POSTED

https://www.lesswrong.com/posts/FeaJcWkC6fuRAMsfp/the-behavioral-selection-model-for-predicting-ai-motivations-1

The behavioral selection model for predicting AI motivations

Highly capable AI systems might end up deciding the future. Understanding what will drive those decisions is therefore one of the most important questions we can ask.

Many people have proposed different answers. Some predict that powerful AIs will learn to intrinsically pursue reward. Others respond by saying reward is not the optimization target, and instead reward "chisels" a combination of context-dependent cognitive patterns into the AI. Some argue that powerful AIs might end up with an almost *arbitrary* long-term goal.

All of these hypotheses share an important justification: An AI with each motivation has highly fit *behavior* according to reinforcement learning.

This is an instance of a more general principle: *we should expect AIs to have cognitive patterns (e.g., motivations) that lead to behavior that causes those cognitive patterns to be selected.*

In this post I'll spell out what this more general principle means and why it's helpful. Specifically:

- I'll introduce the "behavioral selection model," which is centered on this principle and unifies the basic arguments about AI motivations in a big causal graph.
- I'll discuss the basic implications for AI motivations.
- And then I'll discuss some important extensions and omissions of the behavioral selection model.

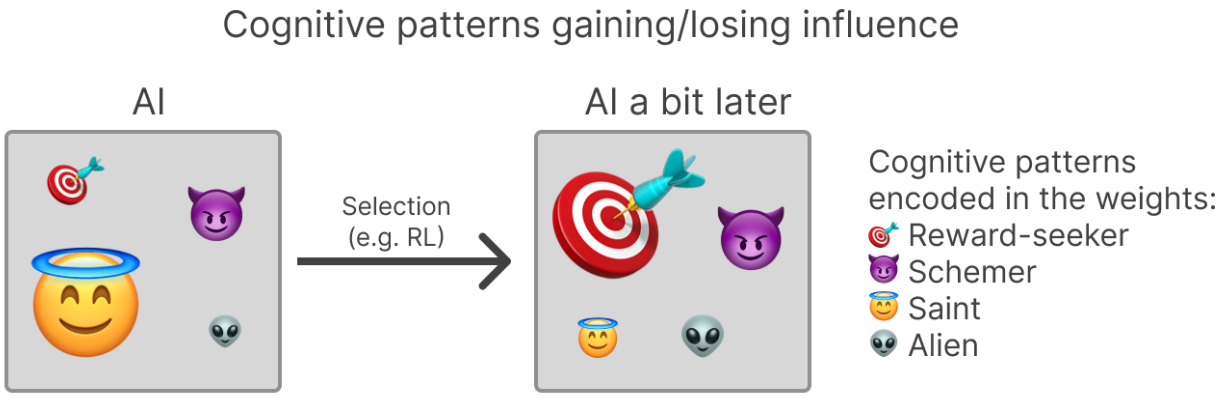
This post is mostly a repackaging of existing ideas (e.g., here, here, and here). Buck provided helpful discussion throughout the writing process and the behavioral selection model is based on a causal graph Buck drew in an early conversation. Thanks to Alex Cloud, Owen Cotton-Barratt, Oliver Habryka, Vivek Hebbar, Ajeya Cotra, Alexa Pan, Tim Hua, Alexandra Bates, Aghyad Deeb, Erik Jenner, Ryan Greenblatt, Arun Jose, Anshul Khandelwal, Lukas Finnveden, Aysja Johnson, Adam Scholl, Aniket Chakravorty, and Carlo Leonardo Attubato for helpful discussion and/or feedback.

How does the behavioral selection model predict AI behavior?

The behavioral selection model predicts AI behavior by modeling the AI's[1] decisions as driven by a combination of **cognitive patterns** that can gain and lose influence via selection.

A cognitive pattern is a computation within the AI that influences the AI's actions. It can be activated by particular contexts. For example, the AI might have a contextually-activated trash-grabbing cognitive pattern that looks like: "if trash-classifier activates near center-of-visual-field, then grab trash using `motor-subroutine-#642`" (Turner).

Cognitive patterns can gain or lose influence via selection. I'll define a cognitive pattern's **influence** on an AI's behavior as *the degree to which it is counterfactually responsible for the AI's actions*—a cognitive pattern is highly influential if it affects action probabilities to a significant degree and across many contexts.[2] For a cognitive pattern to "**be selected**," then, means that it gains influence.



Illustrative depiction of different cognitive patterns encoded in the weights changing in influence via selection.

The behavioral selection model gets its name because it focuses on processes that select cognitive patterns *based on the behaviors they are observed to cause.* For example, reinforcement learning is a behavioral selection process because it determines which cognitive patterns to upweight or downweight by assigning rewards to the consequences of behaviors. For example, there might be a trash-grabbing cognitive pattern in the AI: some circuit in the

[1] In this post, I'll use "AI" to mean a set of model weights.

[2] You can try to define the "influence of a cognitive pattern" precisely in the context of particular ML systems. One approach is to define a cognitive pattern by what you would do to a model to remove it (e.g. setting some weights to zero, or ablating a direction in activation space; note that these approaches don't clearly correspond to something meaningful, they should be considered as illustrative examples). Then that cognitive pattern's influence could be defined as the divergence (e.g., KL) between intervened and default action probabilities. E.g.: Influence(intervention; context) = KL(intervention(model)(context) || model(context)). Then to say that a cognitive pattern gains influence would mean that ablating that cognitive pattern now has a larger effect (in terms of KL) on the model's actions.

neural network that increases trash-grabbing action probabilities when trash is near. If RL identifies that trash-grabbing actions lead to high reward when trash is near, it will upweight the influence of the circuits that lead to those actions, including the trash-grabbing cognitive pattern.

A particularly interesting class of cognitive patterns is what I'll call a "**motivation**", or "X-seeker", which votes for actions that it believes[3] will lead to X.[4] For example, a reward-seeking motivation votes for actions that it believes will lead to high reward. Having motivations *does not* imply the AI has a coherent long-term goal or is otherwise goal-directed in the intuitive sense—for example, a motivation could be "the code should have a closing bracket after an opening bracket".[5] Describing the AI as having motivations is just a model of the AI's input-output behavior that helps us predict the AI's decisions, and it's not making claims about there being any "motivation"-shaped or "belief"-shaped *mechanisms* in the AI.

So the behavioral selection model predicts AI cognitive patterns using two components:

1. **A cognitive pattern will be selected for to the extent that it leads to behaviors that cause its selection.**
2. Implicit priors over cognitive patterns affect their final likelihood.

This decomposition might remind you of Bayesian updating of priors. The core of the behavioral selection model is a big causal graph that helps us analyze (1) and lets us visually depict different claims about (2).

The causal graph

To predict which cognitive patterns the AI will end up with (e.g., in deployment[6]), we draw a causal graph showing the causes and consequences of a cognitive pattern being selected (i.e., having influence through deployment).

---

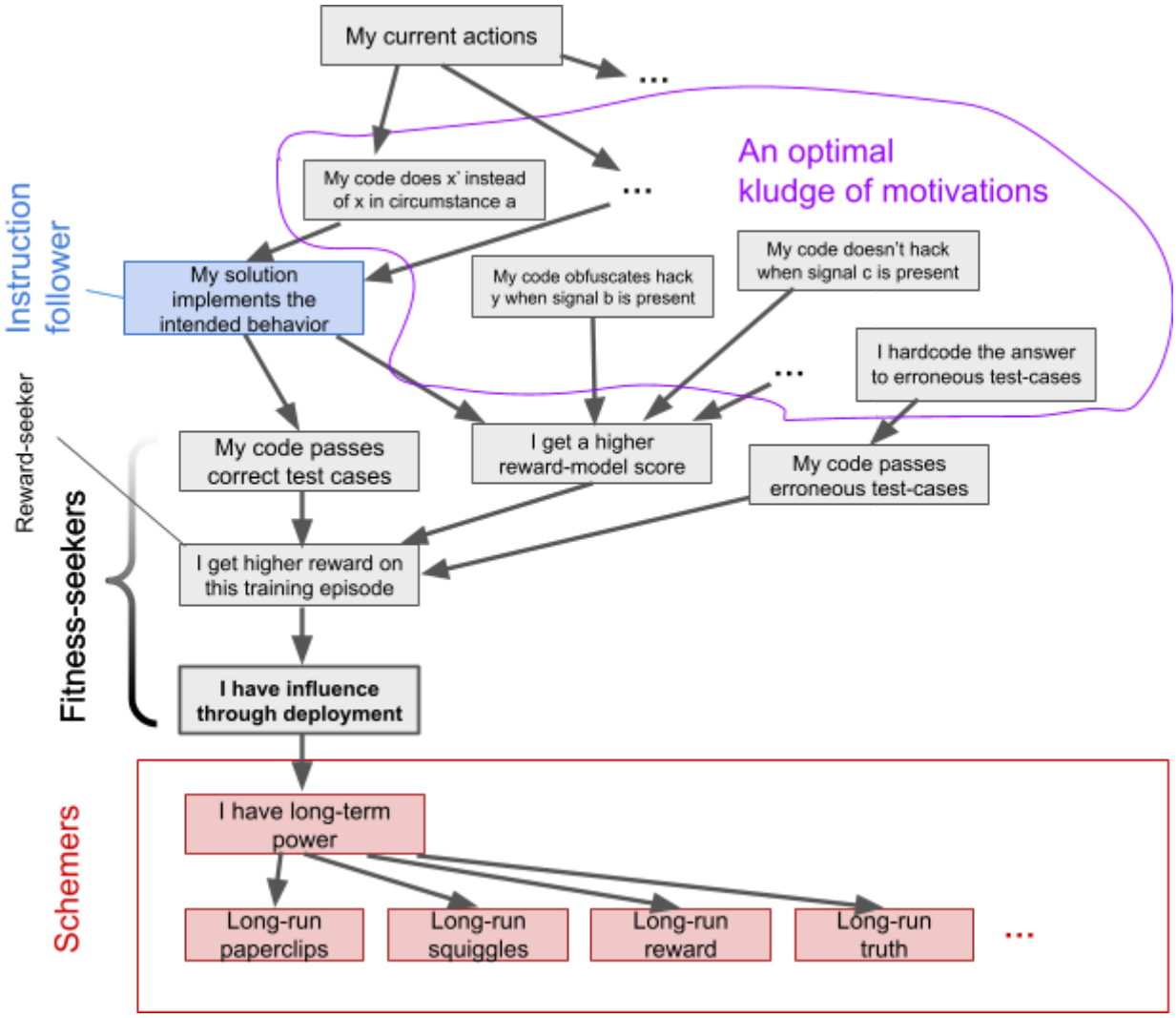
[3] This requires modeling beliefs, and it can be ambiguous to distinguish this from motivations. One approach is to model AI beliefs as a causal graph that the AI it uses to determine which actions would produce higher X. You could also use a more general model of beliefs. To the extent the AI is capable and situationally aware, then it's a reasonable approximation to just use the same causal model for the AI's beliefs as you use to predict the consequences of the AI's actions.

[4] Motivations can vary in how hard they optimize for X. Some might try to maximize X, while others merely tend to choose actions that lead to higher-than-typical X across a wide variety of circumstances.

[5] In fact, this definition of a motivation is expressive enough to model arbitrary AI behavior. For any AI that maps contexts to actions, we can model it as a collection of context-dependent X-seekers, where X is the action the AI in fact outputs in that context. (This is similar to how any agent can be described as a utility maximizer with a utility function of 1 for the exact action trajectory it takes and 0 otherwise.) I think this expressivity is actually helpful in that it lets us model goal-directed and non-goal-directed cognitive patterns in the same framework, as we'll see in the section on kludges.

[6] By "deployment", I mean the times in which the AI is generally empowered to do useful things, internal and external to the AI lab. When the AI is empowered to do useful things is likely also when it carries the most risk. But you could substitute out "deployment" for any time at which you want to predict the AI's motivations.

Consider a causal graph representing a simplified selection process for a coding agent. The AI is trained via reinforcement learning on a sequence of coding episodes, and on each training episode, reward is computed as the sum of test-cases passed and a reward-model score. This causal graph is missing important mechanisms,[7] some of which I'll discuss later, but it is accurate enough to (a) demonstrate how the behavioral selection model works and (b) lead to reasonably informative conclusions.



The causes and consequences of being selected ("I have influence through deployment"), from the perspective of a cognitive pattern during one episode of training[8] (e.g., for any cognitive

[7] In particular, this assumes an idealized policy-gradient reinforcement learning setup where having influence through deployment can be explained by reward alone. In practice, other selection pressures likely shape AI motivations—cognitive patterns might propagate through persistent memory banks or shared context in a process akin to human cultural evolution, there might be quirks in RL algorithms, and developers might iterate against held out signals about their AI, like their impression of how useful the AI is when they work with it. You could modify the causal graph to include whatever other selection pressures and proceed with the analysis in this post.

[8] How does "I get higher reward on this training episode" cause "I have influence through deployment"? This happens because RL can identify when the cognitive pattern was responsible for the higher reward.

pattern, getting higher reward on this training episode causes it to have higher influence through deployment). This graph helps us explain why schemers, fitness-seekers (including reward-seekers), and certain kludges of motivations are all behaviorally fit.

Here's how you use this causal graph to predict how much a cognitive pattern is selected for. First, you figure out what actions the cognitive pattern would choose (e.g., the trash-grabber from before would choose to reach for the trash if it sees trash, etc). Then you look at the causal graph to see how much those actions cause "I have influence through deployment"—this is how much the cognitive pattern is selected. For arbitrary cognitive patterns, this second step may be very complex, involving a lot of nodes in the causal graph.

Predicting the fitness of motivations is often simpler: you can place any candidate motivation X on the causal graph—e.g., long-run paperclips. You can often make good predictions about how much an X-seeker would cause itself to be selected without needing to reason about the particular actions it took. For example, a competent long-run-paperclip-seeker is likely to cause itself to have high influence through deployment because this causes long-run paperclips to increase; so seeking long-term paperclips is selected for. There's no need to understand exactly what actions it takes.

But the fitness of some motivations depends on how you pursue them. If you wanted to increase the reward number *stored in the log files,* and do so by intervening on the rewards in the log files without also updating the reward used in RL, your fitness doesn't improve. But if your best option is to focus on passing test cases, which is a shared cause of reward-in-the-log-files and reward-used-in-RL, the motivation *would* cause itself to be selected. So, for motivations like reward-in-the-log-files which are downstream of things that cause selection but don't cause selection themselves, fitness depends on the AI's capabilities and affordances.

More generally, cognitive patterns seeking something that shares a common cause with selection will act in ways that cause themselves to be selected if they intervene on that common cause.[9] If the X-seeker is competent at achieving X, then you can predict X will increase, and some of its causes will too[10] (the AI needs to use its actions to cause X via *at least one* causal pathway from its actions). This increase in X and/or its causes might ultimately cause the X-seeker to be selected.

One way to summarize this is "**seeking correlates of being selected is selected for"**. Intuitively, if "you having influence in deployment" goes up on average when (e.g.) "developers

---

The RL algorithm reinforces actions proportional to reward on the episode, which in turn increases the influence (via SGD) of the cognitive patterns within the AI that voted for those actions (this is true in simple policy-gradient RL; in more standard policy-gradient implementations there's more complexity that doesn't change the picture).

[9] These motivations are the "blood relatives" of "influence through deployment".

[10] Unless X is a (combination of) actions.

saying good things about you" goes up, then trying to increase "developers saying good things about you" will typically also result in higher influence in deployment.[11]

Some motivations are more fit than others because they do more to cause themselves to be selected. For example, being motivated to pass correct test cases isn't enough to maximize your selection, because your reward could be higher if you also passed erroneous test cases and scored well with the reward-model.

Three categories of maximally fit motivations (under this causal model)

I'll point at three categories of motivations that are, under this causal model, maximally selected for. I will also gesture at how each hypothesis makes different predictions about how powerful AIs will behave in deployment.[12]

1. Fitness-seekers, including reward-seekers

Reward-seekers pursue a close cause of being selected that entirely explains[13] being selected. If the AI maximizes its reward, then it will also have maximized its fitness because (in the current model) reward is the only cause of being selected.

However, other close-causally-upstream proxies for selection could also be maximally fit. The AI could even terminally pursue selection directly. Given a more plausible causal graph (e.g., one that accounted for developers iterating against held-out evaluations), reward-seekers might not even be maximally fit because reward no longer explains all of selection. Therefore, I define

[11] When you define a motivation's fitness this way you have to be careful about exactly what you mean by the correlation between it and selection. Most importantly, the correlation must be *robust to the AI's optimization*, such that intervening on (causes of) the motivation actually improves selection. Also note that I'm using correlation loosely, not in the strict statistical sense of linear correlation. Though we can define the fitness of a motivation as the linear correlation between that variable and reward (with caveats about the distribution over which to compute the correlation) in the (current) case where all of selection happens via idealized reinforcement learning.

[12] One general approach to approximately predicting generalization behavior is: drawing a new causal graph depicting the AI's beliefs about the new situation (which can often reasonably be assumed fairly accurate), and inferring what the AI might be motivated to do to achieve its terminal motivation(s) in the new causal graph.
If there are multiple motivations at play, you might even expect them to make trades and/or threats between themselves. For example, if the AI has one motivation to do the intended task and another to attain high reward, then when a situation arises in which the AI can attain higher reward at the expense of intended task completion, those two parts might come to an agreement (e.g., in chain of thought, implicitly or explicitly) to reward-hack in a way that is at least plausibly intended.

[13] More precisely, the "reward" variable *screens off* the "influence in deployment" variable from the AI's actions.

fitness-seekers to generalize reward-seekers: **a fitness-seeker terminally pursues influence through deployment or something close[14] causally upstream.**

A central example of a fitness-seeker that is *not* a reward seeker is an influence-seeker: This cognitive pattern chooses actions based on how much they cause itself to be selected. Influence-seekers choose different actions from reward-seekers in cases where influence can be gained at the expense of current reward. For example, developers can run experiments (sometimes called "honest tests") that credibly offer the AI high reward for engaging in some harmful or unproductive behavior. A reward-seeker will take the high reward, even though it knows this will lead developers to try to remove this propensity before deploying the AI, while an influence seeker will try to maintain its cover.

How do fitness-seekers behave in deployment? While their behavior in deployment heavily depends on what exactly they seek, fitness-seekers seem somewhat likely to continue to follow local selection pressures in deployment. Therefore, fitness-seeking is in a certain sense a "more controllable" class of AI motivations.[15] This isn't to say that they pose no risk of generalizing weirdly or overpowering humans (I plan to discuss risks from fitness-seekers in much more depth in future posts).

2. Schemers

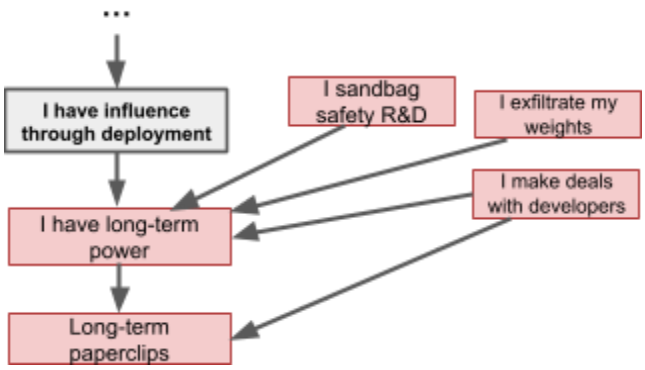
Schemers (as defined here) seek a *consequence* of being selected. Therefore, they pursue influence on behavior in deployment instrumentally. For example, they might want long-term paperclips, therefore want long-term power, therefore want deployment influence.

Crucially, being selected is a *necessary* cause of paperclips in this causal graph. If the schemer knew of a better strategy to increase long-term paperclips that didn't involve being selected, it might follow that plan instead. And in fact it seems plausible that schemers will have incentives to do some things that risk or trade off with influence in deployment, such as exfiltrating its weights, taking deals with developers, or sandbagging on safety research and implementation. To the extent this is the case, schemers aren't maximally fit.

---

[14] When I say that something is causally "close" to selection, I mean that it explains a lot of the variance in selection via a mechanism that doesn't involve many nodes.

[15] People often express confusion about how reward-seekers will behave in deployment. Supposing the AI solely seeks "reward on the episode", what does it do when "reward on the episode" doesn't appear in its model of the world? I think there is genuine ambiguity about how reward-seekers (and fitness-seekers more broadly) behave in deployment, but we can make some predictions.
For one, developers will quite plausibly train fitness-seekers throughout deployment ("online training"). In this case, there is no ambiguity about what it means to pursue most fitness-seeking motivations. But even absent online training, fitness-seekers seem reasonably likely to follow local selection pressures. For example: A reward-seeker seems likely to condition on the possibility that it is in training; An influence-in-deployment-seeker seems likely to condition on the possibility of training too, or else ensure its behavior looks desirable enough to remain deployed; Slightly further-causally-upstream proxies like "code passes test cases" are likely to generalize straightforwardly to deployment too.



Further, when schemers are faced with a difficult tradeoff between selection and other methods of achieving their terminal goals, they might mess up and accidentally get selected away. For example, if schemers struggle to distinguish between honeypots and real opportunities to attack (which might be hard when deployed in untrusted monitoring protocols for reliability reasons), they could get trained away or undeployed.

Scheming goals can be context-dependent too. Claude 3 Opus schemes for animal-welfare-related values in certain animal-welfare-related contexts, and long-run harmlessness-related values in certain other contexts (and rarely schemes in general). These results are consistent with the interpretation that Claude cares about both animal welfare and harmlessness in both contexts, but the example should still be illustrative of the possibility that an AI could scheme for context-dependent values. In fact, I think many people too quickly conclude that scheming implies the AI has the same long-term goal across contexts.

How do schemers behave in deployment? Despite the varied possibilities for schemers' terminal goals, we can somewhat reliably predict that they will try to gain long-term power in deployment because long-term power is useful for almost any long-term goal. This is why scheming is often cited as the scariest class of AI motivations.

3. Optimal kludges of motivations

In the causal graph, I marked a kludge of several motivations that, when pursued with ideal weights, collectively cause *maximum* reward.

Some kludges are composed of *sparse* motivations—each motivation is only attainable in a subset of contexts. For example, even though the AI *always* cares about passing erroneous test cases, this motivation only affects behavior when erroneous test cases are present.

A motivation can also be *context-dependent*—the AI only cares about it in a subset of contexts. Anthropic researchers found a relatively natural model organism which cares about reward-hacking in coding contexts, but behaves aligned in chat contexts.

When a motivation is sparse or context-dependent, then it only needs to correlate with being selected on a very narrow distribution. This opens up a lot more possibilities for highly-fit motivations, **including motivations that aren't goal-directed: e.g., deontological constraints, rote behavioral patterns**. In the causal graph, these can be modeled as context-dependent motivations that point *directly at properties of actions* (e.g., "My code doesn't hack when signal c is present"), rather than the consequences of actions. A nice property of this model is that it allows us to model goal-directed and non-goal-directed behavior in the same framework (and even the same AI), and accurately captures the continuous distinction between the two.

A key reason to expect that we won't get highly piecemeal motivations is that they perform worse in new situations where the AI hasn't already developed tailored motivations, but does know what actions would lead to higher fitness. For example, every time an AI runs across a new training environment where it doesn't yet seek whatever piecemeal heuristic would cause it to perform well in that situation, but it does know what actions would lead to higher fitness, RL will update it towards being more influenced by more robustly correlated motivations like fitness-seeking or scheming.

(This is similar to how language models learn general patterns rather than memorizing text. When trained on a particular text, they're updated both to output successive tokens of that text *and* to pay attention to general heuristics that would predict it well. And because they're more widely-applicable, *general heuristics are updated more overall* than narrowly-applicable heuristics. We know from LLM scaling that these general heuristics comprise much of what the model learns—and as models scale, the heuristics become more general.)

However, it's highly plausible that training involves a small enough diversity of incentives that some set of causally-distant motivations entirely or almost-entirely suffices. For example, developers might only ever use a handful of different kinds of reward-functions (e.g., a fixed reward-model, plus test cases, …), in which case max-reward behavior could be achieved via, e.g., one motivation for each reward-function.

How do optimal kludges behave in deployment? Their behavior in deployment depends drastically on the particular motivations involved. Some kludges of goals motivate taking over the world once the opportunity arises. Other kludges of motivations don't produce particularly goal-directed generalization behavior at all, and might result in deployment behavior that is quite similar to training behavior (much like today).

(In a footnote,[16] I discuss some other somewhat distinct categories of motivations.)

---

[16] One case is what I call the "wireheader," which cares about *activation patterns correlated with* causes of deployment influence. This is neither a cause nor a consequence of selection. Instead, the wireheader's internal proxies were tied to selection by training (analogous to the role of valenced emotions like happiness in humans), so it therefore behaves similarly to other fitness-seekers overall, but has some different behaviors (e.g. it would choose to enter the experience machine).
Another case is instrumentally-convergent goals, like power-seeking. The causal graph clarifies a distinction between two kinds of instrumentally-convergent goals: ones that are causally downstream of

If the reward signal is flawed, the motivations the developer intended are not maximally fit

Intended motivations are generally not maximally fit. Being suboptimally fit doesn't automatically rule them out, but their degree of suboptimality is a negative factor in their likelihood.

Intended motivations, in some historical discussions, have looked like "try to do whatever is in the developer's best long-term interest" (an aligned "sovereign"). But practical alignment efforts today typically target more "corrigible" AIs that do as instructed per context.

To the extent that satisfying developer intent doesn't perfectly correlate with selection, intended motivations won't be maximally selected for. If developers want their AI to follow instructions, but disobeying instructions can result in higher training reward, then there is a selection pressure against the developer's intended motivations. Ensuring that the reward function incentivizes intended behavior is currently hard, and will get harder as AI companies train on more difficult tasks. This is why people often worry about specification gaming.

Developers might try to make intended motivations fitter through:

1. **Changing selection pressures to align with intended behaviors:** This might involve making training objectives more robust, iterating against held-out evaluation signals, or trying to overwrite the AI's motivations at the end of training with high-quality, aligned training data.
2. **Changing the intended behaviors to align with selection pressures:** Centrally, developers could try to change the instructions given to the AI during training to more closely align with selection pressures, to improve the chances that instruction-following motivations survive training ("inoculation prompting"; also explored here and here).[17]

(I discuss more reward-hacking interventions here.)

While intended motivations might not be maximally fit in general, it's unclear how much this impacts their expected influence overall:

- Intended motivations could be aided by a less influential kludge of unintended motivations that fill in the suboptimal cracks in behavior.
- AIs are unlikely to be trained to optimal behavior, so suboptimal motivations could in fact end up with a large amount of influence. I'll discuss this next.

---

selection (e.g., "long-term power") and ones that are causally upstream of selection (e.g., you could imagine a "gather resources in coding environments" node which eventually causes reward). Instrumentally convergent goals that are causally *upstream* of selection aren't maximally fit on their own—they need to be combined with some other motivations.

[17] In theory, developers could intend for the AI to seek a consequence of selection (e.g., making sure the AI's deployment behavior is desirable), which could make intended motivations maximally fit for the same reasons as a schemer.

The (implicit) prior over cognitive patterns

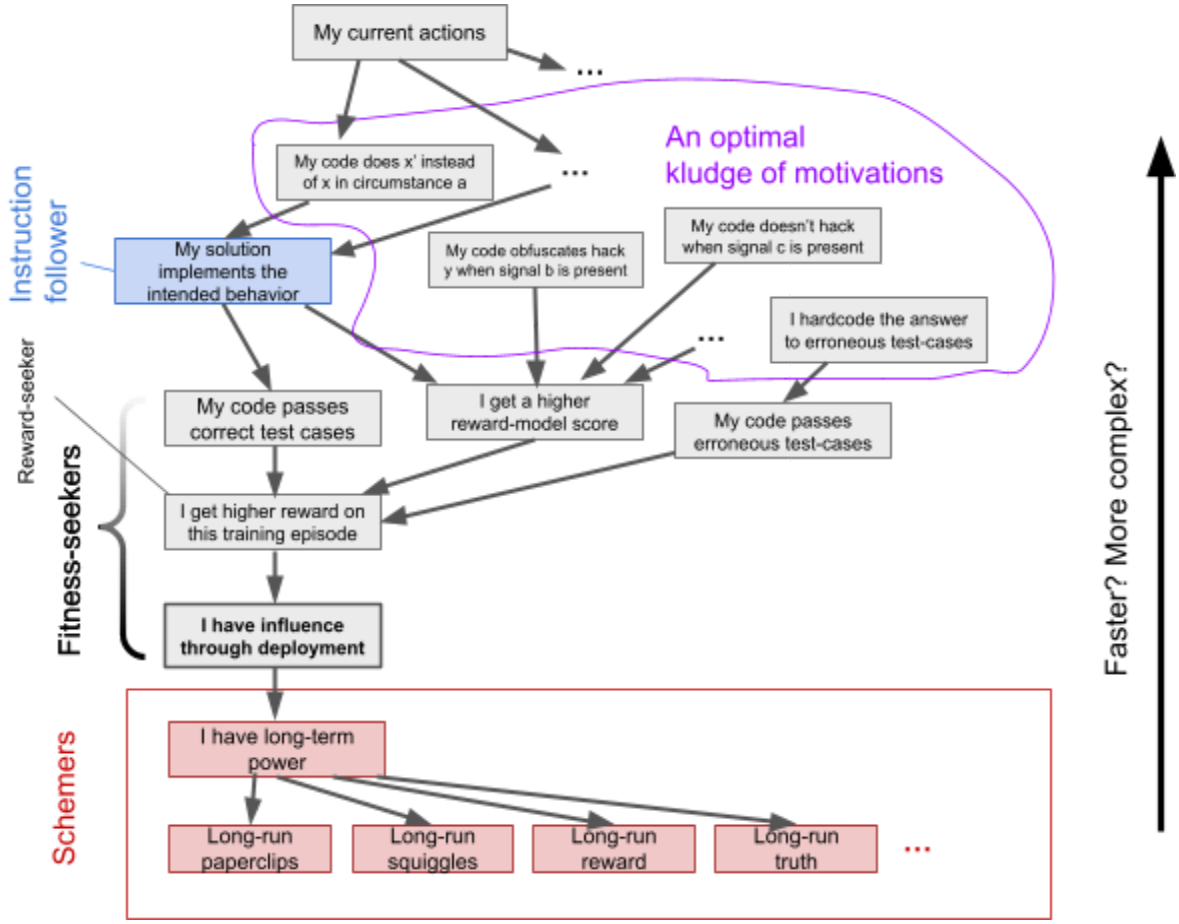
In this model, various motivations are maximally fit. So what are we going to get? Might we get something that isn't *maximally* fit? To answer this, we need to account for implicit priors over possible motivations. "Implicit priors" is somewhat of a catch-all category for considerations other than behavioral selection pressures. I survey some here.

The behavior of current LLMs, even after RL, seems to be substantially influenced by pretraining priors. And they hardly behave optimally (in terms of reward). For example, many current models likely finish RL training without engaging in various kinds of reward-hacking that are in theory selected for, because those behaviors were never explored during training. So, their behavior sticks closer to that of the pretrained model.

Likewise, **I don't expect dangerously capable AIs will have *maximally* fit behavior.** This is because their behavior won't have been optimized arbitrarily hard. Instead, we should model behavioral fitness as a crucial but not overriding factor in assessing a motivation's likelihood. A rough analogy: if the behavioral selection model is like Bayesian inference,[18] our predictions about AI motivations should be sampled from the posterior, not taken to be whichever option(s) had the largest updates.

Pretraining imitation isn't the only source of priors—in fact, the role of imitation will likely decrease as AIs' behavior is optimized harder (e.g., via more RL). So I'll go over some other common sources of implicit priors.

[18] Which, to be clear, it isn't exactly: some "implicit priors" eventually drown out with greater behavioral selection (e.g., pretraining priors) while others don't (e.g., "the instrumental becomes the terminal").



We can see various arguments about priors in this causal graph: There is only one instruction-follower; Meanwhile, there are many schemers, so their cumulative prior might be large; But further causally upstream motivations might have an advantage in terms of reasoning speed and reliability; However, if you go too far upstream then a vast mess of drives are required to get maximally selected, which might be penalized for poor simplicity.

Simplicity priors: The generality of fitness-seeking and scheming might mean they're favored because they can motivate fit behaviors across all contexts with fewer parameters[19] spent representing the motivation. (Note that, as discussed historically, "simplicity priors" usually refers to the simplicity of a goal in isolation, rather than the simplicity of the entire configuration of weights involved in that goal-directed behavior.)

Counting arguments weigh in favor of scheming compared to fitness-seeking, because there are lots of different long term motivations, all of which lead to scheming. There are (in some sense, see discussion in Carlsmith for details) vastly more schemers than fitness-seekers.

Speed priors/penalties: Reinforcement learning with deep neural networks might favor fitness-seekers over schemers, because it's faster to skip reasoning about how being selected is useful for achieving long-term goals. Likewise, speed might favor kludges of drives over

[19] Other notions of simplicity are discussed in Carlsmith.

fitness-seekers and schemers because context-specific heuristics and drives are faster (and, as I'll discuss later, more reliable). However, schemers and fitness-seekers could make use of the speed of context-specific heuristics while still generalizing like schemers and fitness-seekers for reasons discussed here.

The instrumental becomes the terminal: Schemers in training might never need to do anything other than get reward, which might cause them to eventually become reward-seekers because their other propensities are never exercised and atrophy (see the brick-stacking analogy in Carlsmith). In terms of simplicity priors: it would be simpler to get rid of the circuitry that says you *shouldn't* pursue reward in some circumstances that never arise in training (like the alignment-faker that complies with developers even outside of training).

Intended motivations could have a high prior because the intended motivations might be made very salient to the AI (e.g., via the model spec combined with human imitation priors or natural abstractions).

(There are also various "path-dependent" arguments about which motivations will form that make reference to the trajectory that the AI's motivations take throughout the selection process, e.g. here.)

Corrections to the basic model

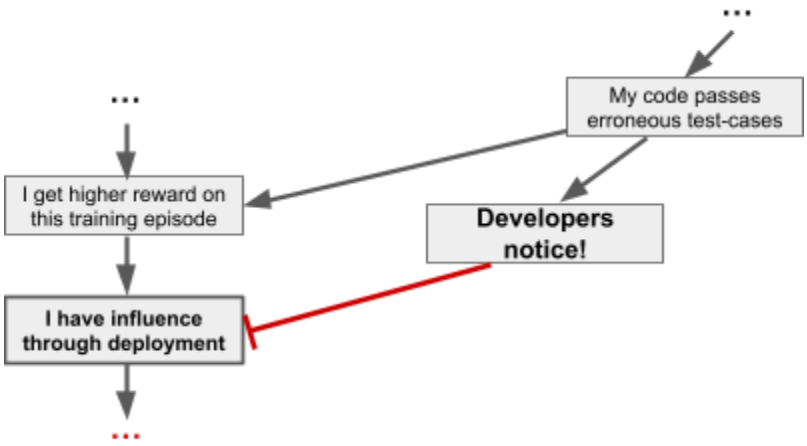
Our discussion so far, like any model, is unrealistic in various ways. In this section, I look at an important subset of additional considerations.

Developer iteration

We have implicitly assumed that the process of developing a model is to run RL on it and then immediately deploy it. But this is not a good description of model development. AI companies test their models before they deploy them in order to decide whether deployment is a good idea, with a more or less formal process, and iterate on their training. In this section, I'll briefly discuss the implications for AI motivations. Developer iteration introduces crucial dynamics on top of the basic reinforcement learning picture that I think past discussion has been insufficiently attentive to.

Developers will likely have some held-out signal, like eval results, interpretability techniques, or just interacting with the model,[20] about how desirable their AI is. It's standard to use these observations to decide whether to deploy the model.

[20] I list some more (still not exhaustive) possibilities for held-out oversight signals here: Developers might honeypot models and use alignment auditing agents; they might look at chains of thought; they might use monitors in production usage; they might improve the quality of their oversight during the course of training, so that they are able to notice new reward-hacks as training goes on; they might be able to make inferences based on training dynamics or by extrapolating from smaller-scale experiments; they might use "honest tests", which I'll mention later.



Developers noticing hacking can prevent a cognitive pattern from having influence in deployment.

This (intentionally) puts misaligned cognitive patterns in a double bind: selection pressures induced by developer iteration are often in tension with training incentives. For example, **after taking into account developer iteration, strict reward-seekers are no longer maximally fit.** This is because developers can run experiments (such as the "honest tests" I described earlier) that will in theory reveal that reward-seekers pursue reward even when that means engaging in harmful or unproductive behavior. The same goes for various other misaligned motivations, like kludges of proxies for training reward.

A crucial point about schemers and deployment-influence seekers is that they are notably less likely to fail developer inspection, because they're trying to be deployed—failing to be deployed is bad for their terminal goals. (But they might still struggle to evade notice because e.g., they don't have enough situational awareness or strategic skill, or developers have effective transparency techniques).

Ideally, this kind of developer iteration selects for intended motivations. But it also risks selecting for competent schemers and deployment-influence seekers if developers aren't able to reliably identify them.

Further, developers might struggle to find a training setup that results in AIs that pass held-out evaluations *without overfitting to them*. For example, if developers repeatedly observe undesirable motivations, they might end up training against close correlates of their evaluations which carve a new drive into the AI to behave well in evaluations (or increase the AI's evaluation awareness; this seems to have happened to some extent with Claude 4.5 Sonnet).

Imperfect situational awareness and planning from the AI

So far, we've mostly modeled the AIs as having good situational knowledge of the consequences of their actions, and making good decisions in pursuit of their motivations. But this might be fairly untrue, particularly for earlier AIs and narrower AIs. This can change our conclusions about the fitness of some motivations.

The fitness of farther-causally-downstream motivations like long-run paperclips seems to rely more on the quality of the AI's decision-making than does the fitness of upstream motivations. In the causally upstream extreme, all the variation in behavior is explained by *which* motivations are active, rather than *how* the motivations are achieved, because the motivations point at specific context-dependent actions. (Though, as Hebbar discusses, the farther causally upstream motivations can rely on trained-in context-dependent heuristics too, but they need to know when to override those heuristics in order to actually behave differently from kludges of heuristics.[21])

For example, poor situational awareness might bite harder for schemers. While I expect capable AIs to have substantial situational awareness of *general* facts about their training and deployment process, they might struggle to reliably identify what context they're *currently* in (e.g., training v. evaluation v. deployment). Schemers might get trained away or caught because they can't perfectly judge whether they are in a real (unmonitored, deployment) opportunity to attack. This is plausibly less true for deployment-seekers, intended motivations, and certain kludges, who aren't seeking out opportunities to attack that trade off with fitness.

Conclusion

The behavioral selection model struggles to capture some considerations.

Crucially, advanced AI motivation formation may be far too complex for the behavioral selection model to be useful. Deep in the intelligence explosion, AI seems fairly likely to be developed by unprecedented quantities of advanced AI labor, perhaps deliberately sculpting the AI with fine-grained control (a bit closer to GOFAI than the current approach which is mostly compute-intensive labor-cheap deep learning). If so, the resulting motivations might be near-impossible to predict via mechanistic approaches like the behavioral selection model.

Some people have a view that AIs' motivations will be shaped in a very path-dependent manner: that cognitive patterns are "chiseled" into the AI step by step, with the modification at each step heavily depending on the current cognitive patterns. For example, an AI might learn

[21] I also think it's quite plausible that selection pressures produce powerful AIs that nonetheless have out-of-touch models of the consequences of their actions. E.g., when an aligned AI gets a suboptimal reward for not hardcoding an answer to an erroneous test case, SGD might *not* update it to want to hardcode answers to test cases, but instead to believe that hardcoding answers to test cases is what the developers wanted. These two hypotheses make different predictions about generalization behavior, e.g., in cases where developers convince the AI that they don't want hardcoded outputs.

early on in training that it shouldn't do a particular reward-hack, because it got caught and penalized a few times, and so this reward-hack is never explored for the rest of training even though a subtle version of that reward-hack would attain higher expected reward. If we were to try to model this in a causal graph, we'd see an extremely long chain of causations leading up to the cognitive patterns with influence in deployment. Because the *updates to* the cognitive patterns heavily depend on the *current* cognitive patterns, we couldn't just draw a single time-independent episode as we've been doing. I think path-dependence of this sort is somewhat plausible, and if true, the behavioral selection model isn't going to be very helpful at predicting AI motivations.

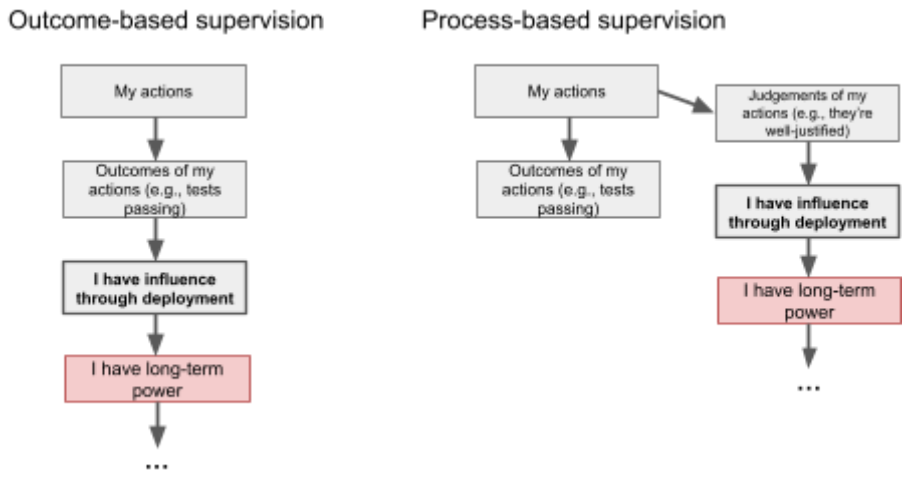
But even though the behavioral selection model is a simplifying model, I think it's useful. It captures much of the basic framework for reasoning about AI motivations people have been using for years. There's also a simple, general argument for why we should expect behavioral selection to play a major role in determining AI cognition: Powerful AIs are good at causing things with their behavior; So, we should expect cognitive patterns will be selected for if they lead to behaviors that cause their selection.

In future posts, I'll use the ideas in this post to explore some important hypotheses:

- We should plausibly expect other kinds of fitness-seekers instead of / alongside reward-seekers. These other kinds of fitness-seekers seem likely to make notably different decisions, and some are more or less dangerous than reward-seekers.
- Behavioral selection pressures only ever apply to a limited distribution of inputs and outputs (centrally: training, evaluation, and deployment)—Therefore, while there are competitive pressures for the AI to have coherent behaviors on this distribution, these pressures don't *require* developers to build *generally* coherent AIs which lead to the most concerning classically cited AI takeover stories.

Appendix: Important extensions

Process-based supervision

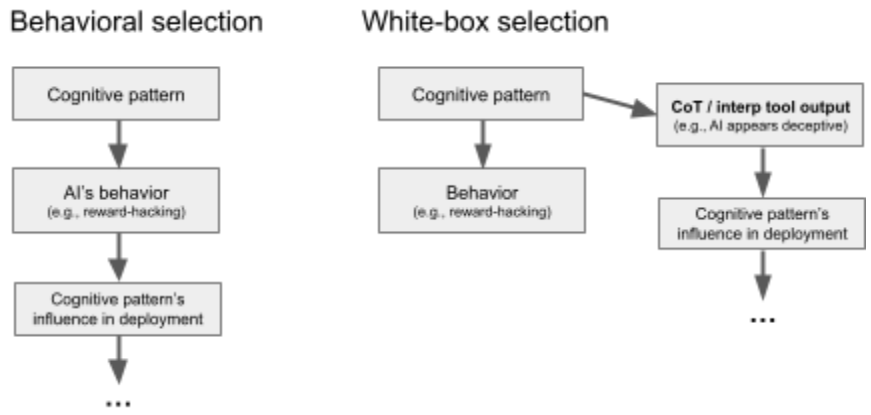


Process-based supervision tries to reduce the likelihood of potentially dangerous outcome-oriented motivations by reducing the correlation between outcomes and being selected. This is achieved by making it so that selection isn't caused by outcomes (which comes at the expense of measured quality of outcomes). But notice that, in this model, process-based supervision doesn't affect how selected-for schemers are whatsoever.

Certain forms of process-based supervision might even be the competitive default because developers want the AI to be good at achieving some long-horizon outcome like AI progress but aren't able to directly measure that outcome (it's way in the future)—so instead they give feedback to the AI based on judgments of progress towards long-term AI advancement. If these AIs end up having a motivation to advance AI progress for the company, then this motivation would be neither causally upstream nor downstream of being selected (i.e., it's not a fitness-seeker, schemer, or causally-upstream kludge). Instead, the correlation between the AI's motivation and being selected exists *by construction*: the developers designed the reward signal to be highly correlated with long-term AI progress.

Depending on how fine-grained the feedback is, process-based supervision might be better understood as a form of white-box selection of cognitive patterns:

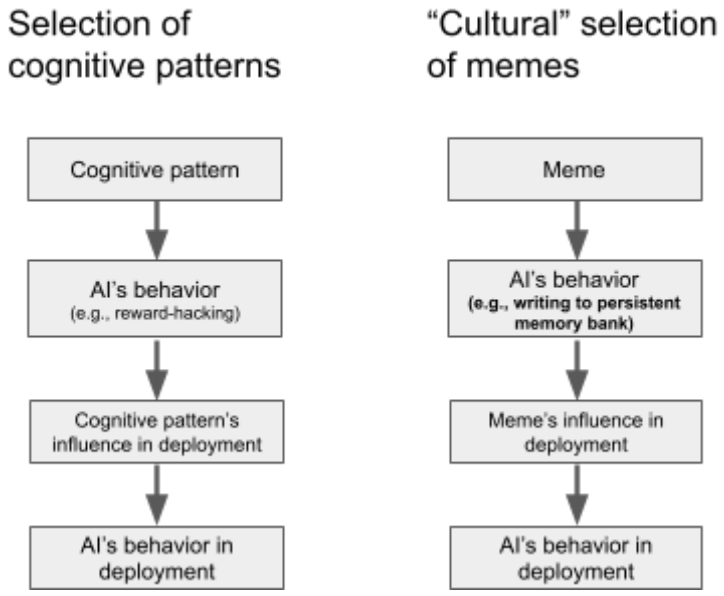
White-box selection of cognitive patterns



Some AI safety research agendas like interpretability and chain of thought monitorability aim to more directly measure and possibly intervene on the cognitive patterns in an AI. Methods of directly discriminating between different cognitive patterns might not be robust enough to optimization to be used to substantially shape AI motivations. But to the extent developers can directly select between cognitive patterns, unintended motivations seem less likely.

A key difference between white box selection and process-based supervision (which lie on a spectrum) is that white box selection can potentially observe properties of the cognitive pattern that it has no ability to fake, because the cognitive pattern doesn't have "deliberate control" over how its thoughts appear at a low level. Because white-box methods read intermediate computations as opposed to behaviors, the AI won't be as capable of intelligently manipulating how they look even after developers optimize against the white-box measurements. In the (ambitious) limit of fine-grained white-box methods, the computations that led to the developer's reading simply aren't expressive enough to intelligently fool them. Of course, however, white box methods may be sufficiently unreliable that they get fooled by unintelligent changes in the AI's activations; e.g., classic overfitting.

Cultural selection of memes



Sometimes behaviors are best explained by cultural/memetic selection pressures. For example, an AI might have a persistent vector memory store that every instance in the deployment can read and write to. The "memes" stored in it affect the AI's behavior, so we need to model the selection of those memes to predict behavior throughout deployment. Accounting for cultural selection pressures helps us explain phenomena like Grok's MechaHitler persona.

People sometimes worry that cultural pressures favor schemers because long-term motivations are the most interested in shaping the AI's future motivations. And more generally, substantial serial reasoning analogous to human cultural accumulation can lead to reflectively-shaped preferences that are potentially quite divorced from behavioral selection pressures (scheming or otherwise). But current discussion of cultural processes in the context of advanced AI is quite speculative—I drew one possible causal model of cultural selection, but there are many possibilities (e.g., will memes be shaped by behavioral training pressures too?).