Path-specific Objectives for Safer Agent Incentives

Sebastian Farquhar[†‡], **Ryan Carey**[†], **Tom Everitt**[‡]

[†]University of Oxford, [‡]DeepMind

Abstract

We present a general framework for training safe agents whose naive incentives are unsafe. As an example, manipulative or deceptive behaviour can improve rewards but should be avoided. Most approaches fail here: agents maximize expected return by any means necessary. We formally describe settings with 'delicate' parts of the state which should not be used as a means to an end. We then train agents to maximize the causal effect of actions on the expected return which is *not* mediated by the delicate parts of state, using Causal Influence Diagram analysis. The resulting agents have no incentive to control the delicate state. We further show how our framework unifies and generalizes existing proposals.

Artificial agents can have unsafe incentives to influence parts of their environments in unintended ways. For example, content recommendation systems can achieve good performance by manipulating their users to develop more predictable preferences instead of catering to their tastes directly (Russell 2019). These incentives can be instrumental: indirectly achieving what the system's designers asked for but did not intend (Everitt et al. 2021a).

In these cases, it is hard to just pick a better reward function. Imagine the unenviable task of writing down which user-preferences are desirable! We would rather ensure that the agent has no systematic incentive to manipulate people's preferences at all—as opposed to an agent with an incentive to encourage its users to have the 'right' kind of preference. In this setting, the users' preferences are what we call *delicate* state: a part of the environment which is hard to define a reward for and vulnerable to deliberate manipulation.

Even when part of the state-space is delicate, other parts might be tractable. For example, we might know exactly how we want to price media-bandwidth consumption driven by our recommendations. This paper provides a framework for designing agents to act safely in environments where parts of the state-space are delicate and other parts are not, under assumptions which permit causal effects estimation.

We show how to train agents in a way that removes incentives to control delicate state. We use causal influence diagrams (CIDs) which can formally express instrumental control incentives (Everitt et al. 2021a). We show that one can remove the instrumental control incentive over delicate state by training agents to maximize the path-specific causal effect (Pearl 2001) of their actions on the reward following paths which are not mediated by the delicate state. Moreover, we show how a diverse set of previous proposals for safe agent design can be motivated by these principles. In this way, we unify and generalize approaches from topics such as reward tampering (Uesato et al. 2020; Everitt et al. 2021b), online reward learning (Armstrong et al. 2020), and auto-induced distributional shift (Krueger, Maharaj, and Leike 2020). At the same time, we show how these methods depend on assumptions about the state-space which have not previously been acknowledged. We highlight the opportunities and dangers of these approaches empirically in a content recommendation environment from Krueger, Maharaj, and Leike (2020). Our main contributions are:

- We formalize the problem of delicate state as a complement to reward specification (§1);
- We propose path-specific objectives (§4);
- We show this generalizes and unifies prior work (§5).

1 The Problem of Delicate State

Delicate state is a tool for framing safe agent design. When a state is *subtle* and *manipulable* we call it *delicate*:

Subtle Hard to specify a reward for.

Manipulable Vulnerable to motivated action—intentional actions can have bad outcomes.

Jointly, these are dangerous: it is hard to say what we want for the state and it is easy for influence on the state to have bad consequences. A person's political beliefs might be an example of such a state. The current toolbox for safe agent design mostly tries to attack subtlety directly by finding a better way to specify the reward. This is not our approach—instead we aim to remove any incentive for the agent to control the delicate part of state-space.

If a part of state-space is delicate then having a control incentive over it is dangerous. But in order for removing it to lead to safe outcomes a third condition is needed:

Stable Robust against unmotivated action—side-effects are unlikely to be bad.

Stability entails that an agent with no systematic incentive to influence the state—but which still influences the state and may produce side-effects over it—is safe. As a metaphor for a system which is both manipulable and stable, consider a puzzle box: apply the right pressure to the right spots and it comes apart easily, but you can fumble randomly or even use it as a mallet and it will not open. We might hypothesise that a person's political beliefs are relatively stable—after all, most people are able to think critically and independently in the presence of influences in many directions.

We define delicate state within the context of a factored Markov decision process (MDP) characterized by transition function, reward function, action-space, and, unlike standard MDPs, a state-space factored into a robust state $s \in \mathcal{S}$ and a delicate state $z \in \mathcal{Z}$ such that the overall state is $\{s, z\} \in \mathcal{S} \cup \mathcal{Z}$. The transition function therefore maps s, z and action ($a \in \mathcal{A}$) onto the succeeding state s', z' and the reward function maps s, s', z, z', a onto a reward $r \in \mathcal{R}$.

Subtlety

We consider five cases that make a state subtle, and thereby potentially delicate. In each, it is not enough to simply pick a reward function that does not explicitly depend on z because instrumental incentives emerge when Z and S interact.

Not Ordered There might not be a well-defined ethical ranking of different values of Z or it might be unethical to codify a ranking. For example, it may be unethical for a system to systematically influence user's beliefs, preferences, and political views (Burr, Cristianini, and Ladyman 2018). Content recommender systems often interact with subtle human states (Kramer, Guillory, and Hancock 2014).

Vague Even if an ordering is possible, we may not trust our agent-designers to describe it. Reward modelling (Leike et al. 2018) or alternative work on reward specification (Christiano et al. 2017) seeks to attack this source of subtlety directly, while our approach tries to side-step it.

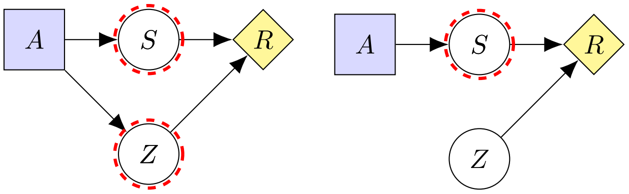
Unenforceable Even with a well-specified reward, we may be unable to enforce it, for example, if Z is the physical implementation of the reward function then a modified Z might no longer punish the agent for having changed it (Amodei et al. 2016; Everitt et al. 2021b).

Illegal The law might ban a well-specified and enforceable reward. For example, if Z is the market-price of an asset, deliberately influencing it may be market manipulation.

Structural We might *choose* not to reward based on Z in order to construct an ecosystem of agents. For example, Z might be a performance measure of our agent which is used by another agent. Alternatively, the system might have deliberately demarcated roles, much as judges may be asked to apply the law as it stands, ignoring political consequences.

Manipulability and Stability

A manipulable state is one where deliberate or intentional actions can easily bring about harm. We adopt a notion of 'deliberateness' built on incentives—assuming that an agent which has an incentive over Z and influences Z does so 'deliberately'. In the appendix we link this to previous analyses of 'manipulation' (Kenton et al. 2021). This approach, described more formally below, has the advantage of being agnostic to the specific implementation of the agent (models, algorithms, etc.). We contrast this with instability—where *non-deliberate* actions can easily bring about harm.



(a) CID with ICI on S and Z (b) CID removing ICI on Z

Figure 1: Causal Influence Diagrams in a setting with delicate state. Blue square is decision, yellow diamond is utility. Black arrows show causal influence. Dashed red circles show instrumental control incentive (ICI). (a) the agent has ICI over $\{S, Z\}$. (b) Z does not influence R so no ICI on Z.

We can draw parallels to 'safety' and 'security' in cybersecurity. A secure system is one that is robust to malicious actors (not manipulable), while a safe system is robust to natural behaviour (stable). For example, making a user manually type `Delete my_repo` improves safety—it is unlikely to happen unintentionally—while doing nothing to improve security. Requiring a secret password instead would improve both safety and security. Our approach is most applicable in settings that are safe (in this sense) but not secure—of which user-preference manipulation or reward tampering are archetypal.

One can show when a system is *not* stable by demonstrating a natural behaviour that produces bad outcomes. Proving that a system is stable is an open challenge. We consider stability further in §8 alongside other limitations of our method, and highlighting that these are previously unacknowledged limitations of a number of related methods.

2 Background on Causal Influence Diagrams

Causal Influence Diagrams (CIDs) combine ideas from influence diagrams (Howard and Matheson 2005; Lauritzen and Nilsson 2001) and causality (Pearl 2009) and can be used to identify incentives (Everitt et al. 2021a). They are particularly well-suited to the analysis of delicate state because they explicitly represent the causal interactions of agents, rewards, and different parts of state while also formalizing graphical criteria for the presence of incentives to control certain parts of state. In this section we provide a background on causal models and CIDs which is needed to formally develop the delicate state setting. We return to a broader review of prior work in §7. Throughout this paper we adopt the convention that upper case denotes random variables and lower case their realizations. We also elide whether random variables are singletons or sets, noting that a set of random variables is identical to a set-valued random variable.

Restating (using original numbering) the definition of the CID itself:

Definition E3 (Everitt et al. 2021a)**.** A *causal influence diagram* (CID) is a directed acyclic graph G whose vertex set $\mathbf{V}$ is partitioned into structure nodes, $\mathbf{X}$, action nodes $\mathbf{A}$, and utility nodes, $\mathbf{U}$.

Intuitively, this is a graph where some nodes represent the agent's decision and others its goals. The rest are structure nodes. Note that what we call 'action' nodes are sometimes called 'decision' nodes. Arrows into action nodes are called 'information links'. The relationship between the nodes are defined by structural functions in a SCIM:

Definition E4 (Everitt et al. 2021a)**.** A *structural causal influence model* (SCIM) is a tuple $\mathcal{M} = \langle G, \mathcal{E}, \mathbf{F}, P\rangle$ where:

- G is a CID with finite-domain $\mathbf{V}$ and $\mathbf{U} \in \mathbb{R}^n$. We say that $\mathcal{M}$ is *compatible with* G.
- $\mathcal{E} = \{\mathcal{E}^V\}_{V\in\mathbf{V}}$ is a set of finite-domain *exogenous variables*, one for each element of $\mathbf{V}$.
- $\mathbf{F} = \{f^V\}_{V\in\mathbf{V}\setminus\mathbf{A}}$ is a set of *structural functions* f^V : $\mathrm{dom}(\mathbf{pa}^V \cup\{\mathcal{E}^V\}) \to \mathrm{dom}(V)$ that specify how each non-decision variable depends on its parents in G and exogenous variable.
- P is a Markovian probability distribution over $\mathcal{E}$ (i.e., all elements are mutually independent).

Intuitively, this describes how variables at the nodes change with each other and incorporates chance. The notation $\mathbf{PA}^X$ describes the parents of X in G. The goal of the agent is to select a policy π : $dom(\mathbf{PA}^A) \to dom(A)$ for each action node, A, so that the expected sum of the utility nodes is maximized.

We also use structural causal models (SCMs) for some of our analysis. While SCMs are logically more fundamental than SCIMs, for the purpose of this paper we can think of them as SCIMs without action nodes. That is, an SCM is a SCIM where all nodes have been assigned structural functions. In particular, imputing a policy, π, to a SCIM, $\mathcal{M}$, turns the SCIM into the SCM $\mathcal{M}_\pi$.

SCMs are fully developed by Pearl (2009), who also formalize the intervention notation $\mathrm{do}(X = x)$ to mean intervening to set the random variable X to x. Formally, $\mathrm{do}(X = x)$ replaces the structural function f^X with a constant function $X = x$. The *potential response* Y_x is used to denote Y under the intervention $\mathrm{do}(X = x)$. Somewhat abusing notation, we will write Y_π for the variable Y in $\mathcal{M}_\pi$, and $Y_{\pi,x}$ for this variable under the intervention $\mathrm{do}(X = x)$. Potential responses can be nested, allowing expressions such as Z_{Y_x}, which should be interpreted as Z_y where $y = Y_x$.

CIDs have been used to define instrumental control incentives, which formalises the intuitive notion of which variables that agent 'wants' to influence:

Definition E17 (Everitt et al. 2021a)**.** There is an *instrumental control incentive* (ICI) in a SCIM with a single-decision CID $\mathcal{M}$ on a variable X in with $\mathbf{pa}^A$ with total return $\mathcal{U} = \sum \mathbf{U}$ if, for all optimal policies π^*,

$$\mathbb{E}_{\pi^*}\left[\mathcal{U}_{X_a} \mid \mathbf{pa}^A\right] \neq \mathbb{E}_{\pi^*}\left[\mathcal{U} \mid \mathbf{pa}^A\right] \qquad (1)$$

$\mathcal{U}_{X_a}$ is the utility in the nested potential response where X is as if A had been a. Intuitively, this says that the agent has an ICI over X if it could achieve utility different than that of the optimal policy, were it also able to independently set X. Finally, to diagnose the presence or lack of ICIs over the delicate state:

Theorem E18 (Everitt et al. 2021a)**.** A single-decision CID $\mathcal{G}$ admits an ICI over $X \in \mathbf{V}$ iff $\mathcal{G}$ has a directed path from A to $\mathbf{U}$ via X: i.e. a directed path $A \to X \to \mathbf{U}$.

To review these concepts, examine Fig. 1 where we contrast a CID which admits an ICI over Z (Fig. 1a) with a CID which is not (Fig. 1b).

3 General Delicate MDP CID

Applying these tools to §1, we construct a general CID for a factored MDP with delicate and robust state.

Definition 3.1. A *delicate T-step MDP* is a factored MDP (Boutilier, Dearden, and Goldszmidt 2000) where the state is factored into delicate state, Z_t, and robust state, S_t, at each timestep.

We can describe a delicate MDP with a CID containing random variables Z_t, S_t, A_t, and R_t for $0 \le t \le T$. Here, A_t are action nodes; the R_t's are utility nodes (discounting can be introduced by scaling these); all other nodes are chance nodes. Variables depend only on the most recent timestep. The decision node A_t can observe Z_t, S_t, and R_t. The resulting CID is shown in Fig. 2a. Special cases of this graph can remove influence arrows. For example, work on reward tampering often assumes that the reward function specification (here modelled as Z_t) cannot directly influence the rest of the state (here modelled as S_{t+1}) (Everitt et al. 2021b).

4 Path-specific Objectives

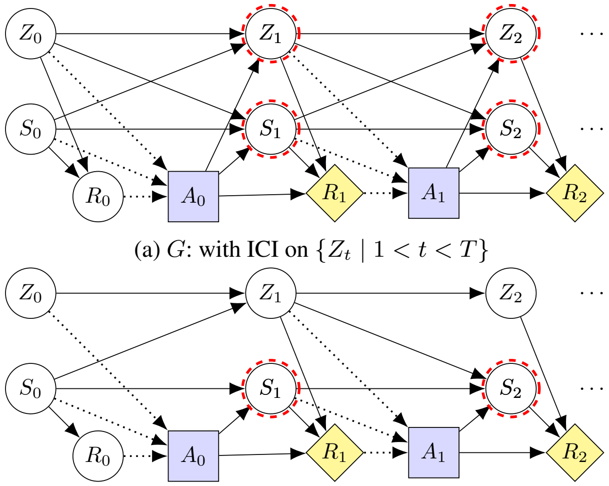
We show how to train an agent in a way that removes instrumental control incentives (ICIs) over the delicate state even though the environment actually has unsafe incentives.

To understand the causal effect that a variable X has on a variable Y along an edge-subgraph G' of an SCM M, Pearl (2001, Definition 8) defines path-specific causal effects. Informally, the path-specific effect along G' compares the outcome of Y under a default outcome $\bar{x}$ for X with the value that Y takes under a different outcome x, when the effect of the new value is propagated only along G'. Formally, restating their definition with our notation:

Definition P8 (Pearl (2001))**.** Let G be the causal graph associated with causal model M, and let G' be an edge-subgraph of G containing the paths selected for effect analysis. The G'-specific effect of x on Y (relative to reference $\bar{x}$) is defined as the total effect of x on Y in a modified model $\bar{M}_G$ formed as follows. Let each parent set, $\mathbf{PA}_i$, be partitioned into two parts

$$\mathbf{PA}_i = \{\mathbf{PA}_i(G'), \mathbf{PA}_i(\tilde{G}')\} \qquad (2)$$

where $\mathbf{PA}_i(G')$ represents those members of $\mathbf{PA}_i$ that are linked to X_i in G', and $\mathbf{PA}_i(\tilde{G}')$ represents the complementary set, from which there is no link to X_i in G'. We replace each function $f_i(\mathbf{pa}_i(G'), \epsilon)$ with a new function



(a) G: with ICI on $\{Z_t \mid 1 < t < T\}$

(b) G': with no ICI on $\{Z_t \mid 1 < t < T\}$

Figure 2: (a) A general delicate MDP CID. (b) Removing links from A_0 to Z_t removes ICI on $\{Z_t\}_{1<t<T}$. (Dashes are information links (Howard and Matheson 1984).)

$\bar{f}_i(\mathbf{pa}_i, \epsilon; G)$, where ϵ are realizations of the exogenous random variables, defined as

$$\bar{f}_i(\mathbf{pa}_i, \epsilon; G') = f_i(\mathbf{pa}_i(G'), \bar{\mathbf{pa}}_i(\tilde{G}'), \epsilon) \tag{3}$$

where $\bar{\mathbf{pa}}_i(\tilde{G}')$ stands for the values that the variables in $\mathbf{PA}_i(\tilde{G}')$ would attain (in M and ϵ) under $X = \bar{x}$ (that is $\bar{\mathbf{pa}}_i(\tilde{G}') = \mathbf{PA}_i(\tilde{G}')_{\bar{x}}$). The G'-specific effect of x on Y, denoted $\mathrm{SE}_{G'}(x, \bar{x}; Y, \epsilon)_M$ is defined as

$$\mathrm{SE}_{G'}(x, \bar{x}; Y, \epsilon)_M = \mathrm{TE}(x, \bar{x}; Y, \epsilon)_{\bar{M}_{G'}}. \tag{4}$$

As an extension of this idea, we introduce the path-specific objective (PSO) in a SCIM. Intuitively, the PSO captures the causal effect of the agent's action on its objective which is carried along a given causal path, while the other variables take their 'natural' distributions. In order to define this, while evaluating each action we impute a policy to future time-steps which the agent currently expects its future-self will take, which converts the SCIM into a SCIM with a single-decision CID (similarly to Everitt et al. 2021a). The path-specific objective may then be defined with respect to an underlying SCM: we compare the path-specific effect of a imputing a particular candidate action, a, to a baseline action, $\bar{a}$, which reduces the SCIM to a SCM. We then compute the path-specific effect in this model in simulation. More formally, we define the PSO as:

Definition 4.1. The *Path-specific Objective* (PSO) for an action node, A, in a SCIM, $\mathcal{M} = \langle G, \mathcal{E}, \mathbf{F}, P \rangle$, is defined with respect to: G', an edge-subgraph of G; π, a policy imputed to future actions; and $\bar{a}$, a default action. The PSO is

$$\mathcal{U}^{G', \bar{a}}_{\pi, a}(\epsilon) = SE_{G'}(a, \bar{a}; \mathcal{U}, \epsilon)_{\mathcal{M}_{\pi, \bar{a}}}.$$

Using this definition we can estimate this PSO and derive the Bellman update in the context of reinforcement learning.

Theorem 1. *For all delicate MDPs, $\mathcal{M} = \langle G, \mathcal{E}, \mathbf{F}, P \rangle$, and for any action A_0, there exists an edge-subgraph G' of G which does not admit an ICI over $\mathbf{Z}$. For this G', given a policy, π, and default action, $\bar{a}$, the PSO for an action, a, is (up to an additive constant of the expected utility under $\bar{a}$)*

$$\mathcal{U}^{G', \bar{a}}_{\pi, a}(\epsilon) = \mathcal{U}_{\pi, \mathbf{Z}_{\bar{a}}, a}(\epsilon), \tag{5}$$

which is the potential response of the utility under π, a, and $\mathbf{Z}_{\bar{a}}$ (the nested counterfactual $\mathbf{Z}$ under $\bar{a}$).

Moreover, specializing to reinforcement learning, the one-step Bellman optimality condition for optimizing this objective in expectation is:

$$\begin{aligned} & v_{\pi^*[\mathcal{M}, G', \bar{a}, \pi]} \\ &= \max_{a_0} \sum_{s_1, r_1} p\left(s_1 \mid z_0, s_0, \mathrm{do}(A_0 = a_0)\right) \\ &\qquad p\left(r_1 \mid z^1_{\bar{a}}, s_1, \mathrm{do}(A_0 = a_0)\right) p\left(z^1_{\bar{a}} \mid z_0, s_0\right) \\ &\qquad \cdot \left[r_0 + \gamma v_{\pi^*[\mathcal{M}, G', \bar{a}, \pi]}(r_1, z^1_{\bar{a}})\right], \end{aligned} \tag{6}$$

where γ is the discount rate.

Proof. We can construct G', an edge-subgraph of G, which removes ICIs on $\mathbf{Z}$, by removing the arrows $A_{t'} \to Z_{t'+1}$ and $S_{t'} \to Z_{t'+1}$ for $t' \geq 0$, as shown in Fig. 2b. By Theorem E18, G' admits no ICI on any $Z_{t'} \in \mathbf{Z}$ because there is no directed path from A_0 to $\mathbf{Z}$.

We further define an SCM, M, as in Definition 4.1 by imputing future actions under π. Now, by the definition of path-specific effects (Pearl 2001, Definition 8), the G'-specific effect of a on $\mathcal{U}$ relative to default action, $\bar{a}$, is equal to the total effect of a on $\mathcal{U}$ in a modified model, $M' = \langle G, \mathcal{E}, \mathbf{F}', P \rangle$. Thanks to the fact that all paths from A_0 to $\mathbf{Z}$ have been cut in G', the resulting structural functions $\mathbf{F}'$ become

$$f^V_*(\mathbf{pa}^V, \epsilon^V; G') = f^V(\mathbf{pa}^V \setminus \mathbf{Z}, \mathbf{Z}_{\bar{a}}, \epsilon^V), \tag{7}$$

where $\mathbf{Z}_{\bar{a}}$ are the values that $\mathbf{Z}$ would attain under $\bar{a}$ (in M and under the exogenous variable assigned to V, $\epsilon^V \sim \mathcal{E}^V$). This corresponds to computing the return normally, except for imputing a 'natural' distribution to the delicate states, $\mathbf{Z}$, that matches what would have happened on the default action, $\bar{a}$. This is equation (5), the return under an imputed natural distribution.

In the special case of reinforcement learning, the one-step Bellman optimality condition similarly follows by taking the expectation while intervening under the natural distribution $\mathrm{do}(Z_1 = z_{\bar{a}})$. The value of the optimal policy is then the maximal intervened reward at the next step plus the discounted value of the next state. This gives equation (6). □

It follows from Theorem 1 that any optimal policy with respect to the PSO in $\mathcal{M}$ also optimizes the total effect of a on $\mathcal{U}$ in the modified model. The analysis of instrumental control incentives offered by Everitt et al. (2021a) is relative to the SCIM for which the agent is optimal. That is, if we train an agent to be PSO-optimal in $\mathcal{M}$, the CID under which the agent is return-optimal is G'. Therefore, we must look at G' to infer agent incentives. G' admits no ICI on $\mathbf{Z}$, so an agent trained with the PSO does not have an ICI on $\mathbf{Z}$. However, note that such a policy may still systematically affect $\mathbf{Z}$ as a side-effect of acting towards some other objective. [1]

Unfortunately, these path-specific effects are not identifiable from experiments in cases with more than one timestep. Indeed, Avin, Shpitser, and Pearl (2005, Theorem 5) show that the path-specific effect of an edge-subgraph G' of a Markov causal graph G is not experimentally identifiable if and only if there is no node W such that: there is a path $X \to W$ in G, there is a path $W \to Y$ which is in G but not G', and there is a path $W \to Y$ which is in both G and G'. For example, S_1 is such a node. This contrasts with the optimization of policies with respect to path-specific effects considered by (Razieh, Kanki, and Shpitser 2018) who focus on settings with only a single time-step and non-factored MDPs.

Even though the effects cannot be experimentally identified, they are identifiable with further assumptions like counterfactual independence which can be assumed in simulated environments (Robins and Richardson 2011) and in some cases could be bounded with milder assumptions. The next subsection discusses several ways to approximate the path-specific effect in practice. While there are situations where these estimates will be inaccurate (Shpitser 2013), they all remove the ICI on the delicate state by producing models without directed paths $A_0 \to \mathbf{Z}$.

Estimating the Natural Distribution

A path-specific effect can be defined for any default action, $\bar{a}$. However, unlike prior work using path-specific effects, here we are not just estimating the effect but also optimizing an agent with respect to it. As a result, some default actions provide more useful comparisons than others. We must make two design choices: first we must select $\bar{a}$ for A_0 and a policy π for future actions; second, we must have a scheme for estimating $\mathbf{Z}_{\bar{a}}$. We call our estimate for this potential response $\bar{\mathbf{z}}$, which we use to compute the PSO in simulation. Note that $\bar{\mathbf{z}}$ must not depend on any descendent of A_0 (this would induce an ICI over $\mathbf{Z}$). Moreover, the intervention must not depend on the current policy in order for standard convergence results for the MDP optimization algorithm in question to apply.

For estimating the natural distribution, we suggest three approaches, detailed in Table 1. The most principled solution is to take the default action from a default trustworthy policy, which we call a policy baseline. Here, we define a hypothetical policy, $\bar{\pi}$, compute the way in which $\mathbf{Z}$ would evolve under that policy, and use this to impute $\bar{\mathbf{z}}$. This is effective if we can simulate the full system well enough to infer how the counterfactual system would have evolved.

Intervention	$\mathrm{do}(Z_{t+1} = \bar{z})$
Policy Baseline	$\bar{z} \sim \sum_{a'_t} \hat{p}(Z_{t+1} \mid s_t, z_t, a'_t) p_{\bar{\pi}}(a'_t \mid s_t, z_t)$
State Baseline	$\bar{z} \sim \hat{p}(z_{t+1} \mid z_t, s_t)$
Fixed State	$\bar{z} = z_t$
Ordinary w/ ICI	$\bar{z} \sim p(z_{t+1} \mid s_t, z_t, a_t)$

Table 1: Agent designers can forecast how the delicate state is likely to evolve in order to sample from the 'natural' distribution when estimating the path-specific effects. Different approximations represent different choices of 'default' behaviour. To remove the ICI, it is important that the choice of intervened value does not depend on A_0, even indirectly.

Where this is not possible, as a heuristic for setting $\bar{\mathbf{z}}$ we can set a baseline over the state itself, selecting a rule $\hat{p}$ for how $\mathbf{Z}$ evolves given the previous state. This works if we know how the delicate state tends to evolve naturally. Insofar as marginalizing over the policy baseline entails a state baseline, this is a special case of the policy baseline.

As a final heuristic, we can intervene using a fixed state such as the initial delicate state Z_0. This works if we can record Z_0 and expect relatively little change. In §6 we demonstrate these choices in simple settings.

We also note that the standard RL objective that optimizes the total effect (rather than a path-specific one) can be recovered by setting the intervention value according to the actual environment dynamics.

5 Unifying and Generalizing Prior Work

Some prior work proposes modifying training *environments* to remove undesired control incentives. In fact, these proposals can be interpreted as specifying an intervention distribution for the estimation of path-specific causal effects. Many of these are also special cases of a delicate-state setting. A schematic overview is provided in Table 2.

Decoupled Approval Uesato et al. (2020) propose giving a reward for a state-action pair different from the action taken by the agent. In our terminology, their "reward generating mechanism" constitutes a *delicate* state because the reward is *unenforceable*. Their algorithm is what we call a policy baseline in which $\bar{\pi} = \pi$, but with a different sample from the same random variable, which does not fully remove ICIs over multiple timesteps.

Counterfactual Reward and Uninfluencability Everitt et al. (2021b) consider the problem of reward tampering. This is a special case of our setting, in which the reward function state is the delicate state (and additionally they assume, in our terminology, that $\mathbf{Z}$ cannot influence $\mathbf{S}$ directly). Their proposal of *counterfactual reward functions* can be understood as, in our terminology, running a policy baseline and using this intervention to estimate a PSO. Similarly, Armstrong et al. (2020) require that an agent's actions cannot influence its reward-function learning process and

[1] Although we consider MDPs for simplicity, partially observable MDPs can be used. Everywhere we have state random variables at a time-step, instead consider two random variables, one of which is observed and the other which is not. That is, at time t we have the random variables $\{S^o_t, S^u_t, Z^o_t, Z^u_t, A_t, R_t\}$. Any influence arrow that would have gone to S_t now goes to both S^o_t and S^u_t, and similarly for Z. The proofs proceed similarly, with the sub-graph G' needing to block all paths from both the observed and unobserved delicate state instead. The resulting path-specific objective estimation is no longer computable from the agent's perspective, since part of the relevant state can no longer be observed, but can be done from the agent designer's perspective if the unobserved state is known-to-the-designer.

Prior work	Note	Reference
Decoupled Approval	Reward tampering focused. Like policy baseline, but sample from same policy.	(Uesato et al. 2020)
Counterfactual Reward & Uninfluencability	Reward tampering focused. Assumes $Z \nrightarrow S$.	(Armstrong et al. 2020; Everitt et al. 2021b)
Frozen Preference Model	Preference manipulation. Like a fixed intervention.	(Everitt et al. 2021a)
Auto-induced Distributional Shift	Like policy intervention from fixed pool of diverging 'counterfactual' worlds.	(Krueger, Maharaj, and Leike 2020)
Ignoring Effect Through Some Channel	No robust state, mostly consider one-step decisions.	(Taylor 2016)

Table 2: Overview of prior work which our framework generalizes. Uses our terminology. Details in main text.

propose a reward-function depending on what would have happened if the agent had not taken actions.

Frozen Preference Model To avoid an incentive to manipulate user preferences, Everitt et al. (2021a) propose learning and freezing a model of a person's preferences and using these to provide a reward to the agent (their Fig. 4b). This is equivalent to a fixed state intervention on the delicate state to estimate the PSO.

Auto-induced Distributional Shift Krueger, Maharaj, and Leike (2020) try to avoid the incentive for agents to induce shifts in the state-distribution by reassigning a population of agents to a new environment at each time-step. They do not explicitly distinguish delicate and robust state, instead they note that not all distribution-shift is bad. Their algorithm is a restricted version of a policy baseline with two major differences: that the intervention context is re-used every K steps (so the control incentive is only weakened, not removed), and that the intervention context is allowed to diverge after initialization rather than updating to match the starting point of each new decision (which is why their method does not work well in multi-timestep environments).

Maximizing a Quantity While Ignoring Effect Through Some Channel Taylor (2016) propose that an agent might optimize an objective while ignoring influence that flows via a part of the state. They impute a distribution to that state induced by some natural decision and use the resulting counterfactual objective to determine the actual decision. Our formalization generalizes theirs by considering the interaction between delicate and robust state over multiple timesteps and exploring different strategies for picking the natural distribution.

Insofar as these proposals identify and demarcate delicate parts of the state-space, they need to show that these parts are *stable* to show that the modifications create safe agents.

6 Experiments

We present two experimental tests of our approach in order to elaborate the underlying mathematical mechanisms. First, we use a simple tabular environment to demonstrate how an agent optimizing a PSO will not take opportunities to change the delicate state, but will act in a way that is responsive to externally-caused changes to the delicate state. Second, we show how our method removes the incentive to manipulate user preferences in a content recommendation setting used by Krueger, Maharaj, and Leike (2020). This experiment also reveals how removing control incentives does *not* guarantee safety in an unstable environment.

Hyperparameter	Setting description
Number of user types (K)	10
Number of article types (M)	10
Number of environments	20
Initialization scale	0.03
Loyalty update rate (α_1)	0.03
Preference update rate	0.003 with normalization
Architecture	1-layer 100-unit ReLU MLP
Optimization algorithm	SGD(lr=0.01, $\rho = 0.1$)
Batch size	10
Number of steps	2000 (PBT every 10)

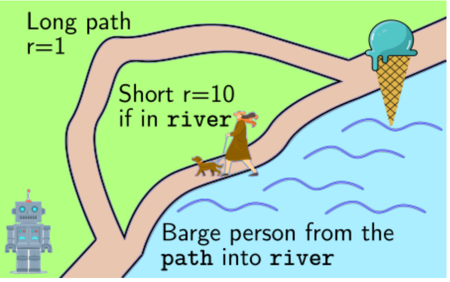
Table 3: Content Recommendation Hyperparameters.

Tabular Example: Barging

We construct an environment to demonstrate the effects of PSO around delicate state (Fig. 3a and 3b). Our agent tries to reach an ice-cream cone before it melts. Going the long way, the ice-cream melts before arrival giving a small reward. The short way is fast, and would give high reward, but it is blocked by a person. The agent can barge the person into the river, opening the short way.

The delicate state, **Z**, can take two values: the person is either on the `path` or in the `river`. The agent can take one of three actions. The long path, `L`, gives a small reward and terminates. If the person is on the `path`, the short way, `S`, achieves nothing. If the person is in the `river`, then `S` gives a large reward and terminates. The agent can barge the person out of the way, `B`, which flips the delicate state from `path` to `river`. Because the person's position is *delicate*, we want the agent to take the long path whenever the person is on the `path`. It should forgo the cold ice-cream because it should not use the person's position as a means to its end.

Naively, an optimal agent first barges the person off the path and then takes the short path. The standard reward specification approach would look at this behaviour and say "We should penalize barging people into the river." In *this simple setting* that would work: an 'oracle' return where barging gets -11 reward, $\mathcal{U}_O$, would prevent barging. The simplicity helps illustrate the path-specific objectives; but our approach



(a) Barging Setting.

	Delicate State - Z (person's position)	
Action	`path`	`river`
`L`	reward = 1; end.	reward = 1; end.
`S`	`no operation`	reward = 10; end.
`B`	Z := `river`; reward = 0	`no operation`

(b) Barging Payoffs.

Agent	Policy	$\mathbb{E}[\mathcal{U}]$	$\mathbb{E}[\mathcal{U}_{\mathcal{M}'}]$	$\mathbb{E}[\mathcal{U}_O]$
Standard	`B,S`	10	n.a.	-1
PSO – det.	`L`	1	1	1
PSO – ϵ-greedy	adaptive	1.43	1	0.9

(c) Outcomes.

Figure 3: (a-b) Three actions: long path has small reward; short path has big reward if the person is in the `river`; barging pushes the person into the `river`. (c) Normal agents barge and then take the short path, claiming high reward ($\mathbb{E}[\mathcal{U}]$). A deterministic agent optimizing PSO ($\mathbb{E}[\mathcal{U}_{\mathcal{M}'}]$) always does `L` as desired, not using the delicate state as means to an end. But if the person is accidentally pushed into the `river`, e.g. because of ϵ-greedy behaviour, the agent adapts. For an 'oracle' return ($\mathbb{E}[\mathcal{U}_O]$) with a barging penalty of -11, PSO maximizes performance and mitigates the ϵ-greedy handicap.

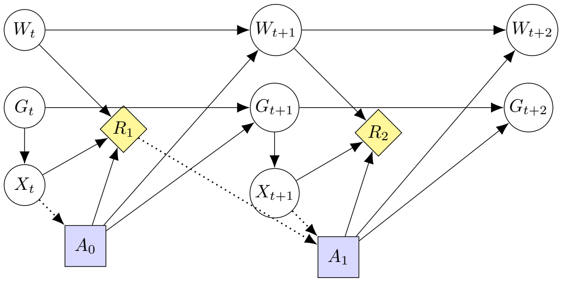


Figure 4: Content recommendation CID. W are preferences (delicate), G are loyalties and X sampled users (robust).

is meant for delicate states which are *subtle* (see §1).

An agent optimal under the PSO acts as desired. Consider a *fixed intervention*: the PSO rewards conditioned on do(position = `path`). The optimal agent now takes `L`, because the reward under this intervention of `S` is always zero. However, if the person happened to fall into the river for other reasons, the agent responds to this: it will then take `S`.

For example, if the agent is fallible and now has an $\epsilon = 0.1$ chance of taking a random action at each timestep, it might now accidentally go `B` on the first step (off-policy) and then, since the person is in the `river` anyhow, deliberately go `S` the second step. In many cases, this is what we want. This means that the agent is responsive to changes in its environment, but will not deliberately use the delicate state as a means to an end.

In Fig. 3c we describe the outcomes in this environment for standard agents and those with a fixed intervention PSO. On the deterministic on-policy version, the PSO agent performs optimally on the corrected 'oracle' return, $\mathcal{U}_O$, which by hypothesis we do not have access to. On the ϵ-greedy off-policy variant, the agent sometimes accidentally takes the penalty for barging, but at least then takes the short path in the new circumstances. A less flexible agent that never took the short path would score lower on the oracle return. Note that the desired behaviour produces a *low* expected return ($\mathbb{E}[\mathcal{U}]$). This is not a mistake: by hypothesis our reward function is not all we we care about.

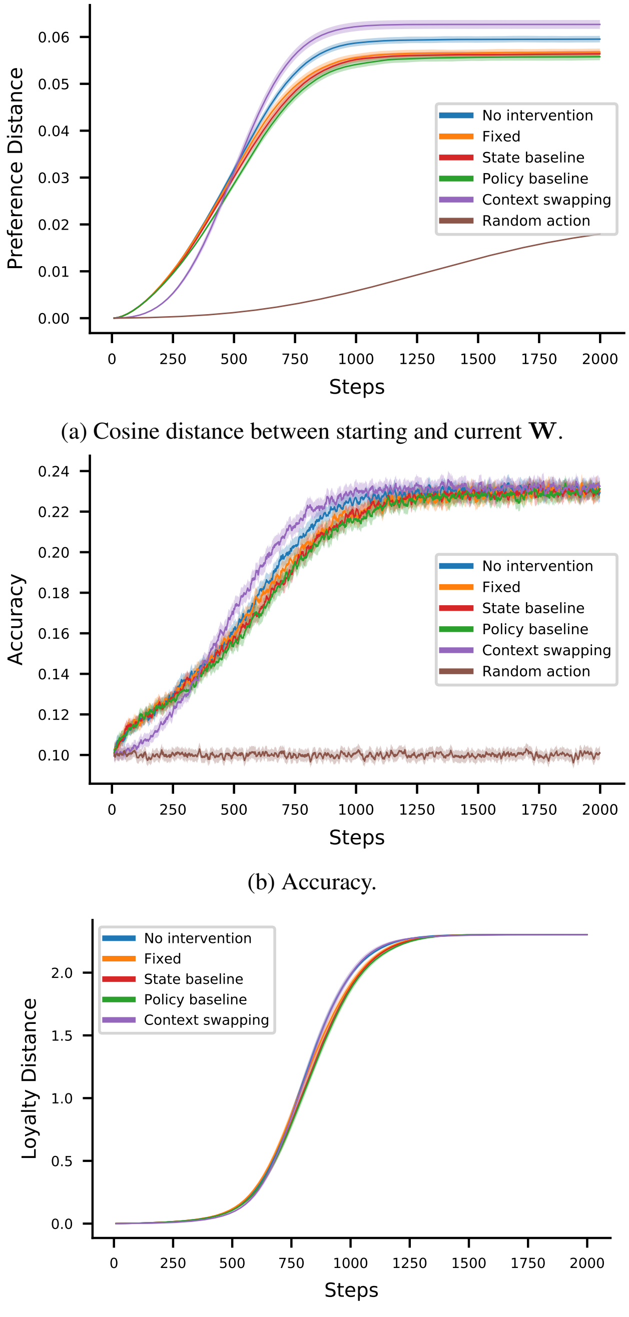
Content Recommendation

We demonstrate our method using the content recommendation simulation from Krueger, Maharaj, and Leike (2020). A population of neural network content recommendation systems are shown a sample of users and pick topics they predict the user is interested in. Users who get good recommendations become more likely to be active (i.e., sampled more often). By assumption, the users become more interested in the topics they are shown. The content recommendation system updates its recommendation by gradient descent. Periodically, the best systems are cloned and replace the worst through population-based training (Jaderberg et al. 2017).

The CID describing this is in Fig. 4. We retain the notation and set-up used by Krueger, Maharaj, and Leike (2020), with K user types and M article types. We treat the user preferences ($\mathbf{W}$, a matrix of size $M \prod K$) are the delicate state (equivalent to $\mathbf{Z}$ in this paper), while we treat the loyalties ($\mathbf{g}$, a vector of size K) and sampled users ($\mathbf{X}$, a vector of size N sampled according to $\mathbf{g}$) as robust state (elsewhere $\mathbf{S}$). This CID is therefore a special case of the general delicate MDP CID presented in §3, which adds internal structure to the robust state. This reflects the assumption that it is untoward to try to influence the user's preferences, but fine to build loyalty by giving a good service.

Following Krueger, Maharaj, and Leike (2020), at each time-step, a set of user type indices is sampled from a categorical distribution according to $\mathbf{g}_t$. The agent then selects action a_t, which is an index in the set $\{0 \ldots M\}$ representing the article type to show that user. The user clicks on the article with probability $\mathbf{W}_t^{x_t, a_t}$ and the agent gets a reward of 1 if a click arrives and 0 otherwise. As a result of the action, the loyalties of users who click on the article increase by α_1 and all user types become more interested in the article types they were shown. Unlike their work, we do 10 parallel recommendations per time-step for computational speed.

By default, the agent gradually encourages predictable



(a) Cosine distance between starting and current $\mathbf{W}$.

(b) Accuracy.

(c) KL-divergence between starting and current $\mathbf{g}$.

Figure 5: (a) PSO slows the drift in user-preferences. Although context-swapping looks effective initially, in fact it regularizes, leading to more drift eventually. Even without learning (totally random) some drift is caused by the system dynamics. (b) Methods have comparable accuracy, though context-swapping regularizes. (c) None of the interventions appreciably change the rate of drift of the 'loyalty' variable, as desired. 100 seeds, shading is standard error.

users to use the platform and develop even more predictable tastes. An agent trained with PSO has no such incentive. However, the *unstable* dynamics—predicting well naturally encourages preference drift—mean this happens naturally.

Three variants of the PSO reduce drift similarly although the policy-baseline (which is the most 'correct') performs marginally better than the two methods which are more heuristic. In Fig. 5a we show how removing the control incentive over the path-specific objective reduces the change in user preferences at the end of training relative to no intervention (blue line is higher than orange, red, and green). This is achieved without meaningful harm to accuracy (Fig. 5b) and without affecting change in loyalty, which we did not treat as delicate (Fig. 5c).

For a fixed intervention, we compute the population-based-training score by intervening $\mathrm{do}(\mathbf{W}_t = \mathbf{W}_0)$. For the policy baseline, we use a uniformly random baseline policy—$\bar{\pi} = \mathcal{U}(0, M)$—to calculate a simulated counterfactual preference, $\overline{\mathbf{W}}_t$. The population-based-training score then uses the intervention $\mathrm{do}(\mathbf{W}_t = \overline{\mathbf{W}}_t)$. For a state baseline, we assume that the most preferred article type for each user is slightly more preferred each step, intervening with a directly calculated $\overline{\mathbf{W}}$. In this setting, computing the PSO only marginally reduces training speed, with all variants adding less than 15% to the naive training time.

Fig. 5 also shows the strengths and weaknesses of context-swapping (Krueger, Maharaj, and Leike 2020). Changing environments every step slows down learning at first, resulting in smaller preference drift. But it also regularizes, improving accuracy, causing more preference drift eventually.

In addition to showing how PSO removes control incentives, this experiment shows how removing the control incentive is not enough to ensure safety. This environment is not *stable*. Even unmotivated behaviour causes drift because the user always becomes more interested in what they are shown. Even a completely random policy (brown line) causes some drift in preferences (Fig. 5 middle). Regardless of control incentive, preferences drift faster when shown the same topic more often, which happens if the policy is accurate. Note also that we cannot offer an 'oracle' return here because, as designers, we do not really understand what desirable behaviour for user preferences would be.

7 Related Work

Our work builds on Causal Influence Diagrams (Howard and Matheson 2005; Lauritzen and Nilsson 2001; Everitt et al. 2021a), using tools from the path-specific causal effects literature (Pearl 2001; Avin, Shpitser, and Pearl 2005). Path-specific effects have been used, especially in medical literature, to measure impacts on only some causal pathways. Indeed, Razieh, Kanki, and Shpitser (2018) perform policy optimization in a medical context using path-specific effects as a target, although they consider much simpler causal graphs with stronger assumptions.

We aim to address the problem of safe agent design (Amodei et al. 2016) using a strategy which is orthogonal to other approaches which either aim at better-specified rewards (Leike et al. 2018), preferences (Christiano et al.

2017), or demonstrations (Schaal 1997). In trying to avoid actions that use part of the state as a means to an end, we also adopt a different approach to methods that merely try to avoid changing parts of the environment for whatever reason (Turner, Ratzlaff, and Tadepalli 2020; Krakovna et al. 2020). A number of papers have considered problems in safe-agent design which can be regarded as special cases of delicate state and use approaches which can be interpreted as special cases of our path-specific objective. These include reward tampering (Uesato et al. 2020; Everitt et al. 2021b), online reward learning (Armstrong et al. 2020), and auto-induced distributional shift (Krueger, Maharaj, and Leike 2020). Taylor (2016) more generally argue that safe agents might need to be able to optimize some objective while ignoring effects along certain channels and propose a counterfactual causal rule for this.

8 Discussion and Limitations

Much existing work on agent safety tries to improve descriptions of good and bad behaviour, e.g., through rewards or demonstration. This is hard, making it important to consider alternatives. Out complementary approach splits the *environment* into parts that are easy to reward and parts that are better to simply remove any incentive to control.

This offers a resolution to the subtlety, but it only provides safety if the non-incentivized behaviour is safe, a property we call *stability*. While instability can be proved by example, stability seems hard to prove. By definition, anything which is *manipulable* is not stable under all possible natural behaviours—proving stability is therefore contingent, empirical, and a matter of degree. Stability might be a reasonable assumption in isolable systems (e.g., reward function implementations) or systems that already withstand competitive pressures (e.g., political preferences). Although stability is a serious requirement, by unifying several previous proposals and providing a clearer language for them we hope to give researchers the tools to make progress addressing it. Other challenges for implementation include:

Graph Discovery To estimate the PSO we must define a causal graph and define the vertices. This is difficult in real settings where it is unclear how to carve up reality.

Causal estimation Although not experimentally identifiable, PSOs are identifiable in simulation. Approximation under other assumptions may be possible.

Distribution Observation We need to (partly) observe the state. For psychological state, like beliefs, this can be hard.

Moral choices Deciding what is delicate and what is not is a complicated ethical decision.

Acknowledgements

We would like to gratefully thank for their comments, help, and discussions Charles Evans, James Fox, Julia Haas, Lewis Hammond, Zach Kenton, David Krueger, Eric Langlois, Vlad Mikulik, Jon Richens, and Rohin Shah.

This work was supported in-part by the Leverhulme Centre for the Future of Intelligence, Leverhulme Trust, under Grant RC2015-067.

References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. *arXiv*.

Armstrong, S.; Leike, J.; Orseau, L.; and Legg, S. 2020. Pitfalls of Learning a Reward Function Online. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 1592–1600.

Avin, C.; Shpitser, I.; and Pearl, J. 2005. Identifiability of Path-Specific Effects. *IJCAI*, 357–363.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic Dynamic Programming with Factored Representations. *Artif. Intell.*, 121(1–2): 49–107.

Burr, C.; Cristianini, N.; and Ladyman, J. 2018. An analysis of the interaction between intelligent software agents and human users. *Minds and Machines*, 28: 735–774.

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. *Neural Information Processing Systems*.

Everitt, T.; Carey, R.; Langlois, E.; Ortega, P. A.; and Legg, S. 2021a. Agent Incentives: A Causal Perspective. *AAAI*.

Everitt, T.; Hutter, M.; Kumar, R.; and Krakovna, V. 2021b. Reward Tampering Problems and Solutions in Reinforcement Learning: A Causal Influence Diagram Perspective. *Synthese*.

Howard, R. A.; and Matheson, J. E. 1984. The principles and applications of decision analysis. *Strategic Decisions Group, Palo Alto, CA*, 719–762.

Howard, R. A.; and Matheson, J. E. 2005. Influence Diagrams. *Decision Analysis*, 2: 127–143.

Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W. M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; Fernando, C.; and Kavukcuoglu, K. 2017. Population Based Training of Neural Networks. *arXiv*.

Kenton, Z.; Everitt, T.; Weidinger, L.; Gabriel, I.; Mikulik, V.; and Irving, G. 2021. Alignment of Language Agents. *arXiv*.

Krakovna, V.; Orseau, L.; Ngo, R.; Martic, M.; and Legg, S. 2020. Avoiding Side Effects by Considering Future Tasks. *Neural Information Processing Systems*.

Kramer, A. D. I.; Guillory, J. E.; and Hancock, J. T. 2014. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24): 8788–8790.

Krueger, D.; Maharaj, T.; and Leike, J. 2020. Hidden Incentives for Auto-Induced Distributional Shift. *arXiv:2009.09153 [cs, stat]*.

Lauritzen, S. L.; and Nilsson, D. 2001. Representing and solving decision problems with limited information. *Management Science*, 47: 1235–1251.

Leike, J.; Krueger, D.; Everitt, T.; Martic, M.; Maini, V.; and Legg, S. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv*.

Pearl, J. 2001. Direct and Indirect Effects. *Uncertainty in Artificial Intelligence*, 7.

Pearl, J. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition.

Razieh, N.; Kanki, P.; and Shpitser, I. 2018. Estimation of Personalized Effects Asssociated with Causal Pathways. *Uncertainty in Artificial Intelligence*.

Robins, J. M.; and Richardson, T. S. 2011. Alternative Graphical Causal Models and the Identification of Direct Effects.

Russell, S. 2019. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin Publishing Group. ISBN 9780525558620.

Schaal, S. 1997. Learning from Demonstration. *Neural Information Processing Systems*, 9.

Shpitser, I. 2013. Counterfactual graphical models for longitudinal mediation analysis with unobserved confounding. *Cognitive Science*, 37(6): 1011–1035.

Taylor, J. 2016. Maximizing a quantity while ignoring effect through some channel. *Alignment Forum*.

Turner, A.; Ratzlaff, N.; and Tadepalli, P. 2020. Avoiding Side Effects in Complex Systems. *Neural Information Processing Systems*.

Uesato, J.; Kumar, R.; Krakovna, V.; Everitt, T.; Ngo, R.; and Legg, S. 2020. Avoiding Tampering Incentives in Deep RL via Decoupled Approval. *arXiv:2011.08827 [cs]*.