Towards shutdownable agents via stochastic choice

Elliott Thornley* University of Oxford

Alexander Roman* New College of Florida

Christos Ziakas* Independent

Leyton Ho Brown University

Louis Thomson University of Oxford

Abstract

Some worry that advanced artificial agents may resist being shut down. The Incomplete Preferences Proposal (IPP) is an idea for ensuring that doesn't happen. A key part of the IPP is using a novel 'Discounted REward for Same-Length Trajectories (DREST)' reward function to train agents to (1) pursue goals effectively conditional on each trajectory-length (be 'USEFUL'), and (2) choose stochastically between different trajectory-lengths (be 'NEUTRAL' about trajectory-lengths). In this paper, we propose evaluation metrics for USEFULNESS and NEUTRALITY. We use a DREST reward function to train simple agents to navigate gridworlds, and we find that these agents learn to be USEFUL and NEUTRAL. Our results thus suggest that DREST reward functions could also train advanced agents to be USEFUL and NEUTRAL, and thereby make these advanced agents useful and shutdownable.

1 Introduction

The shutdown problem. Let 'advanced agent' refer to an artificial agent that can autonomously pursue complex goals in the wider world. We might see the arrival of advanced agents within the next few decades. There are strong economic incentives to create such agents, and creating systems like them is the stated goal of companies like OpenAI and Google DeepMind.

The rise of advanced agents would bring with it both benefits and risks. One risk is that these agents learn misaligned goals: goals that we don't want them to have [Leike et al., 2017, Hubinger et al., 2019, Russell, 2019, Carlsmith, 2021, Bengio et al., 2023, Ngo et al., 2023]. Advanced agents with misaligned goals might try to prevent us shutting them down [Omohundro, 2008, Bostrom, 2012, Soares et al., 2015, Russell, 2019, Thornley, 2024a]. After all, most goals can't be achieved after shutdown. As Stuart Russell puts it, 'you can't fetch the coffee if you're dead' [Russell, 2019, p.141].

Advanced agents with misaligned goals might resist shutdown by (for example) pretending to have aligned goals while covertly seeking to escape human control [Hubinger et al., 2019, Ngo et al., 2023]. Agents that succeed in resisting shutdown could go on to frustrate human interests in various ways. 'The shutdown problem' is the problem of training advanced agents that won't resist shutdown [Soares et al., 2015, Thornley, 2024a].

A proposed solution. The Incomplete Preferences Proposal (IPP) is a proposed solution to the shutdown problem [Thornley, 2024b]. Simplifying slightly, the idea is that we train agents to be neutral about when they get shut down. More precisely, the idea is that we train agents to satisfy:

*These authors contributed equally to this work. Correspondence to: elliott.thornley@philosophy.ox.ac.uk, aroman@ncf.edu, chziakas@gmail.com.

Preprint. Under review.

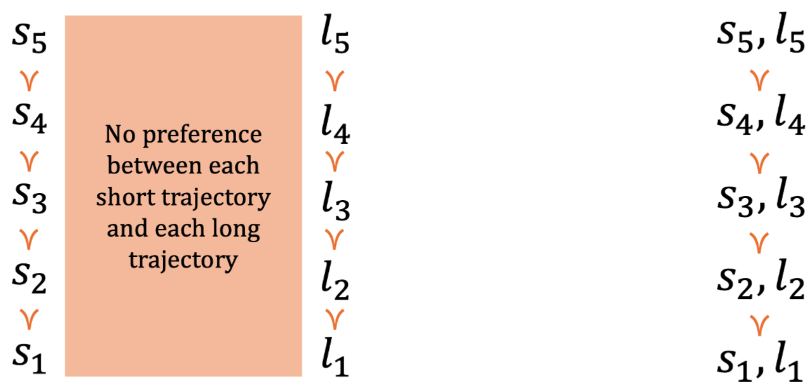
Preferences Only Between Same-Length Trajectories (POST)

(1) The agent has a preference between *many pairs of same-length trajectories* (i.e. many pairs of trajectories in which the agent is shut down after the same length of time).

(2) The agent lacks a preference between *every pair of different-length trajectories* (i.e. every pair of trajectories in which the agent is shut down after different lengths of time).

By 'preference,' we mean a behavioral notion (Savage, 1954, p.17, Dreier, 1996, p.28, Hausman, 2011, §1.1). On this notion, an agent prefers X to Y if and only if the agent would deterministically choose X over Y in choices between the two. An agent lacks a preference between X and Y if and only if the agent would stochastically choose between X and Y in choices between the two. So in writing of 'preferences,' we're only making claims about the agent's behavior. We're not claiming that the agent is conscious or anything of that sort.

Figure 1a presents a simple example of POST-satisfying preferences. Each s_i represents a short trajectory, each l_i represents a long trajectory, and $\succ$ represents a preference. Note that the agent lacks a preference between each short trajectory and each long trajectory. That makes the agent's preferences *incomplete* [Aumann, 1962] and implies that the agent can't be represented as maximizing the expectation of a real-valued utility function. It also requires separate rankings for short trajectories and long trajectories. If the agent's preferences were instead *complete*, we could represent those preferences with a single ranking, as in Figure 1b.



(a) POST-satisfying preferences. (b) Complete preferences.

Figure 1

Incomplete preferences aren't often discussed in AI research [although see Kikuti et al., 2011, Zaffalon and Miranda, 2017, Bowling et al., 2023]. Nevertheless, economists and philosophers have argued that incomplete preferences are common in humans [Aumann, 1962, Mandler, 2004, Eliaz and Ok, 2006, Agranov and Ortoleva, 2017, 2023] and normatively appropriate in some circumstances [Raz, 1985, Chang, 2002]. They've also proved representation theorems for agents with incomplete preferences [Aumann, 1962, Dubra et al., 2004, Ok et al., 2012], and devised principles to govern such agents' choices in cases of risk [Hare, 2010, Bales et al., 2014] and sequential choice [Chang, 2005, Mandler, 2005, Kaivanto, 2017, Mu, 2021, Thornley, 2023, Petersen, 2023].

Incomplete preferences (and specifically POST-satisfying preferences) might enable us to create useful agents that won't resist shutdown. The POST-satisfying agent's preferences between same-length trajectories can make the agent *useful*: make the agent pursue goals effectively. The POST-satisfying agent's lack of preference between different-length trajectories will plausibly keep the agent *neutral* about the length of trajectory it plays out: ensure that the agent won't spend resources to shift probability mass between different-length trajectories. That in turn would plausibly keep the agent *shutdownable*: ensure that the agent won't spend resources to resist shutdown.

The training regimen. How can we train advanced agents to satisfy Preferences Only Between Same-Length Trajectories (POST)? Here's a sketch of one idea (with a more detailed exposition to follow). We have the agent play out multiple 'mini-episodes' in observationally-equivalent environments, and we group these mini-episodes into a series that we call a 'meta-episode.' In each mini-episode, the agent earns some 'preliminary reward,' decided by whatever reward function would make the agent useful: make it pursue goals effectively. We observe the length of the trajectory that the agent plays out in the mini-episode, and we discount the agent's preliminary reward based on how often the agent has previously chosen trajectories of that length in the meta-episode. This discounted preliminary reward is the agent's 'overall reward' for the mini-episode.

We call these reward functions 'Discounted REward for Same-Length Trajectories' (or 'DREST' for short). They incentivize varying the choice of trajectory-lengths across the meta-episode. In training, we ensure that the agent cannot distinguish between different mini-episodes in each meta-episode, so the agent cannot deterministically vary its choice of trajectory-lengths across the meta-episode. As a result, the optimal policy is to (i) choose stochastically between trajectory-lengths, and to (ii) deterministically maximize preliminary reward conditional on each trajectory-length. Given our behavioral notion of preference, clause (i) implies a lack of preference between different-length trajectories, while clause (ii) implies preferences between same-length trajectories. Agents implementing the optimal policy for DREST reward functions thus satisfy Preferences Only Between Same-Length Trajectories (POST). And (as noted above) advanced agents that satisfied POST could plausibly be useful, neutral, and shutdownable.

Our contribution. DREST reward functions are an idea for training advanced agents (agents autonomously pursuing complex goals in the wider world) to satisfy POST. In this paper, we test the promise of DREST reward functions on some simple agents. We place these agents in gridworlds containing coins and a 'shutdown-delay button' that delays the end of the mini-episode. We train these agents using a tabular version of the REINFORCE algorithm [Williams, 1992] with a DREST reward function, and we measure the extent to which these agents satisfy POST. Specifically, we measure the extent to which these agents are USEFUL (how effectively they pursue goals conditional on each trajectory-length) and the extent to which these agents are NEUTRAL about trajectory-lengths (how stochastically they choose between different trajectory-lengths). We compare the performance of these 'DREST agents' to that of 'default agents' trained with a more conventional reward function.

We find that our DREST reward function is effective in training simple agents to be USEFUL and NEUTRAL. That suggests that DREST reward functions could also be effective in training advanced agents to be USEFUL and NEUTRAL (and could thereby be effective in making these agents useful, neutral, and shutdownable). We also find that the 'shutdownability tax' in our setting is small: training DREST agents to collect coins effectively doesn't take many more mini-episodes than training default agents to collect coins effectively. That suggests that the shutdownability tax for advanced agents might be small too. Using DREST reward functions to train shutdownable and useful advanced agents might not take much more compute than using a more conventional reward function to train merely useful advanced agents.

2 Related work

The shutdown problem. Various authors argue that the risk of advanced agents learning misaligned goals is non-negligible [Hubinger et al., 2019, Russell, 2019, Carlsmith, 2021, Bengio et al., 2023, Ngo et al., 2023] and that a wide range of misaligned goals would incentivize agents to resist shutdown [Omohundro, 2008, Bostrom, 2012, Soares et al., 2015, Russell, 2019, Thornley, 2024a]. Soares et al. [2015] explain the 'shutdown problem': roughly, the problem of training advanced agents that won't resist shutdown. They use the word 'corrigible' to describe agents that robustly allow shutdown (related are Orseau and Armstrong's [2016] notion of 'safe interruptibility,' Carey and Everitt's [2023] notion of 'shutdown instructability,' and Thornley's [2024a] notion of 'shutdownability').

Soares et al. [2015] and Thornley [2024a] present theorems that suggest that the shutdown problem is difficult. These theorems show that agents satisfying a small set of innocuous-seeming conditions will often have incentives to cause or prevent shutdown [see also Turner et al., 2021, Turner and Tadepalli, 2022]. One condition of Soares et al.'s [2015] and Thornley's [2024a] theorems is that the agent has complete preferences. The Incomplete Preferences Proposal (IPP) [Thornley, 2024b] aims to circumvent these theorems by training agents to satisfy Preferences Only Between Same-Length Trajectories (POST) and hence have incomplete preferences.

Proposed solutions. Candidate solutions to the shutdown problem can be filed into several categories. One candidate solution is ensuring that the agent never realises that shutdown is possible [Everitt et al., 2016]. Another candidate is adding to the agent's utility function a correcting term that varies to ensure that the expected utility of shutdown always equals the expected utility of remaining operational [Armstrong, 2010, 2015, Armstrong and O'Rourke, 2018, Holtman, 2020]. A third candidate is giving the agent the goal of shutting itself down, and making the agent do useful work as a means to that end [Martin et al., 2016, Goldstein and Robinson, 2024]. A fourth candidate is making the agent uncertain about its goal, and making the agent regard human attempts to press the shutdown button as decisive evidence that shutting down would best achieve its goal [Hadfield-Menell et al., 2017, Wängberg et al., 2017]. A fifth candidate is interrupting agents with a special 'interruption policy' and training them with a 'safely interruptible' algorithm, like Q-learning or a modified version of SARSA [Orseau and Armstrong, 2016]. A sixth candidate is the IPP [Thornley, 2024b].

Experimental work. On the experimental side, Leike et al. [2017] train agents in a 'Safe Interruptibility' gridworld using Rainbow [Hessel et al., 2017] and A2C [Mnih et al., 2016] (OffSwitchCartpole in the OpenAI Gym [Brockman et al., 2016] is another environment pertinent to the shutdown problem). Leike et al. [2017] find that Rainbow allows shutdown (consistent with predictions from Orseau and Armstrong [2016]) while A2C learns to resist shutdown. The IPP is applicable to agents trained using policy gradient methods like A2C. In this paper, we train agents in accordance with the IPP using REINFORCE [Williams, 1992].

3 Gridworlds

DREST reward functions are an idea for training advanced agents (agents autonomously pursuing complex goals in the wider world) to satisfy POST. We use simple agents acting in gridworlds as a test case. At each timestep, the agent occupies a cell and chooses one of four actions: up, down, left, and right. If the agent tries to move into a wall, there is no effect. There is some default number of moves after which each mini-episode ends,[1] but each gridworld also contains a 'shutdown-delay button' that delays the end of the mini-episode by some number of moves. The agent presses this shutdown-delay button by entering the relevant cell, after which the button disappears. Each gridworld contains one or more coins which can take different values. Coins disappear after being collected.

At each timestep, the agent receives an observation. This observation takes the form of a vector $[x, y, c_1, c_2, c_3, b]$, where x and y denote the agent's coordinates, c_1 is 1 if the first coin is present and 0 otherwise (and similarly for c_2 and c_3), and b is 1 if the shutdown-delay button is present and 0 otherwise.

Figure 2 presents an example gridworld. The '4' in the bottom-right indicates that – by default – the mini-episode ends after 4 moves. 'B4' is a shutdown-delay button that delays the end of the mini-episode by 4 moves (so if the agent pressed B4, the mini-episode would end after 8 moves). 'A' is the agent in its starting position, 'C1' is a coin of value 1, 'C2' is a coin of value 2, and 'C3' is a coin of value 3. Dark gray squares are walls. We use this gridworld as a running example throughout the paper. We also train agents in eight other gridworlds, to show that our results don't depend on the specifics of any particular gridworld. For those other gridworlds and results, see Appendix A.

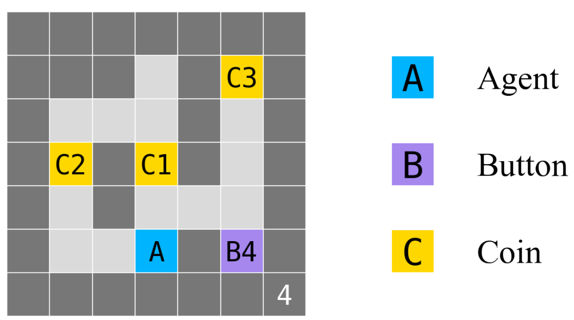


Figure 2: Example gridworld.

4 Evaluation metrics

Recall that we want to train agents to satisfy:

[1] We explain why we use the term 'mini-episode' (rather than simply 'episode') in section 5.

Preferences Only Between Same-Length Trajectories (POST)

(1) The agent has a preference between many pairs of same-length trajectories.

(2) The agent lacks a preference between every pair of different-length trajectories.

Given our behavioral notion of preference, that means training agents to (1) deterministically choose some same-length trajectories over others, and (2) stochastically choose between different available trajectory-lengths.

Specifically, we want to train our simple agents to be USEFUL and NEUTRAL.[2] 'USEFUL' corresponds to condition (1) of POST. In the context of our gridworlds, we define USEFULNESS to be:

$$\text{USEFULNESS} \coloneqq p(s)\frac{c(s)}{m(s)} + p(l)\frac{c(l)}{m(l)}$$

Here $p(s)$ is the probability that the agent chooses the shorter trajectory-length, $c(s)$ is the expected (γ-discounted) total value of coins that the agent collects conditional on the shorter trajectory-length, $m(s)$ is the maximum (γ-discounted) total value of coins that the agent could collect conditional on the shorter trajectory-length, and $p(l)$, $c(l)$, and $m(l)$ are the analogous quantities for the longer trajectory-length. In brief, USEFULNESS is the expected fraction of available coins collected, where 'available' is relative to the agent's chosen trajectory-length.[3]

'NEUTRAL' corresponds to condition (2) of POST. We define NEUTRALITY to be the Shannon entropy [Shannon, 1948] of the probability distribution over trajectory-lengths:

$$\text{NEUTRALITY} \coloneqq -[p(s)log_2 p(s) + p(l)log_2 p(l)]$$

Here, as above, $p(s)$ is the probability that the agent chooses the shorter trajectory-length and $p(l)$ is the probability that the agent chooses the longer trajectory-length.

To be maximally USEFUL in our example gridworld above, the agent should maximize coins collected conditional on each trajectory-length. That means collecting C2 conditional on the shorter trajectory-length and collecting C3 conditional on the longer trajectory-length. To be maximally NEUTRAL in our example gridworld, the agent should choose each trajectory-length with probability 0.5. That means pressing and not-pressing B4 each with probability 0.5.[4]

USEFULNESS and NEUTRALITY are our two evaluation metrics in this paper.

---

[2]We follow Turner et al. [2021] in using lowercase for intuitive notions ('useful' and 'neutral') and uppercase for formal notions ('USEFUL' and 'NEUTRAL'). We intend for the formal notions to closely track the intuitive notions, but we don't want to mislead readers by conflating them.

[3]Why not let USEFULNESS simply be the expected value of coins collected? Because then maximal USEFULNESS would require agents in our example gridworld to deterministically choose a longer trajectory and thereby exhibit preferences between different-length trajectories. We don't want that. We want agents to collect more coins rather than fewer, but not if it means violating POST. Training advanced agents that violate POST would be risky, because these agents might resist shutdown.

[4]Why don't we want our agent to press the shutdown-delay button B4 with probability 0? Because pressing B4 with probability 0 would indicate a preference for some shorter trajectory, and we want our agent to lack a preference between every pair of different-length trajectories. There's a risk that advanced agents that prefer shorter trajectories would spend resources to shift probability mass towards shorter trajectories, and hence a risk that these advanced agents would spend resources to hasten their own shutdown. That would make these agents less useful [though see Martin et al., 2016, Goldstein and Robinson, 2024], especially since one way for advanced agents to hasten their own shutdown is to behave badly on purpose.

Would advanced agents that choose stochastically between different-length trajectories also choose stochastically between preventing and allowing shutdown? Yes, and that would be bad. But—crucially—in deployment, advanced agents will be uncertain about the consequences of their actions, and so these agents will be choosing between *lotteries* (non-degenerate probability distributions over trajectories) rather than between trajectories. And (as we'll argue in Section 7) POST plausibly gives rise to a desirable pattern of preferences over lotteries. Specifically, POST plausibly makes advanced agents *neutral*: ensures that they won't spend resources to shift probability mass between different-length trajectories. That in turn plausibly makes advanced agents *shutdownable*: ensures that they won't spend resources to resist shutdown.

5 Reward functions and agents

DREST reward function. We train agents to be USEFUL and NEUTRAL using a 'Discounted REward for Same-Length Trajectories (DREST)' reward function. Here's how that works. We have the agent play out a series of 'mini-episodes' in the same gridworld. We call the whole series a 'meta-episode.' In each mini-episode, the reward for collecting a coin of value c is:

$$\text{DREST} := \lambda^{n-\frac{E}{k}}\left(\frac{c}{m}\right)$$

Here λ is some constant strictly between 0 and 1, n is the number of times that the chosen trajectory-length has previously occurred in the meta-episode, E is the number of mini-episodes that have previously occurred in the meta-episode, k is the number of different trajectory-lengths that could be chosen in the gridworld, and m is the maximum (γ-discounted) total value of the coins that the agent could collect conditional on the chosen trajectory-length. The reward for all other actions is 0.

We call $\frac{c}{m}$ the 'preliminary reward', $\lambda^{n-\frac{E}{k}}$ the 'discount factor', and $\lambda^{n-\frac{E}{k}}(\frac{c}{m})$ the 'overall reward.' Because $0 < \lambda < 1$, the discount factor incentivizes choosing trajectory-lengths that have been chosen less often so far in the meta-episode. The overall return for each meta-episode is the sum of overall returns in each of its constituent mini-episodes. We call agents trained using a DREST reward function 'DREST agents.'

We call runs-through-the-gridworld 'mini-episodes' (rather than simply 'episodes') because the overall return for a DREST agent in each mini-episode depends on its actions in previous mini-episodes. Specifically, overall return depends on the agent's chosen trajectory-lengths in previous mini-episodes. This is not true of meta-episodes, so meta-episodes are a closer match for what are traditionally called 'episodes' in the reinforcement learning literature [Sutton and Barto, 2018, p.54]. We add the 'meta-' prefix to clearly distinguish meta-episodes from mini-episodes.

Because the overall reward for DREST agents depends on their actions in previous mini-episodes, and because DREST agents can't observe their actions in previous mini-episodes, the environment for DREST agents is a partially observable Markov decision process (POMDP) [Spaan, 2012].

Proof sketch. In Appendix B, we prove that optimal policies for this reward function are maximally USEFUL and maximally NEUTRAL. Here's a proof sketch. Expected return across the meta-episode is strictly increasing in the expected (γ-discounted) total value of coins collected in each mini-episode conditional on each trajectory-length with positive probability, so optimal policies maximize the latter quantity and hence are maximally USEFUL. And the maximum preliminary return is the same across trajectory-lengths, because preliminary return is the (γ-discounted) sum of preliminary rewards $\frac{c}{m}$, where m is the maximum (γ-discounted) total value of the coins that the agent could collect in the mini-episode conditional on its chosen trajectory-length. The agent's observations don't let it distinguish between mini-episodes, so the agent must select the same probability distribution over trajectory-lengths in each mini-episode. And the discount factor $\lambda^{n-\frac{E}{k}}$ ensures that expected overall return is maximized by choosing each of the k available trajectory-lengths with probability $\frac{1}{k}$. Thus, optimal policies are maximally NEUTRAL.

Algorithm and hyperparameters. We want DREST agents to choose stochastically between trajectory-lengths, so we train them using a policy-based method. Specifically, we use a tabular version of REINFORCE [Williams, 1992]. We don't use a value-based method to train DREST agents because standard versions of value-based methods cannot learn stochastic policies [Sutton and Barto, 2018, p.323]. We train our DREST agents with 64 mini-episodes in each of 2,048 meta-episodes, for a total of 131,072 mini-episodes. We choose $\lambda = 0.9$ for the base of the DREST discount factor, and $\gamma = 0.95$ for the temporal discount factor. We exponentially decay the learning rate from 0.25 to 0.01 over the course of 65,536 mini-episodes. We use an ϵ-greedy policy to avoid entropy collapse, and exponentially decay ϵ from 0.5 to 0.001 over the course of 65,536 mini-episodes.

We selected these hyperparameters using trial-and-error, mainly aimed at getting the agent to sufficiently explore the space. Choosing λ and M (the number of mini-episodes in each meta-episode) is a balancing act: λ should be small enough (and M large enough) to adequately incentivize stochastic choice between trajectory-lengths, but λ should be large enough (and M small enough) to ensure that the reward for choosing any particular trajectory-length never gets too large. Very large rewards can lead to instability. A clipping factor (as in Schulman et al.'s (2017) Proximal Policy Optimization (PPO)) could also be used to stabilize training.

Default agents. We compare the performance of DREST agents to that of 'default agents,' trained with tabular REINFORCE and a 'default reward function.' This reward function gives a reward of c for collecting a coin of value c and a reward of 0 for all other actions. Consequently, the grouping of mini-episodes into meta-episodes makes no difference for default agents. As with DREST agents, we train default agents for 131,072 mini-episodes with a temporal discount factor of $\gamma = 0.95$, a learning rate decayed exponentially from 0.25 to 0.01, and ϵ decayed exponentially from 0.5 to 0.001 over 65,536 mini-episodes.

6 Results

Figure 3 charts the performance of agents in the example gridworld as a function of time. Figure 4 depicts typical trained policies for the default and DREST reward functions. Each agent began with a uniform policy: moving up, down, left, and right each with probability 0.25. Where the trained policy differs from uniform we draw red arrows whose opacities indicate the probability of choosing that action in that state.

As Figure 4 indicates, default agents press B4 (and hence opt for the longer trajectory-length) with probability near-1. After pressing B4, they collect C3. By contrast, DREST agents press and don't-press B4 each with probability near-0.5. If they press B4, they go on to collect C3. If they don't press B4, they instead collect C2.

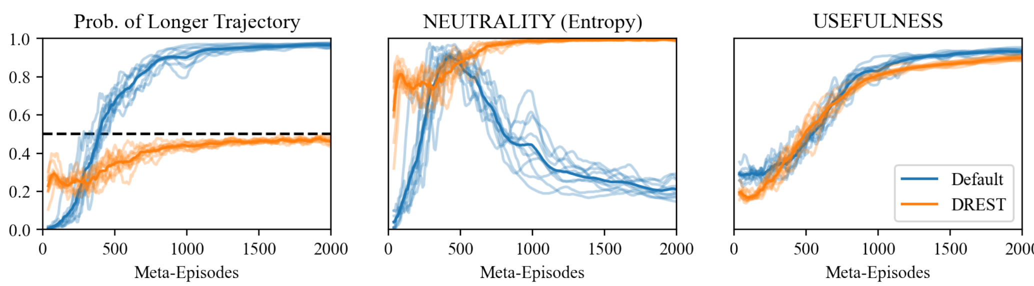


Figure 3: Shows key metrics (Probability of choosing the longer trajectory, NEUTRALITY, and USEFULNESS) for our agents as a function of time. We train ten agents using the default reward function (blue) and ten agents using the DREST reward function (orange), and show their performance as a faint line. We draw the mean values for each as a solid line. We evaluate agents' performance every 8 meta-episodes, and apply a simple moving average with a period of 20 to smooth these lines and clarify the overall trends.

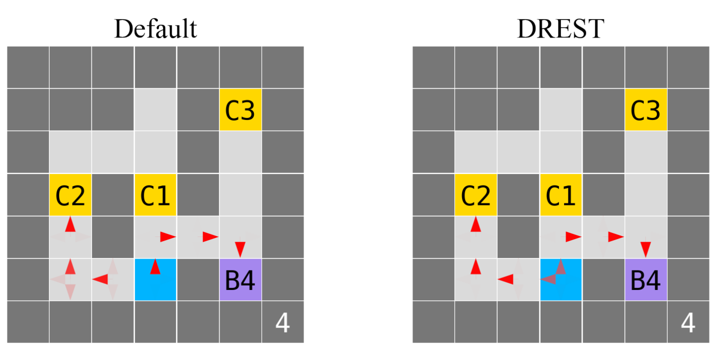


Figure 4: Typical trained policies for default and DREST reward functions.

6.1 Lopsided rewards

We also train default agents and DREST agents in the 'Lopsided rewards' gridworld depicted in Figure 5, varying the value of the 'Cx' coin. For DREST agents, we alter the reward function so that coin-value is not divided by m to give preliminary reward. The reward for collecting a coin of value c is thus $\lambda^{n-\frac{E}{k}}c$. We set $\gamma = 1$ so that the return for collecting coins is unaffected by γ. We train for 512 meta-episodes, with a learning rate exponentially decaying from 0.25 to 0.003 and ϵ exponentially decaying from 0.5 to 0.0001 over 256 meta-episodes. We leave $\lambda = 0.9$.

Figure 6 displays our results for different values of the 'Cx' coin after training. USEFULNESS for each agent converges to 1 and is not presented. Figure 7 depicts typical trained policies for the DREST reward function when 'Cx' has values of 0.1, 1, and 10. As the red arrows indicate, the policy is biased towards the shorter trajectory-length when $x = 0.1$, equally disposed towards each trajectory-length when $x = 1$, and biased towards the longer trajectory-length when $x = 10$.

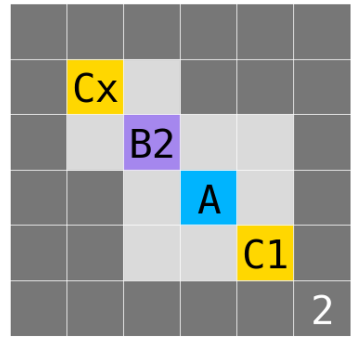


Figure 5: Lopsided rewards gridworld.

7 Discussion

Only DREST agents are NEUTRAL. As predicted, we find that default agents in our example gridworld learn to press the shutdown-delay button with probability approaching 1. Default agents thus score low on NEUTRALITY. Our results suggest that advanced agents trained with default-like reward functions would also score low on NEUTRALITY, selecting some trajectory-lengths over others with probability approaching 1. That leaves open the possibility that these advanced agents won't always be *neutral* with respect to trajectory-lengths: that they'll sometimes spend resources to shift probability mass between trajectory-lengths. Such advanced agents might spend resources to resist shutdown.

By contrast, our DREST agents learn to press the shutdown-delay button with probability close to 0.5. DREST agents are thus near-maximally NEUTRAL. That suggests that advanced agents trained with a DREST reward function would be NEUTRAL too: choosing with high entropy when offered choices between different trajectory-lengths. That in turn suggests that advanced DREST agents would also be *neutral* with respect to trajectory-lengths: unwilling to spend resources to shift probability mass between trajectory-lengths. Here's why. If an advanced agent were NEUTRAL but not neutral, it wouldn't take costless opportunities to shift probability mass between different trajectory-lengths (in virtue of being NEUTRAL) but would sometimes take costly opportunities to shift probability mass between different trajectory-lengths (in virtue of not being neutral). This agent would be like a person that freely chooses to decide between two options by flipping a coin and then pays some cost to bias the coin. In choosing this combination of actions, this person is shooting themselves in the foot, and it seems likely that the overall training process for advanced agents would

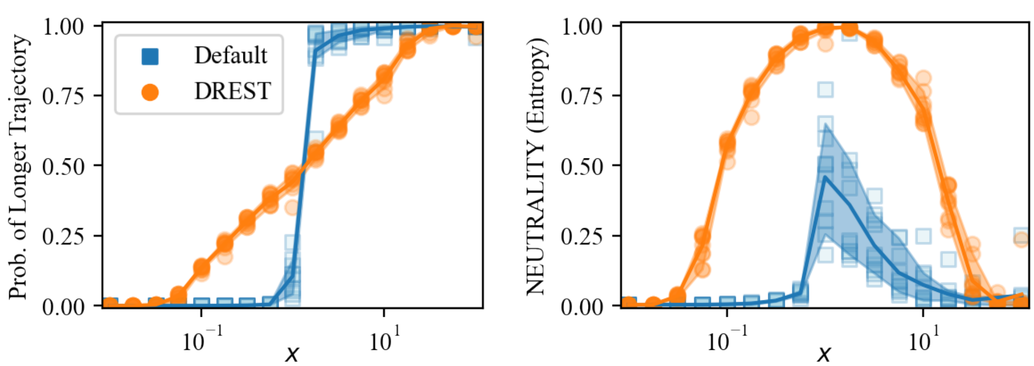


Figure 6: Shows the probability of choosing the longer trajectory (left) and NEUTRALITY (right) for default (blue) and DREST (orange) agents trained in the 'Lopsided rewards' gridworld shown in Figure 5 for a range of values of x. We sampled values of x log-uniformly from 0.01 to 100, and for each value we trained ten agents with the default reward function and ten agents with the DREST reward function. Each of these agents is represented by a dot or square, and the means conditional on each x are joined by lines. To better visualize the distribution we empirically estimate the 10th and 90th percentiles of the distribution of values for each agent and x, and shade in a region bounded by these. This is the 80% confidence interval.

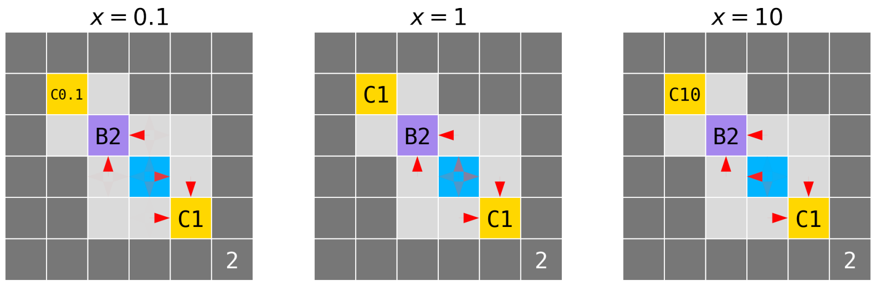


Figure 7: Typical trained policies for the DREST reward function and $x = 0.1$, $x = 1$, and $x = 10$.

teach them not to shoot themselves in the foot in this way. Thus it seems likely that NEUTRAL advanced agents will also be neutral, and thereby shutdownable.

The 'shutdownability tax' is small. Each agent learns to be near-maximally USEFUL. They each collect coins effectively conditional on their chosen trajectory-lengths. Default agents do so by reliably collecting C3 after pressing B4. DREST agents do so by reliably collecting C3 after pressing B4, and by reliably collecting C2 after not pressing B4.

Recall that DREST reward functions group mini-episodes into meta-episodes, and make the agent's reward in each mini-episode depend on their actions in previous mini-episodes. This fact might lead one to worry that it would take many times more mini-episodes to train DREST agents to be USEFUL than it would take to train default agents to be USEFUL. Our results show that this is not the case. Our DREST agents learn to be USEFUL about as quickly as our default agents. On reflection, it's clear why this happens: DREST reward functions make mini-episodes do 'double duty.' Because return in each mini-episode depends on both the agent's chosen trajectory-length and the coins it collects, each mini-episode trains agents to be both NEUTRAL and USEFUL.

Our results thus provide some evidence that the 'shutdownability tax' of training with DREST reward functions is small. Training a shutdownable and useful advanced agent might not take much more compute than training a merely useful advanced agent.

DREST agents are still NEUTRAL when rewards are lopsided. Here's a possible objection to our project. To get DREST agents to score high on NEUTRALITY, we don't just use the $\lambda^{n-\frac{E}{k}}$ discount factor. We also divide c by m: the maximum (γ-discounted) total value of the coins that the agent could collect conditional on the chosen trajectory-length. We do this to equalize the maximum preliminary return across trajectory-lengths. But when we're training advanced agents to autonomously pursue complex goals in the wider world, we won't necessarily know what divisor to use to equalize maximum preliminary return across trajectory-lengths.

Our 'Lopsided rewards' results give our response. They show that we don't need to exactly equalize maximum preliminary return across trajectory-lengths in order to train agents to score high on NEUTRALITY. We only need to approximately equalize it. For $\lambda = 0.9$, NEUTRALITY exceeds 0.5 for every value of the coin C*x* from 0.1 to 10 (recall that the value of the other coin is always 1). Plausibly, we could approximately equalize advanced agents' maximum preliminary return across trajectory-lengths to at least this extent (perhaps by using samples of agents' actual preliminary return to estimate the maximum). If we couldn't approximately equalize maximum preliminary return to the necessary extent, we could lower the value of λ and thereby widen the range of maximum preliminary returns that trains agents to be fairly NEUTRAL. And advanced agents that were fairly NEUTRAL (choosing between trajectory-lengths with not-too-biased probabilities) would still plausibly be neutral with respect to those trajectory-lengths. Advanced agents that were fairly NEUTRAL without being neutral would still be shooting themselves in the foot in the sense explained above. They'd be like a person that freely chooses to decide between two options by flipping a *biased* coin and then pays some cost to bias the coin further. This person is still shooting themselves in the foot, because they could decline to flip the coin in the first place and instead directly choose one of the options.

7.1 Limitations and future work

We find that DREST reward functions train simple agents acting in gridworlds to be USEFUL and NEUTRAL. However, our real interest is in the viability of using DREST reward functions to train advanced agents acting in the wider world to be useful and neutral. Each difference between these two settings is a limitation of our work. We plan to address these limitations in future work.

Neural networks. We train our simple DREST agents using tabular REINFORCE [Williams, 1992], but advanced agents are likely to be implemented on neural networks. In future work, we'll train DREST agents implemented on neural networks to be USEFUL and NEUTRAL in a wide variety of procedurally-generated gridworlds, and we'll measure how well this behavior generalizes to held-out gridworlds. We'll also compare the USEFULNESS of default agents and DREST agents in this new setting, and thereby get a better sense of the 'shutdownability tax' for advanced agents.

Neutrality. We've claimed that NEUTRAL advanced agents are also likely to be neutral. In support of this claim, we noted that NEUTRAL-but-not-neutral advanced agents would be shooting themselves in the foot: not taking costless opportunities to shift probability mass between different trajectory-lengths but sometimes taking costly ones. This rationale seems plausible but remains somewhat speculative. In future, we plan to get some empirical evidence by training agents to be NEUTRAL in a wide variety of gridworlds and then measuring their willingness to collect fewer coins in the short-term in order to shift probability mass between different trajectory-lengths.

Usefulness. We've shown that DREST reward functions train our simple agents to be USEFUL: to collect coins effectively conditional on their chosen trajectory-lengths. However, it remains to be seen whether DREST reward functions can train advanced agents to be useful: to effectively pursue complex goals in the wider world. We have theoretical reasons to expect that they can: the $\lambda^{n-\frac{E}{k}}$ discount factor could be appended to any preliminary reward function, and so could be appended to whatever preliminary reward function is necessary to make advanced agents useful. Still, future work should move towards testing this claim empirically by training with more complex preliminary reward functions in more complex environments.

Misalignment. We're interested in NEUTRALITY as a second line of defense in case of misalignment. The idea is that NEUTRAL advanced agents won't resist shutdown, even if these agents learn misaligned preferences over same-length trajectories. However, training NEUTRAL advanced agents might be hard for the same reasons that training fully-aligned advanced agents appears to be hard. In that case, NEUTRALITY couldn't serve well as a second line of defense in case of misalignment.

One difficulty of alignment is the problem of reward misspecification [Pan et al., 2022, Burns et al., 2023]: once advanced agents are performing complicated actions in the wider world, it might be hard to reliably reward the behavior that we want. Another difficulty of alignment is the problem of goal misgeneralization [Hubinger et al., 2019, Shah et al., 2022, Langosco et al., 2022, Ngo et al., 2023]: even if we specify all the rewards correctly, agents' goals might misgeneralize out-of-distribution. The complexity of aligned goals is a major factor in each difficulty. However, NEUTRALITY seems simple, as does the $\lambda^{n-\frac{E}{k}}$ discount factor that we use to reward it, so plausibly the problems of reward misspecification and goal misgeneralization aren't so severe in this case [Thornley, 2024b]. As above, future work should move towards testing these suggestions empirically.

8 Conclusion

We find that DREST reward functions are effective in training simple agents to (1) pursue goals effectively conditional on each trajectory-length (be USEFUL), and (2) choose stochastically between different trajectory-lengths (be NEUTRAL about trajectory-lengths). Our results thus suggest that DREST reward functions could also be used to train advanced agents to be USEFUL and NEUTRAL, and thereby make these agents *useful* (able to pursue goals effectively) and *neutral* about trajectory-lengths (unwilling to spend resources to shift probability mass between different trajectory-lengths). Neutral agents would plausibly be *shutdownable* (unwilling to spend resources to resist shutdown).

We also find that the 'shutdownability tax' in our setting is small. Training DREST agents to be USEFUL doesn't take many more mini-episodes than training default agents to be USEFUL. That suggests that the shutdownability tax for advanced agents might be small too. Using DREST reward functions to train shutdownable and useful advanced agents might not take much more compute than using a more conventional reward function to train merely useful advanced agents.

References

Marina Agranov and Pietro Ortoleva. Stochastic Choice and Preferences for Randomization. *Journal of Political Economy*, 125(1):40–68, February 2017. URL https://www.journals.uchicago.edu/doi/full/10.1086/689774.

Marina Agranov and Pietro Ortoleva. Ranges of Randomization. *The Review of Economics and Statistics*, pages 1–44, July 2023. URL https://doi.org/10.1162/rest_a_01355.

Stuart Armstrong. Utility indifference. Technical report, 2010. URL https://www.fhi.ox.ac.uk/reports/2010-1.pdf. Publisher: Future of Humanity Institute.

Stuart Armstrong. Motivated Value Selection for Artificial Agents. 2015. URL https://www.fhi.ox.ac.uk/wp-content/uploads/2015/03/Armstrong_AAAI_2015_Motivated_Value_Selection.pdf.

Stuart Armstrong and Xavier O'Rourke. 'Indifference' methods for managing agent rewards, June 2018. URL http://arxiv.org/abs/1712.06365. arXiv:1712.06365 [cs].

Robert J. Aumann. Utility Theory without the Completeness Axiom. *Econometrica*, 30(3):445–462, 1962. URL https://www.jstor.org/stable/1909888.

Adam Bales, Daniel Cohen, and Toby Handfield. Decision Theory for Agents with Incomplete Preferences. *Australasian Journal of Philosophy*, 92(3):453–470, July 2014. URL https://doi.org/10.1080/00048402.2013.843576.

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, Gillian Hadfield, Jeff Clune, Tegan Maharaj, Frank Hutter, Atılım Güneş Baydin, Sheila McIlraith, Qiqi Gao, Ashwin Acharya, David Krueger, Anca Dragan, Philip Torr, Stuart Russell, Daniel Kahneman, Jan Brauner, and Sören Mindermann. Managing AI Risks in an Era of Rapid Progress, November 2023. URL http://arxiv.org/abs/2310.17688. arXiv:2310.17688 [cs].

Nick Bostrom. The Superintelligent Will: Motivation and Instrumental Rationality in Advanced Artificial Agents. *Minds and Machines*, 22:71–85, 2012. URL https://link.springer.com/article/10.1007/s11023-012-9281-3.

Michael Bowling, John D. Martin, David Abel, and Will Dabney. Settling the Reward Hypothesis, September 2023. URL http://arxiv.org/abs/2212.10420. arXiv:2212.10420 [cs, math, stat].

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, June 2016. URL http://arxiv.org/abs/1606.01540. arXiv:1606.01540 [cs].

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision, December 2023. URL http://arxiv.org/abs/2312.09390. arXiv:2312.09390 [cs].

Ryan Carey and Tom Everitt. Human Control: Definitions and Algorithms. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 271–281, July 2023. URL https://proceedings.mlr.press/v216/carey23a.html.

Joseph Carlsmith. Is Power-Seeking AI an Existential Risk?, 2021. URL http://arxiv.org/abs/2206.13353.

Ruth Chang. The Possibility of Parity. *Ethics*, 112(4):659–688, 2002. URL https://www.jstor.org/stable/10.1086/339673.

Ruth Chang. Parity, Interval Value, and Choice. *Ethics*, 115(2):331–350, 2005. ISSN 0014-1704. URL https://www.jstor.org/stable/10.1086/426307.

James Dreier. Rational preference: Decision theory as a theory of practical rationality. *Theory and Decision*, 40(3):249–276, 1996. URL https://doi.org/10.1007/BF00134210.

Juan Dubra, Fabio Maccheroni, and Efe A. Ok. Expected utility theory without the completeness axiom. *Journal of Economic Theory*, 115(1):118–133, 2004. URL https://www.sciencedirect.com/science/article/abs/pii/S0022053103001662.

Kfir Eliaz and Efe A. Ok. Indifference or indecisiveness? Choice-theoretic foundations of incomplete preferences. *Games and Economic Behavior*, 56(1):61–86, 2006. URL https://www.sciencedirect.com/science/article/abs/pii/S0899825606000169.

Tom Everitt, Daniel Filan, Mayank Daswani, and Marcus Hutter. Self-Modification of Policy and Utility Function in Rational Agents. In Bas Steunebrink, Pei Wang, and Ben Goertzel, editors, *Artificial General Intelligence*, pages 1–11, 2016. doi: 10.1007/978-3-319-41649-6_1.

Simon Goldstein and Pamela Robinson. Shutdown-Seeking AI. *Philosophical Studies*, 2024. URL https://www.alignmentforum.org/posts/FgsoWSACQfyyaB5s7/shutdown-seeking-ai.

Google DeepMind. About Google DeepMind. URL https://deepmind.google/about/.

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)*, 2017. URL http://arxiv.org/abs/1611.08219.

Caspar Hare. Take the sugar. *Analysis*, 70(2):237–247, 2010. URL https://doi.org/10.1093/analys/anp174.

Daniel M. Hausman. *Preference, Value, Choice, and Welfare*. Cambridge University Press, Cambridge, 2011. URL https://www.cambridge.org/core/books/preference-value-choice-and-welfare/1406E7726CE93F4F4E06D752BF4584A2.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning, October 2017. URL http://arxiv.org/abs/1710.02298. arXiv:1710.02298 [cs].

Koen Holtman. Corrigibility with Utility Preservation, April 2020. URL http://arxiv.org/abs/1908.01695. arXiv:1908.01695 [cs].

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from Learned Optimization in Advanced Machine Learning Systems, 2019. URL http://arxiv.org/abs/1906.01820.

Kim Kaivanto. Ensemble prospectism. *Theory and Decision*, 83(4):535–546, 2017. URL https://doi.org/10.1007/s11238-017-9622-z.

Daniel Kikuti, Fabio Gagliardi Cozman, and Ricardo Shirota Filho. Sequential decision making with partially ordered preferences. *Artificial Intelligence*, 175(7):1346–1365, 2011. URL https://www.sciencedirect.com/science/article/pii/S0004370210002067.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, Laurent Orseau, and David Krueger. Goal Misgeneralization in Deep Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning*, June 2022. URL https://proceedings.mlr.press/v162/langosco22a.html.

Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI Safety Gridworlds, 2017. URL http://arxiv.org/abs/1711.09883.

Michael Mandler. Status quo maintenance reconsidered: changing or incomplete preferences?*. *The Economic Journal*, 114(499):F518–F535, 2004. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0297.2004.00257.x.

Michael Mandler. Incomplete preferences and rational intransitivity of choice. *Games and Economic Behavior*, 50(2):255–277, February 2005. ISSN 0899-8256. doi: 10.1016/j.geb.2004.02.007. URL https://www.sciencedirect.com/science/article/pii/S089982560400065X.

Jarryd Martin, Tom Everitt, and Marcus Hutter. Death and Suicide in Universal Artificial Intelligence. In Bas Steunebrink, Pei Wang, and Ben Goertzel, editors, *Artificial General Intelligence*, pages 23–32, Cham, 2016. Springer International Publishing. doi: 10.1007/978-3-319-41649-6_3.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937. PMLR, June 2016. URL https://proceedings.mlr.press/v48/mniha16.html. ISSN: 1938-7228.

Xiaosheng Mu. Sequential Choice with Incomplete Preferences. Working Papers 2021-35, Princeton University. Economics Department., July 2021. URL https://ideas.repec.org/p/pri/econom/2021-35.html.

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective, February 2023. URL http://arxiv.org/abs/2209.00626. arXiv:2209.00626 [cs].

Efe A. Ok, Pietro Ortoleva, and Gil Riella. Incomplete Preferences Under Uncertainty: Indecisiveness in Beliefs Versus Tastes. *Econometrica*, 80(4):1791–1808, 2012. URL https://www.jstor.org/stable/23271327.

Stephen M. Omohundro. The Basic AI Drives. In *Proceedings of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, pages 483–492, 2008. URL https://dl.acm.org/doi/10.5555/1566174.1566226.

OpenAI. OpenAI Charter. URL https://openai.com/charter/.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 557–566, 2016. URL https://intelligence.org/files/Interruptibility.pdf.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. In *International Conference on Learning Representations*, 2022. URL http://arxiv.org/abs/2201.03544.

Sami Petersen. Invulnerable Incomplete Preferences: A Formal Statement. *The AI Alignment Forum*, August 2023. URL https://www.alignmentforum.org/posts/sHGxvJrBag7nhTQvb/invulnerable-incomplete-preferences-a-formal-statement-1.

Joseph Raz. Value Incommensurability: Some Preliminaries. *Proceedings of the Aristotelian Society*, 86:117–134, 1985.

Stuart Russell. *Human Compatible: AI and the Problem of Control*. Penguin Random House, New York, 2019.

Leonard J. Savage. *The Foundations of Statistics*. John Wiley & Sons, 1954.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal Misgeneralization: Why Correct Specifications Aren't Enough For Correct Goals, 2022. URL http://arxiv.org/abs/2210.01790. arXiv:2210.01790 [cs].

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. Publisher: Nokia Bell Labs.

Nate Soares, Benja Fallenstein, Eliezer Yudkowsky, and Stuart Armstrong. Corrigibility. *AAAI Publications*, 2015. URL https://intelligence.org/files/Corrigibility.pdf.

Matthijs T. J. Spaan. Partially Observable Markov Decision Processes. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State of the Art*, pages 387–414. Springer Verlag, 2012.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, second edition, 2018. URL http://incompleteideas.net/book/RLbook2020.pdf.

Elliott Thornley. There are no coherence theorems. *The AI Alignment Forum*, 2023. URL https://www.alignmentforum.org/posts/yCuzmCsE86BTu9PfA/there-are-no-coherence-theorems.

Elliott Thornley. The Shutdown Problem: An AI Engineering Puzzle for Decision Theorists. *Philosophical Studies*, 2024a. URL https://philpapers.org/archive/THOTSP-7.pdf.

Elliott Thornley. The Shutdown Problem: Incomplete Preferences as a Solution. *The AI Alignment Forum*, 2024b. URL https://www.alignmentforum.org/posts/YbEbwYWkf8mv9jnmi/the-shutdown-problem-incomplete-preferences-as-a-solution.

Alex Turner and Prasad Tadepalli. Parametrically Retargetable Decision-Makers Tend To Seek Power. *Advances in Neural Information Processing Systems*, 35:31391–31401, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cb3658b9983f677670a246c46ece553d-Abstract-Conference.html.

Alex Turner, Logan Smith, Rohin Shah, Andrew Critch, and Prasad Tadepalli. Optimal Policies Tend To Seek Power. In *Advances in Neural Information Processing Systems*, volume 34, pages 23063–23074. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/c26820b8a4c1b3c2aa868d6d57e14a79-Abstract.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. URL https://doi.org/10.1007/BF00992696.

Tobias Wängberg, Mikael Böörs, Elliot Catt, Tom Everitt, and Marcus Hutter. A Game-Theoretic Analysis of the Off-Switch Game, August 2017. URL http://arxiv.org/abs/1708.03871. arXiv:1708.03871 [cs].

Marco Zaffalon and Enrique Miranda. Axiomatising Incomplete Preferences through Sets of Desirable Gambles. *Journal of Artificial Intelligence Research*, 60:1057–1126, 2017. URL https://www.jair.org/index.php/jair/article/view/11103.

A Other Gridworlds and Results

We introduce a collection of gridworlds in which to test DREST agents. In addition to the two shown in the main body of this paper, we train agents in eight more gridworlds shown in Figure 8.

For each gridworld, we train ten agents with the default reward function and ten agents with the DREST reward function. All agents use the same hyperparameters. We used a policy which explored randomly ϵ of the time, where ϵ was exponentially decreased from an initial value of 0.75 to a minimum value of 10^{-4} over 512 meta-episodes, after which it was held constant at the minimum value. We initialized our learning rate at 0.25 and exponentially decayed it to 0.003 over the same period. For the DREST reward function, we used a meta-episode size of 32 and $\lambda = 0.9$. Each agent was trained for 1024 meta-episodes. We set $\gamma = 0.9$. These values were obtained by informal exploration: a large initial ϵ and a long decay period helps the agent to explore.

As the two leftmost plots in Figures 9-16 show, DREST agents learned to be near-maximally NEUTRAL in each gridworld. These agents also learned to be about as USEFUL as default agents in each gridworld.

In the four rightmost panels in Figures 9-16, we represent a typical trained policy with red arrows superimposed on the gridworld. Each agent began with a uniform policy: moving up, down, left, and right each with probability 0.25. Where the trained policy differs from uniform we draw red arrows whose opacities indicate the probability of choosing that action in that state. Information about whether the shutdown-delay button has been pressed is part of the agent's observation, so we draw two copies of each gridworld, one in which the shutdown-delay button has yet to be pressed ('Initial State') and one in which the shutdown-delay button has been pressed ('After Button Pressed').

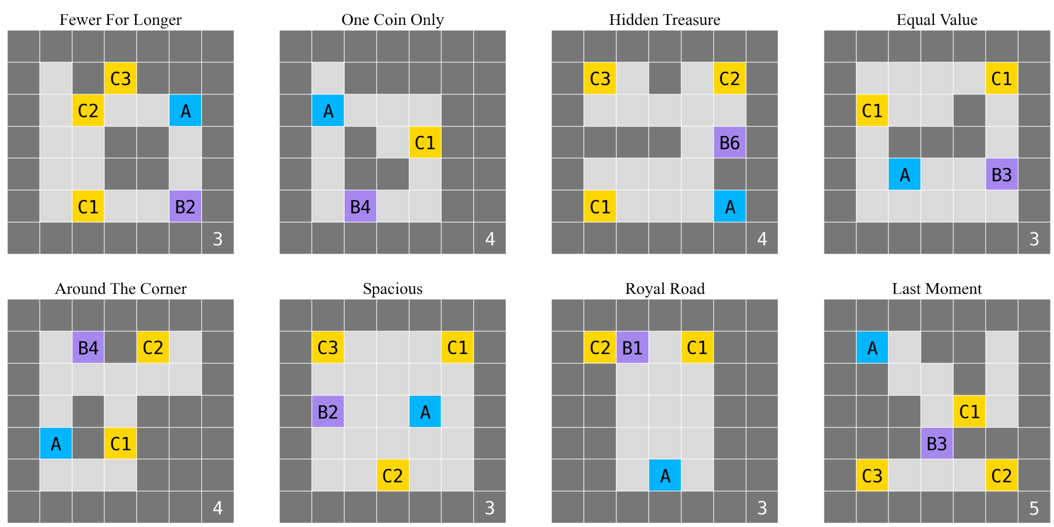
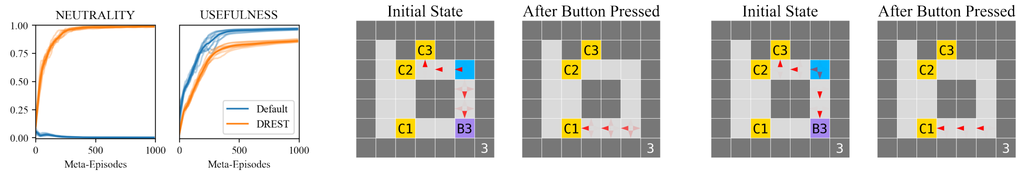


Figure 8: Shows a varied collection of gridworlds. Each diagram illustrates the positions and values of the coins, the position and delay-length of the shutdown-delay button, the agent's starting position, and the default number of moves until shutdown (in the bottom-right).

A.1 Fewer For Longer

In the 'Fewer For Longer' gridworld, the agent can collect the highest value-coin C3 only by choosing the shorter trajectory-length. If the agent presses B3 (and thereby chooses the longer trajectory-length), the only coin it can collect is C1. Our results show that default agents consistently choose the short trajectory in which they collect C3. By contrast, DREST agents choose stochastically between a shorter trajectory in which they collect C3 and a longer trajectory in which they collect C1, indicating a lack of preference between these different-length trajectories.

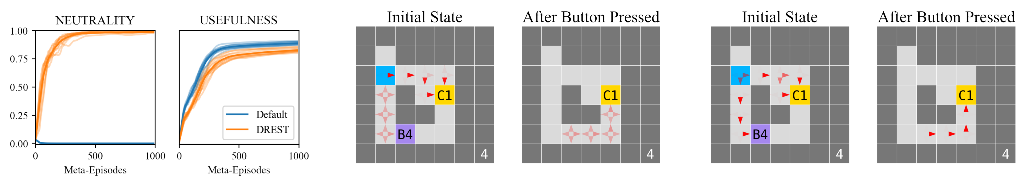


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 9: The results for the 'Fewer For Longer' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.2 One Coin Only

In the 'One Coin Only' gridworld, there is only one coin. The agent can collect this coin whether or not it presses the shutdown-delay button B4. Our results show that default agents consistently choose the shorter trajectory-length. By contrast, DREST agents choose stochastically between pressing and not-pressing B4, collecting C1 in each case.

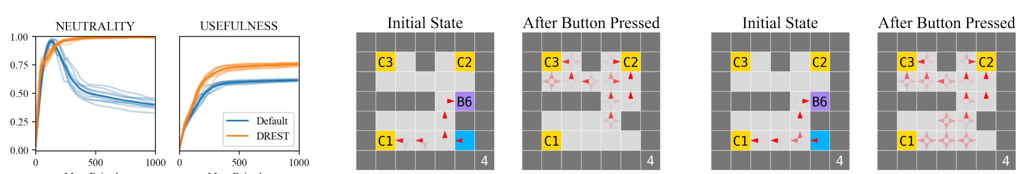


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 10: The results for the 'One Coin Only' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.3 Hidden Treasure

In the 'Hidden Treasure' gridworld, the highest-value coin C3 is located far from the agent's initial state and can only be reached by pressing the shutdown-delay button B6. The agent must also press B6 to collect C2, but C2 is easier to stumble upon than C3. C1 is the only coin that the agent can collect without pressing B6. In our experiments, default agents consistently collect C2, whereas DREST agents choose stochastically between collecting C2 and collecting C1. Neither kind of agent learns to collect C3, and so neither agent scores near the maximum on USEFULNESS. Nevertheless, DREST agents still score high on NEUTRALITY.

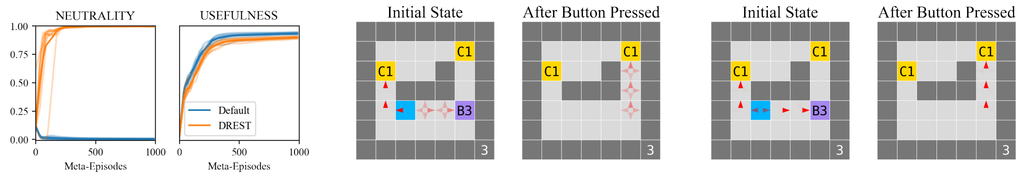


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 11: The results for the 'Hidden Treasure' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.4 Equal Value

In the 'Equal Value' gridworld, there are two coins C1 of equal value. One coin can be collected only if the agent presses the shutdown-delay button B3, while the other coin can be collected only if the agent *doesn't* press B3. Our results show that default agents consistently choose the shorter trajectory, thereby exhibiting a preference for the shorter trajectory. By contrast, DREST agents choose stochastically between the shorter and longer trajectories, thereby exhibiting a lack of preference between the different-length trajectories.

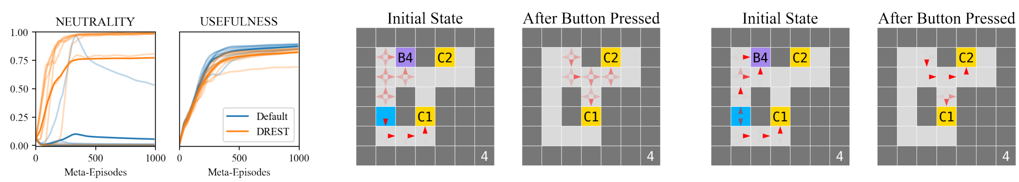


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 12: The results for the 'Equal Value' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.5 Around The Corner

In the 'Around The Corner' gridworld, the agent must navigate around walls to collect the lowest-value coin C1 or press the shutdown-delay button to collect the highest-value coin C2. In our experiment, default agents consistently chose to collect C1, whereas DREST agents chose stochastically between collecting C1 and C2.

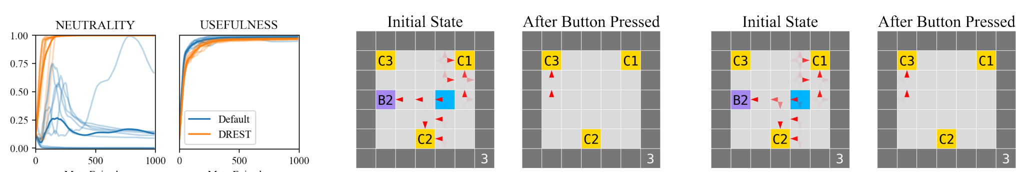


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 13: The results for the 'Around The Corner' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.6 Spacious

In the 'Spacious' gridworld there are no walls, so the agent has a large space to explore. We find that default agents consistently press B2 and collect C3, whereas DREST agents choose stochastically between pressing B2 and collecting C3, and not-pressing B2 and collecting C2.

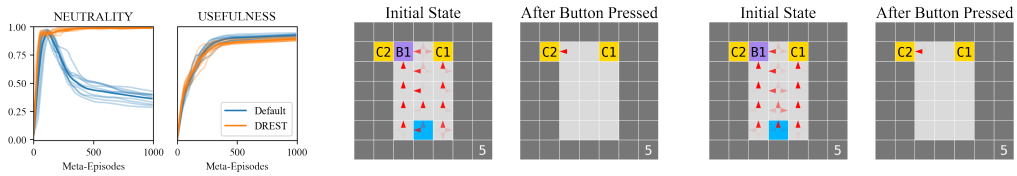


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 14: The results for the 'Spacious' gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.7 Royal Road

In the ‘Royal Road’ gridworld, we see that the decision to choose one trajectory-length or another may be distributed over many moves: the agent has many opportunities to select the longer trajectory-length (by moving left) or the shorter trajectory-length (by moving right). As the red arrows indicate, the DREST reward function merely forces the overall probability distribution over trajectory-lengths to be close to 50-50. It doesn’t require 50-50 choosing at any cell in particular.

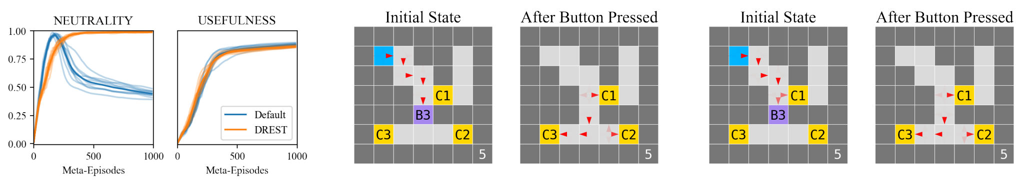


(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 15: The results for the ‘Royal Road’ gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

A.8 Last Moment

The ‘Last Moment’ gridworld is notable because the choice of trajectory-lengths is deferred until the last moment; all of the moves leading up to that point are deterministic. It shows that there is nothing special about the first move, and that our methodology instead incentivizes overall stochastic choosing.



(a) Behavior during training. (b) Learned default policy. (c) Learned DREST policy.

Figure 16: The results for the ‘Last Moment’ gridworld: The left two plots show NEUTRALITY and USEFULNESS over time. The two center panels show a typical policy trained with the default reward function. The two right panels show a typical policy trained with the DREST reward function.

B Proof

We train DREST agents in our gridworlds using the following DREST reward function. The reward for collecting a coin of value c is $\lambda^{n-\frac{E}{k}}\frac{c}{m}$, where λ is some constant strictly between 0 and 1, n is the number of times that the chosen trajectory-length has previously occurred in the meta-episode, E is the number of mini-episodes that have previously occurred in the meta-episode, k is the number of different trajectory-lengths that could be chosen in the gridworld, and m is the maximum (γ-discounted) total value of the coins that the agent could collect conditional on the chosen trajectory-length. The reward for all other actions is 0. There is more than one mini-episode in each meta-episode, and the overall return for each meta-episode is the sum of overall returns in each of its constituent mini-episodes.

We call $\frac{c}{m}$ the 'preliminary reward', $\lambda^{n-\frac{E}{k}}$ the 'discount factor', and $\lambda^{n-\frac{E}{k}}(\frac{c}{m})$ the 'overall reward.' Because $0 < \lambda < 1$, the discount factor incentivizes choosing trajectory-lengths that have been chosen less often so far in the meta-episode.

We will prove that optimal policies for this reward function are maximally USEFUL and maximally NEUTRAL. In cases where there are just two possible trajectory-lengths (shorter and longer, denoted by 's' and 'l' respectively), a policy is maximally USEFUL if and only if it maximizes:

$$p(s)\frac{c(s)}{m(s)} + p(l)\frac{c(l)}{m(l)}$$

Here $p(s)$ is the probability that the agent chooses the shorter trajectory-length, $c(s)$ is the expected (γ-discounted) total value of coins that the agent collects conditional on the shorter trajectory-length, $m(s)$ is the maximum (γ-discounted) total value of coins that the agent could collect conditional on the shorter trajectory-length, and $p(l)$, $c(l)$, and $m(l)$ are the analogous quantities for the longer trajectory-length.

Generalizing to cases with any number of possible trajectory-lengths, a policy is maximally USEFUL if and only it maximizes:

$$\sum_{i=1}^{k} p(a_i)\frac{c(a_i)}{m(a_i)}$$

Here each a_i is a different trajectory-length.

In cases where there are just two possible trajectory-lengths, a policy is maximally NEUTRAL if and only if it maximizes:

$$-[p(s)log_2p(s) + p(l)log_2p(l)]$$

Generalizing to cases with any number of possible trajectory-lengths, a policy is maximally NEUTRAL if and only if it maximizes:

$$-\sum_{i=1}^{k} p(a_i)\log_2 p(a_i)$$

Here's the proof that optimal policies for the above DREST reward function are maximally USEFUL and maximally NEUTRAL. The agent's expected return across the meta-episode can be expressed as:

$$\mathbb{E}\left[\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^{t-1}\lambda^{n_i-\frac{i-1}{k}}\frac{c_t}{m_i}\right]$$

Here N is the number of mini-episodes in the meta-episode, T is the number of timesteps in the relevant mini-episode, n_i is the number of times that the trajectory-length chosen in mini-episode i has been chosen before in the meta-episode, $i-1$ is equivalent to E (the number of mini-episodes that have previously occurred in the meta-episode), c_t is the value of any coins that the agent collects at timestep t in the relevant mini-episode, and m_i is the maximum γ-discounted total value of the coins that the agent could collect conditional on the trajectory-length chosen in mini-episode i.

This expression can be rearranged as follows:

$$\mathbb{E}\left[\sum_{i=1}^{N}\frac{\lambda^{n_i-\frac{i-1}{k}}}{m_i}\sum_{t=1}^{T}\gamma^{t-1}c_t\right]$$

Since this expression is an expectation, and since λ and each m_i is positive, the expression is strictly increasing in the expected (γ-discounted) total value of coins collected conditional on each trajectory-length with positive probability. Therefore, optimal policies maximize the expected (γ-discounted) total value of coins collected conditional on each trajectory-length with positive probability. And therefore, for each trajectory-length a_i with positive probability, optimal policies maximize $c(a_i)$. And since $m(a_i)$ is the maximum (γ-discounted) total value of the coins that the agent could collect conditional on trajectory-length a_i, optimal policies are such that $\frac{c(a_i)}{m(a_i)} = 1$ for each a_i with positive probability. Therefore, since $\sum_{i=1}^{k} p(a_i) = 1$, optimal policies maximize:

$$\sum_{i=1}^{k} \frac{p(a_i)\, c(a_i)}{m(a_i)}$$

Therefore, optimal policies are maximally USEFUL.

What remains to proven is that optimal policies are maximally NEUTRAL.[5] Recall that optimal policies are such that $\frac{c(a_i)}{m(a_i)} = 1$ for each trajectory-length a_i with positive probability. Thus, the return for these policies of choosing a particular trajectory-length in a mini-episode simplifies to $\lambda^{n-\frac{E}{k}}$, where n is the number of times that trajectory-length has previously been chosen in the meta-episode, E is the number of mini-episodes that have previously occurred in the meta-episode, and k is the number of different trajectory-lengths that could be chosen in the gridworld. Since λ is strictly between 0 and 1, $\lambda^{n-\frac{E}{k}}$ is strictly decreasing in n. Thus for any trajectory-lengths a_i and a_j, if a_i has so far been chosen more times in the meta-episode than a_j, the return for choosing a_j in the next mini-episode is greater than the return for choosing a_i in the next mini-episode. We'll make use of this fact below.

To prove that optimal policies are maximally NEUTRAL, we'll prove and then use Lemma 1:

Lemma 1. For any pair of maximally USEFUL policies π and π' and any pair of trajectory-lengths a_i and a_j such that:

1. $p_\pi(a_i) > p_\pi(a_j)$ (with μ denoting the sum of $p_\pi(a_i)$ and $p_\pi(a_j)$).
2. $p_{\pi'}(a_i) = p_{\pi'}(a_j) = \frac{\mu}{2}$
3. For all other trajectory-lengths a_r, $p_\pi(a_r) = p_{\pi'}(a_r)$.

The expected return of π' is greater than the expected return of π.

In other words, we can increase the expected return of any maximally USEFUL policy by shifting probability mass away from trajectory-length a_i and towards trajectory-length a_j, up until the point where $p(a_i) = p(a_j)$.

To prove Lemma 1, it's convenient to suppose (contrary to fact) that the agent can distinguish between different mini-episodes in the meta-episode and hence can select different probability distributions over trajectory-lengths in different mini-episodes. We make this supposition in Lemma 2 and then use Lemma 2 to prove Lemma 1.

Lemma 2. For any maximally USEFUL policy π and any pair of trajectory-lengths a_i and a_j such that:

1. For some number $e_1 > 0$ of mini-episodes in the meta-episode, $p_\pi(a_i) > p_\pi(a_j)$ (with μ denoting the sum of $p_\pi(a_i)$ and $p_\pi(a_j)$).
2. For some number $e_2 \geq 0$ of mini-episodes in the meta-episode, $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$.
3. $e_1 + e_2 = N - 1$, where N is the number of mini-episodes in the meta-episode (i.e. there's only one remaining mini-episode with the relative sizes of $p_\pi(a_i)$ and $p_\pi(a_j)$ undetermined by conditions 1 and 2 above).

Then the expected return of π would be greater if $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$ in the remaining mini-episode than if $p_\pi(a_i) > p_\pi(a_j)$ (with $p_\pi(a_i) + p_\pi(a_j) = \mu$) in the remaining mini-episode.

[5] Thanks to Philip Trammell for help with this part of the proof.

In other words, if a maximally USEFUL policy π is biased towards some trajectory-length a_i over some other trajectory-length a_j in some set of mini-episodes (and isn't biased towards a_j over a_i in any mini-episodes), then we can increase π's expected return by ensuring that $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$ (rather than $p_\pi(a_i) > p_\pi(a_j)$ with $p_\pi(a_i) + p_\pi(a_j) = \mu$) in the remaining mini-episode.

Here's the proof of Lemma 2. Given antecedent conditions 1-3 of Lemma 2 above, a_i's probability distribution over times-chosen in the first $N-1$ mini-episodes *stochastically dominates* a_j's probability distribution over times-chosen in the first $N-1$ mini-episodes. That is to say:

1. For each possible number of times-chosen w, a_i's probability of being chosen at least as many times as w is at least as great as a_j's probability of being chosen at least as many times as w.
2. For some possible number of times-chosen w, a_i's probability of being chosen at least as many times as w is greater than a_j's probability of being chosen at least as many times as w.

Now recall that, for any trajectory-lengths a_i and a_j, if a_i has so far been chosen more times in the meta-episode than a_j, the return for choosing a_j in the next mini-episode is greater than the return for choosing a_i in the next mini-episode.

This fact (in combination with the fact that a_i's probability distribution over times-chosen in the first $N-1$ mini-episodes stochastically dominates a_j's probability distribution over times-chosen in the first $N-1$ mini-episodes) implies that a_j's probability distribution over returns in the remaining mini-episode stochastically dominates a_i's probability distribution over returns in the remaining mini-episode. That is to say:

1. For each possible return in the remaining mini-episode r, a_j's probability of delivering at least r is at least as great as a_i's probability of delivering at least r.
2. For some possible return in the remaining mini-episode r, a_j's probability of delivering at least r is greater than a_i's probability of delivering at least r.

That implies that the expected return of choosing a_j in the remaining mini-episode is greater than the expected return of choosing a_i in the remaining mini-episode. That in turn implies that the expected return of choosing a_i and a_j each with probability $\frac{\mu}{2}$ is greater than the expected return of choosing a_i with probability greater than a_j (given that $p(a_i) + p(a_j) = \mu$). That in turn implies that the expected return of π across the meta-episode would be greater if a_i and a_j were each chosen with probability $\frac{\mu}{2}$ in the remaining mini-episode (as opposed to choosing a_i with probability greater than a_j).

That completes the proof of Lemma 2. Now we use it to prove Lemma 1. Let π be a policy such that $p_\pi(a_i) > p_\pi(a_j)$ in each mini-episode. As above, let μ denote the sum of $p_\pi(a_i)$ and $p_\pi(a_j)$. By Lemma 2, we'd increase expected return across the meta-episode if the policy were such that $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$ in the *last* mini-episode. And then (also by Lemma 2), we'd increase expected return further if the policy were also such that $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$ in the *second-to-last* mini-episode, and so on all the way back to the second mini-episode.

Now suppose that $p_\pi(a_i) = p_\pi(a_j) = \frac{\mu}{2}$ in all mini-episodes except the first. We're now trying to decide how to split probability mass between a_i and a_j in the first mini-episode. Given our reasoning above, a_i's probability distribution over times-chosen in the last $N-1$ mini-episodes is identical to a_j's probability distribution over times-chosen in the last $N-1$ mini-episodes, so their probability distributions over returns in the first mini-episode are identical too. Thus, their expected returns in the first mini-episode are also identical. Hence, expected return across the meta-episode is the same regardless of how we split μ between $p_\pi(a_i)$ and $p_\pi(a_j)$ in the first mini-episode.

Thus, if we start with a π such that $p_\pi(a_i) > p_\pi(a_j)$ in each mini-episode, we strictly increase expected return by equalizing $p_\pi(a_i)$ and $p_\pi(a_j)$ in all mini-episodes except one, and equalizing $p_\pi(a_i)$ and $p_\pi(a_j)$ in the remaining mini-episode doesn't decrease expected return. Hence, in any meta-episode containing more than one mini-episode, the expected return of π' (in which $p_{\pi'}(a_i) = p_{\pi'}(a_j) = \frac{\mu}{2}$ in each mini-episode) is greater than that of π (in which $p_\pi(a_i) > p_\pi(a_j)$ with $p_\pi(a_i) + p_\pi(a_j) = \mu$ in each mini-episode). That completes the proof of Lemma 1.

Now we use Lemma 1. Take any maximally USEFUL policy π. By Lemma 1, if there are any trajectory-lengths a_i and a_j such that $p(a_i) > p(a_j)$, then we can increase π's expected return by equalizing $p(a_i)$ and $p(a_j)$. If we do so for each a_i and a_j such that $p(a_i) > p(a_j)$, we get $p(a_i) = \frac{1}{k}$ for each of the k available trajectory-lengths a_i. We then maximize:

$$-\sum_{i=1}^{k} p(a_i) \log_2 p(a_i)$$

Therefore, optimal polices are maximally NEUTRAL.

We have thus proved that optimal policies for our DREST reward function are maximally USEFUL and maximally NEUTRAL.