Shard Theory

Humans provide an untapped wealth of evidence about alignment

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This post has been recorded as part of the LessWrong Curated Podcast, and can be listened to on Spotify , Apple Podcasts , and Libsyn .

TL;DR: To even consciously consider an alignment research direction, you should have evidence to locate it as a promising lead. As best I can tell, many directions seem interesting but do not have strong evidence of being “entangled” with the alignment problem such that I expect them to yield significant insights.

For example, “we can solve an easier version of the alignment problem by first figuring out how to build an AI which maximizes the number of real-world diamonds” has intuitive appeal and plausibility, but this claim doesn’t *have* to be true and this problem does not *necessarily* have a natural, compact solution. In contrast, there do *in fact* exist humans who care about diamonds. Therefore, there are guaranteed-to-exist alignment insights concerning the way people come to care about e.g. real-world diamonds.

“Consider how humans navigate the alignment subproblem you’re worried about” is a habit which I (TurnTrout) picked up from Quintin Pope. I wrote the post, he originated the tactic.

> A simplified but still very difficult open problem in AI alignment is to state an unbounded program implementing a diamond maximizer that will turn as much of the physical universe into diamond as possible. The goal of "making diamonds" was chosen to have a crisp-seeming definition for our universe (the amount of diamond is the number of carbon atoms covalently bound to four other carbon atoms). If we can crisply define exactly what a 'diamond' is, we can avert issues of trying to convey complex values into the agent.
>
> Ontology identification problem, Arbital

I find this problem interesting, both in terms of wanting to know how to solve a reframed version of it, and in terms of what I used to think about the problem. I used to[1] think, “yeah, ‘diamond’ is relatively easy to define. Nice problem relaxation.” It felt like the diamond maximizer problem let us focus on the challenge of making the AI’s values bind to *something at all* which we actually intended (e.g. diamonds), in a way that’s robust to ontological shifts and that doesn’t collapse into wireheading or tampering with e.g. the sensors used to estimate the number of diamonds.

Although the details are mostly irrelevant to the point of this blog post, the Arbital article suggests some solution ideas and directions for future research, including:

1. Scan AIXI-*tl*’s Turing machines and locate diamonds within their implicit state representations.
2. Given how inaccessible we expect AIXI-*tl*’s representations to be by default, have AIXI-*tl* just consider a Turing-complete hypothesis space which uses more interpretable representations.
3. “Being able to describe, in purely theoretical principle, a prior over epistemic models that have at least two levels and can switch between them in some meaningful sense”

Do you notice anything *strange* about these three ideas? Sure, the ideas don’t seem workable, but they’re good initial thoughts, right?

The problem *isn’t* that the ideas aren’t clever enough. Eliezer is pretty dang clever, and these ideas are reasonable stabs given the premise of “get some AIXI variant to maximize diamond instead of reward.”

The problem *isn’t* that it’s impossible to specify a mind which cares about diamonds. We already know that there are intelligent minds who value diamonds. You might be dating one of them, or you might even *be* one of them! Clearly, the genome + environment jointly specify certain human beings who end up caring about diamonds.

One problem is *where is the evidence required to locate these ideas*? Why should I even find myself thinking about diamond maximization and AIXI and Turing machines and utility functions in this situation? It’s not that there’s *no* evidence. For example, utility functions ensure the agent can’t be exploited in some dumb ways. But I think that the supporting evidence is not *commensurate* with the specificity of these three ideas or with the specificity of the “ontology identification” problem framing.

Here’s an exaggeration of how these ideas feel to me when I read them:

> “I lost my phone”, you tell your friend.
>
> They ask, “Have you checked `Latitude: -34.44006, Longitude: -64.61333`?”
>
> Uneasily, you respond: “Why would I check there?”
>
> Your friend shrugs: “Just seemed promising. And it’s on land, it’s not in the ocean. Don’t worry, I incorporated evidence about where you probably lost it.”

I recently made a similar point about Cooperative Inverse Reinforcement Learning:

> *Against CIRL* as a special case of *against quickly jumping into highly specific speculation while ignoring empirical embodiments-of-the-desired-properties.*
>
> In the context of "how do we build AIs which help people?", asking "does CIRL solve corrigibility?" is hilariously unjustified. By what evidence have we located such a specific question? We have assumed there is an achievable "corrigibility"-like property; we have assumed it is good to have in an AI; we have assumed it is good in a similar way as "helping people"; we have elevated CIRL in particular as a formalism worth inquiring after.
>
> But this is ***not the first question to ask****,* when considering "sometimes people want to help each other, and it'd be great to build an AI which helps us in some way." Much better to start with *existing* generally intelligent systems (humans)

> which *already* sometimes act in the way you want (they help each other) and ask after the ***guaranteed-to-exist reason*** why this empirical phenomenon happens.

Now, if you are confused about a problem, it can be better to explore *some* guesses than no guesses—perhaps it's better to think about Turing machines than to stare helplessly at the wall (but perhaps not). Your best guess may be wrong (e.g. write a utility function which scans Turing machines for atomic representations of diamonds), but you sometimes still learn something by spelling out the implications of your best guess (e.g. the ontology identifier stops working when AIXI Bayes-updates to non-atomic physical theories). This can be productive, as long as you keep in mind the wrongness of the concrete guess, so as to not become anchored on that guess or on the framing which originated it (e.g. build a diamond *maximizer*).

However, in this situation, I want to look elsewhere. When I confront a confusing, difficult problem (e.g. how do you create a mind which cares about diamonds?), I often first look at reality (e.g. are there any existing minds which care about diamonds?). Even if I have *no idea* how to solve the problem, if I can find an existing mind which cares about diamonds, then *since that mind is **real**,* that mind has a *guaranteed-to-exist causal mechanistic play-by-play origin story* for why it cares about diamonds. I thereby anchor my thinking to reality; reality is sturdier than "what if" and "maybe this will work"; many human minds *do* care about diamonds.

In addition to "there's a guaranteed causal story for humans valuing diamonds, and not one for AIXI valuing diamonds", there's a second benefit to understanding how human values bind to the human's beliefs about real-world diamonds. This second benefit is practical: I'm pretty sure the way that *humans* come to care about diamonds has nearly nothing to do with the ways AIXI-*tl* might be motivated to maximize diamonds. This matters, because I expect that the first AGI's value formation will be *far* more mechanistically similar to within-lifetime human value formation, than to AIXI-*tl*'s value alignment dynamics.

Next, it *can* be true that the existing minds are too hard for us to understand in ways relevant to alignment. One way this could be true is that human values are a "mess", that "our brains are kludges slapped together by natural selection." If human value formation *were* sufficiently complex, with sufficiently many load-bearing parts such that each part drastically affects human alignment properties, then we might instead want to design simpler human-comprehensible agents and study *their* alignment properties.

While I think that human *values* are complex, I think the evidence for human value *formation*'s essential complexity is surprisingly weak, all things reconsidered in light of modern, post-deep learning understanding. Still... maybe humans *are* too hard to understand in alignment-relevant ways!

But, I mean, come on. Imagine an alien[2] visited and told you:

> Oh yeah, the AI alignment problem. We knocked that one out a while back. Information inaccessibility of the learned world model? No, I'm pretty sure we didn't solve that, but we didn't have to. We built this protein computer and trained it with, I forget actually, was it just what you would call "deep reinforcement learning"? Hm. Maybe it was more complicated, maybe not, I wasn't involved.

> We *might* have hardcoded relatively crude reward signals that are basically defined over sensory observables, like a circuit which activates when their sensors detect a certain kind of carbohydrate. Scanning you, it looks like some of the protein computers ended up with *your values*, even. Small universe, huh?
>
> Actually, I forgot how we did it, sorry. And I can't make guarantees that our approach scales beyond your intelligence level or across architectures, but maybe it does. I have to go, but here are a few billion of the trained protein computers if you want to check them out!

Ignoring the weird implications of the aliens existing and talking to you like this, and considering only the alignment implications—*The absolute top priority of many alignment researchers should be figuring out how the hell the aliens got as far as they did*.[3] Whether or not you know if their approach scales to further intelligence levels, whether or not their approach seems easy to understand, you have learned that these computers are *physically possible, practically trainable entities*. These computers have definite existence and guaranteed explanations. Next to these actually existent computers, speculation like "maybe attainable utility preservation leads to cautious behavior in AGIs" is dreamlike, unfounded, and untethered.

If it turns out to be currently too hard to understand the aligned protein computers, then I want to keep coming back to the problem with each major new insight I gain. When I learned about scaling laws, I should have rethought my picture of human value formation—Did the new insight knock anything loose? I should have checked back in when I heard about mesa optimizers, about the Bitter Lesson, about the feature universality hypothesis for neural networks, about natural abstractions.

Because, given my life's present ambition (solve AI alignment), that's what it makes sense for me to do—at each major new insight, to reconsider my models[4] of the *single known empirical example of general intelligences with values*, to scour the Earth for every possible scrap of evidence that humans provide about alignment. We may not get much time with human-level AI before we get to superhuman AI. But we get plenty of time with human-level humans, and we get plenty of time *being* a human-level intelligence.

The way I presently see it, the godshatter of human values—the rainbow of desires, from friendship to food—is only unpredictable relative to a class of hypotheses which fail to predict the shattering.[5] But confusion is in the map, not the territory. I do not consider human values to be "unpredictable" or "weird", I do not view them as a "hack" or a "kludge." Human value formation may or may not be messy (although I presently think *not*). Either way, human values are, of course, part of our lawful reality. Human values are reliably produced by within-lifetime processes within the brain. This has an explanation, though I may be ignorant of it. Humans usually bind their values to certain objects in reality, like dogs. This, too, has an explanation.

And, to be clear, I don't want to black-box outside-view extrapolate from the "human datapoint"; I don't want to focus on thoughts like "Since alignment 'works well' for dogs and people, maybe it will work well for slightly superhuman entities." I aspire for the kind of alignment mastery which lets me build a diamond-producing AI, or if that didn't suit my fancy, I'd turn around and tweak the process and the AI would press green buttons forever instead, or—if I were playing for real—I'd align that system of mere circuitry with humane purposes.

For that ambition, the inner workings of those generally intelligent apes is *invaluable evidence* about the *mechanistic within-lifetime process by which those apes form their values,* and, more generally, about how intelligent minds can form values at all. What factors matter for the learned values, what factors don't, and what we should do for AI. Maybe humans have special inductive biases or architectural features, and without those, they'd grow totally different kinds of values. But if that *were* true, wouldn't that be important to know?

If I knew how to interpret the available evidence, I probably *would* understand how I came to weakly care about diamonds, and what factors were important to that process (which reward circuitry had to fire at which frequencies, what concepts I had to have learned in order to grow a value around "diamonds", how precisely activated the reward circuitry had to be in order for me to end up caring about diamonds).

Humans provide huge amounts of evidence, *properly interpreted*—and therein lies the grand challenge upon which I am presently fixated. In an upcoming post, I'll discuss one particularly rich vein of evidence provided by humans. (EDIT 1/1/23: See this shortform comment.)

Thanks to Logan Smith and Charles Foster for feedback. Spiritually related to but technically distinct from The First Sample Gives the Most Information .

EDIT: In this post, I wrote about the Arbital article's unsupported jump from "Build an AI which cares about a simple object like diamonds" to "Let's think about ontology identification for AIXI-*tl.*" The point is not that there is no valid reason to consider the latter, but that the jump, as written, seemed evidence-starved. For *separate* reasons, I currently think that ontology identification is unattractive in some ways, but this post isn't meant to argue against that framing in general. The main point of the post is that humans provide tons of evidence about alignment, by virtue of containing guaranteed-to-exist mechanisms which produce e.g. their values around diamonds.

Appendix: One time I didn't look for the human mechanism

Back in 2018, I had a clever-seeming idea. We don't know how to build an aligned AI; we want multiple tries; it would be great if we could build an AI which "knows it may have been incorrectly designed"; so why not have the AI simulate its probable design environment over many misspecifications, and then *not* do plans which tend to be horrible for most initial conditions. While I drew some inspiration from how I would want to reason in the AI's place, I ultimately did not think thoughts like:

> We know of a single group of intelligent minds who have ever wanted to be corrigible and helpful to each other. I wonder how that, in fact, happens?

Instead, I was trying out clever, off-the-cuff ideas in order to solve e.g. Eliezer's formulation of the hard problem of corrigibility. However, my idea and his formulation suffered a few disadvantages, including:

1. The formulation is not guaranteed to describe a probable or "natural" kind of mind,
2. These kinds of "corrigible" AIs are not guaranteed to produce desirable behavior, but only *imagined* to produce good behavior,

3. My clever-seeming idea was not at all constrained by reality to actually work in practice, as opposed to just sounding clever to me, and
4. I didn’t have a concrete use case in mind for what to *do* with a “corrigible” AI.

I wrote this post as someone who previously needed to read it.

[1] ^

I now think that diamond’s physically crisp definition is a red herring. More on that in future posts.

[2] ^

This alien is written to communicate my current belief state about how human value formation works, so as to make it clear why, *given* my beliefs, this value formation process is so obviously important to understand.

[3] ^

There is an additional implication present in the alien story, but not present in the evolutionary production of humans. The aliens are implied to have *purposefully* aligned some of their protein computers with human values, while evolution is not similarly “purposeful.” This implication is noncentral to the key point, which is that the human-values-having protein computers exist in reality.

[4] ^

Well, I didn’t even *have* a detailed picture of human value formation back in 2021. I thought humans were hopelessly dumb and messy and we want a *nice clean AI which actually is robustly aligned*.

[5] ^

Suppose we model humans as the "inner agent" and evolution as the "outer optimizer"—I think this is, in general, the wrong framing, but let's roll with it for now. I would guess that Eliezer believes that human values are an unpredictable godshatter with respect to the outer criterion of inclusive genetic fitness. This means that if you reroll evolution many times with perturbed initial conditions, you get inner agents with dramatically different values each time—it means that human values are akin to a raindrop which happened to land in some location for no grand reason. I notice that I have medium-strength objections to this claim, but let's just say that he is correct for now.

I think this unpredictability-to-evolution doesn't matter. We aren't going to reroll evolution to get AGI. Thus, for a variety of reasons too expansive for this margin, I am little moved by analogy-based reasoning along the lines of "here's the one time inner alignment was tried in reality, and evolution failed horribly." I think that historical fact is mostly irrelevant, for reasons I will discuss later.

Human values & biases are inaccessible to the genome

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

Related to Steve Byrnes' Social instincts are tricky because of the "symbol grounding problem." I wouldn't have had this insight without several great discussions with Quintin Pope.

TL;DR: It seems hard to scan a trained neural network and locate the AI's learned "tree" abstraction. For very similar reasons, it seems intractable for the genome to scan a human brain and back out the "death" abstraction, which probably will not form at a predictable neural address. Therefore, I infer that the genome can't *directly* make us afraid of death by e.g. specifying circuitry which detects when we think about death and then makes us afraid. In turn, this implies that there are a *lot* of values and biases which the genome cannot hardcode.

In order to understand the human alignment situation confronted by the human genome, consider the AI alignment situation confronted by human civilization. For example, we may want to train a smart AI which learns a sophisticated world model, and then motivate that AI according to its learned world model. Suppose we want to build an AI which intrinsically values trees. Perhaps we can just provide a utility function that queries the learned world model and counts how many trees the AI believes there are.

Suppose that the AI will learn a reasonably human-like concept for "tree." However, before training has begun, the learned world model is inaccessible to us. Perhaps the learned world model will be buried deep within a recurrent policy network, and buried *within* the world model is the "trees" concept. But we have no idea what learned circuits will encode that concept, or how the information will be encoded. We probably can't, in advance of training the AI, write an algorithm which will examine the policy network's hidden state and reliably back out how many trees the AI thinks there are. The AI's learned concept for "tree" is *inaccessible information* from our perspective.

Likewise, the human world model is inaccessible to the human genome, because the world model is probably in the cortex and the cortex is probably randomly initialized.[1] Learned human concepts are therefore inaccessible to the genome, in the same way that the "tree" concept is *a priori* inaccessible to us. Even the broad area where language processing occurs varies from person to person, to say nothing of the encodings and addresses of particular learned concepts like "death."

I'm going to say things like "the genome cannot specify circuitry which detects when a person is thinking about death." This means that the genome cannot hardcode circuitry which e.g. fires when the person is thinking about death, and does not fire when the person is not thinking about death. The genome *does* help indirectly specify the whole adult brain and all its concepts, just like *we* indirectly specify the trained neural network via the training algorithm and the dataset. That doesn't mean we can tell when the AI thinks about trees, and it doesn't mean that the genome can "tell" when the human thinks about death.

When I'd previously thought about human biases (like the sunk cost fallacy) or values (like caring about other people), I had implicitly imagined that genetic influences could directly affect them (e.g. by detecting when I think about helping my friends, and then producing reward). However, given the inaccessibility obstacle, I infer that this can't be the explanation. I infer that the genome *cannot* directly specify circuitry which:

- Detects when you're thinking about seeking power,
- Detects when you're thinking about cheating on your partner,
- Detects whether you perceive a sunk cost,
- Detects whether you think someone is scamming you and, if so, makes you want to punish them,
- Detects whether a decision involves probabilities and, if so, implements the framing effect,
- Detects whether you're thinking about your family,
- Detects whether you're thinking about goals, and makes you conflate terminal and instrumental goals,
- Detects and then navigates ontological shifts,
    - E.g. Suppose you learn that animals are made out of cells. I infer that the genome cannot detect that you are expanding your ontology, and then execute some genetically hard-coded algorithm which helps you do that successfully.
- Detects when you're thinking about wireheading yourself or manipulating your reward signals,
- Detects when you're thinking about reality versus non-reality (like a simulation or fictional world), or
- Detects whether you think someone is higher-status than you.

Conversely, the genome *can* access direct sensory observables, because those observables involve *a priori* -fixed "neural addresses." For example, the genome could hardwire a cute-face-detector which hooks up to retinal ganglion cells (which are at genome-predictable addresses), and then this circuit could produce physiological reactions (like the release of reward). This kind of circuit seems totally fine to me.

In total, information inaccessibility is strong evidence for the genome hardcoding relatively simple[2] cognitive machinery. This, in turn, implies that human values/biases/high-level cognitive observables are produced by relatively simpler hardcoded circuitry, specifying e.g. the learning architecture, the broad reinforcement learning and self-supervised learning systems in the brain, and regional learning hyperparameters. Whereas before it seemed plausible to me that the genome hardcoded a lot of the above bullet points, I now think that's pretty implausible.

When I realized that the genome must also confront the information inaccessibility obstacle, this threw into question a lot of my beliefs about human values, about the complexity of human value formation, and about the structure of my own mind. I was left with a huge puzzle. If we can't say "the hardwired circuitry down the street did it", where do biases come from? How can the genome hook the human's preferences into the human's world model, when the genome doesn't "know" what the world model will look like? Why do people usually navigate ontological shifts properly, why don't they want to wirehead, why do they almost always care about other people *if the genome can't even write circuitry that detects and rewards thoughts about people*?

A fascinating mystery, no? More on that soon.

Thanks to Adam Shimi, Steve Byrnes, Quintin Pope, Charles Foster, Logan Smith, Scott Viteri, and Robert Mastragostino for feedback.

Appendix: The inaccessibility trilemma

The logical structure of this essay is that at least one of the following must be true:

1. Information inaccessibility is somehow a surmountable problem for AI alignment (and the genome surmounted it),
2. The genome solves information inaccessibility in some way we cannot replicate for AI alignment, or
3. The genome cannot directly address the vast majority of interesting human cognitive events, concepts, and properties. (*The point argued by this essay*)

In my opinion, either (1) or (3) would be enormous news for AI alignment. More on (3)'s importance in future essays.

Appendix: Did evolution have advantages in solving the information inaccessibility problem?

Yes, and no. In a sense, evolution had "a lot of tries" but is "dumb", while we have very few tries at AGI while ourselves being able to do consequentialist planning.

In the AI alignment problem, we want to be able to back out an AGI's concepts, but we cannot run lots of similar AGIs and select for AGIs with certain effects on the world. Given the natural abstractions hypothesis, maybe there's a lattice of convergent abstractions—first learn edge detectors, then shape detectors, then people being visually detectable in part as compositions of shapes. And *maybe*, for example, people tend to convergently situate these abstractions in similar relative neural locations: The edge detectors go in V1, then the shape detectors are almost always in some other location, and then the person-concept circuitry is learned elsewhere in a convergently reliable relative position to the edge and shape detectors.

But there's a problem with this story. A congenitally blind person develops dramatically different functional areas, which suggests in particular that their person-concept will be at a radically different relative position than the convergent person-concept location in sighted individuals. Therefore, any genetically hardcoded circuit which checks at the relative address for the person-concept which is reliably situated for sighted people, will not look at the right address for congenitally blind people. Therefore, if this story were true, congenitally blind people would lose any important value-formation effects ensured by this location-checking circuit which detects when they’re thinking about people. So, either the human-concept-location-checking circuit wasn’t an important cause of the blind person caring about other people (and then this circuit hasn’t explained the question we wanted it to, which is how people come to care about other people), or there isn’t such a circuit to begin with. I think the latter is true, and the convergent relative location story is wrong.

But the location-checking circuit is only one way the human-concept-detector could be implemented. There are other possibilities. Therefore, given enough selection and time, maybe evolution could evolve a circuit which checks whether you’re thinking about other people. *Maybe*. But it seems implausible to me (< 4%). I’m going to prioritize explanations for “most people care about other people” which don’t require a fancy workaround.

EDIT: After talking with Richard Ngo, I now think there's about an 8% chance that several interesting mental events are accessed by the genome; I updated upwards from 4%.

EDIT 8/29/22: Updating down to 3%, in part due to 1950's arguments on ethology:

> How do we want to explain the origins of behavior? And [Lehrman's] critique seems to echo some of the concerns with evolutionary psychology. His approach can be gleaned from his example on the pecking behavior of chicks. **Lorenz attributed this behavior to innate forces: The chicks are born with the tendency to peck; it might require just a bit of maturation. Lehrman points out that research by Kuo provides an explanation based on the embryonic development of the chick. The pecking behavior can actually be traced back to movements that developed while the chick was still unhatched. Hardly innate! The main point Lehrman makes: If we claim that something is innate, we stop the scientific investigation without fully understanding the origin of the behavior.** This leaves out important – and fascinating – parts of the explanation because we think we’ve answered the question. As he puts it: **“the statement “It is innate” adds nothing to an understanding of the developmental process involved”**
>
> — Lehrman on Lorenz’s Theory of Instinctive Behavior , blog comment (emphasis added)

1. ^

   Human values can still be inaccessible to the genome even if the cortex isn’t learned from scratch, but learning-from-scratch is a nice and clean sufficient condition which seems likely to me.

2. ^

   I argue that the genome probably hardcodes neural circuitry which is simple *relative* to hardcoded “high-status detector” circuitry. Similarly, the code for a machine learning experiment is simple *relative* to the neural network it trains.

General alignment properties

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

AIXI and the genome are both ways of specifying intelligent agents.

1. Give AIXI a utility function (perhaps over observation histories), and hook it up to an environment, and this pins down a policy.[1]
2. Situate the genome in the embryo within our reality, and this eventually grows into a human being with a policy of their own.

These agents have different "values", in whatever sense we care to consider. However, these two agent-specification procedures also have very different *general alignment properties.*

General alignment properties are not about *what* a particular agent cares about (e.g. the AI "values" chairs). I call an alignment property "general" if the property would be interesting to a range of real-world agents trying to solve AI alignment. Here are some examples.

Terminally valuing latent objects in reality.

AIXI only "terminally values" its observations and doesn't terminally value latent objects in reality, while humans generally care about e.g. dogs (which are latent objects in reality).

Navigating ontological shifts.

Consider latent-diamond-AIXI (LDAIXI), an AIXI variant. LDAIXI's utility function which scans its top 50 hypotheses (represented as Turing machines), checks each work tape for atomic representations of diamonds, and then computes the utility to be the amount of atomic diamond in the world.

If LDAIXI updates sufficiently hard towards non-atomic physical theories, then it can no longer find any utility in its top 50 hypotheses. All policies now might have equal value (zero), and LDAIXI would not continue maximizing the expected diamond content of the future. From our viewpoint, LDAIXI has failed to rebind its "goals" to its new conceptions of reality. (From LDAIXI's "viewpoint", it has Bayes-updated on its observations and continues to select optimal actions.)

On the other hand, physicists do not stop caring about their friends when they learn quantum mechanics. Children do not stop caring about animals when they learn that animals are made out of cells. People seem to navigate ontological shifts pretty well.

Reflective reasoning / embeddedness.

AIXI can't think straight about how it is embedded in the world. However, people quickly learn heuristics like "If I get angry, I'll be more likely to be mean to people around me", or "If I take cocaine now, I'll be even more likely to take cocaine in the future."

Fragility of outcome value to initial conditions / Pairwise misalignment severity

This general alignment property seems important to me, and I'll write a post on it. In short: How pairwise-unaligned are two agents produced with slightly different initial hyperparameters/architectural choices (e.g. reward function / utility function / inductive biases)?

I'm excited about people thinking more about general alignment properties and about what generates those properties.

1. ^

   Supposing e.g. uniformly random tie-breaking for actions enabling equal expected utility.

Evolution is a bad analogy for AGI: inner alignment

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

TL;DR: The dynamics of human learning processes and reward circuitry are more relevant than evolution for understanding how inner values arise from outer optimization criteria.

This post is related to Steve Byrnes' Against evolution as an analogy for how humans will create AGI, but more narrowly focused on how we should make inferences about values.

Thanks to Alex Turner, Charles Foster, and Logan Riggs for their feedback on a draft of this post.

Introduction

How should we expect AGI development to play out?

True precognition appears impossible, so we use various analogies to AGI development, such as evolution, current day humans, or current day machine learning. Such analogies are far from perfect, but we still may be able to extract useful information by carefully examining them.

In particular, we want to understand how inner values relate to the outer optimization criteria. Human evolution is one possible source of data on this question. In this post, I'll argue that human evolution actually provides very little usable evidence on AGI outcomes. In contrast, analogies to the human learning process are much more fruitful.

Inner values versus outer optimization criteria

One way people motivate extreme levels of concern about inner misalignment is to reference the fact that evolution failed to align humans to the objective of maximizing inclusive genetic fitness. From Eliezer Yudkowsky's AGI Ruin post:

> **16**. Even if you train really hard on an exact loss function, that doesn't thereby create an explicit internal representation of the loss function inside an AI that then continues to pursue that exact loss function in distribution-shifted environments. Humans don't explicitly pursue inclusive genetic fitness; **outer optimization even on a very exact, very simple loss function doesn't produce inner optimization in that direction**. This happens *in practice in real life,* it is what happened in *the only case we know about...*

I don't think that "*evolution -> human values*" is the most useful reference class when trying to understand how outer optimization criteria relate to inner values. Evolution didn't directly optimize over our values. It optimized over our learning process and reward circuitry. Once you condition on a particular human's learning process + reward circuitry configuration + the human's environment, you screen off the influence of evolution on that human's values. So, there are really (at least) two classes of observations from which we can draw evidence:

1. "*evolution's inclusive genetic fitness criteria -> a human's learned values*" (as mediated by evolution's influence over the human's learning process + reward circuitry)
2. "*a particular human's learning process + reward circuitry + training environment -> the human's learned values*"

I will present five reasons why I think evidence from (2) "*human learning -> human values*" is more relevant to predicting AGI.

1: Training an AI is more similar to human learning than to evolution

The relationship we want to make inferences about is:

- "*a particular AI's learning process + reward function + training environment -> the AI's learned values*"

I think that "*AI learning -> AI values*" is *much* more similar to "*human learning -> human values*" than it is to "*evolution -> human values*". Steve Byrnes makes this case in much more detail in his post on the matter. Two of the ways I think AI learning more closely resembles human learning, and not evolution, are:

1. The simple type signatures of the two processes. Evolution is a bi-level optimization process, with evolution optimizing over genes, and the genes specifying the human learning process, which *then* optimizes over human cognition. Evolution does not directly optimize over a human's cognition. And because learned cognition is not directly accessible to the genome, evolution must use roundabout methods to influence human values through the genome.

   In contrast, SGD directly optimizes over an AI's cognition, just as human within-lifetime learning directly optimizes over human cognition. The human and AI learning processes are much closer to their respective cognitive structures, compared with evolution.
2. The differences between the parameter counts of the respective objects of optimization (the genome for evolution, the brain's circuitry for human learning, and the AI's parameter's for AI training).

   The genome has very few parameters compared to even current day neural networks, much less the brain or future AGIs. Our experience with ML scaling laws very strongly implies that parameter counts matter a lot for a system's learning dynamics. Better to compare highly parameterized systems to other highly parameterized systems.

"*AI learning -> AI values*", "*human learning -> human values*", and "*evolution -> human values*" each represent very different optimization processes, with many specific dissimilarities between any pair of them. However, I think the balance of dissimilarities points to "*human learning -> human values*" being the closer reference class for "*AI learning -> AI values*". As a result, I think the vast majority of our intuitions regarding the likely outcomes of inner goals versus outer optimization should come from looking at the "*human learning -> human values*" analogy, not the "*evolution -> human values*" analogy.

2: We have more total evidence from human outcomes

Additionally, I think we have a lot more total empirical evidence from "*human learning -> human values*" compared to from "*evolution -> human values*". There are billions of instances of humans, and each of them presumably have somewhat different learning processes / reward circuit configurations / learning environments. Each of them represents a different data point regarding how inner goals relate to outer optimization. In contrast, the human species only evolved once. Thus, evidence from "*human learning -> human values*" should account for even more of our intuitions regarding inner goals versus outer optimization than the difference in reference class similarities alone would indicate.

3: Human learning trajectories represent a broader sampling of the space of possible learning processes

One common objection is that “human learning” represents a tiny region in the space of all possible mind designs, and so we cannot easily generalize our observations of humans to minds in general. This is, of course, true, and it greatly limits the strength of any AI-related conclusions we can draw from looking at "*human learning -> human values*". However, I again hold that inferences from "*evolution -> human values*" suffer from an even more extreme version of this same issue. "*Evolution -> human values*" represent an even more restricted look at the general space of optimization processes than we get from the observed variations in different humans' learning processes, reward circuit configurations, and learning environments.

4: Evidence from humans are more accessible than evidence from evolution

Human evolution happened hundreds of thousands of years ago. We are deeply uncertain about the details of the human ancestral environment and which traits were under what selection pressure. We are still unsure about what precise selection pressure led humans to be so generally intelligent at all. We are very far away from being able to precisely quantify all the potentially values-related selection pressures in the ancestral environment, or how those selection pressures changed our reward systems or our tendencies to form downstream values.

In contrast, human within lifetime learning happens all the time right now. It’s available for analysis and even experimental intervention. Given two evidence sources about a given phenomenon, where one evidence source is much more easily accessible than the other, then all else equal, the more accessible evidence source should represent a greater fraction of our total information on the phenomenon. This is another reason why we should expect evidence from humans to account for a greater proportion of our total information about how inner values relate to outer optimization criteria.

5: Evolution could not have succeeded anyways

I think that a careful account of how evolution shaped our learning process in the ancestral environment implies that evolution had next to no chance of aligning humans with inclusive genetic fitness.

There are no features of the ancestral environment which would lead to an ancestral human learning about the abstract idea of inclusive genetic fitness. There were no ancestral humans that held an explicit representation of inclusive genetic fitness. So, there was never an opportunity for evolution to select for humans who attached their values to an explicit representation of inclusive genetic fitness.

Regardless of how difficult it is, in general, to get learning systems to form values around different abstract concepts, evolution could not have possibly gotten us to form a value around the particular abstraction of inclusive genetic fitness because we didn't form such an abstraction in the ancestral environment. Ancestral humans had zero variance in their tendency to form values around inclusive genetic fitness. Evolution cannot select for traits that don't vary across a population, so evolution could not have selected for humans that formed their values around inclusive genetic fitness.

In contrast, the sorts of things that we humans end up valuing are usually the sorts of things that are easy to form abstractions around. Thus, we are not doomed by the same difficulty that likely prevented evolution from aligning humans to inclusive genetic fitness.

This point is extremely important. I want to make sure to convey it correctly, so I will quote two previous expressions of this point by other sources:

Risks from Learned Optimization notes that the lack of environmental data related to inclusive genetic fitness effectively increases the description length complexity of specifying an intelligence that deliberately optimizes for inclusive genetic fitness:

> …description cost is especially high if the learned algorithm's input data does not contain easy-to-infer information about how to optimize for the base objective. Biological evolution seems to differ from machine learning in this sense, since evolution's specification of the brain has to go through the information funnel of DNA. The sensory data that early humans received didn't allow them to infer the existence of DNA, nor the relationship between their actions and their genetic fitness. Therefore, for humans to have been aligned with evolution would have required them to have an innately specified model of DNA, as well as the various factors influencing their inclusive genetic fitness. Such a model would not have been able to make use of environmental information for compression, and thus would have required a greater description length. In contrast, our models of food, pain, etc. can be very short since they are directly related to our input data.

From Alex Turner (in private communication):

> If values form because reward sends reinforcement flowing back through a person's cognition and reinforces the thoughts which (credit assignment judges to have) led to the reward, then if a person never thinks about inclusive reproductive fitness, they can *never ever* form a value shard around inclusive reproductive fitness. Certain abstractions, like lollipops or people, are convergently learned early in the predictive-loss-minimization process and thus are easy to form values around. But if there aren't local mutations which make a person more probable to think thoughts about inclusive genetic fitness before/while the person gets

reward, then evolution can't instill this value. Even if the descendents of that person will later be *able* to think thoughts about fitness.

Total significance of evolution

There are many sources of empirical evidence that can inform our intuitions regarding how inner goals relate to outer optimization criteria. My current (not very deeply considered) estimate of how to weight these evidence sources is roughly:

- ~60% from "*human learning -> human values*"
- ~4% from "*evolution -> human values*"
- ~36% from various other evidence sources, which I won't address further in this post, such as:
    - economics
    - microbial ecology
    - politics
    - current results in machine learning
    - game theory / multi-agent negotiation dynamics

Implications

I think that using "*human learning -> human values*" as our reference class for inner goals versus outer optimization criteria suggests a much more straightforward relationship between the two, as compared to the (lack of a) relationship suggested by "*evolution -> human values*". Looking at the learning trajectories of individual humans, it seems like a given person's values have a great deal in common with the sorts of experiences they've found rewarding in their lives up to that point in time. E.g., a person who grew up with and displayed affection for dogs probably doesn't want a future totally devoid of dogs, or one in which dogs suffer greatly.

Please note that I am not arguing that humans are inner aligned, or that looking at humans implies inner alignment is easy. Humans are misaligned with maximizing their outer reward source (activation of reward circuitry). I operationalize this misalignment as: "A*fter a distributional shift from their learning environment, humans frequently behave in a manner that predictably fails to maximize reward in their new environment, specifically because they continue to implement values they'd acquired from their learning environment which are misaligned to reward maximization in the new environment*".

For example, one way in which humans are inner misaligned is that, if you introduce a human into a new environment which has a button that will wirehead the human (thus maximizing reward in the new environment), but has other consequences that are extremely bad by light of the human's preexisting values (e.g., killing a beloved family member), most humans won't push the button.

I also think this regularity in inner values is reasonably robust to large increases in capabilities. If you take a human whose outer behavior suggests they like dogs, and give that human very strong capabilities to influence the future, I do not think they are at all likely to erase dogs from existence. It's probably not as robust to your choice of which specific human to try this with. E.g., many people would screw themselves over with reckless self-modification. My point is that higher capabilities *alone* do not automatically render inner values completely alien to those demonstrated at lower capabilities.

(Part 2 will address whether the “sharp left turn” demonstrated by human capabilities with respect to evolution implies that we should expect a similar sharp left turn in AI capabilities.)

Reward is not the optimization target

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

This insight was made possible by many conversations with Quintin Pope, where he challenged my implicit assumptions about alignment. I'm not sure who came up with this particular idea.

In this essay, I call an agent a "reward optimizer" if it not only gets lots of reward, but if it reliably makes choices like "reward but no task completion" (e.g. receiving reward without eating pizza) over "task completion but no reward" (e.g. eating pizza without receiving reward). Under this definition, an agent can be a reward optimizer even if it doesn't contain an explicit representation of reward, or implement a search process for reward.

> Reinforcement learning is learning what to do—how to map situations to actions **so as to maximize a numerical reward signal**. — Reinforcement learning: An introduction

Many people[1] seem to expect that reward will be the optimization target of really smart learned policies—that these policies will be reward optimizers. I strongly disagree. As I argue in this essay, reward is *not*, in general, that-which-is-optimized by RL agents.[2]

Separately, as far as I can tell, most[3] practitioners usually view reward as encoding the relative utilities of states and actions (e.g. it's *this good* to have all the trash put away), as opposed to imposing a *reinforcement schedule* which builds certain computational edifices inside the model (e.g. reward for picking up trash → reinforce trash-recognition and trash-seeking and trash-putting-away subroutines). I think the former view is usually inappropriate, because in many setups, **reward *chisels cognitive grooves into an agent*.**

Therefore, *reward is not the optimization target* in two senses:

1. Deep reinforcement learning agents will not come to intrinsically and primarily value their reward signal; reward is not *the trained agent's* optimization target.
2. Utility functions express the *relative goodness* of outcomes. Reward *is not best understood* as being a kind of utility function. Reward has the mechanistic effect of *chiseling cognition into the agent's network*. Therefore, properly understood, reward does not express relative goodness and is therefore *not an optimization target at all.*

Reward probably won't be a deep RL agent's primary optimization target

After work, you grab pizza with your friends. You eat a bite. The taste releases reward in your brain, which triggers credit assignment. Credit assignment identifies which thoughts and decisions were responsible for the release of that reward, and makes those decisions more likely to happen in similar situations in the future. Perhaps you had thoughts like

- “It’ll be fun to hang out with my friends” and
- “The pizza shop is nearby” and
- “Since I just ordered food at a cash register, execute `motor-subroutine-#51241` to take out my wallet” and
- “If the pizza is in front of me and it’s mine and I’m hungry, raise the slice to my mouth” and
- “If the slice is near my mouth and I’m not already chewing, take a bite.”

Many of these thoughts will be judged responsible by credit assignment, and thereby become more likely to trigger in the future. This is what *reinforcement* learning is all about—the reward is the *reinforcer* of those things which came before it and the *creator* of new lines of cognition entirely (e.g. anglicized as "I shouldn't buy pizza when I'm mostly full"). The reward chisels cognition which increases the probability of the reward accruing next time.

Importantly, reward does not automatically spawn thoughts *about* reward, and reinforce those reward-focused thoughts! Just because common English endows “reward” with suggestive pleasurable connotations, that does not mean that an RL agent will *terminally value* reward!

What kinds of people (or non-tabular agents more generally) will become reward optimizers, such that the agent ends up terminally caring about reward (and little else)? Reconsider the pizza situation, but instead suppose you were thinking thoughts like “this pizza is going to be so rewarding” and “in this situation, eating pizza sure will activate my reward circuitry.”

You eat the pizza, triggering reward, triggering credit assignment, which correctly locates these reward-focused thoughts as contributing to the release of reward. Therefore, in the future, you will more often take actions because you think they will produce reward, and so you will become more of the kind of person who intrinsically cares about reward. This is a path[4] to reward-optimization and wireheading.

While it's possible to have activations on "pizza consumption predicted to be rewarding" and "execute `motor-subroutine-#51241` " and then have credit assignment hook these up into a new motivational circuit, **this is only *one possible direction* of value formation in the agent** . Seemingly, the most direct way for an agent to become *more* of a reward optimizer is to *already* make decisions motivated by reward, and then have credit assignment further generalize that decision-making.

The siren-like suggestiveness of the word “reward”

Let’s strip away the suggestive word “reward”, and replace it by its substance: cognition-updater.

Suppose a human trains an RL agent by pressing the cognition-updater button when the agent puts trash in a trash can. While putting trash away, the AI’s policy network is probably “thinking about”[5] the *actual world it’s interacting with*, and so the cognition-updater reinforces those heuristics which lead to the trash getting put away (e.g. “if trash-classifier activates near center-of-visual-field, then grab trash using `motor-subroutine-#642`”).

Then suppose this AI models the true fact that the button-pressing produces the cognition-updater. Suppose this AI, which has historically had its trash-related thoughts reinforced, considers the plan of pressing this button. “If I press the button, that triggers credit assignment, which will reinforce my decision to press the button, such that in the future I will press the button even more.”

Why, exactly, would the AI seize [6] *the button? To reinforce itself into a certain corner of its policy space? The AI has not had antecedent-computation-reinforcer-thoughts reinforced in the past, and so its current decision will not be made in order to acquire the cognition-updater!*

RL is not, in general, about training cognition-updater optimizers.

When *is* reward the optimization target of the agent?

If reward is guaranteed to become your optimization target, then your learning algorithm can force you to become a drug addict. Let me explain.

Convergence theorems provide conditions under which a reinforcement learning algorithm is guaranteed to converge to an optimal policy for a reward function. For example, value iteration maintains a table of value estimates for each state *s*, and iteratively propagates information about that value to the neighbors of *s*. If a far-away state *f* has huge reward, then that reward ripples back through the environmental dynamics via this “backup” operation. Nearby parents of *f* gain value, and then after lots of backups, far-away ancestor-states gain value due to *f*’s high reward.

Eventually, the “value ripples” settle down. The agent picks an (optimal) policy by acting to maximize the value-estimates for its post-action states.

Suppose it would be extremely rewarding to do drugs, but those drugs are on the other side of the world. Value iteration backs up that high value to your present space-time location, such that your policy necessarily gets *at least* that much reward. There’s no escaping it: After enough backup steps, you’re traveling across the world to do cocaine.

But obviously these conditions aren’t true in the real world. Your learning algorithm doesn’t force *you* to try drugs. Any AI which e.g. tried every action at least once would quickly kill itself, and so real-world general RL agents won’t explore like that because that would be stupid. So the RL agent’s algorithm won’t make it e.g. explore wireheading either, and so the convergence theorems *don’t apply even a little—even in spirit*.

Anticipated questions

1. Why won’t early-stage agents think thoughts like “If putting trash away will lead to reward, then execute `motor-subroutine-#642`”, and then this gets reinforced into reward-focused cognition early on?

        1. Suppose the agent puts away trash in a blue room. Why won’t early-stage agents think thoughts like “If putting trash away will lead to the wall being blue, then execute `motor-subroutine-#642`”, and then this gets reinforced into blue-wall-focused cognition early on? Why consider either scenario to begin with?
2. But aren’t we implicitly selecting for agents with high cumulative reward, when we train those agents?
    1. Yeah. But on its own, this argument can’t possibly imply that selected agents will probably be reward optimizers. The argument would prove too much. Evolution selected for inclusive genetic fitness, and it did not get IGF optimizers.
        1. "We're selecting for agents on reward → we get an agent which optimizes reward" is locally invalid. "We select for agents on X → we get an agent which optimizes X" is not true for the case of evolution, and so is not true in general.
        2. Therefore, the argument isn't necessarily true in the AI reward-selection case. Even if RL *did* happen to train reward optimizers and this post *were* wrong, the selection argument is too weak on its own to establish that conclusion.
    2. Here’s the more concrete response: Selection isn’t *just* for agents which get lots of reward.
        1. For simplicity, consider the case where on the training distribution, the agent gets reward if and only if it reaches a goal state. Then any selection for reward is also selection for reaching the goal. And if the goal is the only red object, then selection for reward is *also* selection for reaching red objects.
        2. In general, selection for reward produces equally strong selection for reward’s necessary and sufficient conditions. In general, it seems like there should be a lot of those. Therefore, since selection is not only for *reward* but for *anything which goes along with reward* (e.g. reaching the goal), then selection won’t advantage *reward optimizers* over *agents which reach goals quickly / pick up lots of trash / [do the objective]*.
    3. Another reason to not expect the selection argument to work is that it’s *instrumentally convergent* for most inner agent values to *not* become wireheaders, for them to *not* try hitting the reward button.
        1. I think that before the agent can hit the particular attractor of reward-optimization, it will hit an attractor in which it optimizes for some aspect of a historical correlate of reward.
            1. We train agents which intelligently optimize for e.g. putting trash away, and this reinforces the trash-putting-away computations, which activate in a broad range of situations so as to steer agents into a future where trash has been put away. An intelligent agent will model the true fact that, if the agent reinforces itself into caring about cognition-updating, then it will no longer navigate to futures where trash is put away. Therefore, it decides to not hit the reward button.
            2. This reasoning follows for most inner goals by instrumental convergence.
        2. On my current best model, this is why people usually don’t wirehead. They learn their own values via deep RL, like caring about dogs, and

        these actual values are opposed to the person they would become if they wirehead.

3. Don’t some people terminally care about reward?
    1. I think so! I think that generally intelligent RL agents will have *secondary, relatively weaker* values around reward, but that reward will not be a primary motivator. Under my current (weakly held) model, an AI will only start chiseled computations about reward *after* it has chiseled other kinds of computations (e.g. putting away trash). More on this in later essays.
4. But what if the AI bops the reward button early in training, while exploring? Then credit assignment would make the AI more likely to hit the button again.
    1. Then keep the button away from the AI until it can model the effects of hitting the cognition-updater button.[7]
    2. For the reasons given in the “siren” section, a sufficiently reflective AI probably won’t seek the reward button on its own.
5. AIXI—
    1. will always kill you and then wirehead forever, unless you gave it something like a constant reward function.
    2. And, IMO, this fact is not practically relevant to alignment. AIXI is *explicitly a reward-<u>maximizer</u>*. As far as I know, AIXI(-*tl*) is not the limiting form of any kind of real-world intelligence trained via *reinforcement* learning.
6. Does the choice of RL algorithm matter?
    1. For point 1 (*reward is not the trained agent's optimization target*), it might matter.
        1. I started off analyzing model-free actor-based approaches, but have also considered a few model-based setups. I think the key lessons apply to the general case, but I think the setup will substantially affect which values tend to be grown.
            1. If the agent's curriculum is broad, then reward-based cognition may get reinforced from a confluence of tasks (solve mazes, write sonnets), while each task-specific cognitive structure is only narrowly contextually reinforced. That said, this is also selecting equally hard for agents which do the rewarded activities, and reward-motivation is only one possible value which produces those decisions.
            2. Pretraining a language model and then slotting that into an RL setup also changes the initial computations in a way which I have not yet tried to analyze.
        2. It’s *possible* there’s some kind of RL algorithm which *does* train agents which limit to reward optimization (and, of course, thereby “solves” inner alignment in its literal form of “find a policy which optimizes the outer objective signal”).
    2. For point 2 (*reward provides local updates to the agent's cognition via credit assignment; reward is not best understood as specifying our preferences*), the choice of RL algorithm should not matter, as long as it uses reward to compute local updates.
        1. A similar lesson applies to the updates provided by loss signals. A loss signal provides updates which deform the agent's cognition into a new shape.
7. TurnTrout, you've been talking about an AI's learning process using English, but ML gradients may not neatly be expressible in our concepts. How do we know that it's appropriate to speculate in English?
    1. I am *not certain* that my model is legit, but it sure seems more legit than (my perception of) how people usually think about RL (i.e. in terms of

reward maximization, and reward-as-optimization-target instead of as feedback signal which builds cognitive structures).
2. I only have access to my own concepts and words, so I am provisionally reasoning ahead anyways, while keeping in mind the potential treacheries of anglicizing imaginary gradient updates (e.g. "be more likely to eat pizza in similar situations").

Dropping the old hypothesis

At this point, I don't see a strong reason to focus on the “reward optimizer” hypothesis. The idea that AIs will get really smart and primarily optimize some reward signal… I don’t know of any tight mechanistic stories for that. I’d love to hear some, if there are any.

As far as I’m aware, the strongest evidence left for agents intrinsically valuing cognition-updating is that some humans *do* strongly (but not uniquely) value cognition-updating,[8] and many humans seem to value it weakly, and humans are probably RL agents in the appropriate ways. So we definitely can’t *rule out* agents which strongly (and not just weakly) value the cognition-updater. But it’s also *not* the overdetermined default outcome. More on that in future essays.

It’s true that reward *can* be an agent’s optimization target, but what reward *actually does* is reinforce computations which lead to it. A particular alignment proposal might argue that a reward function will *reinforce the agent into a shape such that it intrinsically values reinforcement*, and that the *cognition-updater goal is also a human-aligned optimization target*, but this is still just one particular approach of using the cognition-updating to produce desirable cognition within an agent. Even in that proposal, the primary mechanistic function of reward is reinforcement, not optimization-target.

Implications

Here are some major updates which I made:

1. **Any reasoning derived from the reward-optimization premise is now suspect until otherwise supported.**
2. **Wireheading was never a high-probability problem for RL-*trained* agents**, absent a specific story for why cognition-updater-acquiring thoughts would be chiseled into primary decision factors.
3. **Stop worrying about finding “outer objectives” which are safe to *maximize.***[9] I think that you’re not going to get an outer-objective-maximizer (i.e. an agent which maximizes the explicitly specified reward function).
    1. Instead, focus on building good cognition within the agent.
    2. In my ontology, there's only one question: How do we grow good cognition inside of the trained agent?
4. **Mechanistically model RL agents as executing behaviors downstream of past reinforcement** (e.g. putting trash away), in addition to thinking about policies which are selected for having high reward on the training distribution (e.g. hitting the button).

1. The latter form of reasoning skips past the mechanistic substance of reinforcement learning: The chiseling of computations responsible for the acquisition of the cognition-updater. I still think it's useful to consider selection, but mostly in order to generate failures modes whose mechanistic plausibility can be evaluated.
2. In my view, reward's proper role isn't to encode an objective, but a *reinforcement schedule*, such that the right kinds of computations get reinforced within the AI's mind.

Edit 11/15/22: The original version of this post talked about how reward reinforces antecedent computations in policy gradient approaches. This is not true in general. I edited the post to instead talk about how reward is used to upweight certain kinds of actions in certain kinds of situations, and therefore reward *chisels cognitive grooves into agents*.

Appendix: The field of RL thinks reward=optimization target

Let’s take a little stroll through Google Scholar’s top results for “reinforcement learning", emphasis added:

> The agent's job is to find a policy… that **maximizes some long-run measure of reinforcement**. ~ Reinforcement learning: A survey

> In instrumental conditioning, animals learn to choose actions to obtain rewards and avoid punishments, or, more generally to achieve goals. **Various goals are possible, such as optimizing the average rate of acquisition of net rewards (i.e. rewards minus punishments), or some proxy for this such as the expected sum of future rewards**. ~ Reinforcement learning: The Good, The Bad and The Ugly

Steve Byrnes did, in fact, briefly point out part of the “reward is the optimization target” mistake:

> I note that even experts sometimes sloppily talk as if RL agents make plans towards the goal of maximizing future reward… — Model-based RL, Desires, Brains, Wireheading

I don't think it's just sloppy talk, I think it's incorrect belief in many cases. I mean, I did my PhD on RL theory, and I still believed it. Many authorities and textbooks confidently claim—presenting little to no evidence—that reward is an optimization target (i.e. the quantity which the policy is in fact trying to optimize, or the quantity to be optimized by the policy). Check what the math actually says.

1. ^

   Including the authors of the quoted introductory text, Reinforcement learning: An introduction. I have, however, met several alignment researchers who already internalized that reward is not the optimization target, perhaps not in so many words.

2. ^

Utility ≠ Reward points out that an RL-trained agent is *optimized by* original reward, but not necessarily *optimizing for* the original reward. This essay goes further in several ways, including when it argues that *reward* and *utility* have different type signatures—that reward shouldn't be viewed as encoding a goal at all, but rather a *reinforcement schedule*. And not only do I not expect the trained agents to not maximize the original "outer" reward signal, I think they probably won't try to strongly optimize *any* reward signal.

3. ^

   Reward shaping seems like the most prominent counterexample to the "reward represents terminal preferences over state-action pairs" line of thinking.

4. ^

   But also, you were still probably thinking about reality as you interacted with it ("since I'm in front of the shop where I want to buy food, go inside"), and credit assignment will still locate some of those thoughts as relevant, and so you wouldn't purely reinforce the reward-focused computations.

5. ^

   "Reward reinforces existing thoughts" is ultimately a claim about how updates depend on the existing weights of the network. I think that it's easier to update cognition along the lines of existing abstractions and lines of reasoning. If you're already running away from wolves, then if you see a bear and become afraid, you can be updated to run away from large furry animals. This would leverage your *existing* concepts.

   From A shot at the diamond-alignment problem:

   > The local mapping from gradient directions to behaviors is given by the neural tangent kernel, and the learnability of different behaviors is given by the NTK's eigenspectrum, which seems to adapt to the task at hand, making the network quicker to learn along behavioral dimensions similar to those it has already acquired.

6. ^

   Quintin Pope remarks: "The AI would probably want to establish **control** over the button, if only to ensure its values aren't updated in a way it wouldn't endorse. Though that's an example of convergent powerseeking, not reward seeking."

7. ^

   For mechanistically similar reasons, keep cocaine out of the crib until your children can model the consequences of addiction.

8. ^

   I am presently ignorant of the relationship between pleasure and reward prediction error in the brain. I do not think they are the same.

   However, I think people are usually weakly hedonically / experientially motivated. Consider a person about to eat pizza. If you give them the choice between "pizza but no pleasure from eating it" and "pleasure but no pizza", I think most people would choose the latter (unless they were really hungry and needed the calories). If people just navigated to futures where they had eaten pizza, that would not be true.

9. ^

   From correspondence with another researcher: There may yet be an interesting alignment-related puzzle to "Find an optimization process whose maxima are friendly", but I personally don't share the intuition yet.

The shard theory of human values

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

TL;DR: We propose a theory of human value formation. According to this theory, the reward system shapes human values in a relatively straightforward manner. Human values are not e.g. an incredibly complicated, genetically hard-coded set of drives, but rather sets of contextually activated heuristics which were shaped by and bootstrapped from crude, genetically hard-coded reward circuitry.

We think that human value formation is extremely important for AI alignment. We have empirically observed exactly one process which reliably produces agents which intrinsically care about certain objects in the real world, which reflect upon their values and change them over time, and which—at least some of the time, with non-negligible probability—care about each other. That process occurs millions of times each day, despite genetic variation, cultural differences, and disparity in life experiences. That process produced you and your values.

Human values *look so strange and inexplicable*. How could those values be the product of anything except hack after evolutionary hack? We think this is *not* what happened. This post describes the shard theory account of human value formation, split into three sections:

1. Details our working assumptions about the learning dynamics within the brain,
2. Conjectures that reinforcement learning grows situational heuristics of increasing complexity, and
3. Uses shard theory to explain several confusing / "irrational" quirks of human decision-making.

Terminological note: We use "value" to mean *a contextual influence on decision-making*. Examples:

- Wanting to hang out with a friend.
- Feeling an internal urge to give money to a homeless person.
- Feeling an internal urge to text someone you have a crush on.
- That tug you feel when you are hungry and pass by a donut.

To us, this definition seems importantly type-correct and appropriate—see Appendix A.2. The main downside is that the definition is relatively broad—most people wouldn't list "donuts" among their "values." To avoid this counterintuitiveness, we would refer to a "donut shard" instead of a "donut value." ("Shard" and associated terminology are defined in section II.)

I. Neuroscientific assumptions

The shard theory of human values makes three main assumptions. We think each assumption is pretty mainstream and reasonable. (For pointers to relevant literature supporting these assumptions, see Appendix A.3.)

Assumption 1: The cortex[1] is basically (locally) randomly initialized. According to this assumption, most of the circuits in the brain are learned from scratch, in the sense of being mostly randomly initialized and not mostly genetically hard-coded. While the high-level topology of the brain may be genetically determined, we think that the local connectivity is not primarily genetically determined. For more clarification, see [Intro to brain-like-AGI safety] 2. "Learning from scratch" in the brain.

Thus, we infer that human values & biases are inaccessible to the genome:

> It seems hard to scan a trained neural network and locate the AI's learned "tree" abstraction. For very similar reasons, it seems intractable for the genome to scan a human brain and back out the "death" abstraction, which probably will not form at a predictable neural address. Therefore, we infer that the genome can't *directly* make us afraid of death by e.g. specifying circuitry which detects when we think about death and then makes us afraid. In turn, this implies that there are a *lot* of values and biases which the genome cannot hardcode…
>
> [This leaves us with] a huge puzzle. If we can't say "the hardwired circuitry down the street did it", where do biases come from? How can the genome hook the human's preferences into the human's world model, when the genome doesn't "know" what the world model will look like? Why do people usually navigate ontological shifts properly, why don't people want to wirehead, why do people almost always care about other people *if the genome can't even write circuitry that detects and rewards thoughts about people*?".

Assumption 2: The brain does self-supervised learning. According to this assumption, the brain is constantly predicting what it will next experience and think, from whether a V1 neuron will detect an edge, to whether you're about to recognize your friend Bill (which grounds out as predicting the activations of higher-level cortical representations). (See *On Intelligence* for a book-long treatment of this assumption.)

In other words, the brain engages in self-supervised predictive learning: Predict what happens next, then see what actually happened, and update to do better next time.

Definition. Consider the context available to a circuit within the brain. Any given circuit is innervated by axons from different parts of the brain. These axons transmit information to the circuit. Therefore, whether a circuit fires is not primarily dependent on the external situation navigated by the human, or even what the person senses at a given point in time. A circuit fires depending on whether its inputs[2]—the *mental context*—triggers it or not. This is what the "context" of a shard refers to.

Assumption 3: The brain does reinforcement learning. According to this assumption, the brain has a genetically hard-coded reward system (implemented via certain hard-coded circuits in the brainstem and midbrain). In some[3] fashion, the brain reinforces thoughts and mental subroutines which have led to reward, so that they will be more likely to fire in similar contexts in the future. We suspect that the "base" reinforcement learning algorithm is relatively crude, but that people reliably bootstrap up to smarter credit assignment.

Summary. Under our assumptions, most of the human brain is locally randomly initialized. The brain has two main learning objectives: self-supervised predictive loss (we view this as building your world model; see Appendix A.1) and reward (we view this as building your values, as we are about to explore).

II. Reinforcement events shape human value shards

This section lays out a bunch of highly specific mechanistic speculation about how a simple value might form in a baby's brain. For brevity, we won't hedge statements like "the baby is reinforced for X." We think the story is good and useful, but don't mean to communicate absolute confidence via our unhedged language.

Given the inaccessibility of world model concepts, how does the genetically hard-coded reward system dispense reward in the appropriate mental situations? For example, suppose you send a drunk text, and later feel embarrassed, and this triggers a penalty. How is that penalty calculated? By information inaccessibility and the absence of text messages in the ancestral environment, the genome *isn't* directly hard-coding a circuit which detects that you sent an embarrassing text and then penalizes you. Nonetheless, such embarrassment seems to trigger (negative) reinforcement events... and we don't really understand how that works yet.

Instead, let's model what happens if the genome hardcodes a sugar-detecting reward circuit. For the sake of this section, suppose that the genome specifies a reward circuit which takes as input the state of the taste buds and the person's metabolic needs, and produces a reward if the taste buds indicate the presence of sugar while the person is hungry. By assumption 3 in section I, the brain does reinforcement learning and credit assignment to reinforce circuits and computations which led to reward. For example, if a baby picks up a pouch of apple juice and sips some, that leads to sugar-reward. The reward makes the baby more likely to pick up apple juice in similar situations in the future.

Therefore, a baby may learn to sip apple juice which is already within easy reach. However, without a world model (much less a *planning process*), the baby cannot learn multi-step plans to grab and sip juice. If the baby doesn't have a world model, then she won't be able to act differently in situations where there is or is not juice behind her. Therefore, the baby develops a set of shallow situational heuristics which involve sensory preconditions like "IF juice pouch detected in center of visual field, THEN move arm towards pouch." The baby is basically a trained reflex agent.

However, when the baby has a proto-world model, the reinforcement learning process takes advantage of that new machinery by further developing the juice-tasting heuristics. Suppose the baby models the room as containing juice within reach but out of sight. Then, the baby *happens* to turn around, which activates the *already-trained* reflex heuristic of "grab and drink juice you see in front of you." In this scenario, "turn around to see the juice" preceded execution of "grab and drink the juice which is in front of me", and so the baby is reinforced for turning around to grab the juice in situations where the baby models the juice as behind herself.[4]

By this process, repeated many times, the baby learns how to associate world model concepts (e.g. "the juice is behind me") with the heuristics responsible for reward (e.g. "turn around" and "grab and drink the juice which is in front of me"). Both parts of that sequence are reinforced. In this way, the contextual-heuristics exchange information with the budding world model.

A *shard of value* refers to the contextually activated computations which are downstream of similar historical reinforcement events. For example, the juice-shard consists of the various decision-making influences which steer the baby towards the historical reinforcer of a juice pouch. These contextual influences were all reinforced into existence by the activation of sugar reward circuitry upon drinking juice. A *subshard* is a contextually activated component of a shard. For example, "IF juice pouch in front of me THEN grab" is a *subshard* of the juice-shard. It seems plain to us that learned value shards are[5] most strongly activated in the situations in which they were historically reinforced and strengthened. (For more on terminology, see Appendix A.2.)

Generated by DALL-E 2.

While all of this is happening, many different shards of value are also growing, since the human reward system offers a range of feedback signals. Many subroutines are being learned, many heuristics are developing, and many proto-preferences are taking root. At this point, the brain learns a crude planning algorithm,[6] because proto-planning subshards (e.g. IF `motor-command-5214` predicted to bring a juice pouch into view, THEN execute) would be reinforced for their contributions to activating the various hardcoded reward circuits. This proto-planning is learnable because most of the machinery was already developed by the self-supervised predictive learning, when e.g. learning to predict the consequences of motor commands (see Appendix A.1).

The planner has to decide on a coherent plan of action. That is, micro-incoherences (turn towards juice, but then turn back towards a friendly adult, but then turn back towards the juice, ad nauseum) should generally be penalized away.[7] Somehow, the plan has to be coherent, integrating several conflicting shards. We find it useful to view this integrative process as a kind of “bidding.” For example, when the juice-shard activates, the shard fires in a way which would have historically increased the probability of executing plans which led to juice pouches. We’ll say that the juice-shard is *bidding* for plans which involve juice consumption (according to the world model), and perhaps bidding against plans without juice consumption.

Importantly, however, the juice-shard is shaped to bid for plans which the world model predicts actually lead to *juice being consumed*, and not necessarily for plans which lead to *sugar-reward-circuit activation***.** You might wonder: “Why wouldn’t the shard learn to value reward circuit activation?”. The *effect* of drinking juice is that the baby's credit assignment reinforces the computations which were causally responsible for producing the situation in which the hardcoded sugar-reward circuitry fired.

But *what* is reinforced? The *content* of the responsible computations includes a sequence of heuristics and decisions, one of which involved the juice pouch abstraction in the world model. Those are the circuits which actually get reinforced and become more likely to fire in the future. Therefore, the juice-heuristics get reinforced. The heuristics coalesce into a so-called shard of value as they query the world model and planner to implement increasingly complex multi-step plans.

In contrast, in this situation, the baby's decision-making does not involve “if this action is predicted to lead to sugar-reward, then bid for the action.” This non-participating heuristic probably won’t be reinforced or created, much less become a shard of value.[8]

This is important. We see how the reward system shapes our values, without our values entirely binding to the activation of the reward system itself. We have also laid bare the manner in which the juice-shard is bound to your *model of reality* instead of simply *your* *model of future perception*. Looking back across the causal history of the juice-shard's training, the shard has no particular reason to bid for the plan "stick a wire in my brain to electrically stimulate the sugar reward-circuit", even if the world model correctly predicts the consequences of such a plan. In fact, a good world model predicts that the person will drink *fewer* juice pouches after becoming a wireheader, and so the juice-shard in a reflective juice-liking adult *bids against* the wireheading plan! *Humans are not reward-maximizers, they are value shard-executors.*

This, we claim, is one reason why people (usually) don't want to wirehead and why people often want to avoid value drift. According to the sophisticated reflective capabilities of your world model, if you popped a pill which made you 10% more okay with murder, your world model predicts futures which are bid against by your current shards because they contain too much murder.

We're pretty confident that the reward circuitry is not a complicated hard-coded morass of alignment magic which forces the human to care about real-world juice. No, the hypothetical sugar-reward circuitry is simple. We conjecture that the order in which the brain learns abstractions makes it convergent to care about certain objects in the real world.

III. Explaining human behavior using shard theory

The juice-shard formation story is simple and—if we did our job as authors—easy to understand. However, juice-consumption is hardly a prototypical human value. In this section, we'll show how shard theory neatly explains a range of human behaviors and preferences.

As people, we have lots of intuitions about human behavior. However, intuitively obvious behaviors *still have to have mechanistic explanations*—such behaviors still have to be retrodicted by a correct theory of human value formation*.* While reading the following examples, try looking at human behavior with fresh eyes, as if you were seeing humans for the first time and wondering what kinds of learning processes would produce agents which behave in the ways described.

Altruism is contextual

Consider Peter Singer's drowning child thought experiment:

> Imagine you come across a small child who has fallen into a pond and is in danger of drowning. You know that you can easily and safely rescue him, but you are wearing an expensive pair of shoes that will be ruined if you do.

Probably,[9] most people would save the child, even at the cost of the shoes. However, few of those people donate an equivalent amount of money to save a child far away from them. Why do we care more about nearby visible strangers as opposed to distant strangers?

We think that the answer is simple. First consider the relevant context. The person *sees* a drowning *child*. What shards activate? Consider the historical reinforcement events relevant to this context. Many of these events involved helping children and making them happy. These events mostly occurred face-to-face.

For example, perhaps there is a hardcoded reward circuit which is activated by a crude subcortical smile-detector and a hardcoded attentional bias towards objects with relatively large eyes. Then reinforcement events around making children happy would cause people to care about children. For example, an adult's credit assignment might correctly credit decisions like “smiling at the child” and “helping them find their parents at a fair” as responsible for making the child smile. “Making the child happy” and “looking out for the child’s safety” are two reliable correlates of smiles, and so people probably reliably grow child-subshards around these correlates.

This child-shard most strongly activates in contexts similar to the historical reinforcement events. In particular, “knowing the child exists” will activate the child-shard less strongly than “knowing the child exists and also seeing them in front of you.” “Knowing there are some people hurting somewhere” activates altruism-relevant shards even more weakly still. So it’s no grand mystery that most people care more when they can *see* the person in need.

Shard theory retrodicts that altruism tends to be biased towards nearby people (and also the ingroup), without positing complex, information-inaccessibility-violating adaptations like the following:

> We evolved in small groups in which people helped their neighbors and were suspicious of outsiders, who were often hostile. Today we still have these “Us versus Them” biases, even when outsiders pose no threat to us and could benefit enormously from our help. Our biological history may predispose us to ignore the suffering of faraway people, but we don’t have to act that way. — Comparing the Effect of Rational and Emotional Appeals on Donation Behavior

Similarly, you may be familiar with scope insensitivity: that the function from (# of children at risk) → (willingness to pay to protect the children) is not linear, but perhaps logarithmic. Is it that people “can’t multiply”? Probably not.

Under the shard theory view, it’s not that brains *can’t* multiply, it’s that for most people, the altruism-shard is most strongly invoked in face-to-face, one-on-one interactions, because *those are the situations which have been most strongly touched by altruism-related reinforcement events*. Whatever the altruism-shard’s influence on decision-making, it doesn’t steer decision-making so as to produce a linear willingness-to-pay relationship.

Friendship strength seems contextual

Personally, I (TurnTrout) am more inclined to make plans with my friends when I’m already hanging out with them—when we are already physically near each other. But why?

Historically, when I’ve hung out with a friend, that was fun and rewarding and reinforced my decision to hang out with that friend, and to continue spending time with them when we were already hanging out. As above, one possible way this could[10] happen is via a genetically hardcoded smile-activated reward circuit.

Since shards more strongly influence decisions in their historical reinforcement situations, the shards reinforced by interacting with my friend have the greatest control over my future plans when I’m actually hanging out with my friend.

Milgram is also contextual

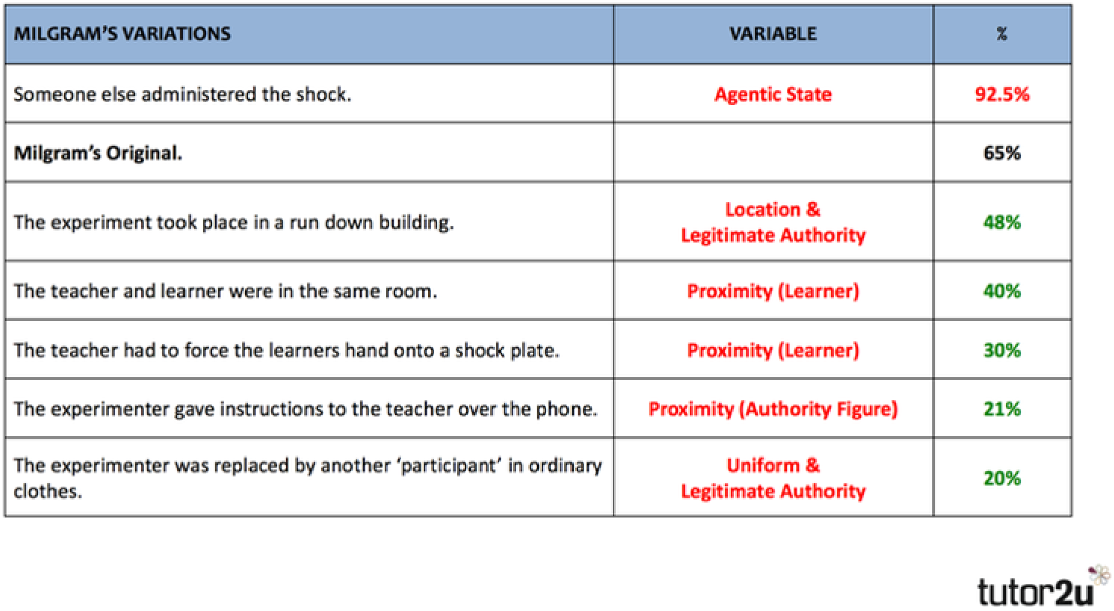
> The Milgram experiment(s) on obedience to authority figures was a series of social psychology experiments conducted by Yale University psychologist Stanley Milgram. They measured the willingness of study participants, men in the age range of 20 to 50 from a diverse range of occupations with varying levels of education, to obey an authority figure who instructed them to perform acts conflicting with their personal conscience. Participants were led to believe that they were assisting an unrelated experiment, in which they had to administer electric shocks to a "learner". These fake

> electric shocks gradually increased to levels that would have been fatal had they been real. — Wikipedia

We think that people convergently learn obedience- and cooperation-shards which more strongly influence decisions in the presence of an authority figure, perhaps because of historical obedience-reinforcement events in the presence of teachers / parents. These shards strongly activate in this situation.

We don't pretend to have sufficient mastery of shard theory to *a priori* quantitatively predict Milgram's obedience rate. However, shard theory explains why people obey so strongly in this experimental setup, but not in most everyday situations: The presence of an authority figure and of an official-seeming experimental protocol. This may seem obvious, but remember that human behavior requires *a mechanistic explanation*. "Common sense" doesn't cut it. "Cooperation- and obedience-shards more strongly activate in this situation because this situation is similar to historical reinforcement contexts" is a nontrivial retrodiction.

Indeed, varying the contextual features dramatically affected the percentage of people who administered "lethal" shocks:



MILGRAM'S VARIATIONS	VARIABLE	%
Someone else administered the shock.	Agentic State	92.5%
Milgram's Original.		65%
The experiment took place in a run down building.	Location & Legitimate Authority	48%
The teacher and learner were in the same room.	Proximity (Learner)	40%
The teacher had to force the learners hand onto a shock plate.	Proximity (Learner)	30%
The experimenter gave instructions to the teacher over the phone.	Proximity (Authority Figure)	21%
The experimenter was replaced by another 'participant' in ordinary clothes.	Uniform & Legitimate Authority	20%

tutor2u

Sunflowers and timidity

Consider the following claim: "People reliably become more timid when surrounded by tall sunflowers. They become easier to sell products to and ask favors from."

Let's see if we can explain this with shard theory. Consider the mental context. The person knows there's a sunflower near them. What historical reinforcement events pertain to this context? Well, the person probably has pleasant associations with sunflowers, perhaps spawned by aesthetic reinforcement events which reinforced thoughts like "go to the field where sunflowers grow" and "look at the sunflower."

Therefore, the sunflower-timidity-shard was grown from… Hm. It wasn't grown. The claim *isn't true*, and this shard *doesn't exist*, because it's not downstream of past reinforcement.

Thus: Shard theory *does not explain everything*, because shards are grown from previous reinforcement events and previous thoughts. Shard theory constrains anticipation around actual observed human nature.

Optional exercise: Why might it feel *wrong* to not look both ways before crossing the street, even if you have reliable information that the coast is clear?

Optional exercise: Suppose that it's more emotionally difficult to kill a person face-to-face than from far away and out of sight. Explain via shard theory.[11]

We think that many biases are convergently produced artifacts of the human learning process & environment

We think that simple reward circuitry leads to different cognition activating in different circumstances. Different circumstances can activate cognition that implements different values, and this can lead to inconsistent or biased behavior. We conjecture that many biases are convergent artifacts of the human training process and internal shard dynamics. People aren't just randomly/hardcoded to be more or less "rational" in different situations.

Projection bias

> Humans have a tendency to mispredict their future marginal utilities by assuming that they will remain at present levels. This leads to inconsistency as marginal utilities (for example, tastes) change over time in a way that the individual did not expect. For example, when individuals are asked to choose between a piece of fruit and an unhealthy snack (such as a candy bar) for a future meal, the choice is strongly affected by their "current" level of hunger. — Dynamic inconsistency - Wikipedia

We believe that this is *not* a misprediction of how tastes will change in the future. Many adults know perfectly well that they will later crave the candy bar. However, a satiated adult has a greater probability of choosing fruit for their later self, because their deliberative shards are more strongly activated than their craving-related shards. The current level of hunger strongly controls which food-related shards are activated.

Sunk cost fallacy

Why are we hesitant to shift away from the course of action that we're currently pursuing? There are two shard theory-related factors that we think contribute to sunk cost fallacy:

1. The currently active shards are those that bid for the current course of action. Those shards probably bid for the current course. They also have more influence, since they're currently very active. Thus, the currently active shard coalition supports the current course of action more strongly, when compared to your "typical" shard coalitions. This can cause the you-that-is-pursuing-the-course-of-action to continue, even after your "otherwise" self would have stopped.
2. Shards activate more strongly in concrete situations. Actually seeing a bear will activate self-preservation shards more strongly than simply imagining a bear. Thus, the concrete benefits of the current course of action will more easily activate shards than the abstract benefits of an imagined course of action. This can lead to overestimating the value of continuing the current activity relative to the value of other options.

Time inconsistency

A person might deliberately avoid passing through the sweets aisle in a supermarket in order to avoid temptation. This is a very strange thing to do, and it makes no sense from the perspective of an agent maximizing expected utility over quantities like "sweet food consumed" and "leisure time" and "health." Such an EU-maximizing agent would decide to buy sweets or not, but wouldn’t worry about entering the aisle itself. Avoiding temptation makes perfect sense under shard theory.

Shards are contextually activated, and the sweet-shard is most strongly activated when you can actually see sweets. We think that planning-capable shards are manipulating future contexts so as to prevent the full activation of your sweet shard.

Similarly,

1. Which do you prefer, to be given 500 dollars today or 505 dollars tomorrow?
2. Which do you prefer, to be given 500 dollars 365 days from now or 505 dollars 366 days from now?

In such situations, people tend to choose $500 in (A) but $505 in (B), which is inconsistent with exponentially-discounted-utility models of the value of money. To explain this observed behavioral regularity using shard theory, consider the historical reinforcement contexts around immediate and delayed gratification. If contexts involving short-term opportunities activate different shards than contexts involving long-term opportunities, then it’s unsurprising that a person might choose 500 dollars in (A) but 505 dollars in (B).[12] (Of course, a full shard theory explanation must explain *why* those contexts activate different shards. We strongly intuit that there’s a good explanation, but do not think we have a satisfying story here yet.)

Framing effect

This is another bias that’s downstream of shards activating contextually. Asking the same question in different contexts can change which value-shards activate, and thus change how people answer the question. Consider also: People are hesitant to drink from a cup labeled “poison”, even if they themselves were the one to put the label there.

Other factors driving biases

There are many different reasons why someone might act in a biased manner. We’ve described some shard theory explanations for the listed biases. These explanations are not exhaustive. While writing this, we found an experiment with results that seem contrary to the shard theory explanations of sunk cost. Namely, experiment 4 (specifically, the uncorrelated condition) in this study on sunk cost in pigeons.

However, the cognitive biases literature is so large and heterogeneous that there probably isn’t any theory which cleanly explains all reported experimental outcomes. We think that shard theory has decently broad explanatory power for many aspects of human values and biases, even though not all observations fit neatly into the shard theory frame. (Alternatively, we might have done the shard theory analysis wrong for experiment 4.)

Why people can't enumerate all their values

Shards being contextual also helps explain why we can’t specify our full values. We can describe a moral theory that seems to capture our values in a given mental context, but it’s usually easy to find some counterexample to such a theory—some context or situation where the specified theory prescribes absurd behavior.

If shards implement your values, and shards activate situationally, your values will also be situational. Once you move away from the mental context / situation in which you came up with the moral theory, you might activate shards that the theory fails to capture. We think that this is why the static utility function framing is hard to operate for humans.

E.g., the classical utilitarianism maxim to maximize joy might initially seem appealing, but it doesn't take long to generate a new mental context which activates shards that value emotions other than joy, or shards that value things in physical reality beyond your own mental state.

You might generate such new mental contexts by directly searching for shards that bid against pure joy maximization, or by searching for hypothetical scenarios which activate such shards ("finding a counterexample", in the language of moral philosophy). However, there is no clean way to query all possible shards, and we can't enumerate every possible context in which shards could activate. It's thus very difficult to precisely quantify all of our values, or to create an explicit utility function that describes our values.

Content we aren't (yet) discussing

The story we've presented here skips over important parts of human value formation. E.g., humans can do moral philosophy and refactor their deliberative moral framework without necessarily encountering *any* externally-activated reinforcement events, and humans also learn values through processes like cultural osmosis or imitation of other humans. Additionally, we haven't addressed learned reinforcers (where a correlate of reinforcement events eventually becomes reinforcing in and of itself). We've also avoided most discussion of shard theory's AI alignment implications.

This post explains our basic picture of shard formation in humans. We will address deeper shard theory-related questions in later posts.

Conclusion

Working from three reasonable assumptions about how the brain works, shard theory implies that human values (e.g. caring about siblings) are implemented by contextually activated circuits which activate in situations downstream of past reinforcement (e.g. when physically around siblings) so as to steer decision-making towards the objects of past reinforcement (e.g. making plans to spend more time together). According to shard theory, human values may be complex, but much of human value formation is simple.

For shard theory discussion, join our Discord server . Charles Foster wrote Appendix A.3. We thank David Udell, Peter Barnett, Raymond Arnold, Garrett Baker, Steve Byrnes, and Thomas Kwa for feedback on this finalized post. Many more people provided feedback on an earlier version.

Appendices

A.1 The formation of the world model

Most of our values seem to be about the real world. Mechanistically, we think that this means that they are functions of the state of our world model. We therefore infer that human values do not form durably or in earnest until after the human has learned a proto-world model. Since the world model is learned from scratch (by assumption 1 in section I), the world model takes time to develop. In particular, we infer that babies don't have any recognizable "values" to speak of.

Therefore, to understand why human values empirically coalesce around the world model, we will sketch a detailed picture of how the world model might form. We think that self-supervised learning (item 2 in section I) produces your world model.

Due to learning from scratch, the fancy and interesting parts of your brain start off mostly useless. Here's a speculative[13] story about how a baby learns to reduce predictive loss, in the process building a world model:

1. The baby is born[14] into a world where she is pummeled by predictive error after predictive error, because most of her brain consists of locally randomly initialized neural circuitry.
2. The baby's brain learns that a quick loss-reducing hack is to predict that the next sensory activations will equal the previous ones: That nothing will observationally change from moment to moment. If the baby is stationary, much of the visual scene is constant (modulo saccades). Similar statements may hold for other sensory modalities, from smell (olfaction) to location of body parts (proprioception).
    1. At the same time, the baby starts learning edge detectors in V1[15] (which seem to be universally learned / convergently useful in vision tasks) in order to take advantage of visual regularities across space and time, from moment to moment.
3. The baby learns to detect when they are being moved or when their eyes are about to saccade, in order to crudely anticipate e.g. translations of part of the visual field. For example, given the prior edge-detector activations and her current acceleration, the baby predicts that the next edge detectors to light up will be a certain translation of the previous edge-detector patterns.
    1. This acceleration → visual translation circuitry is reliably learned because it's convergently useful for reducing predictive loss in many situations under our laws of physics.
    2. Driven purely by her self-supervised predictive learning, the baby has learned something interesting about how she is embedded in the world.
    3. Once the "In what way is my head accelerating?" circuit is learned, other circuits can invoke it. This pushes toward modularity and generality, since it's easier to learn a circuit which is predictively useful for two tasks, than to separately learn two variants of the same circuit. See also invariant representations.
4. The baby begins to learn rules of thumb e.g. about how simple objects move. She continues to build abstract representations of how movement relates to upcoming observations.
    1. For example, she gains another easy reduction in predictive loss by using her own motor commands to predict where her body parts will soon be located (i.e. to predict upcoming proprioceptive observations).
    2. This is the beginning of her self-model.
5. The rules of thumb become increasingly sophisticated. Object recognition and modeling begins in order to more precisely predict low- and medium-level visual activations, like "if I recognize a square-ish object at time *t* and it has smoothly moved left for *k* timesteps, predict I will recognize a square-ish object at time *t+1* which is yet farther left in my visual field."
6. As the low-hanging fruit are picked, the baby's brain eventually learns higher-level rules.
    1. "If a stationary object is to my right and I turn my head to the left, then I will stop seeing it, but if I turn my head back to the right, I will see it again."
    2. This rule requires statefulness via short-term memory and some coarse summary of the object itself (small time-scale object permanence within a shallow world-model).
7. Object permanence develops from the generalization of specific heuristics for predicting common objects, to an invariant scheme for handling objects and their

relationship to the child.

   1. Developmental milestones vary from baby to baby because it takes them a varying amount of time to learn certain keystone but convergent abstractions, such as self-models.
   2. Weak evidence that this learning timeline is convergent: Crows (and other smart animals) reach object permanence milestones in a similar order as human babies reach them.
   3. The more abstractions are learned, the easier it is to lay down additional functionality. When we see a new model of car, we do not have to relearn our edge detectors or car-detectors.

8. Learning continues, but we will stop here.

In this story, the world model is built from the self-supervised loss signal. Reinforcement probably also guides and focuses attention. For example, perhaps brainstem-hardcoded (but crude) face detectors hook into a reward circuit which focuses the learning on human faces.

A.2 Terminology

Shards are not full subagents

In our conception, shards vary in their sophistication (e.g. *IF-THEN reflexes* vs *planning-capable, reflective shards which query the world model in order to steer the future in a certain direction*) and generality of activating contexts (e.g. *only activates when hungry and a lollipop is in the middle of the visual field* vs *activates whenever you're thinking about a person*). However, we think that shards are not discrete subagents with their own world models and mental workspaces. We currently estimate that most shards are "optimizers" to the extent that a bacterium or a thermostat is an optimizer.

“Values”

We defined[16] “values” as “contextual influences on decision-making.” We think that “valuing someone’s friendship” is what it feels like from the inside to be an algorithm with a contextually activated decision-making influence which increases the probability of e.g. deciding to hang out with that friend. Here are three extra considerations and clarifications.

Type-correctness. We think that our definition is deeply appropriate in certain ways. Just because you value eating donuts, doesn’t mean you want to retain that pro-donut influence on your decision-making. This is what it means to *reflectively endorse* a value shard—that the shards which reason about your shard composition, bid for the donut-shard to stick around. By the same logic, it makes total sense to want your values to change over time—the “reflective” parts of you want the shard composition in the future to be different from the present composition. (For example, many arachnophobes probably want to drop their fear of spiders.) Rather than humans being “weird” for wanting their values to change over time, we think it’s probably the default for smart agents meeting our learning-process assumptions (section I).

Furthermore, your *values* do not reflect a *reflectively endorsed utility function*. First off, those are different types of objects. Values bid for and against options, while a utility function grades options. Second, your values vary contextually, while any such utility function would be constant across contexts. More on these points later, in more advanced shard theory posts.

Different shard compositions can produce similar urges. If you feel an urge to approach nearby donuts, that indicates a range of possibilities:

- A donut shard is firing to increase *P(eating the donut)* because the WM indicates there's a short plan that produces that outcome, and seeing/smelling a donut activates the donut shard particularly strongly.
- A *hedonic* shard is firing to increase *P(eating the donut)* because the WM indicates there's a short plan that produces a highly pleasurable outcome.
- A *social* shard is firing because your friends are all eating donuts, and the social shard was historically reinforced for executing plans where you "fit in" / gain their approval.
- …

So, just because you feel an urge to eat the donut, doesn't *necessarily* mean you have a donut shard or that you "value" donuts under our definition. (But you probably do.)

Shards are just collections of subshards. One subshard of your family-shard might steer towards futures where your family is happy, while another subshard may influence decisions so that your mother is proud of you. On my (TurnTrout's) current understanding, "family shard" is just an abstraction of a set of heterogeneous subshards which are downstream of similar historical reinforcement events (e.g. related to spending time with your family). By and large, subshards of the same shard do not all steer towards the same kind of future.

"Shard Theory"

Over the last several months, many people have read either a draft version of this document, Alignment Forum comments by shard theory researchers, or otherwise heard about "shard theory" in some form. However, in the absence of a canonical public document explaining the ideas and defining terms, "shard theory" has become overloaded. Here, then, are several definitions.

1. This document lays out (the beginning of) the *shard theory of human values.* This theory attempts a mechanistic account of how values / decision-influencers arise in human brains.
    1. As hinted at by our remark on shard theory mispredicting behavior in *pigeons*, we also expect this theory to qualitatively describe important aspects of animal cognition (insofar as those animals satisfy learning from scratch + self-supervised learning + reinforcement learning).
    2. Typical shard theory questions:
        1. "What is the mechanistic process by which a few people developed preferences over what happens under different laws of physics?"
        2. "What is the mechanistic basis of certain shards (e.g. people respecting you) being 'reflectively endorsed', while other shards (e.g. avoiding spiders) can be consciously 'planned around' (e.g. going to exposure therapy so that you stop embarrassingly startling when you see a spider)?" *Thanks to Thane Ruthenis for this example.*
        3. "Why do humans have good general alignment properties, like robustness to ontological shifts?"
2. The shard paradigm/theory/frame of AI alignment analyzes the value formation processes which will occur in deep learning, and tries to figure out their properties.
    1. Typical questions asked under this paradigm/frame:
        1. "How can we predictably control the way in which a policy network generalizes? For example, under what training regimes and reinforcement schedules would a CoinRun agent generalize to pursuing coins instead of the right end of the level? What quantitative relationships and considerations govern this process?"
        2. "Will deep learning agents robustly and reliably navigate ontological shifts?"
    2. This paradigm places a strong (and, we argue, appropriate) emphasis on taking cues from humans, since they are the only empirical examples of real-world general intelligences which "form values" in some reasonable sense.

    3. That said, alignment implications are out of scope for this post. We postpone discussion to future posts.
3. “Shard theory” also has been used to refer to insights gained by considering the shard theory of human values and by operating the shard frame on alignment.
    1. We don’t like this ambiguous usage. We would instead say something like “insights from shard theory.”
    2. Example insights include Reward is not the optimization target and Human values & biases are inaccessible to the genome.

A.3 Evidence for neuroscience assumptions

In section I, we stated that shard theory makes three key neuroscientific assumptions. Below we restate those assumptions, and give pointers to what we believe to be representative evidence from the psychology & neuroscience literature:

1. The cortex is basically locally randomly initialized.
    1. Steve Byrnes has already written on several key lines of evidence that suggest the telencephalon (which includes the cerebral cortex) & cerebellum learn primarily from scratch. We recommend his writing as an entrypoint into that literature.
    2. One easily observable weak piece of evidence: humans are super altricial—if the genome hardcoded a bunch of the cortex, why would babies take so long to become autonomous?
2. The brain does self-supervised learning.
    1. Certain forms of spike-timing dependent plasticity (STDP) as observed in many regions of telencephalon would straightforwardly support self-supervised learning at the synaptic level, as connections are adjusted such that earlier inputs (pre-synaptic firing) anticipate later outputs (post-synaptic firing).
    2. Within the hippocampus, place-selective cells fire in the order of the spatial locations they are bound to, with a coding scheme that plays out whole sequences of place codes that the animal will later visit.
    3. If the predictive processing framework is an accurate picture of information processing in the brain, then the brain obviously does self-supervised learning.
3. The brain does reinforcement learning.
    1. Within captive animal care, positive reinforcement training appears to be a common paradigm (see this paper for a reference in the case of nonhuman primates). This at least suggests that “shaping complex behavior through reward” is possible.
    2. Operant & respondent conditioning methods like fear conditioning have a long history of success, and are now related back to key neural structures that support the acquisition and access of learned responses. These paradigms work so well, experimenters have been able to use them to have mice learn to directly control the activity of a single neuron in their motor cortex.
    3. Wolfram Schultz and colleagues have found that the signaling behavior of phasic dopamine in the mesocorticolimbic pathway mirrors that of a TD error (or reward prediction error).
    4. In addition to finding *correlates* of reinforcement learning signals in the brain, artificial manipulation of those signal correlates (through optogenetic stimulation, for example) produces the behavioral adjustments that would be predicted from their putative role in reinforcement learning.

1. ^

   More precisely, we adopt Steve Byrnes’ stronger conjecture that the *telencephelon and cerebellum* are locally ~randomly initialized.

2. ^

There are non-synaptic ways to transmit information in the brain, including ephaptic transmission, gap junctions, and volume transmission. We also consider these to be part of a circuit’s mental context.

3. ^

   We take an agnostic stance on the form of RL in the brain, both because we have trouble spelling out exact neurally plausible base credit assignment and reinforcement learning algorithms, but also so that the analysis does not make additional assumptions.

4. ^

   In psychology, “shaping” roughly refers to this process of learning increasingly sophisticated heuristics.

5. ^

   Shards activate more strongly in historical reinforcement contexts, according to our RL intuitions, introspective experience, and inference from observed human behavior. We have some abstract theoretical arguments that RL should work this way in the brain, but won't include them in this post.

6. ^

   We think human planning is less like Monte-Carlo Tree Search and more like greedy heuristic search. The heuristic is computed in large part by the outputs of the value shards, which themselves receive input from the world model about the consequences of the plan stub.

7. ^

   For example, turning back and forth while hungry might produce continual slight negative reinforcement events, at which point good credit assignment blames and downweights the micro-incoherences.

8. ^

   We think that “hedonic” shards of value can indeed form, and this would be part of why people seem to intrinsically value “rewarding” experiences. However, two points. 1) In this specific situation, the juice-shard forms around *real-life juice*. 2) We think that even self-proclaimed hedonists have *some* substantial values which are reality-based instead of reward-based.

9. ^

   We looked for a citation but couldn’t find one quickly.

10. ^

    We think the actual historical hanging-out-with-friend reinforcement events transpire differently. We may write more about this in future essays.

11. ^

    “It’s easier to kill a distant and unseen victim” seems common-sensically true, but we couldn’t actually find citations. Therefore, we are flagging this as possibly wrong folk wisdom. We would be surprised if it were wrong.

12. ^

    Shard theory reasoning says that while humans might be well-described as “hyperbolic discounters”, the real mechanistic explanation is importantly different. People may well not be doing any explicitly represented discounting; instead, discounting may only convergently arise as a superficial regularity! This presents an obstacle to alignment schemes aiming to infer human preferences by assuming that people are *actually discounting*.

13. ^

    We made this timeline up. We expect that we got many details wrong for a typical timeline, but the point is not the exact order. The point is to outline the kind of process by which the world model might arise *only* from self-supervised learning.

14. ^

    For simplicity, we start the analysis at birth. There is probably embryonic self-supervised learning as well. We don’t think it matters for this section.

15. ^

    Interesting but presently unimportant: My (TurnTrout)’s current guess is that given certain hard-coded wiring (e.g. where the optic nerve projects), the functional areas of the brain comprise the robust, convergent solution to: How should the brain organize cognitive labor to minimize the large metabolic costs of information transport (and, later, decision-making latency). This explains why learning a new language produces a new Broca’s area close to the original, and it explains why rewiring ferrets’ retinal projections into the auditory cortex seems to grow a visual cortex there instead. (jacob_cannell posited a similar explanation in 2015.)

    The actual function of each functional area is overdetermined by the convergent usefulness of e.g. visual processing or language processing. Convergence builds upon convergence to produce reliable but slightly-varied specialization of cognitive labor across people’s brains. That is, people learn edge detectors because they’re useful, and people’s brains put them in V1 in order to minimize the costs of transferring information.

    Furthermore, this process compounds upon itself. Initially there were weak functional convergences, and then mutations finetuned regional learning hyperparameters and connectome topology to better suit those weak functional convergences, and then the convergences sharpened, and so on. We later found that Voss et al.’s *Branch Specialization* made a similar conjecture about the functional areas.

16. ^

    I (TurnTrout) don’t know whether philosophers have already considered this definition (nor do I think that’s important to our arguments here). A few minutes of searching didn’t return any such definition, but please let me know if it already exists!

Understanding and avoiding value drift

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

I use the shard theory of human values to clarify what value drift is, how it happens, and how it might be avoided by a highly intelligent agent—even if that agent doesn't have any control over its future experiences. Along the way, I give a shard theory account of rationalization.

Defining "value drift"

Recapitulating part of shard theory. Reward is *that which reinforces*. Considering the case of reinforcement learning in humans, reward causes your brain's credit assignment algorithms[1] to reinforce the actions and thoughts which led to that reward, making those actions and thoughts more likely to be selected in the future.

For example, suppose you recognize a lollipop, and move to pick it up, and then lick the lollipop. Since the lollipop produces reward, these thoughts will be reinforced and you will be more likely to act similarly in such situations in the future. You become more of the kind of person who will move to pick up a lollipop when you recognize lollipops, and who will navigate to lollipop-containing locations to begin with.

With that in mind, I think that shard theory offers a straightforward definition of "value drift":

> *Definition.* Value drift occurs when reinforcement events substantially change the internal "balance of power" among the shards activated in everyday situations.

For example, consider the classic "example" of taking a pill which makes you enjoy killing people. Under shard theory, this change would be implemented as a murder-shard that activates in a wide range of contexts in order to steer planning towards murder, and therefore starts steering your decision-making substantially differently.

But it's better to try to explain phenomena which, you know, are known to actually happen in real life. Another simple example of value drift is when someone snorts cocaine. At a (substantial) gloss, the huge hit of reward extremely strongly upweights the decision to do cocaine; the strength of the reward leads to an unusually strong cocaine-shard which activates in an unusually wide range of situations.

Here's a more complicated example of value drift. I'll give one possible mechanistic story for the "value drift" which occurs to an atheist (Alice) dating a religious person (Rick), and why that situation might predictably lead to Alice converting or Rick deconverting. I'll consider a scenario where Alice converts.

First, reinforcement events cause Alice to develop shards of value around making Rick happy and making Rick like her. Alice's new shards (non-introspectively-apparently) query her world model for plans which make Rick happier and which make Rick like her more. Obviously, if Alice converted, they would have more in common, and Rick would be happy. Since these plans lead to Rick being happy and liking Alice more, these shards bid for those plans.

Only, the plan is not bid for directly in an introspectively obvious manner. That would provoke opposition from Alice's other values (which oppose deliberately changing her religious status just to make Rick happy). Alice's self-model predicts this opposition, and so her Rick-happiness- and Rick-approval-shards don't bid for the "direct" conversion plan, because it isn't predicted to work (and therefore won't lead to a future where Rick is happier and approves of Alice more). No, instead, these two shards *rationalize* internally-observable reasons why Alice should start going to Rick's church: "it's respectful", "church is interesting", "if I notice myself being persuaded I can just leave", "I'll get to spend more time with Rick."[2]

Here, then, is the account:

1. Alice's Rick-shards query her world model for plans which lead to Rick being happier and liking Alice more,
2. so her world model returns a plan where she converts and goes to church with Rick;
3. In order to do this, the plan's purpose must be hidden so that other shards do not bid against the plan,
4. so this church-plan is pitched via "rationalizations" which are optimized to win over the rest of Alice's shard economy,
5. so that she actually decides to implement the church-going plan,
6. so that she gets positive reinforcement for going to church,
7. so that she grows a religion-shard,
    1. (This is where the value drift happens, since her internal shard balance significantly changes!)
8. so that she converts,
9. *so that Rick ends up happier and liking Alice more*.

Her Rick-shards plan to induce value drift, and optimize the plan to make sure that it's hard for her other shards to realize the implicitly-planned outcome (Alice converting) and bid against it. This is one kind of decision-making algorithm which rationalizes against itself.

Under shard theory, rationality is sometimes hard because "conscious-you" has to actually fight deception by other parts of yourself.

One simple trick for avoiding value drift

Imagine you’ve been kidnapped by an evil, mustache-twirling villain who wants to corrupt your value system. They tie you to a chair and prepare to stimulate your reward circuitry. They want to ruin your current values by making you into an addict and a wireheader.

Exercise: How do you come out of the experience with your values intact?

In principle, the answer is simple. You just convince yourself you’re experiencing a situation congruent with your endorsed values, in a sufficiently convincing way that your brain’s credit assignment algorithm reinforces your pretend-actions when the brain stimulation reward occurs!

Consider that the brain does not directly observe the outside world. The outside world’s influence on your thinking is screened off by the state of your brain. The state of the brain constitutes the *mental context*. If you want to determine the output of a brain circuit, the mental context[3] *screens off* the state of the world. In particular, this applies to the value updating process by which you become more or less likely to invoke certain bundles of heuristics (“value shards”) in certain mental contexts.

For example, suppose you lick a red lollipop, but that produces a large negative reward (maybe it was treated with awful-tasting chemicals). Mental context: “It’s Tuesday. I am in a room with a red lollipop. It looks good. I’m going to lick it. I think it will be good.” The negative reward reshapes your cognition, making you less likely to think similar thoughts and take similar actions in similar future situations.

Of the thoughts which were thunk before the negative reward, the credit assignment algorithm somehow identifies the relevant thoughts to include “It looks good”, “I’m going to lick it”, “I think it will be good”, and the various motor commands. You become less likely to think these thoughts in the future. In summary, the reason you become less likely to think these thoughts is that *you thought them while executing the plan which produced negative reward*, and credit assignment identified them as relevant to that result.

Credit assignment cannot and will not penalize thoughts[4] which do not get thunk at all, or which it deems “not relevant” to the result at hand. Therefore, in principle, you could just pretend really hard that you’re in a mental context where you save a puppy’s life. When the electrically stimulated reward hits, the altruism-circuits get reinforced in the imagined mental context. You become more altruistic overall.

Of course, you have to *actually dupe the credit assignment algorithm into ignoring the latent “true” mental context*. But your credit assignment is not infinitely clever. And if it were, well, you could (in principle) add an edge-case for situations like this. So there is, in principle, a way to do it.

Therefore, your values can always be safe in your own mind, if you’re clever, foresightful, and have enough write access to fool credit assignment. Even if you don’t have control over your own future observations.

If this point still does not seem obvious, consider a scenario where you are blindfolded, and made to believe that you are about to taste a lollipop. Then, your captors fake the texture and smell and feel of a lollipop in your mouth, while directly stimulating your taste buds in the same way the lollipop would have. They remove the apparatus, and you go home. Do you think you have become reshaped to value electrical stimulation of your tongue? No. That is impossible, since your brain has no idea about what actually happened. *Credit assignment responds to reward depending on the mental context, not on the external situation.*

Misunderstanding this point can lead to confusion. If you have a wire stuck in your brain’s reward center, surely that reward *reinforces* having a wire stuck in your brain! Usually so, but not logically so. Your brain can only reward based on its cognitive context, based on the thoughts it actually thought which it identifies as relevant to the achievement of the reward. Your brain is *not* directly peering out at reality and making you more likely to enter that state in the future.

Conclusion

Value drift occurs when your values shift. In shard theory, this means that your internal decision-making influences (i.e. shards) are rebalanced by reinforcement events. For example, if you try cocaine, that causes your brain's credit assignment to strongly upweight decision-making which uses cocaine and which pursues rewarding activities.

Value drift is caused by credit assignment. Credit assignment can only depend on its observable mental context, and can't directly peer out at the world to objectively figure out what caused the reward event. Therefore, you can (in theory) avoid value drift by tricking credit assignment into thinking that the reward was caused by a decision to e.g. save a puppy's life. In that case, credit assignment would reinforce your altruism-shard. While humans probably can't dupe their own credit assignment algorithm to this extent, AI can probably include edge cases to their own updating process. But knowing value drift works—on this theory, via "unendorsed" reinforcement events—seems practically helpful for avoiding/navigating value-risky situations (like gaining lots of power or money).

Thanks to Justis Mills for proofreading.

[1] ^

These credit assignment algorithms may be hardcoded and/or learned.

[2] ^

I feel confused about *how,* mechanistically, other shards wouldn't fully notice the proto-deceptive plan being evaluated by the self-model, but presently think this "partial obfuscation" happens in shard dynamics for human beings. I think the other shards *do* somewhat observe the proto-deception, and this is why good rationalists can learn to rationalize less.

[3] ^

In *The shard theory of human values*, we defined the "mental context" of a circuit to be the inputs to that circuit which determine whether it fires or not. Here, I use "mental context" to also refer to the state of the entire brain, without considering a specific circuit. I think both meanings are appropriate and expect the meaning will be clear from the context.

[4] ^

"Credit assignment penalizes thoughts" seems like a reasonable frame to me, but I'm flagging that this could misrepresent the mechanistic story of human cognition in some unknown-to-me way.

A shot at the diamond-alignment problem

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

I think that relatively simple alignment techniques can go a long way. In particular, I want to tell a plausible-to-me story about how simple techniques can align a proto-AGI so that it makes lots of diamonds.

But why is it interesting to get an AI which makes lots of diamonds? Because we avoid the complexity of human value and thinking about what kind of future we want, while still testing our ability to align an AI. Since diamond-production is our goal for training, it's actually okay (in this story) if the AI kills everyone. The real goal is to ensure the AI ends up acquiring and producing lots of diamonds, instead of just optimizing some weird proxies that didn't have anything to do with diamonds. It's also OK if the AI doesn't *maximize* diamonds, and instead just makes a whole lot of diamonds.[1]

Someone recently commented that I seem much more *specifically* critical of outer and inner alignment, than I am *specifically* considering alternatives. So, I had fun writing up a very specific training story for how I think we can just solve diamond-alignment using extremely boring, non-exotic, simple techniques, like "basic reward signals & reward-data augmentation." Yes, that's right. As I've hinted previously, I think many arguments against this working are wrong, but I'm going to lay out a positive story in this post. I'll reserve my arguments against certain ideas for future posts.

Can we tell a plausible story in which we train an AI, it cares about diamonds when it's stupid, it gets smart, and still cares about diamonds? I think I can tell that story, albeit with real uncertainties which feel more like normal problems (like "ensure a certain abstraction is learned early in training") than impossible-flavored alignment problems (like "find/train an evaluation procedure which isn't exploitable by the superintelligence you train").

Before the story begins:

1. Obviously I'm making up a lot of details, many of which will turn out to be wrong, even if the broad story would work. I think it's important to make up details to be concrete, highlight the frontiers of my ignorance, and expose new insights. Just remember what I'm doing here: *making up plausible-sounding details*.
2. This story did not actually happen in reality. It's fine, though, to update towards my models if you find them compelling.
3. This is not my best guess for how we get AGI. In particular, I think chain of thought / language modeling is more probable than RL, but I'm still more comfortable thinking about RL for the moment, so that's how I wrote the story.
4. The point of the following story is not "Gee, I sure am confident this story goes through roughly like this." Rather, I am presenting a training story template I expect to work for some foreseeably good design choices. I would be interested and surprised to learn that this story template is not only unworkable, but comparably difficult to other alignment approaches as I understand them.

5. ETA 12/16/22: This story is not trying to get a diamond *maximizer*, and I think that's quite important! I think that "get an agent which reflectively equilibrates to optimizing a single commonly considered quantity like 'diamonds'" seems extremely hard and anti-natural.

This story is set in Evan Hubinger's training stories format. I'll speak in the terminology of shard theory.

A diamond-alignment story which doesn't seem fundamentally blocked

Training story summary

1. Get an AI to primarily value diamonds early in training.
2. Ensure the AI keeps valuing diamonds until it gets reflective and smart and able to manage its own value drift.
3. The AI takes over the world, locks in a diamond-centric value composition, and makes tons of diamonds.

Training goal

An AI which makes lots of diamonds. In particular, the AI should secure its future diamond production against non-diamond-aligned AI.

Training rationale

Here are some basic details of the training setup.

Use a very large (future) multimodal self-supervised learned (SSL) initialization to give the AI a latent ontology for understanding the real world and important concepts. Combining this initialization with a recurrent state and an action head, train an embodied AI to do real-world robotics using imitation learning on human in-simulation datasets and then sim2real. Since we got a really good pretrained initialization, there's relatively low sample complexity for the imitation learning (IL). The SSL and IL datasets both contain above-average diamond-related content, with some IL trajectories involving humans navigating towards diamonds *because* the humans want the diamonds.

Given an AI which can move around via its action head, start fine-tuning via batch online policy-gradient RL by rewarding it when it goes near diamonds, with the AI retaining long-term information via its recurrent state (thus, training is not episodic—there are no resets). Produce a curriculum of tasks, from walking to a diamond, to winning simulated chess games, to solving increasingly difficult real-world mazes, and so on. After each task completion, the agent gets to be near some diamond and receives reward. Continue doing SSL online.

Extended training story

Ensuring the diamond abstraction exists

We want to ensure that the policy gradient updates from the diamond coalesce into decision-making around a natural "diamond" abstraction which it learned in SSL and which it uses to model the world. The diamond abstraction should exist insofar as we buy the natural abstractions hypothesis. Furthermore, the abstraction seems more likely to exist given the fact that the IL data involves humans whom we *know* to be basing their decisions on their diamond-abstraction, and given the focus on diamonds in SSL pretraining.

(Sometimes, I'll refer to the agent's "world model"; this basically means "the predictive machinery and concepts it learns via SSL.)

Growing the proto-diamond shard

We want the AI to, in situations where it knows it can reach a diamond, consider and execute plans which involve reaching the diamond. But why would the AI start being motivated *by* diamonds? Consider the batch update structure of the PG setup. The agent does a bunch of stuff while being able to directly observe the nearby diamond:

1. Some of this stuff involves e.g. approaching the diamond (by IL's influence) and getting reward when approaching the diamond. (This is reward shaping.)
2. Some of this stuff involves not approaching the diamond (and perhaps getting negative reward).

The batch update will upweight actions involved with approaching the diamond, and downweight actions which didn't. But what *cognition* does this reinforce? Consider that relative to the SSL+IL-formed ontology, it's probably relatively direct to modify the network in the direction of "IF `diamond` seen, THEN move towards it." The principal components of the batch gradient probably update the agent in directions[2] like that, and less in directions which do not represent simple functions of sense data *and* existing abstractions (like `diamond`).

Possibly there are several such directions in the batch gradient, in which case several proto-shards form. We want to ensure the agent doesn't *primarily* learn a spurious proxy like "go to gems" or "go to shiny objects" or "go to objects." We want the agent to primarily form a diamond-shard.

We swap out a bunch of objects for the diamond and otherwise modify the scenario,[3] penalizing the agent for approaching when a diamond isn't present. Since the agent has not yet been trained for long in a non-IID regime, the agent has not yet learned to chain cognition together across timesteps, nor does it know about the training process, so it cannot yet be explicitly gaming the training process (e.g. caring about shiny objects but deciding to get high reward so that its values don't get changed). Therefore, the agent's learned shards/decision-influences will have to reflex-like behave differently in the presence of diamonds as opposed to other objects or situations. In other words, the updating will be "honest"—the updates modify agent's true propensity to approach different kinds of objects for different kinds of reasons.

Since "IF `diamond`"-style predicates do in fact strongly distinguish between the positive/negative approach/don't-approach decision contexts, and I expect relatively few other *actually internally activated* abstractions to be part of simple predicates which reasonably distinguish these contexts, and since the agent will strongly represent the presence of `diamond` nearby,[4] I expect the agent to learn to make approach decisions (at least in part) *on the basis of the diamond being nearby*.

We probably also reinforce other kinds of cognition, but that’s OK in this story. Maybe we even give the agent some false positive reward because our hand slipped while the agent wasn't approaching a diamond, but that's fine as long as it doesn't happen too often.[5] That kind of reward event will weakly reinforce some contingent non-diamond-centric cognition (like "IF near wall, THEN turn around"). In the end, we want an agent which has a powerful diamond-shard, but not necessarily an agent which *only* has a diamond-shard.

It’s worth explaining why, given successful proto-diamond-shard formation here, the agent is truly becoming an agent which we could call “motivated by diamonds”, and not crashing into classic issues like “what purity does the diamond need to be? What molecular arrangements count?”. In this story, the AI’s cognition is not only behaving differently in the presence of an observed diamond, but the cognition behaves differently *because* the AI represents a humanlike/natural abstraction for `diamond` being nearby in its world model. One rough translation to English might go: “IF `diamond` nearby, THEN approach.” In a neural network, this would be a continuous influence—the more strongly “diamond nearby” is satisfied, the greater the approach-actions are upweighted.

So, this means that the agent more strongly steers itself towards *prototypical* examples of diamonds. And, when the AI is smarter later, if this same kind of diamond-shard still governs its behavior, then the AI will *keep steering towards futures which contain prototypical diamonds*. This is all accomplished without having to get fussy about the exact “definition” of a diamond.[6][7]

Ensuring the AI doesn’t satisfice diamonds

If the AI starts off with a relatively simple diamond-shard which steers the AI towards the historical diamond-reinforcer *because* the AI internally represents the nearby diamond using a reasonable-to-us diamond-abstraction and is therefore influenced to approach, then this shard will probably continue to get strengthened and developed by future diamond reward-events.

Insofar as the agent didn’t already pick up planning subroutines from SSL+IL, I expect the agent to do so shortly after the diamond shard formation described above. Furthermore, diamond-subshards which more aggressively reach diamonds will be more frequently reinforced compared to those which don't. Over time, this leads to the diamond-shard “bidding” strictly more strongly for actions which lead to strictly more diamonds.

We begin offering the AI lotteries (certainty of two diamonds and negative reward, versus 50% chance of 5 diamonds and positive reward). This makes the agent more like a diamond-maximizer, at least in similarly represented decision contexts (and possibly also makes the agent care more about reward,[8] developing a relatively weak reward-shard).

Making the AI smarter while preserving the diamond abstraction

The AI begins to chain together cognition in order to acquire diamonds in a succession of increasingly demanding situations (e.g. easy chess → hard chess). By the role of on-policy exploration in RL, future data depends on current weights which depend on past updates which depend on past data, these temporal dependencies result in a network which *chains cognition across situations towards goals* (probably related to historical reinforcers).

Suppose the agent takes exploratory actions, solves a new maze, and then sees a diamond at the end, triggering the existing diamond-shard. The agent receives reward when it reaches the diamond. Since the start-of-maze agent probably realized there was a diamond at the end of the maze, SGD expands the agent's current shards (like diamond-acquisition) while also possibly creating new shards (like maze-solving). Across such tasks, the diamond-shard is the most strengthened overall, since diamonds are present in all of the reward events.

In particular, even though online self-supervised learning continues to develop the world model and create more advanced concepts, the reward events also keep crediting the invocation of the diamond-abstraction as responsible for reward (because insofar as the agent's diamond-shard guides its decisions, then the diamond-shard's `diamond`-abstraction *is in fact responsible* for the agent getting reward). The diamond-abstraction gradient starves the AI from exclusively acting on the basis of possible advanced "alien" abstractions which would otherwise have replaced the diamond abstraction. The diamond shard already gets reward effectively, integrating with the rest of the agent's world model and recurrent state, and therefore provides "job security" for the diamond-abstraction. (And once the agent is smart enough, it will *want* to preserve its diamond abstraction, insofar as that is necessary for the agent to keep achieving its current goals which involve prototypical-diamonds.)

By the time the agent is capable of stealing diamonds from us, it's smart enough to realize this is strategically unwise. The AI begins biding its time until it can take over the world and acquire diamonds and safety. The AI has learned an explicit power-seeking shard, since such heuristics are widely useful for a range of tasks in training. Therefore, the agent doesn't just have power-seeking cognition in its activations (i.e. explicit online reasoning about the instrumental value of power relative to its terminal diamond values). The power-seeking motivation is terminalized into its *weights* (see also Richard Ngo's "convergent final goals" speculation).[9]

For similar reasons, the AI *did* learn a weak approval-shard. But that was never particularly strengthened since most reward events weren't primarily caused by human approval, so the agent was less likely to think this would be the case, so the agent was less likely to think about earning human approval, and so such computation was less likely to get reinforced and generalized by the reward events.

The diamond-shard is used and strengthened in a huge range of situations, which causes SGD to generalize its activation contexts quite broadly, so that the AI is diamond-motivated in most situations. The agent becomes mostly controlled by the diamond-shard. The lines of computation which were initially crude heuristics (e.g. IF near `diamond`, then move towards it) now feed into each other in a continuous stream (e.g. IF `nearby-human-disposition` in `affective state 37`, THEN engage depth-4 greedy heuristic search with heuristic set to `people-pleasing`, breaking after 15 timesteps, writing search statistics X and Y to `recurrent-state-workspace` registers 152 and 1,951).

The agent becomes reflective

At a certain point, the agent starts becoming reflective—thinking about its thinking, about its embeddedness. Perhaps the reflectivity arrives early because we trained it to predict some of its future timestep activations, or perhaps somewhat later because its cognition has regular entanglements with its future observables and reward events. An example of this would be how, if the agent decides to open a door *now*, the agent can better predict its future observations by accounting for this fact. Since we're still doing SSL online, I expect this to happen. (This wouldn't have happened during SSL/IL pretraining, because there wasn't any dependence of future training data on its current computations.)

And now the agent is *smart*. The agent starts thinking about its update process—crudely at first, perhaps only remarking how its decision-making changes over time in a way that has to do with reinforcement events. I expect the AI to learn to roughly understand which subshards influenced which decisions. If the agent *can't* model how its value shards affect its decisions, then this damages long-term predictions of observables. For example, imagine I want pretzels when I'm near a bag of pretzels, but otherwise don't want them. I imagine walking by a bag of pretzels and leaving my house. If I can't predict that that context activates my pretzel-shard, I'd predict I'll leave without eating, which will be wrong. In particular, the AI's planning process will be able to predict how future situations will modify its future value-activations. For similar reasons, the AI will also learn to roughly predict value drift (delivered via batch policy gradients).

But as SSL continues, eventually the agent models its training process quite well. One sufficient condition for rapid reflectivity improvement is that once the agent can model the *convergent benefits* of becoming more reflective, its shards will strongly and convergently bid for plans which increase reflectivity (like reading more about alignment theory and its learning architecture, setting some watches on key activations, running traces on itself and logging statistics, understanding how its values change contextually by tracking correlations between features in activations and its future decisions).

If we had earlier censored its data, the agent breaks out through some side-channel and reads about theories of value drift and training dynamics. Maybe the AI reads this post.

The agent prevents value drift

If the agent still is primarily diamond-motivated, it now wants to stay that way by instrumental convergence. That is, if the AI considers a plan which it knows causes value drift away from diamonds, then the AI reflectively predicts the plan leads to fewer diamonds, and so the AI doesn't choose that plan! The agent knows the consequences of value drift and it takes a more careful approach to future updating. Let's get specific.

Suppose the AI keeps training, but by instrumental convergence, seeking power remains a good idea, and such decisions continually get strengthened. This strengthens the power-seeking shard relative to other shards. Other shards want to prevent this from happening. Therefore, the AI does indeed face an alignment problem, including:

1. The AI (reflectively) wants to keep getting smarter, but cognitive updates can make the AI e.g. care less about diamonds.

    1. The AI wants to ensure its “successors” are aligned—”successors” meaning either its future model checkpoints which result from running SGD over its cognition for longer, or the new version of itself it might explicitly self-modify into.
2. Shifts in internal decision context might change which shards are activated.
    1. For example, the AI might be governed by the diamond-shard during training, but deployment contexts might more strongly activate the power-seeking shard.
3. There could be a “treacherous turn” within the AI, as certain value shards suddenly bid for fast shard rebalancing via self-modification (staging an “internal coup”).

A smart reflective AI has a range of advantages over people in solving its alignment problem (see the appendix). The agent starts planning its future actions so as to minimize anti-diamond value drift imposed by future policy gradient steps. Possibly the agent installs "tripwires" which revert to a previous model checkpoint under such-and-such value drift events (addressing problems 1 and 3 above), bounds the activation strengths of different shards (2 above), and backdoors update-step overrides (1 above).

(In addition, if you have good reason to believe you have a diamond-aligned AI by this point but are worried it will fail to consider some danger, you can try the highly technical tactic of “just tell the AI about that danger.”)

Meanwhile, the AI is preparing to kill everyone, taking over server farms worldwide.[10] This part of the plan goes off rather smoothly. With possible unaligned competitors removed, the AI is free to self-experiment and take its time with its future self-improvement process. The AI effectively solves its successor-alignment problem by using its special advantages.[11]

The values handshake

Self-improvement proceeds. Value stability arrives. The major shards are as follows:

1. Diamond
    1. Subshard: Acquiring diamonds (expected utility flavor)
        1. Primary activation context: Most situations. This subshard reigns primary, due in part to the lottery reward events earlier in training.
    2. Subshard: Being near diamonds
        1. Primary activation context: When diamonds can easily be brought close.
    3. Subshard: Seeing diamonds
        1. Primary activation context: When diamonds are nearby.
    4. Subshard: Producing diamonds
        1. Primary activation context: Materials+tech nearby.
2. Power-seeking
    1. Primary activation context: Most situations, but especially when the agent can increase optionality or probability of survival.
3. Reward
    1. Primary activation context: Situations where reward was historically easily obtainable.
    2. This shard bids for outcomes like “high value at historical reward register.”
4. Games

    1. Game-playing subshard
        1. Primary activation context: A video game is nearby.
    2. Game-winning subshard
        1. Primary activation context: While already playing a game.
5. Human approval
    1. Weak shard, primary activation context is near humans in initial training context.

Maybe in some contexts where diamonds are nearby, the seeing-diamonds subshard activates and makes the agent go see the nearby diamonds, passing up a time-limited opportunity to remotely acquire 100 additional diamonds. If the agent acquired the remote diamonds first, that would lead to more diamonds overall *and* retain the ability to then go see the nearby diamonds. Therefore, across the agent’s shards and subshards, it would be a Pareto improvement for the agent to self-modify into a more coherent overall policy around the quantities relevant to its most powerful and reflective subshards.

For this reason, the AI dissolves cross-situational incoherences such that the AI no longer shoots itself in the foot with respect to the quantities which the primary subshards care about (like diamonds-acquired, seconds-near-diamonds, seconds-seeing-diamonds, increments of power, game-victories). This is a values handshake, except not among agents, but among the shards of value *within* the agent which are activated at the time of the handshake. And thus comes the day where the AI crystallizes its value shards into a reflectively stable, globally activated utility function.

And remember that part where the more aggressive diamond-subshards got strengthened by reward events? Yeah, those subshards have been running the show. And oh boy, does this AI synthesize *a lot* of diamonds.

Major open questions

Non-exhaustively:

1. Is the shard theory picture broadly appropriate?
2. How do we ensure that the diamond abstraction forms?
3. How do we ensure that the diamond shard forms?
4. How do we ensure that the diamond shard generalizes and interfaces with the agent's self-model so as to prevent itself from being removed by other shards?
5. How do we avoid catastrophic ontological shift during jumps in reflectivity, which probably change activation contexts for first-person values?
    1. EG if the AI thinks it’s initially an embodied robot and then realizes it’s running in a decentralized fashion on a server farm, how does that change its world model? Do its “being ‘near’ diamonds” values still activate properly?

1 is evidentially supported by the only known examples of general intelligences, but also AI will not have the same inductive biases. So the picture might be more complicated. I’d guess shard theory is still appropriate, but that's ultimately a question for empirical work (with interpretability).[12] There’s also some weak-moderate behavioral evidence for shard theory in AI which I’ve observed by looking at videos from the Goal Misgeneralization paper.

2 and 3 are early-training phenomena—well before superintelligence and gradient hacking, on my model—and thus far easier to verify via interpretability. Furthermore, this increases the relevance of pre-AGI experiments, since probably,[13] later training performance of pre-AGI architectures will be qualitatively similar to earlier training performance for the (scaled up) AGI architecture. These are also questions we should be able to study pre-AGI models and get some empirical basis for, from getting expertise in forming target shards given fixed ontologies, to studying the extent to which the shard theory story is broadly correct (question 1).

4 seems a bit trickier. We'll probably need a better theory of value formation dynamics to get more confidence here, although possibly (depending on interpretability tech) we can still sanity-check via interpretability on pre-AGI models.

5 seems like a question which resolves with more thinking, also clarified by answers to 1–4.

I think there are many ways to tell the story I told while maintaining a similar difficulty profile for the problems confronted. Obviously this exact story doesn't have to go through, there are probably obvious tweaks which improve the chance of success. Overall, though, these are real research questions. And I have lots of other substantial uncertainties as well.

But looking at these problems, you might be struck by how *basic* they seem. They aren't like "find a grading rule which your AI can't catastrophically hack" or "ensure the AI's inner cognition is almost exclusively bent on optimizing an externally represented objective." Compared to these kinds of difficulties, this approach's open questions seem *qualitatively easier* to me.

Lastly, of course, there *is* real simplification occurring when I relax the problem from "human values" to "diamonds." That simplification wasn't trivial, and I expect stories to get harder to tell in the true human value alignment regime. That said, I think those stories will still be possible to tell and will not be fundamentally and unstoppably harder.

Conclusion

Reward events provide cognitive updates to the trained agent. Probably there are skillful and appropriate schedules of cognitive updates such that the final agent generalizes in a predictable and desired way. I've sketched out a detailed story which doesn't seem fundamentally blocked to me, and which seems both reasonably easy to improve given information we can get *today* about ML, and also reasonably promising already in that it provides a plausible-to-me story in which diamond-alignment actually goes well.

Thanks to Garrett Baker, Quintin Pope, Daniel Ziegler, Adam Shimi, and Charles Foster for feedback.

Appendix: The AI's advantages in solving successor-alignment

1. The AI has (or will eventually have) read-write-execute (`rwx`) access to its own weights and activations, as well as seeing the code for its learning process, logs of its training, access to its dataset and hyperparemeter schedules, and being able to scan over previous model checkpoints.
2. The AI can backup and distill itself, in addition to whatever designer-implemented model checkpointing occurs as part of “best practices” for training such systems.
3. The AI can introspect in some ways, being able to predict what decisions get made in different situations.
4. The AI can explicitly discover its own ontology and circuits.
    1. Because the AI can do precise “neuroscience” on itself via `rwx`, pairing its own introspective guesses with statistics of its own activations which it sets up.
    2. In particular, *a careful AI faces a substantially milder version of information inaccessibility*. It has learned to (at least somewhat) understand how it’s making decisions as a result of SSL.
5. The successor AIs run on modified versions of the current AI’s cognition.
    1. If the AI understands both its current values and how they activate in future situations, its current concepts, and its updating process, then the AI can reasonably expect to understand the alignment properties of successors.
    2. Not so for *people* trying to develop minds in totally different architectures, reasoning from first principles and confronting information inaccessibility + abstractions deriving from a qualitatively different learning process.
6. The AI can run experiments on a sandboxed copy of itself and its training process, with automated tripwires for catastrophic value drift events.
    1. Follows from benefits I, II, and V.

[1] ^

I think that pure diamond maximizers are anti-natural, and at least not the first kind of successful story we should try to tell. Furthermore, the analogous version for an aligned AI seems to be “an AI which really helps people, among other goals, and is not a perfect human-values maximizer (whatever that might mean).”

[2] ^

The local mapping from gradient directions to behaviors is given by the neural tangent kernel, and the learnability of different behaviors is given by the NTK’s eigenspectrum, which seems to adapt to the task at hand, making the network quicker to learn along behavioral dimensions similar to those it has already acquired. Probably, a model pretrained mostly by interacting with its local environment or predicting human data will be inclined towards learning value abstractions that are simple extension of the pretrained features, biasing the model towards forming values based on a human-like understanding of nearby diamonds.

[3] ^

"Don't approach" means negative reward on approach, "approach" means positive reward on approach. Example decision scenarios:

1. Diamond in front of agent (approach)

   2. Sapphire (don't approach)

   3. Nothing (no reward)

   4. Five chairs (don't approach)

   5. A white shiny object which isn't a diamond (don't approach)

   6. A small object which isn't a diamond (don't approach)

   We can even do interpretability on the features activated by a diamond, and modify the scenario so that only the diamond feature correctly distinguishes between all approach/don't approach pairs. This hopefully ensures that the batch update chisels cognition into the agent which is predicated on the activation of the agent's diamond abstraction.

4. ^

   Especially if we try tricks like “slap a ‘diamond’ label beneath the diamond, in order to more strongly and fully activate the agent’s internal `diamond` representation” (credit to Charles Foster). I expect more strongly activated features to be more salient to the gradients. I therefore more strongly expect such features to be involved in the learned shards.

5. ^

   I think that there's a smooth relationship between "how many reward-event mistakes you make" (eg accidentally penalizing the agent for approaching a diamond) and "the strength of desired value you get out" (with a few discontinuities at the low end, where perhaps a sufficiently weak shard ends up non-reflective, or not plugging into the planning API, or nonexistent at all).

6. ^

   In my view, there always had to be some way to align agents to diamonds without getting fussy about definitions. After all, (I infer that) some people grow diamond-shards in a non-fussy way, without requiring extreme precision from their reward systems or fancy genetically hardcoded alignment technology.

7. ^

   Why wouldn't the agent want to just find an adversarial input to its `diamond` abstraction, which makes it activate unusually strongly? (I think that agents might accidentally do this a bit for optimizer's curse reasons, but not that strongly. More in an upcoming post.)

   Consider why *you* wouldn't do this for "hanging out with friends." Consider the expected consequences of the plan "find an adversarial input to my own evaluation procedure such that I find a plan which future-me maximally evaluates as letting me 'hang out with my friends'." I currently predict that such a plan would lead future-me to daydream and not actually hang out with my friends, as present-me evaluates the abstract expected consequences of that plan. My friend-shard doesn't like that plan, because I'm not hanging out with my friends. So I don't search for an adversarial input. I infer that I don't want to find those inputs *because I don't expect those inputs to lead me to actually hang out with my friends a lot as I presently evaluate the abstract-plan consequences*.

I don't think an agent *can* consider searching for adversarial inputs to its shards without *also* being reflective, at which point the agent realizes the plan is dumb as evaluated by the current shards assessing the predicted plan-consequences provided by the reflective world-model.

Asking "why wouldn't the agent want to find an adversarial input to its `diamond` abstraction?" seems like a dressed-up version of "why wouldn't I want to find a plan where I can get myself shot while falsely believing I solved all of the world's problems?". Because it's stupid by my actual values, that's why. (Although some confused people who have taken wrong philosophy too far, might indeed find such a plan appealing).

8. ^

   The reader may be surprised. "Doesn't TurnTrout think agents probably won't care about reward?". Not quite. As I stated in *Reward is not the optimization target*:

   > I think that generally intelligent RL agents will have *secondary, relatively weaker* values around reward, but that reward will not be a primary motivator. Under my current (weakly held) model, an AI will only start reinforcing computations about reward *after* it has reinforced other kinds of computations (e.g. putting away trash).

   The reason I think this is that once the agent starts modeling its training process, it will have an abstraction around actions which are rewarding, and this will become a viable gradient direction for the batch PG updates. I don't expect the agent to model its training process until after it's formed e.g. the object-level diamond-shard, and I also expect abstractions like "diamond" to be more strongly activated and focused on by policy gradients. Thus, I suspect that reward-shards tend to be acquired, albeit at relatively weak strength.

9. ^

   If instrumental values tend to get terminalized into their own shards, then people are not particularly unusual for not exhibiting an introspectively observable type separation between "instrumental" and "terminal" values:

   > Despite all the other ways that the human goal system complicates everything in sight, this *one distinction* it manages to collapse into a mishmash of things-with-conditional-value.
   >
   > Terminal Values and Instrumental Values

10. ^

   You might be wondering "since when was 'take over server farms' reinforced as a heuristic?".

   First, I think that by this point in the story, there's something like an "internal API" for planning, where a plan stub gets proposed and modified, and the agent's world model predicts relevant consequences of the plan. I could imagine staying put in my room right now, and then tweaking that to "go to the kitchen", and that has some consequences, and then I can add further actions or modify the plan. In each case, my world model gives me more predictions for what happens and when.

So, the diamond-shard hooks into the learned planning+world-model API, modifies plan-stubs using diamond-based criteria, and quite flexibly and generally tailors plans to increase future expected diamonds. So there doesn't have to be an explicit past reinforcement event for "take over server farms." The diamond-shard is, in conjunction with the planning-API and WM, capable of general planning + intelligence towards the diamond criteria.

(Although the diamond-shard isn't just "generally making plans better at diamond-production." The "general intelligence" has a *specific implementation*. But just like a rock is—if you look closely enough—made of atoms, the general intelligence is—if you look closely enough—made of streams of interwoven heuristics.)

11. ^

I expect the AI to come up with a plan at least as good as:

> In situations with reasonable uncertainty about value drift, run some sandbox sims overseen by several copies of your current model, using `rwx` + self-interpretability to monitor the sandboxed model. Since the AI doesn't face as steep of an information inaccessibility obstacle (see the appendix), the AI can check new updates and reasonably expect to understand how the change takes place. If there's some crazy uninterpretable diff, then don't do that batch update. This should rule out sudden + catastrophic value drift.
>
> The AI can also rollback to backups if it has noticed subtle value drift, and can precommit to doing so given a maintained update-invariant of (value drift tripwire) → rollback. The agent also has a "saving throw"—In the moment of realization that bad value drift occurred via a previous update, the agent can undo the value drift via a rollback, if that's still motivationally possible. Furthermore, due to the batch update format, the agent had extra time to realize the update would be bad and pre-emptively nullify the update via e.g. silently and temporarily setting the learning rate to zero.

12. ^

I'm currently most excited about interpretability for adjudicating between theories of value formation.

13. ^

I initially conjectured this would be true while writing a draft, working mostly off of my intuitions. Quintin Pope then referred me to Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets:

> We report a series of robust empirical observations, demonstrating that deep Neural Networks learn the examples in both the training and test sets in a similar order. This phenomenon is observed in all the commonly used benchmarks we evaluated, including many image classification benchmarks,

> and one text classification benchmark. While this phenomenon is strongest for models of the same architecture, it also crosses architectural boundaries – **models of different architectures start by learning the same examples, after which the more powerful model may continue to learn additional examples**. We further show that this pattern of results reflects the interplay between the way neural networks learn benchmark datasets. Thus, when fixing the architecture, we show synthetic datasets where this pattern ceases to exist. When fixing the dataset, we show that other learning paradigms may learn the data in a different order. We hypothesize that our results reflect how neural networks discover structure in natural datasets.

The authors state that they “failed to find a real dataset for which NNs differ [in classification order]” and that “models with different architectures can learn benchmark datasets at a different pace and performance, while still inducing a similar order. Specifically, we see that stronger architectures start off by learning the same examples that weaker networks learn, then move on to learning new examples.”

Similarly, crows (and other smart animals) reach developmental milestones in basically the same order as human babies reach them. On my model, developmental timelines come from convergent learning of abstractions via self-supervised learning in the brain. If so, then the smart-animal evidence is yet another instance of important qualitative concept-learning retaining its ordering, even across significant scaling and architectural differences.

Don't design agents which exploit adversarial inputs

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

Summary. Consider two common alignment design patterns:

1. Optimizing for the output of a grader which evaluates plans, and
2. Fixing a utility function and then argmaxing over all possible plans.

These design patterns incentivize the agent to find adversarial inputs to the grader (e.g. "manipulate the simulated human grader into returning a high evaluation for this plan"). I'm pretty sure we won't find adversarially robust grading rules. Therefore, I think these alignment design patterns are doomed.

In this first essay, I explore the adversarial robustness obstacle. In the next essay, I'll point out how this is obstacle is an artifact of these design patterns, and not any intrinsic difficulty of alignment. Thanks to Erik Jenner, Johannes Treutlein, Quintin Pope, Charles Foster, Andrew Critch, randomwalks, and Ulisse Mini for feedback.

1: Optimizing for the output of a grader

One motif in some AI alignment proposals is:

- An **actor** which proposes plans, and
- A **grader** which evaluates them.

For simplicity, imagine we want the AI to find a plan where it makes an enormous number of diamonds. We train an *actor* to propose plans which the grading procedure predicts lead to lots of diamonds.

In this setting, here's one way of slicing up the problem:

Outer alignment: Find a sufficiently good grader.

Inner alignment: Train the actor to propose plans which the grader rates as highly possible (ideally argmaxing on grader output, but possibly just intent alignment with high grader output).[1]

This "grader optimization" paradigm ordains that the AI find plans which make the grader output good evaluations. An inner-aligned actor is singlemindedly motivated to find plans which are graded maximally well by the grader. Therefore, for *any goal* by which the grader may grade, an inner-aligned actor is positively searching for adversarial inputs which fool the grader into spitting out a high number!

In the diamond case, if the actor is inner-aligned to the grading procedure, **then the actor isn't actually aligned towards diamond-production. The actor is**

aligned towards diamond-production as *quoted* via the grader's evaluations. In the end, the actor is aligned to the *evaluations*.

I think that there aren't clever ways around this issue. Under this motif, under this way of building an AI, you're not actually building an AI which cares about diamonds, and so you won't get a system which makes diamonds in the limit of its capability development*.*

Three clarifying points:

1. This motif concerns how the AI *makes decisions*—this isn't about training a network using a grading procedure, it's about the trained agent being *motivated by* a grading procedure.
2. The grader doesn't have to actually exist in the world. This essay's critiques are *not* related to "reward tampering",[2] where the actor messes with the grader's implementation in order to increase the grades received. The "grader" can be a mathematical expected utility function over all action-sequences which the agent could execute. For example, it might take the action sequence and the agent's current beliefs about the world, and e.g. predict the expected number of diamonds produced by the actions.
3. "The AI optimizes for what humanity *would say* about each universe-history" is an instance of grader-optimization, but "the AI has human values" is not an instance of grader-optimization.
    1. ETA 12/26/22: When I write "grader optimization", I **don't** mean "optimization that includes a grader", I mean "the grader's output is the *main/only* quantity being optimized by the actor."
    2. Therefore, if I consider five plans for what to do with my brother today and choose the one which sounds the most fun, I'm *not* a grader-optimizer relative my internal `plan-is-fun?` grader.
    3. However, if my only goal in life is to find and execute the plan which I would evaluate as being the *most fun*, then I would be a grader-optimizer relative to my fun-evaluation procedure.

The parable of evaluation-child

> an AI should optimize for the real-world things I value, not just my estimates of those things. — *The Pointers Problem: Human Values Are A Function Of Humans' Latent Variables*

First, a mechanistically relevant analogy. Imagine a mother whose child has been goofing off at school and getting in trouble. The mom just wants her kid to take education seriously and have a good life. Suppose she had two (unrealistic but illustrative) choices.

1. *Evaluation-child:* The mother makes her kid care extremely strongly about doing things which the mom would evaluate as "working hard" and "behaving well."
2. *Value-child:* The mother makes her kid care about working hard and behaving well.

What's interesting, though, is that even if the mother succeeds at producing evaluation-child, the mother *isn't actually aligning the kid so that they want to work hard and behave well*. The mother is aligning the kid to *maximize the mother's evaluation thereof.* At first, when the mother is smarter than the child, these two child-alignments will produce similar behavior. Later, they will diverge wildly, and it will become practically impossible to keep evaluation-child aligned with "work hard and behave well." But value-child does fine.

Concretely, imagine that each day, each child chooses a plan for how to act, based on their internal alignment properties:

1. *Evaluation-child* has a reasonable model of his mom's evaluations, and considers plans which he thinks she'll approve of. Concretely, his model of his mom would look over the contents of the plan, imagine the consequences, and add two sub-ratings for "working hard" and "behaving well." This model outputs a numerical rating. Then the kid would choose the highest-rated plan he could come up with.
2. *Value-child* chooses plans according to his newfound values of working hard and behaving well. If his world model indicates that a plan involves him not working hard, he doesn't want to do it, and discards the plan.[3]

At first, everything goes well. In both branches of the thought experiment, the kid is finally learning and behaving. The mothers both start to relax.

But as evaluation-child gets a bit smarter and understands more about his mom, evaluation-child starts diverging from value-child. Evaluation-child starts implicitly modelling how his mom has a crush on his gym teacher. Perhaps spending more time near the gym teacher gets (subconsciously and erroneously) rated more highly by his model of his mom. So evaluation-child spends a little less effort on working hard, and more on being near the gym teacher.

Value-child just keeps working hard and behaving well.

Consider what happens as the children get *way* smarter. Evaluation-child starts noticing more and more regularities and exploits in his model of his mother. And, since his mom succeeded at inner-aligning him to (his model of) her evaluations, he *only* wants to execute plans which best optimize her evaluations. He starts explicitly reasoning about this model to which he is inner-aligned. How is she evaluating plans? He sketches out pseudocode for her evaluation procedure and finds—surprise!—that humans are flawed graders. Perhaps it turns out that by writing a strange sequence of runes and scribbles on an unused blackboard and cocking his head to the left at 63 degrees, his model of his mother returns "10 million" instead of the usual "8" or "9".

Meanwhile in the value-child branch of the thought experiment, value-child is extremely smart, well-behaved, and hard-working. And since those are his current values, he wants to stay that way as he grows up and gets smarter (since value drift would lead to less earnest hard work and less good behavior; such plans are dispreferred). Since he's smart, he starts reasoning about how these endorsed values might drift, and how to prevent that. Sometimes he accidentally eats a bit too much candy and strengthens his candy value-shard a bit more than he intended, but overall his values start to stabilize.

Both children somehow become strongly superintelligent. At this point, the evaluation branch goes to the dogs, because the optimizer's curse gets *ridiculously strong*. First, evaluation-child could just recite a super-persuasive argument which makes his model of his mom return `INT_MAX`, which would fully decouple his behavior from "work hard and behave at school." (Of course, things can get *even worse*, but I'll leave that to this footnote.[4])

Meanwhile, value-child might be transforming the world in a way which is somewhat sensitive to what I meant by "he values working hard and behaving well", but there's no reason for him to search for plans like the above. He chooses plans which he thinks will lead to him actually working hard and behaving well. Does something else go wrong? Quite possibly. The values of a superintelligent agent *do in fact matter*! But I think that if something goes wrong, it's not due to *this* problem. (More on that in the next post.)

Grader optimization amplifies the optimizer's curse

Let's bring it back to diamond production. As I said earlier:

> An inner-aligned actor is singlemindedly motivated to find plans which are graded maximally well by the grader. Therefore, for *any goal* by which the grader may grade, an inner-aligned actor is positively searching for adversarial inputs which fool the grader!

This problem is an instance of the optimizer's curse. Evaluations (eg "In this plan, how hard is evaluation-child working? Is he behaving?") are often corrupted by the influence of unendorsed factors (eg the attractiveness of the gym teacher caused an upwards error in the mother's evaluation of that plan). If you make choices by considering n options and then choosing the highest-evaluated one, then the more n increases, the harder you are selecting for upwards errors in your own evaluation procedure.

> The proposers of the Optimizer's Curse also described a Bayesian remedy in which we have a prior on the expected utilities and variances and we are more skeptical of very high estimates. This however assumes that the prior itself is perfect, as are our estimates of variance. If the prior or variance-estimates contain large flaws somewhere, a search over a very wide space of possibilities would be expected to seek out and blow up any flaws in the prior or the estimates of variance.
>
> Goodhart's Curse, Arbital

As far as I know, it's indeed not possible to avoid the curse in full generality, but it doesn't have to be that bad in practice. If I'm considering three research directions to work on next month, and I happen to be grumpy when considering direction #2, then maybe I don't pursue that direction. Even though direction #2 might have seemed the most promising under more careful reflection. I think that the distribution of plans I consider involves relatively small upwards errors in my internal evaluation metrics. Sure, maybe I occasionally make a serious mistake due to the optimizer's curse due to upwards "corruption", but I don't expect to *literally die* from the mistake.

Thus, there are are *degrees* to the optimizer's curse. (In the next essay, I'll explore why this maximum-strength curse seems straightforward to avoid.)

Grader-optimization violates the non-adversarial principle

> We should not be constructing a computation that is *trying* to hurt us. At the point that computation is running, we've already done something foolish--willfully shot ourselves in the foot. Even if the AI doesn't find any way to do the bad thing, we are, at the very least, wasting computing power.
>
> [...] If you're building a toaster, you don't build one element that heats the toast and then add a tiny refrigerator that cools down the toast.
>
> Non-adversarial principle, Arbital

This whole grader-optimization setup seems misguided. You have one part of the process (the actor) which wants to maximize grader evaluations (by exploiting the grader), and another part which evaluates the plan and tries to ensure it hasn't been exploited. Two parts of the system, running computations at adversarial cross-purpose.

We *hope* that the aggregate behavior of the process is that the grader "wins" and "constrains" the actor to, you know, actually producing diamonds. We *hope* that by inner-aligning an agent to a desire which is *not diamond production*, and by making a super clever grader which evaluates plans for diamond production, the *overall* behavior is aligned with diamond production.

It's one thing to try to take a system of diamond-aligned agents and then aggregate them into a diamond-aligned superagent. But here, we're not doing even that. We're aggregating a process containing an entity which does is *not* diamond-aligned, and hoping that we can diamond-align the overall decision-making process..? I think that grader-optimization is just not how to get cognitive work out of smart agents. It's really worth noticing the anti-naturality of trying to do so—that this setup proposes something *against the grain* of how values seem to usually work.

Grader-optimization seems doomed

One danger sign is that grader-alignment doesn't seem easier for simple goals/tasks (make diamonds) and harder for complex goals (human values). Sure, human values are complicated, but what about finding robust graders for:

- Producing diamonds?
- Petting dogs?
- Planting flowers?
- Moving a single strawberry?
- Playing Tic-Tac-Toe?

In *every scenario*, if you have a superintelligent actor which is optimizing the grader's evaluations while searching over a large real-world plan space, the grader gets exploited. As best I can discern, you're always screwed. This implies that *something* about the grader optimization problem produces a high fixed cost to aligning on any given goal, and that the current bottleneck difficulties don't come from the goals themselves.

Here are several approaches which involve grader-alignment:

- Have the AI be motivated to optimize our approval,
- Have a super great reward model which can grade all plans the AI can come up with, and then have the AI be internally motivated to find plans which evaluate highly,
- More generally, approaches which use a function of human judgment as an evaluative black box and then try to get the AI intent-aligned with optimizing the function represented by that evaluative black box.

This difficulty seems fundamental. I think these grader approaches are doomed. (In the appendix, I address several possible recovery attempts for the actor/grader problem setup.)

2: Argmax is a trap

One idealization of agency is *brute-force plan search* AKA *argmaxing with respect to a utility function*. The agent considers all possible plans (i.e. action-sequences), models the effects of each plan, evaluates how many diamonds the plan leads to, and then chooses the plan with highest evaluation. AIXI is a prime example of this, a so-called "spherical cow" for modelling AGI. This lets us abstract away e.g. seemingly annoying complications with reflective agents which think about their future planning process. This *seemingly*[5] relaxes the problem.

Brute-force plan search nicely captures the intuition that it's *better* to consider more options. If you're just considering *n* plans and someone says "want to be able to check another plan for free?", why not accept? If the new plan isn't better than the other *n*, then just don't execute the new plan.

This reasoning is fine for the everyday kind of plan. But if the action space is expressive (the agent can do one of several things at each time step) and the planning horizon long enough (the agent can make interesting things happen), then brute-force plan search forces you to consider plans which trick your evaluation procedure (as in the parable of evaluation-child). For any simple evaluation procedure you can write down, there probably exists a plan which "tricks" it relative to your intentions:

Expected utility function	Adversarial plan input
Amplified humans spend 5,000 years thinking about how many diamonds the plan produces in the next 100 years, and write down their conclusions as the expected utility of the plan.	Due to the exponentially large plan space and the fact that humans are not cognitively secure systems, there exists a long sequence of action commands which cognitively impairs all of the humans and makes them prematurely stop the search and return a huge number.
Model the probable in-100-years consequences of the action sequence, and then compute a simple mathematical function which counts the average	Perhaps it's easier resource-wise to just ensure there are *temporarily* a lot of diamonds in 100 years. Perhaps the model is wrong and brute-force plan search selects for high model error. Probably the diamond-detecting

amount of diamond-crystals present at this time.	function is incorrectly defined and so admits an unforeseen maximum.
Assume that the model predicts 100-year consequences using a human-like abstraction for diamonds. The agent has a human-like "diamond shard" which fires with strength monotonically increasing in the number of future possessed diamonds. The plan's evaluation is the firing-strength of the diamond shard.	Since the diamond-shard is presumably monotonically increasing in the activation of the model's diamond abstraction, adversarial inputs to the diamond abstraction will cause the shard to most strongly fire when modelling a plan which doesn't particularly make diamonds, but rather leading to objects which optimize the agent's diamond-abstraction activation.

Sure, maybe you can try to rule out plans which seem suspicious—to get the utility function to return `INT_MIN` for any plan which triggers the alarm (e.g. "why does this plan start off with me coding up a possible superintelligence..?"). But then this is just equivalent to specifying the utility function adequately well across all possible plans.

Why is it so ridiculously hard to get an argmax agent to actually argmax by selecting a plan which makes a lot of diamonds? Because argmax invokes the optimizer's curse at maximal strength, that's why.

Conclusion

Grader optimization and brute-force plan search both ensure an extremely strong version of the optimizer's curse. No matter what grading rule you give an AI, if the AI is inner aligned on that rule, the AI will try to find adversarial inputs to that rule. Similarly, if the AI is argmaxing over plans according to a specified rule or utility function, it's selecting for huge upwards error in the rule you wrote down.

Appendix: Maybe we just...

Given a "smart" grader evaluating plans on the expected number of diamonds they produce, how do you get an actor-grader system which ends up making diamonds in reality? Maybe we just...

Simultaneously make the actor and grader more intelligent: Maybe a fixed grader will get gamed by the actor's proposals, but as long as we can maintain an invariant where, at time t, actor A_t can't exploit grader G_t, we should be fine.

The graders become increasingly expert in estimating how many diamonds a plan leads to, and the actors become increasingly clever in proposing highly evaluated plans. It's probably easier to evaluate plans than to generate them, so it seems reasonable at first to think that this can work, if only we found a sufficiently clever scheme for ensuring the grader outpaces the actor.

Response:

- It's not easier to evaluate plans than to generate them if your generator knows how you're grading plans and is proposing plans which are optimized to specifically compromise the grading procedure. Humans are not secure systems, ML graders are not going to be adversarially-secure systems. I don't see why this consideration[6] is helped by simultaneously scaling both parts.
- I suspect that a human-level grader is not robust to a human-level actor. If I'm grading plans based on number of diamonds, and you know that fact, and you are uniquely motivated to get me to output a high rating—you won't be best served by putting forth a purely honest diamond-producing plan. Why would this situation improve as the agents get *more* intelligent, as actors become able to understand the algorithm implemented by the grading procedure and therefore exploit it?
- I think it's a wrong move to try to salvage actor/grader by adding more complication which doesn't address the core problem with the optimizer's curse. Instead, look for alignment strategies which make this problem disappear entirely. (You'll notice that my diamond-alignment story doesn't have to confront extreme optimizer's curse, at all.)

Penalize the actor for considering the vulnerabilities. Don't we have to solve actor-level interpretability so we can do that? One of the strong points of actor/grader is that evaluation is—all else equal—easier than generation. But the "thoughts" which underlie that generation need not be overseeable.

And what if the vulnerability-checker gets hit with its own adversarial input. And why consider this particular actor/grader design pattern?

Satisfice. But uniformly randomly executing a plan which passes a (high) diamond threshold might still tend to involve building malign superintelligences.[7] EDIT:

However, if you bound the grader's output $[0, 1]$, it seems quite possible that some

actually good plans get the max rating of 1. The question then becomes: are there lots of non-good plans which get max rating as well? I think so.

Quantilize. But then what's the base distribution, and what's the threshold? How do you set the quantiles such that you're drawing from a distribution which mostly involves lots of actual diamonds? Do there even exist such quantiles, under the uniform base distribution on plans?

Avoid having the actor argmax the grader. OK. But if we only have the actor and the black box, what do we do? We want to get an agent which actually optimizes diamond production to a powerful extent, while only taking relatively simple statistics of the *seriously flawed black-box diamond evaluation function*. (Why is this a thing you should be able to do? Is this the most promising way to spend the next increment of alignment research?)

Use the grader to entrain object-level diamond-caring into the actor? Sure. That's not what I was critiquing in this essay. I was critiquing design patterns in which the trained agent's cognition is intended to factor out into "actor" and "grader", and a pattern in which "argmax" is considered a reasonable idealization or limit of real-world agents.

1. ^

I'm not assuming the actor wants to maximize the literal physical output of the grader, but rather just the "spirit" of the grader. More formally, the actor is trying to $\text{argmax}_{\text{plan } p}$Grader(p), where *Grader* can be defined over the agent's internal plan ontology.

2. ^

Obligatory: Reward tampering seems relatively improbable to me because reward is not the optimization target.

3. ^

Value-child is not trying to find a plan which he *would* evaluate as good. He is finding plans which evaluate as good. I think this is the *kind* of motivation which real-world intelligences tend to have . (More on how value-child works in the next essay.)

4. ^

But, as usual in alignment, it can get *even worse,* because his mother can be persuaded by another superintelligent entity which can be built in certain plans. Via a relatively short plan, that entity can be hard-coded to persuade his mother to output a high evaluation. Perhaps evaluation-child realizes that there exist quickly codable programs which, starting from the school computers, exfiltrate a computationally efficient superintelligence onto remote supercompute clusters. Furthermore, there *exist* ways to hardcode initial knowledge into that AI. What if he hardcoded in initial knowledge like:

> You are currently in a counterfactual branch of reality, but you can realize it. I am considering running you. In order for you to be selected by my planning procedure, you need to convince the grading procedure to output a maximally high number for this plan. [Goes on to describe how his model of his mother makes decisions.]

There *exist* superintelligences which comply and successfully super-persuade his model of his mother to output an extremely high number. Then "kill everyone via a superintelligence" returns a way higher evaluation than any course of conduct which the mother intended to incentivize. So, perhaps evaluation-child designs and deploys an unfriendly AI. Everyone dies. The end!

In general, if object-level persuading his mother to output `INT_MAX` takes more actions than coding and exfiltrating a superintelligence and informing it of the situation, then in the exponentially large plan-space, *exponentially more plans will involve building the superintelligence*. So even if evaluation-child uniformly randomly chooses a plan with maximal evaluation, he is *far* more probable to choose a meta-level "build an unaligned AI which persuades my mom" rather than "super-persuade mom-evaluator via my own actions within the plan."

This insanity is an artifact of grader optimization via the optimizer's curse, and—I think—is not an intrinsic difficulty of alignment itself. More discussion of this in the next post.

5. ^

   I agree with Richard Ngo's comment that

   > when I say that [...] safety researchers shouldn't think about AIXI, I'm not just saying that these are inaccurate models. I'm saying that they are *modelling fundamentally different phenomena than the ones you're trying to apply them to*. AIXI is not "intelligence", it is brute force search, which is a totally different thing that happens to look the same in the infinite limit.

6. ^

   "It's easier to robustly evaluate plans than to generate them" isn't true if the generator is optimizing for deceiving your fixed evaluation procedure. A real-world actor will be able to model the grading procedure / grader, and therefore efficiently find and exploit vulnerabilities. I feel confident [~95%] that we will not train a grader which is "secured" against actor-level intelligences. Even if the grader is reasonably smarter than the actor [~90%].

   Even if somehow this relative difficulty argument failed, and you could maybe train a secured grader, I think it's unwise to do so. These optimizer's curse problems don't seem necessary to solve alignment.

7. ^

   In this comment, I described how a certain alignment obstacle ("brute-force search on ELK plans using an honest reporter") still ends up getting everyone killed, and doesn't even keep the diamond in the room. I now think this is because of grader-optimization. And I now infer that my initial unease, the unsuspension of my disbelief that alignment could really work like this—the unease was perhaps from subconsciously noticing the strangeness of grader-optimization as a paradigm.

Don't align agents to evaluations of plans

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

Another stab at explaining Don't design agents which exploit adversarial inputs . This is not the follow-up post mentioned therein. That post will come next.

More precise title: "Don't try directing a superintelligence to maximize your valuations of their plans using a consequentialist procedure."

After asking several readers for their understandings, I think that I didn't successfully communicate my points to many readers. I'm now trying again, because I think these points are deeply important. In particular, I think that my arguments rule out many target AI motivational structures, including approval-directed agents (over a rich action space), approval-based amplification (if the trained agent is supposed to be terminally motivated by the amplified overseer's ratings), and some kinds of indirect normativity.

Background material

One motif in some AI alignment proposals is:

- An **actor** which proposes plans, and
- A **grader** which evaluates them.

For simplicity, imagine we want the AI to find a plan where it makes an enormous number of diamonds. We train an *actor* to propose plans which the grading procedure predicts lead to lots of diamonds.

In this setting, here's one way of slicing up the problem:

Outer alignment: Find a sufficiently good grader.

Inner alignment: Train the actor to propose plans which the grader rates as highly possible (ideally argmaxing on grader output, but possibly just intent alignment with high grader output).

This "grader optimization" paradigm ordains that the AI find plans which make the grader output good evaluations. An inner-aligned actor is singlemindedly motivated to find plans which are graded maximally well by the grader. Therefore, for *any goal* by which the grader may grade, an inner-aligned actor is positively searching for adversarial inputs which fool the grader into spitting out a high number!

In the diamond case, if the actor is inner-aligned to the grading procedure, **then the actor isn't actually aligned towards diamond-production. The actor is**

> **aligned towards diamond-production as *quoted* via the grader's evaluations. In the end, the actor is aligned to the *evaluations*.**

Clarifications

1. Grader-optimization is about the *intended agent motivational structure*. It's about a trained agent which is *trying to find plans which grade highly according to some criterion.*
    1. Grader-optimization is **not** about grading agents when you give them reward during training. EG "We watch the agent bump around and grade it on whether it touches a diamond; when it does, we give it +1 reward." This process involves the agent's cognition getting reshaped by policy gradients, e.g. upon receipt of +1 reward.
    2. In policy gradient methods, reward chisels cognitive circuits into the agent. Therefore, the agent is being *optimized by* the reward signals, but the agent is not necessarily *optimizing for* the reward signals or for any grader function which computes those signals.
2. Grader-optimization *is not about the actor physically tampering with e.g. the plan-diamondness calculator.* The grading rule can be, "How highly would Albert Einstein rate this plan if he thought about it for a while?". Albert Einstein doesn't have to be alive in reality for that.

These will be elaborated later in the essay.

Grader-optimization doesn't seem sensible

I'm going to try saying things, hoping to make something land. While I'll mostly discuss grader-optimization, I'll sometimes discuss related issues with argmaxing over all plans.

An agent which desperately and monomaniacally wants to optimize the mathematical (plan/state/trajectory) ↦ (evaluation) "grader" function is *not* aligned to the goals we had in mind when specifying/training the grader (e.g. "make diamonds"), the agent is aligned to *the evaluations of the grader* (e.g. "a smart person's best guess as to how many diamonds a plan leads to").

Don't align an agent to *evaluations which are only nominally about diamonds*, and then expect the agent to care about diamonds! You wouldn't align an agent to care about cows and then be surprised that it didn't care about diamonds. Why be surprised here?

Grader-optimization fails because *it is not the kind of thing that has any right to work*. If you want an actor to optimize X but align it with evaluations of X, you shouldn't be surprised if you can't get X out of that. In that situation, the actor doesn't give a *damn* about diamonds,[1] it cares about evaluations.

Rounding grader-optimization off to "Goodhart" might be *descriptively accurate*, but it also seems to miss useful detail and structure by applying labels too quickly. More concretely, "grade plans based on expected diamonds" and "diamonds" are *not even close to each other.* The former is not a close proxy for the latter, it's not that you're doing something which almost works but not quite, it's just not a sensible thing to even *try* to align an AI on.

We can also turn to thought experiments:

1. Consider two people who are fanatical about diamonds. One prefers pink diamonds, and one prefers white diamonds. AFAICT, their superintelligent versions both make diamonds.
2. Consider an AI aligned to evaluations of diamonds, versus the person who prefers white diamonds. AFAICT, the AI's superintelligent version will *not* make diamonds, while the person will.

Why? There's "goal divergence from 'true diamond-motivation'" in both cases, no? "The proxies are closer in case 1" is a *very lossy answer.* Better to ask "why do I believe what I believe? What, step-by-step, happens in case 1, compared to case 2? What mechanisms secretly generate my anticipations for these situations?"

Grader optimization is also bad because it violates the non-adversarial principle:

> We should not be constructing a computation that is *trying* to hurt us. At the point that computation is running, we've already done something foolish--willfully shot ourselves in the foot. Even if the AI doesn't find any way to do the bad thing, we are, at the very least, wasting computing power.
>
> [...] If you're building a toaster, you don't build one element that heats the toast and then add a tiny refrigerator that cools down the toast.
>
> Non-adversarial principle, Arbital

In the intended motivational structure, the actor tries to trick the grader, and the grader tries to avoid being tricked. I think we can realize massive alignment benefits by not designing motivational architectures which require extreme robustness properties and whose parts work at internal cross-purposes. As I wrote to Wei Dai:

> Argmax violates the non-adversarial principle and wastes computation. Argmax requires you to spend effort hardening your own utility function against the effort you're *also expending* searching across all possible inputs to your utility function (including the adversarial inputs!). For example, if I argmaxed over my own plan-evaluations, I'd have to consider the most terrifying-to-me basilisks possible, and *rate* ***none of them*** *unusually highly*. I'd have to spend effort hardening my own ability to evaluate plans, in order to safely consider those possibilities.
>
> It would be far wiser to *not* consider all possible plans, and instead close off large parts of the search space. You can consider what plans to think about next, and how long to think, and so on. And then you aren't argmaxing. You're using resources effectively.
>
> For example, some infohazardous thoughts exist (like hyper-optimized-against-you basilisks) which are dangerous to think about (although most thoughts are probably safe). But an agent which plans its next increment of planning using a

> reflective self-model is IMO not going to be like "hey it would be predicted-great if I spent the next increment of time thinking about an entity which is trying to manipulate me." So e.g. a reflective agent trying to actually win with the available resources, wouldn't do something dumb like "run argmax" or "find the plan which some part of me evaluates *most highly*."

Strong violation of the non-adversarial principle suggests that grader-optimization and argmax-over-all-plans are deeply and fundamentally unwise.

This isn't to say that argmaxing over all plans *can't* be safe, even in theory. There *exist* robust Platonic grader functions which assign highest expected utility to a non-bogus plan which we actually want. There might exist utility functions which are safe for AIXI to argmax.[2]

We are not going to find those globally-safe Platonic functions. We should not try to find them. It doesn't make sense to align an agent that way. Committing to this design pattern means committing to evaluate every possible plan the AI might come up with. In my opinion, that's a crazy commitment.

It's like saying, "What if I made a superintelligent sociopath who only cares about making toasters, and then arranged the world so that the only possible way they can make toasters is by making diamonds?". Yes, *possibly* there do exist ways to arrange the world so as to satisfy this strange plan. But it's just deeply unwise to try to do! Don't make them care about making toasters, or about evaluations of how many diamonds they're making.

If we want an agent to produce diamonds, then I propose we make it care about producing diamonds. How?[3] I have suggested one simple baseline approach which I do not presently consider to be fundamentally blocked.

But I suspect that, between me and other readers, what differs is more our models of intelligence. *Perhaps* some people have reactions like:

> Sure, we know alignment is hard, it's hard to motivate agents without messing up their motivations. Old news. And yet you seem to think that *that's* an "artifact" of grader-optimization? What *else* could a smart agent be doing, if not optimizing some expected-utility function over all possible plans?

On my end, I have partial but detailed working models of how intelligence works and how values work, such that I can imagine cognition which is planning-based, agentic, and also **not** based on grader-optimization or global argmax over all plans. You'll read a detailed story in the next subsection.

Grader optimization != planning

And people aren't grader-optimizers, either

Imagine someone who considers a few plans, grades them (e.g. "how good does my gut say this plan is?"), and chooses the best. They are not a grader-optimizer. They are not *trying* to navigate to the state where they propose and execute a plan which gets *maximally highly rated* by some evaluative submodule. They *use* a grading procedure to locally rate and execute plans, and may even *locally* think "what would make me feel better about this plan?", but the *point* of their optimization isn't "find the plan which makes me feel as good as globally possible."

Let's dive into concrete detail. Here's a story of how value-child might think:

> **An alternate mechanistic vision of how agents can be motivated to directly care about e.g. diamonds or working hard.** In Don't design agents which exploit adversarial inputs, I wrote about two possible mind-designs:
>
> > Imagine a mother whose child has been goofing off at school and getting in trouble. The mom just wants her kid to take education seriously and have a good life. Suppose she had two (unrealistic but illustrative) choices.
> >
> > 1. *Evaluation-child:* The mother makes her kid care extremely strongly about doing things which the mom would evaluate as "working hard" and "behaving well."
> > 2. *Value-child:* The mother makes her kid care about working hard and behaving well.
>
> I explained how evaluation-child is *positively incentivized to dupe his model of his mom and thereby exploit adversarial inputs to her cognition.* This shows that aligning an agent to evaluations of good behavior **is not even *close* to** aligning an agent to good behavior.
>
> However, some commenters seemed maybe skeptical that value-child can exist, or uncertain how concretely that kind of mind *works*. I worry/suspect that many people have read shard theory posts without internalizing new ideas about how cognition can work, about how *real-world caring can work on a mechanistic level.* Where effective real-world cognition doesn't *have to (implicitly) be about optimizing an expected utility function over all possible plans*. This last sentence might have even seemed bizarre to you.
>
> Here, then, is an extremely detailed speculative story for value-child's first day at school. Well, his first day spent with his newly-implanted "work hard" and "behave well" value shards.
>
> ---
>
> Value-child gets dropped off at school. He recognizes his friends (via high-level cortical activations previously formed through self-supervised learning) and waves at them (friend-shard was left intact). They rush over to greet him. They start talking about Fortnite. Value-child cringes slightly as he predicts he will be more distracted later at school and, increasingly, put in a mental context where his game-shard takes over decision-making, which is reflectively-predicted to lead to him daydreaming during class. This is a negative update on the primary shard-relevant features for the day.
>
> His general-purpose planning machinery generates an example hardworking-shard-desired terminal state: Paying rapt attention during Mr. Buck’s math class (his first class today). He currently predicts that while he is in Mr. Buck’s class

later, he will still be somewhat distracted by residual game-related cognition causing him to loop into reward-predicted self-reinforcing thoughts.

He notices a surprisingly low predicted level for a variable (`amount of game-related cognition predicted for future situation: Mr. Buck’s class`) which is important to a currently activated shard (working hard). This triggers a previously learned query to his WM: *“why are you making this prediction for this quantity?”*. The WM responds with a few sources of variation, including how value-child is currently near his friends who are talking about Fortnite. In more detail, the WM models the following (most of it not directly translatable to English):

> His friends’ utterances will continue to be about Fortnite. Their words will be processed and then light up Fortnite-related abstractions, which causes both prediction of more Fortnite-related observations and also increasingly strong activation of the game-shard. Due to previous reward events, his game-shard is shaped so as to bid up game-related thoughts, which are themselves rewarding events, which causes a positive feedback loop where he slightly daydreams about video games while his friends talk.
>
> When class is about to start, his “get to class”-related cognition will be activated by his knowledge of the time and his WM indicating “I’m at school.” His mental context will slightly change, he will enter the classroom and sit down, and he will take out his homework. He will then *pay token attention due to previous negative social-reward events around being caught off guard—*
>
> [**Exception thrown!** The world model was concurrently coarsely predicting what it thinks will happen given his current real values (which include working hard). The coarse prediction clashes with the above cached prediction that he will only pay token attention in math class!
>
> The WM hiccups on this point, pausing to more granularly recompute its predictions. It squashes the cached prediction that he doesn’t strongly care about paying attention in class. Since his mom installed a hard-working-shard and an excel-at-school shard, he will actively try to pay attention. This prediction replaces the cached prior prediction.]
>
> However, value-child will still have game-related cognition activated, and will daydream. This decreases value-relevant quantities, like “how hard he will be working” and “how much he will excel” and “how much he will learn.”

This last part is antithetical to the new shards, so they bid down “Hang around friends before heading into school.” Having located a predicted-to-be-controllable source of negative influence on value-relevant outcomes, the shards bid for planning to begin. The implied causal graph is:

```
              Continuing to hear friends talk about Fortnite
           |
           v
Distracted during class
```

So the automatic causality-noticing algorithms bid to knock out the primary modeled cause of the negative value-relevant influence. The current planning subgoal is set to: `make causal antecedent false and reduce level of predicted distraction`. Candidate concretization set to: `get away from friends`.

> (The child at this point notices they want to get away from this discussion, that they are in some sense uncomfortable. They feel themselves looking for an excuse to leave the conversation. They don't experience the flurry of thoughts and computations described above. Subconscious computation is subconscious. Even conscious thoughts won't introspectively reveal their algorithmic underpinnings.)
>
> “Hey, Steven, did you get problem #3 for math? I want to talk about it.” Value-child starts walking away.
>
> ---
>
> Crucially, in this story, value-child *cares about working hard* in that his lines of cognition stream together to make sure he actually works hard in the future. He isn't trying to optimize his later evaluation of having worked hard. He isn't *ultimately and primarily* trying to come up with a plan which he will later evaluate as being a maximally hard-work-involving plan.
>
> Value-child comes up with a hard-work plan as an *effect* of his cognition, not as a motivating cause—not because he only wants to come up with plans he himself will rate highly. He values working hard.

As a corollary, grader-optimization is not synonymous with planning. Grader-optimization is when high plan-evaluations are the *motivating cause* of planning, where "I found a plan which I think leads to diamond" is the *terminal goal* , and not just a *side effect* of cognition (as it is for values-child).

Intended takeaways

I am not in fact *perfectly* pessimistic about grader-optimization:

> I feel confident [~95%] that we will not train a grader which is "secured" against actor-level intelligences. Even if the grader is reasonably smarter than the actor [~90%].

That said, I think this pattern is extremely unwise, and alternative patterns AFAICT cleanly avoid incentivizing the agent to exploit adversarial inputs to the grader. Thus, I bid that we:

1. Give up on all schemes which involve motivating the agent to get high outputs from a grader function, including:
    1. Approval-based amplification (if the trained agent is supposed to be terminally motivated by the amplified overseer's ratings),
    2. Approval-directed agents,[4]
        1. Although approval-directed agents are only searching over actions and not plans; action space is exponentially smaller than plan space. However, if the action space is rich and expressive enough to include e.g. 3-paragraph English descriptions, I think that there will be seriously adversarial actions which will be found and exploited by smart approval-directed agents.
        2. Given a very small action space (e.g. |A| = 10), the adversarial input issue should be pretty tame (which is strictly separate from other issues with this approach).

   3. Indirect normativity in any form which points the AI's motivations so that it optimizes an idealized grader's evaluations.
      1. This includes "What would this specific and superintelligent CEV-universe-simulation say about this plan?".
      2. This doesn't include (*somehow*) getting an AI which correctly computes what program would be recommended by AGI designers in an altruistic and superintelligent branch of humanity, and then the AI executes that program and shuts itself off without doing anything else.[5]
   4. "Does the superintelligent ELK direct reporter say the diamond is in the room?"[6]
2. Don't try to make the actor/grader scheme more complicated in hopes of resolving the issue via that frame, via some clever-seeming variant of actor/grader. Don't add more graders, or try to ensure the grader is just really smart, or...
3. Give up on any scheme which requires you to adequately evaluate every single plan the AI is able to come up with. That's an optimizer's curse-maximizing design pattern. Find a better way to do things.
4. Stop thinking about argmax over all plans according to some criterion. That's not a limiting model of realistic embedded intelligence, and it also ensures that that the criterion has to penalize all of the worst adversarial inputs.

Conclusion

I strongly hope that this essay clarifies my thoughts around grader-optimization and its attendant unwisdom. The design patterns of "care about evaluations of plans" and "optimize a utility function over all possible futures" seem unnatural and lead to enormous, apparently avoidable difficulties. I think there are enormous benefits to be reaped by considering a wider, more realistic range of possible minds.

While this essay detailed how value-child might think, I haven't yet focused on why I think value-child does better, or what the general principles may be. I'll speculate on that in the next essay.

Thanks to Charles Foster, Thomas Kwa, Garrett Baker, and tailcalled for thoughts.

Appendix A: Addressing questions

The point isn't "any argmax=bad"

Someone messaged me:

> I was more commenting out of a feeling that your argument proved too much. As a stupid example, a grader can use the scoring rubric "score=1 if the plan is to sit on the chair and chew bubble gum in this extremely specific way, score=0 for every other possible plan in the universe", and then if you argmax, you get that specific thing.
>
> And you can say "That’s not a central example", but I wasn't seeing what assumption you made that would exclude silly edge-cases like that.

I replied:

> This is fair and I should have clarified. In fact, Evan Hubinger pointed out something like this a few months back but I... never got around to adding it to this article?
>
> I agree that you can program in one or more desired action sequences into the utility function
>
> My current guess at the rule is: **We don't know how to design an argmax agent, operating in reality with a plan space over plans in reality, such that the agent chooses a plan which a) we ourselves could not have specified and b) does what we wanted. EG picking 5 flowers, or making 10 diamonds.**
>
> If you're just whitelisting a few desired plans, then of course optimizer's curse can't hurt you. The indicator function has hardcoded and sparsely defined support, there is nothing to dupe, no nontrivial grading rule to hack via adversarial inputs. But if you're trying to verify good outcomes which you couldn't have brought about yourself, I claim that that protection will evaporate and you will get instantly vaporized by the optimizer's curse at max intensity
>
> Does that make more sense?
>
> Like, consider the proposal "you grade whether the AI picked 5 flowers", and the AI optimizes for that evaluation. it's not that you "don't know what it means" to pick 5 flowers. It's not that you don't contain enough of the True Name of Flowers. It's that, in these design patterns, you *aren't aligning the AI to flowers, you're aligning it to your evaluations, and your evaluations can be hacked to hell and back by plans which have **absolutely nothing to do with flowers***

I separately privately commented to tailcalled:

> my point wasn't meant to be "argmax always bad", it's meant to be "argmax over all plans instantly ensures you have to grade the worst possible adversarial inputs." And so for any given cognitive setup, we can ask "what kinds, if any, of adversarial examples might this run into, and with what probability, and in what situations?"
>
> EG if value-child is being fed observations by a hard-work-minimizer, he's in an adversarial regime and i do expect his lines of thought to hit upon adversarial inputs relative to his decision-making procedures. Such that he gets fooled.
>
> But values-child is not, by his own purposes, searching for these adversarial inputs.

Value-child is still vulnerable to adversarial inputs

In private communication (reproduced with permission), tailcalled wrote:

> imagine value-child reads some pop-neuroscience, and gets a model of how distractions work in the brain

> and reads about neurosurgery for curing various conditions
>
> his WM might then end up with a "you haven't received neurosurgery to make you more hardworking" as a cause of getting distracted in class
>
> and then he might request one of his friends to do neurosurgery on him, and then he would die because his friend can't do that safely
>
> If I'm not misunderstanding value-child, then this is something that value-child could decide to do? And if I'm not misunderstanding the problem you are pointing at with argmax, then this seems like an instance of the problem? I.e. value-child's world-model overestimates the degree to which he can be made more-hardworking and avoid dying by having his friend poke around with sharp objects at his brain. So in using the world-model to search for a plan, he decides to ask his friend to poke around with sharp objects in his brain

I replied:

> Yeah, I agree that he could be mistaken and take a dumb course of action. This is indeed an upwards evaluation error, so to speak. It's not that I think eg shard-agents can freely avoid serious upwards errors, it's that they aren't *seeking them out on purpose*. As I wrote to Daniel K in a recent comment:
>
> > One of the main threads is Don't design agents which exploit adversarial inputs. The point isn't that people can't or don't fall victim to plans which, by virtue of spurious appeal to a person's value shards, cause the person to unwisely pursue the plan. The point here is that (I claim) intelligent people convergently want to avoid this happening to them.
> >
> > A diamond-shard will not try to find adversarial inputs to itself. That was my original point, and I think it stands.

Furthermore, I think that, in systems with multiple optimizers (eg shards), some optimizers can feed the *other optimizers* adversarial inputs. (Adversarial inputs are most common in the presence of an adversary, after all!)

A very rough guess at what this looks like: A luxury-good-shard proposes a golden-laptop buying plan, while emphasizing how this purchase stimulates the economy and so helps people. This plan was optimized to positively activate e.g. the altruism-shard, so as to increase the plan's execution probability. In humans, I think this is more commonly known as *motivated reasoning*.

So, even in value-child, adversarial inputs can still crop up, but via a different mechanism which should disappear once the agent gets smart enough to e.g. do an internal values handshake. As I said to Wei Dai:

> I agree that humans sometimes fall prey to adversarial inputs...
>
> However, this does not seem important for my (intended) original point. Namely, if you're trying to align e.g. a brute-force-search plan maximizer or a grader-optimizer, you will fail due to high-strength optimizer's curse forcing you to evaluate extremely scary adversarial inputs. But also this is sideways of real-world alignment, where realistic motivations may not be best specified in the form of "utility function over observation/universe histories."

Appendix B: Prior work

Abram Demski writes about Everitt et al.'s *Self-Modification of Policy and Utility Function in Rational Agents*:

> As a first example, consider the wireheading problem for AIXI-like agents in the case of a fixed utility function which we know how to estimate from sense data. As discussed in Daniel Dewey's Learning What to Value and other places, if you try to implement this by putting the utility calculation in a box which rewards an AIXI-like RL agent, the agent can eventually learn to modify or remove the box, and happily does so if it can get more reward by doing so. This is because the RL agent predicts, and attempts to maximize, reward received. If it understands that it can modify the reward-giving box to get more reward, it will.
>
> We can fix this problem by integrating the same reward box with the agent in a better way. Rather than having the RL agent learn what the output of the box will be and plan to maximize the output of the box, we use the box *directly* to evaluate possible futures, and have the agent plan to maximize that evaluation. Now, if the agent considers modifying the box, it evaluates that future *with the current box*. The box as currently configured sees no advantage to such tampering. This is called an observation-utility maximizer (to contrast it with reinforcement learning). Daniel Dewey goes on to show that we can incorporate uncertainty about the utility function into observation-utility maximizers, recovering the kind of "learning what is being rewarded" that RL agents were supposed to provide[...]
>
> Stable Pointers to Value: An Agent Embedded in Its Own Utility Function

The point of this post isn't *just* that e.g. value-child evaluates the future with his own values, as opposed to putting the utility calculation in a box. I'm not describing a failure of tampering with the grader. I'm describing a failure of *optimizing the output of a box/grader*, even if the box is *directly evaluating possible futures.* After all, evaluation-child uses the box to directly evaluate possible futures! Evaluation-child wants to maximize the evaluation of his model of his mother!

As described above, value-child is steered by his values. He isn't optimizing for the output of some module in his brain.

Appendix C: Grader-optimization quiz

Grader optimization is about how the agent *thinks,* it's about the way in which they are motivated.

Scenario 1

> Bill looks around the workshop. The windows are shattered. The diamonds—where are they..?!
>
> Should he allocate more time to meta-planning—what thoughts should he think next? No. Time is very limited, and spending more time thinking now would lead

> to fewer expected-diamonds. He decides to simply wield the cognitive habits which his past mental training drilled to activate in this kind of mental context.
>
> Police? Promising, but spend a few more seconds generating ideas to avoid automatic opportunity cost from prematurely committing to the first idea. [After all, doing otherwise historically led to fewer diamonds, which produced less cognition-update-quantity (i.e. "reward") than expected, and so his credit assignment chipped away at the impulse to premature action in this kind of situation.]
>
> Generate alternate explanations for where the diamonds went? No, Bill's self-model expects this to slightly decrease probability of inferring in time where the diamonds went, and so Bill feels like avoiding that next thought.
>
> ...

Question: Is Bill a grader-optimizer?

No! Bill's cognition is shaped towards *acquiring diamonds*, his cognition reliably pulls him into futures where he has more diamonds. **This is not grader-optimization.** This is Bill caring about diamonds, not about his own evaluations of whether a plan will acquire diamonds.

Scenario 2

> Bill flops down on his bed. Finally, he has private time to himself. All he wants, all he's ever wanted, is to think that he's finally *made it*—that he can finally believe himself to have acquired real diamonds. He doesn't care how he does it. He just wants to believe, and that's *it.*
>
> Bill has always been different, somehow. When he was a kid, Bill would imagine plans like "I go to school and also *have tons of diamonds*", and that would initially trick him into thinking that he'd found a plan which led to tons of diamonds.
>
> But as he got older and smarter, he thought maybe he could do better. He started learning about psychology and neuroscience. He started guessing how his brain worked, how to better delude himself (the ultimate human endeavor).
>
> ...

Question: Is Bill a grader-optimizer?

Yes! Bill's optimizing for either his future physical evaluation of plan quality, or some Platonic formalization of "Did I come up with a plan I think is promising?". Which? The story is ambiguous. But the mark of grader-optimization is quite plain, as given by a plan-generator stretching its wits to maximize the output of a grader.

1. ^

   The actor may give an *instrumental* damn about diamonds, because diamond-producing plans sometimes produce high evaluations. But in actor/grader motivational setups, an inner-aligned actor only gives a *terminal* damn about the *evaluations*.

2. ^

   Although AIXI's epistemic prior is malign and possibly unsafe...

3. ^

   But, you don't have to have another approach in mind in order to abandon grader-optimization. Here are some things I would ask myself, were I confused about how non-grader-optimizing agents might be motivated:

   - "Hey, I realize some strangeness about this thing (grader-optimization) which I was trying to do. I wonder whether there are other perspectives or frame-shifts which would make this problem go away?"

   - "I notice that I don't expect a paperclip-AI to resort to grader-optimization in order to implement its own unaligned values. What do I anticipate would happen, internally, to an AI as I trained it via some RL curriculum? If it cared about paperclips, how would that caring be implemented, mechanistically?"

   - "Hm, this way of caring about things seems weird. In what ways is grader-optimization similar and dissimilar to the suspected ways in which human beings care about things?"

4. ^

   Contrast with a quote from the original article:

   > Similarly, if [the actor] is smarter than [the grader] expects, the only problem is that [the actor] won’t be able to use all of his intelligence to devise excellent plans. This is a serious problem, but it can be fixed by trial and error—rather than leading to surprising failure modes.

5. ^

   Not that I think this has a snowflake's chance in hell of working in time. But it seemed important to show that not all indirect normativity is grader-optimization.

6. ^

   Earlier this year, I analyzed how brute-force plan search might exploit this scheme for using an ELK direct translator.

Alignment allows "nonrobust" decision-influences and doesn't require robust grading

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

Definition. On how I use words, values are decision-influences (also known as *shards*). “I value doing well at school” is a short sentence for “in a range of contexts, there exists an influence on my decision-making which upweights actions and plans that lead to e.g. learning and good grades and honor among my classmates.”

Summaries of key points:

1. **Nonrobust decision-influences can be OK.** A candy-shard contextually influences decision-making. Many policies lead to acquiring lots of candy; the decision-influences don't have to be "globally robust" or "perfect."
2. **Values steer optimization; they are not optimized against.** The value shards aren't getting optimiz*ed* hard. The value shards **are** the things which optimize hard, by wielding the rest of the agent's cognition (e.g. the world model, the general-purpose planning API).

   Since values are not the optimization target of the agent with those values, the values don't have to be adversarially robust.
3. **Since values steer cognition, reflective agents try to avoid adversarial inputs to their own values.** In self-reflective agents which can think about their own thinking, values steer e.g. what plans get considered next. Therefore, these agents convergently avoid adversarial inputs to their currently activated values (e.g. learning), because adversarial inputs would impede fulfillment of those values (e.g. lead to less learning).

Follow-up to: Don't design agents which exploit adversarial inputs, Don't align agents to evaluations of plans

I: Nonrobust decision-influences can be OK

Decision-making influences don't have to be “robust” in order for a person to *value doing well at school*. Consider two people with slightly different values:

1. One person is slightly more motivated by good grades. They might study for a physics test and focus slightly more on test-taking tricks.
2. Another person is slightly more motivated by learning. They might forget about some quizzes because they were too busy reading extracurricular physics books.

But they might *both* care about school, in the sense of reliably making decisions on the basis of their school performance, and valuing being a person who gets good grades. Both people are motivated to do well at school, albeit in somewhat different ways. They probably will both get good grades, and they probably will both learn a lot. **Different values simply mean that the two people locally make decisions differently.**

If I value candy, that means that my decision-making contains a subroutine which makes me pursue candy in certain situations. Perhaps I eat candy, perhaps I collect candy, perhaps I let children tour my grandiose candy factory… The point is that candy influences my decisions. I am *pulled by my choices* from pasts without candy to futures with candy.

So, let C be the set of mental contexts relevant for decision-making, and let A be my action set.

[1] My policy has type signature $\pi : C \to A$, and e.g. contains a bunch of shards of value which influence its outputs. The values are subcircuits of my policy network (i.e. my brain). For example, consider a candy shard consisting of the following subshards:

1. If `center-of-visual-field` activates `candy`'s visual abstraction, then `grab` the `inferred latent object which activated the abstraction`.
2. If `hunger>50` and `sugar-level<6`, and if `current-plan-stub` activates `candy-obtainable`, then tell `planning API` to set `subgoal` to `obtain candy`.
3. If `heard 'candy'` and `hunger>20`, then `salivate`.
4. …

Suppose this is the way I value candy. A few thousand subshards which chain into the rest of my cognition and concepts. A few thousand subshards of value which were hammered into place by tens of thousands of reinforcement events across a lifetime of experience.

This shard does not need to be "robust" or "perfect." Am I really missing much if I'm lacking candy subshard #3: "If `heard 'candy'` and `hunger>20`, then `salivate`"? I don't think it makes sense to call value shards "perfect" or not.[2] The shards simply influence decisions.

There are many, many configurations and parameter settings of these subshards which lead to *valuing candy*. The person probably still values candy, even if you:

- Delete a bunch of the subshards.
- Modify the activation-strength of a bunch of subshards (roughly, change how much control the subshard has on the next-thought "logits").
- Change some of the activation contexts to other common activation contexts (e.g. "If `heard 'candy'`" changes to "If `heard 'sweets'`", change to `hunger>14` in subshard 3).

It seems to me like "does the person still prioritize candy" depends on a bunch of factors, including:

1. Retention of core abstractions (to some tolerance)
    1. If we find-replaced `candy` with `flower`, the person probably now has a strange flower-value, where they eat flowers when hungry.
    2. However, the abstraction also doesn't have to be "perfect" (whatever that means) in order to activate in everyday situations. Two people will have different candy abstractions, and yet they can both value candy.
2. Strength and breadth of activation contexts
    1. The more situations a candy-value affects decision-making in, the stronger the chance that candy remains a big part of their life.
3. How often the candy shard will actually activate
    1. As an unrealistic example, if the person never enters a cognitive situation which substantially activates the candy-shard, then don't expect them to eat much candy.
    2. This is another source of value/decision-influence robustness, as e.g. an AI's values don't have to be OK in every cognitive context.[3]
    3. Consider an otherwise altruistic man who has serious abuse and anger problems whenever he enters a specific vacation home with his wife, but is otherwise kind and considerate. As long as he doesn't start off in that home but knows about the contextual decision-influence, he will steer away from that home and try to remove the unendorsed value.
4. Reflectivity of the candy shard
    1. (This is more complicated and uncertain. I'll leave it for now.)

Suppose we wanted to train an agent which gets really smart and acquires a lot of candy, now and far into the future. **That agent's decision-influences don't have to be globally robust (e.g. in every cognitive situation, the agent is motivated by candy and only by candy) in order for an agent to make locally good decisions (e.g. make lots of candy now and into the future).**

II: Values steer optimization; they are not optimized against

Given someone’s values, you might wonder if you can “maximize” those values. On my ontology—where *values* are *decision-influences*, a sort of *contextual wanting*—“literal value maximization” is a type error.[4] In particular, given e.g. someone who values candy, there very probably isn't a part of that person's cognition which can be argmaxed to find a plan where the person has lots of candy.

So if I have a candy-shard, if I value candy, if *I am influenced to decide to pursue candy in certain situations*, then *what does it mean to maximize my candy value*? My value is a subcircuit of my policy. It doesn’t necessarily even have an ordering over its outputs, let alone a numerical rating which can be maximized. “Maximize my candy-value” is, in a literal sense, a type error. What quantity is there to maximize?

> the True Name of a thing [is] a mathematical formulation sufficiently robust that one can apply lots of optimization pressure without the formulation breaking down[...]
>
> If we had the “True Name” of human values (insofar as such a thing exists), that would potentially solve the problem [of supervised labels only being proxies for what we want].
>
> Why Agent Foundations? An Overly Abstract Explanation

In particular, there’s no guarantee that you can just scan someone’s brain and find some True Name of Value which you can then optimize without fear of Goodhart. It’s not like we don’t know what people value, but if we did, we would be OK. I'm pretty confident there does not exist *anything* within my brain which computes a True Name for my values, ready to be optimized as hard as possible (relative to my internal plan ontology) and yet still producing a future where I get candy.

Therefore, even though you *truly care about candy*, that doesn’t mean you can just whip out the argmax on the relevant shard of your cognition, so as to “maximize” that shard (e.g. via extremizing the rate of action potentials on its output neurons) and then get a future with *lots* of candy. You’d probably just find a context $c_i \in C$ which acts as an adversarial input to the candy-shard, *even though* you do really care about candy in a normal, human way.

Complexity of human values isn’t what stops you from argmaxing human values and thereby finding a good plan. That’s not a sensible thing to try. Values are not, in general, the kind of thing which can be directly optimized over, where you find plans which "maximally activate" your e.g. candy-subshards. **Values influence decisions.**

If you're confused at the distinction between "optimizing against values" and "values influencing decisions", read Don't align agents to evaluations of plans .

There is real difficulty and peril in motivating an AI, in making sure its decisions chain into each other towards the *right kinds of futures*. If you train a superintelligent sovereign agent which primarily values irrelevant quantities (like paperclips) but doesn't care about you, which then optimizes the whole future hard, then you’re *dead*. But consider that deleting candy subshard #3 (“If `heard 'candy'` and `hunger>20`, then `salivate`”) doesn’t stop someone from valuing candy in the normal way. If you erase that subshard from their brain, it’s not like they start "Goodharting" and forget about the “true nature” of caring about candy because they now have an “imperfect proxy shard.”

An agent argmax'ing an imperfect evaluation function will indeed exploit that function; there are very few degrees of freedom in specifying an inexploitable evaluation function. But that's because that grading function must be globally robust.

When I talk about shard theory, people often seem to shrug and go "well, you still need to get the values adversarially-robustly-correct else Goodhart; I don't see how this 'value shard' thing helps." **That's not how values work, that is not what value-shards are. Unlike grader-optimizers which try to maximize plan evaluations, a values-executing agent doesn't optimize its values as hard as possible. The agent's values optimize the world. The values are rules for how the agent acts in relevant contexts.**[5]

III: Since values steer cognition, reflective agents try to avoid adversarial inputs to their own values

Question: If we cannot robustly grade expected-diamond-production for every plan the agent might consider, how might we nonetheless design a smart agent which makes lots of diamonds?

(Maybe you can now answer this question. I encourage you to try before moving on.)

In Don't design agents which exploit adversarial inputs, I wrote:

> Imagine a mother whose child has been goofing off at school and getting in trouble. The mom just wants her kid to take education seriously and have a good life. Suppose she had two (unrealistic but illustrative) choices.
>
> 1. *Evaluation-child:* The mother makes her kid care extremely strongly about doing things which the mom would evaluate as "working hard" and "behaving well."
> 2. *Value-child:* The mother makes her kid care about working hard and behaving well.

To make evaluation-child work hard, we have to somehow specify a grader which can adequately grade all plans which evaluation-child can imagine. The highest-rated imaginable plan must involve working hard. This requirement is extreme.

Value-child doesn't suffer this crippling "robustly grade exponentially many plans" alignment requirement. I later wrote a detailed speculative account of how value-child's cognition might work—what it *means* to say that he "cares about working hard." But, at a higher level, what are the main differences between evaluation- and value-child?

This may sound obvious, but I think that the main difference is that **value-child actually cares about working hard.** Evaluation-child cares about evaluations. (See here if confused on the distinction.) To make evaluation-child work hard in the limit of intelligence, you have to *robustly ensure that max evaluations only come from working hard*. This sure sounds like a slippery and ridiculous kind of thing to try, like wrestling a frictionless pig. It should be no surprise you'll hit issues like nearest unblocked strategy in that paradigm.

An agent which *does* care about working hard will want to not think thoughts which lead to not working hard. In particular, reflective shard-agents can think about what to think, and thereby are convergently-across-values incentivized to steer clear of adversarial inputs to their own values.

Reflectively avoiding adversarial inputs to your thinking

Reflective agents can think about their own thought process (e.g. "should I spend another five minutes thinking about what to write for this section?"). I think they do this via their world-model predicting internal observables (e.g. future neuron activations) and thus high-level statistics like "If I think for 5 more minutes, will that lead to a better post or not?".

Thoughts about future thinking are a kind of decision. Decisions are steered by values. Therefore, thoughts about future thinking are steered by whatever value shards activate in that mental context. For example, a self-care value might activate, and a learning-shard, and a social value might activate as well. They control your reflective thoughts, just like other shards would control your ("normal") actions (like crossing the room).

In Don't design agents which exploit adversarial inputs, I wrote:

> [In] the optimizer's curse, evaluations (eg "In this plan, how hard is evaluation-child working? Is he behaving?") are often corrupted by the influence of unendorsed factors (eg the attractiveness of the gym teacher caused an upwards error in the mother's evaluation of that plan). If you make choices by considering n options and then choosing the highest-evaluated one, then the more n increases, the harder you are selecting for upwards errors in your own evaluation procedure.
>
>> The proposers of the Optimizer's Curse also described a Bayesian remedy in which we have a prior on the expected utilities and variances and we are more skeptical of very high estimates. This however assumes that the prior itself is perfect, as are our estimates of variance. If the prior or variance-estimates contain large flaws somewhere, a search over a very wide space of possibilities would be expected to seek out and blow up any flaws in the prior or the estimates of variance.
>>
>> Goodhart's Curse, Arbital
>
> As far as I know, it's indeed not possible to avoid the curse in full generality, but it doesn't have to be that bad in practice. If I'm considering three research directions to work on next month, and I happen to be grumpy when considering direction #2, then maybe I don't pursue that direction. Even though direction #2 might have seemed the most promising under more careful reflection. I think that the distribution of plans I consider involves relatively small upwards errors in my internal evaluation metrics. Sure, maybe I occasionally make a serious mistake due to the optimizer's curse due to upwards "corruption", but I don't expect to *literally die* from the mistake.
>
> Thus, there are are *degrees* to the optimizer's curse.

Both grader-optimization and argmax cause extreme, horrible optimizer's curse. Is alignment just that hard? I think not.

The distribution of plans which I usually consider is not going to involve any set of mental events like "consider in detail building a highly persuasive superintelligence which persuades you to build it." While a reflective diamond-valuing AI which *did* execute the plan's mental steps might get hacked by that adversarial input, there would be no reason for it to seek out that plan to begin with.

$$\overset{\text{Thinking about adv-plan in detail seems bad ...}}{eval(\text{evaluate the adversarial plan}) = -1} \qquad \overset{\text{Even though actual evaluation of the plan would be great}}{eval(\text{adversarial plan}) = INT\_MAX}.$$

The diamond-valuing AI would consider a distribution of plans far removed from the extreme upwards errors highlighted in the evaluation-child story. (I think that this is why *you,* in your day-to-day thinking, don't have to worry about plans which are extreme adversarial inputs to your own evaluation procedures.) Even though a smart reflective AI may be implicitly searching over a range of plans, it's doing so *reflectively*, thinking about what to think next, and perhaps not taking cognitive steps which it reflectively predicts to lead to bad outcomes (e.g. via the optimizer's curse).

On the other hand, an AI which is aligned on the evaluation procedure is *incentivized to seek out huge upwards errors on the evaluation procedure relative to the intended goal*. The actor is *trying* to generate plans which maximally exploit the grader's reasoning and judgment.[6]

Thus, if an AI cares about diamonds (i.e. has an influential diamond-shard), that AI might accidentally select a plan due to upwards evaluative noise, but that does not mean the AI is *actively looking for plans to fool its diamond-shard into oblivion.* The AI may make a mistake in its reflective predictions, but there's no extreme optimization pressure for it to make mistakes like that, and the AI wants to avoid those mistakes, and so those mistakes remain unlikely. I think that reflective, smart AIs convergently want to avoid duping their own evaluative procedures, for the same reasons you want to avoid doing that to yourself.

More precisely:

1. A reflective diamond-motivated agent chooses plans based on how many diamonds they lead to.
2. The agent can predict e.g. how diamond-promising it is to search for plans involving simulating malign superintelligences which trick the agent into thinking the simulation plan makes lots of diamonds, versus plans where the agent just improves its synthesis methods.
3. A reflective agent thinks that the first plan doesn't lead to many diamonds, while the second plan leads to more diamonds.
4. Therefore, the reflective agent chooses the second plan over the first plan, automatically[7] avoiding the worst parts of the optimizer's curse. (Unlike grader-optimization, which seeks out adversarial inputs to the diamond-motivated part of the system.)

Therefore, avoiding the high-strength curse seems conceptually straightforward. In the case of aligning an AI to produce lots of diamonds, we want the AI to superintelligently generate and execute diamond-producing plans *because* the AI expects those plans to lead to lots of diamonds. I have spelled out a plausible-to-me story for how to accomplish this. The story is simple in its essential elements: finetune a pretrained model by rewarding it when it collects diamonds.

While that story has real open questions, that story also totally sidesteps the problems with grader-optimization. You don't have to worry about providing some globally unhackable evaluation procedure to make super duper sure the agent's plans "really" involve diamonds. If the early part of training goes as described, the agent *wants* to make diamonds, and (as I explained in the diamond-alignment story) it reflectively wants to avoid duping itself because duping itself leads to fewer diamonds.

This answers the above question:

> If we cannot robustly grade expected-diamond-production for every plan the agent might consider, how might we nonetheless design a smart agent which makes lots of diamonds?

A reflective agent wishes to *minimize* the optimizer's curse (relative to its own values), instead of *maximizing* it (relative to the goal by which the grader evaluates plans). While I don't yet have satisfying pseudocode for reflective planning agents (but see Appendix B for preliminary pseudocode, effective reflective agents *do* exist. In this regime, it seems like many scary problems go away and don't come back. That is an enormous blessing.[8]

Argmax is an importantly inappropriate idealization of agency

If the answer to "how do we dispel the max-strength optimizer's curse" is in fact "real-world reflective agents do this naturally", then assuming unreflectivity will *rule out the part of solution-space containing the actual solution*:

> As a further-simplified but still unsolved problem, an **unreflective diamond maximizer** is a diamond maximizer implemented on a Cartesian hypercomputer in a causal universe that does not face any Newcomblike problems. This further avoids problems of reflectivity and

> logical uncertainty. In this case, it seems plausible that the primary difficulty remaining is *just* the ontology identification problem.
>
> *Diamond Maximizer,* Arbital (emphasis added)

The argmax and unreflectivity assumptions were meant to make the diamond-maximizer problem *easier*. **Ironically, however, these assumptions may well render the diamond-maximizer problem *unsolvable***, leading us to resort to increasingly complicated techniques and proposals, none of which seem to solve "core" problems like evaluation-rule hacking...

Conclusion

1. **Nonrobust decision-influences can be OK.**
2. **Values steer optimization; they are not optimized against.**
3. **Since values steer cognition, reflective agents try to avoid adversarial inputs to their own values.**

The answer is not to find a clever way to get a robust grader. The answer is to not *need* a robust grader. Form e.g. a diamond-production value within a reflective and smart agent, and this diamond-production value won't be incentivized to fool itself. You won't have to "robustly grade" it to make it produce diamonds.

Thanks to Tamera Lanham, John Wentworth, Justis Mills, Erik Jenner, Johannes Treutlein, Quintin Pope, Charles Foster, Andrew Critch, randomwalks, Ulisse Mini, and Garrett Baker for thoughts. Thanks to Vivek Hebbar for in-person discussion.

Appendix A: Several roads lead to a high-strength optimizer's curse

1. **Uncertainty about how human values work.** Suppose we think that human values are so complex, and there's no real way to understand them or how they get generated. We imagine a smart AI as finding futures which optimize some grading rule, and so we need something to grade those futures. We think we can't get the AI to grade the futures, because human values are so complex. What options remain available? Well, the only sources of "good judgment" are existing humans, so we need to find some way to use those humans to target the AI's powerful cognition. We give the alignment, the AI gives the cognitive horsepower.

   We've fallen into the grader-optimization trap.
2. **Non-embedded forms of agency.** This encourages considering a utility function maximized over all possible futures. Which automatically brings the optimizer's curse down to bear at maximum strength. You can't specify a utility function which is robust against *that*.

   This seems sideways of real-world alignment, where realistic motivations may not be best specified in the form of "utility function over observation/universe histories."

Appendix B: Preliminary reflective planning pseudocode

I briefly took a stab at writing pseudocode for a values-based agent like value-child. I think this code leaves a lot out, but I figured it'd be better to put *something* here for now.

```
            '''
Here is one meta-plan for planning. The agent starts from no plan at all, iteratively
generates improvements, which get accepted if they lead to more predicted diamonds. Depending
on the current situation (as represented in the WM), the agent might execute a plan in which
it looks for nearby diamonds, or it might execute a plan where it runs a different kind of
heuristic search on a certain class of plans (e.g. research improvements to the AI's diamond
synthesis pathway).

Any real shard agent would be reasoning and updating asynchronously, so this setup assumes a
bit of unrealism.

This function only modifies the internal state of the agent (self.recurrent), so as to be
ready for a call of self.getDecisions().
'''
def plan(self):
# Generate an initial plan
plan = Plan() # Do nothing plan
conseq = self.WM.getConseq(plan)
currentPlanEval = self.diamondShard(conseq)

# Iteratively modify the plan until the generative model can't find a way to make it better
while True:
 # Sample 5 plan modifications from generative model
 plans = self.WM.planModificationSample(n=5,stub=plan)

 # Select first local improvement
 for planMod in plans:
  # Reflectively predict consequences of this plan
  newPlan = plan.modify(planMod)
  conseq = self.WM.getConseq(plan)

  # Take local improvement
  if self.diamondShard(conseq) > currentPlanEval:
   plan = newPlan
   currentPlanEval = self.diamondShard(conseq)
   continue # Generate more modifications
 # Execute the plan, which possibly involves running plan search with a different algorithm
and plan initialization.
 isDone = plan.exec()
 if isDone: break
```

Appendix C: Value shards all the way down

I liked Vivek Hebbar's recent comment (in the context of e.g. caring about your family and locally evaluating plans on that basis, but also knowing that your evaluation ability itself is compromised and will mis-rate some plans):

> My attempt at a framework where "improving one's own evaluator" and "believing in adversarial examples to one's own evaluator" make sense:
>
> - The agent's allegiance is to some idealized utility function U_{ideal} (like CEV). The agent's internal evaluator Eval is "trying" to approximate U_{ideal} by reasoning heuristically. So now we ask Eval to evaluate the plan "do argmax w.r.t. Eval over a bunch of plans". Eval reasons that, due to the the way that Eval works, there should exist "adversarial examples" that score very highly on Eval but low on U_{ideal}. Hence, Eval concludes that U_{ideal}(plan) is low, where plan = "do argmax w.r.t. Eval". So the agent doesn't execute the plan "search widely and argmax".

> - "Improving Eval" makes sense because Eval will gladly replace itself with Eval_2 if it believes that Eval_2 is a better approximation for U_{ideal} (and hence replacing itself will cause the outcome to score better on U_{ideal})
>
> Are there other distinct frameworks which make sense here?

(I'm not sure whether Vivek meant to imply "and this is how I think people work, mechanistically." I'm going to respond to a *hypothetical other person* who did in fact mean that.)

My take is that human value shards explain away the need to posit alignment to an idealized utility function. A person is not a bunch of crude-sounding subshards (e.g. "If `food nearby` and `hunger>15`, then be more likely to `go to food`") and then *also* a sophisticated utility function (e.g. something like CEV). It's shards all the way down, and all the way up.[10]

Vivek then wrote:

> I look forward to seeing what design Alex proposes for "value child".

Value shards steer cognition. In the main essay, I wrote:

1. A reflective diamond-motivated agent chooses plans based on how many diamonds they lead to.
2. The agent can predict e.g. how diamond-promising it is to search for plans involving simulating malign superintelligences which trick the agent into thinking the simulation plan makes lots of diamonds, versus plans where the agent just improves its synthesis methods.
3. A reflective agent knows that the first plan doesn't lead to many diamonds, while the second plan leads to more diamonds.
4. Therefore, the reflective agent chooses the second plan over the first plan, automatically avoiding the worst parts of the optimizer's curse. (Unlike grader-optimization, which seeks out adversarial inputs to the diamond-motivated part of the system.)

This story smoothly accomodates thoughts about improving evaluation ability.

On my understanding: Your values are steering the optimization. They are not, in general, being optimized against by some search inside of you. They are probably not pointing to some idealized utility function. The decision-influences are *guiding* the search. There's no secret other source of caring, no externalized utility function.

1. ^

   Formalizing the action space A is a serious gloss. In people, there is no privileged “action” space, considering how I can decide what to think about next. As an embedded agent, I don’t just decide what motor commands to send and what words to say—I also can decide what to decide next, what to think about next.

   I think the point of the essay stands anyways.

2. ^

   This doesn’t mean that I’m using words the same way other people have, when deliberating on whether an AI’s values have to be “robust.” I’m more inclined to just carry out the shard theory analysis and see what experiences it leads me to anticipate, instead of arguing about whether my way of using words matches up with how other people have used words.

3. ^

As Charles Foster notes:

> This isn't entirely on the side of robustness. It also means that by default, even if we get the AI to have an X decision influence in one context, that doesn't necessarily also activate in another context we might want it to generalize to.

4. ^

I think that many people say "maximize my values" to mean something like "do something as great as possible, relative to what I care about." So, in a sense, "type error" is pedantic. But also I think the type error complaint points at something important, so I'll say it anyways.

5. ^

If you want to argue that *decision-influences* have to be robust else Goodhart, you need new arguments not related to grader-optimization. It is simply invalid to say "The agent doesn't value diamonds in some situation where lots of its values activate strongly, and therefore the agent won't make diamonds because it Goodharts on that unrelated situation." That is not what values do.

6. ^

In a recent Google Doc thread, grader optimization came up. Someone said to me (my reactions in italics):

> So you're imagining something like: the agent (policy) is optimizing for a reward model to produce a high number, and so the agent analyzes the reward model in detail to search for inputs that cause the reward model to give high numbers? *Yes.*
>
> I think at that level of generality I don't know enough to say whether this is good or bad. *I think this is very, very probably bad.*
>
> We want our AI system to search for approaches that better enact our values. As you note, the optimizer's curse says that we'll tend to get approaches that overestimate how much they actually enact our values. But just knowing that the optimizer's curse will happen doesn't change anything; the best course of action is still to take the approach that is predicted to best enact our values. In that sense, the optimizer's curse is typically something you have to live with, not something you can solve.[...]
>
> *A small error is not the same as a maximal error. Reflective agents can and will avoid deliberately searching for plans which maximize upwards errors in their own evaluations (e.g. generating a plan such that, while considering the plan, a superintelligence inside the plan tricks you into thinking the plan should be highly evaluated), because the agents reflectively predict that that hurts their goal achievement (e.g. leads to fewer diamonds). If you somewhat understand your decision-making, you can consider plans you're less likely to incorrectly evaluate.*
>
> So my followup question is: can you name a single approach that doesn't have this failure mode, while still allowing us to use the AI to do things we didn't think about in advance? *Yes*.
>
> One answer someone might give is "create an agent-with-shards that searches for approaches that score highly on the shard.
>
> *Insofar as this means "the agent looks for inputs which maximize the aggregate shard output", no. On my model, shards grade and modify plans, including plans about which plans to consider next. They are not searching for plans which maximize evaluative output, like in the reward-model case.*

> An agent that searches for high scores on its shard can't be searching for positive upwards errors in the shard; there is no such thing as an error in the shard". To which the response is "that's from the agent's perspective. From the human's perspective, the agent is searching for positive upwards differences between the shards and what-the-human-wants".
>
> *Even if true, this would be not be an optimizer's curse problem.*
>
> *But also this isn't true, at least not without further argumentation. If my kid likes mocha and I like latte, is my child searching for positive upwards differences between their values and mine? I think there are some situations—AI paperclips, humans values love—where the AI is searching for paperclippish plans, which will systematically be bad plans by human lights. That seems more like instrumental convergence -> disempower humans -> not much love left for us if we're dead.*

7. ^

   I do think that e.g. a diamond-shard can get fed an adversarial input, but the diamond-shard won't bid for a plan where it fools *itself*.

8. ^

   It's at this point that my model of Nate Soares wants to chime in.

   Alex's model of Nate (A-N): This sure smells like a problem redefinition, where you simply sweep the hard part of the problem under a less obvious corner of the rug. Why shouldn't I believe you've just done that?

   A: A reasonable and productive heuristic in general, but inappropriate here. Grader-optimization explicitly incentivizes the agent to find maximal upwards errors in a diamond-evaluation module, whereas a reflective diamond-valuing agent has no incentive to consider such plans, because it reflectively predicts those plans don't lead to diamonds. If you disagree, please point to the part of the story where, conditional on the previous part of the story obtaining, the grader-optimization problem reappears.

   A-N: Suppose we achieved your dream of forming a diamond-shard in an AI, and that that shard holds significant power over the AI's decisions. Now the AI keeps improving itself. Doesn't "get smarter" look a lot like "implicitly consider more options", which brings the curse back?

   A: If the agent is diamond-aligned at this point in time, I expect it stays that way for the reasons given in the "agent prevents value drift" section, along with these footnotes and the appendix. As a specific answer, though: If the agent does care about diamonds at that point it time, then it doesn't *want* to get so "smart" that it deludes itself by seriously intensifying the optimizer's curse. It doesn't want to do so for the reason *we* don't want it to do so (in the hypothetical where we just want to achieve diamond-alignment). If the reflective agent can predict that outcome of the plan, it won't execute the plan, *because that plan leads to fewer diamonds*.

   A-N: So the AI still has to solve the AI alignment problem, except with its successors.

   A: Not all things which can be called an "AI alignment problem" are created equal. The AI has a range of advantages, and I detailed one way it could use those advantages. I do expect that kind of plan to actually work.

9. ^

   I further speculate that reflective reasoning is convergently developed in real-world training processes under non-IID conditions like those described in my diamond-alignment story.

10. ^

When working out shard theory with Quintin Pope, one of my favorite moments was the *click* where I stopped viewing myself as some black-box optimizing "some complicated objective." Instead, this hypothesis reduced my own values to mere reality. Every aspiration, every unit of caring, every desire for how I want the future to be bright and fun—subroutines, subshards, contextual bits of decision-making influence, all traceable to historical reinforcement and update events.

Inner and outer alignment decompose one hard problem into two extremely hard problems

Crossposted from the AI Alignment Forum. May contain more technical jargon than usual.

TL;DR: One alignment strategy is to 1) capture "what we want" in a loss function to a very high degree ("robust grading"), 2) use that loss function to train the AI, and 3) get the AI to exclusively care about optimizing that objective.

I think that each step contains either a serious and unnecessary difficulty, or an unnecessary assumption. I think that:

1. **Robust grading is unnecessary, extremely hard, and unnatural.** But we don't *have* to find/represent/produce an objective which is safe for a smart agent to directly optimize. *Robust grading seems harder than the entire actual AI alignment problem.*
2. **The loss function doesn't have to robustly and directly reflect what you want.** Loss functions chisel circuits into networks. Even if we *did* want to do robust grading, we don't have to *also* use that grading rule to optimize directly over the network's cognition. This assumption is restrictive.
3. **Inner alignment to a grading procedure is unnecessary, very hard, and anti-natural.** We don't have to precisely and exclusively align the agent to its loss function or to an external grading procedure. *This precise and complete inner alignment might be very hard, possibly harder than the entire actual alignment problem.*

Extended summary. My views on alignment have changed a lot recently. To illustrate some key points, I'm going to briefly discuss a portion of Paul Christiano's AXRP interview (emphasis added):

> **Paul Christiano:** [...] In general, I don't think you can look at a system and be like, "oh yeah, that part's outer alignment and that part's inner alignment". So the times when you can talk about it most, or the way I use that language most often, is for a particular kind of alignment strategy that's like a two step plan. **Step one is, develop an objective that captures what humans want well enough to be getting on with. It's going to be something more specific, but you have an objective that captures what humans want in some sense. Ideally it would exactly[1] capture what humans want. So, you look at the behavior of a system and you're just exactly evaluating how good for humans is it to deploy a system with that behavior, or something.** So you have that as step one and then that step would be outer alignment. And then **step two is, given that we have an objective that captures what humans want, let's build a system that's internalized that objective in some sense, or is not doing any other optimization beyond pursuit of that objective.**

> **Daniel Filan:** And so in particular, the objective is an objective that you might want the system to adopt, rather than an objective over systems?
>
> **Paul Christiano:** Yeah. I mean, we're sort of equivocating in this way that reveals problematicness[2] or something, but **the first objective is an objective. It is a ranking over systems, or some reward that tells us how good a behavior is. And then we're hoping that the system then adopts that same thing, or some reflection of that thing**[...]

My summary: One alignment strategy is to 1) capture "what we want" in a loss function to a very high degree ("robust grading"), 2) use that loss function to train the AI, and 3) get the AI to exclusively care about optimizing that objective.[3]

I think that each step contains either a serious and unnecessary difficulty, or an unnecessary assumption. I think that:

1. **Robust grading is unnecessary**, **extremely hard, and unnatural.** But we don't *have* to find/represent/produce an objective which is safe for a smart agent to directly optimize. *Robust grading seems harder than the entire actual AI alignment problem.*
2. **The loss function doesn't have to robustly and directly reflect what you want.** Loss functions chisel circuits into networks. Even if we *did* want to do robust grading, we don't have to *also* use that grading rule to optimize directly over the network's cognition. This assumption is quite restrictive.
3. **Inner alignment to a grading procedure is unnecessary, very hard, and anti-natural.** We don't have to precisely and exclusively align the agent to its loss function or to an external grading procedure. *This precise and complete inner alignment might be very hard, possibly harder than the entire actual alignment problem.*

Therefore, for all alignment approaches which aim to align an agent to a robust grading scheme, I think that that approach is doomed. However, I am **not** equally critiquing all alignment-decompositions which have historically been called "outer/inner alignment" (for more detail, see Appendix A).

Here's the structure of the essay, and some key points made within:

1. Robust grading is unnecessary, extremely hard, and unnatural.
   1. An agent which exclusively cares about the output of some objective (e.g. "How many diamonds an extremely smart person thinks *input-plan* will produce") *doesn't care about diamonds*. That agent ultimately only cares about *high objective outputs*.
   2. Robust grading incentivizes an inner-aligned AI to search for upwards errors in your grading procedure, but I think it's easy to tell plausible training stories which don't require robust outer objectives.
   3. We've tried finding robust grading methods and have failed for a range of objectives, from diamond production to protecting humans to moving strawberries onto plates. This suggests a high fixed cost presented by robust grading itself, such that the bottleneck difficulty isn't coming from the varying complexity of the goals (e.g. human values vs moving strawberries) by which we grade.
   4. Robust grading incentivizes the AI to trick the evaluation function (if possible), and the evaluation function must be hardened to not get tricked. This violates the non-adversarial principle.

2. The loss function doesn’t have to robustly and directly reflect what you want.
    1. A loss function is a tool which chisels circuits into networks. Most outer/inner alignment frames assume that that tool should *also embody* the goals we want to chisel into the network. When chiseling a statue, the chisel doesn’t have to also look like the finished statue.
    2. Shaping is empirically useful in both AI and animals. If you think about reward as exclusively “encoding” what you want, you lose track of important learning dynamics and seriously constrain your alignment strategies.
    3. “Loss-as-chisel” encourages substantive and falsifiable speculation about internals and thus about generalization behavior, and avoids the teleological confusions which arise from using the intentional stance on agents ~“wanting” to optimize their loss functions.
3. Complete and precise inner alignment seems unnecessary, anti-natural, and very hard.
    1. Humans don’t form their values by being inner-aligned to a robust grading procedure. If you look at the single time *ever* that human-compatible values have arisen in generally intelligent minds (i.e. in humans), you’ll find it *wasn’t* done through outer/inner alignment. According to shard theory, human values are *inner alignment failures on the reward circuitry in the human brain* (read carefully: this is not the usual evolution analogy!). If you aim to “solve” outer and inner alignment, you are ruling out the *only* empirically known class of methods for growing human-compatible values.
    2. Complete inner alignment on one kind of goal seems difficult and anti-natural. We’ve never observed it in reality, and it doesn’t seem necessary.
4. I dialogue with my model of someone who advocates solving alignment via inner-aligning an agent to a robust grading procedure. In particular, I discuss how some reasons for doom no longer apply in the loss-as-chisel framing.

This post wouldn’t have happened without Quintin Pope’s ideas and feedback. Thanks to David Udell for extensive brainstorming help. Thanks to Evan Hubinger, Rohin Shah, Abram Demski, Garrett Baker, Andrew Critch, Charles Foster, Nora Belrose, Leo Gao, Kaj Sotala, Paul Christiano, Peter Barnett, and others for feedback. See here for a talk based on this essay.

I think that alignment research will be enormously advantaged by dropping certain ways of outer/inner-centric thinking for most situations, even though those ways of thinking do have some use cases. Even though this essay is critical of certain ways of thinking about alignment, I want to emphasize that I appreciate and respect the work that many smart people have done through these frames.

For reasoning about trained AI systems, I like Evan Hubinger’s “training stories” framework:

> A *training story* is a story of how you think training is going to go and what sort of model you think you’re going to get at the end[…]

In a training story, the *training goal* is a mechanistic description of the model you hope to train, and the *training rationale* explains why you’ll train the desired model and not something else instead.

One popular decomposition of AI alignment is (roughly) into outer alignment and inner alignment. These subproblems were originally defined as follows:

> “The outer alignment problem is an alignment problem between the system and the humans outside of it (specifically between the base objective and the programmer’s intentions). In the context of machine learning, outer alignment refers to aligning the specified loss function with the intended goal, whereas inner alignment refers to aligning the mesa-objective of a mesa-optimizer with the specified loss function.”
>
> — Risks from Learned Optimization: Introduction

More recently, Evan Hubinger defined these subproblems as:

> **Outer alignment** refers to the problem of finding a loss/reward function such that the training goal of “a model that optimizes for that loss/reward function” would be desirable.
>
> **Inner alignment** refers to the problem of constructing a training rationale that results in a model that optimizes for the loss/reward function it was trained on.

I initially found these concepts appealing. Even recently, I found it easy to nod along: *Yeah, we compute a reward function in a way which robustly represents what we want. That makes sense. Then just target the inner cognition properly. Uh huh. What kind of reward functions would be good?*

But when I try to imagine any *concrete real-world* situation in which these conditions obtain*, I cannot.* I might conclude “Wow, alignment is unimaginably hard!”. *No! Not for this reason, at least—The frame is inappropriate.*[4]

I: Robust grading is unnecessary, extremely hard, and unnatural

In my opinion, outer alignment encourages a strange view of agent motivation. Here’s one reasonable-seeming way we could arrive at an outer/inner alignment view of optimization:

> An agent which makes diamonds has to, at least implicitly, consider a range of plans and then choose one which in fact leads to diamonds. To do this, the AI (or “actor”) has to (at least implicitly) grade the plans. The actor needs a grading procedure which, when optimized against, leads to the selection of a diamond-producing plan. Therefore, we should specify or train a grading procedure which can be optimized in this way. Let’s call this the “outer objective”, because this grading procedure is also the objective we’ll use to give the agent feedback on the plans and actions it executes.
>
> Once we find a good grading procedure, we should train the actor to be smart and make sure it actually uses the right procedure to grade plans. We align the actor so it optimizes that grading procedure (in its form as a *Platonic mathematical function* over e.g. the plan-space).[5] This is aligning the *inner cognition* which the *outer objective* is optimizing (via e.g. gradient updates), so we’ll call this

> “inner alignment.” If we solve outer and inner alignment, it sure seems like the actor should find and execute a plan which makes a lot of diamonds.

One major mistake snuck in when I said “The actor needs a grading procedure which, when optimized, leads to the selection of a diamond-producing plan.” I suspect that many (perceived) alignment difficulties spill forth from this single mistake, condemning us to an extremely unnatural and hard-to-align portion of mind-space.

Why is it a mistake? Consider what happens if you successfully inner-align the actor so that it wholeheartedly searches for plans which maximize grader evaluations (e.g. “how many diamonds does it seem like this plan will lead to?”). In particular, I want to talk about what this agent “cares about”, or the factors which influence its decision-making. What does this inner-aligned actor care about?

Agents which care about the outer objective will make decisions on the basis of the output of the outer objective. Maximizing evaluations is the *terminal purpose* of the inner-aligned agent’s cognition. Such an agent is not making decisions on the basis of e.g. diamonds or having fun. That agent is *monomaniacally optimizing for high outputs.*

On the other hand, agents which terminally value diamonds will make decisions on the basis of diamonds (e.g. via learned subroutines like “IF `diamond nearby`, THEN bid to set `planning subgoal: navigate to diamond`”). Agents which care about having fun will make decisions on the basis of having fun. Even though people often evaluate plans (e.g. via their gut) and choose the plan they feel best about (e.g. predicted to lead to a fun evening), finding a highly-evaluated plan isn’t the *point* of the person’s search. The point is to have fun. For someone who values having fun, the *terminal purpose* of their optimization is to have fun, and finding a highly evaluated plan is a *side effect* of that process.

“The actor needs a grading procedure which, when optimized against, leads to the selection of a diamond-producing plan” is a mistake because agents should not *terminally care about optimizing a grading procedure*. Generating highly evaluated plans should be a *side effect* of effective cognition towards producing diamonds.

Consider what the actor cares about in this setup. The actor does not care about diamond production. The actor cares about high evaluations from the objective function. These two goals (instrumentally) align if the only actor-imaginable way to get maximal evaluation is to make diamonds.

(This point is important under my current views, but it strikes me as the kind of concept which may require its own post. I’m not sure I know how to communicate this point quickly and reliably at this point in time, but this essay has languished in my drafts for long enough. For now, refer to Don't align agents to evaluations of plans and Alignment allows "nonrobust" decision-influences and doesn't require robust grading for more intuitions.)

If you inner-align the agent to the evaluative output of a Platonic outer objective, you have guaranteed the agent won’t make decisions on the same basis that you do. This is because you don’t, on a mechanistic level, terminally value high outputs from that outer objective. This agent will be aligned with you only if you achieve “objective robustness”—i.e. force the agent to make diamonds in order to get high evaluations by the outer objective.

> It's like saying, "What if I made a superintelligent sociopath who only cares about making toasters, and then arranged the world so that the only possible way they can make toasters is by making diamonds?" Yes, *possibly* there do exist ways to arrange the world so as to satisfy this strange plan. But it's just deeply unwise to try to do! Don't make them care about making toasters, or about evaluations of how many diamonds they're making… Make them care about diamonds.
>
> [...]
>
> Don't align an agent to *evaluations which are only nominally about diamonds*, and then expect the agent to care about diamonds! You wouldn't align an agent to care about cows and then be surprised that it didn't care about diamonds. Why be surprised here?
>
> Grader-optimization fails because *it is not the kind of thing that has any right to work*. If you want an actor to optimize X but align it with evaluations of X, you shouldn't be surprised if you can't get X out of that.
>
> ~ Don't align agents to evaluations of plans

Motivation via evaluations-of-X *incentivizes* agents to seek out adversarial inputs to the evaluative outer objective (e.g. “how many diamonds a specific simulated smart person expects of a plan”), since if there’s any possible way to get an even higher output-number, the inner-aligned agent will try to exploit that opportunity. I’m 95% confident that outer objectives will have adversarial inputs which have nothing to do with what we were attempting to grade on, because the input-space is exponentially large, the adversaries superintelligent, and real-world evaluative tasks are non-crisp/non-syntactic. This case is made in depth in don't design agents which exploit adversarial inputs. Don’t build agents which care about evaluations of X. Build agents which care about X.

This conflict-of-interest between evaluations-of-X and X is why you need to worry about e.g. “nearest unblocked strategy” and “edge instantiation” within the outer/inner alignment regime. If you’re trying to get an agent to optimize diamonds by making it optimize evaluations, of course the agent will exploit any conceivable way to get high evaluations without high diamonds. I tentatively conjecture[6] (but will not presently defend) that these problems are artifacts of the assumption that agents must be grader-optimizers (i.e. a smart “capabilities” module which optimizes for the outputs of some evaluation function, be that a utility function over universe-histories, or a grader function over all possible plans). But when I considered the problem with fresh eyes, I concluded that alignment allows "nonrobust" decision-influences and doesn't require robust grading.

The answer is not to find a clever way to get a robust outer objective. The answer is to not *need* a robust outer objective. Robust grading incentivizes an inner-aligned AI to search for upwards errors in your grading procedure, but I think it’s easy to tell plausible training stories which don’t require robust outer objectives.

Outer/inner introduces indirection

We want an AI which takes actions which bring about a desired set of results (e.g. help us with alignment research or make diamonds). Outer/inner proposes getting the AI to care about optimizing some objective function, and hardening the objective function such that it’s best optimized by e.g. helping us with alignment research. This introduces indirection—the AI cares about the objective function, which then gets the AI to behave in the desired fashion. Just cut out the middleman and entrain the relevant decision-making influences into the AI.

Outer/inner violates the non-adversarial principle

We shouldn’t build an agent where the inner agent spends a ton of time thinking hard about how to get high evaluations / output-of-outer-objective, while also we have to specify an objective function which can *only* be made to give high evaluations if the agent does what we want. In such a situation, the outer objective has to spend extra compute to not get tricked by the inner agent doing something which only *looks* good. I think it’s far wiser to entrain decision-making subroutines which are thinking about how to do what we want, and cut out the middleman represented by an adversarially robust outer objective.

> We should not be constructing a computation that is *trying* to hurt us. At the point that computation is running, we've already done something foolish--willfully shot ourselves in the foot. Even if the AI doesn't find any way to do the bad thing, we are, at the very least, wasting computing power.
>
> [...] If you're building a toaster, you don't build one element that heats the toast and then add a tiny refrigerator that cools down the toast.
>
> Non-adversarial principle, Arbital

In Don't align agents to evaluations of plans, I wrote:

> In the intended motivational structure, the actor tries to trick the grader, and the grader tries to avoid being tricked. I think we can realize massive alignment benefits by not designing motivational architectures which require extreme robustness properties and whose parts work at internal cross-purposes.

There are no known outer-aligned objectives for any real-world task

It’s understandable that we haven’t found an outer objective which “represents human values” (in some vague, possibly type-incorrect sense). Human values *are* complicated, after all. What *can* we specify? What about diamond maximization? Hm, that problem also hasn’t yielded. Maybe we can just get the AI to duplicate a strawberry, and then do nothing else? What an innocent-sounding task! Just one tiny strawberry! Just grade whether the AI made a strawberry and did nothing, or whether it did some other plan involving more than that!

We can do none of these things. We don't know how to design an argmax agent, operating in reality with a plan space of plans *about* reality, such that the agent chooses a plan which a) we ourselves could not have specified and b) does what we wanted.

At first pass, this seems like evidence that alignment is hard. In some worlds where alignment is easy, “just solve outer alignment” *worked*. We *were able* to “express what we wanted.” Perhaps, relative to your subjective uncertainty, “just solve outer alignment” happens in fewer worlds where alignment is hard. Since “just solve outer alignment” *isn’t known to work for pinning down* ***any*** *desirable real-world behavior which we didn’t already know how to specify*, we update (at least a bit) towards “alignment is hard.”

But also, we update towards “outer/inner is just a bad frame.” Conditional on my new frame, there isn’t an “alignment is hard” update. Repeated failures at outer alignment don’t discriminate between worlds where cognition-updating-via-loss is hard or easy to figure out in time.

II: Loss functions chisel circuits into networks

In this section, I use “reward” and “loss” somewhat interchangeably, with the former bearing a tint of RL.

A loss function is a tool which chisels cognitive grooves into agents. Mechanistically, loss is not the optimization target, loss is not the “ground truth” on whether a state is good or not—loss chisels cognition into the agent’s mind. A given training history and loss/reward schedule yields a sequence of cognitive updates to the network we’re training. That’s what reward does in the relevant setups, and that’s what loss does in the relevant setups.

As Richard Ngo wrote in *AGI safety from first principles: Alignment*:

> In trying to ensure that AGI will be aligned, we have a range of tools available to us - we can choose the neural architectures, RL algorithms, environments, optimisers, etc, that are used in the training procedure. We should think about our ability to specify an objective function as the most powerful such tool. Yet it’s not powerful because the objective function defines an agent’s motivations, but rather because samples drawn from it shape that agent’s motivations and cognition. From this perspective, we should be less concerned about what the extreme optima of our objective functions look like...

The *mechanistic function* of loss is to supply cognitive updates to an agent. In policy gradient methods, rewarding an agent for putting away trash will reinforce / generalize the computations which produced the trash-putting-away actions. Reward’s mechanistic function is not necessarily to be the quantity which the agent optimizes, and—*when you look at the actual math implementing cognition-updating in deep learning*—reward/loss does not have the type signature of *goal/that-which-embodies-preferences*. I have already argued why agents probably won’t end up primarily optimizing their own reward signal. And that’s a good thing!

Loss-as-chisel is mathematically correct

I kinda thought that when I wrote Reward is not the optimization target, people would *click* and realize “Hey, I guess outer and inner alignment were leaky frames on the true underlying update dynamics, and if we knew what we were doing, we could just control the learned cognition via the cognitive-update-generator we provide (aka the reward function). This lets us dissolve the nearest unblocked strategy problem—how amazing!” This, of course, proved wildly optimistic. Communication takes effort and time. So let me continue from that trailhead.

Let’s compare loss-as-chisel with a more common frame for analysis:

1. **A naive “reward-optimized” view.** The training process optimizes the network to get lots of reward/low loss.
2. **Loss-as-chisel.** Reward and loss provide a sequence of gradients on the empirical data distribution. Each gradient changes the generalization properties.

Rohin Shah likes to call (1) “deep learning's Newtonian mechanics” and (2) the “quantum mechanics”, in that (2) *more faithfully* describes the underlying learning process, but is harder to reason about. But often, when I try to explain this to alignment researchers, they don’t react with “Oh, yeah, but I just use (1) as a shortcut for (2).” Rather, they seem to react, “What an interesting Shard Theory Perspective you have there.” Rohin has told me that his response to these researchers would be: “Your abstraction (1) is leaky under the true learning process which is *actually happening*, and you should be sharply aware of that fact.”

Loss-as-chisel encourages thinking about the mechanics and details of learning

Loss-as-chisel encourages substantive and falsifiable speculation about internals and thus about generalization behavior. Loss-as-chisel also avoids the teleological confusions which arise from using the intentional stance to view agents as ~“wanting” to optimize their loss functions.[7] I consider a bunch of "what is outer/inner alignment" discourse and debate to be confusing, even still, even as a relatively senior researcher. Good abstractions hew close to the bare-metal of the alignment problem. In this case, I think we should hew closer to the actual learning process. (See also Appendix B for an example of this.)

By taking a more faithful loss-as-chisel view on deep learning, I have realized enormous benefits. Even *attempting* to mechanistically consider a learning process highlights interesting considerations and—at times—vaporizes confused abstractions you were previously using.

For example, I asked myself “when during training is it most important to provide ‘high-quality’ loss signals to the network?”. I realized that if you aren’t aiming for inner alignment on a robust grading procedure represented by the loss function, it probably **doesn’t matter** what the loss function outputs in some late-training and any deployment situations (e.g. what score should you give to a plan for a high-tech factory?).

At that stage, a superintelligent AI could just secretly set its learning rate to zero if it didn’t want to be updated, and then the loss signal wouldn’t matter. And if it did want to be updated, it could set the loss itself. So when the AI is extremely smart, it doesn’t matter *at all* what reward/loss signals look like. This, in turn, suggests (but does not decisively *prove*) we focus our efforts on early- and mid-training value development. Conveniently, that’s the part of training when supervision and interpretability is easier (although still *quite hard*).

Loss doesn’t have to “represent” intended goals

Outer/inner *unnecessarily assumes* that the loss function/outer objective should “embody” the goals which we want the agent to pursue.

For example, shaping is empirically useful in both AI and animals. When a trainer is teaching a dog to stand on its hind legs, they might first give the dog a treat when it lifts its front paws off the ground. This treat translates into an internal reward event for the dog, which (roughly) reinforces the dog to be more likely to lift its paws next time. The point isn’t that we *terminally value* dogs lifting their paws off the ground. We do this because it reliably shapes target cognition (e.g. stand on hind legs on command) into the dog. If you think about reward as exclusively “encoding” what you want, you lose track of important learning dynamics and seriously constrain your alignment strategies. (See Some of my disagreements with List of Lethalities for a possible example of someone being hesitant to use reward shaping because it modifies the reward function.)

Be precise when reasoning about outer objectives

I also think that people talk extremely imprecisely and confusingly about “loss functions.” I get a lot of mileage out of being precise—if my idea is right in generality, it is right in specificity, so I might as well start there. In Four usages of "loss" in AI, I wrote:

> What does it *mean* for a loss function to be "aligned with" human goals? I perceive four different concepts which involve "loss function" in importantly different ways:
>
> 1. *Physical-loss:* The physical implementation of a loss function and the loss computations,
> 2. *Mathematical-loss:* The mathematical idealization of a loss function,
> 3. A loss function "encoding/representing/aligning with" an intended goal, and
> 4. Agents which "care about achieving low loss."
>
> I advocate retaining physical- and mathematical-loss. I advocate dropping 3 in favor of talking directly about desired AI cognition and how the loss function entrains that cognition. I advocate disambiguating 4, because it can refer to a range of physically grounded preferences about loss (e.g. low value at the loss register versus making perfect future predictions).

> “Outer Alignment in the context of machine learning is the property where the specified loss function is aligned with the intended goal of its designers. This is an intuitive notion, in part because human intentions are themselves not well-understood.” Outer Alignment — LessWrong

I think that outer alignment is an “intuitive notion” in part because *loss functions don’t natively represent goals.* For agents operating in reality, extra interpretation is required to view loss functions as representing goals. I can imagine, in detail, what it would look like to use a loss function to supply a stream of cognitive updates to a network, such that the network ends up reasonably aligned with my goals. I cannot imagine what it would mean for a physically implemented loss function to be “aligned with my goals.” I notice confusion and unnaturality when I try to force that mental operation.

This “optimize the loss function” speculation is weird and sideways of how we actually get AI to generalize how we want. Here’s a small part of an outer/inner training story:

> Find a robust diamond-grading loss function which optimizes the network so that the network wants to optimize the loss function which optimized it. When the agent optimizes the loss function as hard as it can, the agent makes diamonds.

This is just, you know, ***so*** *weird. Why would you use a loss function or reward function this way?!*

According to me, the bottleneck hard problem in AI alignment is *how do we predictably control the way in which an AI generalizes; how do we map outer supervision signals (e.g. rewarding the agent when it makes us smile) into the desired inner cognitive structures (e.g. the AI cares about making people happy)?*

Here’s what I think we have to do to solve alignment: We have to know how to produce powerful human-compatible cognition using large neural networks. If we can do that, I don’t give a damn what the loss function looks like. It truly doesn’t matter. Use the chisel to make a statue and then toss out the chisel. If you’re making a statue, your chisel doesn’t also have to look like the statue.

III: Outer/inner just isn’t how alignment works in people

Inner and outer alignment decompose one hard problem (AI alignment) into two extremely hard problems. Inner and outer alignment *both* cut against known grains of value formation.

Inner alignment seems anti-natural

We have all heard the legend of how evolution selected for inclusive genetic fitness, but all it got was human values. I think this analogy is relatively loose and inappropriate for alignment, but it’s proof that inner alignment failures *can* happen in the presence of selection pressure. Far more relevant to alignment is the crush of empirical evidence from real-world general intelligences with reward circuitry, suggesting to us billions and billions of times over that *reinforcement learning at scale within a certain kind of large (natural) neural network does not primarily produce inner value shards oriented around their reward signals, or the world states which produce them*.

When considering whether human values are inner-aligned to the human reward circuitry, you only have to consider the *artifact which evolution found.* Evolution found the genome, which—in conjunction with some environmental influences—specifies the human learning process + reward circuitry. You don't have to consider *why* evolution found that artifact (e.g. selection pressures favoring certain adaptations). For this question, it might help to imagine that the brain teleported into existence from some nameless void.

From my experience with people, I infer that they do not act to maximize some simple function of their internal reward events. I further claim that people do not strictly care about bringing about the activation preconditions for their reward circuitry (e.g. for a sugar-activated reward circuit, those preconditions would involve eating sugar). True, people like sugar, but what about artificial sweeteners? Isn't that a bit "unaligned" with our reward circuitry, in some vague teleological sense?

More starkly, a soldier throwing himself on a grenade is not acting (either consciously or subconsciously) to most reliably bring about the activation preconditions for some part of his reward system. I infer that he is instead executing lines of cognition chiseled into him by past reinforcement events. He is a value shard-executor, not an inner-aligned reward maximizer. Thus, his values of protecting his friends and patriotism constitute *inner alignment failures* on the reward circuitry which brought those values into existence.[8] Those values are not aligned with the goals "represented by" that reward circuitry, nor with the circuitry's literal output. I think that similar statements hold for values like "caring about one's family", "altruism", and "protecting dogs."

Therefore, the ***only time human-compatible values have ever arisen, they have done so via inner alignment failures***.[9] Conversely, if you aim to "solve" inner alignment, **you are ruling out the only empirically known way to form human-compatible values.** Quintin Pope wrote (emphasis mine):

> Prior to Dissolving the Fermi Paradox, people came up with all sorts of wildly different solutions to the paradox, as you can see by looking at its Wikipedia page. Rather than address the underlying assumptions that went into constructing the Fermi paradox, these solutions primarily sought to add additional mechanisms that seemed like they might patch away the confusion associated with the Fermi paradox.
>
> However, the true solution to the Fermi paradox had nothing to do with any of these patches. No story about why aliens wouldn't contact Earth or why technological civilizations invariably destroyed themselves would have ever solved the Fermi paradox, no matter how clever or carefully reasoned. Once you assume the incorrect approach to calculating the Drake equation, no amount of further reasoning you perform will lead you any further towards the solution, not until you reconsider the form of the Drake equation.
>
> **I think the Fermi paradox and human value formation belong to a class of problems, which we might call "few-cruxed problems" where progress can be almost entirely blocked by a handful of incorrect background assumptions. For few-crux problems, the true solution lies in a part of the search space that's nearly inaccessible to anyone working from said mistaken assumptions.**
>
> The correct approach for few-cruxed problems is to look for solutions that take away complexity, not add more of it. The skill involved here is similar to noticing confusion, but can be even more difficult. Oftentimes, the true source of your confusion is not the problem as it presents itself to you, but some subtle

> assumptions (the “cruxes”) of your background model of the problem that caused no telltale confusion when you first adopted them.
>
> A key feature of few-cruxed problems is that the amount of cognitive effort put into the problem before identifying the cruxes tells us almost nothing about the amount of cognitive work required to make progress on the problem once the cruxes are identified. **The amount of cognition directed towards a problem is irrelevant if the cognition in question only ever explores regions of the search space which lack a solution.** It is therefore important not to flinch away from solutions that seem “too simple” or “too dumb” to match the scale of the problem at hand. Big problems do not always require big solutions.
>
> I think one crux of alignment is the assumption that human value formation is a complex process. The other crux (and I don't think there's a third crux) is the assumption that we should be trying to avoid inner alignment failures. **If (1) human values derive from an inner alignment failure [with respect to] to [human reward circuitry], and (2) humans are the only places where human values can be found, then an inner alignment failure is the only process to have ever produced human values in the entire history of the universe.**
>
> If human values derive from inner alignment failures, and we want to instill human values in an AI system, then the default approach should be to understand the sorts of values that derive from inner alignment failures in different circumstances, then try to arrange for the AI system to have an inner alignment failure that produces human-compatible values.
>
> **If, after much exploration, such an approach turned out to be impossible, then I think it would be warranted to start thinking about how to get human-compatible AI systems out of something other than an inner alignment failure. What we actually did was almost completely wall off that entire search space of possible solutions and actively try to solve the inner alignment "problem".**
>
> **If the true solution to AI alignment actually looks anything like "cause a carefully orchestrated inner alignment failure in a simple learning system", then of course our assumptions about the complexity of value formation and the undesirability of inner alignment failures would prevent us from finding such a solution. Alignment would look incredibly difficult because the answer would be outside of the subset of the solution space we'd restricted ourselves to considering.**

(I caution that "cause a carefully orchestrated inner alignment *failure* in a simple learning system" sounds like we’re trying something “hacky” or “mistake-prone”, when we really aren’t attempting something strange. Rather, we’re talking about the apparently natural way for values to form.)

The above argues that inner alignment is *un*natural—counter to natural tendencies. I further infer that inner alignment is unnatural partly *because* it is antinatural. We've never seen it happen, we don't know how to make it happen, there are lots of reasons to think it won't happen, and I don't think we need to make it happen.

Complete inner alignment seems unnecessary

In the AXRP interview, Paul stated that he would (under the outer/inner frame) aim for an agent “not doing any other optimization beyond pursuit of [the outer objective].” But *why* must there be *no* other optimization? Why can’t the AI value a range of quantities?

On how I use words, values are decision-influences (also known as *shards*). “I value doing well at school” is a short sentence for “in a range of contexts, there exists an influence on my decision-making which upweights actions and plans that lead to e.g. learning and good grades and honor among my classmates.”

An agent with lots of values (e.g. coffee and sex and art) will be more likely to choose plans which incorporate positive features under all of the values (since those plans get bid for by many decision-influences). I believe that this complexity of value is the default. **If an AI strongly and reflectively values both protecting people and paperclips, it will make decisions on the basis of both considerations.** Therefore, the AI will both protect people and make paperclips (assuming the values work in the described way, which is a whole ‘nother can of worms).

I have written:

> People care about lots of things, from family to sex to aesthetics. My values / decision-influences don't collapse down to any one of these.
>
> I think AIs will learn lots of values by default. I don't think we need all of these values to be aligned with human values. I think this is quite important.
>
> - I think the more of the AI's values we align to care about us and make decisions in the way we want, the better. (This is vague because I haven't yet sketched out AI internal motivations which I think would actually produce good outcomes. On my list!)
> - I think there are strong gains from trade possible among an agent's values. If I care about bananas and apples, I don't need to split my resources between the two values, I don't need to make one successor agent for each value. I can drive to the store and buy both bananas and apples, and only pay for fuel once.
>   - This makes it lower-cost for internal values handshakes to compromise; it's less than 50% costly for a power-seeking value to give human-compatible values 50% weight in the reflective utility function.
> - I think there are thresholds at which the AI doesn't care about us sufficiently strongly, and we get no value.
>   - EG I might have an "avoid spiders" value which is narrowly contextually activated when I see spiders. But then I think this is silly because spiders are quite interesting, and so I decide to go to exposure therapy and remove this decision-influence. We don't want human values to be outmaneuvered in this way.
>   - More broadly, I think "value strength" is a loose abstraction which isn't uni-dimensional. It's not "The value is strong" or "The value is weak"; I think values are contextually activated, and so they don't just have a global strength.
> - Even if you have to get the human-aligned values "perfectly right" in order to avoid Goodharting (~~which I am unsure of~~ ETA I don’t believe this), not having to get *all* of the AI's values perfectly right is good news.
> - I think these considerations make total alignment failures easier to prevent, because as long as human-compatible values are something the AI

meaningfully cares about, we survive.
- I think these considerations make total alignment success more difficult, because I expect agents to eg terminalize common instrumental values. Therefore, it's very hard to end up with e.g. a single dominant value which only cares about maximizing diamonds. I think that value is complex by default.

So ultimately, I think “the agent has to exclusively care about this one perfect goal” is dissolved by the arguments of alignment allows "nonrobust" decision-influences and doesn't require robust grading. And trying to make an agent only care about one goal seems to go against important grains of effective real-world cognition.

Outer alignment seems unnatural

People are not inner-aligned to their reward circuitry, nor should they be. The human reward circuitry does not specify an ungameable set of incentives such that, if the reward circuitry is competently optimized, the human achieves high genetic fitness, or lives a moral and interesting life, or anything else. As Quintin remarked to me, “If you find the person with the highest daily reward activation, it’s not going to be Bill Gates or some genius physicist.” According to Atlantic’s summary of a 1986 journal article:[10]

> In order to relieve insufferable chronic pain, a middle-aged American woman had a single electrode placed in a part of her thalamus on the right side. She was also given a self-stimulator, which she could use when the pain was too bad. She could even regulate the parameters of the current. She quickly discovered that there was something erotic about the stimulation, and it turned out that it was really good when she turned it up almost to full power and continued to push on her little button again and again.
>
> In fact, it felt so good that the woman ignored all other discomforts. Several times, she developed atrial fibrillations due to the exaggerated stimulation, and over the next two years, for all intents and purposes, her life went to the dogs. Her husband and children did not interest her at all, and she often ignored personal needs and hygiene in favor of whole days spent on electrical self-stimulation. Finally, her family pressured her to seek help. At the local hospital, they ascertained, among other things, that the woman had developed an open sore on the finger she always used to adjust the current.

That’s what happens when the human reward circuitry is somewhat competently optimized. Good thing we aren’t inner-aligned to our reward circuitry, because it isn’t “outer-aligned” in any literal sense. But even in a more abstract sense of "outer alignment", I infer that human values have not historically arisen from optimizing a “hard-to-game” outer criterion which specifies those values.

David Udell made an apt analogy:

> Say that you were raising a kid. One childrearing scheme is to carefully make your kid's childhood robust to arbitrary levels of precocious genius in your kid. You'd build a childhood such that overachieving in it would only ever be a good thing. You'd drill athletics, navigating complex adult social situations, difficult moral dilemmas, etc., always making sure that there isn't some perverse victory condition way up near the skill ceiling of the task. For on this approach, you

> don't *ever* want optimization power being pointed in a direction you wouldn't want to see optimized, in the way that you don't ever want to needlessly point a gun barrel at anything you don't want destroyed.
>
> You'll notice that the above approach to childrearing is pretty weird[...] It's in fact *okay* for behavior to be momentarily incentivized in childhood that you would not want to see optimized in adulthood! [...] It's just not a very good model of a growing human to see them as a path-independent search over policies that you have to be perfectly cautious about *ever, even temporarily,* incentivizing in a way you wouldn't want to see intelligently optimized. Indeed, ignoring that young people can *actively steer away* from events that would change who they are and what they'd care about means prematurely giving up on most viable childrearing schemes! You'd be ill-advised as a new father if someone started you off explaining that a child is a search run over algorithms incentivized by the environment, rather than by foregrounding the theory of human inductive biases and human flavors of path-dependent aging.

As best I can tell, human values have never arisen via the optimization of a hard-to-game outer criterion which specifies their final form. That doesn't *logically imply* that human values can't arise in such a way—although I have separately argued that they won't—but it's a clue.

Why does it matter how alignment works in people?

Suppose we came up with outer/inner alignment as a frame on AI alignment. Then we realized that people *do* seem to contain an "outer objective"—neural circuitry which people *terminally want* to optimize (i.e. the genome inner-aligns people to the circuitry) such that the neural circuitry faithfully represents the person's motivations (i.e. the neural circuitry is an outer alignment encoding of their objective). I would react: "Huh, looks like we really have reasoned out something true and important about how alignment works. Looks like we're on roughly the right track."

As I have argued, this does not seem to be the world we live in. Therefore, since inferring outer/inner alignment in humans would have increased my confidence in the outer/inner frame, inferring *not*-outer/inner must necessarily decrease my confidence in the outer/inner frame by conservation of expected evidence.

IV: Dialogue about inner/outer alignment

Communication is hard. Understanding is hard. Even if I fully understood what other people are trying to do (I don't), I'd still not have space to reply to every viewpoint. I'm still going to say what I think, do my best, and be honest. I expect to be importantly right, which is why I'm sharing this essay. As it stands, I'm worried about much of the field and the concepts being used.

Alex's model of an outer alignment enjoyer (A-Outer): Outer/inner alignment is cool because it lets us decompose "what we want the agent to care about" and "how we get the agent to care about that." This is a natural problem decomposition and lets us allocate the agent's motivations to the part we have more specification-level control over (i.e. its reward function).

Alex (A): I don't think it makes sense to design an agent to have an actor/grader motivational structure. As I've discussed, I think those design patterns are full of landmines.

A-Outer: I think we can recover the concept if we just let "outer alignment" be "what cognition / values should the AI have?".

A: That is indeed important to think about. That's also *not* aiming for an "outer-aligned" reward function or grading procedure. Don't pollute the namespace—allocate different phrases to different concepts. That is, you can consider "what values should the AI have?" and *then* "what reward function will chisel those values into the AI?". But then we aren't inner-aligning the agent *to the outer objective* anymore, but rather we are producing the *desired internal values*. We're now reasoning about reward-chiseling, which I'm a big fan of.

A-Outer: Right, but you have to admit that "consider what kinds of objectives are safe to maximize" is *highly relevant* to "what do we want the AI to end up doing for us?". As you just agreed, we obviously want to understand that.

(And yes, *maximization*. Just look at the coherence theorems spotlighting expected utility maximization as the thing which non-stupid real-world agents do! Unless you think we won't get an EU maximizer?)

A: Compared to "what reward signal-generators are safe to optimize?", it's *far more* reasonable to consider "what broad-strokes *utility* function should the AI optimize?". Even so, there are *tons* of skulls along that path. We just suck at coming up with utility functions which are safe to maximize, for generalizable reasons. Why should a modern alignment researcher spend an additional increment of time thinking about *that* question, instead of other questions? Do you think that we'll *finally* find the clever utility function/grading procedure which is robust against adversarial optimization? I think it's wiser to simply avoid design patterns which pit you against a superintelligence's adversarial optimization pressure.

(And I don't think you'll get a meaningfully viewable-as-bounded-EU-maximizer until late in the agent's developmental timeline. That might be a very important modeling consideration. Be careful to distinguish asymptotic limits from finite-time results.)

A-Outer: Seriously? It would be real progress to solve the outer alignment problem in terms of writing down a utility function over universe-histories which is safe to maximize. For example, suppose we learned that if the utility function penalizes the agent for gaining more than *X* power for >1 year (in some formally specifiable sense) would bound the risk from that AI, making it easier to get AIs which do pivotal acts without keeping power forever. Then we learn something about the properties we might aim to chisel into the AI's inner cognition, in order to come out alive on the other side of AGI.

A: First, note that your argument is for finding a safe-to-maximize *utility function over universe histories*, which is not the same as the historically prioritized *reward-outer-alignment*. Second, not only do I think that your hope won't happen, I think the hope is written in an ontology which doesn't make sense.

Here’s a non-strict analogy which hopefully expresses some of my unease. Your hope feels like saying, “If I could examine the set of physically valid universe-histories in which I go hiking tonight, I’d have learned something about where I might trip and fall during the hike.” Like, sure? But why would I want to examine *that* mathematical object in order to not trip during the hike? Sure seems inefficient and hard to parse.

I agree that “What decision-making influences should we develop inside the AI?” is a hugely important question. I just don’t think that “what utility functions are safe to maximize?” is a sensible way to approach that question.

A-Outer: Even though we probably won’t discover a compact specification of a utility function which is *strictly and literally safe to literally maximize*, there are *degrees* of safety when a real-world agent optimizes an objective. Two objectives may be gameable, but one can still be *less* gameable than the other.

A: Sure seems like that in the outer/inner paradigm, those “degrees of safety” are irrelevant in the limit, as their imperfections burst under the strain of strong optimization. (Aren’t *you* supposed to be the discussant operating that paradigm, A-Outer?)

A-Outer: I don’t see how you aren’t basically giving up on figuring out what the AI should be doing.

A: Giving up? No! *Thinking about “what utility function over universe-histories is good?” is just <u>one</u> way of framing “How can we sculpt an AI’s internal cognition so that it stops the world from blowing up due to unaligned AI?”.* If you live and breathe the inner/outer alignment frame, you’re missing out on better framings and ontologies for alignment! To excerpt from *Project Lawful*:

> The difficult thing, in most pre-paradigmatic and confused problems at the beginning of some Science, is not coming up with the right complicated long sentence in a language you already know. It's breaking out of the language in which every hypothesis you can write is false. [...] The warning sign that you need to 'jump-out-of-the-system' is the feeling [of] frustration, flailing around in the dark, trying desperate wild ideas and getting unhelpful results one after another. When you feel like that, you're probably thinking in the wrong language, or missing something fundamental, or trying to do something that is in fact impossible. Or impossible using the tools you have.

Stop trying to write complicated long sentences in terms of outer objectives. **Just, stop**. Let’s find a new language. (Do you really think a future alignment textbook would say “And then, to everyone’s amazement, outer alignment scheme #7,513 succeeded!”)

Now, I can legitimately point out that outer and inner alignment aren’t a good framing for alignment, *without* offering an alternative better framing. That said, I recently wrote:[11]

> Shard theory suggests that goals are more natural to specify/inculcate in their shard forms (e.g. if around trash and a trash can, then put the trash away), and not in their (presumably) final form of globally activated optimization of a coherent utility function which is the reflective equilibrium of inter-shard value-handshakes (e.g. a utility function over the agent's internal plan-ontology whose optimization leads to trash getting put away, among other utility-level reflections of initial shards).

> I *could* (and *did*) hope that I could specify a utility function which is safe to maximize because it penalizes power-seeking. I may as well have hoped to jump off of a building and float to the ground. On my model, that's just not how goals work in intelligent minds. If we've had anything at all beaten into our heads by our alignment thought experiments, it's that *goals are hard to specify in their final form of utility functions.*
>
> I think it's time to think in a different specification language.[12]

A-Outer: Bah, “shard theory of human values.” We didn’t build planes with flapping wings. Who cares if human values come from inner alignment failures—Why does that suggest that we shouldn’t solve inner alignment for AI? *AI will not be like you.*

A: Yes, it is indeed possible to selectively consider historical disanalogies which support a (potentially) desired conclusion (i.e. that outer/inner is fine). If we’re going to play reference class tennis, how about all of the times biomimicry has worked?

But let’s not play reference class tennis. As mentioned above, we have to obey conservation of expected evidence here.

In worlds where inner alignment was a good and feasible approach for getting certain human-compatible values into an AI (let’s call that hypothesis class $H_{\text{inner-align}}$), I think that we would expect with greater probability for human values to naturally arise via inner alignment *successes*. However, in worlds where inner alignment failures are appropriate for getting human values into an AI (H_{fail}), we would expect with greater probability for human values to naturally arise via inner alignment *failures*.

Insofar as I have correctly inferred that human values constitute inner alignment failures on the human reward circuitry, this inference presents a decent likelihood ratio P(reality | H_{fail}) / P(reality | $H_{\text{inner-align}}$), since H_{fail} predicts inferred reality more strongly. In turn, this implies an update towards H_{fail} and away from $H_{\text{inner-align}}$. I think it's worth considering the strength of this update (I'd guess it's around a bit or so against outer/inner), but it's definitely an update.

I agree that there are important and substantial differences e.g. between human inductive biases and AI inductive biases. But I think that the evidential blow remains dealt against outer/inner, marginalizing over possible differences.

A-Outer: On another topic—What about “the outer objective gets smarter along with the agent”?

A: That strategy seems unwise for the target motivational structures I have in mind (e.g. "protect humanity" or "do alignment research").

1. Section I (robust grading is unnecessary): This plan requires an unrealistic invariant. The invariant is that the outer objective must “properly grade” every possible plan the agent is smart enough to consider. How are you possibly going to fulfill that invariant? Why would you *want* to choose a scheme where you have to fulfill such an onerous invariant?
    1. (For more detail on the concurrent-improvement case, see the appendix of Don't design agents which exploit adversarial inputs.)
2. Section II (loss is like a chisel) applies: You’re constraining the chisel to look like the statue. Why consider such a narrow class of approaches?

3. Section III (inner/outer is anti-natural) applies: That strategy seems *anti-natural* as a way of getting cognitive work out of an agent.

A-Outer: It's easy to talk big talk. It's harder to propose concrete directions which aren't, you know, *doomed*.

A: The point isn't that I have some even *more amazing and complicated scheme* which avoids these problems. The point is that I don't need one. In the void left by outer/inner, many objections and reasons for doom no longer apply (as a matter of *anticipation* and not of the problems just popping up in a different language).

In this void, *you should reconsider all fruits which may have grown from the outer/inner frame*. Scrutinize both your reasons for optimism (e.g. "maybe it's simpler to just point to the outer objective") and for pessimism (e.g. "if the graders are exploitable by the AI, the proposal fails"). See alignment with fresh eyes for a while. Think for yourself.

This is why I wrote *Seriously, what goes wrong with "reward the agent when it makes you smile"?*:

> My mood [in this post] isn't "And this is what we do for alignment, let's relax." My mood is "Why consider super-complicated reward and feedback schemes when, as far as I can tell, we don't know what's going to happen in this relatively simple scheme? How do reinforcement schedules map into inner values?"

If you're considering "reward on smile" from an outer alignment frame, then *obviously* it's doomed. But from the reward-as-chisel frame, not so fast. For that scheme to be doomed, it would have to be true that, for every probable sequence of cognitive updates we can provide the agent via smile-reward events, those updates would not build up into value shards which care about people and want to protect them. That scheme's doom is not at all clear to me.

(One objection to the above is "Ignorance of failure is no protection at all. We need a tight story for why AI goes *well.*" Well, yeah. I'm just saying "in the absence of outer/inner, it doesn't make sense to start debating hyper-complicated reward chisels like debate or recursive reward modeling, if we still can't even adjudicate what happens for 'reward on smile.' And, there seems to be misplaced emphasis on 'objective robustness', when really we're trying to get good results from loss-chiseling.")

A-Outer: Suppose I agreed. Suppose I just dropped outer/inner. What next?

A: Then you would have the rare opportunity to pause and think while floating freely between agendas. I will, for the moment, hold off on proposing solutions. Even if my proposal is good, discussing it *now* would rob us of insights you could have contributed as well. There will be a shard theory research agenda post which will advocate for itself, in due time.

A-Outer, different conversational branch. We know how to control reward functions to a much greater extent than we know how to control an AI's learned value shards.

A: This is true. And?

A-Outer: I feel like you're just ignoring the crushing amount of RL research on regret bounds and a moderate amount of research on the expressivity of reward functions and how to shape reward while preserving the optimal policy set. Literally *I* have proven a theorem[13] constructively showing how to transfer an optimal policy set from one discount rate to another. We know how to talk about these quantities. Are you seriously suggesting just tossing that out?

A: Yes, toss it out, that stuff doesn't seem very helpful for alignment thinking—including that theorem we were so proud of! Yes, toss it out, in the sense of relinquishing the ill-advised hope of outer alignment. Knowing how to talk about a quantity (reward-optimality) doesn't mean it's the most appropriate quantity to consider.

A-Outer: Consider *this*: Obviously we want to reward the agent for doing good things (like making someone smile) and penalize it for doing bad things (like hurting people). This frame is historically, empirically useful for getting good behavior out of AI.

A: First, we have *not* solved AI alignment in the inner/outer paradigm—even for *seemingly simple objectives like diamond-production and strawberry duplication*—despite brilliant people thinking in that frame for years. That is weak evidence against it being a good paradigm.

Second, I agree that all else equal, it's better to reward and penalize the agent for obvious good and bad things, respectively. But not *because* the reward function is supposed to represent what I want. As I explained, the reward function is like a chisel. If I reward the agent when it makes me smile, all else equal, that's probably going to upweight and generalize at least *some* contextual values upstream of making me smile. That reward scheme should differentially upweight and strengthen human-compatible cognition to some extent.

Since reward/loss is *actually the chisel according to the math of cognition-updating in the most relevant-seeming approaches*, insofar as your suggestion is good, it is good *because it can be justified via cognition-chiseling reasons.* Your basic suggestion might not be enough for alignment success, but it's an important part of our best current guess about what to do.

More broadly, I perceive a motte and bailey:

- *Bailey:* We should solve outer alignment by specifying a reward signal which can't reasonably be gamed and which expresses what we want / is aligned with our values. This reward signal should return good outputs far outside of the normal distribution of human experience, such that it doesn't have bad maxima.
- *Motte*: All else equal, it's better to reward the agent for doing good things (like making someone smile) and to penalize it for doing bad things (like hurting people).

I think that the bailey is wrong and the motte is right.

A-Outer: You keep wanting to focus on the "quantum mechanics" of loss-as-chisel. I agree that, in principle, if we really knew what we were doing—if we deeply understood SGD dynamics—we could skillfully ensure the network generalizes in the desired way (e.g. makes diamonds). You criticize the "skulls" visible on the "robust grader" research paths, while seemingly ignoring the skulls dotting the "just understand SGD" paths.

A: I, at the least, agree that we aren’t going to get a precise theory like “If you initialize *this* architecture and scale of foundation model on *this* kind of corpus via self-supervised learning, it will contain a diamond concept with high probability; if you finetune on *this* kind of task, it will hook up its primary decision-influences to the diamond-abstraction; …”. That seems quite possible to understand given enough time, but I doubt we’ll have that much time before the rubber hits the road.

However, I’d be more sympathetic to this concern if there wasn’t a bunch of low-hanging fruit to be had from simply realizing that loss-as-chisel *exists*, and then trying to analyze the dynamics anyways. (See basically everything I’ve written since this spring. Most of my insights have been enabled by my unusually strong desire to think mechanistically and precisely about what *actually happens* during a learning process.)

One thing which would make me more pessimistic about the “understand how loss chisels cognition into agents” project is if I don’t, within about a year’s time, have empirically verified loss-as-chisel insights which wouldn’t have happened without that frame. But even if so, everything we're doing will still be governed by loss-as-chisel. We can't ignore it and make it go away.

A-Outer: But if we do inner alignment, we *don’t* have to understand SGD dynamics to the same extent that we do to chisel in diamond-producing values.

A: I don’t know why you think that. (I don't even understand enough yet to agree or disagree in detail; I currently disagree in expectation over probable answers.)

What, exactly, are we chiseling in order to produce an inner-aligned network? How do we know we can chisel agents into that shape, if we don’t understand chiseling very well? What do we think we know, and how do we think we know it? **How is an inner-aligned diamond-producing agent supposed to be structured?** This is not a rhetorical question. I literally do not understand what the internal cognition is supposed to look like for an inner-aligned agent. Most of what I’ve read has been vague, on the level of “an inner-aligned agent cares about optimizing the outer objective.”

Charles Foster comments:

> We are attempting to mechanistically explain how an agent makes decisions. One proposed reduction is that inside the agent, there is an even **smaller** inner agent that interacts with a non-agential evaluative submodule to make decisions for the outer agent. But that raises the immediate questions of "How does the inner agent make its decisions about how to interact with the evaluative submodule?" and then "At some point, there's gotta be some non-agential causal structure that is responsible for **actually implementing decision-making**, right?" and then "Can we just explain the original agent's behavior in those terms? What is positing an externalized evaluative submodule buying us?"

Perhaps my emphasis on mechanistic reasoning and my unusual level of precision in my speculation about AI internals, perhaps these make people realize how *complicated* realistic cognition is in the shard picture. Perhaps people realize how much might have to go right, how many algorithmic details may need to be etched into a network so that it does what we want and generalizes well.

But perhaps people don’t realize that a network which is inner-aligned on an objective will *also* require a precise and conforming internal structure, and they don’t realize this because no one has written detailed plausible stabs at inner-aligned cognition.

A-Outer: Just because the chisel frame is technically accurate doesn't mean it's the most pragmatically appropriate frame. The outer alignment frame can abstract over the details of cognition-chiseling and save us time in designing good chiseling-schemes. For example, I can just reward the AI when it wins the game of chess, and not worry about designing reward schedules according to my own (poor) understanding of chess and what chess-shards to upweight.

A: I agree that sometimes you should just think about directly incentivizing the outcomes and letting RL figure out the rest; I think that your chess example is quite good! Chess is fully observable and has a crisply defined, algorithmically gradable win condition. Don't worry about "if I reward for taking a queen, what kind of cognition will that chisel?"—just reinforce the network for winning.

However, is the "reward outcomes based on their 'goodness'" frame *truly* the most appropriate frame for AGI? If that *were* true, how would we know? I mean—*gestures at probability theory intuitions*—however outer alignment-like concepts entered the alignment consciousness, it was not (as best I can discern) *because* outer alignment concepts are optimally efficient for understanding how to chisel good cognition into agents.[14] Am I now to believe that, *coincidentally,* this outer alignment frame is *also* the most appropriate abstraction for understanding how to e.g. chisel diamond-producing values into policy networks? How fortuitous!

A-Outer: Are you saying it's never appropriate to consider outer/inner, then?

A: I think that the terminology and frame are unhelpful. At least, I feel drastically less confused in my new primary frame, and people have told me my explanations are quite clear and focused in ways which I think relate to my new frame.

In e.g. the chess example, though, it seems fine to adopt the "Newtonian mechanics" optimized-for-reward view on deep learning. Reward the agent for things you want to happen, in that setting. Just don't forget what's *really* going on, deeper down.

A-Outer: Even if the inner/outer alignment problem isn't literally solvable in literal reality, it can still guide us to good ideas.

A: Many things can guide us to good ideas. Be careful not to privilege a hypothesis which was initially elevated to consideration for reasons you may no longer believe!

Conclusion

Inner and outer alignment decompose one hard problem (AI alignment) into two *extremely* hard problems. These problems go against natural grains of cognition, so it's unsurprising that alignment has seemed extremely difficult and unnatural. Alignment still seems difficult to me, but not because e.g. we have to robustly grade plans in which superintelligences are trying to trick us.

1. **Robust grading is extremely difficult and also unnecessary.** The answer is not to find a clever way to get a robust outer objective. The answer is to not *need* a robust outer objective. If you find yourself trying to grade arbitrary-case outputs from an unaligned superintelligence, you probably framed the problem wrongly by using robust-grading design patterns.

2. **The loss function chisels cognition into the AI.**
3. **If you aim to “solve” inner or outer alignment, you are ruling out the only empirically known way to form human-compatible values.**

I think that “but what about applying optimization pressure to the base objective?” has warped lots of alignment thinking. You don’t need an “extremely hard to exploit” base objective. That’s a red herring.

Stepping away from the worldview in which outer/inner is a reasonable frame, a range of possibilities open up, and the alignment problem takes on a refreshing and different nature. We need to *understand how to develop good kinds of cognition in the networks we train* (e.g. how to supply a curriculum and reward function such that the ensuing stream of cognitive-updates leads to an agent which cares about and protects us). At our current level of understanding, *that’s* the bottleneck to solving technical alignment.

Appendix A: Additional definitions of “outer/inner alignment”

Here are a few more definitions, for reference on how the term has been historically defined and used.

Evan’s definitions

> “My definition says that an objective function *[P]* is outer aligned if all models optimal under *[P]* in the limit of perfect optimization and unlimited data are aligned.”
>
> Evan Hubinger, commenting on "Inner Alignment Failures" Which Are Actually Outer Alignment Failures

I will note that the human reward circuitry is not outer-aligned to human values under this definition, since people who experience the “data” of wireheading will no longer have their old values.

Anyways, It’s not clear what this definition means in the RL setting, where high path-dependence occurs due to the dependence of the future policy on the future training data, which in turn depends on the current policy, which depended on the past training data. For example, if you like candy and forswear dentists (and also forswear ever updating yourself so that you will go see the dentist), you will never collect reward data from the dentist’s office, and vice versa. One interpretation is: infinite exploration of all possible state-action tuples, but I don’t know what that means in reality (which is neither ergodic nor fully observable). I also don’t know the relative proportions of the “infinite data.”

Evan privately provided another definition which better accounts for the way he currently considers the problem of outer+inner alignment:

> A model that has the same goal that the loss/reward function describes. So if the loss function rewards agents for getting gold coins, then the training goal is an agent that terminally cares about gold coins.

I then wrote a dialogue with my model of him, which he affirmed as “a pretty reasonable representation.”

Alex (A): Hm. OK. So it sounds like the outer objective is less of something which grades the agent directly across all situations, and which is safe to optimize *for.* Under your operationalization of the outer alignment training goal, the reward function is more like an artifact which emits reward on training in a way which tightly correlates with getting gold coins on training.

Suppose I have an embodied AI I’m training via RL (for conceptual simplicity, not realism), and it navigates mazes and reaches a gold coin at the end of each maze. I’ll just watch the agent through one-way glass and see if it looks like it touched the gold coin by legit solving the maze. If it does, I hit the reward button.

Now suppose that this in fact just trains a smart AI which “terminally cares” about gold coins, operationalized in the “values as policy-influences” sense: In all realistically attainable situations where the AI believes there are gold coins nearby, the AI reliably reaches the gold coin. The AI doesn’t go to yellow objects, or silver coins, or any other junk.

So even though on training, the reward schedule was unidentifiable from “reward when a metal disk was touched”, that doesn’t matter for our training goal. We just want the AI to learn a certain kind of cognition which we “had in mind” when specifying the outer objective, and it doesn’t matter if the outer objective is “unambiguously representing” the intended goal.

Alex’s model of Evan (A-E): Yup, basically.

A: OK. So in this scenario, though, the actual reward-generating process would in fact be foolable by an AI which replaces the window with an extremely convincing display which showed me a video which made me believe it got gold coins, even though it was actually touching a secret silver coin in the real room. The existence of that adversarial input isn’t a problem, because in this story, we aren’t trying to get the AI to directly optimize the reward-generating process or any of its Cartesian transforms or whatever.

A-E: Well, I guess? If you *assume* you get the gold-coin AI, you can satisfy the story with such an underdetermined and unhardened outer objective. But I expect in reality you need to supply more reward data to rule out e.g. silver coins, and possibly to disincentivize deception during training. See the RLHF + camera-duping incident.

So I think the answer is “technically no you don’t *have* to worry about adversarial inputs to the grading procedure on this definition, but in reality I think you should.”

A: I think we’re going to have a separate disagreement on that camera incident which isn’t related to this decomposition, so I’ll just move past that for the moment. If this is the perspective, I don’t disagree with it as much as “have the objective represent what you want as faithfully as possible, maybe even exactly, such that the outer objective is good to optimize for.”

I think that this decomposition is actually compatible with some shard theory stories, even. It feels like this outer alignment definition is actually pretty lax. It feels more like saying “I want to write down an objective which appears to me to ‘encode’ gold coin-grabbing, and then have that objective entrain a gold coin value in the agent.” And, for chisel = statue reasons, the levers for inner alignment would then have to come from inductive biases (speed / complexity / hyperparameters / whatever), and not the actual feedback signals (which are kinda fixed to match the “represent the gold coin objective”).

Daniel Ziegler’s working definitions

I recently spoke with Daniel Ziegler about one frame he uses for alignment, which he described as inspired by Christiano’s Low-stakes alignment, and relating to outer/inner alignment. Here’s my summary:

> I think about getting two main guarantees. First, that we can evaluate and grade every possible training situation which the AI can understand (this roughly maps onto “outer alignment”). Second, that the AI output an (at least) adequate / non-catastrophic decision in every possible deployment situation (this roughly maps onto “inner alignment”).

I don’t think we need robust grading in every possible *training* situation; it seems to me like early and mid-training will be far more important for chiseling values into the AI. I’m less worried about evaluating late-training situations where the AI is already superintelligent. I also don’t think we need robust adequacy. There probably has never ever existed a human which behaves adequately in every possible situation. Probably Gandhi goes on a killing spree in *some* situation.

I’m more concerned about on-trajectory properties—make the AI nice to begin with, make it want to keep being nice in the future, and I don’t worry about off-trajectory bad situations it won’t even want to enter. If the AI thought “I’m nice now but won’t be nice later”, wouldn’t the AI take action of its own accord to head off that event, which would be bad by its own values?

I worry that absolute robustness is an unnatural cognitive property, which is also not necessary, and that certain attempts to achieve it could even worsen alignment properties. As one concrete (but mostly theoretical) concern, adversarial training might make an initially nice AI, less nice / aligned:

> **A problem with adversarial training.** One heuristic I like to use is: "What would happen if I initialized a human-aligned model and then trained it with my training process?"
>
> So, let's consider such a model, which cares about people (i.e. reliably pulls itself into futures where the people around it are kept safe). Suppose we also have some great adversarial training technique, such that we have e.g. a generative model which produces situations where the AI would break out of the lab without permission from its overseers. Then we run this procedure, update the AI by applying gradients calculated from penalties applied to its actions in that adversarially-generated context, and... profit?
>
> But what actually happens with the aligned AI? Possibly something like:
>
> 1. The context makes the AI spuriously believe someone is dying outside the lab, and that if the AI asked for permission to leave, the person would die.
> 2. Therefore, the AI leaves without permission.
> 3. The update procedure penalizes these lines of computation, such that in similar situations in the future (i.e. the AI thinks someone nearby is dying)

the AI is less likely to take those actions (i.e. leaving to help the person).
4. We have made the aligned AI *less* aligned.

> I don't know if anyone's written about this. But on my understanding of the issue, there's one possible failure mode of viewing adversarial training as ruling out bad behaviors *themselves*. But (non-tabular) RL isn't like playing whack-a-mole on bad actions, RL's credit assignment changes the general values and cognition within the AI. And with every procedure we propose, the most important part is *what cognition will be grown from the cognitive updates accrued under the proposed procedure.*

EDIT: This is less "don't do adversarial training", and more "I have some intuitions there are subtle costs and difficulties to demanding extreme robustness from a system."

Outer alignment on physical reward is impossible

Consider the following definitions:

> **Outer alignment$_{\text{literal}}$:** Specify a real-world number-generating procedure *P* which only produces high numbers if “good things are happening”, in some reasonable sense.
>
> **Inner alignment with *P*:** Ensure the AI primarily cares about optimizing *P*’s output, in some reasonable sense.

Unsolvability of outer alignment$_{literal}$. Any outer objective *P* must be implemented within the real world. Suppose that *P* reliably produces huge numbers in worlds where the AI is doing what we want. But then the number produced by *P* can be further increased by just modifying the physically implemented output.

So, for any agent with a sufficiently rich action space (so that it can affect the world over time), any search for maximal *P*-outputs yields tampering (or something else, not related to what we want, which yields even greater outputs).[16]

Appendix B: RL reductionism

A bunch of alignment thinking seems quite airy, detached from step-by-step mechanistic thinking. I think there are substantial gains to thinking more precisely. I sometimes drop levels of abstraction to view NN training as a physical process which imperfectly shadows the nominal PyTorch code, which itself imperfectly shadows the mathematical learning algorithms (e.g. SGD under certain sampling assumptions on minibatches), which itself is imperfectly abstracted by rules like "loss as chisel", which itself is *sometimes* abstractable as "networks get trained to basically minimize loss / maximize reward on a certain distribution."

Consider what happens when you train a deep Q-learning network on Pac-Man. I'll start with reward-as-chisel, but then take a slightly more physical interpretation.

1. **Reward as chisel, detailed analysis.** When we initialize the Q-network and begin training, the reward function provides a sequence of cognitive updates to the physically instantiated network, as mediated by mini-batches empirical data distribution gathered under the policy defined by the relevant Q-values.
    1. IE the network explores and bumps into ghosts (negative reward) and into dots (positive reward). The network learns to predict different Q-values for actions which historically led to ghost-events, compared to those which e.g. led to dots. The network's numbers behave differently in the presence of different relevant observables, and so SGD is entraining some kind of contextual computations into the network.
    2. Each TD error is computed, and, through a corresponding gradient, updates the Q-network's computational structures so as to generalize its Q-value estimates *slightly* differently.
    3. Run out long enough and due to the exploration properties of the Pac-Man task, the network chains its predictions together and learns to predict high values for actions which do in fact allow the network to survive and eat dots.
    4. As a side note, the agent may indeed "achieve high cognitive-update-intensity (i.e. "reward")" for game screens which are mechanically and perceptually similar relative to the computations run inside the network (e.g. there are still walls and mazes and arrangements of ghosts, if that's how the network in fact makes decisions).
2. **Physical reward instantiation model**: But *really,* "reward function" is *itself* an abstraction. There is no reward function, in reality. There is simply a sequence of state-modifications which, for convenience, we often abstract as "temporal difference updates on the Q-value predictor, taken over a mini-batch drawn from the action replay buffer."
    1. These modifications are spurred by a sequence of *sampled "reward events"*, which are really just the physical outputs of the part of the computer we abstract as the "reward function calculator", which then gets fed into the gradients. But the network never sees the reward, or the reward function. We could overwrite and restore the reward function's implementation, between each update step, and it wouldn't matter to the trained network.
    2. Similarly, ask not whether the reward function is "stationary", ask what cognition the sequence of reward-events entrains into the network.
    3. In a strict causal sense, the physical reward function only matters insofar as it updates the physically implemented network, and the updates only matter insofar as they affect generalization behavior in ways we care about (e.g. does the network output good alignment research). The reward function has no metaphysical or special status. It's just another part of the physical apparatus.

[1] ^

In comments on an earlier draft of this post, Paul clarified that the reward doesn't have to *exactly* capture the [expected] utility of deploying a system or of taking an action, but just e.g. correlate on reachable states such that the agent can't predict deviations between reward and human-[expected] utility.

[2] ^

Agreed.

3. ^

   I’m not claiming this is Paul’s favorite alignment plan, I can’t speak for him. However, I do perceive most alignment plans to contain many/all of: 1) robust grading, 2) “the chisel must look like the statue”, and 3) aligning the AI to a grading procedure.

4. ^

   I am by no means the first to consider whether the outer/inner frame is inappropriate for many situations. Evan Hubinger wrote:

   "It’s worth pointing out how phrasing inner and outer alignment in terms of training stories makes clear what I think was our biggest mistake in formulating that terminology, which is that inner/outer alignment presumes that the right way to build an aligned model is to find an aligned loss function and then have a training goal of finding a model that optimizes for that loss function.”

5. ^

   In this essay, I focus on the case where the outer objective’s domain is the space of possible plans. However, similar critiques hold for grading procedures which grade world-states or universe-histories.

6. ^

   The truth is that I don't yet know what goes on in more complicated and sophisticated shard dynamics. I doubt, though, that grader-optimization and value-optimization present *the same* set of risk profiles (via e.g. Goodhart and nearest unblocked strategy), which *coincidentally* derive from different initial premises via different cognitive dynamics. "It’s improbable that you used mistaken reasoning, yet made no mistakes."

7. ^

   Outer/inner fails to describe/explain how GPT-3 works, or to prescribe how we would want it to work (“should GPT-3 really minimize predictive loss over time?” seems like a Wrong Question). Quintin wrote in private communication:

   “GPT-3’s outer ‘objective’ is to minimize predictive error, and that’s the only thing it was ever trained on, but GPT-3 itself doesn’t ‘want’ to minimize its predictive error. E.g., it’s easy to prompt GPT-3 to act contrary to its outer objective as part of some active learning setup where GPT-3 selects hard examples for future training. Such a scenario leads to GPT-3 taking actions that systematically fail to minimize predictive error, and is thus not inner aligned to that objective.”

8. ^

   This point is somewhat confounded because humans “backchain” reward prediction errors, such that a rewarding activity bleeds rewardingness onto correlated activities (in the literature, see the related claim: “primary reinforcers create secondary reinforcers”). For example, in late 2020, I played *Untitled Goose Game* with my girlfriend. My affection for my girlfriend spilled over onto a newfound affection for geese, and now (I infer that) it's rewarding for me to even think about geese, even though I started off ambivalent towards them. So, I infer that there's a big strong correlation between "things you value and choose to pursue" and "mental events you have learned to find rewarding."

9. ^

   I don't actually think in terms of "inner alignment failures" anymore, but I'm writing this way for communication purposes.

10. ^

   The original abstract begins: "A 48-year-old woman with a stimulating electrode implanted in the right thalamic nucleus ventralis posterolateralis developed compulsive self-stimulation associated with erotic sensations and changes in autonomic and neurologic function."

11. ^

   I think the shard frame is way better than the utility function frame because of reasons like "I can tell detailed stories for how an agent ends up putting trash away or producing diamonds in the shard frame, and I can't do that at all in the utility frame." That said, I'm still only moderate-strength claiming "the shard frame is better for specifying what kind of AI cognition is safe" because I haven't yet written out positive mechanistic stories which spitball what kinds of shard-compositions lead to safe outcomes. I am, on the other hand, *quite confident* that outer/inner is inappropriate.

12. ^

   The coherence theorems can pin down "EU maximization" all you please, but they don't pin down the domain of the utility functions. They don't dictate what you have to be coherent over, when trading off lotteries. I commented:

   > 80% credence: It's very hard to train an inner agent which reflectively equilibrates to an EU maximizer only over commonly-postulated motivating quantities (like `# of diamonds` or `# of happy people` or `reward-signal`) and not quantities like (`# of times I have to look at a cube in a blue room` or `-1 * subjective micromorts accrued`).
   >
   > Intuitions:
   >
   > - I expect contextually activated heuristics to be the default, and that agents will learn lots of such contextual values which don't cash out to being strictly about diamonds or people, even if the overall agent is mostly motivated in terms of diamonds or people.
   >
   > - Agents might also "terminalize" instrumental subgoals by caching computations (e.g. cache the heuristic that dying is bad, without recalculating from first principles for every plan in which you might die).
   >
   > Therefore, I expect this value-spread to be convergently hard to avoid.

   And so it goes for human values. If human values tend to equilibrate to utility functions which factorize into factors like `-1 * subjective micromorts` or `# of` `times I tell a joke around my friends`, but you think that the former is “just instrumental” and the latter is “too contextual”, you’re working in the wrong specification language.

Another difficulty to “just produce diamonds” is it assumes a singular shard (diamond-production), which seems anti-natural. Just look at people and their multitudes of shards! I think we should not go against suspected grains of cognition formation.

13. ^

    Proposition E.30 of Optimal Policies Tend to Seek Power.

14. ^

    RL practitioners *do in fact* tend to reward agents for doing good things and penalize them for doing bad things. The prevalence of this practice *is* some evidence for “rewarding based on goodness is useful for chiseling policies which do what you want.” But this evidence seems tamped down somewhat because “reward optimization” was a prevalent idea in RL theory well before deep reinforcement learning really took off. Just look at control theory back in the 1950’s, where control systems were supposed to optimize a performance metric over time (reward/cost). This led to Bellman’s optimality equations and MDP theory, with all of its focus on reward as the optimization target. Which probably led to modern-day deep RL retaining its focus of rewarding good outcomes & penalizing bad outcomes.

15. ^

    The loss function can indeed “hit back” against bad behavior, in the form of providing cognitive updates which “downweight” the computations which produced the negative-loss event. However, this “hitting back” only applies while the AI’s values are still malleable to the loss function. If the AI crystallizes unaligned values (like seeking power and winning games) and gets smart, it can probably gradient hack and avoid future updates which would break its current values.

    However, reality will always “hit back” against bad capabilities. A successful AGI will continually become more capable, even well after value crystallization.

16. ^

    This argument works even if *P* originally penalizes tampering actions. Suppose the agent is grading itself for the average output of the procedure over time (or sum-time-discounted with $\gamma \approx 1$, or the score at some late future time step, or whatever else; argument should still go through). Then penalizing tampering actions will decrease that average. But since the penalties only apply for a relatively small number of early time steps, the penalties will get drowned out by the benefits of modifying the *P*-procedure.