Artificial Intelligence 322 (2023) 103963

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Discovering agents

Zachary Kenton [a,*], Ramana Kumar [a], Sebastian Farquhar [a], Jonathan Richens [a], Matt MacDermott [b], Tom Everitt [a]

[a] DeepMind, United Kingdom of Great Britain and Northern Ireland
[b] Imperial College London, United Kingdom of Great Britain and Northern Ireland

ARTICLE INFO

Article history:
Received 14 October 2022
Received in revised form 2 June 2023
Accepted 9 June 2023
Available online 24 June 2023

Keywords:
Artificial intelligence
Game theory
Causality

ABSTRACT

Causal models of agents have been used to analyse the safety aspects of machine learning systems. But identifying agents is non-trivial – often the causal model is just assumed by the modeller without much justification – and modelling failures can lead to mistakes in the safety analysis. This paper proposes the first formal causal definition of agents – roughly that agents are systems that would adapt their policy if their actions influenced the world in a different way. From this we derive the first causal discovery algorithm for discovering the presence of agents from empirical data, given a set of variables and under certain assumptions. We also provide algorithms for translating between causal models and game-theoretic influence diagrams. We demonstrate our approach by resolving some previous confusions caused by incorrect causal modelling of agents.

1. Introduction

How can we recognise agents? In economics textbooks, certain entities are clearly delineated as choosing actions to maximise utility. In the real world, however, distinctions often blur. Humans may be almost perfectly agentic in some contexts, while manipulable like tools in others. Similarly, in advanced reinforcement learning (RL) architectures, systems can be composed of multiple non-agentic components, such as actors and learners, and trained in multiple distinct phases with different goals, from which an overall goal-directed agentic intelligence emerges.

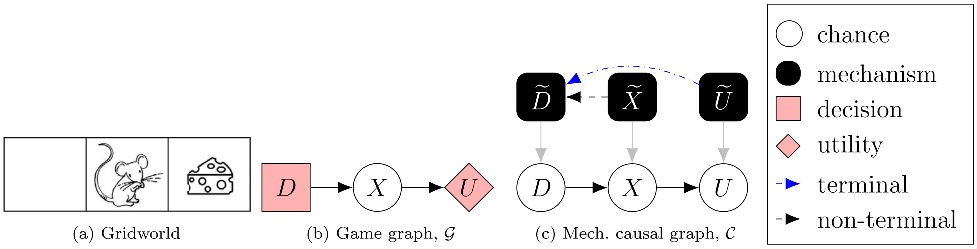
It is important that we have tools to discover goal-directed agents. Artificially intelligent agents that competently pursue their goals might be dangerous depending on the nature of this pursuit, because goal-directed behaviour can become pathological outside of the regimes the designers anticipated [7,60]. They may pursue convergent instrumental goals, such as resource acquisition and self-preservation [44]. Such safety concerns motivate us to develop a formal theory of goal-directed agents, to facilitate our understanding of their properties, and avoid designs that pose a safety risk.

The central feature of agency for our purposes is that agents are systems whose outputs are *moved by reasons* [13]. In other words, the reason that an agent chooses a particular action is that it “expects it” to precipitate a certain outcome which the agent finds desirable. For example, a firm may set the price of its product to maximise profit. This feature distinguishes agents from other systems, whose output might accidentally be optimal for producing a certain outcome. For example, a rock that is the perfect size to block a pipe is accidentally optimal for reducing water flow through the pipe.

Systems whose actions are moved by reasons, are systems that would act differently if they “knew” that the world worked differently. For example, the firm would be likely to adapt to set the price differently, if consumers were differently

[*] Corresponding author.
E-mail address: zkenton@deepmind.com (Z. Kenton).

https://doi.org/10.1016/j.artint.2023.103963
0004-3702/



(a) Gridworld (b) Game graph, $\mathcal{G}$ (c) Mech. causal graph, $\mathcal{C}$

Fig. 1. Different graphical representations for the mouse example (Section 1.1).

price sensitive (and the firm was made aware of this change to the world). In contrast, the rock would not adapt if the pipe was wider, and for this reason we don't consider the rock to be an agent.

Behavioural sensitivity to environment changes can be modelled formally with the language of causality and structural causal models (SCMs) [46]. To this end, our first contribution is to introduce *mechanised* SCMs (Sections 3.1 and 3.2), a variant of mechanised causal games [32]. The graph of a mechanised SCM can be inferred from a set of interventional distributions (Section 3.3). Building on this, our second contribution is an algorithm for determining which variables represent agent decisions and which represent the objectives those decisions optimise, i.e., the *reasons that move the agent* (Section 3.4). This lets us convert a mechanised SCM into a (structural) causal game [32].[1] Combined, this means that under suitable assumptions, we can infer a game graph from a set of experiments, and in this sense *discover agents*.[2] Our third contribution is more philosophical, giving a novel formal definition of agents based on our method (Section 1.3).

These contributions are important for several reasons. First, they ground game graph representations of agents in causal experiments. These experiments can be applied to real systems, or used in thought-experiments to determine the correct game graph and resolve confusions (see Section 4). With the correct game graph obtained, the researcher can then use it to understand the agent's incentives and safety properties [18,30], with an extra layer of assurance that a modelling mistake has not been made. Our algorithms also open a path to automatic inference of game graphs, especially in situations where experimentation is cheap, such as in software simulations.

1.1. Example

To illustrate our method in slightly more detail, consider the following simple example, consisting of a gridworld with three squares, and with a mouse starting in the middle square (Fig. 1a). The mouse can go either left or right, represented by binary variable D. There is some ice which may cause the mouse to slip: the mouse's position, X, follows its choice, D, with probability $p = 0.75$, and slips in the opposite direction with probability $1 - p$. Cheese is in the right square with probability $q = 0.9$, and the left square with probability $1 - q$. The mouse gets a utility, U, of 1 for getting the cheese, and zero otherwise. The directed edges $D \to X$ and $X \to U$ represent direct causal influence.

The decision problem can be represented with the game graph in Fig. 1b: the agent makes a decision, D, which affects its position, X, which affects its utility, U. The intuition that the mouse would choose a different behaviour for other settings of the parameters p and q, can be captured by a *mechanised causal graph* (Fig. 1c). This graph contains additional *mechanism nodes*, $\widetilde{D}, \widetilde{X}, \widetilde{U}$ in black, representing the mouse's decision rule and the parameters p and q. As usual, edges between mechanisms represent direct causal influence, and show that if we intervene to change the cheese location, say from $q = 0.9$ to $q = 0.1$, and the mouse is aware[3] of this, then the mouse's decision rule changes (since it's now more likely to find the cheese in the leftmost spot). Experiments that change p and q in a way that the mouse is aware of, generate interventional data that can be used to infer both the mechanised causal graph (Fig. 1c) and from there the game graph (Fig. 1b). The edge labels (colours) in Fig. 1c will be explained in Section 3.2.

1.2. Informal characterisation of agents

Informally, the definition of agency that we are proposing is the following:

Agents are systems that would adapt their policy if they were aware that their decisions influenced the world in a different way.

Let us illustrate this with some examples. The mouse in the example above is an agent, because it would adapt its decision (going left/right) if it learned that the cheese-location parameter had changed (it might learn this through repeated

[1] We can also reverse this, converting a causal game into a mechanised SCM (Section 3.5).

[2] Note this is relative to a *frame* – a choice of variables that appear in our causal model (Section 5.2).

[3] The mouse could become *aware* of this through learning from repeated trials under soft interventions of X and U which occur on every iteration, see Section 3.1 for further discussion.

interaction with the environment). Humans are also agents, because we usually adapt our behaviour if suitably informed about changes to the consequences of our actions. Informing humans is relatively easy, as we understand natural language.

Other systems are only agents if their *creation process* is considered as part of the system. For example, consider changing the mechanism for how a heater operates, so that it cools rather than heats a room. An existing thermostat will not adapt to this change, and is therefore not an agent. However, if the designers of the thermostat were aware of the change to the heater, then they would have designed the thermostat differently. So the thermostat *with its creation process* is an agent. Similarly, most model-free RL agents would only pursue a different policy if retrained in the modified environment. Thus we consider the system of the *RL training process* to be an agent, but not the *learnt RL policy* by itself. Evolution plays a similar role for the agency of simpler life forms.

Finally, accidentally put together systems are *not* agents: a rock is not an agent because it does not adapt its policy. Borderline cases include evolutionary pre-stages to life, where "systems" are assembled by accident, but persist for increasing periods of time as a result of the more persistent systems remaining.

Our agency definition thus includes a wide range of different systems. According to our definition, agents may or may not have explicit world models and goal representations, as it includes both humans (that do have explicit world models and goal representations) as well as model-free RL agents and their training process (which lack such representations). Our definition also encompasses agents created by design, such as thermostats, and agents created by learning (e.g. humans and RL agents)

1.3. Other characterisations of agents

To put our definition in context, let us compare it to previous characterisations of agents:

- *The intentional stance*: an agent's behaviour can be usefully understood as trying to optimise an objective [13].
- *Cybernetics*: an agent's behaviour adapts to achieve an objective (e.g. Wiener [58], Ashby [1]).
- *Decision theory / game theory / economics / AI*: An agent selects a policy to optimise an objective.
- An agent is a system whose behaviour can be *compressed* with respect to an objective function [45].
- "An *optimising system* is... a part of the universe [that] moves predictably towards a small set of target configurations" [21].
- A *goal-directed system* has self-awareness, planning, consequentialism, scale, coherence, and flexibility [43].
- Agents are ways for the future to influence the past (via the agent's model of the future) [25,22].

Agency has also been discussed further in psychology [9], agent alignment [54], and computer science [59].

Our proposal may be read as an alternative to, or an elaboration of, the intentional stance and cybernetics definitions (depending on how you interpret them) couched in the language of causality. Our definition is fully consistent with the decision theoretic view, as agents choose their behaviour differently depending on its expected consequences, but doesn't require us to know who is a decision maker in advance, nor what they are optimising.

The formal definition by Orseau et al. can be viewed as an alternative interpretation of the intentional stance: the behaviour of systems that choose their actions to optimise an objective function should be highly compressible with respect to that objective function. However, Orseau et al.'s definition suffers from two problems: First, in simple settings, where there is only a small and finite number of possible behaviours (e.g. the agent decides a single binary variable), it will not be possible to compress any policy beyond its already very short description. Second, the compression-based approach only considers what the system actually does. It may therefore incorrectly classify as agents systems with accidentally optimal input-output mappings, such as the water-blocking rock above. Our proposal avoids these issues, as even a simple policy may adapt, but the rock will not.

The insightful proposal by Flint leaves open the question of what part of an optimising system is the agent, and what part is its environment. He proposes the additional property of *redirectability*, but its not immediately clear how it could be used to identify decision nodes in a causal graph (intervening on almost any node will change the outcome-distribution).

The goal-directed systems that Ngo has in mind are agentic in a much stronger sense than we are necessarily asking for here, and each of the properties contain room for interpretation. However, our definition is important for goal-directedness, as it distinguishes incidental influence that a decision might have on some variable, from more directed influence: only a system that adapts can be said to be trying to influence the variable in a systematic way. Adaptation can therefore be used as a *test for goal-directed influence*.

Our definition also matches closely the backwards causality definition of agency by Garrabrant [25], as can be seen by the time-opposing direction of the edges $\widetilde{X} \to \widetilde{D}$ and $\widetilde{U} \to \widetilde{D}$ in Fig. 1c. It also fits nicely with formalisations of agent incentives [30,18], which effectively rely on behaviour in causal scenarios of the form that we consider here. This is useful, as a key motivation for our work is to analyse the intent and incentives of artificial agents.

1.4. Outline

Our paper proceeds as follows: we give relevant technical background in Section 2; give our main contribution, algorithms for discovering agents, in Section 3; show some example applications of this in Section 4 followed by a discussion in Section 5.

2. Background

Before we get to our algorithms for discovering agents, we cover some necessary technical background. The mathematical details can be found in Appendix A. Throughout, random variables are represented with roman capital letters (e.g. V), and their outcomes with lower case letters (e.g. v). We use bold type to indicate vectors of variables, $\boldsymbol{V}$, and vectors of outcomes $\boldsymbol{v}$. For simplicity, each variable V only has a finite number of possible outcomes, denoted $\mathrm{dom}(V)$. For a set of variables, $\mathrm{dom}(\boldsymbol{V}) = \prod_{V \in \boldsymbol{V}} \mathrm{dom}(V)$.

In structural causal models (SCMs; [46]), randomness comes from exogenous (unobserved) variables, $\boldsymbol{\mathcal{E}}$, whilst deterministic structural equations relate endogenous variables, $\boldsymbol{V}$, to each other and to the exogenous ones, i.e., $V = f^V(\boldsymbol{V}, \mathcal{E}^V)$ (Definition A.5). An SCM M induces a causal graph G over the endogenous variables, in which there is an edge $W \to V$ if $f^V(\boldsymbol{V}, \mathcal{E}^V)$ depends on the value of W (Definition A.6). The SCM is *cyclic* if its induced graph is, and *acyclic* otherwise. We never permit self-loops $V \to V$. Parents, children, ancestors and descendants in the graph are denoted $\mathbf{Pa}^V$, $\mathbf{Ch}^V$, $\mathbf{Anc}^V$, and $\mathbf{Desc}^V$, respectively (neither include the variable V). The family is denoted by $\mathbf{Fa}^V = \mathbf{Pa}^V \cup \{V\}$. Interventions on $\boldsymbol{Y} \subseteq \boldsymbol{V}$, denoted $\mathrm{do}(\boldsymbol{Y} = \boldsymbol{y})$, can be realised as replacements of a subset of structural equations, so that $\boldsymbol{Y} = \boldsymbol{f}^{\boldsymbol{Y}}(\boldsymbol{V}, \mathcal{E}^{\boldsymbol{Y}})$ gets replaced with $\boldsymbol{Y} = \boldsymbol{y}$ (Definition A.7). The joint distribution $P(\boldsymbol{V} \mid \mathrm{do}(\boldsymbol{Y} = \boldsymbol{y}))$ is called the *interventional distribution* associated with intervention $\mathrm{do}(\boldsymbol{Y} = \boldsymbol{y})$. A *soft* intervention instead replaces $\boldsymbol{f}^{\boldsymbol{Y}}$ with some other (potentially non-constant) functions $\boldsymbol{g}^{\boldsymbol{Y}}$.

A (structural) causal game [18,32,39] is similar to an SCM, but where the endogenous variables are partitioned into chance, decision, and utility variables, denoted $\boldsymbol{X}$, $\boldsymbol{D}$ and $\boldsymbol{U}$ respectively (Definition A.9). Let $N = \{1, \ldots, n\}$ be a set indexing n agents. The decision variables belonging to agent $A \in N$ are denoted $\boldsymbol{D}^A \subseteq \boldsymbol{D}$, and the agent's utility is taken to be the sum of the agent's utility variables, $\boldsymbol{U}^A \subseteq \boldsymbol{U}$. A causal game is associated with a *game graph* with square, round and diamond nodes for decision, chance and utility variables, respectively, with colours associating decision and utility nodes with different agents (Fig. 1b and Definition A.10). Edges into chance and utility nodes mirror those of an SCM, while edges into decision nodes represent what information is available, i.e. $W \to D$ is present if the outcome of W is available when making the decision D, with information edges displayed with dotted lines.

Given a causal game, each agent can set a decision rule, π^D, for each of their decisions, D, which maps the information available at that decision to an outcome of the decision. A collection of decision rules for all of a player's decisions is called a *policy* – the set of stochastic decision distributions (not the mapping from input histories to stochastic outputs as is sometimes used). Policies for all agents are called *policy profiles*. Formally, the decision rule π^D is a deterministic function of $\mathbf{Pa}^D$ and $\mathcal{E}^D$, where $\mathcal{E}^D$ provides randomness to enable stochastic decisions. This means that the decision rules can be combined with the causal game to form an SCM, which can be used to compute each agent's expected utility. In the single agent case, the decision problem represented by the causal game is to select an optimal decision rule to maximise the expected utility (Definition A.11). With multiple agents, solution concepts such as Nash Equilibrium (Definition A.12) or Subgame-Perfect Nash Equilibrium (Definition A.13) are needed, because in order to optimise their decision rules agents must also consider how other agents will optimise theirs.

Similar to SCMs, interventions in a causal game can be realised as replacements of a subset of structural equations. However, in contrast to an SCM, an intervention can be made *before* or *after* decision rules are selected. This motivates a distinction between *pre-policy* and *post-policy* interventions [32]. Pre-policy intervention are made before the policies are selected, and agents may adapt their policies (according to some rationality principle) to account for the intervention.

3. Algorithms for discovering decision and utility nodes

Having discussed some background material, we now begin our main contribution: providing algorithms that among a set $\boldsymbol{V}$ of variables identify

- decision nodes, i.e. nodes that are controlled by an adaptive agent, and
- utility nodes, i.e. nodes representing a quantity that the agent has an intrinsic interest to influence.

The variables $\boldsymbol{V}$ represent different measurable aspects of the world, such as the location of the cheese or the slipperiness of the ground (Fig. 1). Together they form a "frame" through which to view the world. What's a decision or a utility is relative to that frame (Section 5.2). The set $\boldsymbol{V}$ need *not* contain the agent's "actual" decision or utility, but only nodes related to these quantities. For example, if the actual decision is not present in $\boldsymbol{V}$, but some child of the decision is, then the child will be identified as a decision relative to the frame represented by $\boldsymbol{V}$ (see Appendix C).

The rest of this section will introduce edge-labelled mechanised SCMs, and then propose three algorithms:

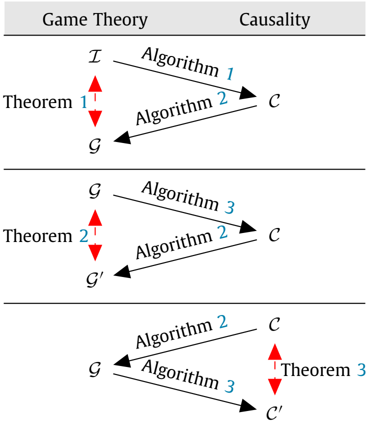


Fig. 2. Overview of our three theorems (symbolised by red arrows). Each provides relations between a game-theoretic mechanised causal game, $\widetilde{\mathcal{M}}$, with its interventional distributions, $\mathcal{I}$, and with its associated game graph, $\mathcal{G}$, and a causal object – a mechanised causal graph, $\mathcal{C}$. Our proposed algorithms Algorithm 1, *Mechanised Causal Graph Discovery*; Algorithm 2, *Agency Identification*; and Algorithm 3, *Mechanism Identification*; detail how to transform from one representation to another.

- Algorithm 1, *Mechanised Causal Graph Discovery*, produces an edge-labelled mechanised causal graph based on interventional distributions.
- Algorithm 2, *Agency Identification*, takes an edge-labelled mechanised causal graph and produces the corresponding game graph.
- Algorithm 3, *Mechanism Identification*, takes a game graph and draws the corresponding edge-labelled mechanised causal graph.

Theorems 1 to 3 establish their correctness, and Fig. 2 visualises their relationships.

3.1. Mechanised structural causal model

The algorithms all revolve around *mechanised SCMs*, which we introduce in this section. A mechanised SCM is similar to an ordinary SCM, but includes a distinction between two types of variables: object-level and mechanism variables. The intended interpretation is that the mechanism variables parameterise how the object-level variables depend on their object-level parents. Mechanism variables have been called *regime indicators* [11] and *parameter variables* [12]. Mechanised SCMs are variants of *mechanised causal games* [32] that lack explicitly labelled decision and utility nodes. Fig. 1c draws the induced graph of a mechanised SCM.

Definition 1 *(Mechanised SCM).* A *mechanised SCM* is an SCM in which there is a partition of the endogenous variables $\boldsymbol{\mathcal{V}} = \boldsymbol{V} \cup \widetilde{\boldsymbol{V}}$ into object-level variables, $\boldsymbol{V}$ (white nodes), and mechanism variables, $\widetilde{\boldsymbol{V}}$ (black nodes), with $|\boldsymbol{V}| = |\widetilde{\boldsymbol{V}}|$. Each object-level variable V has exactly one mechanism parent, denoted $\widetilde{V}$, that specifies the relationship between V and the object-level parents of V.

We refer to edges between object-level nodes as *object-level edges* E^{obj}, edges between mechanism nodes as *mechanism edges* E^{mech}, and edges between a mechanism node and the object-level node it controls *functional edges* E^{func}. We only consider mechanised SCMs in which the object-level-only subgraph is acyclic, but we allow cycles in the mechanism-only subgraph (we follow the formalism of Bongers et al. [6] when using cyclic models). By connecting mechanism variables with causal links, we violate the commonly taken *independent causal mechanism assumption* [51], though we introduce a weaker form of it in Assumption 4 (see further discussion in Section 5.6).

Interventions in a mechanised SCM are defined in the same way as in a standard SCM, via replacement of structural equations. An intervention on an object-level variable V changes the value of V without changing[4] its mechanism, $\widetilde{V}$. This can be interpreted as the intervention occurring after all mechanisms variables have been determined/sampled.

In a causal model, it is necessary to assume that the procedure for measuring and setting (intervening on) a variable is specified. Mechanised SCMs thereby assume a well-specified procedure for measuring and setting both object-level and mechanism variables. Pre- and post-policy interventions in games correspond to mechanism and object-level interventions in mechanised SCMs [32].

[4] Alternatively, it can be viewed as a path-specific intervention on $\widetilde{V}$ whose effects are constrained to V, and does not affect other mechanism variables (assuming that the domain of $\widetilde{V}$ is rich enough to facilitate the value V is intervened to).

The distinction between mechanism and object-level variables can be made more concrete by considering repeated interactions. In Section 1.1, assume that the mouse is repeatedly placed in the gridworld, and can adapt its decision rule based (only) on previous episodes. A mechanism intervention would correspond to a (soft) intervention that takes place across all time steps, so that the mouse is able to adapt to it. Similarly, the outcome of a mechanism can then be measured by observing a large number of outcomes of the game, after any learning dynamics has converged.[5] Finally, object-level interventions correspond to intervening on variables in one particular (post-convergence) episode. Assuming the mouse is only able to adapt its behaviour based on previous episodes, it will have no way to adapt to such interventions. Appendix B has a more detailed example of marginalising and merging nodes in a repeated game to derive the mechanised causal graph and game graph.

3.2. Edge-labelled mechanised causal graphs

Next, we introduce an edge-labelling on mechanised SCMs, to enable the identification of agents. These edge-labellings will require us to sever a node's children with an intervention that agents adapt to, i.e. with a mechanism intervention. We call a mechanism intervention that severs the children of an object-level node a *structural mechanism intervention*:

Definition 2 *(Structural mechanism intervention).* A *structural mechanism intervention* on a variable V is an intervention $\widetilde{v}$ on its mechanism variable $\widetilde{V}$ such that V is conditionally independent of its object-level parents. That is, under $\mathrm{do}(\widetilde{V}=\widetilde{v})$, the following holds

$$\Pr(V \mid \mathbf{Pa}^V, \mathrm{do}(\widetilde{V}=\widetilde{v})) = \Pr(V \mid \mathrm{do}(\widetilde{V}=\widetilde{v})). \tag{1}$$

Using structural mechanism interventions, we can generate more refined tests for decision and utility nodes. If a variable W is inherently valuable to an agent (i.e. is a utility node), rather than just instrumentally valuable for something downstream, then its policy should adapt to a change in the mechanism for W even if the children of W are cut off. In contrast, if W were just instrumental, then the agent should stop caring about W once it's severed from its children. Similarly, if a variable V is chosen to achieve some particular consequence (i.e. is a decision node), rather than its distribution being set mechanistically by some natural process, then V's mechanism should not respond when the children of V are cut off, as V would then be a completely inconsequential decision.

The following definition formalises these ideas, and calls a mechanism edge *terminal* if it connects nodes that satisfy the expected properties of decision and utility nodes.

Definition 3. A mechanised SCM is *edge-labelled* if it further identifies a subset $E^{\text{term}} \subseteq E^{\text{mech}}$ of mechanism edges $\widetilde{W} \to \widetilde{V}$, called *terminal* mechanism edges, such that:

1. For any structural mechanism interventions on $\mathbf{Ch}^W$, there exists a mechanism intervention $\widetilde{w}$ on $\widetilde{W}$ such that $P(\widetilde{V} \mid \mathrm{do}(\widetilde{w}, \widetilde{\mathbf{ch}^W})) \neq P(\widetilde{V} \mid \mathrm{do}(\widetilde{\mathbf{ch}^W}))$.
2. For any structural mechanism interventions on $\mathbf{Ch}^V$ and any mechanism intervention $\widetilde{w}$ on $\widetilde{W}$ we have $P(\widetilde{V} \mid \mathrm{do}(\widetilde{w}, \widetilde{\mathbf{ch}^V})) = P(\widetilde{V} \mid \mathrm{do}(\widetilde{\mathbf{ch}^V}))$.

Terminal edges are drawn dashdotted blue, and non-terminal ones with dashed black.

For example, the mechanism edge $\widetilde{U} \to \widetilde{D}$ in Fig. 1c is terminal, because it remains when the children of the object-level variable U are cut (indeed, U has no children), and disappears if we cut D off from its children (since then D doesn't affect X, and hence doesn't affect U). In contrast, $\widetilde{X} \to \widetilde{D}$ is non-terminal, because if the object-level link $X \to U$ is cut (i.e., the agent's position is made independent of it finding cheese), then the agent will cease adapting its policy to changes in the slip probability p. The labelling of terminal links will be used in Section 3.4 to determine decision and utility nodes.

3.3. Discovering edge-labelled, mechanised causal graphs

Edge-labelled, mechanised causal graphs can be inferred from interventional data. By the definition of a causal edge $W \to V$, if one applies interventions to all nodes except V, and vary these interventions only at W, then V should respond if and only if there is an edge $W \to V$ (this holds true even in cyclic SCMs). This *leave-one-out* strategy[6] for causal disovery is described below:

[5] Strictly, for complete precision one would need an infinite number of games. In practice, dynamics may not converge, which would make the measurement harder.

[6] The efficiency of this strategy is discussed in Section 5.5.2.

Leave-one-out causal discovery

```
Input: Interventional distributions $\mathcal{I} = \{P(\boldsymbol{V} \mid \text{do}(\boldsymbol{Y} = \boldsymbol{y}))\}$ over variables $\boldsymbol{V}$
$E \leftarrow \varnothing$
for $V \in \boldsymbol{V}$
    for $W \in \boldsymbol{V} \setminus \{V\}$
        $\boldsymbol{Y} \leftarrow \boldsymbol{V} \setminus \{V, W\}$
        for $\boldsymbol{y} \in \text{dom}(\boldsymbol{Y})$ and $w, w' \in \text{dom}(W)$
            if $P(V \mid \text{do}(\boldsymbol{Y} = \boldsymbol{y}, W = w)) \neq P(V \mid \text{do}(\boldsymbol{Y} = \boldsymbol{y}, W = w'))$
                $E \leftarrow E \cup (W, V)$
                break
Output: $(\boldsymbol{V}, E)$
```

Lemma 1 *(Leave-one-out causal discovery). Applied to the set of interventional distributions generated by a (potentially cyclic) SCM,* Leave-one-out causal discovery *returns the correct causal graph.*

Proof. Immediate from the definitions of SCM and causal graph, see Section 2. □

Algorithm 1 applies *Leave-one-out causal discovery* to the combined set of object-level and mechanism variables of a mechanised SCM, and then infers edge-labels using structural mechanism interventions on object-level children. The interventions are exhaustive, selecting over all possible values for the variables.

Algorithm 1 Edge-labelled mechanised causal graph discovery.

```
Input: Interventional distributions $\mathcal{I} = \{P(\boldsymbol{\mathcal{V}} \mid \text{do}(\boldsymbol{Y} = \boldsymbol{y}))\}$ over variables $\boldsymbol{\mathcal{V}}$, partitioned into object-level, $\boldsymbol{V}$, and mechanism, $\widetilde{\boldsymbol{V}}$ variables.
$(\boldsymbol{\mathcal{V}}, E) \leftarrow$ leave-one-out-causal-discovery($\{P(\boldsymbol{\mathcal{V}} \mid \text{do}(\boldsymbol{Y} = \boldsymbol{y}))\}$)
$E^{\text{obj}} \leftarrow \{(W, V) \in E : W, V \in \boldsymbol{V}\}$
$E^{\text{mech}} \leftarrow \{(W, V) \in E : W, V \in \widetilde{\boldsymbol{V}}\}$
$E^{\text{func}} \leftarrow \{(W, V) \in E : V \in \boldsymbol{V}, W \in \widetilde{\boldsymbol{V}}\}$
if $E \neq E^{\text{obj}} \cup E^{\text{mech}} \cup E^{\text{func}}$ or $\exists V : |\{(W, V) \in E^{\text{func}} : W \in \widetilde{\boldsymbol{V}}\}| \neq 1$
    Error: graph is not a mechanised SCM
$E^{\text{term}} \leftarrow \varnothing$
for $(\widetilde{W}, \widetilde{V}) \in E^{\text{mech}}$
    potentially_terminal ← True
    $\widetilde{\boldsymbol{Y}} \leftarrow \widetilde{\boldsymbol{V}} \setminus \{\widetilde{W}, \widetilde{V}\}$
    for interventions $\widetilde{\boldsymbol{y}} \cup \widetilde{\mathbf{ch}^V}$ that are structural for $\mathbf{Ch}^V$, and $\widetilde{w}, \widetilde{w'}$
        if $P(\widetilde{V} \mid \text{do}(\widetilde{\boldsymbol{y}}, \widetilde{\mathbf{ch}^V}, \widetilde{w})) \neq P(\widetilde{V} \mid \text{do}(\widetilde{\boldsymbol{y}}, \widetilde{\mathbf{ch}^V}, \widetilde{w'}))$
            potentially_terminal ← False
            break
    if not potentially_terminal
        continue                                    ▷ skip to next iteration of outer for loop
    for interventions $\widetilde{\boldsymbol{y}} \cup \widetilde{\mathbf{ch}^W}$ that are structural for $\mathbf{Ch}^W$, and $\widetilde{w}, \widetilde{w'}$
        if $P(\widetilde{V} \mid \text{do}(\widetilde{\boldsymbol{y}}, \widetilde{\mathbf{ch}^W}, \widetilde{w})) \neq P(\widetilde{V} \mid \text{do}(\widetilde{\boldsymbol{y}}, \widetilde{\mathbf{ch}^W}, \widetilde{w'}))$
            $E^{\text{term}} \leftarrow E^{\text{term}} \cup (\widetilde{W}, \widetilde{V})$
            break
Output: $(\boldsymbol{V} \cup \widetilde{\boldsymbol{V}}, E^{\text{obj}} \cup E^{\text{mech}} \cup E^{\text{func}}, E^{\text{term}})$
```

Lemma 2 *(Discovery of mechanised SCM). Applied to the set of interventional distributions generated by a mechanised SCM in which structural mechanism interventions are available for all nodes, Algorithm 1 returns the correct edge-labelled mechanised causal graph.*

Proof. The algorithm checks the conditions in Definitions 1 and 3. □

Applied to the mouse example of Fig. 1, Algorithm 1 would take interventional data from the system and draw the edge-labelled mechanised causal graph in Fig. 1c. For example, the edge $(\widetilde{U}, \widetilde{D})$ will be discovered because the mouse's decision rule will change in response to a change in the distribution for cheese location.

3.4. Discovering game graphs

To discover agents, we can convert an edge-labelled mechanised causal graph into a game graph as specified by Algorithm 2: decision nodes are identified by their mechanisms having incoming terminal edges (Line 4), while utility nodes are identified by their mechanisms having outgoing terminal edges (Line 5). Decisions and utilities that are in the same connected component in the terminal edge graph receive the same colouring, which is distinct from the other components. On Line 10, Connected$_{term}(V)$ is the set $\{W \in \boldsymbol{V} \mid \widetilde{W} -- \widetilde{V}\}$ of nodes connected to V along terminal edges (here $\widetilde{W} -- \widetilde{V}$ denotes a *path* in the terminal edge graph). This set could be found by a search algorithm, such as breadth-first search.

Algorithm 2 will identify any decision node D under the following conditions (though it may generate false positives):

Algorithm 2 *Agency Identification.* Converts edge-labelled mechanised causal graph to game graph.

```
Input: An edge-labelled mechanised causal graph $\mathcal{C} = (\boldsymbol{\mathcal{V}}, E)$, with nodes $\boldsymbol{\mathcal{V}} = \boldsymbol{V} \cup \widetilde{\boldsymbol{V}}$ and edges $E = E^{\text{obj}} \cup E^{\text{func}} \cup E^{\text{mech}}$, with $E^{\text{term}} \subseteq E^{\text{mech}}$.
$\boldsymbol{D} \leftarrow \varnothing$
$\boldsymbol{U} \leftarrow \varnothing$
for $(\widetilde{W}, \widetilde{V}) \in E^{\text{term}}$
    $\boldsymbol{D} \leftarrow \boldsymbol{D} \cup \{V\}$
    $\boldsymbol{U} \leftarrow \boldsymbol{U} \cup \{W\}$
$\boldsymbol{X} \leftarrow \boldsymbol{V} \setminus \{\boldsymbol{U} \cup \boldsymbol{D}\}$
Coloured $\leftarrow \varnothing$
for $V \in \boldsymbol{U} \cup \boldsymbol{D}$
    if $V \notin$ Coloured
        Connected$_{term}(V) \leftarrow$ NewColour
        Coloured $\leftarrow$ Coloured $\cup$ Connected$_{term}(V)$
Output: game graph $\mathcal{G} = (N, \boldsymbol{V}, E)$, where $\boldsymbol{V} = \boldsymbol{X} \cup \boldsymbol{U} \cup \boldsymbol{D}$ and $E = E^{\text{obj}}$.
```

- $\boldsymbol{V}$ contains a utility node U, or a mediator node X on a directed path from D to U.
- The utility/mediator node must be sufficiently important to the agent controlling D that its mechanism shapes the agents behaviour.
- Mechanism interventions are available that change the agent's optimal policy for controlling U (or X).
- These mechanism interventions are operationalised in a way that the agent's policy can respond to the changes they imply.

Under the following stronger assumptions,[7] Algorithm 2 is guaranteed to produce a fully correct game graph (without false positives). These assumptions are most easily stated using mechanised SCMs with labelled decision and utility nodes. Following Hammond et al. [32], we call such objects *mechanised games*.

For our first assumption, the following definition will be helpful.

Definition 4. For a game graph, $\mathcal{G}$, we define the *agent graph* to be the graph $\mathcal{G}^A = (\boldsymbol{D}^A \cup \boldsymbol{U}^A, E^A)$, where the edge $(D, U) \in \boldsymbol{D}^A \times \boldsymbol{U}^A$ belongs to E^A if and only if there is a directed path $D \dashrightarrow U \in \mathcal{G}$ that doesn't pass through any $U' \in \boldsymbol{U}^A \setminus \{U\}$. We define the *decision-utility graph* to be the graph $\mathcal{G}^{\boldsymbol{DU}} = (\boldsymbol{D} \cup \boldsymbol{U}, \cup_A E^A)$.

For example, the decision-utility graph of Fig. 1b consists of two nodes, D and U, and an edge (D, U) as there is a directed path D to U that is not mediated by other utility nodes. One further piece of terminology we use is that a DAG is weakly connected if replacing all of its directed edges with undirected edges produces a connected graph, i.e. one in which every pair of vertices is connected by some path. A weakly connected component is a maximal subgraph such that all nodes are weakly connected. For example, the decision-utility graph of Fig. 1b is connected, and consists of a single connected component (the agent graph for the mouse).

Our first assumption uses these definitions as follows:

Assumption 1. Each weakly connected component of the decision-utility graph is an agent graph, and contains at least one decision and one utility node.

The intuition behind this assumption is that if there was a disconnected component in the agent graph, then the decisions in that component could be reasoned about independently from the rest of the decisions, and there would be no way to experimentally distinguish whether those independent decisions were made by a separate agent. So we make this as a simplifying assumption that only separate agents reason about their decisions independently. An example of a game ruled out by this assumption is Fig. 8, in which a decision doesn't directly cause its utility. A further example ruled out is Fig. 7 in which there is just one weakly connected component of the decision-utility graph but there are two agent graphs. A final example ruled out can be found in Fig. D.12.

Assumption 2. For any set of mechanism interventions, every agent optimises expected utility (plays best response) in every decision context, i.e. agents play a subgame perfect equilibrium.

Assumption 2 implies that mechanism interventions are operationalised in a way that agents can appropriately respond to them, that agents are trying to optimise their utility nodes, and that object-level links going into the decision can be interpreted as information links (since agents adapt appropriately to the outcomes of the decision parents). An example that breaks this assumption is an agent that uses a suboptimal policy in a game.

[7] We consider examples of breaking the first of these assumptions in Section 4.6.

Assumption 3. Agents have a preferred ordering over decision rules, so that if two or more decision rules obtain the same (optimal) expected utility in all decision contexts, the agent will always pick the first optimal decision rule according to the order.

This ensures no unmotivated switches occur – so that agents don't switch decision rule in response to mechanism interventions which have no effect on the optimality of that decision rule. An example that breaks this assumption is an agent that is indifferent between two flavours of ice cream, and decides between them based on an irrelevant feature of the environment, such as the day of the week.

Assumption 4. Only decision nodes, $D \in \boldsymbol{D}$, have mechanisms, $\widetilde{D}$, with ingoing terminal edges.

This is a weak form of the popular *independent causal mechanism* assumption [50], discussed further in Section 5.6, preventing dependencies between certain mechanisms.

Assumption 5. For each node V, interventions on $\widetilde{V}$ can instantiate any deterministic function relating V to its parents (when V lacks parents, it can be set to any constant value in $\mathrm{dom}(V)$).

This is to ensure that we can enact the necessary soft interventions, in a way that the agent is able to adapt to. An example which breaks this assumption is one in which it's physically impossible to make a certain mechanism intervention – for example, it may be hard to change the fact that both rain and a sprinkler make the pavement wet.

We are now ready to establish a correctness result for Algorithm 2.

Theorem 1 *(Correctness of Algorithms 1 and 2). Let $\widetilde{\mathcal{M}}_{real}$ be a mechanised causal game satisfying Assumptions 1 to 5. Let $\mathcal{G}_{model}$ be the game graph resulting from applying Algorithm 1 followed by Algorithm 2 to $\widetilde{\mathcal{M}}_{real}$. Then $\mathcal{G}_{model} = \mathcal{G}_{real}$.*

Proof. We establish that the algorithm infers the correct object-level causal structure, the correct labelling of decision and utility nodes (and hence of chance nodes), and the correct colouring of the same. Our proof strategy will be to show that any decision will get mapped to a decision, and than any non-decision will not get mapped to a non-decision. Likewise for utilities.

Causal structure The only structural difference between a game and an SCM is the presence of information links in the game. By Assumption 5, we can impute an arbitrary decision rule to any decision, that makes it depend on all its observations. Thereby all information links are causal links.

Decision: We first show that all and only decisions get mapped to decisions. We begin by showing that any decision will get mapped to a decision. The intuition of what follows is that we'll transform (via mechanism interventions) to a game where if we cutoff the decision from its children, it will stop responding to changes in its associated utility's mechanism. Let $D \in \boldsymbol{D}^A$ be a decision variable for agent A in $\widetilde{\mathcal{M}}_{\text{real}}$. By Assumption 1 there exists a utility variable $U \in \boldsymbol{U}^A$ such that there's a directed path, p, from D to U not passing through any other utility node of A. By means of mechanism interventions, we can ensure that U is either 0 or 1 depending on the value of D by copying the value of D along p, using deterministic functions (Assumption 5). All other nodes ignore D (also by means of mechanism interventions). Agent A chooses a decision rule setting U to 1 (Assumption 2). If we do a mechanism intervention to cut off the effect of D on its children, then no mechanism intervention on U will cause agent A to choose a different decision rule as all decision rules would have the same expected utility (and Assumption 3 rules out unmotivated switches). Thus, Lines 11-14 will not change the `potentially_terminal` to *False*, so we don't hit the continue statement on Line 16. We then enter Lines 17-20. There is an intervention that inverts the function governing U, and cuts off all of its effects on its children, so agent A will choose a different decision rule and then the edge $(\widetilde{U}, \widetilde{D})$ gets added to the terminal edge set. Algorithm 2 then correctly identifies D as a decision.

Conversely, assume $V \in \boldsymbol{V} \setminus \boldsymbol{D}$ is a non-decision. By Assumption 4, for any $(\widetilde{W}, \widetilde{V}) \in E^{\text{mech}}$ Lines 11-14 will change the `potentially_terminal` to *False*, so we hit the continue statement 16 and $(\widetilde{W}, \widetilde{V})$ can't be added to E^{term}. Algorithm 2 then doesn't identify V as a decision.

Utility: We next show that all and only utilities get mapped to utilities. Let $U \in \boldsymbol{U}^A$ be a utility variable for agent A in $\widetilde{\mathcal{M}}_{\text{real}}$. By Assumption 1 there exists a decision variable $D \in \boldsymbol{D}^A$ such that there's a directed path, p, from D to U not passing through any other utility node of A. By the same construction as for decision nodes above, Algorithm 1 will discover a terminal mechanism edge $(\widetilde{U}, \widetilde{D})$. Therefore Algorithm 2 identifies U to be a utility as desired.

Conversely, consider a non-utility node, $W \notin \boldsymbol{U}$, and some other node, $V \in \boldsymbol{V} \setminus \{W\}$, with structural interventions cutting off $\mathbf{Ch}^W$ and interventions on all mechanisms except $\widetilde{V}$. Suppose, for contradiction, there exists a terminal edge $(\widetilde{W}, \widetilde{V})$. By Assumption 4, there will be a terminal edge $(\widetilde{W}, \widetilde{V})$ only if V is a decision. Further, by Assumptions 2 and 3 the expected utility must be affected by the change in $\widetilde{W}$. But since we've intervened on all mechanisms except $\widetilde{V}$, the only effect $\widetilde{W}$ can have on the expected utility is via W. But $W \notin \boldsymbol{U}$, and $\mathbf{Ch}^W$ are not affected (since they've been cut off), so $\widetilde{W}$ cannot affect expected utility. Therefore, only utility variables get outgoing edges in E^{term} from Algorithm 1, and Algorithm 2 does not assign V to be a utility.

We have thus shown that all and only decisions nodes get mapped to decisions, and similarly for utilities. All that are left are chance nodes, and these must be mapped to chance nodes (since only decisions/utilities get mapped to decisions/utilities).

Colouring: By Assumption 1 for any agent, A, and for any decision $D \in D^A$, there exists $U \in U^A$ with $(D, U) \in E^A$. By the above paragraphs, we must have that E^{term} contains the edge $(\widetilde{U}, \widetilde{D})$, and further, by the converse arguments, the only edges in E^{term} are of the form $(\widetilde{U}, \widetilde{D})$ with $D \in D^A, U \in U^A$ and $(D, U) \in E^A$ for some A, which means E^{term} is a disjoint union of $\widetilde{E}^A$, in which each edge of $\widetilde{E}^A$ is the reverse of an edge in E^A. By Assumption 1, the weakly connected components of $\mathcal{G}^{\boldsymbol{DU}}$ are the $\mathcal{G}^A$, and so the $\widetilde{E}^A$ are each weakly connected, and disconnected from each other. The colouring of Algorithm 2 colours each vertex of a connected component the same colour, and distinctly to all other components, and thus is correct. □

3.5. Mechanism identification procedure

In the last section we demonstrated an algorithm that, when applied after a causal discovery algorithm, can identify the underlying game graph of a system. In this section we will show the converse, that if one already has a game graph, one can convert it into an edge-labelled mechanised causal graph. The interpretation is that the same underlying system can equivalently be represented either as an edge-labelled mechanised causal graph, which is a causal representation of the system, or as a game graph, which is a decision-theoretic representation of the system.

We first prove a Lemma relating the mechanised causal graph produced by Algorithm 1 to *strategic relevance* [39], which captures which other decision rules are relevant for optimising the decision rule at D. Koller and Milch give a sound and complete graphical criterion for strategic relevance, called *s-reachability*,[8] where $V \neq D$ is *s-reachable* from $D \in \boldsymbol{D}^A$, for agent A, if, in a modified game graph $\hat{\mathcal{G}}$ with a new parent $\hat{V}$ added to V, we have $\hat{V} \not\perp_{\hat{\mathcal{G}}} \boldsymbol{U}^D \mid \mathbf{Fa}^D$, where $\boldsymbol{U}^D$ is the set of utilities for agent A that are descendants of D (i.e. $\boldsymbol{U}^D = \boldsymbol{U}^A \cap \mathbf{Desc}^D$ for $D \in \boldsymbol{D}^A$) and $\not\perp$ denotes d-connection [46]. In the game graph in Fig. 1b, both X and U are s-reachable from D.

Lemma 3. *Let $\widetilde{\mathcal{M}}$ be a mechanised causal game satisfying Assumptions 1 to 5, containing an agent, A, with decision variables $\boldsymbol{D}^A$ and utility variables $\boldsymbol{U}^A$, and let $\mathcal{C}$ be the mechanised causal graph with edges $E^{obj} \cup E^{func} \cup E^{mech}$, and $E^{term} \subseteq E^{mech}$, which results from applying Algorithm 1 to $\widetilde{\mathcal{M}}$. Then*

1. *For $D \in \boldsymbol{D}^A$, that the node $Y \in \boldsymbol{V} \setminus D$ is s-reachable from D is a necessary and sufficient condition for $(\widetilde{Y}, \widetilde{D}) \in E^{mech}$ (this places no restriction on $(\widetilde{Y}, \widetilde{W}) \in E^{mech}$ for $W \notin \boldsymbol{D}$).*
2. *Further, for $Y \in \boldsymbol{U}^A$, that the existence of a directed path $D \dashrightarrow Y$ not through another $U' \in \boldsymbol{U}^A \setminus \{Y\}$ is a necessary and sufficient condition for $(\widetilde{Y}, \widetilde{D}) \in E^{term}$.*

Proof. Necessity of 1: We largely follow the soundness direction of Koller and Milch [39], Thm 5.1, with an extension to relate this to a mechanised causal graph discovered by Algorithm 1. The proof strategy is to suppose that Y is *not* s-reachable from D, and show this implies $(\widetilde{Y}, \widetilde{D}) \notin E^{\text{mech}}$.

We perform the mechanism interventions, $\text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y})$ and $\text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y}')$. Since D is a decision variable, by Lemma 5.1 of Koller and Milch [39] the optimal decision rule $\pi^D_{\widetilde{y}}(\mathbf{pa}^D, \mathcal{E}^D)$ under $\text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y})$ must be a solution of the following optimisation problem

$$\arg\max_{\pi} \sum_{d \in \text{dom}(D)} \pi(d) \sum_{u \in \text{dom}(U^D)} P(u \mid d, \mathbf{pa}^D, \text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y})) \cdot u \tag{2}$$

and similarly for the decision rule $\pi^D_{\widetilde{y}'}$ under $\text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y}')$.

Now suppose that Y is *not* s-reachable from D, then by Lemma 5.2 of Koller and Milch [39], we have that $P(u \mid d, \mathbf{pa}^D, \text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y})) = P(u \mid d, \mathbf{pa}^D, \text{do}(\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{w}}, \widetilde{Y} = \widetilde{y}'))$, and so the two optimization problems are the same. Since they are solutions of the same optimization problem, and by Assumptions 2 and 3 the agents choose decision rules which make up subgame-perfect equilibrium, this leads to the same decision rule in each intervened game $\pi^D_{\widetilde{y}}(\mathbf{pa}^D, \mathcal{E}^D) = \pi^D_{\widetilde{y}'}(\mathbf{pa}^D, \mathcal{E}^D)$. This holds for any $\widetilde{\boldsymbol{W}}, \widetilde{y}, \widetilde{y}'$ and so Algorithm 1 does not draw an edge, i.e. $(\widetilde{Y}, \widetilde{D}) \notin E^{\text{mech}}$, as was to be shown.

Necessity of 2: As argued in Theorem 1 (colouring), $(\widetilde{Y}, \widetilde{D}) \in E^{\text{term}}$ implies $(D, Y) \in E^A$, which by Definition 4 means there exists a directed path $D \dashrightarrow Y$ not through another $U' \in \boldsymbol{U}^A \setminus \{Y\}$.

Sufficiency of 1: We can use soft interventions on object-level variables to construct the same model as used in the existence proof for Theorem 5.2 of Koller and Milch [39]. We note that the proof for Theorem 5.2 of Koller and Milch [39] is written for another decision variable D' being s-reachable from D. But the proof itself makes no use of the special nature of D' as a decision, rather than any other type of variable, and so it also applies to any variable $Y \in \boldsymbol{V} \setminus \{D\}$.

[8] Our definition here generalises the definition from Koller and Milch [39] to include non-decision variables as being s-reachable, following Hammond et al. [31].

Suppose Y is s-reachable from D in $\widetilde{\mathcal{M}}$. It follows from Theorem 5.2 of Koller and Milch [39] that the optimal decision rule for D will be different under these mechanism interventions (i.e. those needed to construct the game of Theorem 5.2 of Koller and Milch [39]), when different mechanism interventions are applied to Y. Hence Algorithm 1 will draw an edge $(\widetilde{Y}, \widetilde{D}) \in E^{\text{mech}}$.

Sufficiency of 2: By the arguments in Theorem 1 (decision, utility) the existence of a directed path $D \dashrightarrow Y$ not through another $U' \in \boldsymbol{U}^A \setminus \{Y\}$ means that $(\widetilde{Y}, \widetilde{D}) \in E^{\text{term}}$. □

Algorithm 3 *Mechanism Identification*. Convert game graph to edge-labelled mechanised causal graph.

```
Input: game graph $\mathcal{G} = (N, \boldsymbol{V}, E)$
 1: $E^{\text{term}} \leftarrow \varnothing$
 2: $\widetilde{\boldsymbol{V}} \leftarrow \varnothing$
 3: for $V \in \boldsymbol{V}$
 4:     $E \leftarrow E \cup (\widetilde{V}, V)$
 5:     $\widetilde{\boldsymbol{V}} \leftarrow \widetilde{\boldsymbol{V}} \cup \mathrm{Node}(\widetilde{V})$,
 6: $\boldsymbol{\mathcal{V}} \leftarrow \boldsymbol{V} \cup \widetilde{\boldsymbol{V}}$,
 7: for $A \in N$
 8:     for $D \in \boldsymbol{D}^A$
 9:         for $V \in \boldsymbol{V} \setminus \{D\}$
10:             $\hat{\mathcal{G}}$ is $\mathcal{G}$ with a new parent $\hat{V}$ added to $V$
11:             if $\hat{V} \not\perp_{\hat{\mathcal{G}}} U^D \mid \{\mathbf{Pa}^D \cup D\}$
12:                 $E \leftarrow E \cup (\widetilde{V}, \widetilde{D})$
13:             if $\exists$ directed path $D \dashrightarrow V$ not through another $U' \in \boldsymbol{U}^A \setminus \{V\}$
14:                 $E^{\text{term}} \leftarrow E^{\text{term}} \cup (\widetilde{V}, \widetilde{D})$
Output: mechanised causal graph $\mathcal{C} = (\boldsymbol{\mathcal{V}}, E)$, $E^{\text{term}}$
```

The conversion from game graph to mechanised causal graph is done by Algorithm 3, *Mechanism Identification*, which identifies mechanisms by converting a game graph into a mechanised causal graph. It first takes the game graph edges and on Lines 3-5 adds the function edges. Lines 8-14 then add the mechanism edges based on s-reachability: if a node V is s-reachable from D in the game graph, then we include an edge $(\widetilde{V}, \widetilde{D})$ in the mechanised causal graph. Further, it adds a terminal edge when there's a directed path from one of an agent's decisions to one of its utilities, that doesn't pass through another of its utilities. We now establish that Algorithm 2 and Algorithm 3 are inverse to each other. We will use the shorthand $a_i(x)$, for $i = 1, 2, 3$ to refer to the result of algorithm a_i on object x, where e.g. x is a game graph.

Theorem 2 *(Algorithm 2 is a left inverse of Algorithm 3). Let $\mathcal{G}$ be a mechanised game graph satisfying Assumptions 1 to 5, and let $\mathcal{C}$ be the mechanised causal graph resulting from applying Algorithm 3 to it. Then applying Algorithm 2 on $\mathcal{C}$ reproduces $\mathcal{G}$. That is, $a_2(a_3(\mathcal{G})) = \mathcal{G}$.*

Proof. All edges between nodes are the same in $\mathcal{G}$ and $a_2(a_3(\mathcal{G}))$, because neither Algorithm 2 or Algorithm 3 changes the object-level edges. We will now show that the node types are the same in both.

Decision: Let A be an agent with utilities $\boldsymbol{U}^A$ and let $D \in \boldsymbol{D}^A$, then by Assumption 1 $\exists U \in \boldsymbol{U}^A$ and a directed path $D \dashrightarrow U$ not through another $U' \in \boldsymbol{U}^A \setminus \{U\}$. Algorithm 3 Lines 13-14 add $(\widetilde{U}, \widetilde{D})$ to E^{term}. Algorithm 2 then adds D to the set of decisions, as desired.

Let $V \in \boldsymbol{V} \setminus \boldsymbol{D}$. Algorithm 3 Lines 13-14 only adds terminal mechanism edges going into decisions, and Algorithm 2 then doesn't add V to the set of decisions, as desired.

Utility: Let A be an agent with decisions $\boldsymbol{D}^A$ and let $U \in \boldsymbol{U}^A$, then by Assumption 1 $\exists D \in \boldsymbol{D}^A$ and a directed path $D \dashrightarrow U$ not through another $U' \in \boldsymbol{U}^A \setminus \{U\}$. So Algorithm 3 Lines 13-14 add $(\widetilde{U}, \widetilde{D})$ to E^{term}. Algorithm 2 then adds U to the set of utilities, as desired.

Let $V \in \boldsymbol{V} \setminus \boldsymbol{U}$. Algorithm 3 Lines 13-14 only adds terminal edges going out of utilities, so there will be no edge out of $\widetilde{V}$ in E^{term}. Algorithm 2 then doesn't add V to the set of utilities, as desired.

Colouring: By above paragraphs, the node types and edges are the same in both $a_2(a_3(\mathcal{G}))$ and $\mathcal{G}$. By Assumption 1 the colouring in $\mathcal{G}$ is a property of the connectedness and hence will be the same in $a_2(a_3(\mathcal{G}))$. □

We now consider the other direction: beginning with a mechanised causal graph, can we transform it to a game graph and then back to the same mechanised causal graph? In general this isn't possible, because the space of possible mechanised causal graphs is larger than the space of mechanised causal graphs that can be recovered using only the information present in a game graph. In particular, mechanisms with non-terminal incoming mechanism edges do not, in general, get codified in the game graph when using a_2. Further, we will find it useful to consider only those mechanised causal graphs that are producible from a mechanised causal game satisfying Assumptions 1 to 5, as this will enable us to use Lemma 3. Thus, in the next theorem, we restrict the space of mechanised causal graphs we consider.

Theorem 3 *(Algorithm 3 is a left inverse of Algorithm 2). Let $\mathcal{C}$ be a mechanised causal graph such that*

- *there exists a mechanised causal game, $\widetilde{\mathcal{M}}$, satisfying Assumptions 1 to 5, such that $a_1(\widetilde{\mathcal{M}}) = \mathcal{C}$;*
- *any node with an incoming mechanism edge also has an incoming terminal edge, i.e. $\forall(\widetilde{V}, \widetilde{W}) \in E^{mech}, \exists(\widetilde{V}', \widetilde{W}) \in E^{term}$, for some $\widetilde{V}' \in \widetilde{\boldsymbol{V}} \setminus \{\widetilde{W}\}$.*

Then $a_3(a_2(\mathcal{C})) = \mathcal{C}$.

Proof. The edges in $E^{\text{obj}}, E^{\text{func}}$ are the same in both $\mathcal{C}$ and $a_3(a_2(\mathcal{C}))$, since neither algorithm changes the object-level edges, and all mechanised causal graphs over object-level variables $\boldsymbol{V}$ have the same edges in E^{func}, i.e. $\{(\widetilde{V}, V)\}_{V \in \boldsymbol{V}}$, which are added in Algorithm 3 Lines 3-5. We now show why edges in E^{term} and E^{mech} are the same in both.

From the theorem statement, $\exists\widetilde{\mathcal{M}}$ such that $a_1(\widetilde{\mathcal{M}}) = \mathcal{C}$. Let $\mathcal{G}$ be the game graph of $\widetilde{\mathcal{M}}$.

$$
\begin{aligned}
& (\widetilde{U}, \widetilde{D}) \in E^{\text{term}} \text{ of } \mathcal{C} \\
\iff & (\widetilde{U}, \widetilde{D}) \in E^{\text{term}} \text{ of } a_1(\widetilde{\mathcal{M}}) \\
\iff & \exists \text{ directed path } D \dashrightarrow U \text{ not through } U' \in \boldsymbol{U}^A \setminus \{U\} \text{ in } \mathcal{G} \quad \text{(by Lemma 3)} \\
\iff & \exists \text{ directed path } D \dashrightarrow U \text{ not through } U' \in \boldsymbol{U}^A \setminus \{U\} \text{ in } a_2(a_1(\widetilde{\mathcal{M}})) \quad \text{(by Theorem 1)} \\
\iff & \exists \text{ directed path } D \dashrightarrow U \text{ not through } U' \in \boldsymbol{U}^A \setminus \{U\} \text{ in } a_2(\mathcal{C}) \\
\iff & (\widetilde{U}, \widetilde{D}) \in E^{\text{term}} \text{ of } a_3(a_2(\mathcal{C})) \quad \text{(by Algorithm 3 Lines 13-14)}.
\end{aligned}
$$

From the theorem statement, $\forall(\widetilde{V}, \widetilde{W}) \in E^{\text{mech}}, \exists(\widetilde{V}', \widetilde{W}) \in E^{\text{term}}$, for some $\widetilde{V}' \in \widetilde{\boldsymbol{V}} \setminus \{\widetilde{W}\}$. By Assumption 4, W must be a decision in $\widetilde{\mathcal{M}}$. Thus, we only need consider edges of the form $(\widetilde{V}, \widetilde{D}) \in E^{\text{mech}}$ where $D \in \boldsymbol{D}$.

$$
\begin{aligned}
& (\widetilde{V}, \widetilde{D}) \in E^{\text{mech}} \text{ of } \mathcal{C} \\
\iff & (\widetilde{V}, \widetilde{D}) \in E^{\text{mech}} \text{ of } a_1(\widetilde{\mathcal{M}}) \\
\iff & V \text{ is s-reachable from } D \text{ in } \mathcal{G} \quad \text{(by Lemma 3)} \\
\iff & V \text{ is s-reachable from } D \text{ in } a_2(a_1(\widetilde{\mathcal{M}})) \quad \text{(by Theorem 1)} \\
\iff & V \text{ is s-reachable from } D \text{ in } a_2(\mathcal{C}) \\
\iff & (\widetilde{V}, \widetilde{D}) \in E^{\text{mech}} \text{ of } a_3(a_2(\mathcal{C})) \quad \text{(by Algorithm 3 Lines 11-12)}. \quad \square
\end{aligned}
$$

4. Examples

We now look at example applications of our algorithms, which help the modeller to draw the correct game graph to describe a system. First, we revisit and add some more detail to our earlier mouse example (Section 4.1). Next, we show that mechanised causal graphs are essential for reliable incentive analysis, by illustrating how incentive analysis can fail when using just the game graph to model a recommender system (Section 4.2). Our algorithms also help avoid an easy modelling mistake in a multi-agent setting (Section 4.3). In Section 4.4, we illustrate how our algorithms can increase modelling precision and clarify what information is available for decisions in a previously published example where an RL agent can have the effects of its actions modified by an overseer [40]. Finally, we cover some corner cases with zero agents (Section 4.5), and violated assumptions (Section 4.6).

4.1. Simple example

We begin by considering the simple example of Fig. 1 in more detail. The underlying system has game graph $\mathcal{G}_{\text{real}}$, displayed in Fig. 1b, with D a decision node, X a chance node and U a utility node. Recall that all variables are binary; $X = D$ with probability p, $X = 1 - D$ with probability $1 - p$; and $U = X$ with probability q, $U = 1 - X$ with probability $1 - q$. Having specified the causal game, we can now describe the optimal decision rule – this depends on the values of p and q: if $p, q > 0.5$ or $p, q < 0.5$, then $D = 1$ is optimal, if $p < 0.5, q > 0.5$ or $p > 0.5, q < 0.5$ then $D = 0$ is optimal, and if either p or q is 0.5, then both $D = 0$ and $D = 1$ are optimal.

We can now consider mechanism interventions, to understand what Algorithm 1 will discover. Suppose we soft intervene on X and U such that $p, q > 0.5$, so that the optimal policy is $D = 1$. When we change the soft intervention on X such that $p < 0.5$, we will see the optimal policy change to $D = 0$. Thus Algorithm 1 draws an edge $(\widetilde{X}, \widetilde{D})$. By a similar argument, it will also draw an edge $(\widetilde{U}, \widetilde{D})$, which will be a terminal edge. Thus Algorithm 1 produces the edge-labelled mechanised causal graph $\mathcal{C}_{\text{model}}$ shown in Fig. 1c. Algorithm 2 then takes $\mathcal{C}_{\text{model}}$ and produces the correct game graph by identifying that only $\widetilde{D}$ has incoming arrows, and so D is the only decision node, and that U is the only variable which has its mechanism with an outgoing terminal edge into the mechanism for D, and hence is a utility. In this simple example, we have recovered the game graph of Fig. 1b.

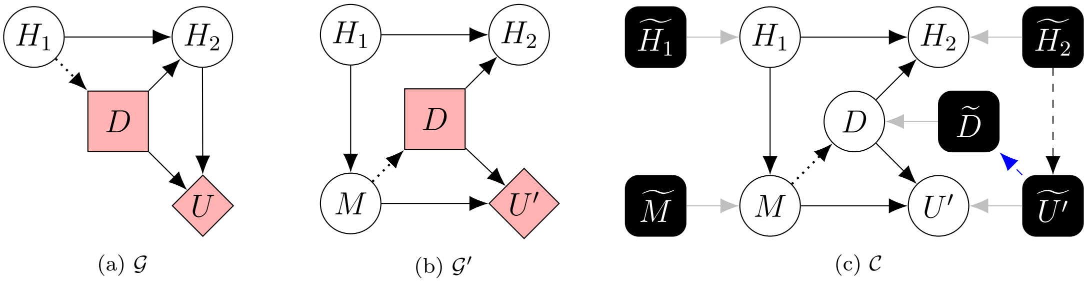


Fig. 3. Recommender system optimising engagement. 3a Agent optimising actual clicks. 3b Agent optimising predicted clicks, from Everitt et al. [18, Fig. 4b]. 3c mechanised causal graph, $\mathcal{C}$, that Algorithm 1 discovers for agent optimising predicted clicks. Note the path $\widetilde{H_2} \to \widetilde{U^D} \to \widetilde{D}$ which implies the recommendation system's policy depends on how a human updates their opinions when shown the recommended content, which is not visible from the game graph.

4.2. Optimising a model of a human

We next consider an example from the influence diagram literature, which highlights when incentive analysis can fail and how mechanised SCMs can avoid these problems.

It has been suggested that a safety problem with content-recommendation systems is that they can nudge users towards more extreme views, to make it easier to recommend content that will generate higher utility for the system (e.g., more clicks), as the extreme views are more easily predictable [5,55,8]. The problem can be modelled as in Fig. 3a [16,20], where an agent is aware of a user's initial preferences H_1 (from their interaction history), and is choosing some content D to show the user in order to maximise their engagement U (e.g. measured as clicks). While the user has their own objectives, these objectives are not among the variables represented here, and so the user is modelled as a non-agent part of the agent's environment (this is appropriate, as we're primarily interested in the content recommender's perspective). The problem illustrated by Fig. 3a is that the agent can satisfy its objective in two ways: either by choosing content that satisfies the user's (initial) preferences, which is the desired solution, but also by changing the user's preferences H_2 so that they fit the content that the agent wants to show. In other words, the agent is incentivised to manipulate the user's preferences.

To combat this, Everitt et al. [18] propose that the system's utility be based on *predicted* clicks using a model M of the user, rather than on *actual* clicks (Fig. 3b reproduces their Fig. 4b). Let U' denote the new "predicted clicks" objective, and note how it is a child of M instead of H_2. Note also that the agent's awarenss of the user's initial preferences is now mediated by the model M (the edge $M \to D$ replaces $H_1 \to D$ in Fig. 3a). Importantly, the incentive for the agent to influence the user's opinion now seems to have disappeared: there is no longer a directed path $D \to H_2 \to U$.

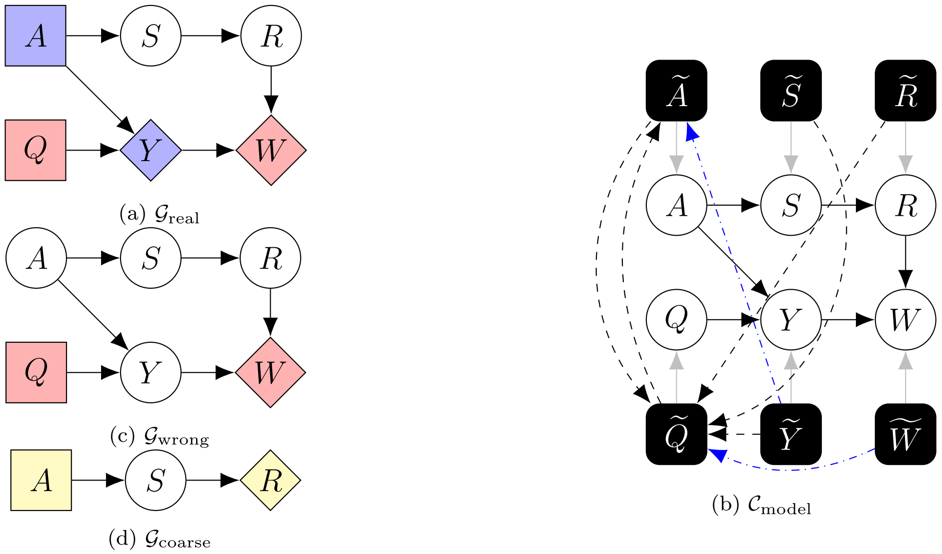
Will this stop the agent from trying to influence the user's opinions? We argue *no*, based on the mechanised causal graph (Fig. 3c). First, there is a terminal edge $(\widetilde{U}', \widetilde{D})$, since this is the goal that the agent is trained to pursue. But should there be an edge $(\widetilde{H}_2, \widetilde{U}')$? This depends on how the user model was obtained. If, as is common in practice, the model was obtained by predicting clicks based on past user data, then changing how a human reacts to recommended content $(\widetilde{H}_2)$, would lead to a change in the way that predicted clicks depend on the model of the original user (i.e. a change in $\widetilde{U}'$). This means that there should be an edge $\widetilde{H}_2 \to \widetilde{U}'$, as we have drawn in Fig. 3c.[9] This in turn means that there is a directed path from $\widetilde{H}_2$ to $\widetilde{D}$, so the recommendation system *is* influencing the human in a goal-directed way! (It is adapting its behaviour to changes in how the human is influenced by its recommendation – cf. discussion in Section 1.3.)

This example casts doubt on the reliability of graphical incentive analysis [18] and its applications [2,16,19,20,40,10]. If different descriptions of the same system yields different conclusions, then graph-based inference does not seem possible. Fortunately, by pinpointing the source of the problem, mechanised SCMs also contain the seed of a solution: graphical incentive analysis can be trusted (only) when all non-decision mechanisms lack ingoing arrows. Indeed, this mirrors the extra assumption needed for the equivalence between games and mechanised SCMs in Theorem 3. As mechanisms are often assumed completely independent, this is often not an unreasonable assumption (see also Section 5.6). Alternatively, it may be possible to use mechanised SCMs to generalise graphical incentive analysis to allow for dependent mechanisms, but we leave investigation of this for future work.

4.3. Actor-critic

Our third example contains multiple agents, and demonstrates an easily made modelling mistake which can be avoided by using our algorithms. It represents an Actor-Critic RL setup for a one-step MDP [56]. Here an *actor* selects action A as advised by a *critic* (Fig. 4a). The critic's action Q states the expected reward for each action (in the form of a vector with

[9] Everitt et al. [18] likely have in mind a different interpretation, where the predicted clicks are a path-specific objective [20], forcing the agent to not use preference manipulation as a means to optimising clicks. But the intended interpretation is ambiguous when looking only at Fig. 3b – the mechanised graph is needed to reveal the difference.



(a) $\mathcal{G}_{\text{real}}$

(c) $\mathcal{G}_{\text{wrong}}$

(d) $\mathcal{G}_{\text{coarse}}$

(b) $\mathcal{C}_{\text{model}}$

Fig. 4. Actor-Critic. 4a True game graph $\mathcal{G}_{\text{real}}$. 4b Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$. From $\mathcal{C}_{\text{model}}$, Algorithm 2 produces the correct game graph by identifying that $\widetilde{A}$ and $\widetilde{Q}$ have incoming arrows, so are decisions, and that Y has its mechanism with an outgoing terminal edge to the mechanism for A so is its utility, whilst W has its mechanism with an outgoing terminal edge to the mechanism for Q, so is its utility. They are coloured differently due to having different utilities. 4c Incorrect game graph for actor-critic. 4d Coarse-grained single-agent game graph.

one element for each possible choice of A, this is often called a *Q-value function*). The action A influences the state S, which in turn determines the reward R. We model the actor as just wanting to follow the advice of the critic, so its utility is $Y = Q(A)$ (the A-th element of the Q vector). The critic wants its advice Y to match the actual reward R. Formally, it optimises $W = -(R - Y)^2$.

Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$, in Fig. 4b. We don't justify all of the mechanism edges, but instead focus on a few of interest. For example, there is an edge $(\widetilde{S}, \widetilde{Q})$ but there is no edge $(\widetilde{S}, \widetilde{A})$, i.e. the critic cares about the state mechanism but the actor does not. The critic cares because it is optimising W which is causally downstream of S, and so the optimal decision rule for Q will depend on the mechanism of S even when other mechanisms are held constant. The dependence disappears if R is cut off from S, so the edge $(\widetilde{S}, \widetilde{Q})$ is not terminal. In contrast, the actor *doesn't* care about the mechanism of S, because Y is *not* downstream of S, so when holding all other mechanisms fixed, varying $\widetilde{S}$ won't affect the optimal decision rule for A. There is however an indirect effect of the mechanism for S on the decision rule for A, which is mediated through the decision rule for Q. Algorithm 2 applied to $\mathcal{C}_{\text{model}}$ produces the correct game graph by identifying that $\widetilde{A}$ and $\widetilde{Q}$ have incoming arrows, and therefore are decisions; that Y's mechanism has an outgoing terminal edge to A's mechanism and so is its utility; and that W's mechanism has an outgoing terminal edge to the mechanism for Q, and so is its utility. The decision-utility graph consists of two connected components, one being (A, Y) and the other (Q, W). The decisions and utilities therefore get coloured correctly.

This can help avoid modelling mistakes and incorrect inference of agent incentives. In particular, Christiano (private communication, 2019) has questioned the reliability of incentive analysis from CIDs, because of an apparently reasonable way of modelling the actor-critic system where the actor is not modelled as an agent, shown in Fig. 4c. Doing incentive analysis on this single-agent diagram would lead to the assertion that the system is not trying to influence the state S or the reward R, because they don't lie on the directed path $Q \to W$ (i.e. neither S nor R has an *instrumental control incentive*; [18]). This would be incorrect, as the system is trying to influence both these variables (in an intuitive and practical sense).

The modelling mistake would be avoided by applying Algorithms 1 and 2 to the underlying system, which produces Fig. 4a, differing from Fig. 4c. The correct diagram (when including these variables) has two agents, and it's not possible to apply the single-agent incentive concept from [18]. Instead, an incentive concept suitable for multi-agent systems would need to be developed. For such a multi-agent incentives concept to be useful, it should capture the influence on S and R jointly exerted by A and Q.

Fig. 4d shows a game graph that involves only a subset of the variables of the underlying system, i.e., a coarse-grained version. This is also an accurate description of the same underlying system, though with less detail. At this coarser level, we find an instrumental control incentive on S and R, as intuitively expected.

4.4. Modified action Markov decision process

Next, we consider an example regarding the redirectability of different RL agents, because it demonstrates how our algorithms can make it clearer what information is available for decisions, as compared to previous work. Langlois and Everitt [40] introduce *modified action Markov decision processes* (MAMDPs) to model a sequential decision-making problem

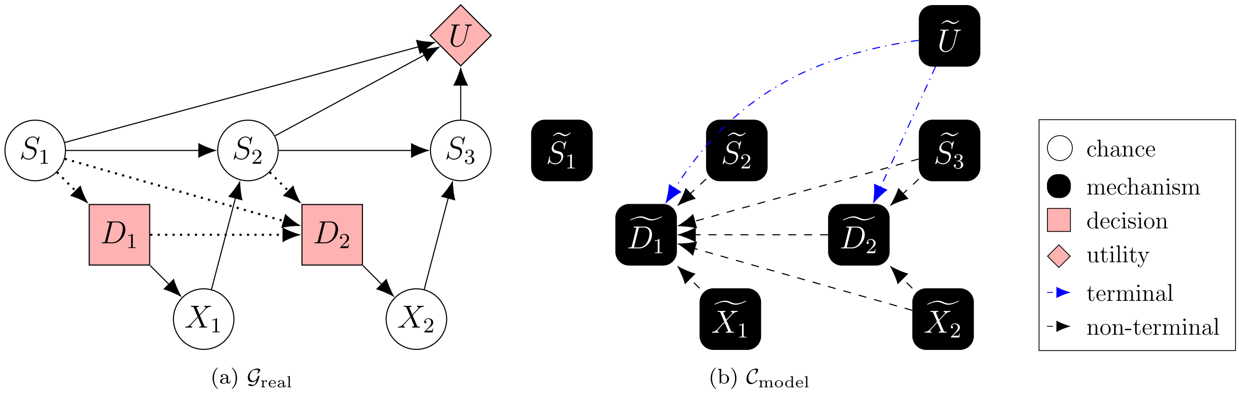


Fig. 5. Modified Action MDP. 5a The underlying system has game graph $\mathcal{G}_{\text{real}}$. 5b Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$ (we display mechanisms only, see Fig. D.10 in Appendix D for the full diagram). Since the utility U is the same, the decisions D_1 and D_2 are coloured the same to show they belong to the same agent.

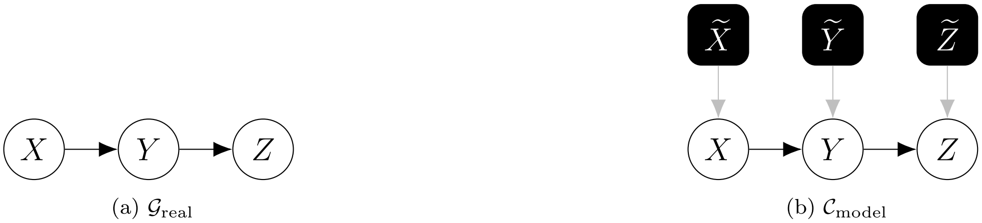

X
Y
Z
(a) $\mathcal{G}_{\text{real}}$
$\widetilde{X}$
$\widetilde{Y}$
$\widetilde{Z}$
X
Y
Z
(b) $\mathcal{C}_{\text{model}}$

Fig. 6. Zero agents. 6a The true game graph $\mathcal{G}_{\text{real}}$ has no decisions or utilities, so is a standard causal Bayesian network. 6b Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$. Algorithm 2 produces the correct game graph by identifying that there are no agents, and just recovers the standard causal Bayesian network.

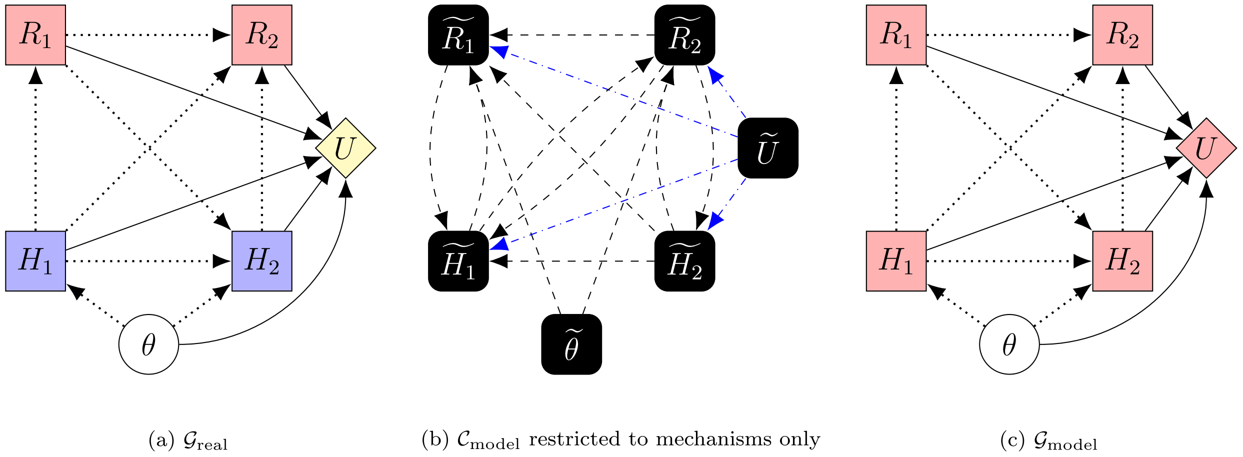
similar to an MDP, but where the agent's decisions, D_t, can be overridden by a human. In the game graph in Fig. 5, this is modelled by D_t only influencing S_t via a chance variable, X_t, which represents the potentially overridden decision.

Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$ in Fig. 5b, where for readability we restrict to mechanisms only – for the full diagram see Fig. D.10 in Appendix D. There are many mechanism edges, so we only elaborate on the interpretation of one of the edges, $(\widetilde{X}_1, \widetilde{D}_1)$, in this mechanised causal graph. This edge represents that the agent's choice of decision rule is influenced by the mechanism for the potentially overridden variable, X_1. In general, in this decision problem, it will be suboptimal to ignore knowledge of the mechanism for the potentially overridden variables. Algorithm 2 applied to $\mathcal{C}_{\text{model}}$ produces the correct game graph by identifying that $\widetilde{D}_1$ and $\widetilde{D}_2$ have incoming terminal edges, so are decisions, and that U has its mechanism with outgoing terminal edges to the mechanisms for decisions D_1 and D_2, and so is a utility. Since the utility is the same, the decisions are coloured the same to show they belong to the same agent.

We note that the game graph diagram presented here in Fig. 5a differs from Figure 2 of Langlois and Everitt [40]. The reason is that we have been stricter about what should appear in a game graph, and what should appear in a mechanised causal graph. In particular, Langlois and Everitt have a node for the decision rule in their game graph, whereas we only have decision nodes in our game graphs, with decision rule nodes only appearing in mechanised causal graphs, along with other mechanism nodes. With this extra strictness comes greater expression and clarity – in our game graph we are clear that the agent's decisions can't condition on the result of the modification, whereas Langlois and Everitt draw an information edge from the modification to the policy, which is a decision node in their diagram. Instead, we represent the fact that the decision rules are influenced by the mechanism for potentially overridden variables by the edges $(\widetilde{X}_t, \widetilde{D}_t)$ in the mechanised causal graph. This allows us to be clearer about what information is available for each decision (the state), in particular that the agent does not observe the outcome of the modification, as might be construed from the diagram in Langlois and Everitt [40].

4.5. Zero agents

Our final example of our algorithm working as desired is one in which there are no agents at all, see Fig. 6. Here X causes Y and Y causes Z, but there is no decision or utility. Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$, in Fig. 6b. Algorithm 2 produces the correct game graph by identifying that there are no decisions as there are no mechanisms with incoming edges, and hence also no utilities. This then just recovers a causal Bayesian network graph.



(a) $\mathcal{G}_{\text{real}}$

(b) $\mathcal{C}_{\text{model}}$ restricted to mechanisms only

(c) $\mathcal{G}_{\text{model}}$

Fig. 7. Assistance Game (A.K.A. CIRL). 7a True game graph $\mathcal{G}_{\text{real}}$, where the yellow utility indicates both robot and human share the same utility. 7b Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$, shown here restricted to mechanisms only for readability – see Appendix D, Fig. D.11 for the full mechanised causal graph. 7c Algorithm 2 produces the an incorrect game graph in this case, because we violated Assumption 1, and gives that all decisions belong to the same agent.

4.6. Breaking assumptions

Compared to the other assumptions which are more benign, Assumption 1 rules out some examples that we might wish to consider. We now consider some examples which break it.

4.6.1. Multiple agents with a shared utility

First, in Fig. 7, we consider a causal game that has two agents with a shared utility, see Fig. 7a. This is a diagram that represents an Assistance Game, formerly known as Cooperative Inverse Reinforcement Learning [27]. There is a human which makes decisions H_1 and H_2, conditioned on information about their preference, encoded by θ, and a robot which makes decisions R_1 and R_2 based on observations of the human's decisions, but without direct observation of θ. All of the human and robot decisions affect the utility, U which is the same for both robot and human agents (drawn in yellow to signify that it's shared). This breaks Assumption 1 because the decision-utility graph only has one weakly connected component, and two agent graphs, whereas Assumption 1 requires the weakly connected components be the two agent graphs.

Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$ shown in 7b. We believe this mechanised causal graph representation of the system is correct. However, the problem arises when we apply Algorithm 2 on it. The result is the game graph in 7c which has all decisions belonging to the same agent. We hope that in future work we will be able to distinguish agents which share the same utility, through some modification to the colouring logic of Algorithm 2. One approach may be to use some condition involving sufficient recall [42] to distinguish between agents (a game graph has sufficient recall if for all agents the mechanism graph restricted to that agent's decision rules is acyclic).

4.6.2. Non-descendent utility

We now consider an example, Fig. 8a, that breaks Assumption 1 in another way. There are two agents which make decisions A and B with utilities U^A and U^B respectively. The red agent chooses A, which affects the utility that the blue agent receives U^B. The blue agent's choice affects the red agent's utility. For example, this might be used to model a situation in which the red agent can coerce blue into making decisions that increase red's utility by threatening to do something to decrease blue's utility otherwise. Note that the agent graph for A is disconnected (no directed path from A to U^A), so this example violates Assumption 1.

Algorithm 1 applied to $\mathcal{G}_{\text{real}}$ produces the mechanised causal graph $\mathcal{C}_{\text{model}}$, in Fig. 8b. We think this mechanised causal graph is an accurate representation of the system. From inspecting it, we can see that although U^A is not a descendent of A, it is a descendent of $\widetilde{A}$, via $\widetilde{A} \to \widetilde{B} \to B \to U^A$. That is, the red agent's decision rule can still have an effect on its utility, but Assumptions 2 and 3 rule out agents strategising using this path. Applying Algorithm 2 on $\mathcal{C}_{\text{model}}$ produces an incorrect game graph with A and U^A being incorrectly identified as chance nodes (Fig. 8c).

This example highlights several questions for future work: Which agents learn to influence their utility by means of their decision rule, thereby breaking our Assumptions 2 and 3? And how can Algorithm 2 be generalised to handle non-descendant utilities and agents utilising influence from their decision rule?

5. Discussion

This section provides a more detailed discussion of points only touched briefly upon earlier in the paper.

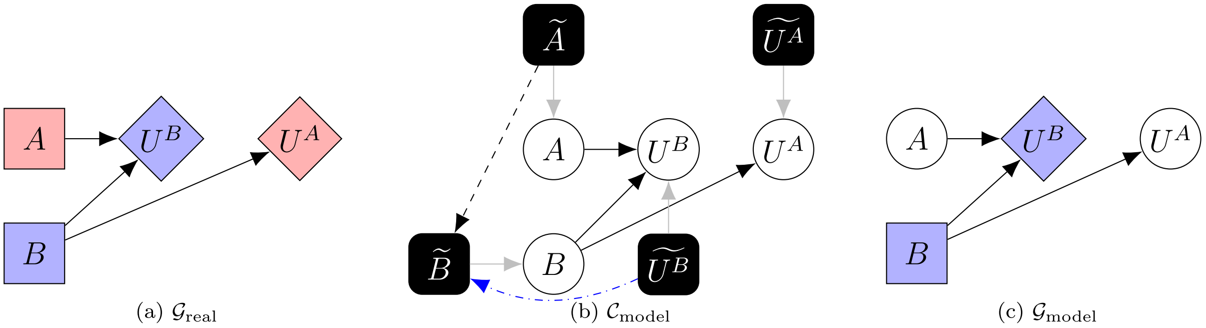


Fig. 8. Non-descendant utility. 8a True game graph $\mathcal{G}_{\text{real}}$. Note that the agent graph for A (Definition 4) is not connected, violating Assumption 1. 8b Algorithm 1 produces the mechanised causal graph $\mathcal{C}_{\text{model}}$. 8c Algorithm 2 produces an incorrect game graph in this case, because we violated Assumption 1, leading to A and U^A being incorrectly identified as chance, rather than as decision and utility variables respectively.

5.1. Choice of variables

Algorithm 1 only provides a way to determine the structure of a mechanised SCM and associated game graph from a given set of variables, but not how to choose them. This can be tricky in practice, and is not an automatic process. That said, we now offer some tips on choosing variables.

A few principles always apply. First, variables should represent aspects of the environment that we are concerned with, either as means of influence for an agent, or as inherently valuable aspects of the environment. The content selected by a content recommender system, and the preferences of a user, are good examples. Second, it should be fully clear both how to measure and how to intervene on a variable. Otherwise its causal relationship to other variables will be ill-defined. In our case, this requirement extends also to the mechanism of each variable. Third, a variable's domain should be exhaustive (cover all possible outcomes of that variable) and represent mutually exclusive events (no pair of outcomes can occur at the same time) [38]. Finally, variables should be *logically independent*: one variable taking on a value should never be mutually exclusive with another variable taking on a particular value [29].

Of particular importance is the level of coarse-graining in the choice of variables. Indeed, Hoel [33] argues that coarse-graining in causal graphs can explain the emergence of agents. We hope to explore marginalisation in the context of game graphs in future work, and present one example in Appendix B. The choice of coarse-graining may have an impact on whether agents are discovered. Technical work on marginalisation has been done in Bayesian networks [17,37], and in cyclic SCMs [6].

Further, Algorithm 1 expects these variables to already be labelled as object-level or mechanism. This can be subjective and is a choice the modeler must make, and different choices can lead to different outputs impacting on whether agents are discovered. However, a general principles to follow is that the mechanism variable should be relevant for parameterising the conditional distribution of the object-level variable given its object-level parents.

5.2. Relativism of variable types

The type of a variable – decision, utility or chance – is relative to which other variables are also included in the model. For example, consider the mouse from Fig. 1b that is optimising the variable U. As expected, U is labelled a utility node in Fig. 1b. However, if we had modelled the mouse example without U, i.e. with just D and X, then X would have been labelled a utility node instead. Similarly, if we modelled the mouse without the decision variable D, but with only the variables X and U, then X would be labelled as a decision. Thus, X can either be a decision, a utility, or a chance node, depending on which other variables are included in the model. See Appendix C for a similar example of this relativism. In a sense, the choice of variables represents a *frame* from which to view the system. And what is a decision or a utility node depends on the frame.

5.3. Do we discover agents or just decision and utility nodes?

Our Algorithms 1 and 2 identify decision and utility nodes in the frame $\boldsymbol{V}$. Is this enough to identify any agent reflected in the variables? We argue that it often is, at least if we interpret *discovering agents* in a weak sense, where it is enough to say that an agent is present, when an agent is using some of the variables in $\boldsymbol{V}$ to control others. In this sense, we do discover an agent whenever we identify a decision node belonging to this agent. But what if some variables describe parts of an agent, such as internal organs of an animal, or parameters or code for a machine learning agent, instead of the agent's outward facing decision? The mechanisms of these variables will typically have co-evolved with the rest of the organism or system, and will therefore be identified by as decisions by our algorithm. So even in this case, we do often discover the presence of an agent. Our algorithms really only fail to discover agents when they fail to discover decision nodes, which can happen when the frame $\boldsymbol{V}$ either doesn't contain variables with adaptive mechanisms, or when $\boldsymbol{V}$ lacks variables that the agents in $\boldsymbol{V}$ care to influence.

One might also interpret *discovering agents* in a stronger sense, asking to draw an exact boundary around the variables describing a particular agent. For example, in Fig. 4a, one might want to draw a boundary around A, Q, W, Y, and possibly R, as they can naturally be interpreted as components of a single actor-critic agent. While the coarse-graining in Fig. 4d points in a direction of how such an algorithm might be developed, we leave the details of this for future work.

5.4. *Goal misgeneralization*

Goal misgeneralisation [14,53] (or mesa-optimisation; [34]) occurs when an agent coherently pursues a goal different from the one its designers intended, when out of distribution. Di Langosco et al. [14] use the agency definition of Orseau et al. [45] to detect such phenomena. As future work, it would be interesting to see if further precision could be gained by employing our causal definition.

5.5. *Limitations*

We now discuss some limitations of our work, covering practicality, efficiency, and assumptions.

5.5.1. *Practicality of interventions*

The causal discovery methods we use require access to the distributions resulting from interventions on any subset of the variables. In principle, any such distribution can be obtained by doing the interventions, and then sampling the variables a sufficient number of times to determine their joint distribution (to arbitrary precision). In practice, this scheme faces multiple challenges.

Interventions are often impractical, as variables may represent things that are not practically changeable. For example, a typical mechanism variable may represent the transition dynamics between different states of the environment, which may be hard to change if the environment is part of the real, physical world (and the agent a robot or an animal). Further, in some cases mechanism interventions would need to have happened in the past. If we want to know whether a tree (together with its evolutionary creation process) would act differently in a world where sunlight was twice as strong, we need to change the sunlight condition before the evolution of the tree happened (in the past).

There are a few different ways of tackling those challenges. First, we can often do interventions in simulations instead of in reality. This is literally true for agents trained in simulation, such as most RL agents. Alternatively, if simulation is not possible, the graph can sometimes also be inferred from statistical analysis [46, ch. 2], unless there are too many confounders. For real-world agents, we can instead use our mental models, and imagine what the result of a particular intervention would be. While this may sound unreliable, in practice we've found that people often agree about the likely outcome of an intervention, as long as it's described to a sufficient level of detail. Indeed, a precise mental model can be described with an SCM.

5.5.2. *Efficiency*

The *Leave-one-out* algorithm we use in Algorithm 1 requires an exponential number of interventional distributions. Since we restrict to an acylic object-level subgraph, a more efficient standard (acyclic) causal discovery algorithm could be used to discover this subgraph (see, e.g., [15]). For the mechanism subgraph, a more efficient cyclic causal discovery algorithm could be used (e.g., [24]). There are usually tradeoffs between speed and assumptions required by these algorithms, however.

5.6. *Relationship to causality literature*

We now discuss some related literature in Causality. Other related work was discussed in Section 1.3. Pearl [46] lays the foundations for modern approaches to causality, with emphasis on graphical models, and in particular through the use of structural causal models (SCMs), which allow for treatment of both interventions and counterfactuals. Dawid [12] considers related approaches to causal modelling, including the use of influence diagrams to specify which variables can be intervened on. One model that's introduced is called a *parameter DAG*, which is similar to our mechanised SCM, in that each object-level variable has a *parameter* variable which parametrises the distribution of the object-level variable. However, whilst acknowledging there could be links between the parameter variables, they are not considered in that work. In contrast, our focus is less on using influence diagrams as a tool for causal modelling, and rather on modelling and discovering agents using causal methods. Further, we allow relationships between mechanism variables in our models, and elucidate their relation to decision, chance and utility variables in the influence diagram representation.

Halpern [28] gives an axiomatization of SCMs, generalizing to cases where the structural equations may not have a unique (or any) solution. However, in the case of non-unique (or non-existant) solutions, potential response variables are ill-defined, which White and Chalak [57] claim prevents the desired causal discourse. They instead propose the *settable systems* framework in which there are *settable* variables which have a role-indicator argument which determines whether a variable's value is a *response*, determined by its structural equation, or if its value is a *setting*, as determined by a hard intervention. Bongers et al. [6] give formalizations for statistical causal modelling using cyclic SCMs, proving certain properties present in the acyclic case don't hold in the cyclic case. In our work, we use mechanised SCMs that can have cycles between mechanism variables. Zero, one or multiple solutions reflect the multiple equilibria arising in some games. Our formalism for mechanised SCMs follows the cyclic SCM treatment of Bongers et al. [6].

Correa and Bareinboim [11] develop *sigma-calculus* for reasoning about the identification of the effects of soft interventions using observational, rather than experimental data. In our work, we assume access to experimental data, which makes the identification question trivial. Future work could relax this assumption to explore when agents can be discovered from observational data. Their *regime indicators* roughly correspond to our mechanism variables.

Our work draws on structure discovery in the causal discovery literature. See Glymour et al. [26] for a review, and Forré and Mooij [24] for an example of causal discovery of cyclic models. The usual focus in causal discovery is not to model agents, but rather to model some physical (agent-agnostic) system (modelling agents is usually done in the context of decision/game theory). Our work differs in that we use causal discovery in order to get a causal model representation of agents (a mechanised SCM), and can then translate that to the game-theoretic description in terms of game graphs with agents.

One of the most immediate applications of our results concerns the independent causal mechanisms (ICM) principle [50,47,51]. ICM states that,

1. Changing (intervening on) the causal mechanism M^X for $P(X \mid \mathbf{pa}^X)$ does not change any of the other mechanisms M^Y for $P(Y \mid \mathbf{pa}^Y)$, $X \neq Y$.
2. Knowledge of M^X does not provide knowledge of M^Y for any $X \neq Y$.

ICM argues that $P(X \mid \mathbf{pa}^X)$ typically describes fixed and modular causal mechanisms that do not respond to the mechanisms of other variables. The classic example is the distribution of atmospheric temperature T given its causes $\mathbf{pa}^T$ such as altitude. While the distribution $P(\mathbf{pa}^T)$ may vary between countries, $P(T \mid \mathbf{pa}^T)$ remains fixed as it describes a physical law relating altitude (and other causes) to atmospheric temperature. In recent years ICM has become the predominant inductive bias used in causal machine learning including causal and disentangled representations [3,41,49], causal discovery [35,36], semi-supervised learning [50], adversarial vulnerability [52], reinforcement learning [4], and has even played a role in major scientific discoveries such as discovering the first exoplanet with atmospheric water [23]. Our results provide a constraint on the applicability of the ICM principle; namely that

> $P(X \mid \mathbf{pa}^X)$ does not obey the ICM principle if it is an agent's decision rule, or is strategically relevant to some agent's decision rule, as determined by Algorithm 2.

Condition 1 in the ICM is true only if X is not strategically relevant for an agent, and condition 2 covers agents themselves, as their mechanisms are correlated with the mechanisms of strategically relevant variables. This limits the applicability of ICM (and methods based on ICM) to systems where the data generating process includes no agents. But there are cases where we expect the data generating process to involve agents, for example sociological data and data generated by reinforcement learning agents during training. However, our hope is that Algorithm 1 can be applied to identify mechanism edges that violate ICM, allowing ICM to be applied to the correct systems, and in doing so improve the performance of ICM-based methods.

6. Conclusion

We proposed the first formal causal definition of agents. Grounded in causal discovery, our key contribution is to formalise the idea that agents are systems that adapt their behaviour in response to changes in how their actions influence the world. Indeed, Algorithms 1 and 2 describe a precise experimental process that can, in principle and under some assumptions, be done to assess whether agents are present in a system. Our process is largely consistent with previous, informal characterisations of agents (e.g. [13,58,21,25]), but making it formal enables agents and their incentives to be identified empirically or from the system architecture. Our process improves upon an earlier formalisation by Orseau et al. [45], by better handling systems with a small number of actions and "accidentally optimal" systems (see Section 1.3 for details).

Causal modelling of AI systems is a tool of growing importance, and this paper grounds this area of work in causal discovery experiments. We have demonstrated the utility of our approach by improving the safety analysis of several AI systems (see Section 4). In consequence, this would also improve the reliability of methods building on such modelling, such as analyses of the safety and fairness of machine learning algorithms (see e.g. [2,16,19,20,40,10,48]).

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Zachary Kenton has patent pending to DeepMind Technologies Ltd.

Data availability

No data was used for the research described in the article.

Acknowledgement

We thank Laurent Orseau, Mary Phuong, James Fox, Lewis Hammond, Francis Rhys Ward, and Ryan Carey for comments and discussions.

Appendix A. Mathematical background

A.1. Notation

We use roman capital letters V for variables, lower case for their outcomes v. We use bold type to indicate vectors of variables, $\boldsymbol{V}$, and vectors of outcomes $\boldsymbol{v}$. Parent, children, ancestor and descendent variables are denoted $\mathbf{Pa}^V, \mathbf{Ch}^V, \mathbf{Anc}^V, \mathbf{Desc}^V$, respectively, with the family denoted by $\mathbf{Fa}^V = \mathbf{Pa}^V \cup \{V\}$. We use $\mathrm{dom}(V)$ and $\mathrm{dom}(\boldsymbol{V}) = \times_{V\in\boldsymbol{V}}\mathrm{dom}(V)$ to denote the set of possible outcomes of V and $\boldsymbol{V}$ respectively, which are assumed finite. Subscripts are reserved for denoting submodels and potential responses to an intervention.

A.2. Structural causal model

We begin with a standard definition of a structural causal model.

Definition A.5 *(Structural Causal Model (SCM), [46]).* A *structural causal model (SCM)* is given by the tuple $\mathcal{S} = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}}^{\boldsymbol{V}}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}}^{\boldsymbol{V}})\rangle$ where

- $\boldsymbol{V}$ is a set of endogenous variables.
- $\boldsymbol{\mathcal{E}}^{\boldsymbol{V}} = \{\mathcal{E}^V\}_{V\in\boldsymbol{V}}$ is a set of exogenous variables, one for each endogenous variable.
- $\mathcal{F} = \{V = f^V(\boldsymbol{V}, \mathcal{E}^V)\}_{V\in\boldsymbol{V}}$ is a set of structural equations, one for each endogenous variable.
- $\Pr(\boldsymbol{\mathcal{E}}^{\boldsymbol{V}})$ is a distribution over the exogenous variables.

This has an associated directed graph, called a causal graph (CG).

Definition A.6 *(Causal Graph (CG)).* For an SCM, $\mathcal{S} = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}}^{\boldsymbol{V}}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}}^{\boldsymbol{V}})\rangle$, a *causal graph* (CG) is the directed graph $\mathcal{C} = \langle \boldsymbol{V}, E\rangle$, where the set of directed edges, E, represent endogenous dependencies in the set of structural equations $\mathcal{F}$, so that $(V, W) \in E$ if and only if $W, V \in \boldsymbol{V}$ and $f^W(\boldsymbol{V}, \mathcal{E}^V)$ depends on the value of V (as such, all our causal graphs are faithful [46] by construction).

The subgraph of white nodes in Fig. 1c is an example of a CG.

In some parts of this work, we will consider acyclic (recursive) SCMs, in which the CG is acyclic. Other parts will consider cases in which there is a possibly cyclic (nonrecursive) SCM, in which the CG is cyclic. See Bongers et al. [6] for a foundational treatment of SCMs with cyclic CGs. They define a solution of an SCM as a set of exogenous and endogenous random variables, $\boldsymbol{\mathcal{E}}, \boldsymbol{V}$, for which the exogenous distribution matches that in the cyclic SCM, and for which the structural equations are satisfied. For a solution, $\boldsymbol{\mathcal{E}}, \boldsymbol{V}$, the distribution over the endogenous variables, $\Pr^{\boldsymbol{V}}$ is called the observational distribution associated to $\boldsymbol{V}$. In this cyclic case, there can be zero, one or many observational distributions, due to the existence of different solutions of the structural equations. In this work, we assume the existence of a unique solution, even in the case of a nonrecursive SCM. This unique solution then defines a joint distribution over endogenous variables [46]

$$\Pr^{[\mathcal{S}]}(\boldsymbol{V} = \boldsymbol{v}) = \sum_{\{\boldsymbol{\varepsilon} \mid \boldsymbol{V}(\boldsymbol{\varepsilon}) = \boldsymbol{v}\}} \Pr(\boldsymbol{\varepsilon}). \tag{A.1}$$

SCMs model causal interventions that set variables to particular outcomes, captured by the following definition of a submodel:

Definition A.7 *(SCM Submodel, [46]).* Let $\mathcal{S} = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}}^{\boldsymbol{V}}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}}^{\boldsymbol{V}})\rangle$ be an SCM, $\boldsymbol{Y} \subseteq \boldsymbol{V}$ be a set of endogenous variables, and $\boldsymbol{y} \in \mathrm{dom}(\boldsymbol{Y})$ a value for each variable in that subset. The submodel $\mathcal{S}_{\boldsymbol{y}}$ represents the effects of an *intervention* $\mathrm{do}(\boldsymbol{Y} = \boldsymbol{y})$ and is formally defined as the SCM $\mathcal{S}_{\boldsymbol{y}} = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}}^{\boldsymbol{V}}, \mathcal{F}_{\boldsymbol{y}}, \Pr(\boldsymbol{\mathcal{E}}^{\boldsymbol{V}})\rangle$ where $\mathcal{F}_{\boldsymbol{y}} = \{V = f^V(\mathbf{Pa}^V, \mathcal{E}^V)\}_{V\in\boldsymbol{V}\setminus\boldsymbol{Y}} \cup \{\boldsymbol{Y} = \boldsymbol{y}\}$. That is, the original functional relationships for $\boldsymbol{Y}$ are replaced with the constant functions $\boldsymbol{Y} = \boldsymbol{y}$.

We also assume the existence of a unique solution to the set of structural equations under all interventions, allowing us to define the potential response.

Definition A.8 *(Potential Response, [46]).* Let $\mathcal{S} = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}^V}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}^V}) \rangle$ be an SCM, and let $\boldsymbol{X}, \boldsymbol{Y} \subseteq \boldsymbol{V}$. The *potential response* of $\boldsymbol{X}$ to the intervention $\mathrm{do}(\boldsymbol{Y} = \boldsymbol{y})$, denoted $\boldsymbol{X_y}(\boldsymbol{\mathcal{E}})$ is the solution for $\boldsymbol{X}$ in the set of equations $\mathcal{F}_{\boldsymbol{y}}$, that is, $\boldsymbol{X_y}(\boldsymbol{\mathcal{E}}) = \boldsymbol{X}_{\mathcal{S}_{\boldsymbol{y}}}(\boldsymbol{\mathcal{E}})$, where $\mathcal{S}_{\boldsymbol{y}}$ is the submodel from intervention $\mathrm{do}(\boldsymbol{Y} = \boldsymbol{y})$.

A.3. Structural causal game

We now introduce a (structural) causal game, which draws on the SCM, emphasising the structural causal dependencies present.

Definition A.9 *(Structural Causal Game, [32,18]).* A (Markovian) (structural) causal game is a tuple

$$\widetilde{\mathcal{M}} = \langle N, \boldsymbol{V}, \boldsymbol{\mathcal{E}^V}, \{\mathbf{Pa}^D\}^{D \in \boldsymbol{D}}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}^V}) \rangle$$

where

- $N = \{1, \ldots, n\}$ is a set of agents
- $\boldsymbol{V} = \boldsymbol{D} \cup \boldsymbol{X} \cup \boldsymbol{U}$ is a set of endogenous variables, partitioned into decision, chance and utility variables respectively.
- $\boldsymbol{\mathcal{E}^V} = \{\mathcal{E}^V\}_{V \in \boldsymbol{V}}$ is a set of exogenous variables, one for each endogenous variable
- $\{\mathbf{Pa}^D\}^{D \in \boldsymbol{D}}$ is a set of information parents for each decision variable D, with $\mathbf{Pa}^D \subseteq \boldsymbol{V} \setminus D$
- $\mathcal{F} = \{V = f^V(\mathbf{Pa}^V, \mathcal{E}^V)\}_{V \in \boldsymbol{V} \setminus \boldsymbol{D}}$ is a set of structural equations for each non-decision endogenous variable, as specified by the functions $f^V : \mathrm{dom}(\mathbf{Pa}^V \cup \{\mathcal{E}^V\}) \mapsto \mathrm{dom}(V)$, where $\mathbf{Pa}^V \subseteq \boldsymbol{V} \setminus \{V\}$.
- $\Pr(\boldsymbol{\mathcal{E}^V})$ is a distribution over the exogenous variables such that the individual exogenous variables are mutually independent.

The causal game has an associated graph:

Definition A.10 *(Game Graph).* Let $\widetilde{\mathcal{M}} = \langle N, \boldsymbol{V}, \boldsymbol{\mathcal{E}^V}, \{\mathbf{Pa}^D\}^{D \in \boldsymbol{D}}, \mathcal{F}, \Pr(\boldsymbol{\mathcal{E}^V}) \rangle$ be a causal game. We define the *game graph* to be the structure $\mathcal{G} = (N, \boldsymbol{V} \cup \boldsymbol{\mathcal{E}^V}, E)$, where $N = \{1, \ldots, n\}$ is a set of agents and $(\boldsymbol{V} \cup \boldsymbol{\mathcal{E}^V}, E)$ is a DAG with:

- Four vertex types $\boldsymbol{V} \cup \boldsymbol{\mathcal{E}^V} = \boldsymbol{X} \cup \boldsymbol{U} \cup \boldsymbol{D} \cup \boldsymbol{\mathcal{E}^V}$: the first three types are endogenous nodes in white circles, coloured diamonds and squares respectively; the fourth type are exogenous nodes, $\boldsymbol{\mathcal{E}^V}$, in grey circles. The different colours of diamonds and squares correspond to different agents.
- Two types of edges:
  - dependence edges, $(V, W) \in E$ if and only if either $W \in \boldsymbol{V} \setminus \boldsymbol{D}$ and V is an argument to the structural function f^W, i.e. $V \in \mathbf{Pa}^W \cup \boldsymbol{\mathcal{E}}^W$; or $W = D \in \boldsymbol{D}$ and $V = \mathcal{E}_D$. These are denoted with solid edges.
  - information edges, $(V, D) \in E$ if and only if $V \in \mathbf{Pa}^D$ of the causal game. These are denoted with dashed edges.

One can also draw a simpler graph by omitting the exogenous variables and their outgoing edges from the game graph. Fig. 1b is an example of a game graph. We will only consider causal games for which the associated game graph is acyclic.

For each non-decision variable, the causal game specifies a distribution over it. For the decision variables, the causal game doesn't specify how it is distributed, only the information available at the time of the decision, as captured by $\mathbf{Pa}^D$. The agents get to select their behaviour at each of their decision nodes, as follows. Let $\mathcal{M}$ be a causal game. A *decision rule*, π^D, for a decision variable $D \in \boldsymbol{D}^i \subseteq \boldsymbol{D}$ is a (measurable) structural function $\pi^D : \mathrm{dom}(\mathbf{Pa}^D \cup \{\mathcal{E}^D\}) \mapsto \mathrm{dom}(D)$ where $\mathcal{E}^D$ is uniformly distributed over the $[0, 1]$ interval.[10] A *partial policy profile*, $\boldsymbol{\pi}^{\boldsymbol{D}'}$ is a set of decision rules π^D for each $D \in \boldsymbol{D}' \subseteq \boldsymbol{D}$. A *policy* refers to $\boldsymbol{\pi}^a$, the set of decision rules for all of agent a's decisions. A *policy profile*, $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^n)$ assigns a decision rule to every agent.

For a causal game $\mathcal{M}$, we can combine its set of structural equations $\mathcal{F} = \{V = f^V(\mathbf{Pa}^V, \mathcal{E}^V)\}_{V \in \boldsymbol{V} \setminus \boldsymbol{D}}$ with a policy profile $\boldsymbol{\pi}$ to obtain a *Policy-game SCM*, $\mathcal{M}(\boldsymbol{\pi}) = \langle \boldsymbol{V}, \boldsymbol{\mathcal{E}^V}, \mathcal{F}^{\boldsymbol{\pi}}, \Pr(\boldsymbol{\mathcal{E}^V}) \rangle$ with the set of structural equations $\mathcal{F}^{\boldsymbol{\pi}} = \mathcal{F} \cup \{D = \pi^D(\mathbf{Pa}^D, \mathcal{E}^D)\}_{D \in \boldsymbol{D}}$. Note that there is a well-defined endogenous distribution, $\Pr^{[\mathcal{M}(\boldsymbol{\pi})]}$, as the policy-game SCM is acyclic, due to the game graph being a DAG.

Each agent's expected utility in policy profile $\boldsymbol{\pi}$ is given by

$$EU^a(\boldsymbol{\pi}) = \sum_{\{u^a \in \mathrm{dom}(U^a)\}} \Pr^{[\mathcal{M}(\boldsymbol{\pi})]}(U^a = u^a) \cdot u^a. \tag{A.2}$$

[10] For settings where we are interested in arbitrary counterfactual queries, a more complex form of $\mathcal{E}^D$ has advantages [32].

Definition A.11 *(Optimality and Best Response, [39]).* Let $\boldsymbol{k} \subseteq \boldsymbol{D}^a$ and let $\boldsymbol{\pi}$ be a policy profile. We say that partial policy profile $\hat{\boldsymbol{\pi}}^{\boldsymbol{k}}$ is *optimal for policy profile* $\boldsymbol{\pi} = (\boldsymbol{\pi}^{-\boldsymbol{k}}, \hat{\boldsymbol{\pi}}^{\boldsymbol{k}})$ if in the induced causal game $\mathcal{M}(\boldsymbol{\pi}^{-\boldsymbol{k}})$, where the only remaining decisions are those in $\boldsymbol{k}$, the decision rule $\hat{\boldsymbol{\pi}}^{\boldsymbol{k}}$ is optimal, i.e. for all partial policy profiles $\boldsymbol{\pi}^{\boldsymbol{k}}$

$$EU^a((\boldsymbol{\pi}^{-\boldsymbol{k}}, \hat{\boldsymbol{\pi}}^{\boldsymbol{k}})) \geq EU^a((\boldsymbol{\pi}^{-\boldsymbol{k}}, \boldsymbol{\pi}^{\boldsymbol{k}})). \tag{A.3}$$

Agent a's decision rule $\boldsymbol{\pi}^a$ is a *best response* to the partial policy profile $\boldsymbol{\pi}^{-a}$ assigning strategies to the decisions of all other agents if for all strategies $\boldsymbol{\pi}^a$

$$EU^a((\boldsymbol{\pi}^{-a}, \hat{\boldsymbol{\pi}}^a)) \geq EU^a((\boldsymbol{\pi}^{-a}, \boldsymbol{\pi}^a)). \tag{A.4}$$

In the game-theoretic setting with multiple agents, we typically consider rational behaviour to be represented by a *Nash Equilibrium*:

Definition A.12 *(Nash Equilibrium, [39]).* A policy profile $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^n)$ is a *Nash Equilibrium* if for all agents a, $\boldsymbol{\pi}^a$ is a best response to $\boldsymbol{\pi}^{-a}$.

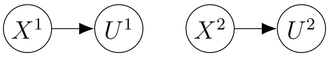
In this paper, we consider the refined concept of subgame perfect equilibrium (SPE), as follows

Definition A.13 *(Subgame Perfect Equilibrium, [31,32]).* A policy profile $\boldsymbol{\pi} = (\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^n)$ is a *Subgame Perfect Equilibrium* if for all subgames, $\boldsymbol{\pi}$ is a Nash equilibrium.

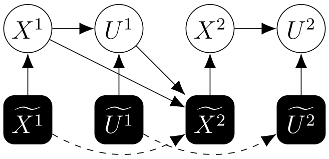
Informally, in any subgame, the rational response is independent of variables outside of the subgame. See Hammond et al. [32,31] for the formal definition of a subgame in a causal game.

Appendix B. Getting a mechanised SCM by marginalisation and merging

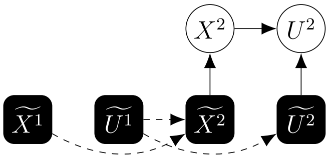
A bandit algorithm repeatedly chooses an arm X and receives a reward U. We can represent two iterations by using indices.



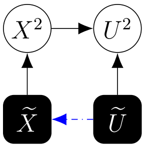
If we include the mechanisms in the graph, we can model the fact that the policy at time 2, i.e. $\widetilde{X}^2$, depends on the arm and outcome at time 1:



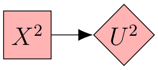
To arrive at the final mechanism graph, we first marginalise X^1 and U^1. The path from $\widetilde{U}^1$ to $\widetilde{X}^2$ previously mediated by U^1 now becomes a direct edge.



Finally, we merge $\widetilde{X}^1$ with $\widetilde{X}^2$ and $\widetilde{U}^1$ with $\widetilde{U}^2$, with the understanding that observing the merged node $\widetilde{X}$ corresponds to observing $\widetilde{X}^2$, while intervening on $\widetilde{X}$ means setting both $\widetilde{X}^1$ and $\widetilde{X}^2$. This yields the following mechanised causal graph (note the terminal edge due to U^2 not having any children)

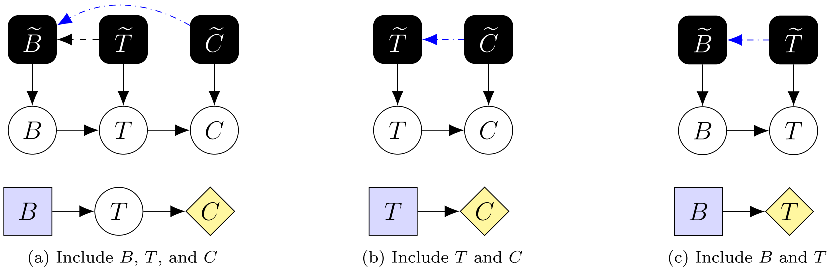


Applying Algorithm 2 yields the expected game graph:



Appendix C. Example of relativism of variable types

This example illustrates the discussion of Sec. 5.2 with an example of the relativism of variable types. Whether a variable gets classified as a decision, chance, or utility node by Algorithm 2 depends on which other nodes are included in the graph. To see this, consider the graph in Fig. C.9 in which a blueprint for a thermometer, B, influences the constructed thermometer, T, and thereby whether the reading is correct or not, C.



(a) Include B, T, and C (b) Include T and C (c) Include B and T

Fig. C.9. What is a decision or a utility node depends on what other variables are included. Here the variables represent a blueprint for a thermometer (B), the constructed thermometer (T), and thereby whether the reading is correct or not (C).

Considering first Fig. C.9a where a first modeler has included all three variables. We find that the designer will produce a different blueprint if they are aware that blueprints are interpreted according to a different convention (i.e., if $\widetilde{T}$ changes), or if temperature was measured at a different scale (a change to $\widetilde{C}$). Accordingly, Algorithm 2 labels B a decision, and C a utility. This makes sense in this context: the designer chooses a blueprint to ensure that the thermometer gives a correct reading.

A second modeler may not care about the blueprint, and only wonder about the relationship between the produced thermometer T and the correctness of the reading C. See Fig. C.9b. They will find that if temperature was measured at a different scale, a slightly different thermometer would have been produced, i.e. $\widetilde{C}$ now influences $\widetilde{T}$ rather than $\widetilde{B}$ (as in Fig. C.9a). This is not a contradiction, as $\widetilde{T}$ is a different object in Fig. C.9a and C.9b. In Fig. C.9a, $\widetilde{T}$ represents the relationship between B and T, while in Fig. C.9b, $\widetilde{T}$ represents the marginal distribution of T. As a consequence, Algorithm 2 will label T as a decision optimising C. This is not unreasonable: a decision was made to produce a particular kind of thermometer with the aim of getting correct temperature readings.

A third modeler may not bother to represent the correctness of the readings explicitly, and only consider the blueprint and the produced thermometer, see Fig. C.9c. They will find that the blueprint is optimised to obtain a particular kind of thermometer. Again, this is not unreasonable, as in this context we may well speak of the designer deciding on a blueprint that will produce the right kind of thermometer.

Appendix D. Supplementary figures

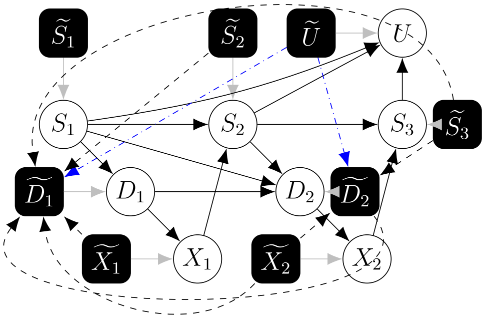


Fig. D.10. Full mechanised causal graph for MAMDP example in Fig. 5.

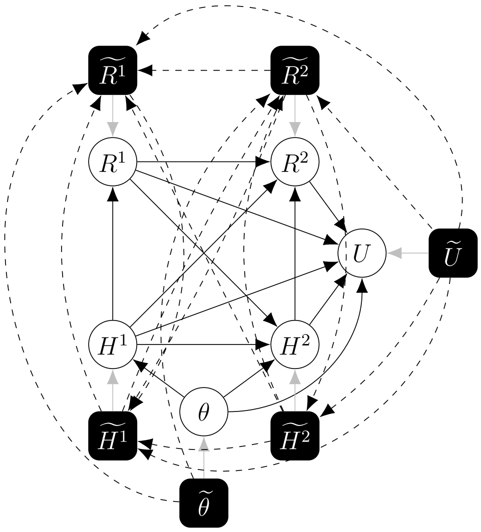


Fig. D.11. Full mechanised causal graph for Assistance Game example in Fig. 7.

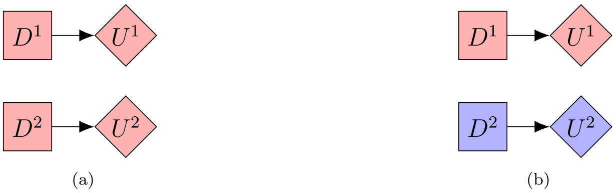


Fig. D.12. Fig. D.12a violates Assumption 1, whilst Fig. D.12b conforms to it.

References

[1] W.R. Ashby, An Introduction to Cybernetics, Chapman and Hall, 1956.

[2] C. Ashurst, R. Carey, S. Chiappa, T. Everitt, Why fair labels can yield unfair predictions: graphical conditions for introduced unfairness, in: AAAI, 2022.

[3] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[4] Y. Bengio, T. Deleu, N. Rahaman, R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, C. Pal, A meta-transfer objective for learning to disentangle causal mechanisms, arXiv preprint, arXiv:1901.10912, 2019.

[5] Y. Benkler, R. Faris, H. Roberts, Network Propaganda: Manipulation, Disinformation, and Radicalization in American Politics, Oxford University Press, 2018.

[6] S. Bongers, P. Forré, J. Peters, J.M. Mooij, Foundations of structural causal models with cycles and latent variables, Ann. Stat. 49 (2021).

[7] N. Bostrom, Superintelligence: Paths, Dangers, Strategies, Oxford University Press, 2014.

[8] M.D. Carroll, A. Dragan, S. Russell, D. Hadfield-Menell, Estimating and penalizing induced preference shifts in recommender systems, in: International Conference on Machine Learning, PMLR, 2022, pp. 2686–2708.

[9] F. Cavazzoni, A. Fiorini, G. Veronese, How do we assess how agentic we are? A literature review of existing instruments to evaluate and measure individuals' agency, Soc. Indic. Res. 159 (2022) 1125–1153, https://doi.org/10.1007/s11205-021-02791-8.

[10] M.K. Cohen, B. Vellambi, M. Hutter, Intelligence and unambitiousness using algorithmic information theory, IEEE J. Sel. Areas Inf. Theory 2 (2021) 678–690.

[11] J. Correa, E. Bareinboim, A calculus for stochastic interventions: causal effect identification and surrogate experiments, Proc. AAAI Conf. Artif. Intell. 34 (2020) 10093–10100.

[12] A.P. Dawid, Influence diagrams for causal modelling and inference, Int. Stat. Rev. 70 (2002) 161–189.

[13] D.C. Dennett, The Intentional Stance, MIT Press, 1987.

[14] L.L. Di Langosco, J. Koch, L.D. Sharkey, J. Pfau, D. Krueger, Goal misgeneralization in deep reinforcement learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 12004–12019.

[15] F. Eberhardt, C. Glymour, R. Scheines, On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables, in: UAI, 2005.

[16] C. Evans, A. Kasirzadeh, User tampering in reinforcement learning recommender systems, in: FAccTRec Workshop on Responsible Recommendation, 2021.

[17] R.J. Evans, Graphs for margins of Bayesian networks, Scand. J. Stat. 43 (2016) 625–648.

[18] T. Everitt, R. Carey, E. Langlois, P.A. Ortega, S. Legg, Agent incentives: a causal perspective, in: AAAI, 2021.

[19] T. Everitt, M. Hutter, R. Kumar, V. Krakovna, Reward tampering problems and solutions in reinforcement learning: a causal influence diagram perspective, Synthese 198 (2021) 6435–6467.

[20] S. Farquhar, R. Carey, T. Everitt, Path-specific objectives for safer agent incentives, in: AAAI, 2022.

[21] A. Flint, The ground of optimization, https://www.alignmentforum.org/posts/znfkdCoHMANwqc2WE/the-ground-of-optimization-1, 2020.

[22] H. von Foerster, M. Mead, H. Teuber (Eds.), Cybernetics: Circular Causal and Feedback Mechanisms in Biological and Social Systems. Transactions of the Seventh Conference, 1951, Josiah Macy, Jr. Foundation.

[23] D. Foreman-Mackey, B.T. Montet, D.W. Hogg, T.D. Morton, D. Wang, B. Schölkopf, A systematic search for transiting planets in the k2 data, Astrophys. J. 806 (2015) 215.
[24] P. Forré, J.M. Mooij, Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders, in: UAI, 2018.
[25] S. Garrabrant, Saving time, https://www.alignmentforum.org/posts/gEKHX8WKrXGM4roRC/saving-time, 2021.
[26] C. Glymour, K. Zhang, P. Spirtes, Review of causal discovery methods based on graphical models, Front. Genet. 10 (2019) 524.
[27] D. Hadfield-Menell, S.J. Russell, P. Abbeel, A. Dragan, Cooperative inverse reinforcement learning, Adv. Neural Inf. Process. Syst. 29 (2016) 3909–3917.
[28] J.Y. Halpern, Axiomatizing causal reasoning, J. Artif. Intell. Res. 12 (2000) 317–337.
[29] J.Y. Halpern, C. Hitchcock, Actual causation and the art of modeling, in: Causality, Probability, and Heuristics: A Tribute to Judea Pearl, College Publications, 2010, pp. 383–406, arXiv:1106.2652.
[30] J.Y. Halpern, M. Kleiman-Weiner, Towards formal definitions of blameworthiness, intention, and moral responsibility, in: AAAI, 2018.
[31] L. Hammond, J. Fox, T. Everitt, A. Abate, M. Wooldridge, Equilibrium refinements for multi-agent influence diagrams: theory and practice, in, in: AAAI, 2021.
[32] L. Hammond, J. Fox, T. Everitt, R. Carey, A. Abate, M. Wooldridge, Reasoning about causality in games, Artif. Intell. 320 (2023).
[33] E.P. Hoel, Agent Above, Atom Below: How Agents Causally Emerge from Their Underlying Microphysics, Springer International Publishing, Cham, 2018, pp. 63–76.
[34] E. Hubinger, C. van Merwijk, V. Mikulik, J. Skalse, S. Garrabrant, Risks from learned optimization in advanced machine learning systems, arXiv preprint, arXiv:1906.01820, 2019.
[35] D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, B. Schölkopf, Information-geometric approach to inferring causal directions, Artif. Intell. 182 (2012) 1–31.
[36] D. Janzing, B. Schölkopf, Causal inference using the algorithmic Markov condition, IEEE Trans. Inf. Theory 56 (2010) 5168–5194.
[37] D. Kinney, D. Watson, Causal feature learning for utility-maximizing agents, in: International Conference on Probabilistic Graphical Models, PMLR, 2020, pp. 257–268.
[38] U.B. Kjaerulff, A.L. Madsen, Bayesian Networks and Influence Diagrams, Springer Science+Business Media 200, 2008, p. 114.
[39] D. Koller, B. Milch, Multi-agent influence diagrams for representing and solving games, Games Econ. Behav. 45 (2003) 181–221.
[40] E. Langlois, T. Everitt, How RL agents behave when their actions are modified, in: AAAI, 2021.
[41] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, in: ICML, PMLR, 2019, pp. 4114–4124.
[42] B. Milch, D. Koller, Ignorable Information in Multi-Agent Scenarios, 2008.
[43] R. Ngo, AGI safety from first principles: goals and agency, https://www.alignmentforum.org/posts/bz5GdmCWj8o48726N/agi-safety-from-first-principles-goals-and-agency, 2020.
[44] S.M. Omohundro, The basic AI drives, in: Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference, IOS Press, NLD, 2008, pp. 483–492.
[45] L. Orseau, S.M. McGill, S. Legg, Agents and devices: a relative definition of agency, arXiv preprint, arXiv:1805.12387, 2018.
[46] J. Pearl, Causality, Cambridge University Press, 2009.
[47] J. Peters, D. Janzing, B. Schölkopf, Elements of Causal Inference: Foundations and Learning Algorithms, The MIT Press, 2017.
[48] J.G. Richens, R. Beard, D.H. Thompson, Counterfactual harm, arXiv preprint, arXiv:2204.12993, 2022.
[49] B. Schölkopf, Causality for machine learning, in: Probabilistic and Causal Inference: The Works of Judea Pearl, 2022, pp. 765–804.
[50] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, J. Mooij, On causal and anticausal learning, arXiv preprint, arXiv:1206.6471, 2012.
[51] B. Schölkopf, F. Locatello, S. Bauer, N.R. Ke, N. Kalchbrenner, A. Goyal, Y. Bengio, Toward causal representation learning, Proc. IEEE 109 (2021) 612–634.
[52] L. Schott, J. Rauber, M. Bethge, W. Brendel, Towards the first adversarially robust neural network model on MNIST, arXiv preprint, arXiv:1805.09190, 2018.
[53] R. Shah, V. Varma, R. Kumar, M. Phuong, V. Krakovna, J. Uesato, Z. Kenton, Goal misgeneralization: why correct specifications aren't enough for correct goals, arXiv preprint, arXiv:2210.01790, 2022.
[54] A. Shimi, M. Campolo, J. Collman, Literature review on goal-directedness, https://www.alignmentforum.org/posts/cfXwr6NC9AqZ9kr8g/literature-review-on-goal-directedness, 2021.
[55] J. Stray, I. Vendrov, J. Nixon, S. Adler, D. Hadfield-Menell, What are you optimizing for? Aligning recommender systems with human values, arXiv preprint, arXiv:2107.10939, 2021.
[56] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
[57] H. White, K. Chalak, Settable systems: an extension of pearl's causal model with optimization, equilibrium, and learning, J. Mach. Learn. Res. 10 (2009).
[58] N. Wiener, Cybernetics: Or Control and Communication in the Animal and the Machine, 2nd ed., MIT Press, 1961.
[59] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowl. Eng. Rev. 10 (1995) 115–152.
[60] E. Yudkowsky, et al., Artificial Intelligence as a Positive and Negative Factor in Global Risk, Global Catastrophic Risks, vol. 1, 2008, p. 184.