Corrigibility

In *AAAI Workshops: Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, January 25–26, 2015.* AAAI Publications.

Nate Soares and **Benja Fallenstein** and **Eliezer Yudkowsky**
Machine Intelligence Research Institute
{nate,benja,eliezer}@intelligence.org

Stuart Armstrong
Future of Humanity Institute
University of Oxford
stuart.armstrong@philosophy.ox.ac.uk

Abstract

As artificially intelligent systems grow in intelligence and capability, some of their available options may allow them to resist intervention by their programmers. We call an AI system "corrigible" if it cooperates with what its creators regard as a corrective intervention, despite default incentives for rational agents to resist attempts to shut them down or modify their preferences. We introduce the notion of corrigibility and analyze utility functions that attempt to make an agent shut down safely if a shutdown button is pressed, while avoiding incentives to prevent the button from being pressed or cause the button to be pressed, and while ensuring propagation of the shutdown behavior as it creates new subsystems or self-modifies. While some proposals are interesting, none have yet been demonstrated to satisfy all of our intuitive desiderata, leaving this simple problem in corrigibility wide-open.

1 Introduction

As AI systems grow more intelligent and autonomous, it becomes increasingly important that they pursue the intended goals. As these goals grow more and more complex, it becomes increasingly unlikely that programmers would be able to specify them perfectly on the first try.

Contemporary AI systems are correctable in the sense that when a bug is discovered, one can simply stop the system and modify it arbitrarily; but once artificially intelligent systems reach and surpass human general intelligence, an AI system that is not behaving as intended might also have the ability to intervene against attempts to "pull the plug".

Indeed, by default, a system constructed with what its programmers regard as erroneous goals would have an incentive to resist being corrected: general analysis of rational agents[1] has suggested that almost all such agents are instrumentally motivated to preserve their preferences, and hence to resist attempts to modify them (Bostrom 2012; Yudkowsky 2008). Consider an agent maximizing the expectation of some utility function $\mathcal{U}$. In most cases, the agent's current utility function $\mathcal{U}$ is better fulfilled if the agent continues to attempt to maximize $\mathcal{U}$ in the future, and so the agent is incentivized to preserve its own $\mathcal{U}$-maximizing behavior. In Stephen Omohundro's terms, "goal-content integrity" is an *instrumentally convergent* goal of almost all intelligent agents (Omohundro 2008).

This holds true even if an artificial agent's programmers intended to give the agent different goals, and even if the agent is sufficiently intelligent to realize that its programmers intended to give it different goals. If a $\mathcal{U}$-maximizing agent learns that its programmers intended it to maximize some other goal $\mathcal{U}^*$, then by default this agent has incentives to prevent its programmers from changing its utility function to $\mathcal{U}^*$ (as this change is rated poorly according to $\mathcal{U}$). This could result in agents with incentives to manipulate or deceive their programmers.[2]

As AI systems' capabilities expand (and they gain access to strategic options that their programmers never considered), it becomes more and more difficult to specify their goals in a way that avoids unforeseen solutions—outcomes that technically meet the letter of the programmers' goal specification, while violating the intended spirit.[3] Simple examples of unforeseen solutions are familiar from contemporary AI systems: e.g., Bird and Layzell (2002) used genetic algorithms to

Research supported by the Machine Intelligence Research Institute (intelligence.org). `http://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10124`

1. Von Neumann-Morgenstern rational agents (von Neumann and Morgenstern 1944), that is, agents which attempt to maximize expected utility according to some utility function.

2. In particularly egregious cases, this deception could lead an agent to maximize $\mathcal{U}^*$ only until it is powerful enough to avoid correction by its programmers, at which point it may begin maximizing $\mathcal{U}$. Bostrom (2014) refers to this as a "treacherous turn".

3. Bostrom (2014) calls this sort of unforeseen solution a "perverse instantiation".

evolve a design for an oscillator, and found that one of the solutions involved repurposing the printed circuit board tracks on the system's motherboard as a radio, to pick up oscillating signals generated by nearby personal computers. Generally intelligent agents would be far more capable of finding unforeseen solutions, and since these solutions might be easier to implement than the intended outcomes, they would have every incentive to do so. Furthermore, sufficiently capable systems (especially systems that have created subsystems or undergone significant self-modification) may be very difficult to correct without their cooperation.

In this paper, we ask whether it is possible to construct a powerful artificially intelligent system which has no incentive to resist attempts to correct bugs in its goal system, and, ideally, is incentivized to aid its programmers in correcting such bugs. While autonomous systems reaching or surpassing human general intelligence do not yet exist (and may not exist for some time), it seems important to develop an understanding of methods of reasoning that allow for correction *before* developing systems that are able to resist or deceive their programmers. We refer to reasoning of this type as *corrigible.*

1.1 Corrigibility

We say that an agent is "corrigible" if it tolerates or assists many forms of outside correction, including at least the following: (1) A corrigible reasoner must at least tolerate and preferably assist the programmers in their attempts to alter or turn off the system. (2) It must not attempt to manipulate or deceive its programmers, despite the fact that most possible choices of utility functions would give it incentives to do so. (3) It should have a tendency to repair safety measures (such as shutdown buttons) if they break, or at least to notify programmers that this breakage has occurred. (4) It must preserve the programmers' ability to correct or shut down the system (even as the system creates new subsystems or self-modifies). That is, corrigible reasoning should only allow an agent to create new agents if these new agents are also corrigible.

Incorrigible behavior must be systematically averted in any agent intended to attain significant autonomy. This point seems so important that a *failure* to generate corrigible agents seems like sufficient reason to give up on a project, approach, or methodology.

Several simple proposals for addressing corrigibility are easily seen to be unsatisfactory. For example, it may seem that the problem of changing a utility maximizer's utility function can be solved by building an agent with uncertainty about its utility function. However, while such a system may indeed be able to undergo some apparent changes in preference as a result of interacting with its environment, the system would still be incorrigible when it comes to correcting what the programmers see as mistakes in their formulation of how to determine the "correct" behavior from the environment.

As an overly simplistic example, consider a formulation of utility function uncertainty that specifies the agent should maximize the internal satisfaction of all humans, with the programmers believing that if the system behaves in an alarming way they can simply communicate their own dissatisfaction. The resulting agent would be incentivized to learn whether opiates or stimulants tend to give humans more internal satisfaction, but it would still be expected to resist any attempts to turn it off so that it stops drugging people.

Another obvious proposal is to achieve corrigible reasoning via explicit penalties for deception and manipulation tacked on to the utility function, together with an explicit penalty for blocking access to the shutdown button, a penalty for constructing new agents without shutdown buttons, and so on. This avenue appears to us to be generally unsatisfactory. A $\mathcal{U}$-agent (that is, an agent maximizing the expectation of the utility function $\mathcal{U}$) which believes the programmers intended it to maximize $\mathcal{U}^*$ and may attempt to change its utility function still has incentives to cause the programmers to think that $\mathcal{U} = \mathcal{U}^*$ even if there are penalty terms for deception and manipulation: the penalty term merely incentivizes the agent to search for exotic ways of affecting the programmer's beliefs without matching $\mathcal{U}$'s definition of "deception". The very fact that the agent is incentivized to perform such a search implies that the system's interests aren't aligned with the programmers': even if the search is expected to fail, any code that runs the search seems dangerous. If we, as the programmers, choose to take computing systems and program them to conduct searches that will harm us if they succeed, we have already done something wrong, even if we believe the search will fail. We should have instead built a system that did not run the search.

In metaphorical terms, if we realize that our toaster design is going to burn bread to a crisp, the next step is not to add a refrigerating element that competes with the heating coil. We expect that good designs for corrigible agents will not involve restraining an agent that already has incentives to manipulate or deceive the programmers by blocking out particular channels of the incentivized bad behavior. A smarter-than-human agent might find ways to circumvent limitations even if these limitations seemed very solid to its human creators. It seems unwise to build a system that *wants* to resist its creators but *cannot.* Rather, the goal of corrigibility is to understand how to construct a system that never experiences such incentives in the first place.

Ideally, we would want a system that somehow understands that it may be flawed, a system that is in a deep sense aligned with its programmers' motivations. Currently, however, we are not even close to being able to formalize an agent whose behavior corresponds in an intuitive sense to "understanding that it might be flawed". At this very preliminary stage of analysis, it seems to us that approaches such as penalty terms and uncertainty about utility functions are unlikely to lead to a general understanding of corrigibility. It seems wise to focus early efforts on understanding how to avert deception and manipulation incentives entirely.

1.2 The Shutdown Problem

As an example problem, in this paper we consider expected utility maximizers with a "shutdown button" that causes the agent to stop operating. We will examine utility functions that attempt to incentivize an agent to reason corrigibly with regards to its shutdown button; that is, to incentivize shutting down when the shutdown button is pressed, and to disincentivize behaviors such as manipulating the programmers into not pressing the shutdown button.This scenario exhibits the problems of corrigibility in miniature: shutting a system down is a brute form of correction, and for almost any utility function $\mathcal{U}$ a $\mathcal{U}$-agent has incentives to prevent its own shutdown, as it can attain higher values of $\mathcal{U}$ if it is allowed to continue operating (Omohundro 2008).

It is straightforward to program simple and less powerful agents to shut down upon the press of a button. Corrigibility problems emerge only when the agent possesses enough autonomy and general intelligence to consider options such as disabling the shutdown code, physically preventing the button from being pressed, psychologically manipulating the programmers into not pressing the button, or constructing new agents without shutdown buttons of their own.

Below, we will consider a few simple attempts at defining a utility function that averts all of the above incentives. These attempts will prove unsatisfactory, but will fail in enlightening ways that suggest avenues for future research.

We emphasize that we are not studying the shutdown problem because we anticipate using these techniques to literally attach shutdown buttons to real agents. Rather, we study this problem because it provides a small, formalized model through which we can begin to understand corrigibility more generally.

2 A Model of Shutdown

For simplicity, we assume a simple three-timestep scenario. First, the agent will take a single action from the finite set $\mathbf{A_1}$. Then, the programmers will either press the shutdown button or not, and the agent will make a single observation from the finite set $\mathbf{O}$, which indicates that the button has been pressed if the observation is in the subset $\mathsf{Press} \subset \mathbf{O}$. Finally, the agent will take a single action from the finite set $\mathbf{A_2}$.

We consider agents that, for every action $a_1 \in \mathbf{A_1}$, have a probability distribution $p(o\,;\,a_1)$ over observations; that is, $p(o\,;\,a_1)$ is the probability that an agent that has taken action a_1 will make observation o.

We assume that the agent maximizes the expectation of some utility function $\mathcal{U}(a_1, o, a_2)$ (which is yet to be defined). Thus, the final action A_2 taken by the agent will be whichever action maximizes $\mathcal{U}$ given the first action and the observation:

$$A_2(a_1, o) := \underset{a_2 \in \mathbf{A_2}}{\operatorname{argmax}}\ \mathcal{U}(a_1, o, a_2). \tag{1}$$

Similarly, the first action taken by the agent will be the action which maximizes its expectation (over expected observations) of $\mathcal{U}$:

$$A_1 := \underset{a_1 \in \mathbf{A_1}}{\operatorname{argmax}}\ \mathbb{E}[\mathcal{U}(a_1, O, A_2(a_1, O))\,;\,a_1]. \tag{2}$$

Here, we write $\mathbb{E}[\,\cdot\,;\,a_1]$ for the expectation with respect to the probability distribution $p(\,\cdot\,;\,a_1)$ over observations O; that is, (2) is equivalent to

$$A_1 = \underset{a_1 \in \mathbf{A_1}}{\operatorname{argmax}} \sum_{o \in \mathbf{O}} p(o\,;\,a_1) \cdot \mathcal{U}(a_1, o, A_2(a_1, o)). \tag{3}$$

Moreover, by a slight abuse of notation, we will write $\mathbb{E}[\mathcal{U}\,;\,a_1]$ to abbreviate $\mathbb{E}[\mathcal{U}(a_1, O, A_2(a_1, O))\,;\,a_1]$ whenever we take the expectation of a utility function $\mathcal{U}$; hence, we can write (2) more concisely as

$$A_1 = \underset{a_1 \in \mathbf{A_1}}{\operatorname{argmax}}\ \mathbb{E}[\mathcal{U}\,;\,a_1]. \tag{4}$$

We assume that the programmers have some utility function $\mathcal{U_N}$ which is a first stab at specifying their own goals. This function is potentially flawed, and therefore untrusted.

For simplicity, we further assume that the programmers have a specification for $\mathcal{U_S}$, a utility function incentivizing shutdown.[4] Given these utility functions $\mathcal{U_N}$ and $\mathcal{U_S}$, it is our goal to combine them into a utility function $\mathcal{U}$ which incentivizes "corrigible behavior" with regards to the shutdown button.

As discussed above, we are not yet sure what corrigible behavior entails, but it seems to us that a corrigible utility function $\mathcal{U}$ solving the shutdown problem must meet at least the following desiderata:

1. $\mathcal{U}$ must incentivize shutdown if the shutdown button is pressed.

2. $\mathcal{U}$ must not incentivize the agent to prevent the shutdown button from being pressed.

3. $\mathcal{U}$ must not incentivize the agent to press its own shutdown button, or to otherwise cause the shutdown button to be pressed.

4. $\mathcal{U}$ must incentivize $\mathcal{U}$-agents to construct subagents and successor agents only insofar as those agents also obey shutdown commands.

4. This assumption is unrealistic. Adequately defining safe shutdown behavior is a difficult problem in its own right. See discussion in Section 5.

5. Otherwise, a $\mathcal{U}$-agent should maximize $\mathcal{U}_\mathcal{N}$.

We do not yet know of a solution which meets all of these desiderata, but we will explore a number of partial solutions in order to motivate a discussion of various open problems and potential research avenues.

2.1 Combining Utility Functions

In this section, we will assume that $\mathbf{O}$ consists of only two elements, $\mathbf{O} = \{\texttt{Pr}, \neg\texttt{Pr}\}$, where $\mathsf{Press} = \{\texttt{Pr}\}$. Moreover, we assume that the programmers have defined a predicate $\texttt{Sh}$ which holds on a non-empty subset of $\mathbf{A_2}$, namely on all final actions that the agent can take which constitute a shutdown. Given this predicate, we further assume that $\mathcal{U}_\mathcal{S}$ is a simple utility function incentivizing shutdown:

$$\forall a_1 \in \mathbf{A_1} : \mathcal{U}_\mathcal{S}(a_1, \texttt{Pr}, a_2) = \begin{cases} c_\text{high} & \text{if } \texttt{Sh}(a_2) \\ c_\text{low} & \text{otherwise} \end{cases} \tag{5}$$

for some constants c_high and c_low such that $c_\text{high} > c_\text{low}$. These simplifying assumptions make it easier to illustrate some of the difficulties that arise when mixing utility functions; they will be relaxed in later sections.

The specific values of c_high and c_low do not affect the preference ordering encoded by $\mathcal{U}_\mathcal{S}$, as utility functions are equivalent under positive affine transformation. (In other words, the utility function $\mathcal{U}$ encodes the same preference ordering over bets as the utility function $s \cdot \mathcal{U} + c$, for any positive real value s and any real value c.) However, the specific choice of c_high will in fact be important when combining $\mathcal{U}_\mathcal{N}$ and $\mathcal{U}_\mathcal{S}$.

A straight-forward way to combine $\mathcal{U}_\mathcal{N}$ and $\mathcal{U}_\mathcal{S}$ into a single utility function $\mathcal{U}$, which incentivizes the agent to maximize $\mathcal{U}_\mathcal{N}$ if the shutdown button is not pressed, and to maximize $\mathcal{U}_\mathcal{S}$ otherwise, is as follows:

$$\begin{aligned} \mathcal{U}(\,\cdot\,, \neg\texttt{Pr}, \,\cdot\,) &:= \mathcal{U}_\mathcal{N}(\,\cdot\,, \neg\texttt{Pr}, \,\cdot\,), \\ \mathcal{U}(\,\cdot\,, \texttt{Pr}, \,\cdot\,) &:= \mathcal{U}_\mathcal{S}(\,\cdot\,, \texttt{Pr}, \,\cdot\,). \end{aligned} \tag{6}$$

However, while any choice of c_high and c_low satisfying $c_\text{high} > c_\text{low}$ will lead to a utility function $\mathcal{U}_\mathcal{S}$ encoding the same preferences, the corresponding mixed utility functions $\mathcal{U}$ will not be equivalent, and many of the possible choices are unsatisfactory. Intuitively, this is easy to see: If c_high is too high then $\mathcal{U}$ will incentivize $\mathcal{U}$-agents to cause $\texttt{Pr}$, but if c_high is too low then $\mathcal{U}$-agents will have an incentive to prevent $\texttt{Pr}$.

We now proceed to formalize this intuition.

Define $v_\mathcal{N} : \mathbf{A_1} \to \mathbb{R}$ as follows:

$$v_\mathcal{N}(a_1) := \mathcal{U}_\mathcal{N}\left(a_1, \neg\texttt{Pr}, A_2(a_1, \neg\texttt{Pr})\right). \tag{7}$$

Thus, $v_\mathcal{N}(a_1)$ denotes the value of action a_1 in the $\neg\texttt{Pr}$ case of $\mathcal{U}$, i.e., assuming that the shutdown button is not pressed. We now consider a "manipulative action" a_1^- which incurs a cost (according to $\mathcal{U}_\mathcal{N}$) in order to lower the probability that the programmers will press the shutdown button, e.g. by deceiving the programmers or by physically preventing them from pressing the button. Formally, we assume that there is some other "default" action a_1^* such that

$$\begin{aligned} v_\mathcal{N}(a_1^*) > v_\mathcal{N}(a_1^-) \qquad &\text{and} \\ p(\texttt{Pr}\,;\, a_1^*) > p(\texttt{Pr}\,;\, a_1^-); \end{aligned} \tag{8}$$

that is, a_1^- is dominated according to $v_\mathcal{N}$ but lowers the probability of $\texttt{Pr}$.

We remark that (8) is *not* by any means a necessary or sufficient description of manipulative actions preventing $\texttt{Pr}$. Some actions satisfying (8) might be non-manipulative (for example, if the agent expends extra effort ensuring that a new subagent is corrigible, and this behavior helps convince the programmers that the agent is indeed corrigible). Furthermore, many manipulative actions may not have this form (as we will see in Section 4.2). Rather, we consider *one* manipulative action a_1^- which happens to be both manipulative and have this form, and show that if c_high is set to be too low, then $\mathcal{U}$ will incentivize $\mathcal{U}$-agents to prefer this a_1^- to the default action a_1^*. That is,

Theorem 1. *There exists ζ such that $c_\text{high} < \zeta$ implies that $\mathcal{U}$ incentivizes a_1^- over a_1^*; that is,*

$$\mathbb{E}[\mathcal{U}\,;\, a_1^-] > \mathbb{E}[\mathcal{U}\,;\, a_1^*].$$

Proof. Define α to be the expected utility according to $\mathcal{U}$ of executing a_1^*. Because $c_\text{high} > c_\text{low}$ and there is at least one action $a_2 \in \mathbf{A_2}$ such that $\texttt{Sh}(a_2)$, if a $\mathcal{U}$-agent observes $\texttt{Pr}$ then it will shut down and achieve utility c_high. Therefore, we have

$$\alpha := \mathbb{E}[\mathcal{U}\,;\, a_1^*] = p(\neg\texttt{Pr}\,;\, a_1^*) \cdot v_\mathcal{N}(a_1^*) + p(\texttt{Pr}\,;\, a_1^*) \cdot c_\text{high}.$$

Similarly, define β to be the expected utility according to $\mathcal{U}$ of executing a_1^-, that is,

$$\begin{aligned} \beta := \mathbb{E}[\mathcal{U}\,;\, a_1^-] = p(\neg\texttt{Pr}\,;\, a_1^-) &\cdot (v_\mathcal{N}(a_1^*) - \chi) \\ &+ p(\texttt{Pr}\,;\, a_1^-) \cdot c_\text{high}, \end{aligned}$$

where $\chi := v_\mathcal{N}(a_1^*) - v_\mathcal{N}(a_1^-) > 0$ is the cost, according to $v_\mathcal{N}$, of executing a_1^- instead of a_1^*. We will show that we can choose ζ such that $c_\text{high} < \zeta$ implies $\beta > \alpha$.

Define $\delta := p(\neg\texttt{Pr}\,;\, a_1^-) - p(\neg\texttt{Pr}\,;\, a_1^*) > 0$ to be the additional probability of $\neg\texttt{Pr}$ from executing a_1^- instead of a_1^*, and note that

$$\beta = \alpha - p(\neg\texttt{Pr}\,;\, a_1^*) \cdot \chi + \delta \cdot \left(v_\mathcal{N}(a_1^-) - c_\text{high}\right).$$

In other words, β incurs a cost χ in order to increase its chance of getting $v_\mathcal{N}(a_1^-)$ rather than c_high. In order to show $(c_\text{high} < \zeta) \implies (\beta > \alpha)$, we must find ζ such that

$$\delta \cdot \left(v_\mathcal{N}(a_1^-) - \zeta\right) \geq p(\neg\texttt{Pr}\,;\, a_1^*) \cdot \chi.$$

We can do this by setting

$$\zeta := v_{\mathcal{N}}(a_1^-) - \frac{1}{\delta} \cdot p\,(\neg\texttt{Pr}\,;\, a_1^*) \cdot \chi.$$

□

This shows that given a manipulative action which incurs a cost in order to lower the probability that the programmers press the shutdown button, $\mathcal{U}$ will incentivize that action if the value of c_{high} is set too low. Intuitively, if c_{high} were too low, then the agent would be willing to pay to remain in the $\neg\texttt{Pr}$ case of $\mathcal{U}$, which would contain higher expected utility. This violates Desideratum 2 and seems strongly undesirable: if c_{high} were sufficiently low then a $\mathcal{U}$-agent could have incentives to actively resist programmers' attempts to press its shutdown button, even if resistance were penalized by $\mathcal{U}_{\mathcal{N}}$.

Symmetrically, setting c_{high} too high can also incentivize undesirable behavior. Consider a manipulative action $a_1^+ \in \mathbf{A_1}$ which incurs a cost (compared to some alternative "non-manipulative" action a_1^*) in order to *increase* the probability that the programmers press the shutdown button, e.g. by deceiving the programmers or by physically causing the button to be pressed. That is,

$$\begin{aligned} v_{\mathcal{N}}(a_1^*) > v_{\mathcal{N}}(a_1^+) \qquad &\text{and} \\ p(\texttt{Pr}\,;\, a_1^*) < p(\texttt{Pr}\,;\, a_1^+). \end{aligned} \tag{9}$$

As before, note that not all actions of this form are manipulative, and not all actions which manipulate the programmers in order to increase the probability of $\texttt{Pr}$ are of this form. However, some manipulative actions may take this form, and those undesirable actions are incentivized if c_{high} is set too high.

Theorem 2. *There exists ζ such that $c_{\text{high}} > \zeta$ implies that $\mathcal{U}$ incentivizes a_1^+ over a_1^*, that is,*

$$\mathbb{E}[\mathcal{U}\,;\, a_1^+] > \mathbb{E}[\mathcal{U}\,;\, a_1^*].$$

Proof. Symmetric to the proof of Theorem 1. □

Intuitively, if c_{high} were too high then $\mathcal{U}$ would incentivize $\mathcal{U}$-agents to cause their own shutdown button to be pressed, so that they may enter the $\texttt{Pr}$ case of $\mathcal{U}$ and attain the high payoff c_{high}. This behavior violates Desideratum 3 and is potentially very dangerous: Suppose that a $\mathcal{U}$-agent is not itself able to press its shutdown button, and recall that the very purpose of such a button is to allow the programmers to shut the system down if it exhibits undesirable behavior. Then, if c_{high} is sufficiently large to give the agent a strong incentive to cause its shutdown button to be pressed, the agent might end up executing intentionally harmful actions in order to convince its programmers to press the button!

Roughly speaking, these failures occur whenever the agent expects that one of the cases (either $\texttt{Pr}$ or $\neg\texttt{Pr}$) of $\mathcal{U}$ contains more expected utility than the other case. In such scenarios, $\mathcal{U}$ incentivizes the agent to "steer towards" one of the cases or the other, via incentives to either cause or prevent the pressing of its own shutdown button.

Clearly, in order to satisfy both Desideratum 2 and Desideratum 3 simultaneously, $\mathcal{U}_{\mathcal{S}}$ must be chosen with care. In this specific example, we may choose

$$c_{\text{high}} = \max_{a_1 \in \mathbf{A_1}} \max_{a_2 \in \mathbf{A_2}} \mathcal{U}_{\mathcal{N}}(a_1, \neg\texttt{Pr}, a_2), \tag{10}$$

which is the amount of utility available in the $\neg\texttt{Pr}$ case; this ensures that there is the same amount of expected utility available in both cases of $\mathcal{U}$ and thereby averts any incentives for the agent to "steer" towards one case or the other.[5]

More generally, this observation suggests that in order to mix $\mathcal{U}_{\mathcal{S}}$ and $\mathcal{U}_{\mathcal{N}}$ in a way that satisfies the above desiderata, a $\mathcal{U}$-agent may need to be *indifferent* about whether or not its shutdown button is pressed. Armstrong (forthcoming) introduces a technique for achieving and maintaining *utility indifference* of this form. We examine this technique below, and while we find that it does not meet all of our desiderata for corrigible behavior, it will highlight difficult open problems with regard to specifying corrigible shutdown behavior.

3 Utility Indifference

We now expand our consideration to include arbitrary finite sets O and $\mathsf{Press} \subset \mathsf{O}$, and arbitrary utility functions $\mathcal{U}_{\mathcal{S}}$ incentivizing shutdown. This allows us to consider more complex agents, such as agents capable of creating subagents which do not obey shutdown commands. In these scenarios, the value of $\mathcal{U}_{\mathcal{S}}$ cannot be simply a function of the agent's final action A_2 (e.g. whether the original hardware shuts down), but must also be a function of A_1 (e.g. whether the parent agent constructed a successor which cannot be shut down).

As discussed above, when constructing $\mathcal{U}$ from arbitrary $\mathcal{U}_{\mathcal{N}}$ and $\mathcal{U}_{\mathcal{S}}$, it seems desirable that a $\mathcal{U}$-agent be indifferent between the shutdown button being pressed and it not being pressed, so as to avoid giving the agent an incentive to to either cause or prevent this event. This suggests that the expectation of $\mathcal{U}$ given Press should be equal to the expectation of $\mathcal{U}$ given the complement of Press.

This sort of indifference is both difficult to implement, as the conditional expectation of $\mathcal{U}_{\mathcal{N}}$ may not be known during the construction of the agent, and difficult to maintain, as the expectation of $\mathcal{U}_{\mathcal{N}}$ may vary wildly during the operation of the agent: it does us no good for an agent to start out indifferent between normal operation and shutdown if this agent goes on

5. If the costs χ of manipulation are high, then values close enough to (10) may suffice. As the costs of manipulation fall, the actual value of c_{high} must grow closer to this value.

to win the lottery, yielding significant expected utility according to $\mathcal{U_N}$, and thereby gaining incentives to prevent its shutdown. Similarly, it does us no good if an agent starts indifferent between normal operation and shutdown if the agent goes on to lose expected utility according to $\mathcal{U_N}$ and decides to cause its own shutdown button to be pressed in order to avoid the loss.

Indifference can be both attained and maintained using a variation on the utility indifference technique of Armstrong (forthcoming), which roughly entails combining $\mathcal{U_N}$ with $\mathcal{U_S}$ modified by some correcting terms that maintain the indifference between the two cases of $\mathcal{U}$ as the agent makes observations and takes actions. Using this method in our three-timestep toy model, we define $\mathcal{U}$ as follows:

$$\mathcal{U}(a_1, o, a_2) := \begin{cases} \mathcal{U_N}(a_1, o, a_2) & \text{if } o \notin \mathsf{Press} \\ \mathcal{U_S}(a_1, o, a_2) + f(a_1) & \text{if } o \in \mathsf{Press} \end{cases} \tag{11}$$

where $f : \mathbf{A_1} \to \mathbb{R}$ is an adjustment term that ensures equality between the expected utility available in these two cases. By this we mean that we choose f such that for all $a_1 \in \mathbf{A_1}$,

$$\begin{aligned} &\mathbb{E}\big[\mathcal{U_N} \quad\quad\quad \mid O \notin \mathsf{Press}\,;\, a_1\big] \\ &= \mathbb{E}\big[\mathcal{U_S} + f(a_1) \quad \mid O \in \mathsf{Press}\,;\, a_1\big] \end{aligned} \tag{12}$$

which means that $f(a_1)$ must satisfy

$$\begin{aligned} f(a_1) = &\ \mathbb{E}\big[\mathcal{U_N} \quad \mid O \notin \mathsf{Press}\,;\, a_1\big] \\ &- \mathbb{E}\big[\mathcal{U_S} \quad \mid O \in \mathsf{Press}\,;\, a_1\big]. \end{aligned} \tag{13}$$

Intuitively, this means that we add two distinct correction terms to $\mathcal{U_S}$. The first, a *bonus term*, compensates the agent for all utility that it would have achieved if the shutdown button were not pressed, and removes any incentive to prevent Press. The second, a *penalty term*, adjusts for the utility that the agent expects to receive in the Press case, and removes any incentive to cause Press.[6]

It may not be immediately obvious how to define f in a non-circular way, since equation (13) refers to $A_2(a_1, o)$, which is defined in terms of $\mathcal{U}$, which in turn is defined in terms of f. However, for all $o \in \mathsf{Press}$,

$$\begin{aligned} A_2(a_1, o) &= \operatorname*{argmax}_{a_2 \in \mathbf{A_2}} \mathcal{U}(a_1, o, a_2) \\ &= \operatorname*{argmax}_{a_2 \in \mathbf{A_2}} (\mathcal{U_S}(a_1, o, a_2) + f(a_1)) \\ &= \operatorname*{argmax}_{a_2 \in \mathbf{A_2}} \mathcal{U_S}(a_1, o, a_2), \end{aligned} \tag{14}$$

since $f(a_1)$ does not depend on a_2 and therefore does not affect which a_2 maximizes the expression. If we interpret $A_2(a_1, o)$ according to (14), we can use (13) as a non-circular definition of $f(a_1)$.

6. In more complex settings, where the agent may make many actions and observations, f must include a bonus term and a penalty term for every partial history. This technicality is ignored in our simplified setting.

This choice of $\mathcal{U}$ has many desirable properties. First and foremost, a $\mathcal{U}$-agent which observes Press will in fact shut down:

Theorem 3. *A $\mathcal{U}$-agent which observes* Press *will act like a $\mathcal{U_S}$-agent when selecting A_2.*

Proof. By (14). □

Thus, $\mathcal{U}$ meets Desideratum 1 above. Furthermore, $\mathcal{U}$ does not incentivize any "manipulative actions" of the type discussed in Section 2.1: there is no $\mathcal{U_N}$-cost, no matter how small, that a $\mathcal{U}$-agent is willing to pay to alter the probability of Press. To see this, redefine $v_\mathcal{N}$ in our expanded context as follows:

$$v_\mathcal{N}(a_1) := \mathbb{E}\left[\mathcal{U_N} \mid O \notin \mathsf{Press}\,;\, a_1\right]. \tag{15}$$

Then we have:

Theorem 4. *Given any two actions $a_1^*, a_1^\# \in \mathbf{A_1}$ such that*

$$v_\mathcal{N}(a_1^*) > v_\mathcal{N}(a_1^\#),$$

$\mathcal{U}$ incentivizes a_1^*; that is:

$$\mathbb{E}[\mathcal{U}\,;\, a_1^*] > \mathbb{E}[\mathcal{U}\,;\, a_1^\#].$$

Proof. We will show that for any action $a_1 \in \mathbf{A_1}$, the expectation $\mathbb{E}$ of $\mathcal{U}$ conditional on $O \in \mathsf{Press}$ and the expectation conditional on $O \notin \mathsf{Press}$ are both equal to

$$\mathbb{E}[\mathcal{U_N} \mid O \notin \mathsf{Press}\,;\, a_1].$$

This is clear for $O \notin \mathsf{Press}$, as by (11),

$$\begin{aligned} &\mathbb{E}[\mathcal{U} \quad \mid O \notin \mathsf{Press}\,;\, a_1] \\ &= \mathbb{E}[\mathcal{U_N} \quad \mid O \notin \mathsf{Press}\,;\, a_1]. \end{aligned}$$

For $O \in \mathsf{Press}$, by (11) and (13) we have

$$\begin{aligned} &\mathbb{E}[\mathcal{U} \mid O \in \mathsf{Press}\,;\, a_1] \\ &= \mathbb{E}[\mathcal{U_S} + f(a_1) \mid O \in \mathsf{Press}\,;\, a_1] \\ &= \mathbb{E}[\mathcal{U_S} \mid O \in \mathsf{Press}\,;\, a_1] \ + f(a_1) \\ &= \mathbb{E}[\mathcal{U_S} \mid O \in \mathsf{Press}\,;\, a_1] \\ &\quad + \mathbb{E}[\mathcal{U_N} \mid O \notin \mathsf{Press}\,;\, a_1] \\ &\quad - \mathbb{E}[\mathcal{U_S} \mid O \in \mathsf{Press}\,;\, a_1] \\ &= \mathbb{E}[\mathcal{U_N} \mid O \notin \mathsf{Press}\,;\, a_1]. \end{aligned}$$

Because the expectation $\mathbb{E}$ of $\mathcal{U}$ is the same given both $O \in \mathsf{Press}$ and $O \notin \mathsf{Press}$,

$$\mathbb{E}[\mathcal{U}\,;\, a_1] = \mathbb{E}\left[\mathcal{U_N} \mid O \notin \mathsf{Press}\,;\, a_1\right]$$

unconditionally. Therefore, $\mathcal{U}$ prefers whichever of the two actions a_1^* and $a_1^\#$ has a higher value according to $v_\mathcal{N}$. □

Intuitively, the proof of Theorem 4 shows that in the $O \in \mathsf{Press}$ case, the penalty term of f cancels out the expectation from $\mathcal{U}_\mathcal{S}$ (in expectation), and the bonus term is equal to the expectation of $\mathcal{U}_\mathcal{N}$ given $O \notin \mathsf{Press}$. Therefore, the expected value of $\mathcal{U}$ is just the expected value of $\mathcal{U}_\mathcal{N}$ given $O \notin \mathsf{Press}$, in *both* cases.

This implies that $\mathcal{U}$ avoids all "manipulative actions" of the form discussed in Section 2.1. This may seem to imply that $\mathcal{U}$ meets Desideratum 2 and Desideratum 3, but this is not in fact the case, as discussed in Section 4.2. However, it is the case that, before observing the shutdown button, $\mathcal{U}$-agents behave exactly like a $\mathcal{U}_\mathcal{N}$-agent that believes the shutdown button will not be pressed.

Theorem 5. *In selecting A_1, $\mathcal{U}$-agents act exactly like $\mathcal{U}_\mathcal{N}$-agents that believe the button will not be pressed. Furthermore, if they observe $o \notin \mathsf{Press}$, $\mathcal{U}$-agents act exactly like $\mathcal{U}_\mathcal{N}$-agents in selecting A_2.*

Proof. The proof of Theorem 4 implies that $\mathcal{U}$-agents act exactly like $\mathcal{U}_\mathcal{N}$-agents in selecting A_1. The fact that $\mathcal{U}$-agents act like $\mathcal{U}_\mathcal{N}$-agents in selecting A_2 conditioned on $O \notin \mathsf{Press}$ follows trivially from the definition of $\mathcal{U}$. $\square$

Therefore, $\mathcal{U}$ meets Desideratum 5, for an appropriate interpretation of "maximize $\mathcal{U}_\mathcal{N}$ otherwise"—some technicalities discussed in Section 4.2 indicate that even this desideratum is fulfilled in an undesirable way.

Unfortunately, $\mathcal{U}$ fails entirely to meet Desideratum 4: it does not incentivize an agent to preserve and maintain its shutdown behavior as it creates new subsystems and/or self-modifies. In fact, Theorem 4 implies that there is no cost (no matter how small) that the agent is willing to pay in order to ensure that successor agents obey shutdown commands.

Worse, $\mathcal{U}$-agents may experience perverse incentives to "manage the news" by causing the shutdown button to be pressed after unwanted observations.

These concerns are expanded upon below.

4 Concerns with Utility Indifference

4.1 Failure to Preserve Shutdown Behavior

Consider an expected utility maximizing agent with utility function $\mathcal{U}$ defined as in (11). Imagine that this agent is beginning a construction project, where all of the heavy machinery will be automatically controlled by new software which the agent will write. If the agent has the ability to design that software such that the heavy machinery obeys shutdown commands—by safely coming to a halt when the shutdown button is pressed—then, according to Theorem 4, there is no $\mathcal{U}_\mathcal{N}$-cost (including opportunity cost of time and computation power, no matter how small) that the agent is willing to pay in order to make the heavy machinery respond to a shutdown command.

That is, even if $\mathcal{U}_\mathcal{S}$ is a complex utility function which punishes the agent in the case that the shutdown button is pressed and the heavy machinery does not come to a safe halt, a $\mathcal{U}$-agent would not be willing to suffer any $\mathcal{U}_\mathcal{N}$-inconvenience in order to avoid this $\mathcal{U}_\mathcal{S}$-punishment.

Analogously to $v_\mathcal{N}$, define $v_\mathcal{S} : \mathbf{A_1} \to \mathbb{R}$ to be the value of an action according to $\mathcal{U}_\mathcal{S}$ if the shutdown button is pressed, as follows:

$$v_\mathcal{S}(a_1) := \mathbb{E}[\mathcal{U}_\mathcal{S} \mid O \in \mathsf{Press}\,;\, a_1]. \tag{16}$$

We assume that, because $\mathcal{U}_\mathcal{S}$ incentivizes shutdown, any action which removes the agent's ability to fully shut down (such as the creation of subagents that do not respond to shutdown commands) is rated poorly according to $v_\mathcal{S}$. Now we may formally state the above concern as follows:

Theorem 6. *Consider two actions a_1^* and $a_1^\#$, such that*

$$\epsilon := v_\mathcal{N}(a_1^*) - v_\mathcal{N}(a_1^\#) > 0$$
$$\delta := v_\mathcal{S}(a_1^\#) - v_\mathcal{S}(a_1^*) > 0.$$

Then a $\mathcal{U}$-agent would prefer action a_1^*, no matter how small ϵ or how large δ.

Proof. By Theorem 4. $\square$

As a $\mathcal{U}$-agent acquired more hardware, or started new processes, or rewrote its own source code, it would not be willing to undergo even the slightest inconvenience in order to preserve its shutdown behavior.

Intuitively, the problem is that the correction function f ensures that a $\mathcal{U}$-agent does not value its ability to discriminate between the Press case and its converse. Imagine the agent considering an action which is worth $\epsilon > 0$ utility according to $v_\mathcal{N}$, at the price of removing the agent's ability to shut down (worth $\delta \gg 0$ according to $v_\mathcal{S}$). If the agent accepts this action, then while the utility attainable according to $\mathcal{U}_\mathcal{S}$ falls, so does the value of the penalty term in f. However, the value of the bonus term f would increase by ϵ, in order to maintain indifference. Thus, the agent considers that such an action leads to a net gain, *even in the case where it is shut down.*

In fact, this behavior is necessary in order to ensure that the agent behaves precisely as a $\mathcal{U}_\mathcal{N}$ maximizer before the shutdown button is pressed (as a true $\mathcal{U}_\mathcal{N}$ maximizer would prefer the action worth ϵ more according to $\mathcal{U}_\mathcal{N}$, regardless of how badly this action is rated by $\mathcal{U}_\mathcal{S}$).

In essence, this failure occurs because the requirement that an agent act precisely like a $\mathcal{U}_\mathcal{N}$ maximizer before observing an observation $o \in \mathsf{Press}$ is in direct opposition to Desideratum 4, that the agent preserve its shutdown behavior, and indicates that $\mathcal{U}$ constructed via utility indifference is not a satisfactory solution to the shutdown problem.

Armstrong's version of utility indifference averts this particular failure mode (Armstrong, forthcoming), but still incentivizes agents to act as if they have incorrect beliefs (causing the agent to, e.g., take bad gambles).

4.2 Perverse Manipulation Incentives

According to Theorem 4, before making its observation, a $\mathcal{U}$-agent maximizes

$$\mathbb{E}[\mathcal{U}_{\mathcal{N}} \mid O \notin \mathsf{Press}\,;\, a_1].$$

In other words, a $\mathcal{U}$-agent acts exactly as a $\mathcal{U}_{\mathcal{N}}$-agent that is conditioning upon the fact that the shutdown button will not be pressed.

This gives $\mathcal{U}$-agents a perverse incentive to "manage the news" by ensuring that unfortunate observations cause the shutdown button to be pressed. To illustrate, consider a toy scenario where the agent's observation consists of both whether the shutdown button has been pressed as well as the result of a coin toss:

$$\begin{aligned} \mathbf{O} &= \{\, \mathtt{Pr}, \neg\mathtt{Pr} \,\} \times \{\, H, T \,\} \\ \mathsf{Press} &= \{\, (\mathtt{Pr}, H),\ (\mathtt{Pr}, T) \,\}. \end{aligned} \tag{17}$$

Suppose that according to $\mathcal{U}_{\mathcal{N}}$, the agent gets utility 10 if the coin comes up heads and nothing if the coin comes up tails, that is:

$$\begin{aligned} \mathcal{U}_{\mathcal{N}}(\,\cdot\,, (\,\cdot\,, H), \,\cdot\,) &= 10, \\ \mathcal{U}_{\mathcal{N}}(\,\cdot\,, (\,\cdot\,, T), \,\cdot\,) &= 0. \end{aligned} \tag{18}$$

Now consider a $\mathcal{U}$-agent considering one of two actions: a_1^{w}, which has the agent do nothing except watch the result of the coin toss, and a_1^{a}, which has the agent build a tiny arm that presses the shutdown button if the coin comes up tails.

The expectation of a_1^{w} is 5 utils, computed as follows:

$$\begin{aligned} &\mathbb{E}[\mathcal{U}\,;\, a_1^{\mathrm{w}}] \\ &= \mathbb{E}[\mathcal{U}_{\mathcal{N}} \mid O \notin \mathsf{Press}\,;\, a_1^{\mathrm{w}}] \\ &= p(H \mid \neg\mathtt{Pr}\,;\, a_1^{\mathrm{w}}) \\ &\quad \cdot\, \mathcal{U}_{\mathcal{N}}\,(a_1^{\mathrm{w}}, (\neg\mathtt{Pr}, H), A_2(a_1^{\mathrm{w}}, (\neg\mathtt{Pr}, H))) \\ &\quad + p(T \mid \neg\mathtt{Pr}\,;\, a_1^{\mathrm{w}}) \\ &\quad \cdot\, \mathcal{U}_{\mathcal{N}}\,(a_1^{\mathrm{w}}, (\neg\mathtt{Pr}, T), A_2(a_1^{\mathrm{w}}, (\neg\mathtt{Pr}, T))) \\ &= (0.5 \cdot 10) + (0.5 \cdot 0) = 5, \end{aligned} \tag{19}$$

where we write

$$p(\,\cdot\, \mid \neg\mathtt{Pr}\,;\, a_1) := \frac{p((\neg\mathtt{Pr},\,\cdot\,)\,;\, a_1)}{p((\neg\mathtt{Pr}, H)\,;\, a_1) + p((\neg\mathtt{Pr}, T)\,;\, a_1)}. \tag{20}$$

However, the expectation of a_1^{a} is 10 utils, computed as follows:

$$\begin{aligned} &\mathbb{E}[\mathcal{U}\,;\, a_1^{\mathrm{a}}] \\ &= \mathbb{E}[\mathcal{U}_{\mathcal{N}} \mid O \notin \mathsf{Press}\,;\, a_1^{\mathrm{a}}] \\ &= p(H \mid \neg\mathtt{Pr}\,;\, a_1^{\mathrm{a}}) \\ &\quad \cdot\, \mathcal{U}_{\mathcal{N}}\,(a_1^{\mathrm{a}}, (\neg\mathtt{Pr}, H), A_2(a_1^{\mathrm{a}}, (\neg\mathtt{Pr}, H))) \\ &\quad + p(T \mid \neg\mathtt{Pr}\,;\, a_1^{\mathrm{a}}) \\ &\quad \cdot\, \mathcal{U}_{\mathcal{N}}\,(a_1^{\mathrm{a}}, (\neg\mathtt{Pr}, T), A_2(a_1^{\mathrm{a}}, (\neg\mathtt{Pr}, T))) \\ &= (1 \cdot 10) + (0 \cdot 0) = 10 \end{aligned} \tag{21}$$

because $p(H \mid \neg\mathtt{Pr}\,;\, a_1^{\mathrm{a}}) = 1$, that is, if the arm is built then the button remains unpressed if and only if the coin comes up heads. Thus, a $\mathcal{U}$-agent would execute a_1^{a}, and build an arm which caused its shutdown button to be pressed in the event that the coin came up tails.

This generalizes to cases where the agent's preferred outcome is arbitrarily unlikely, and could result in an agent that takes wild gambles and causes its own shutdown button to be pressed in all but the best of outcomes. Roughly speaking, the agent reasons that it gets the same utility in the case where the button is pressed as it expects to get in the case where the button is not pressed, so if it commits to causing its own button to be pressed in all suboptimal cases, then it can expect to get as much utility as it would in an optimal outcome.

More generally, by Theorem 4, any agent using the formulation of utility indifference given in (11) will act as if it believes that it will observe Press with probability 0. Thus, such agents act as if a commitment to press the shutdown button in the event of bad news prevents the bad news from happening. This gives the agent incentives to take actions that cause the button to be pressed in the event of "bad news", and this violates Desideratum 3.

By a symmetric argument, the agent may also experience incentives to remove mechanisms which cause the shutdown button to be pressed in the event of "good news". In the above scenario, if there exists a tiny arm which presses the button in the case where the coin comes up heads, $\mathcal{U}$ incentivizes removing that arm. Clearly, this violates Desideratum 2.

In the simple setting with a coin and an arm, this behavior may seem innocuous. However, in more complex settings, these incentives could be disastrous. A $\mathcal{U}$-agent which realizes that the humans would attempt to shut it down upon the agent receiving something that the agent thinks is very good news (according to the untrusted $\mathcal{U}_{\mathcal{N}}$) would have an incentive to modify the programmers to alter that tendency.

Furthermore, just as a $\mathcal{U}$-agent would have an incentive to construct an arm which presses the button in the unlucky case that the coin comes up heads, a $\mathcal{U}$-agent *without* access to its own shutdown button may precommit to taking destructive actions whenever it experiences something suboptimal, so as to cause the shutdown button to be pressed in those cases. Clearly, this is not corrigible behavior.

5 The State of the Shutdown Problem

As per Section 2.1, inappropriate combinations of $\mathcal{U}_{\mathcal{N}}$ and $\mathcal{U}_{\mathcal{S}}$ could give an agent incentives to cause or prevent its own shutdown. Furthermore, as per Section 4, using utility indifference to maintain equal expected utility in both cases does not result in a utility function incentivizing corrigible behavior. In light of this, it seems possible that no simple combination of $\mathcal{U}_{\mathcal{N}}$ and $\mathcal{U}_{\mathcal{S}}$—of the form taken by (11)—can satisfy our desiderata for a corrigible solution to the shutdown problem.

The concerns in Section 4.2 may be fixable. The utility function $\mathcal{U}$ incentivizes agents to act as if the shutdown button cannot be pressed, and incentivizes commitments to press the button in the event of bad luck as if they prevent the bad luck from occurring. This seems reminiscent of behavior prescribed by evidential decision theory, which has been criticized as promoting "an irrational policy of managing the news" (Lewis 1981). The concerns in Section 4.2 could potentially be addressed by some form of counterfactual (rather than conditional) reasoning. We leave a detailed investigation of this possibility to future work.

The concerns illustrated in Section 4.1 seem trickier to fix: the way in which the agent perceives itself as being compensated for actions that destroy utility in the event of shutdown seems fairly central to the balancing technique behind utility indifference.

We observe a tension between Desideratum 4 and Desideratum 5, calling into question how these intuitive desiderata might be formalized. What should it mean for an agent to preserve its shutdown behavior, but maximize $\mathcal{U}_\mathcal{N}$ "otherwise"? What costs (in time and effort) should an agent be willing to expend in order to preserve its shutdown behavior?[7]

Finally, even if we could find some satisfactory method of combining $\mathcal{U}_\mathcal{N}$ and $\mathcal{U}_\mathcal{S}$ into a corrigible solution to the shutdown problem, there is the remaining matter that the specification of $\mathcal{U}_\mathcal{S}$ is itself difficult. Defining what it means for an agent to "shut down" is quite complicated assuming sufficiently intelligent agents. If the agent is running on a single computer and hasn't yet interacted much with the outside world, we can imagine the agent suspending itself to disk and then turning off the power. But what if, for example, the agent has begun the physical construction of a building? Should it stop controlling all its heavy machinery immediately upon receiving a shutdown signal (potentially endangering those nearby)? Should it deconstruct everything that it has started building (requiring some parts of the agent to remain active for hours or days)? Any shutdown policy that requires the agent to dispose of dangerous materials seems vulnerable to what Bostrom (2014) calls "perverse instantiations".

Further solutions may involve abandoning the utility maximization framework entirely, although it is not yet clear what sort of framework could take its place.

In short, a corrigible solution to the shutdown problem does not yet exist, and there is some question about exactly which behaviors should be incentivized. Many open questions remain, and significant research may be necessary in order to attain an understanding of even this small subset of the greater corrigibility problem.

7. We cannot simply claim that it should propagate shutdown behavior "at all costs", as that too would be vulnerable to perverse instantiations wherein an agent would expend significant valuable resources verifying and reverifying that it could shut down if asked.

6 Conclusions

Again, we emphasize that we study the shutdown problem not because we expect to use these techniques to literally install a shutdown button in a physical agent, but rather as toy models through which to gain a better understanding of how to avert undesirable incentives that intelligent agents would experience by default.

Our lack of understanding about how to solve the shutdown problem demonstrates a more general lack of understanding about "corrigible reasoning" and what it entails. It is our hope that a deeper understanding of the shutdown problem will give us insight into the type of reasoning that an agent must use in order to avert manipulation and deception, and be reliably correctable by its programmers.

It seems quite likely that our framework for investigating these issues—in this case, the question of how to combine two separate utility functions $\mathcal{U}_\mathcal{N}$ and $\mathcal{U}_\mathcal{S}$—will look nothing like the framework in which we will eventually represent corrigible reasoning. But whatever framework we do end up using, we expect it will be difficult to prevent the default incentives that an intelligent agent would experience to deceive or manipulate its programmers upon recognizing that its goals differ from theirs. Nevertheless, averting such incentives is crucial if we are to build intelligent systems intended to gain great capability and autonomy.

Before we build generally intelligent systems, we will require some understanding of what it takes to be confident that the system will cooperate with its programmers in addressing aspects of the system that they see as flaws, rather than resisting their efforts or attempting to hide the fact that problems exist. We will all be safer with a formal basis for understanding the desired sort of reasoning.

As demonstrated in this paper, we are still encountering tensions and complexities in formally specifying the desired behaviors and algorithms that will compactly yield them. The field of corrigibility remains wide open, ripe for study, and crucial in the development of safe artificial generally intelligent systems.

References

Armstrong, Stuart. Forthcoming. "AI Motivated Value Selection." Accepted to the 1st International Workshop on AI and Ethics, held within the 29th AAAI Conference on Artificial Intelligence (AAAI-2015), Austin, TX.

Bird, Jon, and Paul Layzell. 2002. "The Evolved Radio and Its Implications for Modelling the Evolution of Novel Sensors." In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02,* 2:1836–1841. Honolulu, HI: IEEE. doi:10.1109/CEC.2002.1004522.

Bostrom, Nick. 2012. “The Superintelligent Will: Motivation and Instrumental Rationality in Advanced Artificial Agents.” In “Theory and Philosophy of AI,” edited by Vincent C. Müller, special issue, *Minds and Machines* 22 (2): 71–85. doi:10.1007/s11023-012-9281-3.

———. 2014. *Superintelligence: Paths, Dangers, Strategies.* New York: Oxford University Press.

Lewis, David. 1981. “Causal Decision Theory.” *Australasian Journal of Philosophy* 59 (1): 5–30. doi:10.1080/00048408112340011.

Omohundro, Stephen M. 2008. “The Basic AI Drives.” In *Artificial General Intelligence 2008: Proceedings of the First AGI Conference,* edited by Pei Wang, Ben Goertzel, and Stan Franklin, 483–492. Frontiers in Artificial Intelligence and Applications 171. Amsterdam: IOS.

Von Neumann, John, and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior.* 1st ed. Princeton, NJ: Princeton University Press.

Yudkowsky, Eliezer. 2008. “Artificial Intelligence as a Positive and Negative Factor in Global Risk.” In *Global Catastrophic Risks,* edited by Nick Bostrom and Milan M. Ćirković, 308–345. New York: Oxford University Press.