A Unifying Framework for Causal Explanation of Sequential Decision Making

Samer B. Nashed\textsuperscript{1}, Saaduddin Mahmud\textsuperscript{1}, Claudia V. Goldman\textsuperscript{2}, Shlomo Zilberstein\textsuperscript{1}

\textsuperscript{1}University of Massachusetts Amherst \hspace{1em} \textsuperscript{2}General Motors

\{snashed, smahmud, shlomo\}@cs.umass.edu, claudia.goldman@gm.com

Abstract

We present a novel framework for causal explanations of stochastic, sequential decision-making systems. Building on the well-studied structural causal model paradigm for causal reasoning, we show how to identify semantically distinct types of explanations for agent actions using a single unified approach. We provide results on the generality of this framework, run time bounds, and offer several approximate techniques. Finally, we discuss several qualitative scenarios that illustrate the framework’s flexibility and efficacy.

1 Introduction

As autonomous decision making becomes ubiquitous, researchers agree that developing trust is required for adoption and proficient use of AI systems by both experts and non-experts (Linegang et al. 2006; Stubbs, Hinds, and Wettergreen 2007; Zhang, Liao, and Bellamy 2020), and it is widely accepted that autonomous agents that can explain their decisions help promote trust (Chen et al. 2018; Hayes and Shah 2017; Mercado et al. 2016). However, there are many challenges in generating such explanations.

Consider, for example, an autonomous vehicle (AV) stopping behind a truck for a long duration. The passenger may wonder whether the AV is waiting for the truck to move, waiting for an opportunity to pass the truck, or perhaps dealing with some technical problem. Generating suitable explanations of such a system is hard due to the complexity of planning, which may involve large state spaces, stochastic actions, imperfect observations, and complicated objectives. Furthermore, useful explanations must somehow reduce the internal reasoning process to a form understandable by a user who likely does not know all of the algorithmic details.

Another challenging aspect is the heterogeneity of possible operational contexts and the interaction with different types of users with different expectations. For example, in the above AV scenario, the explanation to a passive passenger would differ from an explanation to a driver who evaluates whether to intervene and take control of the vehicle. Moreover, different planning, learning, and decision-making algorithms may not afford the same mechanisms for explanation due to fundamental differences in available information. Hence, it is important to develop unifying methods that can produce explanations for a wide range of sequential decision-making techniques and deployment scenarios.

Debate on the definition, taxonomy, and purpose of explanations has been well-represented in the cognitive science, psychology, and philosophy literature for a long time. While still active, there are several insights for which there is broad consensus (Miller 2019; Mittelstadt, Russell, and Wachter 2019), and we use this knowledge to motivate our approach. Scholars studying explanations mostly agree that requests for explanations are often motivated by a mismatch between the mental model of the requester and a logical conclusion based on an observation (Heider 1958; Hesslow 1988; Hilton and Slugoski 1986; Hilton 1996; Lombozro 2006; Williams, Lombozro, and Rehder 2013), which creates a form of generalized model reconciliation problem (Chakraborti et al. 2017). Researchers also agree that explanations often require counterfactual analysis (Mackie 1980; Lipton 1990; Hilton 1990; Lombozro 2012), which in turn requires causal determination (Woodward 2005; Salmon 2006; Lombozro 2010). There are several computational paradigms for causal analysis, including those based on conditional logic (Lewis 1974; Giordano and Schwind 2004), and statistics (Freedman 2007). Among the most well-studied paradigms is the structural causal model (SCM) (Halpern and Pearl 2000).

The primary contribution of this paper is a framework, based on SCMs, for applying causal analysis to sequential decision-making agents that unifies previous work and provides a well-founded mechanism for different types of explanations. Here, types of explanations refer to different sets of variables in the decision-making model, including the state factors and the reward, transition, and value functions. We show how to create a SCM representing the computations needed to derive a policy for a Markov decision process (MDP) and apply several forms of causal inference to identify variables that cause certain aspects of agent behavior. Once identified, those variables can be used to generate accurate, concise explanations.

We also provide theoretical results regarding the domain of problems for which this method is exact, and some preliminary run time bounds for the algorithms presented. Furthermore, we offer several approximate techniques for large, expensive problems, and problems for which the topology of the underlying causal graph prevents exact inference. Finally, we discuss several qualitative scenarios that illustrate how our approach not only produces sensible causes of agent behavior, but also uses a single framework to identify causal variables from a variety of MDP components.
2 Related Work

Automatically generating explanations is a growing field of research that can be divided roughly into two main categories. The first studies different methods for explaining decisions of black-box machine learning algorithms (Moithilal, Sharma, and Tan 2020; Lucic, Haned, and de Rijke 2020; Karimi, Schölkopf, and Valera 2021). These works often use the terms explainable or interpretable machine learning (XML). The second focuses primarily on explaining the outputs of planning algorithms, or modifying planning algorithms so that they produce plans that are inherently more explainable. These works typically use the term explainable planning (XAIP). A large portion of XAIP research has been devoted to deterministic planners, or analyzing plans after they have been executed (Fox, Long, and Magazzeni 2017; Chakraborti et al. 2019; Chakraborti, Sreedharan, and Kambhampati 2020). However, many applications operate in stochastic domains and require explanations in real time.

Research on explanations of stochastic planners, such as MDPs, is relatively sparse, but there are several notable existing efforts. Elizalde et al. (2009) identify important state factors by looking at how the value function would change were they to perturb that state factor’s value, and Khan et al. (2009) present a technique to explain policies for factored MDPs by analyzing the expected occupancy frequency of states with extreme reward values. Similarly, Jooza-paitis et al. (2019) analyze how extreme reward values impact action selection in decomposed-reward RL agents, and Bertram and Peng (2018) look at reward sources in deterministic MDPs. Wang et al. (2016) try to explain policies of partially observable MDPs by communicating the relative likelihoods of different events or levels of belief. However, research clearly indicates that humans are not good at using this kind of numerical information (Miller 2019).

While these approaches have limited scope in the explanations they provide, they are also computationally cheap and easy to implement. More complex methods have been developed that attempt explanation via summarization. Pouget et al. (2020) identify key state-action pairs via spectrum-based fault localization, and Russell et al. (2019) use decision trees to approximate a given policy and analyze the decision nodes to determine which state factors are most influential for immediate reward. Panigutti et al. (2020) used similar methods to explain classifiers.

Recently, the use of structural causal models for explaining MDPs has been proposed by Madumal et al. (2020), who used SCMs to encode the influence of particular actions available to the agent. This approach was used in a model-free, reinforcement learning setting to learn the structural equations as multivariate regression models during training. However, it requires several strong assumptions including the prior availability of a graph representing causal direction between variables, discrete actions, and the existence of sink states. In contrast to the above methods, our proposed unifying framework allows causal analysis of all components of MDPs using a single set of algorithms. Moreover, our framework is theoretically well-justified as it rests on a concrete theory of causality, and can be easily extended for cases where approximate reasoning is required.

3 Background

Here, we provide an overview of concepts and notation for structural causal models and Markov decision processes.

3.1 Structural Causal Models

The structural causal model approach (Halpern and Pearl 2000; Halpern and Pearl 2005) partitions variables into two sets. The first contains variables considered possible causes of some event, and the second contains variables that, while possibly causally relevant, are assumed to be fixed for a given analysis. The decision of which variables to put into each class is a design choice that we revisit later.

Formally, a structural causal model (SCM) is a signature

$$ S = \langle \mathcal{U}, \mathcal{V}, \mathcal{M} \rangle $$

(describing a causal reasoning problem. The set \( \mathcal{U} \), called the context, is a set of exogenous variables that describe some condition of the world. The set \( \mathcal{V} \) is a set of endogenous variables. In our case, these variables are internal to either the online or offline reasoning process of the agent. All variables in the world are either elements of \( \mathcal{U} \) or elements of \( \mathcal{V} \), and \( \mathcal{U} \cap \mathcal{V} = \emptyset \). The final component, \( \mathcal{M} \), is a set of equations which completely describe the causal effects of variables in \( \mathcal{U} \) and \( \mathcal{V} \) on other variables in \( \mathcal{V} \).

A causal graph is a DAG where nodes are variables and edges denote cause-effect relations. A layered causal graph (Fig. 1) is defined given an event \( \phi \), for which we want to determine causes, and a set of variables \( X \subseteq \mathcal{V} \), which we would like to evaluate as causal or not. A layered causal graph is a directed graph whose nodes are partitioned into non-intersecting layers \( (S_1^k, \ldots, S_0^k) \), where for every edge \( A \to B \) in the graph there exists some \( i \in \{1, \ldots, k\} \) such that \( A \in S_i^k \) and \( B \in S_i^{k-1} \). Further, \( X \subseteq S_k^k \), and \( \phi \in S_0^0 \).

3.2 Markov Decision Processes

A Markov decision process (MDP) is a model for reasoning in fully observable, stochastic environments (Bellman 1952), described as a tuple \( \langle S, A, T, R, d \rangle \). \( S \) is a finite set of states, where \( s \in S \) may be expressed in terms of a set of state factors, \( \{f_1, f_2, \ldots, f_N\} \), such that \( s \) indexes a unique assignment of variables to the factors \( f \); \( A \) is a finite set of actions; \( T : S \times A \times S \to [0, 1] \) represents the probability of reaching a state \( s' \in S \) after performing an action \( a \in A \) in a state \( s \in S \); \( R : S \times A \times S \to \mathbb{R} \) represents the expected immediate reward of reaching a state \( s' \in S \) after performing an action \( a \in A \) in a state \( s \in S \); and \( d : S \to [0, 1] \) represents the probability of starting in a state \( s \in S \). A solution to an MDP is a policy \( \pi : S \to A \) indicating that an action \( \pi(s) \in A \) should be performed in a state \( s \in S \). A policy \( \pi \) induces a value function \( V^\pi : S \to \mathbb{R} \) representing the expected discounted cumulative reward \( V^\pi(s) \in \mathbb{R} \) for each state \( s \in S \) given a discount factor \( 0 \leq \gamma < 1 \). An optimal policy \( \pi^* \) maximizes the expected discounted cumulative reward for every state \( s \in S \) by satisfying the Bellman optimality equation

$$ V^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s') [R(s, a, s') + \gamma V^*(s')] \, . $$

4 Structural Causal Models for MDPs

At a high level, our method constructs a causal model of the computation used to determine the policy of the MDP and
then uses this model to determine causes for agent actions, which can later be used as components of an explanation. This process follows four steps. First, a causal graph is generated from the relevant MDP components. Second, the resulting graph is converted into a layered causal graph. Third, the layered graph is pruned to remove any nodes and edges determined irrelevant. Finally, a recursive algorithm identifies a set of causal variables in the pruned graph. The main benefit of this approach is that it provides a principled, general framework for causal inference on MDPs that simultaneously supports several types of explanations—both novel methods as well as some resembling those proposed earlier.

In our layered graphical representation of MDPs we call layer \( S^0 \) the policy layer. This layer contains Boolean variables of the form \( \pi_{sa} = [\pi(s) = a] \) that collectively represent the policy of the MDP agent. For an MDP with \(|A|\) actions and \(|S|\) states, this layer has \(|S||A|\) variables, one for each state-action combination. Depending on the type of analysis, layers \( S^1 \) through \( S^k \) will represent different reasoning processes, which correspond to different definitions of \( V \). However, there is an overall similarity in that, given a layered causal graph and the values for all nodes without incoming edges, we can derive the action that the agent would take by passing variable values along edges and combining them to compute new variables in subsequent layers of the graph until we reach layer \( S^0 \), at which point the selected action is clear. We will first examine this process in more detail using layered MDPs as an example (§4), and then discuss general MDPs and approximate methods (§5).

### 4.1 Causal Models for Layered MDPs

We begin with the special case of a layered MDP for which our methods are exact (up to discretization). Layered MDPs are MDPs whose planning graphs are also layered graphs. In particular they contain two important subclasses: tree-structured MDPs and all finite-horizon MDPs. To use causal reasoning, we need to create a SCM representation of the MDP. Although it is possible to create a single, monolithic causal graph, this is not helpful since it does not afford any additional types of inference, along with being much less computationally efficient and requiring substantial bookkeeping to maintain the layered property. Thus, we analyze two causal models that together can answer causal queries about many parts of MDPs considered in previous literature.

The first model is for queries about the causality of state factors. Here, we let \( \mathcal{U} = \{ \emptyset \} \), and

\[
\mathcal{V} = \pi_{sa} \cup s(f) \cup f_i \quad \forall s, a, i,
\]

where \( \pi_{sa} \) denotes whether or not action \( a \) is taken in state \( s \) under policy \( \pi \), \( s(f) \) is a Boolean identity function representing whether the agent is in state \( s \) while observing state factors \( f \), and \( f_i \) denotes the \( i \)th state factor. Finally, we let \( \mathcal{M} \) be the set of equations mapping sets of state factors to states and states to actions.

\[
\mathcal{M}_F := f_i = f_i^t, \quad \text{for } i = 1, \ldots, n.
\]

Here \( f_i^t \) is the value of state factor \( i \) at time \( t \). A given set of state factors \( \{f_1, \ldots, f_n\} \in F \) determines the state \( s \in S \).

\[
\mathcal{M}_S := (S = s_i) = (f_1 = f_1^t) \land \ldots \land (f_n = f_n^t), \quad \forall s \in S.
\]

Lastly, we have the set of equations that represent action selection given the state.

\[
\mathcal{M}_A := (\pi(s) = a) = \pi_{sa} \land s \quad \forall s \in S, a \in A.
\]

Thus we define \( \mathcal{M} := \mathcal{M}_F \cup \mathcal{M}_S \cup \mathcal{M}_A \).

We note a few features of this graph, shown in Fig. 2, for comparison with subsequent graphs. First, this graph represents a fixed policy. We cannot, for example, change state factors to produce a different policy. We can however understand how different state factors affect action selection. Second, this graph has a fixed number of layers. As long as the MDP has a finite number of states and state factors, the graph will always have 3 layers. Last, unlike other causal graphs we will generate, this graph permits exact inference regardless of the topology of the underlying MDP.

The second causal model we present is used to analyze how reward functions, transition functions, and value functions are causally linked to action selection. Here, we let \( \mathcal{U} = \{ \gamma \} \) since it is essential for computing the ultimate effect of other variables in the system, but we are unlikely to consider this a direct cause of any behavior we want to explain. Further, we let

\[
\mathcal{V} = T(s, a, s') \cup R(s, a, s') \cup V(s) \cup \pi_{sa} \quad \forall s, a, s'.
\]

Finally, we let \( \mathcal{M} \) be the set of equations generated to solve for a policy, for instance by value iteration, the first two of which do not depend on other endogenous variables.

\[
\mathcal{M}_R := R_a^{ss'} = R(s, a, s'), \quad \forall s, a, s';
\]

\[
\mathcal{M}_T := T_a^{ss'} = T(s, a, s'), \quad \forall s, a, s'.
\]

The set of equations for the value at each state \( s \in S \) is

\[
\mathcal{M}_V := V(s) = \max_a \sum_{s' \in S} T_a^{ss'} [R_a^{ss'} + \gamma V(s')], \quad \forall s \in S.
\]

Lastly, we have the set of equations that represent action selection given the value function.
transitions, values, and partial policies on action selection.

Theorem 3. Let $M$ be a finite-horizon MDP and let $H$ be the planning graph representing the choice of action at state $s$. If $G$ is a layered causal graph representing $M$ at state $s$, then $G$ preserves cause-effect relationships in the reasoning process for action selection in $M$ at $s$.

Proof: Since $M$ is finite-horizon, then by Lemma 1 we can create an equivalent layered MDP and associated planning graph $H$. We can construct a function $\psi$ that induces a homomorphism $\psi : G \rightarrow H$ in the following way: map all nodes representing variables $V(s_i)$ or $T(s, a, s_i)$ in $G$, to the node representing $s_i$ in the planning graph $H$. Since the homomorphism $\psi : G \rightarrow H$ exists, and since MDPs may be represented as Bayesian networks, then by Lemma 2, $G$ captures all cause-effect relationships for action selection.

4.2 Causal Inference for Layered MDPs

We prove this by providing an algorithm for constructing layered MDPs. For any $s$, start at state $s$ and run BFS without an explored list until all paths of length $h$ have been explored. Next, append “i” to state IDs at the $i$th level of the tree. Last, aggregate any duplicate nodes, preserving their edges. The resulting planning graph is layered.

Lemma 2. If $G$ and $H$ are causal graphs of finite Bayesian networks, and there exists a homomorphism $G \rightarrow H$, then $G$ and $H$ preserve cause-effect relationships.

Proof: This result follows from Jacobs et al. (2019); Otsuka and Saigo (2022).

Definition 4.1. A layered Markov decision process is an MDP where, for all states $s \in S$, the planning graph whose root represents state $s$ is a layered graph.

It is also possible to move certain variables from $\mathcal{V}$ to the context $\mathcal{U}$, which can reduce complexity at the cost of eliminating certain variables from causal analysis. For example, we could move all reward variables. This will not compromise the underlying computation, but as we will see later reduces the number of variables that need to be tested counterfactually for causal influence. An important limitation is that the number of layers changes depending on the MDP. Layered MDPs are a particularly nice case because the structural causal models formed by our construction automatically form layered causal graphs. Given $M$ and a starting state $s_0$, we can construct a layered causal graph using the layered structure of the MDP. The following lemmas and theorem bound the class of MDP reasoning problems for which such layered graphs may be constructed exactly.

Lemma 1. Given a finite-horizon MDP with horizon $h$, there exists an equivalent layered MDP conditioned on the current state $s$, for every state $s \in S$.

**Proof:** We prove this by providing an algorithm for constructing layered MDPs. For any $s$, start at state $s$ and run BFS without an explored list until all paths of length $h$ have been explored. Next, append “i” to state IDs at the $i$th level of the tree. Last, aggregate any duplicate nodes, preserving their edges. The resulting planning graph is layered.

Lemma 2. If $G$ and $H$ are causal graphs of finite Bayesian networks, and there exists a homomorphism $G \rightarrow H$, then $G$ and $H$ preserve cause-effect relationships.

**Proof:** This result follows from Jacobs et al. (2019); Otsuka and Saigo (2022).
those variables. Given an event $\phi$, defined as a logical expression, for instance $\phi = (\neg a \land b)$, a weak cause of $\phi$ satisfies the following conditions:

1. Given the context $U = u$ and $X = x$, $\phi$ holds.
2. Some $W \subseteq (V \setminus X)$ and some $\bar{x}$ and $w$ exist such that using these values produces $\neg \phi$.
3. Some $W \subseteq (V \setminus X)$ and $w$ exist such that for all $W' \subseteq W$, $Z \subseteq V \setminus (X \cup W')$, where $w' = w|W'$ and $z = Z$ given $U = u$, $\phi$ holds when $X = x$.

Essentially, item 3 is saying that, given context $U = u$, $X = x$ alone is sufficient to cause $\phi$, independent of some other endogenous variables $W$.

**Definition 4.3.** An actual cause is a weak cause with the following property: $X$ is minimal. That is, no such $X' \subset X$ exists such that $X'$ is also a weak cause.

We now introduce two additional constructs, also from Eiter and Lukasiewicz (2006), that they use to establish a theorem on identification of weak causes. These constructs are general and do not have particular meaning with respect to MDPs. First, however, some notation. For every tuple $\phi$, given context $U = u$, and the assignments of variables $X = x$ and $Y = y$, $w' = w|W'$ refers to the restriction of $w$ to $W'$. Last, $[x|y] = x|(X \setminus Y) \cup y$.

$$R^0 = \{(p, q, F) | F \subseteq S^0, p, q \subseteq D(F), \exists u \in D(S^0 \setminus F) \forall p, q \in D(F) :$$

$$p \in p \iff \neg \phi_{pq}(u),$$

$$q \in q \iff \phi_{[q|Z(u)]|w}(u),$$

$$\forall W' \subseteq S^0 \setminus F, w' = w|W', Z \subseteq F \setminus S^k,$$

and

$$R^i = \{(p, q, F) | F \subset S^i, p, q \subseteq D(F), \exists u \in D(S^0 \setminus F) \exists r' \in R^{i-1} \forall p, q \in D(F) :$$

$$p \in p \iff F_{pq}(u) \in p',$$

$$q \in q \iff F_{[q|Z(u)]|w}(u) \in q',$$

$$\forall W' \subseteq S^i \setminus F, w' = w|W', Z \subseteq F \setminus S^k,$$ for $i > 0.$

**Theorem 4.** (From Eiter and Lukasiewicz (2006)) Let $S = (U, V, M)$ be a causal model. Let $X \subseteq V$, $x \in D(X), u \in D(U)$, and let $\phi$ be an event. Let $(S^0, \ldots, S^k)$ be a layered causal graph with respect to $X$, and let $R^k$ be defined as above. Then, $X = x$ is a weak cause of $\phi$ under $u$ in $S$ iff 

1. $X(u) = x$ and $\phi(u)$ in $S$, and 
2. $\exists (p, q, X) \in R^k$ such that $p \neq \phi$ and $x \in q$.

Algorithm 1 determines weak causes given a layered causal graph by constructing and checking sets $R^0, \ldots, R^k$, and generating sets of causal variables $X_E \subseteq X$, each of which satisfies the definition of weak cause with respect to event $\phi$. We will walk through Algorithm 2, as 1 is similar. The outer loop (line 4) looks at all possible subsets of variables, $F$, in the $i$th layer. Variables not in $F$ are assigned values $w$ one at a time, eventually looping over all possible sets of values (line 6). Then, for every tuple $R^i$ from layer $i - 1$ (line 7), we check the conditions for $p \in p$ (lines 8-10) and $q \in q$ (lines 11-23). Finally, for a given set $F$, we add all the qualifying $p, q$ to the tuple $R$ (line 24). One challenge in dealing with MDPs is the continuous domain of variables representing the transition and reward functions. In this paper we assume a discretization scheme, for which there are many reasonable choices. For example, since the transition function is bounded, one may create several equal intervals within $[0, 1]$. For the reward function, one could take the set of reward values actually realized within the MDP.

**Proposition 4.1.** Let $G$ be a layered causal graph representing an MDP, $X \subseteq S^k$, and $\phi \subseteq S^0$. Further, let $T$ and $R$ be discrete domains of possible values for $T(s, a, s')$ and $R(s, a, s')$. Executing Algorithm 1 on $G, X, \phi$ will produce sets of variables $X_W$ for which every $x_w \in X_W, x_w \subseteq X$ is a weak cause of $\phi$ under discretizations $T$ and $R$.

**Proof Sketch:** Algorithm 1 computes which pairs of values $W = w$ and $F = f$ falsify $\phi$, and which $F = f$ are robust to $W = w$. This produces unique tuples $R^i = (p, q, F)$. Algorithm 2 follows the same process, but instead of checking $\phi$, checks all tuples in $R^i$ (line 7), since any tuple in $R^i$ represents a weak cause of some tuple in $R^{i-1}$. Thus, weak causes of tuples in $R^i$ must be weak causes of $\phi$.

Algorithm 1 supports causal queries on graphs like those in Fig. 3, but it is not efficient. If we restrict the set of possible causal variables $X$ to the set of state factors of the MDP,
we get a layered causal graph like that in Fig. 2. For this problem formulation, we can use a simpler algorithm similar to that presented by Bertossi et al. (2020) based on the concept of responsibility and blame originating in Chockler and Halpern (2004). Similar to Algorithm 1, Algorithm 3 iterates through possible weak causal sets (line 4), and then progressively checks larger contingency sets \( W \) for possible assignments \( w \) that satisfy definition 4.2 (lines 5-8). Lines 11-15 and 19-22 check conditions 2 and 3, respectively. Finally, lines 23-27 compute the responsibility score, \( \rho \), and use it to determine if the set \( F \) is a weak cause. In addition to finding weak causal sets consistent with definition 4.2, the responsibility score provides a ranking over causal sets.

Given a set of weak causes \( C_W \), we can determine actual causes by iterating over \( C_W \) from smallest to largest, finding common subsets and eliminating supersets since they violate the minimality of actual causes. For causal queries made with respect to event \( \phi \) and potential causal variables \( X \), inference may be sped up by removing some irrelevant variables. Eiter and Lukasiewicz (2006) provide the following conditions for removing a variable \( v \). 1) \( v \in X \) is not connected via variables in \( \bar{V} \setminus X \) to \( \phi \), and 2) \( v \) is neither a direct parent of a variable in \( \phi \) or part of a chain connecting \( X \) to \( \phi \). Graphs that remove such variables are called strongly reduced. Given a layered causal graph \( G_{s_0} \), Algorithm 4 produces a strongly reduced layered causal graph \( G_{s_0}^{\phi \ X} \).

5 Generalization and Approximation

Although layered MDPs represent a large class of potential problems, Algorithms 1 and 2 have three limitations. First, they cannot represent infinite horizon problems, and second, even though the graph itself is straightforward to build for finite horizon problems, very large problems or problems with large horizons may still be prohibitively expensive to analyze. Finally, these algorithms cannot run when presented with continuous state factors. In this section we propose several algorithms to relax some of these limitations, either by constructing smaller, approximate causal models or by approximating more expensive inference processes.

5.1 Approximate Causal Models for MDPs

There are many possible methods for building approximate causal graphs. Our approach for arbitrary MDPs removes edges from the original causal graph to produce a layered causal graph that represents a simplified, finite-horizon version of the original MDP. To construct the original graph, we use an algorithm based on Iwasaki et al. (1986). Algorithm 5 generates a causal graph given a structural causal model by first constructing a bipartite graph, where variables (\( V \)) and equations (\( M \)) are nodes, and edges exist between variable nodes and equation nodes if that equation contains that variable. Given the bipartite graph, Hopkins-Karp is run to produce a perfect matching. This perfect matching is used to build a directed (causal) graph containing only variables.

The resulting causal graph may not be unique if there are circular dependencies in \( M \) (Iwasaki and Simon 1986; Trave-Massuyes and Pons 1997). In particular, since the Bellman update equation is a recurrence relation between the values of (potentially) all states in the MDP, cycles are almost guaranteed in the general case and we must remove them to satisfy the definition of a layered causal graph. We consider a finite horizon \( h \), beginning at state \( s_0 \), and let variables associated with states not reachable within \( h \) actions form causal ‘leaves’ and remove their incoming causal edges. Remaining non-layered structures are corrected by
executing Algorithm 6 on $G$. Proposition 4.2. and together produce one or more layered causal graphs. These algorithms may be pre-computed before deployment, processing step, while Algorithm 6 is run once for each state.

\[ \text{Algorithm 4 REDUCE CAUSAL GRAPH} \]

1: Input: Layered causal graph $G_{s_0}$, explanans $X$, event $\phi$
2: Output: Strongly reduced layered causal graph $G_{s_0}^\phi$
3: $G_{s_0}^\phi \leftarrow G_{s_0}$
4: for all $x \in V$ do
5: if $\exists y \in \phi$ then
6: remove $x$ and its edges from $G_{s_0}^\phi$
7: if $\forall y \text{ paths from } x \text{ to } y$, $\exists x' \in X \text{ along the path then}$
8: remove $x$ and its edges from $G_{s_0}^\phi$
9: for all $v \in V \setminus (X \cup \phi)$ do
10: if $\exists y \in \phi$ such that $v$ is a direct parent and $\exists x \in X$, $y \in \phi$
11: remove $v$ and its edges from $G_{s_0}^\phi$
12: return $G_{s_0}^\phi$

\[ \text{Algorithm 5 CONSTRUCT CAUSAL GRAPH} \]

1: Input: Set of variables $V$, set of equations $M$
2: Output: Causal graph $G$
3: $B \leftarrow \text{CONSTRUCTBIPARTITE}(V, M)$
4: $E_{PM} \leftarrow \text{HOPCRAFT-KARP}(B)$
5: $V \leftarrow V$, $E \leftarrow \emptyset$
6: for all $v \in V$ do
7: if $Q$ is a node in $B$ representing an equation.
8: for all $e, (v, Q) \in \text{EDGES}(v)$ do
9: if $e \in E_{PM}$ then
10: $V_Q$ is the set of variables in equation $Q$.
11: for all $v' \in V_Q$, $v' \neq v$ do
12: $E \leftarrow E \cup \text{EDGE}(v', v)$
13: else
14: for all $v' \in V_Q$, $v' \neq v$ do
15: $E \leftarrow E \cup \text{EDGE}(v, v')$
16: $G \leftarrow \{E, V\}$
17: return $G$

removing edges such that states farther from $s_0$ causally influence states nearer to $s_0$ (lines 11-14). In a sense, the causal chains flow from the horizon at time $t+h$ back in time to the present time $t$. These operations are executed simultaneously in Algorithm 6 which, given state $s_0$, removes edges from the original causal graph $G$ to produce a layered causal graph $G_{s_0}$ that can be used for causal analysis whenever the agent is in state $s_0$. Thus, Algorithm 5 is a single preprocessing step, while Algorithm 6 is run once for each state. These algorithms may be pre-computed before deployment, and together produce one or more layered causal graphs.

\[ \text{Proposition 4.2. If causal graph } G \text{ represents an MDP, executing Algorithm 6 on } G \text{ produces a layered causal graph.} \]

Proof: Algorithm 6 labels states exactly once. Each label represents the minimum length path in the planning graph from state $s_0$. For every vertex (line 11) labeled $i$ (layer $i$), only incident edges with labels $> i$ are preserved. Since the labels are minimal all remaining edges must come from vertices labeled $i+1$. Thus the resulting graph will have a finite set of layers, where edges only exist between nodes in layer $i$ to layer $i-1$ for all $i \in \{1, \ldots, h, \infty\}$.

5.2 Approximate Causal Inference for MDPs

Often, complete models are too expensive for exhaustive inference. Moreover, we may wish to apply more restrictive versions of definition 4.2, or extend the applicability of algorithms like 3 to continuous states factors. We address these problems via tweaks to algorithms presented previously.

If state factors have continuous domains, we cannot directly apply algorithm 3 as it requires iterating over all possible states. However, we can approximate the same inference by replacing lines 7 and 8 with a single for loop over vectors $w$ of size $\beta$ generated via sampling. Sampled input-output pairs are then constructed and counted in the same way and the responsibility score still indicates weak causality. The challenge is then to determine a sampling domain such that the samples have reasonable coverage over important counterfactual scenarios without becoming too numerous.

There are several other techniques applicable to both discrete and continuous domains. One class involves limiting the size of possible sets $W$ and $Z$, the main benefit of which is the drastic reduction in problem complexity. These restrictions of course diverge from definition 4.2, but can be made in a principled way that more or less preserves an order over the possible results. In particular, one may set $Z = \emptyset$ and or $|W| \leq \kappa$ for some $\kappa << |V|$. In the state factor case, the latter restriction is equivalent to requiring $\rho \geq \epsilon > 0$ rather than $\rho > 0$, for weak causality. Moreover, while we use one definition of causality, Halpern and Pearl, and later Halpern, have proposed several related definitions (Halpern 2015). We do not argue against definitions beyond 4.2.

Often, reward and transition functions in MDPs are not sampled randomly, but instead based on a high-level rule. For example, a reward might be proportional to the value of a particular state factor. Another example is the transition function in classic grid world problems where, regardless of location, agents move to the desired cell with 80% probability and slide left or right the other 20% of the time. When a problem has these structures we can replace loops over, for example, all possible values of $T(s, a, s')$ for all $s, a, s'$, and instead simply loop over the set of high-level rules, which is much, much smaller. These high-level rules constrain the transition and reward function to some manifold, and we can discretize this manifold to gain substantial speed up without sacrificing important possible worlds.

Finally, in some cases we may want to perform causal
| Alg. | time | space | Complexity | Bottleneck |
|------|------|-------|------------|------------|
| 1/2  |      |       | $2^{\lceil \frac{|E|}{|V|} \rceil} k^2$ | Enumerating causal sets |
|      |      |       | $|V|^2 k^2$ | Storing | R, $R^-$ |
| 3    | time | space | $2^{\lceil \frac{|E|}{|V|} \rceil} |D(F)|$ | Enumerating causal sets |
|      |      |       | $2^{\lceil \frac{|E|}{|V|} \rceil}$ | Storing weak causes |
| 4    | time | space | max$(|S|^3 k^3, k)$ | Connectivity checks + Path finding |
|      |      |       | $|S|^2 k$ | Storing all paths |
| 5    | time | space | $|V| + |E| \sqrt{|V|}$ | Bipartite construction + Hoperof-Karp |
|      |      |       | $3|V| + |A|k$ | Bipartite, Matching + Final graph |
| 6    | time | space | $|S|h + |S|^2 + |E| |S|^2$ | Connectivity checks + Label comparisons |
|      |      |       | Reachability matrix | |

Table 1: Worst-case time and space bounds for algorithms 1-6.

analysis on scenarios in the distant future. If this distance is great enough, the branching factor of the planning graph and causal graph may create too many variables, even when limiting domains or the sets $W$ and $Z$. In these cases, we can use a form of beam search to limit the number of operations that must be performed at each layer of the causal graph. The key idea is to measure the influence of variables on $\phi$ or on the previous layer and then keep only the top $m$ most influential variables as the search progresses through the layers. This measure is specific to the event $\phi$ and possibly also the set of potential explanans $X$.

6 Results

Here, we present preliminary bounds on the asymptotic run time and memory use for algorithms presented in §4 and §5, along with a discussion on the tightness of these bounds in practice and potential for pre-computation, caching, or lazy computation. We also present qualitative results from a small problem illustrating different possible causal queries.

6.1 Theoretical Analysis

Below, $|S|$, $|A|$, and $|F|$ denote the size of the state and action spaces and the number of state factors in an MDP, respectively. We will denote the number of variables and edges in causal models for state factor influence as $|V_{SF}|$ and $|E_{SF}|$, respectively, and will remove the subscript when referring to the causal model for all other MDP components.

$$|V_{SF}| = |F| + |S||S||A|,$$  
$$|E_{SF}| = |F| |S||S||(S||A|),$$  
and $|V| = |S|(k + |A|(k + k||S| + 1))$,  
$|E| = k|S|^2|A|\Omega$

Here, $\mathcal{V}$ and $\mathcal{E}$ are upper bounds, with actual values changing due to the topology of the MDP, $k$ is the number of layers in the layered causal graph, and $\Omega = (2/|A| + 1/|S| + 1 + 1/\varepsilon)$. Table 1 provides a summary of the worst-case time and space bounds, along with the limiting operations.

In practice, there are many instances where worst-case bounds are likely to be loose. In Algorithm 1, we must check all possible assignments of values to $X$, and $X$ may be as large as $\mathcal{V}$. However, usually $|X| \ll |\mathcal{V}|$, especially after applying Algorithm 4. Ultimately though, this runtime is not scalable, and we present alternatives designed to avoid running Algorithm 1. Similarly, in Algorithm 4, in the worst case we must check the connectivity between all pairs of variables. In practice this is a very loose bound since, again, typically $|X| \ll |\mathcal{V}|$. Bounds for Algorithm 3, while not as loose, are also poor estimates of in-practice cost, since it can employ short-circuiting.

There are also some bounds whose tightness depends on the connectivity of the MDP. For Algorithms 5 and 6 the bounds given are for fully-connected MDPs. However, most MDPs are not dense and therefore the quantity $|\mathcal{E}|$ will be significantly smaller. Of course, if we know that we will be building a causal graph representing an MDP, and not any structural causal model, we can construct it in $O(|V| + |\mathcal{E}|)$. This is the most likely case in practice. When converting an arbitrary causal graph to a layered one, Algorithm 6 is also faster if the MDP is sparse. Moreover, if $h$ is small compared to the width of the MDP, runtime will significantly decrease since any nodes labeled $\infty$ can be handled in linear time.

The most useful pre-computation steps are to construct the layered causal graphs for all states in the MDP, regardless of how $\phi$ and $X$ are specified, and to compute any connectivity, reachability, or path information that might be needed later. Given an $X$ and $\phi$ online, reductions and causal model approximations can then be applied quickly.

6.2 Example Explanations

Here, we present qualitative results from several related scenarios in an example domain. The purpose of these experiments is primarily to show 1) how our approach can handle many semantically different types of causal queries, corresponding to several different conceptions of MDP explanation in the literature, and 2) how our algorithms produce sensible, helpful explanans, that in some cases uncover surprising or subtle aspects of agent decision making. While communicating such explanations to a human user in a natural language is beyond the scope of this paper, in practice, there are existing methods for inserting variables uncovered in causal analysis into natural language templates. Hence, we focus on generating the best causal set of variables. As there are no state-of-the-art systems with the same capabilities, and thorough empirical evaluation requires several other systems as well as human subjects testing, we present qualitative results with some analysis. First, however, we present one last formalism to aid in comparison.

Definition 6.1. Y-type explanations are explanations whose explanans come from the set of variables $Y$. For example, F-type explanations use the set of state factors $F$. We use $R$ for reward, $T$ for transition, and $V$ for value of a state.

We have identified 4 types of explanation: F (Elizalde et al. 2009; Russell et al. 2019), R (Khan et al. 2009; Juozapaitis et al. 2019; Bertram and Wei 2018), T (Wang et al. 2016), and V (Pouget et al. 2020), where $V$-type explanations may relate to either the value function or specific states. Each of these papers defines metrics, algorithms, and definitions particular to their choice of $Y$. The following examples will show how using SCMs for MDPs allows for any choice of $Y$, with relatively little modification.

We now detail the MDP representing our example domain. Consider a robot navigating a 2D environment with several goal states. In addition to $(x, y)$ location, the agent also has a time to failure $c$, and can sense whether the current terrain $t$ is unremarkable, suitable for servicing repairs,

\footnote{Goal states are not required, but help illustrate explanations.}
or may cause the agent to get stuck. Thus, the state factors are \( x \in \{1, \ldots, 9\}, y \in \{1, \ldots, 6\}, c \in \{0, \ldots, 5\}, t \in \{\text{NORMAL}(N), \text{SERVICE CENTER}(SC), \text{DANGEROUS}(D)\}\). The actions are \( A = \{\text{UP}, \text{LEFT}, \text{RIGHT}, \text{SERVICE}\}\). If the agent breaks down and is not in a service state, it requires an expensive mobile service procedure and suffers \(-10\) reward. Receiving service has a reward of \(-1\) or \(2\), depending on the scenario. Goal state rewards vary across different scenarios. All other actions have \(-1\) reward. Last, transitions are deterministic except for actions \text{LEFT} and \text{RIGHT} when taken at \((5,3)\), which result in agent locations of \((4.3)\) (with probability \(T_L = 0.6\)) or \((4.4)\), and \((6,3)\) (with probability \(T_R = 0.01\)) or \((6,4)\), respectively. One possible domain is shown in Figure 4, and Table 2 shows the value of all configurable variables and the current state \(s_0\) for each scenario. For all explanations, we considered event \( \phi = \pi_{s_0}(\pi(s_0))\), which is a single Boolean variable representing taking action \(\pi(s_0)\) in state \(s_0\). For each type of explanation \(Y\), we set \(X = Y\). Thus, we considered the power set of those variables (i.e., \(F, R, T, V\)) and calculated their MeanRESP score using Algorithm 4. For clarity, we only report weak or actual causes of size 1.

**Scenario 1:** The cause for action \text{SERVICE} is state factor \(t\) (\(\rho = 0.64\)). In this scenario, since \(R_C > 0\), it is always optimal to service, as it prevents failure in normal cells and is better than the \(-1\) reward for taking other actions.

**Scenarios 2 and 3:** There are two causes for taking the action \text{UP} in scenario 2. The terrain type \(t\) (\(\rho = 0.50\)) and the location (\(\rho = 0.26\)). We have the same two causes in scenario 3, but the action is \text{LEFT} and the location now has \(\rho = 0.60\). This increase is due to the unique topology at \((5,3)\) compared to \((5,2)\), making \text{LEFT} a less common action under this policy. In both cases, \(t\) is a cause as if \(t\) was set to \text{SC} the agent would have taken action \text{SERVICE}.

**Scenario 4:** Here, the agent first takes action \text{LEFT} and gets service done at \((3,3)\). Then, it heads toward goal 2. The agent does not directly go toward goal 2 as it will break down at \((4,5)\), requiring expensive mobile service. Our framework identified the location (\(\rho = 0.68\)), time to failure (\(\rho = 0.85\)) as causes, and \(t\) (\(\rho = 0.60\)). Note that unlike in the previous three scenarios time to failure is both a cause and has the highest responsibility.

**Scenario 5:** \(R_1\) (\(\rho = 0.30\)) and \(R_3\) (\(\rho = 0.55\)) are causes for the action \text{LEFT}. Because we bound \(R = [-100, 100]\), no values for \(R_2\) or \(R_4\) can change the outcome. \(R_4\) is already the maximum, and \(R_2\) alone is not a cause due to its relatively weak effect on the expected value of that subtree.

**Scenario 6:** Here, only \(R_3\) (\(\rho = 0.55\)) is causal. Since \(R_C > 0\), the agent exploits this by repeatedly taking service at \((3,3)\). Hence, in this setting \(R_1\) alone does not affect the agent’s policy since goal 1 (and 4) will never be visited. This exemplifies discovery of a misspecified objective.

**Scenarios 7 and 8:** In scenario 7 due to the imbalance in rewards to the right, the only weak cause for the action \text{LEFT} is \(T_R\) (\(\rho = 0.66\)). If the uncertainty is reduced, the agent will take action \text{RIGHT}. The same is not true for scenario 8, in which both \(T_R\) and \(T_L\) are weak causes.

**Scenario 9:** When the agent takes action \text{LEFT} at \((5,3)\) it will most likely (\(\rho = 0.6\)) end up in the least rewarding goal, goal 2. Using beam search, we find that although the most likely outcome is to arrive at goal 2, the most influential set of trajectories in terms of action selection leads to goal 1, since it contributes most to the expected value.

These scenarios encompass all four types of explanation for taking action \text{LEFT} at \((5,3)\), providing several different types of insight into the effect of different variables on the policy. Some explanations were more enlightening than others in particular scenarios, and this underscores the importance of generality and flexibility when generating explanations since it is rarely clear before deployment which types of explanation will be most effective in which scenarios. Most existing methods only analyze a single component of the MDP and therefore can only produce a fraction of these explanations. This is a critical limitation when applying these explanation techniques to MDPs in the many domains we may want them. Moreover, it is evident from scenario 6 that our framework, besides being used to generate explanations that engender user trust in AI systems, may also be used to debug such systems.

### 7 Conclusion
We present a unifying framework for causal analysis of Markov decision processes based on structural causal models, motivated by generating explanations of MDP agent behavior. This framework provides a solid theoretical foundation for explainable sequential decision making, offers many potential approximate techniques to increase efficiency, and can simultaneously support causal queries using different decision problem components, which have not been possible prior to this work. While this approach is well-grounded, there are several questions left for future empirical investigation, most notably the many choices one could make on a deployed system and their impact on the usability of the explanations for people interacting with the system.

| ID | Y | \((x, y, t, c)\) | \(R_1\) | \(R_2\) | \(R_3\) | \(R_4\) | Action |
|----|---|-----------------|-------|-------|-------|-------|--------|
| 1  | \(F\) | \((5,1), SC, 5\) | 80    | -50   | 40    | 100   | 2      | SERVICE |
| 2  | \(F\) | \((5,2), N, 4\) | 80    | -50   | -40   | 100   | 2      | UP      |
| 3  | \(F\) | \((5,3), N, 3\) | 80    | -50   | 40    | 100   | 2      | LEFT    |
| 4  | \(F\) | \((4,3), N, 2\) | 80    | 90    | 40    | 100   | 1      | LEFT    |
| 5  | \(R\) | \((5,3), N, 3\) | 80    | -50   | 40    | 100   | 2      | LEFT    |
| 6  | \(R\) | \((5,3), N, 3\) | 80    | -50   | 40    | 100   | 2      | LEFT    |
| 7  | \(T\) | \((5,3), N, 3\) | 80    | 80    | 70    | 100   | 1      | LEFT    |
| 8  | \(T\) | \((5,3), N, 3\) | 80    | -50   | 40    | 100   | 1      | LEFT    |
| 9  | \(V\) | \((5,3), N, 3\) | 80    | -50   | 40    | 100   | 1      | LEFT    |
References
Bellman, R. 1952. On the theory of dynamic programming. National Academy of Sciences of the United States of America 38(8):716.
Bertossi, L.; Li, J.; Schleich, M.; Suciu, D.; and Vagena, Z. 2020. Causality-based explanation of classification outcomes. arXiv preprint arXiv:2003.06868.
Bertram, J., and Wei, P. 2018. Explainable deterministic MDPs. arXiv preprint arXiv:1806.03492.
Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. arXiv preprint arXiv:1701.08317.
Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D. E.; and Kambhampati, S. 2020. The emerging landscape of explainable automated planning & decision making. In International Joint Conference on Artificial Intelligence (IJCAI), 4803–4811.
Chen, J. Y.; Lakhmani, S. G.; Stowers, K.; Selkowitz, A. R.; Wright, J. L.; and Barnes, M. 2018. Situation awareness-based agent transparency and human-autonomy teaming effectiveness. Theoretical Issues in Ergonomics Science 19(3):259–282.
Chockler, H., and Halpern, J. Y. 2004. Responsibility and blame: A structural-model approach. Journal of Artificial Intelligence Research 22:93–115.
Eiter, T., and Lukasiewicz, T. 2006. Causes and explanations in the structural-model approach: Tractable cases. Artificial Intelligence 170(6-7):542–580.
Elizalde, F.; Sucar, E.; Noguez, J.; and Reyes, A. 2009. Generating explanations based on Markov decision processes. In Mexican International Conference on Artificial Intelligence, 51–62. Springer.
Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. arXiv preprint arXiv:1709.10256.
Freedman, D. A. 2007. Statistical models for causation. The SAGE Handbook of Social Science Methodology 127–146.
Giordano, L., and Schwind, C. 2004. Conditional logic of actions and causation. Artificial intelligence 157(1-2):239–279.
Halpern, J. Y., and Pearl, J. 2000. Causes and explanations: A structural-model approach. Part I: Causes. The British Journal for the Philosophy of Science 52(3):613–622.
Halpern, J. Y., and Pearl, J. 2005. Causes and explanations: A structural-model approach. Part II: Explanations. The British Journal for the Philosophy of Science 56(4):889–911.
Halpern, J. 2015. A modification of the Halpern-Pearl definition of causality. In Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 3022–3033.
Hayes, B., and Shah, J. A. 2017. Improving robot controller transparency through autonomous policy explanation. In ACM/IEEE International Conference on Human-Robot Interaction (HRI), 303–312.
Heider, F. 1958. The psychology of interpersonal relations. Wiley, New York.
Hesslow, G. 1988. The problem of causal selection. Contemporary Science and Natural Explanation: Commonsense Conceptions of Causality 11–32.
Hilton, D. J., and Slugoski, B. R. 1986. Knowledge-based causal attribution: The abnormal conditions focus model. Psychological Review 93(1):75.
Hilton, D. J. 1990. Conversational processes and causal explanation. Psychological Bulletin 107(1):65.
Hilton, D. J. 1996. Mental models and causal explanation: Judgements of probable cause and explanatory relevance. Thinking & Reasoning 2(4):273–308.
Iwasaki, Y., and Simon, H. A. 1986. Causality in device behavior. Artificial intelligence 29(1):3–32.
Jacobs, B.; Kissinger, A.; and Zanasi, F. 2019. Causal inference by string diagram surgery. In International Conference on Foundations of Software Science and Computation Structures, 313–329. Springer.
Juozapaitis, Z.; Koul, A.; Fern, A.; Erwig, M.; and Doshi-Velez, F. 2019. Explainable reinforcement learning via reward decomposition. In IJCAI/ECAI Workshop on Explainable Artificial Intelligence.
Karimi, A.-H.; Schölkopf, B.; and Valera, I. 2021. Algorithmic recourse: From counterfactual explanations to interventions. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 353–362.
Khan, O.; Poupart, P.; and Black, J. 2009. Minimal sufficient explanations for factored Markov decision processes. In International Conference on Automated Planning and Scheduling (ICAPS), volume 19.
Lewis, D. 1974. Causation. The Journal of Philosophy 70(17):556–567.
Linegang, M. P.; Stoner, H. A.; Patterson, M. J.; Seppelt, B. D.; Hoffman, J. D.; Crittendon, Z. B.; and Lee, J. D. 2006. Human-automation collaboration in dynamic mission planning: A challenge requiring an ecological approach. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 50(23):2482–2486.
Lipton, P. 1990. Contrastive explanation. Royal Institute of Philosophy Supplements 27:247–266.
Lombrozo, T. 2006. The structure and function of explanations. Trends in Cognitive Sciences 10(10):464–470.
Lombrozo, T. 2010. Causal-explanatory pluralism: How intentions, functions, and mechanisms influence causal ascriptions. Cognitive Psychology 61(4):303–332.
Lombrozo, T. 2012. Explanation and abductive inference. In Holyoak, K. J., and Morrison, R. G., eds., The Oxford Handbook of Thinking and Reasoning. Oxford University Press. 260–276.
Lucic, A.; Haned, H.; and de Rijke, M. 2020. Why does my model fail? Contrastive local explanations for retail forecasting. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 90–98.

Mackie, J. L. 1980. The cement of the universe: A study of causation. Clarendon Press.

Madumal, P.; Miller, T.; Sonenberg, L.; and Vetere, F. 2020. Explainable reinforcement learning through a causal lens. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2493–2500.

Mercado, J. E.; Rupp, M. A.; Chen, J. Y.; Barnes, M. J.; Barber, D.; and Procci, K. 2016. Intelligent agent transparency in human-agent teaming for Multi-UxV management. Human Factors 58(3):401–415.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence 267:1–38.

Mittelstadt, B.; Russell, C.; and Wachter, S. 2019. Explaining explanations in AI. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 279–288.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 607–617.

Otsuka, J., and Saigo, H. 2022. On the equivalence of causal models: A category-theoretic approach. arXiv preprint arXiv:2201.06981.

Panigutti, C.; Perotti, A.; and Pedreschi, D. 2020. Doctor XAI: An ontology-based approach to black-box sequential data classification explanations. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 629–639.

Pouget, H.; Chockler, H.; Sun, Y.; and Kroening, D. 2020. Ranking policy decisions. arXiv preprint arXiv:2008.13607.

Russell, J., and Santos, E. 2019. Explaining reward functions in Markov decision processes. In Thirty-Second International FLAIRS Conference.

Salmon, W. C. 2006. Four decades of scientific explanation. University of Pittsburgh press.

Stubbs, K.; Hinds, P. J.; and Wettergreen, D. 2007. Autonomy and common ground in human-robot interaction: A field study. IEEE Intelligent Systems 22(2):42–50.

Trave-Massuyes, L., and Pons, R. 1997. Causal ordering for multiple mode systems. In Proceedings of the Eleventh International Workshop on Qualitative Reasoning, 203–214.

Wang, N.; Pynadath, D. V.; and Hill, S. G. 2016. The impact of POMDP-generated explanations on trust and performance in human-robot teams. In International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 997–1005.

Williams, J. J.; Lombrozo, T.; and Rehder, B. 2013. The hazards of explanation: Overgeneralization in the face of exceptions. Journal of Experimental Psychology: General 142(4):1006.

Woodward, J. 2005. Making things happen: A theory of causal explanation. Oxford university press.

Zhang, Y.; Liao, Q. V.; and Bellamy, R. K. 2020. Effect of confidence and explanation on accuracy and trust calibration in Al-assisted decision making. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency, 295–305.