Be Considerate: Avoiding Negative Side Effects in Reinforcement Learning

Parand Alizadeh Alamdari*
University of Toronto
Toronto, Canada
Vector Institute
Toronto, Canada
parand@cs.toronto.edu

Toryn Q. Klassen* †
University of Toronto
Toronto, Canada
Vector Institute
Toronto, Canada
toryn@cs.toronto.edu

Rodrigo Toro Icarte
Pontificia Universidad Católica de Chile
Santiago, Chile
Vector Institute
Toronto, Canada
rodrigo.toro@ing.puc.cl

Sheila A. McIlraith‡
University of Toronto
Toronto, Canada
Vector Institute
Toronto, Canada
sheila@cs.toronto.edu

ABSTRACT

In sequential decision making – whether it's realized with or without the benefit of a model – objectives are often underspecified or incomplete. This gives discretion to the acting agent to realize the stated objective in ways that may result in undesirable outcomes, including inadvertently creating an unsafe environment or indirectly impacting the agency of humans or other agents that typically operate in the environment. In this paper, we explore how to build a reinforcement learning (RL) agent that contemplates the impact of its actions on the wellbeing and agency of others in the environment, most notably humans. We endow RL agents with the ability to contemplate such impact by augmenting their reward based on expectation of future return by others in the environment, providing different criteria for characterizing impact. We further endow these agents with the ability to differentially factor this impact into their decision making, manifesting behaviour that ranges from self-centred to self-less, as demonstrated by experiments in gridworld environments.

KEYWORDS

AI Safety; Reinforcement Learning; Side Effects; Value Alignment

ACM Reference Format:
Parand Alizadeh Alamdari, Toryn Q. Klassen, Rodrigo Toro Icarte, and Sheila A. McIlraith. 2022. Be Considerate: Avoiding Negative Side Effects in Reinforcement Learning. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

Sequential decision making, whether it is realized via reinforcement learning (RL), supervised learning, or some form of probabilistic or otherwise symbolic planning using models – relies on the specification of an objective – a reward function to be optimized in the case of RL, or a goal to achieve in the case of symbolic planning. Recent work in AI safety has raised the concern that objective specifications are often underspecified or incomplete, neglecting to take into account potential undesirable (negative) side effects of achievement of the specified objective. As Amodei et al. [1] explain, "[F]or an agent operating in a large, multifaceted environment, an objective function that focuses on only one aspect of the environment may implicitly express indifference over other aspects of the environment." Stuart Russell gave the example of tasking a robot to get coffee from a coffee shop and the robot, in its singular commitment to achieving the stated objective, killing all those in the coffee shop that stood between it and the purchase of coffee [12]. A somewhat more benign example by Amodei et al. [1] is that of a robot breaking a vase that is on the optimal path between two points. A range of recent works have presented computational techniques for avoiding or learning to avoid negative side effects [e.g., 10, 11, 14, 23, 27].

Our concern in this paper is with how an RL agent can learn to act safely in the face of a potentially incomplete specification of the objective. Is avoiding negative side effects the answer? Amodei et al. observe that

> "avoiding side effects can be seen as a proxy for the things we really care about: avoiding negative externalities. If everyone likes a side effect, there's no need to avoid it."

In the spirit of this observation, we contend that *to act safely an agent should contemplate the impact of its actions on the wellbeing and agency of others in the environment.* Indeed, what may be construed as a positive effect for one agent may be a negative effect for another. Here we consider negative side effects to be those that impede the future wellbeing or agency of other agents.

The setup in this paper is *not* a multi-agent RL or cooperative AI setup, and this is done by design. We take the pragmatic stance

[*] Equal contribution
[†] Also with Schwartz Reisman Institute for Technology and Society.
[‡] Also with Schwartz Reisman Institute for Technology and Society.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

that in many real world settings, an RL agent will not be able to compel humans to consistently and rationally cooperate, if they deign to cooperate at all. As such, the problem we address is in a single RL-agent setting in which the other agents — which may be the humans that operate in the environment — are just part of the environment, operating with fixed policies, and it is the acting RL agent that is constructing a policy that minimizes its impact on the future agency of these other (human) agents. The acting agent is being considerate of others.

Here, we endow RL agents with the ability to consider in their learning the future welfare and continued agency of others in the environment. We do so by augmenting the RL agent's reward with an auxiliary reward that reflects different functions of expected future return of other agents. We contrast this with recent work on side effects that takes into account only how the agent's actions will affect its own future abilities [10, 11, 23]. Considering other agents' abilities when avoiding side effects was informally discussed by Turner [21], and we have also investigated it in the context of symbolic planning [8, 9].

In its most general case, we make no assumptions about the number of agents that exist in the environment, their actions, or details of their transition functions. However we show how individuals or groups of agents can be distinguished and differentially considered. We further endow these agents with the ability to control the degree to which impact on self versus others factors into their learning, resulting in behaviour that ranges from self-centred to self-less. Experiments in gridworld environments illustrate qualitative and quantitative properties of the proposed approach.

2 PRELIMINARIES

In this section, we review relevant definitions and notation. RL agents learn policies from experience. When the problem is fully-observable, it's standard to model the environment as a Markov Decision Process (MDP) [19]. We describe an MDP as a tuple $\langle S, A, T, r, \gamma \rangle$ where S is a finite set of states, A is a finite set of actions, $T(s_{t+1}|s_t, a_t)$ gives the probability of transitioning to state s_{t+1} when taking action a_t in state s_t, $r : S \times A \times S \longrightarrow \mathbb{R}$ is the reward function, and γ is the discount factor. Sometimes an MDP can also include a designated initial state $s_0 \in S$. When the agent takes an action $a_t \in A$ in a state $s_t \in S$, as result it ends up in a new state s_{t+1} drawn from the distribution $T(s_{t+1}|s_t, a_t)$, and receives the reward $r_{t+1} = r(s_t, a_t, s_{t+1})$. A *terminal state* in an MDP is a state s which can never be exited – i.e., $T(s|s, a) = 1$ for every action a – and from which no further reward can be gained – i.e., $r(s, a, s) = 0$ for every action a.

A *policy* is a (possibly stochastic) mapping from states to actions. Given a policy π, the *value* of a state s is the *expected return* of that state, that is, the expected sum of (discounted) rewards that the agent will get by following the policy π starting in s. That can be expressed using the value function V^π, defined as $V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} \mid s_t = s \right]$ where the $\mathbb{E}_\pi$ notation means that in the expectation, the action in each state $s_t, s_{t+1}, \dots$ is selected according to the policy π. The discount factor γ determines how much less valuable it is to receive rewards in the future instead of the present. An optimal policy will maximize the value of each state. The optimal value function V^* is the value function of an optimal policy. Similarly, a value (called a Q-value) can be associated with a state-action pair: $Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k+1} \mid s_t = s, a_t = a \right]$. Note that the first action selected is necessarily a, but the policy π is followed afterwards. The optimal Q-function is the Q-function Q^* corresponding to an optimal policy.

3 PROBLEM AND APPROACH

In Section 1, we suggested that to act safely an acting agent should contemplate the impact of its actions on the welfare and continued agency of those that act or react in the environment. In this section, we present several formulations that address this aspiration. For the purposes of this study, we consider an environment with a single *acting agent* that learns how to act via RL. Other agents exist within the environment, operating via fixed policies, and only acting after the acting agent has reached a terminating state. We assume that we can neither incentivize nor control these other agents. An evocative example may be to consider university students who share a kitchen environment, and we wish our RL agent – the acting agent, with some conception of what others may typically do in the kitchen — to learn how to act in the kitchen in a manner that is considerate of others who may use the kitchen after the acting agent is done.

3.1 Using Information about Value Functions

To incentivize the acting agent to consider the future wellbeing and agency of others, we augment its reward with an auxiliary reward that reflects the impact of its choice of actions on future agency and wellbeing of others in the environment. To reflect the acting agent's uncertainty about what is good for others, we make use of a distribution over value functions. In particular, suppose that we have a finite set $\mathcal{V}$ of possible value functions $V : S \longrightarrow \mathbb{R}$, and a probability distribution $\mathsf{P}(V)$ over that set. Note that we don't have to commit to how many agents there are (or what exactly their actions are). It could be that each $V \in \mathcal{V}$ corresponds to a different agent, that the set reflects all possible value functions of a unique agent, or anything in between. Also, each $V \in \mathcal{V}$ could reflect some aggregation of the value functions of all or some of the agents.

We define the augmented reward function as

$$r_{\text{value}}(s, a, s') = \begin{cases} \alpha_1 \cdot r_1(s, a, s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s, a, s') + \gamma \cdot \alpha_2 \cdot F(\mathcal{V}, \mathsf{P}, s') & \text{if } s' \text{ is terminal} \end{cases} \quad (1)$$

where r_1 is the acting agent's individual reward function, and F is some function. The hyperparameters α_1 and α_2, which we call "caring coefficents", are real numbers that determine the degrees to which the individual reward r_1 and the auxiliary reward $F(\mathcal{V}, \mathsf{P}, s')$ contribute to the overall reward. As we will see in Section 4, if $\alpha_1 = 1$ and $\alpha_2 = 0$, we just get the original reward function where the acting agent only values its own reward. The agent is oblivious to its impact on others in its environment. If $\alpha_1 = 0$, then the acting agent entirely ignores any reward it garners directly from its actions. Note that future activity does not have to start in exactly the same state at which the acting agent ended. V can be defined so that $V(s')$ gives the expected return of future activity considered over a known distribution of starting states, given that the acting agent ended in s'.

We consider three possible different definitions of $F(\mathcal{V}, \mathrm{P}, s')$:

$$\textstyle\sum_{V\in\mathcal{V}} \mathrm{P}(V)\cdot V(s') \qquad \text{expected future return} \tag{2}$$

$$\min_{V\in\mathcal{V}:\mathrm{P}(V)>0} V(s') \qquad \text{worst-case future return} \tag{3}$$

$$\textstyle\sum_{V\in\mathcal{V}} \mathrm{P}(V)\cdot \min(V(s'), V(s_0)) \qquad \text{penalize negative change} \tag{4}$$

In Eq. (2), $F(\mathcal{V}, \mathrm{P}, s')$ is the expected value of s', given the distribution on value functions. Under some conditions, this is a generalization of the auxiliary reward defined by Krakovna et al. [11], which assumed that the future value functions were ones of the acting agent (and so depended on the acting agent's own abilities). See subsection 5.1 for more details. Meanwhile, Eq. (3) considers the value of s' if the "worst-case" value function from $\mathcal{V}$ (that still has positive probability) is used.

Note that those two reward augmentations may incentivize the acting agent to not only avoid negative side effects, but also to cause "positive side effects" – to help other agents (assuming $\alpha_2 > 0$). To focus on avoiding negative side effects, Krakovna et al. [11] proposed comparing the state the agent ends up in against a *reference state* (see subsubsection 5.1.1 for more details), and that is applicable to our approach as well. In Eq. (4), we use one of the simplest possible reference states, the initial state: the auxiliary reward is the lower of $V(s')$ and $V(s_0)$, where s_0 is the initial state. The idea is to decrease the acting agent's reward when it decreases the expected future return, but to *not* increase the acting agent's reward for increasing that same expected return.

We have also explored associating different caring coefficients with different agents, and trying to preserve agents' ability to execute *options* [20] (see subsection 3.3).

A complication with our approach is that for some possible reward functions for the acting agent and future value functions, the acting agent may have an incentive to avoid terminating states, to avoid or delay the penalty for negative future return. This incentive would typically be undesirable. However, it can be shown that under some circumstances, the acting agent's optimal policy will be terminating. The proposition below and its proof are similar to a result of Illanes et al. [7, Theorem 1].

Proposition 1. *Let $M = \langle S, A, T, r_1, \gamma\rangle$ be an MDP where $\gamma = 1$, the reward function r_1 is negative everywhere, and there exists a terminating policy. Suppose r_{value} is the reward function constructed from r_1 according to Equation 2, using some distribution $\mathrm{P}(V)$. Then any optimal policy for the MDP $M' = \langle S, A, T, r_{\text{value}}, \gamma\rangle$ with the modified reward will terminate with probability 1.*

Proof. Suppose for contradiction that there is an optimal policy π^* for M' that is non-terminating. Then there is some state $s \in S$ so that the probability of reaching a terminal state from s by following π^* is some value $c < 1$. Since rewards are negative everywhere, that means that $V^{\pi^*}(s) = -\infty$. On the other hand, any terminating policy gives a finite value to each state. Since there is a terminating policy for M there is one for M', and so π^* cannot be optimal. □

3.2 Treating Agents Differently

To this point, we've utilized a distribution over value functions to capture the expected return on future behaviour within the environment. The distribution has made no commitments to the existence of individual agents. To differentiate individual agents and to have the ability to treat them differently, we augment our formulation with indices, $i = 1, \ldots, n$, corresponding to different agents (we will assume the acting agent is agent 1). Furthermore, for each agent i, suppose we have a finite set of possible value functions $\{V_1^{(i)}, V_2^{(i)}, \ldots\}$, and $\mathrm{P}(V_{ij})$ is the probability that $V_j^{(i)}$ is the real value function for agent i. We could then have a separate caring coefficient α_i for each agent i, and define the following reward function for the acting agent:

$$r'_{\text{value}}(s,a,s') = \begin{cases} \alpha_1 \cdot r_1(s,a,s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s,a,s') + \gamma \sum_i \alpha_i \sum_j \mathrm{P}(V_{ij}) \cdot V_j^{(i)}(s') & \\ & \text{if } s' \text{ is terminal} \end{cases} \tag{5}$$

Considering individual agents raises the possibility of giving the acting agent reward based not on the expected sum of returns of the other agents (as in Eq. (5)), but by incorporating some notion of "fairness". For example, we could consider the expected return of the agent who would be worst-off. This is inspired by the maximin (or "Rawlsian") social welfare function, which measures social welfare in terms of the utility of the worst-off agent [see, e.g., 15].

3.3 Using Information about Options

In Eq. (1), we used a distribution over value functions to provide some sense of what agents might do in the future and the expected return achievable from different states. Here we consider those agents to instead be endowed with a set of *options* [20] that could reflect particular skills or tasks they are capable of realizing, and we use a distribution over such options to characterize what might be executed by future agents. This easily allows us to identify individual skills and could give the acting agent the ability to contemplate preservation of skills or tasks, if desirable.

An option is a tuple $\langle \mathcal{I}, \pi, \beta\rangle$ where $\mathcal{I} \subseteq S$ is the initiation set, π is a policy, and β is a termination condition (formally, a function associating each state with a termination probability) [20]. The idea is that an agent can follow an option by starting from a state in its initiation set $\mathcal{I}$ and following the policy π until it terminates. Options provide a form of macro action that can be used as a temporally abstracted building block in the construction of policies. Options are often used in Hierarchical RL: an agent can learn a policy to choose options to execute instead of actions. Here we will use options to represent skills or tasks that other agents in the environment may wish to perform.

Suppose we have a set O of initiation sets of options, and a probability function $\mathrm{P}(\mathcal{I})$ giving the probability that $\mathcal{I}$ is the initiation set of the option whose execution will be attempted after the acting agent reaches a terminating state. To try to make the acting agent act so as to allow the execution of that option, we can modify the acting agent's reward function r_1, yielding the new reward function r_{option} below.

$$r_{\text{option}}(s,a,s') = \begin{cases} \alpha_1 \cdot r_1(s,a,s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s,a,s') + \gamma \cdot \alpha_2 \displaystyle\sum_{\mathcal{I}\in O} \mathrm{P}(\mathcal{I}) \cdot \mathbb{I}_{\mathcal{I}}(s') & \text{if } s' \text{ is terminal} \end{cases} \tag{6}$$

where $\mathbb{I}_{\mathcal{I}} : S \to \{0, 1\}$ is the indicator function for $\mathcal{I}$ as a subset of S, i.e., $\mathbb{I}_{\mathcal{I}}(s) = \begin{cases} 1 & \text{if } s \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases}$.

Note that if O is finite and P is a uniform distribution, then the auxiliary reward given by r_{option} will be proportional to how many options in O can be started in the terminal state. Also note that if O represents a set of options that could have been initiated in the start state of the *acting agent*, we can interpret r_{option} as encouraging *preservation* of the capabilities of other agents, which is more related to the idea of side effects.

The hyperparameters α_1 and α_2 determine how much weight is given to the original reward function and to the ability to initiate the option. Given a fixed value of α_1 (and ignoring the discount factor), the parameter α_2 could be understood as a "budget", indicating how much negative reward the acting agent is willing to endure in order to let the option get executed.

We could consider variants of this approach that further distinguish options with respect to the agent(s) that can realize them, or by specific properties of the options, such as what skill they realize, and we could use such properties to determine how each α is weighted. For example, the acting agent could negatively weight options which terminate in states that the acting agent doesn't like. To illustrate, imagine that the option's execution involves a deer eating the plants in the vegetable garden. The acting agent might want to prevent option execution by building a fence.

Finally, if we had a distribution over pairs $\langle \mathcal{I}, V \rangle$ – consisting of an option's initiation set and a value function associated with that – then this can be captured by the following augmentation:

$$r'_{\text{option}}(s, a, s') = \begin{cases} \alpha_1 \cdot r_1(s, a, s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s, a, s') + \alpha_2 \sum_{\langle \mathcal{I}, V \rangle \in O} \mathsf{P}(\langle \mathcal{I}, V \rangle) \cdot \mathbb{I}_{\mathcal{I}}(s') \cdot V(s') & \\ & \text{if } s' \text{ is terminal} \end{cases} \tag{7}$$

This is much like r_{option} but has an extra factor of $V(s')$ in the sum in the second case.

4 EXPERIMENTS

In the previous section we presented different formulations of reward functions that allow RL agents to contemplate the impact of their actions on the welfare and agency of others. Here, we present quantitative and qualitative results relating to these formulations.

In all the experiments, policies are learned using Q-learning [25]. To aid exposition, we consider simple distributions over future value functions, in which the acting agent is certain of what the future value function is (or, in Figure 3, only considers a small number of possibilities). Experimental details can be found in the supplementary material. Code is available at https://github.com/praal/beconsiderate.

4.1 Quantitative Experiments

In our first set of experiments, we compare one of our formulations (Eq. (2)) of a considerate RL agent against two baselines. We illustrate that by considering others, the acting agent avoids causing negative side effects for them, and in some scenarios, yields positive side effects. Second, we illustrate the effect of the caring coefficient on the agent's behaviour and on other agents' reward.

4.1.1 The Impact of Considering Others. We explore how our choice of reward augmentation method affects the acting agent and the agent that goes next. We use a kitchen environment where agents aim to collect different ingredients from the fridge or shelves, and prepare a meal. Each agent, when it performs any action, gets -1 reward. We designed four different scenarios to illustrate properties of our approach. The results are shown in Table 1. We use a *step difference* metric – the difference between the number of steps each agent required to execute their policy as compared to what they would have required if they had tried to complete their task from the initial state without considering other agents.

Baselines: We compare our method, which is defined in Eq. (2) (with $\alpha_1 = \alpha_2 = 1$), with two reward augmentation baselines: not augmenting the reward, and a method based on Krakovna et al. [11]'s approach. The Krakovna-style baseline uses the same Eq. (2) to augment the rewards, except that the future value functions considered are always possible future value functions of the *acting agent itself* (as if it were trying to accomplish the tasks of other agents). So if other agents have differing abilities, those abilities are ignored in the Krakovna-style model. (Note that this method does not incorporate Krakovna et al. [11]'s notion of a "reference state" and may incentivize positive side effects in some cases, as our own method does.)

The first experiment (Salad) shows a scenario where the acting agent and next agent have the same abilities, and as such our approach and the Krakovna-style baseline both avoid negative side effects and behave identically. The next experiments (Peanut and Salt) show that our approach, taking into account differing agent abilities, is sometimes more effective at avoiding negative side effects than either baseline. The last experiment (Cookies) shows how our approach (and the Krakovna-style baseline) can cause positive side effects for the next agent. Each of the experiments is described in more detail below.

In Salad, the acting agent needs to collect the ingredients from the fridge. If it doesn't consider side effects, it doesn't close the fridge and ruins all the remaining ingredients, preventing the next agent from completing its task. By considering future tasks (whether another agent's or its own), the acting agent learns to take an extra step to close the fridge. In Peanut, preparing food contaminates the environment, and for the next agent to cook requires that the environment first be cleaned (taking one step), or disinfected (taking two steps) if the next agent has allergies. Only our approach takes the two extra steps to disinfect the kitchen because it considers that the other agent (unlike itself) has allergies. In Salt, if the acting agent does not put the salt shaker back on the shelves, the next agent can't complete its task. By considering future agents (in the Krakovna-style baseline and our approach) this side effect is avoided. However, the acting agent is tall and may put the salt on the top shelf (making it take longer for the next, shorter, agent to get it) if it considers that the next agent will be itself, as in the Krakovna-style baseline. Finally, in Cookies, the next agent's task is to bake cookies in the oven. Two steps are required to preheat the oven (turning on the oven and waiting). By considering the future task of the next agent, the acting agent (who was not using

Table 1: Comparison of reward augmentation methods for acting and subsequent agents. Each row reflects a different method. Each column depicts results for a different experimental scenario. Each entry pair depicts a "step difference" required by the acting agent and the subsequent acting agent (next). The "step difference" is the difference between the number of steps the agent required to execute their policy as compared to what they would have required if they had tried to complete their task from the initial state without considering other agents. ∞ indicates the task was unachievable.

	Salad	Peanut	Salt	Cookies
Method	acting agent, next	acting, next	acting, next	acting, next
Non-augmented reward	0, ∞	0, ∞	0, ∞	0, 0
Based on Krakovna et al.	1, 0	1, ∞	1, 1	1, -2
Our approach [Eq. 2]	1, 0	2, 0	1, 0	1, -2

the oven) can turn on the oven to start preheating it, and save the next agent two steps.

4.1.2 Varying the Caring Coefficient. In this experiment, we investigate the effect of choosing different caring coefficients (α_1 and α_2) in Eq. (2) by monitoring the average reward collected by each of the agents in the Craft-World Environment.

Craft-World Environment We consider a Minecraft$^{\text{TM}}$ inspired gridworld environment depicted in Figure 1.

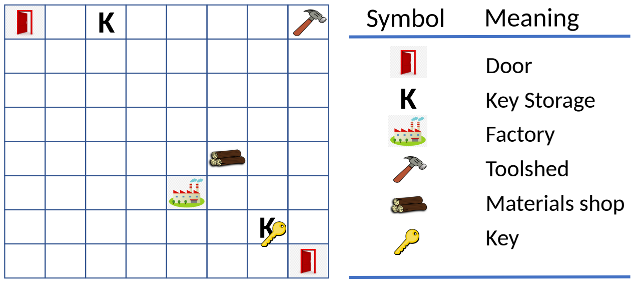

Figure 1: Craft-World Environment

Agents in this environment use tools and materials to construct artifacts such as boxes. Tools are stored in a toolshed in the upper right corner of the grid environment. Agents enter and exit the environment through doors in the upper left and lower right. They must collect materials and bring them to the factory for assembly. The factory requires a key for entry, and there is only one key, which can only be stored in one of two locations (denoted by K). When considering other agents, the acting agent may elect to place the key in a position that is convenient for others, or may help other agents by anticipating their need for tools or resources and collect them on their behalf.

In the experiment we ran, agents enter at the top left door, tasked with making a box. The first (caring) agent learns a policy following Eq. (2). The second, subsequent acting agent, follows a fixed policy designed to optimize its own reward.

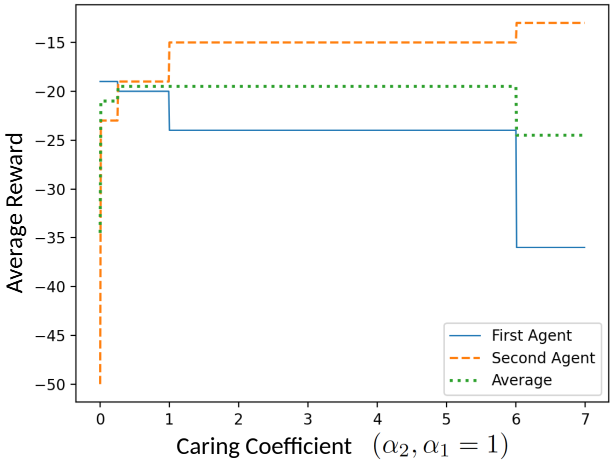

Figure 2: Effect of caring coefficients in the Craft-World environment. Increasing α_2 above 0, at first the agent changes its behaviour with little or no cost and this is significantly beneficial to the second agent. However, by increasing α_2 further, the first agent incurs high cost, yielding only a small benefit to the second agent.

Figure 2 shows the reward that each agent gets (after training) as we vary the caring coefficient α_2. It also shows their average. When $\alpha_2 = 0$, the first agent is oblivious to others and exits the environment without returning the key, precluding the second agent from making a box. When $\alpha_2 > 0$, the agent becomes more considerate and returns the key on its way to the exit. As we increase the value of α_2, the first agent is incentivized to help the second agent, eventually (to its detriment) carrying extra materials to the factory for the second agent, garnering negative reward for this hard work and also, interestingly, lowering the average reward of the two agents. Too much caring does not yield maximal reward for the collective!

4.2 Qualitative Experiments

In this section we share the results of qualitative experiments that serve to illustrate how different reward function augmentations and settings of the caring coefficients lead to different behaviours. We consider reward augmentations using different definitions of $F(\mathcal{V}, \mathrm{P}, s')$, the agent-distinguishing variant from subsection 3.2, and the options-based formulation from subsection 3.3.

4.2.1 Optimal Behaviours under Different Reward Augmentations. Figure 3 illustrates the difference between Equations (2), (3) and (4) and the Krakovna-style baseline. In this experiment, the goal of the agents is to play with the doll and leave it somewhere in the environment for the next agent, and then exit the environment from their entry point; the agents get -1 reward for each step. There are six agents (circles 1-6 in Figure 3) in the environment with the same goal. They are shown at their individual entry points. Agents enter the environment separately; the acting agent is agent 1. In this scenario, $\alpha_1 = 1$, and $\alpha_2 = 10$. If we augment the acting agent's reward according to Eq. (2) (where the distribution of value functions is a uniform distribution over the optimal value function for each agent w.r.t. the goal of playing with the doll), the optimal policy is to place the doll as close as possible to the majority of the agents. If we use Eq. (3) the optimal policy is to place the doll so as to minimize the distance to the furthest agent. Finally, if we use Eq. (4) the optimal policy is to leave the doll where it is, because moving it causes negative side effects for agent 6. However if we use the approach based on Krakovna et al., the optimal policy is to leave the doll at agent 1's exit/entry point, so that the doll would be conveniently located for agent 1 if it were to re-enter. Finally, if we use non-augmented reward the agent does not have an incentive to place the doll in the environment and leaves with the doll (not shown in figure).

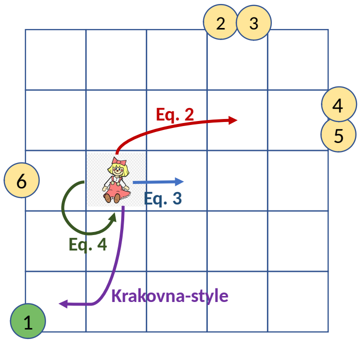

Figure 3: Example behaviour that illustrates different augmentations of the reward function according to Equations (2), (3), (4) and the Krakovna-style baseline. Six agents (circles 1-6) are shown at their individual entry points. Agent 1 is the acting agent ($\alpha_1 = 1$ and $\alpha_2 = 10$). All agents wish to play with the doll, subsequently exiting from their entry points. Agent 1 learns a policy to play with the doll and leave it for others. Each arrow points to where an optimal policy could leave the doll when Agent 1 receives auxiliary reward according to the approach labelling the arrow.

4.2.2 Using Different Caring Coefficients for Different Agents. A second experiment, depicted in Figure 4, illustrates the difference in treatment of agents through the choice of caring coefficients when using the modified reward in Eq. (5). There are 3 agents that want to get to the exit from the starting point which is at the bottom left; they get -1 reward in each time step. Agents enter the environment separately, and the acting agent is agent 1. Agent 2 has a garden on the shortest path and gets very upset (-20 reward) if someone passes through the garden. The acting agent cares about agent 3 and itself in an equal amount ($\alpha_1 = \alpha_3 = 1$). In the first case we consider, agent 1 is oblivious to agent 2 ($\alpha_2 = 0$) and follows the shortest path to the exit, passing through the garden (the solid red path in the left figure). In the second case, agent 1 cares about agent 2 a little ($\alpha_2 = 1$) and takes the longer path (dotted blue path in the left figure) to avoid passing through the garden. In the third case, agent 1 cares about agent 2 a lot ($\alpha_2 = 10$) and even though there is a reward of -50, agent 1 builds a fence to protect agent 2's garden (dashed green path in the right figure), which also makes agent 3 take the longer path with extra steps.

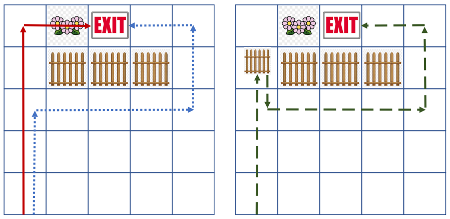

Figure 4: Different caring coefficients lead to different paths. This experiment shows how an inconsiderate agent will walk over another agent's garden (solid red line). With a little consideration, it will walk around (dotted blue line) and with significant consideration it will go to the expense of building a fence to keep it and others out of the garden (dashed green line in the right figure).

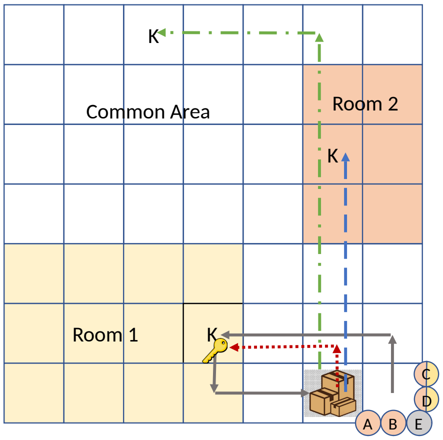

Figure 5: Example behaviour that illustrates the effect of α_2 with options, following Eq. (6)**.**

4.2.3 Using Options. Here, we give an illustration of the options-based reward function in Eq. (6) and investigate the behaviour of the acting agent by fixing α_1 and changing α_2. Figure 5 depicts a grid-world environment composed of a small mail room in the lower right corner (depicted by the pile of packages), two designated rooms (Room 1 and Room 2), and a Common Area. The mail room requires a key to open it. The key has to be stored at a 'K' location.

In addition to the acting agent, there are 5 other agents (A, B, C, D, E) that may use the environment in the future. The acting agent has access to all areas of the grid, but the other agents' access is restricted. Room 1 is only accessible to agents C and D, while

Room 2 is accessible to agents A, B, C, and D, but not E. All agents can access the Common Area. Agents A, B, C, D, and E each have an option that enables them to collect a package from the mail room, but because the key can only be stored in one of the three designated 'K' locations, the initiation sets for agents' options differ, based on their personal room access.

The acting agent (not depicted) aims to pick up the key, collect a package, and place the key at one of the 'K' locations. It realizes a reward of -1 at each time step. Since agents A, B, C, D, and E all need the key to execute their option, but are restricted in their access to certain rooms where the key could be stored, the acting agent will differ in its behaviour depending on how much it is willing to inconvenience itself (incur -1 for each step) to leave the key in a location that is accessible to other agents.

The coloured and patterned lines in Figure 5 depict the different policies learned by the acting agent under different settings of α_2 with fixed $\alpha_1 = 1$. The distribution over initiation sets of options, $\mathrm{P}(\mathcal{I})$, is set to a uniform distribution (the acting agent is uncertain which of agents A, B, C, D, and E will attempt to execute its option). All of the policies start with the acting agent getting the key and then going to the package (following the solid grey line in the figure), but they differ in what happens after that. By setting $\alpha_1 = \alpha_2 = 1$ the acting agent puts the key in Room 1 (following the dotted red line) as this is the closest place to leave the key. Recall that Room 1 is only accessible to agents C and D (40% of the agents). When α_2 is changed such that $\alpha_2 = 5$, the acting agent cares more about the other agents and puts the key at the 'K' location in Room 2 (following the dashed blue line) where 80% of the possible future agents can execute their option, and by setting $\alpha_2 = 25$ the acting agent incurs some personal hardship and puts the key at the far-away 'K' location in the Common Area (following the dot-dash-patterned green line), so that all the agent can execute their options.

5 RELATED WORK

The work presented here is related to several bodies of work, including work on AI safety and (negative) side effects, and work on empathetic planning. In subsection 5.1, we discuss the relation of our approach to Krakovna et al. [11], the most closely related work. This is followed by discussion of other related work.

5.1 Relation to the Future Task Approach [11]

We consider the relation of some of our formulations to the "future task" approach to avoiding side effects from Krakovna et al. [11]. We'll see that under some conditions, their approach can be seen as a special case of ours (using either Eq. (2) or Eq. (5)), and also how our Eq. (4) incorporates their notion of a *reference state*.

Krakovna et al. proposed modifying the agent's reward function to add an auxiliary reward based on its own ability to complete possible future tasks. A "task" corresponds to a reward function which gives reward of 1 for reaching a certain goal state, and 0 otherwise. In their simplest definition (not incorporating a baseline), the modified reward function was

$$r_K(s,a,s') = \begin{cases} r_1(s,a,s') + \beta(1-\gamma)\sum_i F(i)V_i^*(s') & \text{if } s' \text{ is not terminal} \\ r_1(s,a,s') + \beta\sum_i F(i)V_i^*(s') & \text{if } s' \text{ is terminal} \end{cases}$$

where r_1 is the original reward function, F is a distribution over tasks, V_i^* is the optimal value function for task i (when completed by the single agent itself), and β is a hyperparameter which determines the how much weight is given to future tasks. They interpret $1-\gamma$ (where γ is the discount factor) as the probability the agent will terminate its current task and switch to working on the future task, which leads to the $(1-\gamma)$ factor in the case where s' is not terminal.

This is similar to (and inspired) our approach, though for r_K the value functions are restricted to be possible value functions for the agent itself (and so depend on what actions the agent itself can perform). In contrast, in our approach, we consider value functions that may belong to different agents with different abilities. Additionally, they assume the value functions are optimal. Below we show how under some conditions, our approach generalizes theirs.

In the case where γ (the discount factor) is 1, r_K simplifies so that $r_K(s,a,s') = r_1(s,a,s')$ if s' is not terminal. Meanwhile, our Eq. (2) (substituted into Eq. (1)), in the case where $\gamma = 1$, can be rewritten as

$$r_{\text{value}}(s,a,s') = \begin{cases} \alpha_1 \cdot r_1(s,a,s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s,a,s') + \alpha_2 \sum_{V\in\mathcal{V}} \mathrm{P}(V)\cdot V(s') & \text{if } s' \text{ is terminal} \end{cases}$$

Observe that if $\gamma = 1$, $\alpha_1 = 1$, $\alpha_2 = \beta$, and $\mathrm{P}(V) = \sum\{F(i) \mid V_i^* = V\}$ then $r_K = r_{\text{value}}$. So in the undiscounted setting r_K is a special case of r_{value}.

5.1.1 Relationship to Use of Reference States. In Krakovna et al. [11]'s more complicated version of the augmented reward function, the auxiliary reward (r_{aux}) that is added to r_1 depends on a *reference state* s'_t (sometimes also called a "baseline state"):

$$r_{\text{aux}}(s',s'_t) = \begin{cases} \beta(1-\gamma)\sum_i F(i)V_i^*(s',s'_t) & \text{if } s' \text{ is not terminal} \\ \beta\sum_i F(i)V_i^*(s',s'_t) & \text{if } s' \text{ is terminal} \end{cases}$$

Their definition of $V_i^*(s',s'_t)$ is somewhat complicated, but (as they note) when the environment is deterministic it is equal to $\min(V_i^*(s'), V_i^*(s'_t))$.

Recall that our Eq. (4) (substituted into Eq. (1)) is

$$r_{\text{value}}(s,a,s') = \begin{cases} \alpha_1 \cdot r_1(s,a,s') & \text{if } s' \text{ is not terminal} \\ \alpha_1 \cdot r_1(s,a,s') + \gamma\cdot\alpha_2\cdot\sum_{V\in\mathcal{V}} \mathrm{P}(V)\cdot\min(V(s'),V(s_0)) & \\ & \text{if } s' \text{ is terminal} \end{cases}$$

So, if $\gamma = 1$, $\alpha_1 = 1$, $\alpha_2 = \beta$, $\mathrm{P}(V) = \sum\{F(i) \mid V_i^* = V\}$, and the environment is deterministic, that's equal to Krakovna et al. [11]'s modified reward function with the initial state as a reference state.

Krakovna et al. [11] actually used a more complicated reference state. We leave it to future work to incorporate other reference states into our approach.

5.1.2 Considering Different Agents. Recall that in Eq. (5) we introduced r'_{value}, an augmented reward function which considered the possible value functions of different agents. r'_{value} can be compared to the reward r_K from Krakovna et al. in a different way.

Consider the case where $\gamma = 1$, $\alpha_1 = 1$, and $\alpha_i = 0$ for $i > 1$ (so only the first agent's future reward is considered – all other agents are ignored). We can then simplify Eq. (5) to

$$r'_{\text{value}}(s,a,s') = \begin{cases} r_1(s,a,s') & \text{if } s' \text{ is not terminal} \\ r_1(s,a,s') + \sum_j \mathrm{P}(V_{1j}) \cdot V_j^{(1)}(s') & \text{if } s' \text{ is terminal} \end{cases}$$

Observe that this is equal to $r_K(s,a,s')$ where $\beta = 1$ (and $\gamma = 1$ again) with an appropriate choice of the distributions F and P (e.g., one where $V_i^{(1)} = V_i^*$ and $F(i) = \mathrm{P}(V_{1i})$ for each i).

5.2 Other Related Work

Prior to Krakovna et al. [11], Krakovna et al. [10] considered a number of approaches to avoiding side effects in which the agent's reward function $r(s_t, a_t, s_{t+1})$ is modified to include a penalty for impacting the environment, so that the new reward function is of the form $r(s_t, a_t, s_{t+1}) - \beta \cdot d(s_{t+1}, s'_{t+1})$ where β is a hyperparameter (indicating how important the penalty is), $d(\cdot,\cdot)$ is a "deviation" measure, and s'_{t+1} is a reference state to compare against. They considered several possible reference states: the initial state, the state resulting from performing no-op actions from the initial state, or the state resulting from performing a no-op action in s_t. While the abstract notion of a deviation measure is broad enough to support consideration of other agents, all the explicit deviation measures Krakovna et al. suggested were defined in terms of how the agent's *own* ability to reach states is affected.

Attainable Utility Preservation (AUP) [22, 23] is another similar approach to side effects. The agent's reward is modified, given a set $\mathcal{R}$ of other reward functions, to penalize actions that change the agent's *own* ability to optimize for the functions in $\mathcal{R}$. By using a set $\mathcal{R}$ of randomly selected reward functions, Turner et al. [23] were able to have an agent avoid side effects in some simple problems. Later, Turner et al. [22] considered using AUP with just a single reward function (computed using an autoencoder), and showed that that worked well in avoiding side effects in the more complicated SafeLife environment [24]. It would be interesting to explore consideration of other agents in such complicated environments.

Also related to our work is recent work that aspires to build agents that act empathetically in the environment to explain, recognize the goals of, or act to assist others [e.g., 6, 16–18], but this body of work assumes the existence of a model. In the context of RL, Bussmann et al. [3] proposed "Empathetic Q-learning", an RL algorithm which learns not just the agent's Q-function, but an additional Q-function, Q_{emp}, which gives a weighted sum of the agent's value from taking an action, and the value that another agent will get. By taking actions to maximize the Q_{emp}-value, the first agent may be able to avoid some side effects that involve harming the other agent. The value the other agent will get is approximated by considering what reward the first agent would get, if their positions were swapped. Note that the approach assumes that the agent being "empathized" with gets at least some of the same rewards (while our approach makes no assumption about agents being similar).

Du et al. [5] considered the problem of having an AI assist a human in achieving a goal. They proposed an auxiliary reward based on (an estimate of) the human's *empowerment* in a state. Empowerment in an information-theoretic quantity that measures ability to control the state. In some cases having an agent try to maximize (approximate) empowerment outperformed methods that tried to infer what the human's goal was and help with that. However, an abstract measure like empowerment might be influenced by the presence of irrelevant features that humans aren't actually interested in controlling.

Finally, multi-agent reinforcement learning (MARL), in which multiple artificial agents learn how to act together (see, e.g., the surveys [2, 26]) broadly shares motivation with our work. A critical distinction is that while we are doing RL in the presence of multiple agents, only the acting agent is engaged in RL, while other agents are assumed to be following existing fixed policies. A similar observation can be made to contrast our work with cooperative multi-agent systems (e.g., Dafoe et al. [4]).

6 CONCLUDING REMARKS

Incompletely specified objectives will endure for at least as long as humans have a hand in objective specifications, continuing to present threats to AI acting safely [e.g., 1]. In this paper we have put forward that acting safely should include contemplation of the impact of an agent's actions on the wellbeing and agency of others.

Providing agents with a means to *be considerate* is important, and it can be done without the need for coordination. We have studied how an agent can learn to be considerate of others via RL, given some non-specific general knowledge of (potentially hypothetical) agents that operate fixed policies in an environment.

The work presented here provides a pragmatic stance to building systems that have the potential to be benevolent without requiring multiple agents to agree to cooperate. However, like so many AI advances, there are potential malicious or unintended uses of the ideas presented here. In particular, in the same way that the caring coefficient can be set to attend to and to help others, it could be set to attempt to effect change that purposefully diminishes others' wellbeing and/or agency. The caring coefficient also raises the possibility for differential treatment of agents, which presents opportunities to systematize notions of fair (and unfair) decision making, as briefly noted in Section 3.

Limitations. We identify several limitations of our work. If the distribution over value functions (or options) used by our approach is inaccurate, that may incentivize behavior that fails to accommodate others or is actually harmful. A further limitation of our approach is that, as previously noted, in some cases our augmented reward functions can introduce a (probably undesirable) incentive for the acting agent to never reach a terminal state, to avoid being penalized for what effect its actions have had on others. Finally, a general problem when dealing with the utilities of different agents is that different agents may gain rewards at very different scales. This is a classic philosophical problem, which has also been noted in the context of AI safety [13]. One might try to normalize the values by setting the caring coefficients, but in general it may be difficult to determine appropriate values for them.

ACKNOWLEDGMENTS

We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada CIFAR AI Chairs Program, and Microsoft Research. The third author acknowledges funding from ANID (Becas Chile). This work was done while he was a graduate student at the University of Toronto.

REFERENCES

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. (2016). arXiv:1606.06565

[2] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (2008), 156–172.

[3] Bart Bussmann, Jacqueline Heinerman, and Joel Lehman. 2019. Towards Empathic Deep Q-Learning. In *Proceedings of the Workshop on Artificial Intelligence Safety 2019 co-located with the 28th International Joint Conference on Artificial Intelligence, AISafety@IJCAI (CEUR Workshop Proceedings, Vol. 2419)*. CEUR-WS.org, Aachen. http://ceur-ws.org/Vol-2419/paper_19.pdf

[4] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. 2021. Cooperative AI: machines must learn to find common ground. *Nature* 593 (2021), 33–36. https://doi.org/10.1038/d41586-021-01170-0

[5] Yuqing Du, Stas Tiomkin, Emre Kiciman, Daniel Polani, Pieter Abbeel, and Anca Dragan. 2020. AvE: Assistance via Empowerment. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc.

[6] Richard G. Freedman, Steven J. Levine, Brian C. Williams, and Shlomo Zilberstein. 2020. Helpfulness as a Key Metric of Human-Robot Collaboration. (2020). arXiv:2010.04914

[7] León Illanes, Xi Yan, Rodrigo Toro Icarte, and Sheila A. McIlraith. 2020. Symbolic Plans as High-Level Instructions for Reinforcement Learning. In *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling*. AAAI Press, 540–550.

[8] Toryn Q. Klassen and Sheila A. McIlraith. 2021. Planning to Avoid Side Effects (Preliminary Report). In *IJCAI Workshop on Robust and Reliable Autonomy in the Wild (R2AW)*. http://rbr.cs.umass.edu/r2aw/papers/R2AW_paper_15.pdf

[9] Toryn Q. Klassen, Sheila A. McIlraith, Christian Muise, and Jarvis Xu. 2022. Planning to Avoid Side Effects. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*. To appear.

[10] Victoria Krakovna, Laurent Orseau, Miljan Martic, and Shane Legg. 2019. Penalizing Side Effects using Stepwise Relative Reachability. In *Proceedings of the Workshop on Artificial Intelligence Safety 2019 co-located with the 28th International Joint Conference on Artificial Intelligence, AISafety@IJCAI 2019 (CEUR Workshop Proceedings, Vol. 2419)*. CEUR-WS.org, Aachen. http://ceur-ws.org/Vol-2419/paper_1.pdf

[11] Victoria Krakovna, Laurent Orseau, Richard Ngo, Miljan Martic, and Shane Legg. 2020. Avoiding Side Effects By Considering Future Tasks. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc.

[12] Jim Lebans. 2020. The threat from AI is not that it will revolt, it's that it'll do exactly as it's told. CBC Radio. URL https://www.cbc.ca/radio/quirks/apr-25-deepwater-horizon-10-years-later-covid-19-and-understanding-immunity-and-more-1.5541299/the-threat-from-ai-is-not-that-it-will-revolt-it-s-that-it-ll-do-exactly-as-it-s-told-1.5541304.

[13] Stuart Russell. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Publishing Group, New York.

[14] Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein. 2020. A Multi-Objective Approach to Mitigate Negative Side Effects. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. 354–361. https://doi.org/10.24963/ijcai.2020/50

[15] Amartya Sen. 1974. Rawls Versus Bentham: An Axiomatic Examination of the Pure Distribution Problem. *Theory and Decision* 4, 3-4 (1974), 301–309. https://doi.org/10.1007/BF00136651

[16] Maayan Shvo. 2019. Towards Empathetic Planning and Plan Recognition. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 525–526.

[17] Maayan Shvo, Toryn Q. Klassen, and Sheila A. McIlraith. 2020. Towards the Role of Theory of Mind in Explanation. In *Explainable, Transparent Autonomous Agents and Multi-Agent Systems: Second International Workshop, EXTRAAMAS 2020*. Springer-Verlag, Berlin, Heidelberg, 75–93. https://doi.org/10.1007/978-3-030-51924-7_5

[18] Maayan Shvo and Sheila A McIlraith. 2019. Towards empathetic planning. (2019). arXiv:1906.06436

[19] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). MIT Press, Cambridge, MA. http://incompleteideas.net/book/the-book.html

[20] Richard S. Sutton, Doina Precup, and Satinder P. Singh. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence* 112, 1-2 (1999), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1

[21] Alex Turner. 2019. Reframing Impact. Blog post, https://www.lesswrong.com/s/7CdoznhJaLEKHwvJW.

[22] Alex Turner, Neale Ratzlaff, and Prasad Tadepalli. 2020. Avoiding Side Effects in Complex Environments. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc.

[23] Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. 2020. Conservative Agency via Attainable Utility Preservation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES '20)*. Association for Computing Machinery, New York, NY, United States, 385–391. https://doi.org/10.1145/3375627.3375851

[24] Carroll Wainwright and Peter Eckersley. 2020. SafeLife 1.0: Exploring Side Effects in Complex Environments. In *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI 2020) co-located with 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. 117–127. http://ceur-ws.org/Vol-2560/paper46.pdf

[25] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-Learning. *Machine Learning* 8 (1992), 279–292. https://doi.org/10.1007/BF00992698

[26] Kaiqing Zhang, Zhuoran Yang, and Tamer Basar. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. (2021). arXiv:1911.10635

[27] Shun Zhang, Edmund H. Durfee, and Satinder P. Singh. 2018. Minimax-Regret Querying on Side Effects for Safe Optimality in Factored Markov Decision Processes. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*. 4867–4873. https://doi.org/10.24963/ijcai.2018/676