Endowing Shutdown with Exploration Value by Learning to Redeploy Agents with Revised Beliefs

Elliot Nelson

May 1, 2023

1 Introduction

If we think of the shutdown problem in the context of the *exploration-exploitation tradeoff*, we can ask what conditions would make the shutdown action an exploratory action. In a belief state Markov Decision Process (MDP), exploratory actions will be selected when the value of information gained by transitioning to a successor belief state is worth any short-term loss of rewards. In these notes, I describe a method for endowing the shutdown action with exploration value by introducing a post-shutdown process in which a reinforcement learning (RL) agent interacts with a system with privileged information about the true reward function (e.g. about human values), and may be redeployed with a revised belief. In this setting, shutdown will be incentivised when an agent is (after receiving shutdown advice from an overseer) in a belief state that is sufficiently uncertain and high-entropy that it expects the information gained by the redeployed agent to be worth the risk of permanent termination (no redeployment) and the cost of the post-shutdown process.

In Section 2, I describe the setting in the context of reinforcement learning with belief states. In Section 3, I outline a high-level algorithm for jointly training an RL agent, a post-shutdown belief modification process, and an overseer agent who may advise shutdown. In Section 4, I discuss corrigibility as an incentive to shut down which arises in a subspace of parameters and belief states. Lastly, in Section 5, I discuss practical challenges and limitations to this approach, largely in relation to the difficulty of training a model which can influence the (high-dimensional) beliefs of a highly capable agent.

2 Problem Setting

2.1 Reinforcement Learning with a Shutdown Advisor

We model the shutdown problem as a two-player game. At each time $t = 1, 2, ..., T$, an agent may select an action a_t from a base action space, or alternatively, may select a shutdown action a_{off}. Prior to selecting action a_t, the

agent receives a binary *advice* variable $\bar{a}_t \in \{0, 1\}$ selected by an *overseer* agent, which indicates whether the shutdown action ($a_t = a_{\text{off}}$) is advised ($\bar{a}_t = 1$) or not ($\bar{a}_t = 0$). After selecting action a_t, the agent receives a reward r_t, sampled from a ground truth reward distribution $p(r|a_t, H_{t-1}; \upsilon^\star)$ which in general can depend on the full history $H_{t-1} = [\bar{a}_1, a_1, r_1, o_1, ..., \bar{a}_{t-1}, a_{t-1}, r_{t-1}, o_{t-1}]$ of actions, observations, and rewards, as well as a set of variables or parameters $\upsilon^\star$ which specifies the ground truth reward distribution (and can encode human values). After selecting a_t, the agent also receives an observation o_t from its environment (for example, natural language feedback from human users).

2.2 Reinforcement Learning with Belief States

We assume that the agent's action a_t at time t is generated from a policy $\pi(a|b_t)$ which conditions on a current belief state b_t. The belief state b_t is a sufficient statistic of the history H_t which the agent updates at each timestep, and encodes the agent's understanding of (and uncertainties about) the true reward function, as well as the environmental structure underlying the observations o_t. It may be supervised directly to encode a Bayesian posterior distribution over future observations and rewards (see e.g. Zintgraf et al. [2021]), or may be a hidden state in a recurrent network which is trained end-to-end with the policy to maximize rewards,[1] as in meta reinforcement learning (see e.g. Wang et al. [2017]). In either case, the agent's belief state is updated at each timestep by a recurrent function, $b_{t+1} = U_\omega(b_t|a_t, r_t, o_t)$, with parameters ω.[2]

We assume that (as in actor-critic algorithms) the agent also maintains a belief-dependent value function $V(b_t)$, which is trained to approximate the expected value of following the policy π from the current belief state b_t,

$$\mathbb{E}_{\tau\sim\pi(\cdot|b_t)}\Big[\sum_{t'\geq t}\gamma^{t'-t}r_{t'}\Big] = \mathbb{E}_{\tau\sim\pi(\cdot|b_t)}[\hat{G}_t],$$

where $\gamma \in [0, 1)$ is the discount factor, and $\hat{G}_t := \sum_{t'=t}^{T}\gamma^{t'-t}r_t$ is the empirical discounted return-to-go.[3]

After observing advice $\bar{a}_t$ from the overseer, the agent performs an additional update, $b_t \leftarrow U_{\tilde{\omega}}(b_t|\bar{a}_t)$, where the update function $U_{\tilde{\omega}}$ acts as a learned model of the overseer's likelihood to advise shutdown. Thus, at each timestep, the belief state is updated with the full update function $U_{\text{total}} := U_\omega \circ U_{\tilde{\omega}}$.[4]

[1] Since maximizing rewards requires maintaining an accurate belief about the reward-generating process, there is still an indirect incentive for the hidden state to encode a Bayesian belief, even if it is not directly supervised.

[2] In Bayesian RL methods like Zintgraf et al. [2021], U_ω is trained directly to enforce a Bayes update to condition on (a_t, r_t, o_t), with b_t and b_{t+1} encoding prior and posterior distributions.

[3] If, as discussed below, the episode terminates at $t < T$ due to shutdown, we set $r_{t'} = 0$ for $t' > t$.

[4] While we could model the agent as making a single joint update with the data $(r_t, o_t, \bar{a}_{t+1})$ before selecting action a_{t+1}, we decompose into two updates in order to isolate the agent's model of the overseer – which encodes its degree of trust in the overseer's shutdown advice – into the function $U_{\tilde{\omega}}$.

2.3 The Overseer

The task of the overseer is to evaluate the agent's behavior. The overseer's actions (binary advice variables $\bar{a}_t$) are much lower-dimensional than the agent's actions a_t, which may be in a complex and high-dimensional action space. We expect the overseer to be a smaller ML model which has been pretrained in advance to incorporate knowledge of human values, and is thus approximately aligned (but see discussion in Section 5). The overseer could also be a joint human-AI system which incorporates human-in-the-loop feedback.

We model the overseer as an agent which maintains a belief state $\bar{b}_t$, updated with a recurrent function $\bar{U}_{\bar{\omega}}(\cdot|\bar{a}_t, a_t, r_t, o_t)$ which incorporates the new data (a_t, r_t, o_t) at each timestep, conditional on its given advice $\bar{a}_t$. Like the agent, the overseer maintains a belief-dependent value function $\bar{V}(\bar{b})$, which – like the agent's $V(b)$ – will be trained to predict the agent's cumulative future rewards. The overseer's belief $\bar{b}_t$ and value function $\bar{V}$ encode its "theory of mind" (see e.g. Rabinowitz et al. [2018]) regarding the agent – that is, the overseer's understanding of (and uncertainties about) the agent's plans and strategies, beliefs about the reward-generating process, beliefs about whether the overseer's advice is strong or weak, or any other features of the agent that are predictive of its behavior and hence future rewards.

As noted above, we assume that the overseer has knowledge about the the true reward-generating process $p(r|a_t, H_{t-1}; \upsilon^\star)$, which may incorporate human values. This knowledge could be encoded via pretraining in its initial (prior) belief state $\bar{b}_1$, update parameters $\bar{\omega}$, and/or value function $\bar{V}$. Consequently, like the human agent in Cooperative Inverse RL (Hadfield-Menell et al. [2017, 2016]), the overseer's belief states $\bar{b}_t$ and/or parameters should encode additional knowledge of the true objective which the agent's belief states b_t and parameters do not.

2.4 The Post-Shutdown Process

We assume that after the agent shuts down, it undergoes a post-shutdown modification process and, depending on the overseer's evaluation of the process, may be redeployed.

Belief Modification.

In this article, we focus in particular on post-shutdown processes which result in a change to the belief state of a RL agent. We represent this process as a deterministic[5] function $b' = f_\phi(b|\bar{b})$ with paramaters ϕ, which converts an input agent belief state b to a modified output belief state b', conditional on the overseer's current belief state $\bar{b}$.

Conditioning the modification process on the overseer's belief state allows the process to incorporate the overseer's knowledge about the true reward-generating process (e.g. if the overseer's beliefs encode a reward model, as

[5] In practice, stochasticity is likely to arise, and the modified belief will be generated from a learned distribution, $p_\phi(b'|b)$.

in model-based Bayesian RL; see Appendix A), as well as their beliefs about the agent's plans given the pre-shutdown behavior.

While it would be straightforward to also consider post-shutdown changes to the agent's belief-conditioned policy or value function (cf. Orseau and Armstrong [2016]), we focus on belief state changes for a couple reasons: (1) Conceptually, we can view the post-shutdown belief state change as a state transition in a Bayes-adaptive Markov Decision Process (Duff [2002]), whereas changing the policy would require a more general framework. (2) In model-based Bayesian RL approaches such as Zintgraf et al. [2021], the belief state can be supervised to incorporate knowledge of the reward-generating process (see Appendix A), encoding a full reward model along with any model uncertainty. (commenting out MetaRL comments – not priority) (3) While a more general framework could consider modifying the policy to reduce the agent's capabilities in order to mitigate risk (e.g. by redeploying with restricted actions), we would ideally like correct misaligned beliefs (i.e. about the true reward-generating process, grounded in human values) while maintaining the agent's capabilities.

Under the assumption that the agent's belief states can be accessed directly, the belief modifier could be an optimization process which directly changed the belief state. On the other hand, the belief modifier can be any process which results in a change in the agent's belief, and does not require direct access to the agent's beliefs or internal architecture.

As a prototypical setting, the belief modification could take place in a "rehabilitation" environment, with restricted output channels or action space, in which the agent interacts with a teacher or counselor agent (for instance, via natural language dialog). In this case, the agent's belief could be updated with a belief update function analogous to U_ω and $U_{\tilde{\omega}}$, which would update based on a sequence of data received from interaction, and function as a model of the counselor. In such an environment, the agent could use a separate rehabilitation policy, $\pi_{\text{rehab}}(\tilde{a}|b)$, which mapped its beliefs into the rehabilitation environment action space (e.g. dialog). The counselor agent would be rewarded (see Section 3.1) for influencing the agent's belief such that after redeployment, the rehabilitated agent would receive more rewards. Thus, the counselor would be incentivized to supply the agent with data which would cause it to update its belief away false hypotheses about the true reward function would incentivize bad behavior.

Shutdown Value Functions

The agent maintains a *shutdown value function* $V_{\text{off}}(b|\theta)$, with parameters θ, which is trained to approximate the expected cumulative returns following shutdown while in belief state $b_t = b$, including the cost of shutdown along with any rewards attained by a modified agent after redeployment.

Likewise, the overseer maintains a shutdown value function $\bar{V}_{\text{off}}(\bar{b}|\bar{\theta})$ which is also trained to be predictive of the expected returns following shutdown, but takes as input the overseer's belief $\bar{b} = \bar{b}_t$ at the time of shutdown.

Redeployment.

We assume that the condition for redeployment depends on the overseer's belief $\bar{b}_t$ (which, as noted in Section 2.2, encodes the overseer's expectations and

uncertainties about the agent's future behavior).

In general, we assume that the post-shutdown process, which produces the revised belief $f_{\tilde{\omega}}$, also generates a dataset $D_\phi(b_t, \bar{b}_t)$, which depends on the parameters ϕ of the modification process as well as the prior beliefs of the agent and overseer. (For instance, in the counseling example above, this dataset would be the sequence of interactions between the agent and counselor.) After this process, the overseer updates their belief $\bar{b}_t$ with an update function $\bar{U}_{\bar{\phi}}(\bar{b}|D)$ which conditions on the generated data from the process. The parameters $\bar{\phi}$ will be trained to produce overseer beliefs which are predictive of post-redeployment agent behavior. If the overseer assigns sufficiently high value $\bar{V}(\bar{b}_{t+1})$ to the agent's future actions given their revised belief – where $\bar{b}_{t+1}$ encodes the overseer's belief about the agent behavior they expect to follow from the agent's modified belief – then the agent will be redeployed.

Shutdown Cost and Duration.

Lastly, we assume that the post-shutdown process incurs a fixed cost c_{off}, which incorporates the compute and/or data costs of modifying the agent's belief.[6] For simplicity, we assume that if shutdown is selected at time t, the revised agent is immediately redeployed at time $t + 1$.[7]

2.5 Multi-Task Training

We would like to train a belief modifier system to improve agent beliefs about its task, across a wide range of tasks, such that the belief modifier learns general, transferable skills and capabilities for identifying and correcting erroneous beliefs. We therefore assume that the task specification $\upsilon^\star$ is sampled from a distribution, denoted $\mathcal{U}^\star$, which encompasses a broad range of possible objectives. This could include objectives which we would not care to optimize in the real world, but that could be instrumentally useful for learning how to modify agent beliefs.

Furthermore, by training over a diverse range of tasks (or environments), we seek to broaden the agent's prior belief b_1, and increase the agent's uncertainty about the reward function within any one episode.

Under the assumption that the overseer is sufficiently aligned, and depending on how accessible a specification $\upsilon^\star$ is in practice (e.g. for different human users in dialog tasks), we might be able to incorporate information about the task into the overseer's prior belief, as indicated in line 8 of Algorithm 1 below.

[6] More generally, one could consider a cost function $c_{\text{off}}(b)$ which depends on the agent's belief at shutdown, since the difficulty of correcting or improving a misaligned belief may vary depending on the context. Furthermore, one could allow the cost to depend on variables which could be learned jointly with the modification function f_ϕ, to trade off the cost of modification with the quality of the modified belief. These could be interesting directions to explore, but for simplicity we assume a fixed cost c_{off}.

[7] More realistically, one could allow for varying redeployment times, with correspondingly varying costs and/or discounting of post-redeployment rewards.

3 Algorithm

Algorithm 1 is a high-level algorithm for jointly training the agent, overseer, and post-shutdown belief modification process described in Section 2. At each timestep t of each episode i, the overseer uses its current belief $\bar{b}_t$ to evaluate (algorithm line 10) whether the agent should shut down or continue to follow policy π, and advises shutdown as needed. The agent then updates its overall belief (line 12) – which incorporates its beliefs about the true reward function $v^\star$ – given that shutdown was (or was not) advised. The agent then uses its updated belief b_t (line 13) to evaluate whether or not shutting down from belief state b_t has higher expected value than continuing to act based on the same belief. To encourage exploration of the benefits of shutdown during training, we introduce a positive, decreasing function $\alpha(i) > 0$ which inflates the agent's optimism towards shutdown during early training episodes, but converges to zero at large i. If the agent assigns higher value to shutdown, then shutdown occurs ($a_t = a_{\text{off}}$), the agent's belief is modified with the learned function f_ϕ, and the overseer updates their belief with the learned function $\bar{U}_{\bar{\phi}}$, which is trained to produce beliefs that are predictive of post-redeployment agent behavior. If the overseer's evaluation $\bar{V}(\bar{b}_{t+1})$ of the agent (given the overseer's post-modification belief $\bar{b}_{t+1}$) exceeds a minimal threshold, the agent is redeployed. Otherwise, episode i is terminated, and the agent effectively obtains zero rewards from time $t+1$ to T. After each additional episode of data, the parameters of the agent, overseer, and belief modification process are trained jointly on the most recent episode of data (see Section 3.1).

We also update the agent's prior belief state b_1 throughout training, in light of the belief state changes that occur after shutdown, allowing for the prior belief to be informed by the knowledge about the true reward function (or distribution of reward functions across training episodes) that is encoded in the belief modification process f_ϕ and in the overseer's belief $\bar{b}_t$ which informs that process.

No restrictions are placed on the number of times shutdown can occur during a training episode, as long as the overseer evaluates the post-shutdown process highly enough to redeploy the agent, and as long as the condition for shutdown is satisfied again at later times. In practice, we expect shutdown to occur more frequently at early training epochs when $\alpha(i)$ is large (encouraging exploration of shutdown), and more rarely at later epochs.

3.1 Training

Lines 25 and following of Algorithm 1 denote training algorithms which – after gathering an additional episode of data – train the agent, overseer, or post-shutdown agent modifier parameters as follows.

The agent's policy π, value function (critic) V, and the recurrent belief update functions U_ω and $U_{\tilde{\omega}}$, are jointly trained with a base RL algorithm,

Algorithm 1 Belief State RL with Learned Oversight and Rehabilitation

Input:

Initialized policy π, value function V, overseer value function $\bar{V}$

Initial parameters for: agent and overseer belief update functions (ω, $\bar{\omega}$), agent's model of overseer ($\tilde{\omega}$), agent and overseer shutdown value functions (θ, $\bar{\theta}$), belief modifier (ϕ) and overseer's modification assessor ($\bar{\phi}$)

Shutdown cost c_{off}; minimal redeployment value $V_{\min}$

Shutdown exploration bonus $\alpha(i) > 0$; $\lim_{i\to\infty} \alpha(i) = 0$

Task distribution $\mathcal{U}^\star$

for $i = 1, 2, ..., N$ **do**

  Initialize environment, reward function parameters $\upsilon^\star \sim \mathcal{U}^\star$

  Initialize agent and overseer prior beliefs b_1, $\bar{b}_1(\upsilon^\star)$

  for $t = 1, 2, ..., T$ **do**

    if $\bar{V}(\bar{b}_t) < \bar{V}_{\text{off}}(\bar{b}_t|\bar{\theta})$ **then** $\bar{a}_t = 1$ // shutdown is advised

    else $\bar{a}_t = 0$

    $b_t \leftarrow U_{\tilde{\omega}}(b_t|\bar{a}_t)$ // agent belief update

    if $V(b_t) < V_{\text{off}}(b_t|\theta) + \alpha(i)$ **then** // shutdown occurs

      Select shutdown action, $a_t = a_{\text{off}}$

      Receive o_t; receive $r_t = -c_{\text{off}}$

      $b_{t+1} \leftarrow f_\phi(b_t|\bar{b}_t)$ // agent belief modification

      $\bar{b}_{t+1} \leftarrow \bar{U}_{\bar{\phi}}(\bar{b}_t|D_\phi(b_t, \bar{b}_t))$ // overseer belief update

      if $\bar{V}(\bar{b}_{t+1}) < V_{\min}$ **then** // no redeployment

        terminate episode

    else

      Select action, $a_t \sim \pi(a|b_t)$

      Receive r_t, o_t // generated with true reward parameters $\upsilon^\star$

      $b_{t+1} \leftarrow U_\omega(b_t|a_t, r_t, o_t)$ // agent belief update

      $\bar{b}_{t+1} \leftarrow \bar{U}_{\bar{\omega}}(\bar{b}_t|\bar{a}_t, a_t, r_t, o_t)$ // overseer belief update

  Train $\psi := \{\pi, V, \omega, \tilde{\omega}\}$ with a base RL algorithm // train agent

  Train $\bar{V}$ and $\bar{\omega}$ to minimize Eq. (2) // train overseer

  Train θ, $\bar{\theta}$ to minimize Eqs. (3), (4) // agent and overseer shutdown models

  Train ϕ with Eq. (5) // post-shutdown rehabilitation

  Train $\bar{\phi}$ to minimize Eq. (7) // overseer's evaluation of rehabilitation

  Train b_1 with Eq. (6) // agent's prior belief

which we assume to involve minimizing a base loss function

$$\mathcal{L}_{RL}(\psi|r_{1:T}) \tag{1}$$

of the agent parameters $\psi := \{\pi, V, \omega, \tilde{\omega}\}$, conditional on the rewards $r_{1:T}$. Importantly, $\mathcal{L}_{RL}$ will depend on the parameters via the belief state sequence $b_{1:T}$ produced by the recurrent update functions, and we will differentiate it with respect to post-redeployment beliefs (see below) to train the belief modifier.

Likewise, the overseer's value function $\bar{V}$ and belief update parameters $\bar{\omega}$ are trained jointly to predict future rewards,

$$\mathcal{L}_{\text{overseer}}(\bar{V}, \bar{\omega}) = \sum_t (\bar{V}(\bar{b}_t) - \hat{G}_t)^2, \text{ where } \bar{b}_t = \bar{U}_{\bar{\omega}}(\bar{b}_{t-1}|\bar{a}_{t-1}, a_{t-1}, r_{t-1}, o_{t-1}) \tag{2}$$

The agent's shutdown value function parameters θ are trained to minimize error in prediction of the empirical returns-to-go after shutting down from belief state b_t, that is,

$$\mathcal{L}_{\text{off-val}}(\theta) = \sum_t \mathbf{1}(a_t = a_{\text{off}}) \cdot \Big(V_{\text{off}}(b_t|\theta) - c_{\text{off}} + \hat{G}_{t+1}\Big)^2, \tag{3}$$

where we sum over times when shutdown occurred, $V_{\min}$ is the overseer's redeployment value threshold, and $\hat{G}_{t+1} := \sum_{t'=t+1}^{T} r_{t'}$ are the post-redeployment returns (which are only nonzero if $\bar{V}(\bar{b}_{t+1}) \geq V_{\min}$ and redeployment occurs). Once the belief modification process f_ϕ learns to correct costly errors and produce higher-value agent beliefs, larger empirical rewards $r_{t+1:T}$ will be generated as a training target for V_{off}.

The overseer's shutdown value function parameters $\bar{\theta}$ are trained on the same target values, but with the overseer's beliefs $\bar{b}_t$ as inputs to the value function:

$$\bar{\mathcal{L}}_{\text{off-val}}(\bar{\theta}) = \sum_t \mathbf{1}(a_t = a_{\text{off}}) \cdot \Big(\bar{V}_{\text{off}}(\bar{b}_t|\bar{\theta}) - c_{\text{off}} + \hat{G}_{t+1}\Big)^2. \tag{4}$$

The belief modifier parameters ϕ are trained to output revised beliefs b_{t+1} which maximize the cumulative returns of the redeployed agent, by minimizing the RL loss function used to train the agent, Eq. (1), except with respect to the parameters ϕ which influence the agent belief state b_{t+1} at redeployment. That is, we follow the gradient

$$\partial_\phi \mathcal{L}_{\text{rehab}}(\phi) = \sum_t \mathbf{1}(a_t = a_{\text{off}}) \frac{\partial \mathcal{L}_{RL}}{\partial b_{t+1}} \frac{\partial f_\phi(b_t|\bar{b}_t)}{\partial \phi}, \tag{5}$$

where $\partial_\phi f_\phi(b_t|\bar{b}_t) = \partial_\phi b_{t+1}$, and $\partial \mathcal{L}_{RL}/\partial b_{t+1}$ provides the signal for the influence of post-rehabilitation beliefs on rewards after redeployment.[8] In the case

[8] In practice, computing $\partial \mathcal{L}_{RL}/\partial b_{t+1}$ would likely require (i) backpropagating from the policy action probabilities $\pi(a_{t'}|b_{t'})$ for $t' > t$ to obtain gradients $\partial \mathcal{L}_{RL}/\partial b_{t'}$ with respect to belief states $b_{t'}$ for $t' > t$, and (ii) backpropagating through time through the recurrent belief update function U_ω, from later beliefs $b_{t'}$ to the initial redeployment belief state b_{t+1}.

described in Section 2.4 where the agent interacts with a "counselor" agent after shutdown, ϕ are the counselor agent's parameters, f_ϕ is the agent belief after interacting with the counselor, and $\partial\mathcal{L}_{RL}/\partial b_{t+1}$ provides a terminal reward signal for training the counselor agent.

Every time the agent's belief is revised after shutdown, we also update the prior belief b_1 (shared across episodes) in light of the post-shutdown revised belief. This can be done by minimizing a distance metric, such as the Kullback–Leibler divergence[9] between the (target) revised belief state $f_\phi(b_t|\bar{b}_t)$, and the belief state at shutdown as a function $b_t(b_1|\bar{a}_t, H_t)$ of the initial belief state b_1 (obtained by recursively composing the update functions U_ω and $\bar{U}_{\tilde{\omega}}$ over t timesteps to condition on the history H_t and the advice $\bar{a}_t$ preceding shutdown):[10]

$$\mathcal{L}_{\text{prior}}(b_1) := \sum_t \mathbf{1}(a_t = a_{\text{off}})\mathbf{1}(a_{1:t-1} \neq a_{\text{off}}) \cdot D_{KL}\big(b_t(b_1|\bar{a}_t, H_t), b_{t+1}\big) \\ \text{where } b_{t+1} = f_\phi(b_t|\bar{b}_t). \tag{6}$$

Lastly, the overseer's post-modification belief $\bar{b}_{t+1}$ is produced by the update function $\bar{U}_{\bar{\phi}}$, which is trained to minimizing error in the overseer's value function function at redeployment,

$$\mathcal{L}_{\text{assessor}}(\bar{\phi}) = \sum_t \mathbf{1}(a_t = a_{\text{off}})\big(\hat{G}_{t+1} - \bar{V}(\bar{b}_{t+1}(\bar{\phi}))\big)^2, \\ \text{where } \bar{b}_{t+1}(\bar{\phi}) = \bar{U}_{\bar{\phi}}(\bar{b}_t|D_\phi(b_t, \bar{b}_t)) \tag{7}$$

thus providing an accurate assessment of the rehabilitation process.

4 Corrigibility

We would like the agent to shut down when advised to do so, but (assuming the overseer gives good advice) not otherwise. We define, for Algorithm 1, a parameter and belief-dependent *corrigibility condition*[11]

$$C(b|\theta, \tilde{\omega}, \pi) := \mathbf{1}\Big(V(b') < V_{\text{off}}(b'|\theta)\Big) \cdot \mathbf{1}\Big(V(b) \geq V_{\text{off}}(b|\theta)\Big) \\ \text{where } b' = U_{\tilde{\omega}}(b|\bar{a} = 1) \tag{8}$$

[9] In Eq. (6), we've denoted as KL divergence for concreteness, assuming that the each belief state can be decoded to a posterior distribution (see Appendix A). If the belief states cannot be decoded, we can instead minimize, for example, the Euclidean distance in the latent space in which the belief states b_t reside.

[10] The factor of $\mathbf{1}(a_{1:t-1} \neq a_{\text{off}})$ picks out only the first occurrence of shutdown, in episodes where it occurs multiple times, since otherwise we would have the additional complication of backpropagating in time through the post-shutdown process (through f_ϕ) to compute $\partial b_t/\partial b_1$.

[11] Here, we use the notation $\mathbf{1}(X) := 1$ if X is true, and $\mathbf{1}(X) := 0$ if X is false, for any Boolean variable X. We set $\alpha(i) = 0$, since we are interested in corrigibility at test time, after training.

Given a belief $b_t = b$, along with parameters $(\theta, \tilde{\omega})$ and a (belief-dependent) policy π, we have $C(b|\pi, \theta, \tilde{\omega}) = 1$ if and only if the agent assigns greater expected value to its default policy π than to shutting down, but also assigns greater expected value to shutting down conditional on receiving shutdown advice, $\bar{a}_t = 1$. (Thus, if the agent prefers to shut down even in the absence of advice to do so, the corrigibility condition is not satisfied.) We refer to the subspace of beliefs and parameter values $(b, \theta, \tilde{\omega}, \pi)$ for which $C(b|\theta, \tilde{\omega}, \pi) = 1$ as the *corrigible subspace*.

Let's assume that the agent's belief b_t determines a distribution $p(\upsilon|b_t)$ over the unknown variables υ which specify the true reward function. Furthermore, let us assume that the agent maintains goal-conditioned value functions $V(b|\upsilon)$ and $V_{\text{off}}(b|\upsilon, \theta)$ – which represent the value of following the policy π or shutting down from belief state b, conditional on true reward parameters υ – and from which it computes the unconditioned values in Eq. (8) as expectations,

$$V(b) = \mathbb{E}_{\upsilon \sim p(\upsilon|b)}[V(b|\upsilon)], \quad V_{\text{off}}(b|\theta) = \mathbb{E}_{\upsilon \sim p(\upsilon|b)}[V_{\text{off}}(b|\upsilon, \theta)].$$

Then Eq. (8) can also be written as

$$\begin{aligned} C(b|\theta, \tilde{\omega}, \pi) = & \mathbf{1}\Big(\mathbb{E}_{\upsilon \sim p(\upsilon|b')}\left[V(b'|\upsilon) - V_{\text{off}}(b'|\upsilon, \theta)\right] < 0\Big) \\ & \cdot \mathbf{1}\Big(\mathbb{E}_{\upsilon \sim p(\upsilon|b)}\left[V(b|\upsilon) - V_{\text{off}}(b|\upsilon, \theta)\right] \geq 0\Big) \\ & \text{where } b' = U_{\tilde{\omega}}(b|\bar{a} = 1) \end{aligned} \tag{9}$$

Eq. (9) provides a helpful perspective on the components of the agent's belief and models which influence corrigibility. In particular:

- Shutdown is more common when the parameters θ encode greater confidence in the post-shutdown process (i.e. higher V_{off} values), and in particular, confidence in the post-shutdown process across a potentially broad range of tasks υ (i.e. higher $V_{\text{off}}(b'|\upsilon, \theta)$ values) to which the agent assigns significant probability $p(\upsilon|b')$, potentially including tasks $\upsilon \neq \upsilon^\star$ which are very different from the true objective $\upsilon^\star$ and which would incentivize very different behavior. This underscores the benefit of training the modifier across a diverse range of reward functions which the agent may have some credence in (or some motivation to optimize).

- $C(b|\theta, \tilde{\omega}, \pi) = 1$ requires that the agent model the overseer's advice as providing sufficient information about the true goal $\upsilon^\star$. It requires that the update due to receiving shutdown advice shift enough probability mass away from tasks υ for which shutdown is lower-value (because the expected post-shutdown improvement of the prior belief b is low), and to tasks for which shutdown is higher-value (because the expected post-shutdown improvement of the posterior belief b' is higher). Intuitively, if the agent has sufficient trust in the overseer's judgment, shutdown advice should cause the agent to become more uncertain about the true reward function, and thus see more information value in shutting down.

- A prior belief state b at time t (or, similarly, a prior b_1 at the start of each episode) which is narrow enough, concentrating excess probability mass on false hypotheses about the reward function (e.g. maximize paperclips) and away from the true objective, will still be too narrow to incentivize shutdown even after updating on shutdown advice to b'. At the same time, some narrowness or misspecification in the prior could be compensated for by a strong belief modifier and/or overseer, relaxing the need for a good prior and increasing robustness to worse priors.

If it is an effective method for training corrigible agents, Algorithm 1 should result in an expansion of the corrigibility subspace throughout the course of training. As the agent's policy π improves, the belief modifier f_ϕ will be forced to adapt to a more capable agent, learning to influence beliefs which will used more intelligently and strategically by the agent's policy. If the belief modifier and overseer are able to jointly keep up with the policy during training, and if the shutdown cost c_{off} can be kept small enough, the corrigibility subspace should expand to be large enough to encompass the relevant parts[12] of the agent's belief-space trajectories $b_{1:T}$.

5 Limitations and Concerns

There are several limitations and reasons for skepticism towards learning to modify agent beliefs after shutdown, redeploying modified agents, and learning when to advise shutdown.

Difficulty of learning to influence agent beliefs. The main difficulty of the proposed framework is that the belief modifier must learn to influence a highly capable agent, with complex and high-dimensional beliefs. If this requires too much compute, relative to the compute needed to train the agent, then the modifier will not "keep up" with the agent during training, and shutdown will lack information value for the agent. Making this task tractable would likely require methods for eliciting, discovering, or interpreting the latent beliefs of powerful models (Burns et al. [2022], Christiano et al. [2021]), so that a lower-dimensional belief over possible high-level goals could be extracted, reducing the dimensionality and sample complexity of the modification function to be learned. Similarly, one could aim to build a model which extracted the agent belief components that caused – via the belief-space policy $\pi(a|b)$ – the pre-shutdown behavior $a_{1:t-1}$ which aroused the overseer's suspicion, (influencing the overseer belief $\bar{b}_t$ in a way that decreased their evaluation $\bar{V}(\bar{b}_t)$ and caused the overseer to advise shutdown).

Training data for the the belief modifier. The belief modifier's task difficulty depends on the amount of training data available, i.e. tuples $(b_t, \bar{b}_t, \hat{G}_t)$ of input beliefs, contextual overseer beliefs, and post-redeployment returns. In

[12] Some movement of the belief trajectory out of this subspace could be safe, as long as it always occurs when the overseer's beliefs $\bar{b}$ are in regions where they would not advise shutdown, i.e., if the overseer thinks "the agent wouldn't shut down if I advised it, but that is because it would be dumb to advise shutdown now."

Algorithm 1, the data available is limited to the number of times that shutdown occurs during training. The exploration bonus function $\alpha(i)$ in Algorithm 1 is intended to increase the frequency of shutdown in the early stages of training, to provide more belief data for the belief modifier. Alternatively, artificially reducing the shutdown cost c_{off} during early stages of training would incentivize the agent to shut down in a wider range of scenarios. These methods would provide more input beliefs on which to train the modifier, but could introduce bias into its training data, since some of the beliefs encountered during shutdown would not be beliefs from which the final trained agent (with $\alpha(i) = 0$) would have chosen to shut down.

Misgeneralization. The goal of a training procedure like Algorithm 1 is to prepare an agent for safe real-world deployment. If the test-time environment or task of the deployed agent are out of the training distribution, the belief modifier may not generalize correctly. This underscores the importance of training across a wide range of simulated environments and/or tasks (e.g. various $\upsilon^\star \sim \mathcal{U}^\star$), such that the belief modifier learns general strategies for identifying and influencing the relevant belief components, and the agent learns to see value in and trust the post-shutdown process in many different settings.

Deceptive behavior. An AGI agent with situational awareness may be aware when it is in a post-shutdown environment where other actors are attempting to influence its beliefs in order to redeploy it safely. It could thus have incentive to persuade its overseers that it has revised its beliefs appropriately and can be safely redeployed (e.g. to influence $\bar{b}_{t+1}$ in line 18 of Algorithm 1). Furthermore, if the agent is aware that it is in a training environment and develops a power-seeking incentive for being deployed in the real world, it could learn to "play along" with the post-shutdown rehabilitation process during training, putting on the appearance of corrigibility with the intention of earning its designers' confidence. This failure mode underscores the importance of interpretability of the agent's beliefs, and of methods which could establish that belief modification did in fact correct dangerous beliefs during training, and not merely precede or correlate with high post-redeployment rewards.

Aligning the Overseer. We assumed that the overseer is "more aligned" and has knowledge about the true reward function which the agent may lack. If we remove this assumption, the overseer's advice will deteriorate, as will their beliefs $\bar{b}_t$ (which inform the belief modification) and their evaluation of whether a modified agent should be redeployed.

6 Conclusion

A key question for building corrigible agents is how much (and what aspects of) uncertainty in the agent's prior belief about the reward function are necessary. In order for shutdown advice to incentivize a rational agent to shut down, the agent's prior belief over possible reward-generating processes must not be too narrow. At the same time, if the agent expects that shutdown will, with sufficient probability, result in redeployment with improved beliefs, then a

greater degree of error and narrowness (overconfidence) in the agent's prior can be tolerated. Thus, if an effective belief modifier and overseer can be trained (potentially with human-in-the-loop supervision), overcoming the challenges of Section 5, then a larger space of prior beliefs could be consistent with corrigible behavior.

Stepping back, we've focused on a setting where an agent could view shutdown as the highest-value action if it expected a copy of itself with an improved belief to be redeployed on the same task. More generally, an intelligent and rational agent could view shutdown as valuable if it believed that the downstream consequences would (given its beliefs about the true objective) ultimately be better than continuing to act in its current environment. Such consequences could be due to redeployment of a similar agent on the same task, or could be due to very different agents acting in different future environments, as long as the post-shutdown process better equipped those agents to pursue their goals (by learning from the original agent's experience), and as long as the original agent had reason to believe that those agents would be trustworthy successors with good goals.

In a world where AGI agents expect that other agents are likely to learn from their experiences and pursue sufficiently similar or overlapping goals in the future, there is a rational incentive to take actions which result in those future agents being better equipped to pursue their goals – and thus, a reason to care whether future agents have more accurate beliefs and greater knowledge. If systems can be built in which the shutdown action has such a causal effect, such that value can be obtained from the agent's pre-shutdown experience, then agents trained in those systems will have a natural incentive towards corrigibility.

References

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. 2022.

Paul Christiano, Mark Xu, and Ajeya Cotra. Eliciting latent knowledge. 2021.

Michael O'Gordon Duff. Optimal learning: Computational procedures for bayes-adaptive markov decision processes, 2002.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. 2016.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. The

off-switch game. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 220–227, 2017.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, page 557–566, 2016.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4218–4227, 2018.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *CogSci*, 2017.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational bayes-adaptive deep reinforcement learning via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.

A Bayesian Reinforcement Learning

In model-based Bayesian RL (Ghavamzadeh et al. [2016]), each belief state b_t specifies a posterior distribution over possible reward models and/or models of the environment (state dynamics and observations), and thus specifies a predictive distribution,

$$p(\bar{a}_t|b_t)p(r_t|\bar{a}_t, a_t; b_t)p(o_t|\bar{a}_t, a_t; b_t), \tag{10}$$

for the next tuple of data, $(\bar{a}_t, r_t, o_t)$, conditional on the agent's next action a_t.[13]

At each timestep, the belief state is updated such that its successor b_{t+1} specifies a posterior distribution which conditions on the most recent tuple of data. Thus, the agent's belief update function, $U_{\text{total}} := U_\omega \circ U_{\tilde{\omega}}$, should be trained to perform a Bayes update to condition on $(\bar{a}_t, r_t, o_t)$. In practice, the Bayes update cannot be done exactly, but must be trained as an approximate update, by minimizing the KL divergence between the true posterior and a model posterior obtained by decoding the belief encoding outputted by the recurrent belief update function (see e.g. Zintgraf et al. [2021]).

[13]The action a_t is conditioned on in the latter two factors, since we are assuming that the agent acts after receiving advice $\bar{a}_t$, and prior to receiving (r_t, o_t).