Automated Discovery of Actual Cause Judgments in Complex Environments

Anonymous authors
Paper under double-blind review

Abstract

Many problems in artificial intelligence, such as explanation, blame attribution, and assessment of harm, can be formally represented within the framework of *actual causality*, which describes the causation of particular observed events that have already occurred. However, existing definitions of actual cause often identify many events as actual causes in large, complex environments, even though many of them rarely influence the outcome. By contrast, humans are able to make highly specific *causal judgments* about a few specific causes for a particular outcome. Prior work in both actual causality and psychology has characterized this as an issue of *normality*, which assigns actual cause based on the "normative" or "common" outcomes. However, current approaches, which often require defining hierarchies of normality, do not scale well to complex and continuous-valued environments. This paper introduces *invariant-set actual-cause judgment (IAJ)*, a novel framework that formalizes established intuitions of normality by using *context-specific independencies* to identify the set of *actual-cause judgments*: a subset of causal variables that are characteristic of producing the outcome. Inspired by actual cause, IAJ captures intuitions about the key variables that describe why an outcome occurs. We then present *joint optimization for minimum actual-cause judgment (JOA)*, a practical algorithm that learns from observational data to infer approximated IAJs. We show empirically that JOA identifies actual-cause judgments with significantly higher accuracy than existing methods across a set of complex, continuous-valued environments.

1 Introduction

A growing body of research in artificial intelligence (AI) focuses on inferring the causes of specific events in a particular observed context. This approach has been applied to various problems in AI such as explanation (Nashed et al., 2023; Beckers, 2022), blame and responsibility (Chockler & Halpern, 2004), assessment of harm (Richens et al., 2022; Beckers et al., 2022), and sequential decision-making (Pitis et al., 2020; Triantafyllou et al., 2022; Gyevnar et al., 2024). Inferring causes in a particular context can often yield different results from traditional causal inference methods that infer general causal relationships between variables (Hall, 2004). This is because while many variables can participate in causal relationships, only a select few may be responsible for the outcome in the particular context that was observed, and this can vary across different contexts. *Actual causality* (Pearl, 2000; Halpern, 2016), which formally describes the causes of particular events in a specific observed context, offers a principled and general framework for addressing this problem.

However, existing approaches in actual causality can struggle to scale effectively to high-dimensional and continuous environments due to their overly permissive nature. In particular, an event qualifies as an actual cause if there is at least one counterfactual scenario where a different outcome could have occurred. For example, consider a robot trying to push a block across a table in a room filled with objects. The position of every single object in the room would be a valid actual cause, including those in the background or far away from the table, since any of them could have been placed in front of the block and prevented it from moving. While this set of causes technically meets the criteria for actual causation, it can be prohibitively large and impractical for AI problems in complex environments.

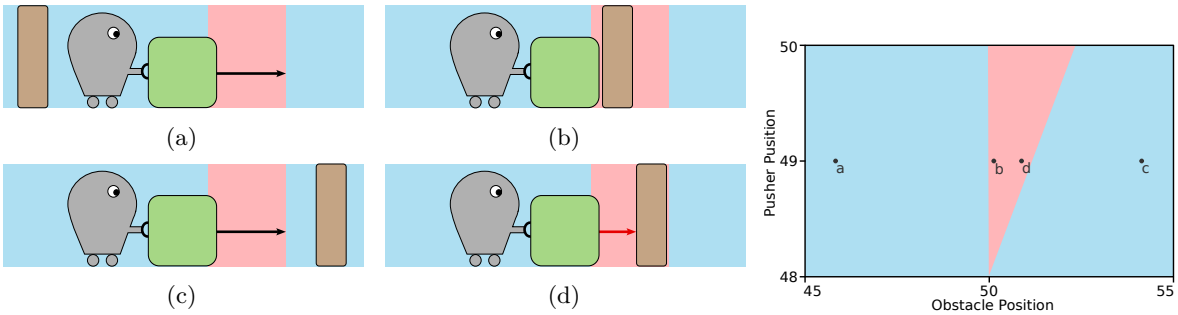


Figure 1: **Invariant preimages** of the block position corresponding to a portion of the state space from Example 2.1. The block is at position 50 and the pusher attempts to move it to position 52. The obstacle has no impact on the block position in states **(a)** and **(c)**, but will obstruct the pusher in states **(b)** and **(d)**. The blue region represents states where only the pusher can affect the block position—an *invariant preimage* of the block position with respect to the pusher. When the observed state is in this region, the obstacle is not an invariant-set actual-cause judgment of the block being pushed successfully.

This problem is broadly described in the actual causality literature as an issue of *normality* (Icard et al., 2017), which describes the degree to which certain events are considered normal, rare, or abnormal. Normality in causal models has traditionally been represented using a rank ordering over states (Halpern & Hitchcock, 2015). However, manually specifying such an ordering is impractical in large and continuous domains with complex dynamics. Furthermore, prior work does not provide guidance on how to manually or automatically obtain this ordering. As a result, this approach is infeasible in many AI settings.

By contrast, humans make *specific judgments* about a narrow subset of causal variables to explain outcomes (Knobe & Fraser, 2008; Gerstenberg et al., 2015; Quillien & Lucas, 2024), handling normality intuitively (Kahneman & Miller, 1986; Willemsen & Kirfel, 2019; Quillien & Lucas, 2024). These judgments can be quite powerful in isolating the key actual causes of an event. Inspired by this, we use the term *actual-cause judgments* to denote a subset of the actual causes judged to be the primary causes of an outcome, based on a formal criterion that is more restrictive than those used to identify actual causes.

This work focuses on the *invariant-set actual-cause judgment (IAJ)* framework for identifying actual-cause judgments, which extends established definitions of actual cause by accounting for normality through *context-specific independencies* (Boutilier et al., 2013) that hold in particular states. In the previous example, the robot was in a state where the final block position would have remained unaffected by most rearrangements of the distant objects, even if their positions changed dramatically. IAJ formalizes this intuition by identifying sets of states in which the outcome is independent of all variables whose values are not IAJs—though they might be considered actual causes. These sets are referred to as *invariant preimages (IVPs)*, based on the mathematical notion of "preimage" as the set of inputs that produce a particular outcome—in this case, the same or "invariant" outcome. As sets, these IVPs partition the space. By identifying which IVP the observed state belongs to, we can directly identify which variables determined the outcome at that particular state. This enables automated inference of IAJs.

To learn the IVPs necessary to infer IAJs in high-dimensional environments, we introduce *joint optimization for minimum actual-cause judgment (JOA)*. JOA learns from observational data to identify which subset of variables determined the outcome at a given state, and returns IAJs of the outcome. We show that IAJ maintains agreed-upon verdicts in a set of examples from the actual causality literature, and we demonstrate that JOA shows impressive results when compared to existing methods in both synthetic and RL domains. By accounting for normality without requiring manual specification of normal and abnormal states, our approach scales IAJ to continuous-state domains and hundreds of dimensions—as compared to prior actual cause inference methods, which are reserved for discrete state spaces with a limited number of unique states.

2 Illustrative Example

The following example illustrates how existing definitions of actual cause can overlook context-specific independence and accordingly permit more causes. We later expand on this with a formal analysis of these prior definitions in Section 4.7, and include additional examples in Appendix F.

Example 2.1 (**1-D Block Pushing**)**.** A pusher, a block, and an obstacle are present in a 1-D world within the range $[0, 100]$. The pusher always moves 2 units to the right. If the block is within 2 units in front of the pusher, it will be pushed until either the pusher stops or an obstacle is encountered. Say the pusher is at position 49, with the block in front of it at position 50 and the obstacle behind it at position 20. Since the block is unobstructed, the pusher successfully moves it forward to position 52.

Prior definitions of actual cause, such as the Halpern-Pearl definitions (Halpern & Pearl, 2005; Halpern, 2015; 2016), would declare that both the pusher and the obstacle are actual causes of the block being pushed forward in this scenario. The pusher would be an actual cause since the block would not have moved forward if the pusher were in any other position. Similarly, the obstacle position would also be an actual cause since the observed outcome would not have happened if the obstacle were placed right in front of the block.

However, this reasoning does not account for how far away the obstacle actually was, or how much it could have influenced the block position from its current location. The existence of a single state where the obstacle impedes the mover is enough to declare the obstacle position as a cause, whether it was 30 meters away or 3 million meters away. Moreover, if there are hundreds of obstacles in the world, then *all* of them would be considered actual causes. While these conclusions are technically correct, they can run counter to human intuition and judgment (Icard et al., 2017; Gerstenberg et al., 2015; Quillien & Lucas, 2024) and are impractical to apply for downstream tasks in large and complex environments. For instance, if a decision-making agent constantly attended to all of these objects at once, its decisions could easily be influenced by even slight movements in the distant objects, resulting in unstable performance.

A naive approach might simply prune out the objects that are far away and rarely or never interact with the agent, or more generally, exclude statistically improbable events in the set of candidate actual causes. However, if we encounter a scenario where a faraway object *actually does* interact with the agent, deprioritizing or eliminating it from the agent's representation can lead to a catastrophic failure. Consequently, the approach to handling distant objects or rarely occurring causes in practice can vary depending on the specific scenario.

This example highlights the issue of *normality*: judgments of actual causation can be influenced by the degree to which certain events are seen as normal or abnormal (Halpern & Hitchcock, 2015). A classic example of this is the *Queen of England problem* (Schaffer, 2005). A gardener goes on vacation, and their flowers die from not being watered. The Queen of England would technically be an actual cause of the flowers dying, as she could have watered them, but did not. This contradicts most human judgments (Beebee, 2004; McGrath, 2005; Livengood & Machery, 2007) since in most scenarios it is unrealistic to expect the Queen of England to visit someone's garden. However, studies show that when the Queen of England actually visits the garden, her actions are prioritized as a cause over that of the gardener, even if they watered the plants too (Beebee, 2004; McGrath, 2005; Livengood & Machery, 2007).

The pusher and obstacle play analogous roles to the gardener and the Queen of England, respectively, with the state corresponding to the above scenario lying in the blue region from Figure 1. The final block position depends on the obstacle position only when the observed state is in the red region, and is independent of the obstacle position in every state from the blue region. We denote the blue region as an *invariant preimage (IVP)* of the block position with respect to the pusher position, since the outcome (block position) is invariant to all remaining inputs (obstacle position) for every state in this region. Knowing that the observed scenario fell in the blue IVP would indicate that the obstacle, or the Queen of England, can be excluded as IAJ.

3 Background

Our definition of invariant-set actual-cause judgment builds on definitions and notation introduced by Halpern and Pearl (Halpern & Pearl, 2005; Halpern, 2015), as well as prior definitions of causal sufficiency and necessity (Beckers, 2021). We reference prior work on inferring actual causes in Section 6, and other related

work in Appendix B. This paper focuses on the problem of determining whether an event $\mathbf{X} = \mathbf{x}$ is an IAJ of an outcome $\mathbf{Y} = \mathbf{y}$ in a particular observed context, for an acyclic structural causal model M with endogenous variables $\mathbf{V}$ and exogenous variables $\mathbf{U}$. Prior work in actual causality often refers to the values $\mathbf{U} = \mathbf{u}$ as the *context*, whereas papers on context-specific independence use *context* to mean the observed state (Boutilier et al., 2013). We adopt the latter definition in this paper and take $\mathbf{U} = \mathbf{u}$ as given for the definitions that follow. However, we note that IAJs are distinct quantities from actual causes, selecting a subset of the actual causes which are dependent on the invarient properties of the dynamics. In Appendix A, we provide an extended discussion of the reasoning for this distinction.

Given an outcome variable $\mathbf{Y}$, we denote the set of all endogenous variables excluding $\mathbf{Y}$ as the set of *state variables* $\mathbf{S} = \mathbf{V} - \mathbf{Y}$, and an assignment to these variables $\mathbf{S} = \mathbf{s}$ as a *state*. Let $\mathcal{R}(\cdot)$ be a function that associates every variable or set of variables $\mathbf{X}$ with a nonempty set of possible values $\mathcal{R}(\mathbf{X})$. For a given state $\mathbf{s}$, the assignment to the variables $\mathbf{X}$ is denoted by $\mathbf{s}_{\mathbf{X}}$. The function in M that returns the value of $\mathbf{Y}$ is given by $\mathbf{Y} = f_{\mathbf{Y}}(\mathbf{s}, \mathbf{u})$, where in an abuse of notation we let $f_{\mathbf{Y}}$ take a full state as an argument and assume it ignores all state variables that are descendants of $\mathbf{Y}$. The Boolean statement $(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}]\mathbf{Y} = \mathbf{y}$ is true when under the model M, the outcome $\mathbf{Y} = \mathbf{y}$ occurs whenever the exogenous variables take on values $\mathbf{u}$ and the intervention $\mathbf{X} \leftarrow \mathbf{x}$ is applied.

We build a definition of IAJ from the following general framework. Existing definitions replace each of these conditions with a specific interpretation of necessity, sufficiency, and minimality. Details on each of these conditions and the witness set can be found in Halpern (2016) and Beckers (2021).

Definition 3.1 (General Framework for Actual Causation)**.** For a given model M and observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is an *actual cause* of the outcome $\mathbf{Y} = \mathbf{y}$ if:

AC1 (*Factual*): $\mathbf{X} = \mathbf{x}$ and outcome $\mathbf{Y} = \mathbf{y}$ actually happened: $\mathbf{s}^*_{\mathbf{X}} = \mathbf{x}$ and $(M, \mathbf{u}) \vDash (\mathbf{Y} = \mathbf{y})$.

AC2(a) (*Necessity*): There exists a counterfactual assignment $\mathbf{X} = \mathbf{x}'$ and witness set $\mathbf{W}$ such that $(\mathbf{X} = \mathbf{x}', \mathbf{W} = \mathbf{s}^*_{\mathbf{W}})$ is not sufficient for $\mathbf{Y} = \mathbf{y}$.

AC2(b) (*Sufficiency*): $(\mathbf{X} = \mathbf{x}, \mathbf{W} = \mathbf{s}^*_{\mathbf{W}})$ is sufficient for $\mathbf{Y} = \mathbf{y}$.

AC3 (*Minimality*): There is no strict subset $\mathbf{Z} \subset \mathbf{X}$ for which the assignment $\mathbf{Z} = \mathbf{z}$ satisfies AC1, AC2(a) and AC2(b), where $\mathbf{z}$ refers to the values of variables $\mathbf{Z}$ in $\mathbf{x}$.

4 Invariant-Set Actual-Cause Judgment

In this section, we introduce a definition of invariant-set actual-cause judgment that formalizes the intuitions of normality and context-specific independence described in Example 2.1. The state space in the example is partitioned into two regions: one where the block position depends only on the pusher position, and one where it depends on both the obstacle and block positions. We establish criteria for defining these partitions and use them to propose a novel description for an actual-cause judgment.

4.1 Invariant Preimage

We define an *invariant preimage* $IVP(\mathbf{X}; \mathbf{Y})$ to be a partition of the state space in which the outcome $\mathbf{Y}$ is invariant to all state variables that are not in $\mathbf{X}$. For any state $\mathbf{s}$ in this set, the outcome is not affected by any change to the values of the remaining variables $\mathbf{C} = \mathbf{S} - \mathbf{X}$ that results in another state $\mathbf{s}'$ in the same partition.

Definition 4.1 (Invariant Preimage)**.** A partition $IVP(\mathbf{X}; \mathbf{Y}) \subseteq \mathcal{R}(\mathbf{S})$ is an *invariant preimage* of the outcome variables $\mathbf{Y}$ with respect to the variables $\mathbf{X}$ if all states $\mathbf{s}$ within the partition satisfy:

$$f_{\mathbf{Y}}(\mathbf{s}_{\mathbf{X}}, \mathbf{s}_{\mathbf{C}}, \mathbf{u}) = f_{\mathbf{Y}}(\mathbf{s}_{\mathbf{X}}, \mathbf{s}'_{\mathbf{C}}, \mathbf{u}) \quad \forall\, \mathbf{s}' \in IVP(\mathbf{X}; \mathbf{Y}) \text{ where } \mathbf{s}'_{\mathbf{X}} = \mathbf{s}_{\mathbf{X}} \tag{1}$$

Each invariant preimage signifies a partition of the state space in which a particular context-specific independence holds. For example, if a state $\mathbf{s}^*$ was present in the set of states given by $IVP(\mathbf{X}; \mathbf{Y})$, then the value of the outcome $\mathbf{Y}$ would remain constant for every other state in the set where $\mathbf{X}$ takes on the same values

$\mathbf{s}^*_{\mathbf{X}}$. We refer to a set of invariant preimages with respect to outcome variable $\mathbf{Y}$ as $\mathcal{I}(\mathbf{Y}) = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_{2^{|\mathbf{S}|}}\}$ where each set in $\mathcal{I}(\mathbf{Y})$ corresponds to a subset of the state variables $\mathbf{S}$.

Figure 1 describes a 2D state space with two invariant preimages. The red region in the center corresponds to $IVP((\mathbf{P}, \mathbf{O}); \mathbf{Y})$ where the block position $\mathbf{Y}$ depends on the pusher position $\mathbf{P}$ and obstacle position $\mathbf{O}$. The large blue region corresponds to $IVP(\mathbf{P}; \mathbf{Y})$, where $\mathbf{Y}$ is unaffected by obstacle position $\mathbf{O}$. In Section 5, we devise discovery algorithms to obtain a set of invariant preimages $\mathcal{I}(\mathbf{Y})$ and a mapping from states to these invariant preimages. For the rest of this section, we use a given $\mathcal{I}(\mathbf{Y})$ to reason about causal sufficiency, necessity, and minimality.

4.2 Causal Sufficiency

In the language of causal models, causal sufficiency is established by treating $\mathbf{X} = \mathbf{x}$ as an intervention and $\mathbf{Y} = \mathbf{y}$ as a consequence of that intervention. However, this does not specify what values the remaining state variables $\mathbf{C}$ should take, and different choices of conditions for $\mathbf{C}$ lead to different definitions of causal sufficiency. If we do not intervene on any other variables and just let them take on their observational values, we would obtain the following definition, reproduced from Beckers (2021):

Definition 4.2 (Weak Sufficiency)**.** For an observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is *weakly sufficient* for the outcome $\mathbf{Y} = \mathbf{y}$ under witness set $\mathbf{W}$ if $(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}$.

However, this definition does not require any intervention on the remaining variables $\mathbf{C}$, allowing their values to be affected by the intervention $\mathbf{X} \leftarrow \mathbf{x}$. As a result, $\mathbf{C}$ is not prevented from influencing the outcome even though $\mathbf{X} = \mathbf{x}$ is considered sufficient. For the given outcome to be solely determined by $\mathbf{X} = \mathbf{x}$, the observed state should be such that the values of $\mathbf{C}$ had no bearing on the outcome $\mathbf{Y}$. This can be described as a *context-specific independence* between $\mathbf{Y}$ and the remaining variables $\mathbf{C}$.

The invariant preimages precisely define this context-specific independence. If the observed state $\mathbf{s}^*$ fell within the set $IVP(\mathbf{X}; \mathbf{Y})$, then we would know that the outcome value in that state depended solely on the variables $\mathbf{X}$. To ensure that this independence holds, we extend the weak sufficiency definition by additionally requiring that the observed state fall within the invariant preimage $IVP(\mathbf{X}; \mathbf{Y})$ for the event $\mathbf{X} = \mathbf{x}$ to be deemed as causally sufficient.

Definition 4.3 (Sufficient Judgement)**.** For an observed state $\mathbf{s}^*$ and set of invariant preimages $\mathcal{I}(\mathbf{Y})$, an event $\mathbf{X} = \mathbf{x}$ is a *Sufficient Judgement* for the outcome $\mathbf{Y} = \mathbf{y}$ if it is weakly sufficient for $\mathbf{Y} = \mathbf{y}$ and $\mathbf{s}^* \in IVP(\mathbf{X}; \mathbf{Y})$.

Under this definition, each actual cause is associated with a context-specific independence property specified by its corresponding invariant preimage.

4.3 Causal Necessity

Necessity is typically treated as a violation of sufficiency in the absence of the candidate actual cause. In the language of SCMs, this is given by the existence of at least one alternative event $\mathbf{X} = \mathbf{x}'$ that can result in a different outcome $\mathbf{Y} \neq \mathbf{y}$, thereby not being sufficient for the actual outcome $\mathbf{Y} = \mathbf{y}$.

Definition 4.4 (Contrastive Necessity)**.** For an observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is *necessary* for the outcome $\mathbf{Y} = \mathbf{y}$ under the witness set $\mathbf{W}$ if there exists an alternative assignment $\mathbf{x}' \in \mathcal{R}(\mathbf{X})$ such that $(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}$.

In practice, this condition can be extended such that an event is necessary only if a change in $\mathbf{X}$ results in a different outcome in at least α percent of states, instead of simply verifying that one such state exists. We refer to this as α-necessity and its corresponding sufficiency condition as α-sufficency. We expand on their definitions and properties in Appendix G, and this is a direction for future work, but for the rest of this section, we use weak sufficiency and contrastive necessity to outline our approach.

4.4 Minimality

In the framework for actual causation given in Definition 3.1, AC3 enforces minimality and the elimination of redundant causes. This is traditionally viewed as a local property of the observed state. On the other hand, minimizing the number of variables in an invariant-set actual-cause judgment also implies minimizing the number of variables corresponding to the IVP that contains the observed state. However, AC3 must hold for any observed state where we want to find actual-cause judgments, and this depends on how states are assigned to IVPs. Hence, minimality is a global property of the set of IVPs.

To formalize global minimality, we introduce the following notation. Each invariant preimage in $\mathcal{I}(\mathbf{Y})$ is associated with a unique binary vector of length $|\mathbf{S}|$, where each element corresponds to a variable in $\mathbf{S}$. The binary vector corresponding to $IVP(\mathbf{X};\mathbf{Y})$ will have a value of one in the indices corresponding to variables in $\mathbf{X}$. We refer to $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$ as a binary-subset pair. Minimality can thereafter be represented using a cost function as follows:

Definition 4.5 (Minimality of binary-subset pair)**.** A set of invariant preimages $\mathcal{I}(\mathbf{Y}) = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_K\}$ and corresponding binary vectors $\mathcal{B}(\mathbf{Y}) = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_K\}$ are minimal if:

$$(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y})) = \underset{\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y})}{\arg\min} \sum_{k=1}^{K} |\mathcal{I}_k||\mathbf{b}_k| \tag{2}$$

We can recover AC3 by showing that if the binary-subset pair $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$ satisfies the minimality criterion in Equation 2 then any event $\mathbf{X} = \mathbf{x}$ that satisfies AC1 and Definitions 4.3 and 4.4 for the outcome $\mathbf{Y} = \mathbf{y}$ must also be minimal.

Theorem 4.6. *For an observed state $\mathbf{s}^*$ and binary-subset pair $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$, if the event $\mathbf{X} = \mathbf{x}$ satisfies AC1, sufficient judgement, and contrastive necessity for the outcome $\mathbf{Y} = \mathbf{y}$ under witness set $\mathbf{W}$, and $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$ is minimal, then there cannot exist a smaller set of variables $\mathbf{Z} \subset \mathbf{X}$ such that $\mathbf{Z} = \mathbf{z}$ also satisfies AC1, sufficient judgement, and contrastive necessity for $\mathbf{Y} = \mathbf{y}$, where $\mathbf{z}$ denotes the values of variables $\mathbf{Z}$ in the assignment $\mathbf{x}$.*

4.5 Actual Causation

By substituting the definitions of sufficient judgement (Definition 4.3), contrastive necessity (Definition 4.4), and minimality (Definition 4.5) in the general framework for actual causation (Definition 3.1), we obtain the following definition of actual-cause judgment, referred to as *invariant-set actual-cause judgment (IAJ)*.

Definition 4.7 (Invariant-Set Actual-Cause Judgment)**.** For a given model M, observed state $\mathbf{s}^*$, and binary-subset pair $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$, the event $\mathbf{X} = \mathbf{x}$ is a invariant-set actual-cause judgment of the outcome $\mathbf{Y} = \mathbf{y}$ if:

AC1 (*Factual*): $\mathbf{X} = \mathbf{x}$ and outcome $\mathbf{Y} = \mathbf{y}$ actually happened, i.e. $\mathbf{s}^*_{\mathbf{X}} = \mathbf{x}$ and $\mathbf{s}^*_{\mathbf{Y}} = \mathbf{y}$.

AC2(a) (*Necessity*): There exists a counterfactual event $\mathbf{x}' \in \mathcal{R}(\mathbf{X})$ and witness set $\mathbf{W}$ such that $(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}$.

AC2(b) (*Sufficiency*): The observed state $\mathbf{s}^*$ lies in the invariant preimage $IVP(\mathbf{X};\mathbf{Y})$

AC3 (*Minimality*): The binary-subset pair $(\mathcal{B}(\mathbf{Y}),\mathcal{I}(\mathbf{Y}))$ minimizes the cost $\mathcal{L} = \sum_{k=1}^{K} |\mathcal{I}_k||\mathbf{b}_k|$

For any definition of actual cause, there may be multiple valid actual causes that satisfy the definition for a given outcome and observed state. This property is also present in IAJ. The IAJ definition makes this explicit by including the minimal binary-subset pair that was used to determine actual-cause judgments. For a given model M, there can be multiple partitionings of the state space that satisfy Equations 1 and 2. Appendix N provides examples of this, showing multiple valid actual-cause judgments identified for each observed state, with each assignment of actual-cause judgments associated with a binary-subset pair.

Like definitions of actual cause, actual-cause judgments are minimal sets of events that are necessary and sufficient for the observed outcome, but are additionally required to satisfy a globally minimized context-specific independence property. Weak sufficiency is dropped from the definition since it is trivially satisfied if

AC1 is satisfied. We show in the next section how the necessity, sufficiency, and minimality properties in IAJ capture the normality intuition outlined in Section 2.

4.6 Normality and the Queen of England

The 1-D Block Pushing and Queen of England examples in Section 2 highlight scenarios where the absence of a rare event is a valid actual cause despite conflicting with human judgments. Empirical studies have shown that human judgments about actual causation can be influenced significantly by how they perceive observed events as normal, abnormal, or rare (Hitchcock & Knobe, 2009; Knobe & Fraser, 2008). As a result, prior work has suggested incorporating *normality* (Icard et al., 2017) into the theory of actual causation.

We focus on two specific issues of normality in this paper. First, a rare event is considered a less significant cause when it does not actually happen. This effect has been studied in various AI problems such as explanation Karimi et al. (2022) and planning Srivastava et al. (2013). Second, when a rare event actually happens, it is perceived as a more significant cause than a routine event (Icard et al., 2017). This effect is known as *abnormal inflation* (Icard et al., 2017), and is extensively studied in the philosophy literature (Halpern & Hitchcock, 2015; Blanchard & Schaffer, 2017).

IAJ addresses both problems using the invariant preimages. First, a given $\mathbf{X} = \mathbf{x}$ is considered an actual-cause judgment only when the observed state falls within $IVP(\mathbf{X}; \mathbf{Y})$, indicating that the outcome is contextually independent of all variables not in $\mathbf{X}$. This ensures that events that could not have influenced the actual outcome are not included as actual-cause judgments. Second, the global minimality objective ensures that when a rare event occurs, it is included in the actual-cause judgment. Minimizing Equation 2 ensures that $IVP(\mathbf{X}; \mathbf{Y})$ has fewer states when $\mathbf{X}$ has more variables. When all states have equal likelihood, an implicit assumption in most prior work on actual causation, a smaller $IVP(\mathbf{X}; \mathbf{Y})$ means that the values of $\mathbf{X}$ are actual causes more rarely. Thus, when the observed state is in a smaller IVP, it indicates that a rare event occurred and was included in the actual cause. The IAJ formulation of minimality *implicitly assumes uniform probabilities over variable assignments* and can be extended to probabilistic domains by weighting states according to their likelihood, which we leave to future work.

4.7 Relationships between IAJ and Prior Definitions of Actual Causation

The Halpern-Pearl definitions of actual causation (Halpern & Pearl, 2005) are widely recognized in the literature, with *Modified HP* (Halpern, 2015) being the most recent definition. Modified HP can be understood as a special case of IAJ where the binary-subset pair is such that all states in the state space fall within a single invariant preimage $IVP(\mathbf{S}; \mathbf{Y})$. We can interpret this as a property of the environment where no context-specific independencies exist and the outcome always depends on all state variables, or just a choice of the set of invariant preimages $\mathcal{I}(\mathbf{Y})$ that disregards context-specific independencies.

Example 2.1 shows how Modified HP can be overly permissive. IAJ was designed to exclude state variables that did not contribute to the outcome and return a smaller set of actual causes. We show that the union of all Modified HP actual causes always contains at least as many variables as the union of all functional actual causes, with the proof in Appendix D.1

Theorem 4.8. *For an observed state* $\mathbf{s}^*$*, if* $\mathbf{X} = \mathbf{x}$ *is a invariant-set actual-cause judgment of* $\mathbf{Y} = \mathbf{y}$ *for any binary-subset pair* $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ *then the assignment of every causal variable* $\mathbf{X}_i = \mathbf{x}_i$ *in* $\mathbf{X} = \mathbf{x}$ *is contained in some Modified HP actual cause* $\mathbf{Z}_i = \mathbf{z}_i$

Direct sufficiency (Beckers, 2021) is a more stringent AC definition that requires the outcome $\mathbf{Y} = \mathbf{y}$ to always occur when the intervention $\mathbf{X} \leftarrow \mathbf{x}$ is applied, regardless of the values of all other variables $\mathbf{C}$. This would be satisfied if the observed state $\mathbf{s}^*$ lies within the invariant preimage $IVP(\mathbf{X}; \mathbf{Y})$. Invariant preimages are defined such that the invariance to $\mathbf{C}$ only needs to hold for a subset of the possible values of $\mathbf{C}$, whereas direct sufficiency requires this invariance to hold for all possible assignments of $\mathbf{C}$.

Theorem 4.9. *For an observed state* $\mathbf{s}^*$*, sufficient judgement and direct sufficiency are equivalent when the invariant preimage* $IVP(\mathbf{X}; \mathbf{Y})$ *that contains* $\mathbf{s}^*$ *shows complete invariance, i.e. all other states* $\mathbf{s}'$ *where* $\mathbf{s}'_{\mathbf{X}} = \mathbf{s}^*_{\mathbf{X}}$ *also lie within* $IVP(\mathbf{X}; \mathbf{Y})$*.*

While sufficient judgment is a statement about contextual invariance, direct sufficiency requires complete invariance, and Modified HP does not specify any invariance. IAJ can therefore be viewed as encompassing a spectrum between direct sufficiency and Modified HP. Appendices E and F expand further on the relationships between IAJ and existing definitions of actual causation. Furthermore, a deeper question might be whether token or specific causality is an "actual judgment" at all—that is, if it should be a binary judgment. While outside the scope of this work, we believe that IAJ lends itself well to softer judgments, and provide discussion of this in Appendix C

5 Discovering invariant-set actual-cause judgments

A key implication of the IAJ definition (Definition 4.7) is that the process of inferring actual-cause judgments boils down to mapping each observed state $\mathbf{s}^*$ to a binary vector $\mathbf{b} \in \mathcal{B}(\mathbf{Y})$. To ensure that Definition 4.7 is satisfied for any valid actual-cause judgment in all possible states, we verify that the binary-subset pair $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ satisfies the following properties:

P1 (*Invariance*): For any set of variables $\mathbf{X}$, every state in its corresponding partition $\mathcal{I}_{\mathbf{X}}$ satisfies Equation 1, implying that its resulting outcome is subset-invariant to the remaining variables $\mathbf{S} - \mathbf{X}$.

P2 (*Necessity*): If a state $\mathbf{s}$ lies within the partition $\mathcal{I}_{\mathbf{X}}$ then $\mathbf{X} = \mathbf{s}_{\mathbf{X}}$ satisfies contrastive necessity (Definition 4.4) with respect to the outcome at state $\mathbf{s}$.

P3 (*Minimality*): The binary-subset pair $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ minimizes the cost function in Equation 2.

5.1 Recovering Binary-Subset Pairs in Discrete Environments

To discover a binary-subset pair that satisfies the above conditions, we start with a naive approach: exhaustively enumerating all possible partitionings of the state space and assignments of binary vectors to these partitions. We then identify and return all binary-subset pairs that satisfy properties **P1**, **P2**, and **P3**. This exhaustive enumeration is infeasible by hand in all but the simplest of scenarios, so we algorithmically recover the binary subset pairs by progressively filtering binaries and state subsets. The specifics of this approach are described in Appendix M, with pseudocode in Algorithm 0.

5.2 Scaling to Continuous Domains via Joint Optimization

To scale to more complex domains, we learn a parameterized function that identifies actual-cause judgments at any given state $\mathbf{s}$ by mapping it to a corresponding binary vector. We represent this using a neural network $h_{\mathbf{Y}}(\mathbf{s}; \theta) : \mathcal{R}(\mathbf{S}) \to \{0, 1\}^{|\mathbf{S}|}$.[1] This exploits the fact that once we have a binary-subset pair that satisfies properties **P1**, **P2**, and **P3**, there is no need to explicitly store or enumerate the partitions as long as the correct binary vector is returned.

In practice, we rarely have access to a fully specified SCM to perform arbitrary interventions and evaluate the above properties, and instead have a dataset $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{y}_i)\}_{i=1}^{N}$ with observations and outcomes. We learn a model of the environment $f(\mathbf{s}, \mathbf{b}; \phi)$ which, given a state $\mathbf{s}$ and a binary vector $\mathbf{b}$, returns a value for the outcome $\mathbf{Y}$ that only depends on the state variables $\mathbf{S}_{\mathbf{b}}$ corresponding to the ones in $\mathbf{b}$. In practice, we implement this by using $\mathbf{b}$ to mask out all variables except for those in $\mathbf{S}_{\mathbf{b}}$.

Algorithm 1 describes our approach for jointly learning a model of the environment and mapping states to binary vectors. To enforce minimality (**P3**), the model $h_{\mathbf{Y}}(\mathbf{s}; \theta)$ is trained to minimize the length of the binary vectors corresponding to each state in the dataset. Since $h_{\mathbf{Y}}(\mathbf{s}; \theta)$ returns a single binary vector for every state, each binary vector corresponds to a unique partition of the state space. To enforce invariance (**P1**), we constrain the output of model $f(\mathbf{s}, \mathbf{b}; \phi)$ to match the true outcome $\mathbf{y}$, since this implies that only the variables $\mathbf{S}_{\mathbf{b}}$ were needed to obtain the outcome at state $\mathbf{s}$. Adding this as an optimization constraint

[1] Note that this makes the assumption of only a single actual-cause judgment in a particular state for a particular $\mathbf{X} = \mathbf{x}, \mathbf{Y} = \mathbf{x}$ because the network maps each pair to a unique value. While this assumption may not always hold, future work can extend this to outputting a set of explanations using an ensemble of models.

and using its Lagrangian dual form results in the following objective:

$$\min_{\theta,\phi} \sum_{i=1}^{N} \left(|h_{\mathbf{Y}}(\mathbf{s}^{(i)};\theta)| + \beta \| f(\mathbf{s}^{(i)}, \mathbf{b}^{(i)};\phi) - \mathbf{y}^{(i)} \| \right) \text{ where } \mathbf{b}^{(i)} = h_{\mathbf{Y}}(\mathbf{s}^{(i)};\theta) \tag{3}$$

Algorithm 1 Joint Optimization for Minimum Actual-Cause Judgment (JOA)

Input: Dataset $\mathcal{D} = \{(\mathbf{s}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{N}$
Output: state-binary mapping $h_{\mathbf{Y}}(\mathbf{s};\theta)$
repeat
  $\mathbf{b}^{(i)} = h_{\mathbf{Y}}(\mathbf{s}^{(i)};\theta)$ **for** $i \in \{1, ..., N\}$
  $\phi := \arg\min_{\phi} \sum_{i=1}^{N} \| f(\mathbf{s}^{(i)}, \mathbf{b}^{(i)};\phi) - \mathbf{y}^{(i)} \|$
  $\theta := \arg\min_{\theta} \sum_{i=1}^{N} \left(|h_{\mathbf{Y}}(\mathbf{s}^{(i)};\theta)| + \beta \| f(\mathbf{s}^{(i)}, \mathbf{b}^{(i)};\phi) - \mathbf{y}^{(i)} \| \right)$
until (θ, ϕ) converges

JOA thus learns to identify invariant-set actual-cause judgments by jointly optimizing the masked forward model f and state-binary mapping $h_{\mathbf{Y}}$. Appendix H contains additional details on the order-invariant neural network architecture used for our implementation of JOA and how it enforces causal necessity.

6 Experiments

In this section, we evaluate the accuracy of our approach in inferring actual-cause judgments in discrete and continuous environments. We assess the verdicts of actual-cause judgments on a set of classic examples from the actual causality literature, using exhaustive search to find exact solutions to the IAJ definition. We assess whether the verdicts from IAJ still match those of actual cause definitions on existing examples, while determining if they differ on examples, like the 1D pusher, where prior definitions provide unintuitive outcomes. We also evaluate JOA in a set of complex, continuous-valued domains and compare against learning algorithms that try to recover a set of state-specific causes for a particular effect. In this work, we assume access to a well-defined set of causal variables. In practice, identifying causal variables is integral for defining a good set of IVPs, and to some extent, this work is inspired by the notion that actual cause can result in unintuitive outcomes when dealing with object-factored settings. Recovering good variables from high-dimensional observations remains an ongoing area of research, which is complementary to this work.

6.1 Exhaustive Search in Discrete Domains

In the philosophy literature, it is common practice to evaluate definitions of actual cause using a set of examples with agreed-upon answers. We use exhaustive search to obtain invariant-set actual-cause judgments for the *Rock Throwing*, *Forest Fire*, *Halt and Charge*, and *Railroad Track* examples from Halpern (2016), *Gang Execution* example from Rosenberg & Glymour (2018), and the binary *AND*, *OR*, and *XOR* functions. Since none of these examples rely on context-specific independence, we would expect the results of IAJ to align with prior definitions.

Our approach obtained the agreed-upon answers on all of the above examples. For instance, *Rock Throwing* is a scenario where Suzy and Billy both throw a rocks at a bottle, but Suzy's rock reaches first and shatters the bottle. Billy's throw was also accurate and would have shattered the bottle had Suzy's rock not arrived first, but most prior work asserts that Suzy's throw is the only actual cause of the bottle shattering Halpern (2016). This scenario corresponds to the last row (1110) in Table 1(a), and the binary vector $\mathbf{b} = 1000$ indicates that our exhaustive search algorithm (Appendix M) identified Suzy's throw as the actual-cause judgment. The full set of tables is in Appendix N.

To verify that IAJ is consistent with the normality intuition outlined in Section 4.6, we run Algorithm 0 on a simplified 1D Mover environment based on Example 2.1, where existing methods would identify the obstacle as an actual cause. As shown in Table 1(b), our approach declares that the obstacle is an actual-cause judgment only in states where it could have impeded the mover.

State				Outcome	Binary
ST	SH	BT	BH	BoS	**b**
0	0	0	0	0	0001
0	0	1	1	1	0001
1	1	0	0	1	1000
1	1	1	0	1	1000

(a)

State		Outcome	Binary	State		Outcome	Binary
m	o	m'	**b**	m	o	m'	**b**
0	1	0	11	2	0	3	10
1	2	1	11	2	1	3	10
0	0	1	10	2	2	3	10
0	2	1	10	3	0	4	10
1	0	2	10	3	1	4	10
1	1	2	10	3	2	4	10

(b)

Table 1: Exhaustive search results. Binary vector **b** indicates the variables whose values are actual-cause judgments. **(a) Rock Throwing**: ST (Suzy Throws), SH (Suzy Hits), BT (Billy Throws), BH (Billy Hits), BoS (Bottle Shatters); full SCM is in Appendix N.8. **(b) 1D Mover**: $m' = m + 1$ if $o \neq m + 1$, otherwise $m' = m$, for variables m (mover position), o (obstacle position), m' (next mover position).

6.2 JOA in Continuous Domains

To assess the accuracy of JOA in inferring actual causes, we evaluate JOA in a set of continuous-valued *Random Vectors* domains with different causal graph structures, and in the RL domains *Mini-Breakout*, *2D Robot Pushing*, and *Spriteworld*.

Baselines: Since traditional actual causality methods require access to a perfect SCM and enumerating all counterfactual states, they are infeasible to evaluate in continuous and high-dimensional domains. However, several approaches in machine learning aim to discover *local causality*, *context-set specific independence*, *interactions* or other related concepts to identify state factors that actually influenced the outcome at a given state. These include *gradient thresholds* (**Grad**) (Wang et al., 2023), *neural attentive circuits* (**Attn**) (Weiss et al., 2022), *counterfactual testing* (**CF**) (Pitis et al., 2020) and *neural contextual decomposition* (**NCD**) (Hwang et al., 2023). NCD is a method for inferring context-set specific independence in graphical models. A modified version of NCD (**NCD-reg**) is used in Hwang et al. (2024a) and includes a notion of minimality through tuned regularization. Further details on these baselines are in Appendix K.

Random Vectors: Each domain is described by a causal DAG over a set of active variables $\mathcal{X} = \{X_1, \dots X_n, Y\}$ and passive variables $\tilde{\mathcal{X}} = \{\tilde{X}_1, \dots \tilde{X}_n, \tilde{X}_Y\}$, where each passive variable $\tilde{X}_t$ is a parent of its corresponding active variable X_t. All variables are d-dimensional. Each active variable X_t is a conditional function of its passive variable and its active variable parents, given by:

$$X_t = f(x_{pa}, \tilde{x}_t) = \begin{cases} f_P(\tilde{x}_t) & \text{if } g(x_{pa}, \tilde{x}_t) \leq \tau \\ f_A(x_{pa}, \tilde{x}_t) & \text{if } g(x_{pa}, \tilde{x}_t) > \tau \end{cases}$$

We use different linear and nonlinear functions for f_A, f_P, and g. The passive variable value $\tilde{x}_t$ is always an actual cause since it is included in both f_P and f_A, meaning $\tilde{x}_t$ always influences the value of X_t. However, the value of $g(x_{pa}, \tilde{x}_t)$ determines whether f_P or f_A is used to obtain the value of X_t. Hence, the values of the active variable parents may or may not be actual causes based on the values of x_{pa} and $\tilde{x}_t$ in the observed state. We generate Random Vectors domains for the following causal graphs (dropping the passive variables for simplicity):

- **1-in**: $X_{a,1} \to Y$
- **2-in**: $X_{a,1} \to Y, X_{a,2} \to Y$
- **3-in**: $X_{a,1} \to Y, X_{a,2} \to Y, X_{a,3} \to Y$
- **3-chain**: $X_{a,1} \to X_{a,2} \to Y$
- **3-m-in**: $[X_{a,1}, X_{a,2}, X_{a,3}] \to Y$
- **d-20**: $X_{a,1} \to Y$ with $d = 20$ instead of 4
- **τ-1**: $X_{a,1} \to Y$ with $< 10\%$ frequency of $X_{a,1}$ being an actual cause

Aside from **d-20**, all variables are 4-dimensional (in the latter, the variables are 20 dimensional). We generate $90k$ states for training and $10k$ states for the test set in each domain. Further details on the Random Vector domains are in Appendix I.

Env	JOA	Grad	Attn	CF	NCD	NCD-Reg
1-in-linear	$\mathbf{3.6 \pm 0.1}$	8.8 ± 3.6	48 ± 0.8	42 ± 0.9	55 ± 0.1	16 ± 9.3
2-in-linear	$\mathbf{2.6 \pm 2.0}$	27 ± 4.0	50 ± 1.1	42 ± 1.5	53 ± 2.1	21 ± 8.1
1-in-nonlinear	$\mathbf{3.2 \pm 0.1}$	20 ± 2.1	49 ± 0.9	46 ± 2.0	61 ± 0.1	13 ± 6.2
2-in-nonlinear	$\mathbf{4.0 \pm 1.0}$	43 ± 3.7	34 ± 3.7	35 ± 3.0	62 ± 2.2	25 ± 10.1
3-in	$\mathbf{4.5 \pm 1.9}$	41 ± 3.8	47 ± 3.7	44 ± 1.1	59 ± 11	42 ± 15.8
3-chain	$\mathbf{9.3 \pm 7.6}$	46 ± 3.2	46 ± 0.5	47 ± 0.5	55 ± 8.3	41 ± 1.8
3-m-in	$\mathbf{2.0 \pm 0.7}$	17 ± 3.5	50 ± 0.5	44 ± 0.3	56 ± 0.3	25 ± 10.5
d-20	$\mathbf{7.7 \pm 1.2}$	$\mathbf{8.0 \pm 0.4}$	43 ± 0.8	45 ± 1.1	47 ± 1.0	42 ± 1.1
τ-1	$\mathbf{0.2 \pm 0.0}$	2.1 ± 0.3	9.9 ± 0.8	7.7 ± 0.5	87 ± 1.2	4.0 ± 1.8
2D Pushing	$\mathbf{13 \pm 1.3}$	22 ± 1.1	39 ± 0.6	41 ± 1.2	58 ± 0.5	27 ± 3.6
Breakout	$\mathbf{3.6 \pm 0.2}$	$\mathbf{10 \pm 7.1}$	16 ± 6.5	35 ± 1.1	71 ± 0.6	22 ± 4.6
Spriteworld-2	$\mathbf{6.6 \pm 1.2}$	16 ± 0.5	31 ± 0.8	35 ± 0.4	71 ± 0.9	21 ± 1.8
Spriteworld-3	$\mathbf{11 \pm 1.7}$	21 ± 0.1	34 ± 0.4	39 ± 0.8	61 ± 0.7	18 ± 2.3
Spriteworld-6	$\mathbf{6.3 \pm 1.3}$	20 ± 0.2	12 ± 2.0	35 ± 0.0	44 ± 0.1	23 ± 1.5

Table 2: Error (%) and Std. Dev. in identifying ground truth actual-cause judgments, 5 seeds per evaluation.

RL Domains: To evaluate how well JOA scales to more realistic domains with complex dynamics, we use the *Mini-Breakout, 2D Robot Pushing, and Spriteworld* environments. In each domain, the simulator indicates when objects interact and can be used to obtain ground truth actual-cause judgments. Additional details on these domains and their causal graphs are in Appendix L.

Results: As shown in Table 2, JOA significantly outperforms all baselines in identifying actual-cause judgments across all domains. However, we also found that JOA could converge to local minima such as the binary vector of all zeros (no actual-cause judgments) or all ones (every causal variable is included in the actual-cause judgment) without careful hyperparameter tuning. Appendix J provides details on our hyperparameter tuning approach, along with training curves and a false-positive/negative assessment.

7 Conclusion

The *invariant-set actual-cause judgment (IAJ)* definition is a step towards bridging the theory of actual causation with the practical objective of scaling to complex environments. We show that IAJ yields a more concise set of actual-cause judgments compared to the actual cause sets of prior definitions. By formalizing established intuitions of normality through context-specific independence and global optimization, automated inference becomes a feasible strategy without requiring manually specified scores or human preferences. We demonstrate this using *joint optimization for minimum actual-cause judgment (JOA)*, showing that it accurately infers actual-cause judgments in continuous-valued and complex domains.

References

Natasha Alechina, Joseph Y Halpern, and Brian Logan. Causality, responsibility and blame in team plans. arXiv preprint arXiv:2005.10297, 2020.

Gadi Aleksandrowicz, Hana Chockler, Joseph Y Halpern, and Alexander Ivrii. The computational complexity of structure-based causality. Journal of Artificial Intelligence Research, 58:431–451, 2017.

Mark D Alicke. Culpable causation. Journal of personality and social psychology, 63(3):368, 1992.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. arXiv preprint arXiv:1909.12000, 2019.

Adam Bear and Joshua Knobe. Normality: Part descriptive, part prescriptive. cognition, 167:25–37, 2017.

Sander Beckers. Causal sufficiency and actual causation. Journal of Philosophical Logic, 50(6):1341–1374, June 2021. doi: 10.1007/s10992-021-09601-z. URL https://doi.org/10.1007/s10992-021-09601-z.

Sander Beckers. Causal explanations and XAI. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), Proceedings of the First Conference on Causal Learning and Reasoning, volume 177 of Proceedings of Machine Learning Research, pp. 90–109. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/beckers22a.html.

Sander Beckers and Joost Vennekens. Combining probabilistic, causal, and normative reasoning in cp-logic. In 2015 AAAI Spring Symposium Series, 2015.

Sander Beckers, Hana Chockler, and Joseph Halpern. A causal analysis of harm. Advances in Neural Information Processing Systems, 35:2365–2376, 2022.

Helen Beebee. Causing and nothingness. In L. A. Paul, E. J. Hall, and J. Collins (eds.), Causation and Counterfactuals, pp. 291–308. MIT Press, 2004.

Alexis Bellot, Kim Branson, and Mihaela van der Schaar. Consistency of mechanistic causal discovery in continuous-time using neural odes. arXiv preprint arXiv:2105.02522, 2021.

Leopoldo Bertossi, Jordan Li, Maximilian Schleich, Dan Suciu, and Zografoula Vagena. Causality-based explanation of classification outcomes. In Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning, pp. 1–10, 2020.

Abhijat Biswas, Badal Arun Pardhi, Caleb Chuck, Jarrett Holtz, Scott Niekum, Henny Admoni, and Alessandro Allievi. Gaze supervision for mitigating causal confusion in driving agents. In 2024 IEEE Intelligent Vehicles Symposium (IV), pp. 2331–2338. IEEE, 2024.

Thomas Blanchard and Jonathan Schaffer. Cause without default. In Helen Beebee, Christopher Hitchcock, and Huw Price (eds.), Making a Difference, pp. 175–214. Oxford University Press, 2017.

Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. arXiv preprint arXiv:1302.3562, 2013.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. Advances in Neural Information Processing Systems, 33:21865–21877, 2020.

Marc J Buehner and Patricia W Cheng. Causal induction: The power pc theory versus the rescorla-wagner model. In Proceedings of the nineteenth annual conference of the Cognitive Science Society, pp. 55–60. Erlbaum Hillsdale, NJ, 1997.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. arXiv preprint arXiv:1811.06272, 2018.

Erin Catto. Box2d. https://github.com/TelloViz/Box2D, 2020. Accessed: 2024-05-21.

Ivi Chatzi, Nina L. Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Counterfactual token generation in large language models. In Biwei Huang and Mathias Drton (eds.), Proceedings of the Fourth Conference on Causal Learning and Reasoning, volume 275 of Proceedings of Machine Learning Research, pp. 1291–1315. PMLR, 07–09 May 2025. URL https://proceedings.mlr.press/v275/chatzi25a.html.

Rohan Chitnis, Tom Silver, Beomjoon Kim, Leslie Kaelbling, and Tomas Lozano-Perez. Camps: Learning context-specific abstractions for efficient planning in factored mdps. In Conference on robot learning, pp. 64–79. PMLR, 2021.

Hana Chockler and Joseph Y Halpern. Responsibility and blame: A structural-model approach. Journal of Artificial Intelligence Research, 22:93–115, 2004.

Jongwook Choi, Sungtae Lee, Xinyu Wang, Sungryull Sohn, and Honglak Lee. Unsupervised object interaction learning with counterfactual dynamics models. In Workshop on Reincarnating Reinforcement Learning at ICLR 2023, 2023.

Caleb Chuck, Supawit Chockchowwat, and Scott Niekum. Hypothesis-driven skill discovery for hierarchical deep reinforcement learning. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5572–5579. IEEE, 2020.

Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger-causal hierarchical skill discovery. arXiv preprint arXiv:2306.09509, 2023.

Caleb Chuck, Fan Feng, Carl Qi, Chang Shi, Siddhant Agarwal, Amy Zhang, and Scott Niekum. Null counterfactual factor interactions for goal-conditioned reinforcement learning. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=2uPZ4aX1VV.

Jukka Corander, Antti Hyttinen, Juha Kontinen, Johan Pensar, and Jouko Väänänen. A logical approach to context-specific independence. Annals of Pure and Applied Logic, 170(9):975–992, 2019.

Paul Dagum, Adam Galper, Eric Horvitz, and Adam Seiver. Uncertain reasoning and forecasting. International Journal of Forecasting, 11(1):73–87, 1995.

Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. Advances in Neural Information Processing Systems, 35:26532–26548, 2022.

Fan Feng and Sara Magliacane. Learning dynamic attribute-factored world models for efficient multi-object reinforcement learning. arXiv preprint arXiv:2307.09205, 2023.

Jörg Frohberg and Frank Binder. Crass: A novel data set and benchmark to test counterfactual reasoning of large language models. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 2126–2140, 2022.

Tobias Gerstenberg. What would have happened? counterfactuals, hypotheticals and causal judgements. Philosophical Transactions of the Royal Society B: Biological Sciences, 377(1866), 2022.

Tobias Gerstenberg, Noah Goodman, David Lagnado, and Josh Tenenbaum. From counterfactual simulation to causal judgment. In Proceedings of the annual meeting of the cognitive science society, volume 36, 2014.

Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Josfhua B Tenenbaum. How, whether, why: Causal judgments as counterfactual contrasts. In Proceedings of the annual meeting of the cognitive science society, volume 37, 2015.

Tobias Gerstenberg, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. A counterfactual simulation model of causal judgments for physical events. Psychological review, 128(5):936, 2021.

Balint Gyevnar, Cheng Wang, Christopher G. Lucas, Shay B. Cohen, and Stefano V. Albrecht. Causal explanations for sequential decision-making in multi-agent systems. In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '24, pp. 771–779, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Abdul Mueed Hafiz, Shabir Ahmad Parah, and Rouf Ul Alam Bhat. Attention mechanisms and deep learning for machine vision: A survey of the state of the art. arXiv preprint arXiv:2106.07550, 2021.

Ned Hall. Two Concepts of Causation. In Causation and Counterfactuals. The MIT Press, 06 2004. ISBN 9780262270663. doi: 10.7551/mitpress/1752.003.0010. URL https://doi.org/10.7551/mitpress/1752.003.0010.

Joseph Y. Halpern. A modification of the halpern-pearl definition of causality. In Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pp. 3022–3033. AAAI Press, 2015. ISBN 9781577357384.

Joseph Y Halpern. Actual causality. MIT Press, 2016.

Joseph Y Halpern and Christopher Hitchcock. Graded causation and defaults. The British Journal for the Philosophy of Science, 2015.

Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. The British journal for the philosophy of science, 2005.

Paul Henne, Laura Niemi, Ángel Pinillos, Felipe De Brigard, and Joshua Knobe. A counterfactual explanation for the action effect in causal judgment. Cognition, 190:157–164, 2019.

Denis J Hilton and Ben R Slugoski. Knowledge-based causal attribution: The abnormal conditions focus model. Psychological review, 93(1):75, 1986.

Christopher Hitchcock. Prevention, preemption, and the principle of sufficient reason. The Philosophical Review, 116(4):495–532, 2007.

Christopher Hitchcock and Joshua Knobe. Cause and norm. The Journal of Philosophy, 106(11):587–612, 2009.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

Inwoo Hwang, Yunhyeok Kwak, Yeon-Ji Song, Byoung-Tak Zhang, and Sanghack Lee. On discovery of local independence over continuous variables via neural contextual decomposition. In Conference on Causal Learning and Reasoning, pp. 448–472. PMLR, 2023.

Inwoo Hwang, Yunhyeok Kwak, Suhyung Choi, Byoung-Tak Zhang, and Sanghack Lee. Quantized local independence discovery for fine-grained causal dynamics learning in reinforcement learning. International Conference on Machine Learning (ICML), 2024a.

Inwoo Hwang, Yunhyeok Kwak, Suhyung Choi, Byoung-Tak Zhang, and Sanghack Lee. Fine-grained causal dynamics learning with quantization for improving robustness in reinforcement learning. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org, 2024b.

Thomas F. Icard, Jonathan F. Kominsky, and Joshua Knobe. Normality and actual causal strength. Cognition, 161:80–93, 2017. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2017.01.010. URL https://www.sciencedirect.com/science/article/pii/S0010027717300100.

Fateme Jamshidi, Sina Akbari, and Negar Kiyavash. Causal imitability under context-specific independence relations. Advances in Neural Information Processing Systems, 36:26810–26830, 2023.

Daniel Kahneman and Dale T Miller. Norm theory: Comparing reality to its alternatives. Psychological review, 93(2):136, 1986.

Daniel Kahneman and Amos Tversky. The simulation heuristic. Judgment under uncertainty, pp. 201–208, 1982.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. ACM Comput. Surv., 55(5), dec 2022. ISSN 0360-0300. doi: 10.1145/3527848. URL https://doi.org/10.1145/3527848.

Harold H Kelley. The processes of causal attribution. American psychologist, 28(2):107, 1973.

Harold H Kelley and John L Michela. Attribution theory and research. Annual review of psychology, 31(1): 457–501, 1980.

Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. Transactions on Machine Learning Research, 2023.

Lara Kirfel and David Lagnado. Causal judgments about atypical actions are influenced by agents' epistemic states. Cognition, 212:104721, 2021.

Lara Kirfel, Thomas Icard, and Tobias Gerstenberg. Inference from explanation. Journal of Experimental Psychology: General, 151(7):1481, 2022.

Joshua Knobe and Ben Fraser. Causal judgment and moral judgment: Two experiments. In Walter Sinnott-Armstrong (ed.), Moral Psychology. MIT Press, 2008.

Joshua Knobe and Scott Shapiro. Proximate cause explained. The University of Chicago Law Review, 88(1): 165–236, 2021.

Jonathan F Kominsky and Jonathan Phillips. Immoral professors and malfunctioning tools: Counterfactual relevance accounts explain the effect of norm violations on causal selection. Cognitive science, 43(11): e12792, 2019.

Jonathan F Kominsky, Jonathan Phillips, Tobias Gerstenberg, David Lagnado, and Joshua Knobe. Causal superseding. Cognition, 137:196–209, 2015.

Anna Kun and Bernard Weiner. Necessary versus sufficient causal schemata for success and failure. Journal of Research in Personality, 7(3):197–207, 1973.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality, 2023.

David A Lagnado and Tobias Gerstenberg. Causation in legal and moral reasoning. Causality in Cognition Lab at Stanford, UNCORRECTED PROOF – FIRSTPROOFS, 2017.

Asmae Lamsaf, Rui Carrilho, João C Neves, and Hugo Proença. Causality, machine learning, and feature selection: a survey. Sensors, 25(8):2373, 2025.

Manuele Leonelli and Gherardo Varando. Context-specific causal discovery for categorical data using staged trees. In International conference on artificial intelligence and statistics, pp. 8871–8888. PMLR, 2023.

Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. Advances in Neural Information Processing Systems, 35:11828–11841, 2022.

Jonathan Livengood and Edourd Machery. The folk probably don't think what you think they think: Experiments on causation by absence. Midwest Studies In Philosophy, 31(1):107–127, 2007. doi: https://doi.org/10.1111/j.1475-4975.2007.00150.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-4975.2007.00150.x.

Chaochao Lu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for generalization in imitation and reinforcement learning. In ICLR2022 workshop on the elements of reasoning: Objects, structure and causality, 2022.

Ni Y Lu, Kun Zhang, and Changhe Yuan. Improving causal discovery by optimal bayesian network learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 8741–8748, 2021.

Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pp. 2493–2500, 2020.

David R Mandel. Effect of counterfactual and factual thinking on causal judgements. Thinking & Reasoning, 9(3):245–265, 2003.

Sarah McGrath. Causation by omission: A dilemma. Philosophical Studies, 123(1-2):125–148, 2005. doi: 10.1007/s11098-004-5216-z.

Alexandra Meliou, Wolfgang Gatterbauer, Katherine F Moore, and Dan Suciu. The complexity of causality and responsibility for query answers and non-answers. arXiv preprint arXiv:1009.2021, 2010.

Peter Menzies et al. Causation in context. Causation, physics, and the constitution of reality, pp. 191–223, 2007.

Adam Morris, Jonathan Phillips, Thomas Icard, Joshua Knobe, Tobias Gerstenberg, and Fiery Cushman. Judgments of actual causation approximate the effectiveness of interventions. Psy ArXiv) doi, 10, 2018.

Samer B Nashed, Saaduddin Mahmud, Claudia V Goldman, and Shlomo Zilberstein. Causal explanations for sequential decision making under uncertainty. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, pp. 2307–2309, 2023.

Samer B Nashed, Saaduddin Mahmud, Claudia V Goldman, and Shlomo Zilberstein. Causal explanations for sequential decision making. Journal of Artificial Intelligence Research, 83, 2025.

Jennifer Neville and David Jensen. Relational dependency networks. Journal of Machine Learning Research, 8(3), 2007.

Henrik Nyman, Johan Pensar, and Jukka Corander. Stratified gaussian graphical models. Communications in Statistics-Theory and Methods, 46(11):5556–5578, 2017.

Judea Pearl. Causality. Cambridge University Press, New York, 2000.

Judea Pearl. Probabilities of causation: three counterfactual interpretations and their identification. In Probabilistic and causal inference: the works of Judea Pearl, pp. 317–372. 2022.

Johan Pensar, Henrik Nyman, Jarno Lintusaari, and Jukka Corander. The role of local partial independence in learning of bayesian networks. International journal of approximate reasoning, 69:91–105, 2016.

Johan Pensar, J Kohonen, and Jukka Corander. Exact bayesian learning of partition directed acyclic graphs. In Journal of Physics: Conference Series, volume 1036, pp. 012017. IOP Publishing, 2018.

José C Perales and David R Shanks. Normative and descriptive accounts of the influence of power and contingency on causal judgement. The Quarterly Journal of Experimental Psychology Section A, 56(6): 977–1007, 2003.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. Advances in Neural Information Processing Systems, 33:3976–3990, 2020.

Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. Advances in Neural Information Processing Systems, 35:18143–18156, 2022.

David Poole and Nevin Lianwen Zhang. Exploiting contextual independence in probabilistic inference. Journal of Artificial Intelligence Research, 18:263–313, 2003.

Tadeg Quillien. When do we think that x caused y? Cognition, 205:104410, 2020.

Tadeg Quillien and Christopher G Lucas. Counterfactuals and the logic of causal selection. Psychological Review, 2023.

Tadeg Quillien and Christopher G Lucas. Counterfactuals and the logic of causal selection. Psychological Review, 131(5):1208, 2024.

Robert A Rescorla. A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. Classical conditioning, Current research and theory, 2:64–69, 1972.

Jonathan Richens, Rory Beard, and Daniel H. Thompson. Counterfactual harm. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 36350–36365. Curran Associates, Inc., 2022.

Ian Rosenberg and Clark Glymour. Review of Joseph Halpern, Actual Causality. PhilSciArchive, 2018.

Jana Samland and Michael R Waldmann. How prescriptive norms influence causal inferences. Cognition, 156:164–176, 2016.

Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious exploration via structured world models yields zero-shot object manipulation. Advances in Neural Information Processing Systems, 35:24170–24183, 2022.

Jonathan Schaffer. Contrastive causation. The Philosophical Review, 114(3):327–358, 2005.

Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. Advances in Neural Information Processing Systems, 34:22905–22918, 2021.

David R Shanks, Susan M Pearson, and Anthony Dickinson. Temporal contiguity and the judgement of causality by human subjects. The Quarterly Journal of Experimental Psychology, 41(2):139–159, 1989.

Scott Shapiro and Joshua Knobe. Proximate cause explained: An essay in experimental jurisprudence.

Yujia Shen, Arthur Choi, and Adnan Darwiche. A new perspective on learning context-specific independence. In International Conference on Probabilistic Graphical Models, pp. 425–436. PMLR, 2020.

Siddharth Srivastava, Xiang Cheng, Stuart J. Russell, and Avi Pfeffer. First-order open-universe pomdps: Formulation and algorithms. In Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2013. URL `https://api.semanticscholar.org/CorpusID:18456943`.

Eric V Strobl. Causal discovery with a mixture of dags. Machine Learning, pp. 1–25, 2022.

Santtu Tikka, Antti Hyttinen, and Juha Karvanen. Identifying causal effects via context-specific independence relations. Advances in neural information processing systems, 32, 2019.

Stelios Triantafyllou, Adish Singla, and Goran Radanovic. Actual causality and responsibility attribution in decentralized partially observable markov decision processes. In Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society, pp. 739–752, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. arXiv preprint arXiv:2206.13452, 2022.

Zizhao Wang, Jiaheng Hu, Peter Stone, and Roberto Martin-Martin. Elden: Exploration via local dependencies. In Advances in Neural Information Processing Systems (NeurIPS), 2023.

Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. Skild: Unsupervised skill discovery guided by factor interactions. Advances in Neural Information Processing Systems, 37:77748–77776, 2024a.

Zizhao Wang, Caroline Wang, Xuesu Xiao, Yuke Zhu, and Peter Stone. Building minimal and reusable causal state abstractions for reinforcement learning. In Proceedings of the aaai conference on artificial intelligence, volume 38, pp. 15778–15786, 2024b.

Zizhao Wang, Kaixin Wang, Li Zhao, Peter Stone, and Jiang Bian. Dyn-o: Building structured world models with object-centric representations. Advances in Neural Information Processing Systems, 38:121788–121812, 2026.

Nicholas Watters, Loic Matthey, Sebastian Borgeaud, Rishabh Kabra, and Alexander Lerchner. Spriteworld: A flexible, configurable reinforcement learning environment. https://github.com/deepmind/spriteworld/, 2019. URL `https://github.com/deepmind/spriteworld/`.

Martin Weiss, Nasim Rahaman, Francesco Locatello, Chris Pal, Yoshua Bengio, Bernhard Schölkopf, Erran Li Li, and Nicolas Ballas. Neural attentive circuits. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 7741–7754. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/32f227c41a0b4e36f65bebb4aeda94a2-Paper-Conference.pdf.

Peter A White. Causal judgement as evaluation of evidence: The use of confirmatory and disconfirmatory information. The Quarterly Journal of Experimental Psychology Section A, 56(3):491–513, 2003.

Pascale Willemsen and Lara Kirfel. Recent empirical work on the relationship between causal judgements and norms. Philosophy Compass, 14(1):e12562, 2019.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. arXiv preprint arXiv:2310.06114, 2023.

Kun Zhang, Biwei Huang, Jiji Zhang, Clark Glymour, and Bernhard Schölkopf. Causal discovery from non-stationary/heterogeneous data: Skeleton estimation and orientation determination. In IJCAI: Proceedings of the Conference, volume 2017, pp. 1347. NIH Public Access, 2017.

Fan Zhu. A new halpern-pearl definition of actual causality by appealing to the default world. Axiomathes, 32(Suppl 2):453–472, 2022.

Yuan Zou, Johan Pensar, and Teemu Roos. Representing local structure in bayesian networks by boolean functions. Pattern Recognition Letters, 95:73–77, 2017.

A Distinction Between Actual Cause and Actual Cause Judgment

In this work, we make a novel distinction between what we describe as actual-cause judgment and the identification of actual causes (Halpern & Pearl, 2005; Halpern, 2015; Beckers, 2021). This is for three reasons: first, we want to distinguish a tool that selects a subset of actual causes from the exhaustive set of actual causes (which can be overly permissive), without removing the underlying philosophical depth of the concept of actual cause. Second, because the ability to select a subset of the explanations can be powerful for machine learning models (Pitis et al., 2022; Chuck et al., 2025; Wang et al., 2026), where leveraging properties of the dynamics—the fact that certain causal factors are only relevant in a subset of states, can improve the generalization capabilities of the model (Lu et al., 2022; Li et al., 2022). Third, we believe that the term causal judgement, drawn from psychology, is a relevant analogy in this way: humans make causal assumptions about the world, which, while they are not always complete, are able to simplify their decisions in a way that aids decision-making (Gerstenberg et al., 2015; Willemsen & Kirfel, 2019). Drawing from the classic "Queen of England example (Schaffer, 2005), a person making a judgment about why their plants died might have to think about all kinds of explanations, from a full model of the weather systems to the every going-on of their neighbors, before deciding how to better care for their garden. Or they might simplify their problem to simply a question of whether they arranged for their plants to be watered. This same kind of simplifying capability is the one that we want to highlight and provide a formal definition for, while drawing from the closely related subject of actual cause.

Thus, in this work, we use *actual-cause judgments* to describe the quantity to be assessed, that subset of actual causes which we hope have the relevant properties. For these kinds of judgements, *invariant-set actual-cause judgment* describes the specific formal definition in causal language which we introduce in this work. We believe other formalisms for actual-cause judgments could provide better descriptions than invariant-set actual-cause judgment, and hope that our description will help spur future work in this subject. In this work, we hope to bring to bear the properties identified in causality and especially actual causality to the kinds of reasoning used in other judgements of relevant features, whether from sequential decisionmaking (Wang et al., 2023; Biswas et al., 2024; Wang et al., 2024a) or model learning (Lu et al., 2022; Li et al., 2022; Lamsaf et al., 2025), especially by providing JOA, which offers a computable algorithm for estimating actual-cause judgments.

However, we recognize that the term "causal judgement" as used in literature is primarily a psychological rather than a philosophical one (Schaffer, 2005; Gerstenberg et al., 2015; Icard et al., 2017; Henne et al., 2019). From a philosophical perspective, there does not appear to be a distinction between "identifying and actual cause" or "$\mathbf{X}$ is an actual cause of $\mathbf{Y}$" (Halpern & Pearl, 2005; Halpern, 2016). This is because actual causes are an objective reality of the causal model, rather than something determined or *judged* by a person. By contrast, in literature, there does not appear to be an accepted definition for causal judgement, and the term is often used interchangeably with "causal attribution " (Kelley, 1973; Kelley & Michela, 1980) or even "causal schmata." (Kun & Weiner, 1973) This is also reasonable from the psychological perspective, where what a person determines to be the cause of an outcome is more important than some objective quantity. Thus, we use the term actual-cause judgment as a bridge for this gap, where we are interested in the set of causal variables that are a useful subset for capturing the underlying statistical property of the dynamics, and so can be judged as the "meaningful causes." We acknowledge that this adds complexity by adding a new distinction for assessing actual causes or causal judgments, but believe this distinction is valuable for characterizing the spectrum of cause and effect.

B Related Work

Our work touches on a wide variety of prior research related to actual causality, context-specific independence, normality, and sequential decision-making. We highlight some of this work below.

B.1 Analogous Concepts in Causal Inference

The problem of an overdetermined set of actual causes, such as the obstacle being an actual cause in Example 2.1 relates to the classic Queen of England problem: *the queen of England's not watering my flowers caused my flowers to wilt* is contrary to most human judgments, but could technically be correct depending on the definition of actual cause. Schaffer (2005) connects this problem to the ambiguity between causation and non-causation, i.e., the non-occurrence of events being causes. This idea is related to *normality* (Halpern & Hitchcock, 2015), which has been observed in human behavior (Icard et al., 2017; Kominsky et al., 2015; Morris et al., 2018) and formally expanded on (Blanchard & Schaffer, 2017). Normality has also been incorporated in causal probabilistic logic (Beckers & Vennekens, 2015), but rarely in continuous domains or for inference with learned models and relationships with statistical correlation in counterfactual space (Quillien, 2020; Quillien & Lucas, 2023; 2024; Kirfel & Lagnado, 2021). However, invariant-set actual-cause judgment extends these definitions by applying context-specific independence and set-partitioning minimality, which formalizes notions of prescriptive causes with statistical correlation. In particular, IAJs represent normality using the invariant preimage, a form of set-based statistical normality.

Actual causality has well-established relationships to abnormality (Hilton & Slugoski, 1986), which is built on the basic causal assumption of describing an actual cause as present *when present and absent when absent* (Kelley & Michela, 1980). Nonetheless, the subject of normality has remained challenging to formalize, because human intuitions appear to be some combination of prescriptive (something is the cause because it should have been some way) and statistical (something is the cause because it is unlikely) (Hitchcock, 2007; Menzies et al., 2007; Hitchcock & Knobe, 2009). In some cases, prescriptive norms are described according to moral or legal reasoning (Kominsky & Phillips, 2019; Lagnado & Gerstenberg, 2017; Knobe & Shapiro, 2021; Shapiro & Knobe), and in other cases, in terms of mechanics (Bear & Knobe, 2017; Samland & Waldmann, 2016). In cases such as these, there are often few solutions more elegant than simply constructing a heuristic gradation of values (Halpern & Hitchcock, 2015).

Outside of the context of normality, the concept of the invariant preimage and identifying subsets of the state space where specific invariances exist is analogous to representing and learning context-specific independencies (Neville & Jensen, 2007), which has been explored in prior work on probabilistic graphical models (Shen et al., 2020; Boutilier et al., 2013). Mixtures of DAG representations (Strobl, 2022), especially using dynamic networks (Dagum et al., 1995; Zhang et al., 2017) frame the invariance using the generative DAG model. Alternatively, actual causal explanations include leveraging shapely values (Nashed et al., 2025) and are modified to handle isomorphism (Zhu, 2022). While actual causality introduces additional constraints related

to necessity, Algorithm 1 bears some resemblance to methods modeling mixture distributions (Bellot et al., 2021; Lu et al., 2021).

B.2 Causal Judgements

In this work, we describe actual-cause judgments as a way to disentangle judgments about the cause of an effect from the philosophical question of simply what could qualify as an actual cause. In Psychology, causal judgments are used to describe the observed human judgments made about causes for an effect, and their corresponding conditions (Kun & Weiner, 1973; Shanks et al., 1989), along with other terms such as causal attribution (Kelley, 1973) or causal culpability (Alicke, 1992). Nonetheless, it appears the dominant modern description is causal judgment, hence inspiring the terminology taken in this work.

As causal judgments serve as a term to describe the psychological phenomena of attributing an actual cause to an event, it is intertwined with the concepts of actual cause. In particular, causal judgments have been observed to be affected by: covariance and the presence of multiple explanations (Kelley, 1973), Necessary and sufficient evidence (Kun & Weiner, 1973; Alicke, 1992; Pearl, 2022), proximity in time between cause and effect (Shanks et al., 1989), distinctions between action and inaction (Henne et al., 2019), and Normality (Willemsen & Kirfel, 2019). These causal judgment models are then assessed in psychological research by constructing questionnaires with slight modifications to identify human models of causality. In this work, however, we are concerned with constructing a synthetic model of causality that can leverage invariance to identify specific causal events, like the interactions of objects or causal variables.

Normality in causal judgments offers likely the closest link to actual-cause judgments. Some of the forms of normality in the psychological literature (Kahneman & Miller, 1986) include statistical or injunctive norms, action and omission, and normative, pragmatic or responsibility views of causality (Willemsen & Kirfel, 2019). These works have build statistical models of normality, include those built on probability power (Buehner & Cheng, 1997; Perales & Shanks, 2003), additive probability (Rescorla, 1972), additive evidence (White, 2003; Kirfel et al., 2022), and counterfactual simulation (Kahneman & Tversky, 1982; Mandel, 2003; Gerstenberg et al., 2021; Gerstenberg, 2022). IAJ carries the closest link to counterfactual simulation, but rather than simulating over a single event, simulates over the set of the invariant preimage.

B.3 Context Specific Independence

Causal relationships that are context-specific have been previously studied in the literature on causal graphical models, where they are formalized using various concepts such as context-specific independence (Poole & Zhang, 2003; Boutilier et al., 2013), partial conditional independence (Pensar et al., 2016), and context-set specific independence (Hwang et al., 2023; 2024b). In these settings, sets of variable assignments in which particular independence relationships hold are referred to as *domains* or *context sets*. The concept of invariant preimage is analogous to these notions, as it represents regions of the state space where the outcome is invariant to certain inputs. A key distinction between invariant preimages and context-specific independence (CSI) is that the former asserts assumes invariance only within a subset of states while the latter assumes complete independence when certain variables are fixed to specific values, though some work (Boutilier et al., 2013; Hwang et al., 2023) has explored relaxing this constraint.

CSI addresses a problem similar to that of actual causality; it describes the contextually *independent* parents in a particular state, while actual causality describes what can intuitively be interpreted as the contextually *dependent* parents of an outcome. Actual causation, and by extension actual-cause judgments emphasizes what *actually happened* and not just any context, which is a requirement for the counterfactual in actual causation to make sense. Additionally, CSI aims to encompass the spectrum between partial conditional independence and marginal independence. The former describes an independence relationship that holds for a single assignment of the other variables, while the latter describes an independence relationship that holds regardless of the values of other variables. Similarly, Theorem 4.9 describes how functional sufficiency captures the spectrum between direct sufficiency and weak sufficiency. By synthesizing the concepts of necessity, sufficiency, and witness sets from actual causation with the notions of CSI and context sets as state-space partitions, this work provides a novel contribution to both fields.

The complexity of identifying context-specific independence is NP-hard (Corander et al., 2019), as is that of identifying blame and responsibility (Aleksandrowicz et al., 2017) with actual causation. In practice, context-specific independence has been inferred using staged trees (Leonelli & Varando, 2023), Bayesian networks with Boolean functions (Zou et al., 2017), weighted adjacency matrices (Brouillard et al., 2020) and stratified Gaussian models (Nyman et al., 2017). Additionally, the subset partitioning for defining context sets in CSI is exactly formulated in *local partial context-specific independence* (LP-CSI) (Pensar et al., 2016), and subsequent work has used Bayesian methods (Pensar et al., 2018) and logical graph structures (Tikka et al., 2019) to recover these partitions. LP-CSI captures the notion of the invariant preimage and tries to recover the set of invariant parents for different states. Though an analogy can be drawn between Bayesian methods and minimality (AC3) since they regularize the model, they do not capture the notion of necessity or the witness set.

JOA (Algorithm 1) can be seen as a function approximation method for learning these partitions that incorporates necessity (AC2a) in the structure of the function approximator. This is especially relevant in similarities to Neural Contextual Decomposition (NCD) (Hwang et al., 2023), which utilizes a forward model masked by outputs from a Gumbel-Bernoulli binary variable. However, NCD does not use joint optimization, regularization, or order-invariant structure to infer LP-CSI graphs. Another work based on NCD that shares many similarities to JOA is Fine-Grained Causal Dynamics Learning (FCDL) (Hwang et al., 2023; 2024a;b), which identifies locally invariant components. FCDL also employs a regularization objective that enforces minimality, but restricts the form of the local invariance function $h_{\mathbf{Y}}$ with a quantized codebook. Unlike these approaches, JOA uses an order-invariant structure and adaptive regularization scheme to capture the necessity condition in actual causation, which is not a requirement for CSI.

B.4 Causal Explanations and Decision Making

Generating causal explanations is one of the most intuitive applications of actual causality (Beckers, 2022) and a recurring focus of prior work, extending to problem settings such as classification (Bertossi et al., 2020), answering queries (Meliou et al., 2010), jurisprudence (Shapiro & Knobe), and language models (Frohberg & Binder, 2022; Kıcıman et al., 2023; Kiciman et al., 2023; Chatzi et al., 2025). Language analysis has, in particular, leveraged actual causation to describe the causal links between evidence and specific outcomes using models of internal state (Chatzi et al., 2025). For RL and sequential decisionmaking domains, past work has focused on explaining the decisions of RL policies (Nashed et al., 2023), imitation learning (Jamshidi et al., 2023) evaluating the degree of responsibility of particular decisions (Triantafyllou et al., 2022) and team games (Alechina et al., 2020; Nashed et al., 2025), decentralized POMDPs (Gyevnar et al., 2024) and studying human participants' explanations of the behavior of RL agents (Madumal et al., 2020). In general, this work draws from but does not directly relate to explanation, though there are several opportunities in this area for future work.

Actual cause judgments have many promising application directions in RL (Buesing et al., 2018), by utilizing state-specific causal judgments in decision-making. Several methods in RL have described this problem as *interaction identification* (Yang et al., 2023), *local causality* (Pitis et al., 2020), *controllability* (Seitzer et al., 2021) and *counterfactuals*. Actual cause offers a more principled way to unify these approaches. State-specific causes have been inferred in RL using gradients (Wang et al., 2023), attention (Pitis et al., 2020), modified Granger-causal tests (Chuck et al., 2023), and physical properties (Chuck et al., 2020). By incorporating invariance into the definition of actual cause, we can categorize these methods according to how they approximate $IVP(\mathbf{X}; \mathbf{Y})$: with arbitrary partitions, or heuristics such as local states or pairwise variables.

Analogous properties to actual-cause judgment approximations have been applied to exploration in RL (Wang et al., 2023), model-based reinforcement learning with graph networks (Feng & Magliacane, 2023), curiosity based methods (Sancaktar et al., 2022). One early direction is counterfactual data augmentation (Pitis et al., 2020), which has been extended to identify regions where a planner (Chitnis et al., 2021) or policy can perform well (Pitis et al., 2022). Alternative methods utilize interventional data (Baradel et al., 2019) applied at object interactions. State-specific causes have also been applied to skill learning, as shown by prior work on identifying actual-cause judgment equivalent properties with pairwise models (Chuck et al., 2020; 2023) and action counterfactuals (Choi et al., 2023). A model of the environment that encodes partial invariances

Method	Invariance	Necessity	Minimality
IAJ	General set partitioning	Arbitrary counterfactuals	Global minimality
Gradient (Wang et al., 2023)	Local state partitioning	Local counterfactuals	Gradient Regularization
Attention (Pitis et al., 2020)	General set partitioning	Not explicitly handled	Entropy Regularization
Granger (Chuck et al., 2023)	Pairwise-variable partitioning	Pairwise counterfactuals	Not used
Physical (Chuck et al., 2020)	Heuristic state partitioning	Rule-based counterfactuals	Physics

Table 3: Summary of key properties of actual cause characterized by related methods.

can be useful for planning, and recent work investigates this through causal structure learning (Ding et al., 2022; Wang et al., 2022; Weiss et al., 2022) and minimal dynamics modeling (Wang et al., 2024b). This work provides a theoretically grounded basis by which to unify these methods, while demonstrating the potential to scale to these contexts and infer actual-cause judgments in large and complex domains.

C Soft Judgments and Degrees of Actual Causation

Actual cause carries a long history of "soft assignments," or assignments which are not strict 0-1 Chockler & Halpern (2004); Halpern & Hitchcock (2015); Icard et al. (2017) in step with the notion of normality or responsibility. Similarly, causal judgments have been investigated with language such as "more of a cause" or "less of a cause," Gerstenberg et al. (2014), rather than a strict cause or non-cause. While the IAJ definition focuses primarily on binary judgments to provide a clear comparison with actual cause and the simplest first step, future work has several avenues for intuitively extending this notion to notions of responsibility or soft causal judgment.

Two possible extensions are as follows: **1)** Leverage the notion of soft-necessity and soft sufficiency as described in Appendix G, which dictates a soft boundary for the percentage of states where the outcome is either invariant to changes in the non-causes (soft sufficiency), or varies as a result of reassignment to the causes. In the existing definitions 4 and 5, these are used to demark a binary decision, but instead could return the soft value (or estimated soft value) itself. These definitions can also be extended to the probabilistic case by taking a weighted fraction of states. **2)** Take advantage of the notion that there may be multiple assignments of minimal IVPs, as is observed in the results in section N. IAJ simply takes the or-aggregation of the binary assignments for the different assignments of IVPs. However, an alternative strategy can be viewed as making a strength-of-cause assumption by reasoning over the multiple "worlds", where each world is a particular partitioning of the state space into variant and invariant regions.

The question of assigning degree of cause to different variables is an important one for which the actual-cause judgment framework, by relating notions from actual cause and causal judgments, offers a promising direction. Furthermore, we believe that there are straightforwad extensions to IAJ and joint optimization for minimum actual-cause judgment that would allow them to meaningfully capture soft judgments in a meaningful way.

D Proofs

Theorem D.6. *For an observed state* $\mathbf{s}^*$ *and binary-subset pair* $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$*, if the event* $\mathbf{X} = \mathbf{x}$ *satisfies AC1, functional sufficiency, and contrastive necessity for the outcome* $\mathbf{Y} = \mathbf{y}$ *under witness set* $\mathbf{W}$*, and* $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ *is minimal, then there cannot exist a smaller set of variables* $\mathbf{Z} \subset \mathbf{X}$ *such that* $\mathbf{Z} = \mathbf{z}$ *also satisfies AC1, functional sufficiency, and contrastive necessity for* $\mathbf{Y} = \mathbf{y}$*, where* $\mathbf{z}$ *denotes the values of variables* $\mathbf{Z}$ *in the assignment* $\mathbf{x}$*.*

Proof. Say that $\mathbf{s}^*$ lies within the invariant preimage $\mathcal{I}_k$. By definition the invariant preimage satisfies Equation 1, and therefore for every state $\mathbf{s} \in \mathcal{I}_k$ we will have:

$$f_{\mathbf{Y}}(\mathbf{s}_{\mathbf{X}}, \mathbf{s}_{\mathbf{C}}, \mathbf{u}) = f_{\mathbf{Y}}(\mathbf{s}_{\mathbf{X}}, \mathbf{s}'_{\mathbf{C}}, \mathbf{u}) \quad \forall\, \mathbf{s}' \in \mathcal{I}_k \text{ where } \mathbf{s}'_{\mathbf{X}} = \mathbf{s}_{\mathbf{X}}.$$

By definition, its corresponding binary vector b_k will have one element equal to one for every variable in $\mathbf{X}$. We first show that $\mathbf{s}^*$ could not have been mapped to an invariant preimage with another binary vector b_j such that $|b_j| > |b_k|$.

Say that $\mathbf{s}^*$ could have instead been mapped to some binary vector b_j with a longer length, i.e. $|b_j| > |b_k|$. By definition there is only one invariant preimage corresponding to a given subset of exogenous variables, and only one binary vector corresponding to each invariant preimage, so $\mathbf{s}^*$ would have to be moved to the invariant preimage $\mathcal{I}_j$ corresponding to the binary vector b_j.

Let L_1 denote the cost of the given binary-subset pair, i.e. $L_1 = \sum_{n=1}^{2^{|\mathbf{S}|}} |\mathcal{I}_n||b_n| = |\mathcal{I}_j||b_j| + |\mathcal{I}_k||b_k| + (|\mathcal{I}_1||b_1| + \cdots |\mathcal{I}_K||b_K|)$

If the state $\mathbf{s}$ was moved from $\mathcal{I}_k$ to $\mathcal{I}_j$ the resultant cost would be $L_2 = (|\mathcal{I}_j| + 1)|b_j| + (|\mathcal{I}_k| - 1)|b_k| + (|\mathcal{I}_1||b_1| + \cdots |\mathcal{I}_K||b_K|) = |b_j| - |b_k| + L_1$.

We know that $|b_j| > |b_k|$ and hence $L_2 > L_1$. Therefore, assigning $\mathbf{s}^*$ to the binary vector $|b_k|$ will have lower cost, and by extending this to all states $\mathbf{s} \in \mathcal{I}_k$ we will have $|b_k| \leq |\mathbf{X}|$.

Now say that $\mathbf{s}^*$ is instead mapped to some binary vector b_j with a shorter length, i.e. $|b_j| < |b_k|$, and its corresponding invariant preimage is $\mathcal{I}_j$. If moving this state to $\mathcal{I}_j$ leads to Equation 1 not being satisfied or necessity (Definition 4.4) not being satisfied, the resultant set of invariant preimages is not valid and therefore $\mathbf{Z} = \mathbf{z}$ cannot be a invariant-set actual-cause judgment . However, if the aforementioned properties are satisfied then since $\mathbf{s}^*$ is not in $IVP(\mathbf{X}; \mathbf{Y})$, either $\mathbf{X} = \mathbf{x}$ could not have been functionally sufficient for $\mathbf{Y} = \mathbf{y}$ or the given binary-subset pair was not minimal, both of which are contradictions. Hence we must have $|b_k| \geq |\mathbf{X}|$.

Since $|b_k| \leq |\mathbf{X}|$ and $|b_k| \geq |\mathbf{X}|$, we have that $|b_k| = |\mathbf{X}|$. Therefore, any actual cause that satisfies functional sufficiency and contrastive necessity with a minimal binary-subset pair must have cardinality $|\mathbf{X}|$. We note that this is true regardless of which witness set is used to determine if $\mathbf{Z} = \mathbf{z}$ was necessary since functional sufficiency would be violated either way. As a result, $\mathbf{Z} = \mathbf{z}$ cannot be a invariant-set actual-cause judgment .

□

D.1 IAJ subset of HP AC

We demonstrate that the set of all variables judged causes by IAJ is a subset of all variables declared causes by Modified HP (which we refer to as HP actual causes). Again, we restate the subset-necessity addition to the definition of IAJ necessity, replacing our existing definition of necessity. Note that on any of the existing examples, there will be no difference between subset-necessity and necessity, but it is possible to construct adversarial cases where a distinction must be made:

Definition D.7 (Subset Necessity)**.** An actual-cause judgment $\mathbf{X}$ is said to satisfy subset necessity if for any strict subset $\mathbf{Z}$ of an actual-cause judgment $\mathbf{X}$ that satisfies necessity, $\mathbf{R} = \mathbf{X}/\mathbf{Z}$ also satisfies subset necessity. If there does not exist a subset of $\mathbf{X}$ that satisfies necessity, then $\mathbf{X}$ must be necessary.

Theorem D.8. *For an observed state $\mathbf{s}^*$, if $\mathbf{X} = \mathbf{x}$ is a invariant-set actual-cause judgment of $\mathbf{Y} = \mathbf{y}$ for any binary-subset pair $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ then the assignment of every causal variable $\mathbf{X}_i = \mathbf{x}_i$ in $\mathbf{X} = \mathbf{x}$ is contained in some Modified HP actual cause $\mathbf{Z}_i = \mathbf{z}_i$*

Proof. We prove this case by contradiction. Suppose that there is a set of variables $\mathbf{R}$ in IAJ $\mathbf{X}$ not in any HP AC. By definition, this must mean that $\mathbf{R}$ does not satisfy HP necessity, or that none of the variables contained are HP minimal.

First, let us analyze the case where $\mathbf{R}$ is not necessary. This will then suggest two cases. Either $\mathbf{Q} = \mathbf{X}/\mathbf{R}$ is necessary, or it is not necessary. If it is necessary, then by subset necessity $\mathbf{R}$ must also be necessary, a contradiction. If it is not necessary, then there must exist some subset $\mathbf{T} \subseteq \mathbf{R}$ such that $\mathbf{Q} \cup \mathbf{T}$ is necessary, otherwise necessity for $\mathbf{X}$ would be violated and $\mathbf{X}$ is not in any IAJ. However, notice the following proof to demonstrate that $\mathbf{T}$ must equal $\mathbf{R}$: consider $\mathbf{T}_0$ as $\mathbf{R}/\mathbf{T}$. If $\mathbf{Q} \cup \mathbf{T}$ is necessary, then by subset necessary $\mathbf{T}_0$ is necessary. If both $\mathbf{Q} \cup \mathbf{T}$ and $\mathbf{T}_0$ are necessary, then $\mathbf{R}$ is necessary, since $\mathbf{T}_0 \subset \mathbf{R}$ is necessary, a

contradiction. If $\mathbf{T}_0$ is empty, then $\mathbf{R}$ must be part of the HP AC that contains the minimal component of $\mathbf{Q}$ and $\mathbf{R}$ that is necessary. In this case, $\mathbf{R}$ will be contained in some HP AC, which contradicts the original premise.

If the set does not satisfy minimality, then there must be a set of strict subsets $\{\mathbf{R}_1, \ldots, \mathbf{R}_k\}$ which is necessary. Notice two cases, either the union of all strict subsets is equal to $\mathbf{R}$, or there exists some set of variables $\mathbf{Q}$ that are not in any necessary subset. In the first case, this means that all the variables in $\mathbf{R}$ are contained in an HP AC, a contradiction. In the second case, this means that $\mathbf{R}$ is not subset necessary, which would imply that $\mathbf{X}$ is not subset necessary, which means that $\mathbf{X}$ is not an IAJ. Consequently, this demonstrates that there cannot exist a set of variables $\mathbf{R}$ in IAJ $\mathbf{X}$ that are not contained in some HP AC. □

The following corollary follows directly from the above theory.

Corollary D.9. *For an observed state $\mathbf{s}^*$, the set of variables included in at least one IAJ, $\mathcal{X}$, is a subset of the set of variables included in at least one HP AC $\mathcal{Z}$, or $\mathcal{X} \subseteq \mathcal{Z}$.*

Proof. Since Theorem D.8 implies that any IAJ is composed of HP ACs, it follows trivially that the set of all IAJs must be contained in the set of all HP ACs. For completeness, by contradiction, suppose that there is a IAJ variable that was not in an HP AC. This would violate Theorem D.8. Since this is true for any variable, it is also true for the set of all variables. □

Note the consequence of this: when looking at all the possible causes, IAJ will always be more restrictive than HP AC. In an AI system, this means that using IAJ will direct the agent towards at most the same number of actual cause judgments as HP actual causes. However, as we have seen in the pusher and other examples, in practice IAJ will ignore many possible actual causes, except when they have a direct effect on the outcome. Overall, this suggests more robust actual cause judgments.

E Direct Sufficiency and Proof of Theorem 4.9

Beckers (2021) provides a definition of *direct sufficiency*, which imposes a stronger constraint by requiring that the outcome is always $\mathbf{Y} = \mathbf{y}$ when the intervention $\mathbf{X} \leftarrow \mathbf{x}$ is applied, regardless of the values of all other variables $\mathbf{C}$. In other words, the outcome should be robust to all possible interventions on the remaining variables $\mathbf{C} = \mathbf{S} - \mathbf{X}$.

Definition E.1 (Direct Sufficiency)**.** For an observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is directly sufficient for the outcome $\mathbf{Y} = \mathbf{y}$ if for all $\mathbf{c} \in \mathcal{R}(\mathbf{C})$ we have that $(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}]\mathbf{Y} = \mathbf{y}$.

Invariant preimages are defined such that the invariance to all remaining variables $\mathbf{C}$ only needs to hold for a subset of possible assignments of $\mathbf{C}$, given by the states in $IVP(\mathbf{X}; \mathbf{Y})$, while direct sufficiency requires that the invariance holds for all assignments of $\mathbf{C}$. In other words, while direct sufficiency is a statement about complete invariance with respect to $\mathbf{C}$, sufficient judgment is a statement about context-specific invariance to $\mathbf{C}$ at the observed state in particular. To illustrate this, we define a stricter notion of invariance as follows:

Definition E.2 (Complete Invariance)**.** An invariant preimage $IVP(\mathbf{X}; \mathbf{Y})$ shows *complete invariance* with respect to a state $\mathbf{s}^*$ if all states $\mathbf{s}'$ where $\mathbf{s}'_{\mathbf{X}} = \mathbf{s}^*_{\mathbf{X}}$ also lie within $IVP(\mathbf{X}; \mathbf{Y})$.

It can be verified that if complete invariance holds for one state $\mathbf{s} \in IVP(\mathbf{X}; \mathbf{Y})$ then it should hold for all other states in $IVP(\mathbf{X}; \mathbf{Y})$. We now complete the proof of Theorem 4.9.

Theorem E.10. *For an observed state $\mathbf{s}^*$, sufficient judgment and direct sufficiency are equivalent when the invariant preimage $IVP(\mathbf{X}; \mathbf{Y})$ that contains $\mathbf{s}^*$ shows complete invariance, i.e. all other states $\mathbf{s}'$ where $\mathbf{s}'_{\mathbf{X}} = \mathbf{s}^*_{\mathbf{X}}$ also lie within $IVP(\mathbf{X}; \mathbf{Y})$.*

Proof. We first demonstrate that a functionally sufficient actual cause under complete invariance implies a directly sufficient one, then show that a directly sufficient actual cause implies a functionally sufficient one with complete invariance.

If the event $\mathbf{X} = \mathbf{x}$ were functionally sufficient for the outcome $\mathbf{Y} = \mathbf{y}$ at the observed state $\mathbf{s}^*$, and the states corresponding to the assignment $\mathbf{X} = \mathbf{s}^*_{\mathbf{X}}$ and all possible assignments of $\mathbf{C}$ were present within $IVP(\mathbf{X}; \mathbf{Y})$, then any arbitrary intervention on $\mathbf{C}$ along with the intervention $\mathbf{X} \leftarrow \mathbf{x}$ would result in a state in $IVP(\mathbf{X}; \mathbf{Y})$. By the definition of invariant preimage, the outcome would remain the same for all of these states. Hence $\mathbf{X} = \mathbf{x}$ would also be directly sufficient for the outcome $\mathbf{Y} = \mathbf{y}$.

Similarly, if $\mathbf{X} = \mathbf{x}$ were directly sufficient for the outcome $\mathbf{Y} = \mathbf{y}$ the outcome would need to remain the same under all possible assignments of the remaining variables $\mathbf{C}$. If the invariant preimage $IVP(\mathbf{X}; \mathbf{Y})$ showed complete invariance with respect to the state $\mathbf{s}^*$, then all of the states corresponding to the assignment $\mathbf{X} = \mathbf{s}^*_{\mathbf{X}}$ and all possible assignments of $\mathbf{C}$ would be present within $IVP(\mathbf{X}; \mathbf{Y})$. One of these states is the observed state $\mathbf{s}^*$. Since direct sufficiency implies weak sufficiency (Proposition 2, Beckers (2021)) and $\mathbf{s}^*$ is in $IVP(\mathbf{X}; \mathbf{Y})$, the event $\mathbf{X} = \mathbf{x}$ is functionally sufficient for the outcome $\mathbf{Y} = \mathbf{y}$.

□

By explicitly encoding context-specific independence, sufficient judgment captures the spectrum between weak sufficiency (no invariance) and direct sufficiency (complete invariance). The above statements are expressed without the witness set for simplicity, but the same holds with a witness since context-specific independence and the assignment of states to invariant preimages are not affected by the choice of witness.

F Additional Examples

In this section, we evaluate the Modified HP (Halpern, 2016) and direct sufficiency (Beckers, 2021) definitions on a few discrete examples that illustrate context-specific independence and illustrate the limitations of these definitions. Beckers (2021) also introduces the strong sufficiency definition, which operates on extended causal chains, and in future work, we plan to extend IAJ to this setting. However, in the following examples, strong sufficiency acts the same way as direct sufficiency since they all involve causes that are direct parents of the outcome.

Pusher	**Obstacle**	**Final**
499	not 501	501
500	not 501 or 502	502
499	501	500
500	501	500
500	502	501
not 499 or 500	all	500

Table 4: **Discrete 1D Pusher** environment definition, mapping states to outcomes

F.1 Discrete 1D Pusher

This example is analogous to the 1D pusher domain in Example 2.1 but uses discrete states. We do not use a continuous state space domain because Modified HP and direct sufficiency do not scale to continuous state spaces in their present form.

The discrete 1D pusher domain contains three causal variables: pusher, obstacle, and final block position. The pusher and obstacle positions can vary in the set $\{0, \ldots, 1000\}$, and the block position in range $\{500, 501, 502\}$. If the agent is within $\{499, 500\}$, the final block position will be $500 + (\text{agent position} - 500 + 2)$, unless the obstacle obstructs the block. Otherwise, the final block position remains constant at 500. We illustrate the dynamics of this environment in Table 4.

Suppose we are in the state corresponding to the first row of the table, where the pusher position is 499, the obstacle position is 200 and the final block position is 501, and we want to determine whether the obstacle *alone* is an actual cause. The pusher, obstacle, and final block positions are denoted as P, X, and Y respectively, and there are no exogenous variables U in this example. We can verify that the obstacle would be an actual cause according to the direct sufficiency and Modified HP definitions.

Modified HP:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 501]Y \neq 501$ is true, as can be seen in the third row of the table. Note in this case the witness set is empty.

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 200]Y = 501$ is true, as can be seen in the first row of the table.

AC3: There is no strict subset of X that could be an actual cause since the empty set does not satisfy AC2a.

Direct Sufficiency and Necessity:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 501, P \leftarrow 499]Y \neq 501$ is true, as can be seen in the third row of the table. This satisfies AC2a because this counterexample demonstrates that there is a counterfactual assignment that is not sufficient. In this case the pusher is in the witness set and set to its actual value (and is the same witness for sufficiency).

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 200, P \leftarrow 499]Y = 501$ is true, as can be seen in the first row of the table. Note that the C term for direct sufficiency $[X \leftarrow x, W \leftarrow w^*, C \leftarrow c]$ is satisfied because the set of variables C in this case is empty.

AC3: There is no strict subset of X that could be an actual cause since the empty set does not satisfy AC2a.

Thus, both definitions imply that the obstacle alone is an actual cause *in every state where the pusher pushes the block*, even though in 99.9% of these cases, the outcome is contextually independent of the obstacle. While this conclusion is not technically wrong, it is ultimately impractical for downstream tasks to utilize the returned actual causes, as they cannot take advantage of the context-specific independence.

F.2 Rocket Ship

A rocket ship has two thrusters, represented by causal variables $T1$ and $T2$, which can turn on or off. If either one of the thrusters fires, the rocket ship will launch ($Y = 1$). If neither thruster fires, the rocket ship will not launch ($Y = 0$). There is also a bird X that is flying around the field, its position taking on values in $\{0, \ldots, 1000\}$. There is one particular position, $X = 500$, where the bird knocks out the wiring and therefore causes the rocket to fail to launch, i.e. $Y = 0$.

T1	T2	Bird	Launch
1	1	not 500	1
1	0	not 500	1
0	1	not 500	1
0	0	not 500	0
1	1	500	0
1	0	500	0
0	1	500	0
0	0	500	0

Table 5: **Rocket Ship** environment

Suppose we are in the state corresponding to the first row of the table, where $T1 = 1, T2 = 1$, the bird position is $X = 200$, and $Y = 1$. We want to determine if the bird position alone is an actual cause. Note that there are no exogenous variables U in this example. We can verify that the bird position alone would be an actual cause according to the direct sufficiency and Modified HP definitions.

Modified HP:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 500]\mathbf{Y} \neq 1$ is true, as can be seen in the fifth row of the table. Note in this case the witness set is empty.

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 200]\mathbf{Y} = 1$ is true, as can be seen in the first row of the table.

AC3: There is no strict subset of the bird position alone whose assignment is an actual cause, since the empty set does not satisfy AC2a.

Direct Sufficiency and Necessity:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 500, T1 \leftarrow 1]Y \neq 1$ is true, as can be seen in the fifth row of the table. This counterexample demonstrates that there is a counterfactual assignment of X that is not sufficient. In this case, the first thruster is in the witness set.

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [X \leftarrow 200, T1 \leftarrow 1, T2 \leftarrow 1]Y = 501$ is true and $(M, \mathbf{u}) \vDash [X \leftarrow 200, T1 \leftarrow 1, T2 \leftarrow 0]Y = 501$ is true, as can be seen in the first and second rows of the table. Note that the C term for direct sufficiency is $T2$ since this is the only unassigned variable, and we vary over all its possible values (1 and 0), and the outcome is still sufficient. As with AC2a, the first thruster is in the witness set.

AC3: There is no strict subset of the bird position alone whose assignment is an actual cause, since the empty set does not satisfy AC2a.

With both definitions, the bird's position alone is an actual cause of the rocket launch when either thruster fires. If an agent encoded this form of invariance, it would assume that the bird was the reason the rocket launched, despite the outcome being dependent on the bird's position in only 0.1% of states. In contrast, the *functional actual cause* definition (Definition 4.7) would assign the values of (T1,T2) as a minimal actual cause in the observed state.

F.3 Offensive Chatbot

Spatial proximity can overlap with context-specific independence in many cases; for example, an object far away from an agent will often not influence its motion. However, this overlap is by no means complete, and spatial proximity or distances between counterfactual scenarios do not serve as a general proxy for context-specific independence. In this section, we provide an example where spatial proximity has no bearing on the outcome.

User	Engineer	Bot
u	not 500	**u**
u	500	0

A chatbot C is designed to repeat (parrot) what the user U inputs, which we represent by having U take on values between $1 - 1000$ and assigning $C = U$. Suppose there is a software engineer E who can do a variety of actions between $1 - 1000$, most of which have no effect. However, if they take the action $E = 500$, then this will cause the chatbot to crash, and output $C = 0$ regardless of what the user inputs.

Suppose we are in the state: User at 500, Engineer at 200, so the chatbot is 500, corresponding to the first row of the table. We want to determine if the Engineer alone is an actual cause.

Modified HP:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [E \leftarrow 500]\mathbf{C} \neq 500$ is true, as can be seen in the second row of the table. Note in this case the witness set is empty.

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [E \leftarrow 200]\mathbf{C} = 500$ is true, as can be seen in the first row of the table.

AC3: There is no strict subset of the engineer alone which is an actual cause since that would imply the empty set, which does not satisfy AC2a.

Direct Sufficiency and Necessity:

AC1: Satisfied since this is the observed state.

AC2a: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [E \leftarrow 500, U \leftarrow 500]Y \neq 500$ is true, as can be seen in the second row of the table. This satisfies AC2a because this counterexample demonstrates that there is a counterfactual assignment that is not sufficient. In this case, the User (U) is in the witness set, and matches.

AC2b: Satisfied for M as the environment since $(M, \mathbf{u}) \vDash [E \leftarrow 200, U \leftarrow 500]Y = 500$ is sufficient, and by putting U in the witness set ensures that there are no other counterfactual assignments.

AC3: There is no strict subset of the engineer among which is an actual cause since that would imply the empty set, which does not satisfy AC2a.

Thus, the engineer is independently an actual cause of the output $U = 500$ under both definitions. Now, suppose that the output is offensive. This suggests that the Engineer is solely responsible for this output *because they did not crash the chatbot*. This is a very strong and counterintuitive enforcement of responsibility. Furthermore, suppose that the engineer is an RL agent instead. If there was any chance of offensive output,

the agent would always destroy the output since it is the sole actual cause, even though the user could have output any value, and the chatbot parrots whatever the user inputs.

This example demonstrates how the proximity of objects is not the key factor in assessing normality in actual causation. Instead, what matters is the size or likelihood of the event set that produces varying outputs. In this case, the engineer could have performed any number of actions other than crashing the bot, and would not have changed the result.

G Soft Necessity and Sufficiency

This section describes the *soft necessity and sufficiency* conditions used in Algorithm 0. These conditions, denoted α_1-necessity and α_0-sufficiency, can be incorporated into IAJ by replacing the contrastive necessity and weak sufficiency conditions. Unlike the invariant preimage and global minimality criterion discussed in the main paper, these properties only provide rough measures of normality through statistical requirements on the number of counterfactual outcomes. We plan to investigate these properties further in future work.

G.1 Soft Necessity

By default, contrastive necessity requires there is only one alternative assignment of $\mathbf{X}$ which is not sufficient for the outcome. We instead require that at least α_1 percent of the assignments of $\mathbf{X}$ are not sufficient for the outcome. Enforcing this would mean that for an event to be considered an actual-cause judgment, deviating from the observed event would result in a change in outcome more often.

Definition G.1 (α_1-necessity)**.** For an observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is α_1-necessary for the outcome $\mathbf{Y} = \mathbf{y}$ under the witness set $\mathbf{W}$ if the fraction of values within its support for which the resultant outcome deviates from the observed outcome $\mathbf{y}$ is at least $\alpha_1 \in (0, 1)$.

$$\frac{1}{|\mathcal{R}(\mathbf{X})|} \sum_{\mathbf{x}' \in \mathcal{R}(\mathbf{X})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) \geq \alpha_1 \tag{4}$$

By increasing the value of the ratio hyperparameter α_1, the event is declared as necessary if a change in $\mathbf{X}$ results in a different outcome in a greater percentage of states, instead of simply verifying that one such state exists. Doing so ensures that not only is the event necessary, but that a change in outcome is a more *normal* or common occurrence, and not just an edge case.

Theorem G.2. *α_1-necessity implies contrastive necessity.*

Proof. When α_1-necessity is satisfied for $\alpha_1 > 0$, we will have:

$$\begin{aligned}
&\frac{1}{|\mathcal{R}(\mathbf{X})|} \sum_{\mathbf{x}' \in \mathcal{R}(\mathbf{X})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) > 0 \\
&\implies \sum_{\mathbf{x}' \in \mathcal{R}(\mathbf{X})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) > 0 \\
&\implies \exists \mathbf{x}' \in \mathcal{R}(\mathbf{X}) \quad (M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}
\end{aligned}$$

Hence contrastive necessity is always satisfied when α_1-necessity is satisfied. However, when α_1-necessity is not satisfied for $\alpha_1 > 0$ we will have:

$$\frac{1}{|\mathcal{R}(\mathbf{X})|} \sum_{\mathbf{x}' \in \mathcal{R}(\mathbf{X})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

$$\implies \sum_{\mathbf{x}' \in \mathcal{R}(\mathbf{X})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

$$\implies (M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}', \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y} \qquad \forall \mathbf{x} \in \mathcal{R}(\mathbf{X})$$

Hence contrastive necessity is not satisfied when α_1-necessity is not satisfied. □

The above result shows that α_1-necessity is a generalization of contrastive necessity; the latter is satisfied when there is at least one counterfactual event that can change the outcome, but α_1-necessity requires a larger set of counterfactual events to be able to change the outcome.

G.2 Soft Sufficiency

Weak sufficiency does not place hard requirements on the assignments of the non-causes, and any value of the non-causes can result in a change in outcome. However, we instead require that at least $(1 - \alpha_0)$ percent of assignments of $\mathbf{X}$ produce the same outcome. This places a limit on the number of assignments of $\mathbf{X}$ for which the outcome is not invariant to the remaining variables.

Definition G.3 (α_0-sufficiency)**.** For an observed state $\mathbf{s}^*$, the event $\mathbf{X} = \mathbf{x}$ is α_0-sufficient for the outcome $\mathbf{Y} = \mathbf{y}$ under the witness set $\mathbf{W}$ if the fraction of values within the support of its complement $\mathbf{C} = \mathbf{S} - (\mathbf{W} \cup \mathbf{X})$ for which the resultant outcome deviates from the observed outcome $\mathbf{y}$ is less than $\alpha_0 \in [0, 1)$.

$$\frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) \leq \alpha_0 \tag{5}$$

The above definition is a generalization of the weak sufficiency (Definition 4.2) and direct sufficiency (Definition E.1) definitions. As with the invariant preimage, α_0-sufficiency captures a sufficiency requirement that is between the strict condition of complete invariance and the weak condition of invariance not being required. We can show that each of these extreme cases can be captured with α_0-sufficiency.

Theorem G.4. *When $\alpha_0 = 0$, α_0-sufficiency is equivalent to direct sufficiency.*

Proof. Substituting $\alpha_0 = 0$ in Equation 5 gives us:

$$\frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

$$\implies \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

Hence there is no assignment to the remaining variables $\mathbf{C} = \mathbf{c}$ for which we can have an alternative outcome $\mathbf{Y} \neq \mathbf{y}$.

$$\implies (M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y} \quad \forall \mathbf{c} \in \mathcal{R}(\mathbf{C})$$

This is the definition of direct sufficiency, and hence α_0-sufficiency implies direct sufficiency. Similarly, when direct sufficiency holds, we have:

$$(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y} \quad \forall \mathbf{c} \in \mathcal{R}(\mathbf{C})$$

Hence, there is no $\mathbf{C} \leftarrow \mathbf{c}$ for which an alternative outcome $\mathbf{Y} \neq \mathbf{y}$ can occur.

$$\implies \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

$$\implies \frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) = 0$$

In other words, α_0-sufficiency holds when direct sufficiency holds. Therefore, α_0-sufficiency is equivalent to direct sufficiency for $\alpha_0 = 0$.

□

Theorem G.5. *α_0-sufficiency implies weak sufficiency, unless AC1 is violated.*

Proof.

$$\frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} \neq \mathbf{y}) \leq \alpha_0$$

$$\implies \frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}) \geq 1 - \alpha_0$$

Since $\alpha_0 \in [0, 1)$, we will have $1 - \alpha_0 > 0$.

$$\implies \frac{1}{|\mathcal{R}(\mathbf{C})|} \sum_{\mathbf{c} \in \mathcal{R}(\mathbf{C})} \mathbb{1}((M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}) > 0$$

$$\implies \exists \mathbf{c} \in \mathcal{R}(\mathbf{C}) \quad (M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{c}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}$$

In other words, at least one state exists where the event $\mathbf{X} = \mathbf{x}$ occurs and the resulting outcome is $\mathbf{Y} = \mathbf{y}$. This trivially holds when AC1 is satisfied since the observed state $\mathbf{s}^*$ will result in the observed outcome $\mathbf{Y} = \mathbf{y}$. However, if the states where the resulting outcome is $\mathbf{Y} = \mathbf{y}$ correspond to some $\mathbf{c} \neq \mathbf{s}^*_{\mathbf{C}}$ instead of the observed values, then AC1 will be violated. When AC1 is satisfied, we will have:

$$(M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{C} \leftarrow \mathbf{s}^*_{\mathbf{C}}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}$$

Since all variables are intervened on and $\mathbf{s}^*_{\mathbf{C}}$ is the observational value of $\mathbf{C}$, we can drop $\mathbf{C} \leftarrow \mathbf{s}^*_{\mathbf{C}}$ from the above equation.

$$\implies (M, \mathbf{u}) \vDash [\mathbf{X} \leftarrow \mathbf{x}, \mathbf{W} \leftarrow \mathbf{s}^*_{\mathbf{W}}]\mathbf{Y} = \mathbf{y}$$

Therefore, weak sufficiency is satisfied when α_0-sufficiency and AC1 are satisfied.

□

From a practical standpoint, most algorithms for discovering actual-cause judgments will run an $O(1)$ check for AC1 and terminate immediately if it fails, so in practice we can assume that α_0-sufficiency implies weak sufficiency. Theorems G.2 and G.5 show that α_1-necessity and α_0-sufficiency are stricter than the Modified HP definition.

In general, the motivation for applying soft necessity and sufficiency is to exclude spurious actual-cause judgments in small state spaces where the number of possible binary vectors exceeds the number of possible states. In these cases, states with equivalent outputs can be sequestered to a single IVP with a smaller corresponding binary vector. In other words, when the outcome space is small, an algorithm for discovering invariant-set actual-cause judgment might return part of a cause instead of the full cause to reduce the cost of the overall assignment. For example, Table 6 shows the results in the 1D mover domain previously outlined in Section 5. The rows corresponding to binary 01 show that when the obstacle obstructs the mover, it is possible to reduce the cost by assigning all states corresponding to the outcome of 1 to the binary 01 such that only the obstacle position is an actual-cause judgment. We prevent this effect by forcing a certain percentage of states to take on certain values. In practice, we used α_0-sufficiency but did not require α_1-necessity to obtain accurate results.

Binary	Mover	Obstacle	Outcome
01	0	0	1
01	0	2	1
01	1	1	2
01	1	2	1
10	0	1	0
10	1	0	2
10	2	0	3
10	2	1	3
10	2	2	3
10	3	0	4
10	3	1	4
10	3	2	4

Table 6: 1D Mover results with weak sufficiency instead of α_0-sufficiency.

Binary	Mover	Obstacle	Outcome
11	0	2	1
01	1	2	1
10	0	0	1
10	0	1	0
10	1	0	2
10	1	1	2
10	2	0	3
10	2	1	3
10	2	2	3
10	3	0	4
10	3	1	4
10	3	2	4

Table 7: 1D Mover results with α_0-sufficiency, but without enforcing invariance (**P5**).

However, α_0-sufficiency and α_1-necessity alone are not enough to represent normality, and the invariant preimage is still essential. Table 7 shows the results in the same 1D Mover example where α_0-sufficiency is included but the invariance property (**P5**) is not enforced. The states $[0, 1]$ and $[1, 1]$ have different outcomes but the binary vector 10 is still assigned, implying that the outcome is not invariant to the obstacle position. For this reason, we set $\alpha_0 \neq 0$ to permit some counterfactual assignments of the candidate actual-cause judgments to result in different outcomes, provided they continue to satisfy the invariance property given by the invariant preimage containing the observed state.

It is worth noting that contrastive necessity and direct sufficiency do not readily scale to complex environments with continuous state spaces, since verifying them requires an enumeration of all counterfactual possibilities. In contrast, the invariance property can be approximated easily through masking, as demonstrated in JOA.

G.3 Applications of α_0-sufficiency: Sequestering

Sequestering is the case in discrete scenarios where a small number of outcomes caused by one process can be assigned to a different invariant preimage by assigning or *sequestering* those outcomes to a particular outcome. This happens because if a particular input is not mapped to a unique outcome, it can be assigned as a pseudo-actual cause of a singular outcome. To illustrate this process, consider the Halt and Charge example from the actual causality Halpern (2016) literature — we evaluate our approach on this example in Appendix N. This scenario is represented in the following table:

Major	Seargent	Corporal
0	0	0
0	1	0
1	0	1
1	1	1
2	0	0
2	1	1

In this scenario, consider the last row where the major does not speak (Major = 2) and the sergeant gives the charging order (Sergeant = 1), resulting in the Corporal charging (Corporal = 1). The result agreed upon in the literature for this scenario is that the Sergeant is an actual cause of the outcome since the Sergeant could have given the charge order to change the outcome. It is also true that the Major is an actual cause in this scenario, since if the Major had given the halt order, the outcome would have changed. Although the literature can be ambiguous on this, the simplest answer would be to consider both of them jointly responsible for the outcome since it is both the Major's choice to stay silent and the Sergeant's order that results in the Corporal's action. This is the verdict given by the direct sufficiency (Beckers, 2021) definition, but not by the

Modified HP (Halpern, 2015) definition. When we use $\alpha_0 = 1.0$ we get the minimum cost binary-subset pairs in Table 8.

Binary	Major	Seargent	Corporal
10	0	0	0
10	0	1	0
10	1	0	1
10	1	1	1
01	2	0	0
01	2	1	1

Binary	Major	Seargent	Corporal
10	0	0	0
10	0	1	0
10	1	0	1
10	1	1	1
10	2	0	0
01	2	1	1

Binary	Major	Seargent	Corporal
10	0	0	0
10	0	1	0
10	1	0	1
10	1	1	1
01	2	0	0
10	2	1	1

Table 8: Sequestering: min-cost binary-subset pairs for Halt and Charge with $\alpha_0 = 1.0$

Notice that while in aggregate the three solutions assign the Major and the Sergeant each to be the actual cause in the last row, the Sergeant is at no point jointly responsible with the Major. This is because we can sequester exactly one Sergeant-dependent outcome to be "caused" by the major taking on the silent action (Major = 2). Let us represent states with a length 3 vector corresponding to the Major, Sergeant, and Corporal, respectively. Notice that in all rows corresponding to the binary vector 10, i.e where the Major is the actual cause, the Sergeant decision can change the outcome in the states $000, 010, 101, 111$ *without assigning any outcomes to the Major taking the silent action (Major = 2).* Because of this, exactly one state where Major = 2 occurred would lead to it being considered a actual-cause judgment. This is because it trivially satisfies necessity: within the IVP corresponding to binary vector 10, the outcome will be trivially invariant to changes in the non-causes (Sergeant) because no other states with that value (ex. Major = 2) are in the IVP.

This issue is linked to the same motivation for direct sufficiency (Beckers, 2021), which enforces that the outcome is robust to perturbations of the non-causes. α_0-sufficiency is a softer version of direct sufficiency, which works alongside the invariant preimage to prevent these singleton sets by forcing the non-causes to be assigned to the appropriate values, as described earlier. Furthermore, because this edge case can only be applied to singleton sets, it is less likely to occur in continuous environments.[2]

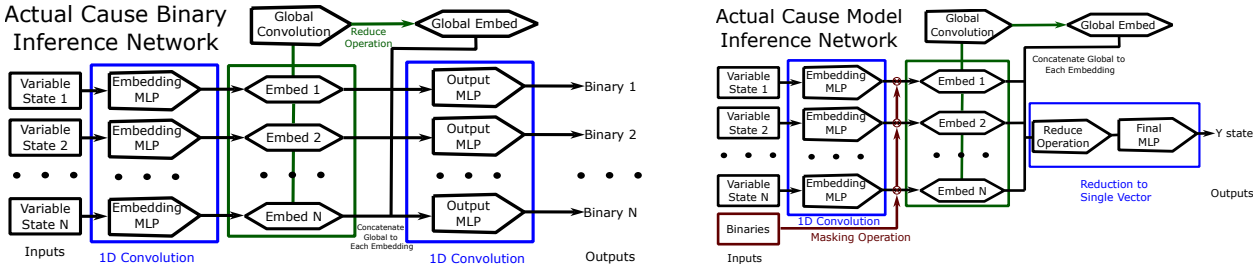
H Network and Training Details

We train two neural networks to identify actual causes from data, one to model the state-binary mapping $h_{\mathbf{Y}}(\mathbf{s}^{(i)}; \theta)$, and the other to model the dynamics $f(\mathbf{s}^{(i)}, \mathbf{b}^{(i)}; \phi)$. Both networks take in an unordered sequence of vectors corresponding to the causal variables; f returns the state of the outcome variable and $h_{\mathbf{Y}}$ returns a binary vector corresponding to the inferred actual cause in that state. In practice, several details are used to ensure the networks are able to learn.

Both networks utilize 1D convolutions to encode variable-wise invariance. A 1D convolution applies the same multi-layer perceptron (MLP) to each input, where each input is the state of a different causal variable. The first 1D convolution produces the embeddings. These embeddings are aggregated by adding together the outputs. This aggregated result allows the model to include global information and is appended to each of the inputs for a second 1D convolution. In $h_{\mathbf{Y}}$, the output of this second 1D convolution is $\mathbf{b}_i$ for each vector. In f, this vector is passed through a final MLP that computes the output. Without having two layers, global information would not be included in $h_{\mathbf{Y}}$, restricting the class of functions that f can represent.

While the networks for f and $h_{\mathbf{Y}}$ share the majority of their architecture, there are a few key differences. For f, a masking operation is applied after the first embedding operation: for any variable k with $\mathbf{b}[k] = 0$, the corresponding embedding of the state (the outcome of the 1D convolutions before reduction) is multiplied by $\mathbf{b}[k]$ to mask it out. Applying the masking operation at the embedding layer prevents the state of the

[2]A similar issue can occur in data-deficient regimes when learning from observational data in the scenario of overfitting, where an inference model might be willing to incur some prediction error to produce a smaller binary because the observation space exceeds the size of the outcome space.



(a) Inference Model (b) Masked Forward Model

Figure 2: **(a)** Computational flow of $h_{\mathbf{Y}}(\mathbf{s};\theta)$, which takes in the observed state for all causal variables and returns a binary vector indicating the actual cause. **(b)** Computational flow of $f(\mathbf{s},\mathbf{b};\phi)$, which takes in the observed state for all causal variables and a binary vector and returns the state of the outcome variable.

variable $\mathbf{s}_k = \mathbf{0}$ being confused with the binary vector mask $\mathbf{b}[k] = 0$. For $h_{\mathbf{Y}}$, the binary vectors are directly returned after the second convolution.

The necessity property is encouraged through the structure of $h_{\mathbf{Y}}(\mathbf{s}^{(i)};\theta)$, which constructs $\mathbf{b}$ by sharing the 1D convolution of $\mathbf{b}^{(i)}[k] = h_{\mathbf{Y}}(\mathbf{s}_k^{(i)},\theta)$ for every $\mathbf{s}_k^{(i)}$. This structure biases the output binary vector to be dependent on the input value, while still utilizing global information. Since the necessity property captures counterfactual dependence on a factor, this constraint biases towards necessity. We opt for this approach instead of explicitly generating counterfactual states and verifying necessity, since without a perfect model this would lead to instability during training. An illustration of the binary network is in Figure 2a, and the masked forward model is in Figure 2b. Hyperparameters are set identically for f and $h_{\mathbf{Y}}$, and the 1D convolutions both use 3 hidden layers of size 256 and embed to size 512.

In practice, $h_{\mathbf{Y}}$ cannot output hard $\{0,1\}$ binary values in training, since this would result in zero gradient flow from the forward model f to the state-binary mapping results. However, if the forward model f received stochastic inputs instead of values in $\{0,1\}$ it would affect the masking operation and result in poor prediction performance. We resolve this using a mixing distribution, where the network for $h_{\mathbf{Y}}$ outputs values in $\widehat{\mathbf{b}} := [0,1]^{|\mathbf{S}|}$. We then use $\widehat{\mathbf{b}}$ as the probability of a Bernoulli random variable to give $\widetilde{\mathbf{b}} := \text{Bern}(\widehat{\mathbf{b}})$, and input $\mathbf{b} := \widetilde{\mathbf{b}} \cdot \widehat{\mathbf{b}}$ to the forward model f. This ensures gradient flow when training $h_{\mathbf{Y}}$, while still returning $\mathbf{b}_i \in \{0,1\}$. As an aside, we tried using a Gumbel-Bernoulli distribution as in Hwang et al. (2023), but it did not yield good results empirically.

I Random Vectors

In this section, we elaborate on some of the details of the *Random Vectors* domains. Each domain is defined by a set of active variables $\mathcal{X} = \{X_1, \ldots X_n, Y\}$, where Y is the outcome variable, as well as a set of passive variables $\tilde{\mathcal{X}} = \{\tilde{X}_1, \ldots \tilde{X}_n, \tilde{X}_Y\}$. All variables in the model are d-dimensional vectors and take on values in $[-1,1]^d$. The model determines their values with R relations: linear functions of the values of parent variables $\mathcal{P}_r \subseteq \mathcal{X} \cup \tilde{\mathcal{X}}$ whose output is one of the active variables in $\mathcal{X}$. If there are R_t relations whose output is the active variable X_t then each of these relations returns a value in $[-\frac{1}{R_t}, \frac{1}{R_t}]^d$

Each active variable $X_t \in \mathcal{X}$, including the outcome, has a corresponding passive variable $\tilde{X}_t$. For each relation where the output variable is X_t, the returned value always depends on the passive variable, but it may or may not depend on the active variables based on their current values. We aim to infer the actual causes of outcome variable Y.

Each relation f_r is associated with four matrices $\mathbf{A_r}, \mathbf{B_r}, \mathbf{C_r}, \mathbf{D_r}$, and is a conditional linear relationship between a target variable X_t, its active variable parents $\mathcal{P}_r$, and its passive variable $\tilde{X}_t$. Let $x_{pa} = [x_i, \ldots]_{i \in \mathcal{P}_t}$ denote the concatenation of all the active variable parent values, and $[x_{pa}, \tilde{x}_t]$ denote the concatenation of the active variable parent values and passive variable value. In the linear case, the output of relation f_r is given by:

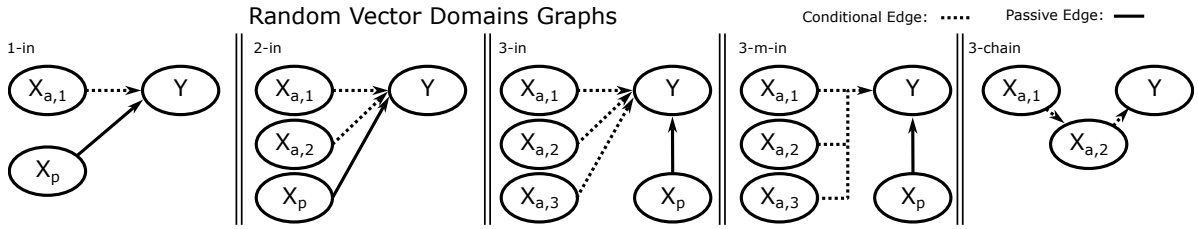


Figure 3: Random Vectors domains, where dotted lines indicate conditional edges. In **3-m-in**, the three edges form a single connection such that the outcome varies according to all of the inputs. τ**-1** and **d-20** use the same graph structure as **1-in**.

$$f_r(x_{pa}, x_t) \coloneqq \mathbb{1}(\mathbf{D}[x_{pa}, \tilde{x}_t] \leq \tau)(\frac{b}{\sqrt{d}}\mathbf{C}\tilde{x}_t) + \mathbb{1}(\mathbf{D}[x_{pa}, \tilde{x}_t] > \tau)(\frac{b}{\sqrt{d} \cdot (|\mathcal{P}_r| + 1)}(\mathbf{A}x_{pa} + \mathbf{B}\tilde{x}_t))$$

Here τ is a threshold hyperparameter, such that the value of X_t depends only on $\tilde{x}_t$ if $\mathbf{D}[x_{pa}, \tilde{x}_t] \leq \tau$, but depends on both x_{pa} and $\tilde{x}_t$ if $\mathbf{D}[x_{pa}, \tilde{x}_t] > \tau$. The value of the indicator $\mathbf{D}[x_{pa}, \tilde{x}_t] > \tau$ therefore determines whether the active variables are actual causes or not, and serves as ground truth for our evaluation. When τ is smaller, the active variables are actual causes less frequently.

In the nonlinear case, the graph structure remains the same and we modified the dynamics to utilize a nonlinear test for each component. The nonlinearity was constructed by replacing a matrix operation $\mathbf{A}x$ with an arbitrary nonlinear function $g(\mathbf{A}x)$. In practice, we used a cos function, though a different function could have been chosen. We considered more complex nonlinear relationships, but the results for JOA were not significantly changed compared to the equivalent linear settings, and it is well established that neural networks are universal function approximators Hornik et al. (1989). We obtain the following altered system dynamics:

$$\begin{aligned} f_r(x_{pa}, x_t) \coloneqq & \mathbb{1}(\cos(\mathbf{D}[x_{pa}, \tilde{x}_t]) \leq \tau)(\frac{b}{\sqrt{d}}\cos(\mathbf{C}\tilde{x}_t)) + \\ & \mathbb{1}(\cos(\mathbf{D}[x_{pa}, \tilde{x}_t]) > \tau)(\frac{b}{\sqrt{d} \cdot (|\mathcal{P}_r| + 1)}(\cos(\mathbf{A}x_{pa}) + \cos(\mathbf{B}\tilde{x}_t))) \end{aligned}$$

Aside from having extremely flexible graph connectivity, these domains are convenient since we can generate multiple versions of these graphs with the same connectivity, change the frequency of interactions by using $\mathbf{D}[x_{pa}, \tilde{x}_t] > \tau$, change the dimensionality of the variables, and distinguish multiple independent interactions from jointly dependent interactions (as illustrated in **3-in** vs. **3-m-in**). Additionally, in these domains, there are none of the shortcuts such as spatial proximity that are sometimes used in other RL domains. This allows us to form a clear distinction between methods based on spatial proximity and methods based on context-specific independence, such as JACI.

These domains are inspired by the bipartite causal relationships that characterize Markov Decision Processes (MDPs) and RL. In such settings, the outcome is the result of a state transition $y \to y'$; we write $\tilde{X}_t$ in place of y and Y in place of y' to match the notation for candidate cause and outcome variables used in the rest of this paper. $\tilde{X}_t$ represents passive dynamics and is always a parent of Y. We collect episodes of length 50, and at the end of each episode, the environment resets to a new initialization of random values between $[-1, 1]$ for all variables. A causal variable in the domain is represented with a length 4 vector clipped between $[-1, 1]$. All variables, including the active variables, have passive dynamics, but for simplicity, we only represent passive dynamics for Y in Figure 3. For simplicity, we write the causal relationships as $\mathbf{A}x_{pa} + \mathbf{B}\tilde{x}_t$, but in practice, they are related by affine functions and there is no closed-form solution to solve for the outcome. Figure 3 shows the connectivity of the different graphs evaluated in this work.

When evaluating inference algorithms in these domains, we compute the accuracy using all the conditional edges. For example, in the **2-in** domain we take the accuracy of $X_{a,1} \to Y$ and $X_{a,2} \to Y$ combined. It is

worth noting that not all edges exhibit exactly 50% rates in the dataset, so guessing always 1 or always 0 for an actual cause might result in accuracy near, but not exactly equal to, 50%. In the same way, as the rate of appearance of conditional edges falls to below $< 5\%$, some errors might be cases where the outcome is truly indistinguishable from sampling ambiguity since the model is essentially recovering the decision boundary. It is also possible that the optimal model under Equation 3 will incur some error to minimize the length of the binary vector, and these cases warrant further study as a possible limitation of JACI, or the Random Vectors domain.

The **3-chain** environment is characteristically different from the other environments in that it does not represent a dynamic relationship. As a result, the passive variable replaces the current state as observed by the agent, but as values are computed from the initial node, the current state is modified. The invariant relationships still hold in this case, so it is unclear why the performance is significantly worse. We also assumed that by adding the relationship, the model would occasionally return $X_{a,1}$ instead of $X_{a,2}$ as a cause, but this did not happen. One simple hypothesis is that the chain dynamics were simply harder to predict, making convergence more difficult, but additional experiments will be needed to determine why the error rate for the chain structure remains high.

J JOA Training Details

The primary hyperparameters that JOA uses are the tradeoff rate β between the minimality criterion for binary vectors and forward modeling loss, and the relative learning rate between the forward modeling network and the binary network that guides the joint optimization. Because the optimization algorithm is searching for a solution in a complex space with very strong local minima (such as $h_{\mathbf{Y}}$ always outputting $\mathbf{0}$ or $\mathbf{1}$), as well as strong constraints induced by the network structure, these hyperparameter values must be carefully chosen. For the **1-in** domain, Figure 4a shows the rate of false positives and false negatives, and Figure 4b shows the error rate during training.

Hand-tuning a fixed β is often impossible and does not always result in recovering the true partitions. Instead, we use an adaptive β based on the following insight: when $\|f(\mathbf{s}^{(i)}, b^{(i)}; \phi) - \mathbf{y}^{(i)}\|$ is large, this indicates that the forward model is not performing well, and masking out additional information is counterproductive since the network probably does not have enough information to determine the outcome. This could be either because of masking or because it is early in training. Instead of a fixed tradeoff value, JOA uses an adaptive tradeoff rate given by $\beta = \widehat{\beta} e^{-\|f(\mathbf{s}^{(i)}, b^{(i)}; \phi) - \mathbf{y}^{(i)}\|}$, where $\widehat{\beta}$ is a hyperparameter. In practice, the inference accuracy of JOA is somewhat robust to values of $\widehat{\beta}$, and we use a value between $1 - 10$ depending on the domain.

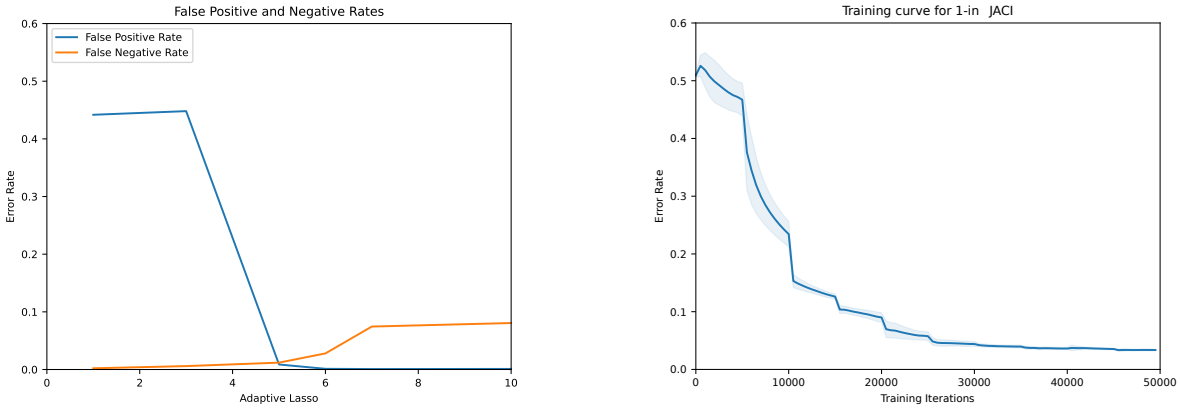
Additionally, the relative learning rates between the forward modeling and binary networks must be tuned to avoid local minima. If the binary network is trained too quickly, it may converge on behavior that only predicts based on the all 0 or all 1 binary, which is a local minima that the forward model cannot escape. Instead, the forward modeling and binary networks must be trained together so that each network can adapt to gradual changes in the other. In practice, this amounts to alternating once per training step between the forward modeling and binary networks, with the learning rate of the binary network initialized to 1/10th that of the modeling network. However, further strategies for this alternation could yield better results. Other hyperparameters and their sensitivity are described in Table 10.

J.1 Compute

While the network architecture for training JOA is fixed across runs, the compute time differs based on the number of objects, which affects the size of the 1D convolutions. Training JOA to convergence took approximately 100K steps with a batch size of 512. Using a TitanX GPU with one CPU results in 7 hours of training for a single run of the **1-in** Random Vector domain, and 14 hours of training for the Spriteworld-6 domain. Other domains varied within this range. We executed 5 trials of JOA on each domain on a 64CPU server with 4 TitanX GPUs and 126Gb of RAM. Note that we ran a substantial number of jobs in parallel since at least 4 jobs could be run on the same GPU.

Env	JOA	Best Baseline	Smallest Class	p-baseline	p-smallest
1-in-linear	3.6 ± 0.1	8.8 ± 3.6	48 ± 0.1	$1.60\text{e}-02$	$1.24\text{e}-11$
2-in-linear	2.6 ± 2.0	27 ± 4.0	50 ± 2.0	$1.30\text{e}-04$	$1.51\text{e}-06$
1-in-nonlinear	3.2 ± 0.1	20 ± 2.1	49 ± 0.1	$2.88\text{e}-05$	$1.09\text{e}-11$
2-in-nonlinear	4.0 ± 1.0	35 ± 3.7	43 ± 1.0	$2.75\text{e}-05$	$2.07\text{e}-07$
3-in	4.5 ± 1.9	41 ± 3.8	47 ± 1.9	$2.16\text{e}-05$	$1.91\text{e}-06$
3-chain	9.3 ± 7.6	46 ± 3.2	47 ± 7.6	$2.86\text{e}-04$	$7.14\text{e}-04$
3-m-in	2.0 ± 0.7	17 ± 3.5	50 ± 0.7	$3.57\text{e}-04$	$2.17\text{e}-08$
d-20	7.7 ± 1.2	$\mathbf{8.0 \pm 0.4}$	47 ± 1.2	$\mathbf{3.12\text{e}-01}$	$4.16\text{e}-07$
τ-1	0.2 ± 0.0	2.1 ± 0.3	10 ± 0.0	$7.22\text{e}-05$	$5.20\text{e}-17$
2D Pushing	13 ± 1.3	22 ± 1.1	42 ± 1.3	$1.47\text{e}-04$	$1.93\text{e}-06$
Breakout	3.6 ± 0.2	$\mathbf{10 \pm 7.1}$	35 ± 0.2	$\mathbf{5.71\text{e}-02}$	$7.90\text{e}-10$
Spriteworld-2	6.6 ± 1.2	16 ± 0.5	35 ± 1.2	$4.28\text{e}-05$	$1.52\text{e}-06$
Spriteworld-3	11 ± 1.7	21 ± 0.1	34 ± 1.7	$9.71\text{e}-05$	$1.41\text{e}-05$
Spriteworld-6	6.3 ± 1.3	12 ± 2.0	35 ± 1.3	$2.96\text{e}-03$	$2.01\text{e}-06$

Table 9: P-value significance tests computed according to the best baseline (p-baseline) or predicting the smallest class (p-smallest). Bolded lines are within $> 5\%$ p-value



(a) (b)

Figure 4: **(a)** Comparison of false positive and false negative rates in the **1-in** domain. The blue line indicates false positive rate (the model guesses an actual-cause judgment when there is none) and the orange line indicates false negative rate (the model guesses no actual-cause judgment when there is one). **(b)** Train curve averaged over 10 runs for **1-in**. Error rate decreases monotonically, even though JOA optimizes Equation 3 instead of the error rate explicitly.

J.2 Significance Tests

We use datasets of 1M transitions in each domain, splitting 90% for training and 10% for testing at random. To ensure that interactions are identified in low-interaction regimes, we upsample interaction states at evaluation.

We provide some basic p-values using a t-test for the JOA results obtained with 5 trials, and comparing against the null hypotheses of (1) always selecting the smallest class with a standard deviation equal to that of JOA (2) the best-performing baseline. Our results are illustrated in Table 9

After extensive hyperparameter tuning on NCD with a regularizer, labeled **NCD-Reg** with a grid search over 40 different values of the lasso value between 0.0001 to 10.0, adaptations to the network architecture and environment specific values of lasso, we did not replicate the results from the associated work (Hwang et al., 2023). Additionally, we tried out an adaptive value of lasso, but could not get improved performance.

Parameter	Values	Sensitivity
Forward Model Learning rate	0.001-0.003	Significant if too high
Interaction Model Learning rate	0.0001-0.0007	Significant
Adaptive lasso rate	1-10	Significant, see Figure 4a
Network layers	6	Fixed across all tasks
Network width	256	Fixed across all tasks
Embedding width	1024	Fixed across all tasks
Reduction Function	sum	Fixed across all tasks
Number of Frames	100k, 500k	Random Vectors or Breakout/Robot Pushing
Number of Training Iterations	200k	Fixed across all tasks
Batch Size	512	Fixed across all tasks, sensitive if too low
Threshold for AC	0.99	Fixed across all tasks

Table 10: Hyperparameters

We think that some of the poor performance of regularized NCD in some cases could be because we were unable to find the right value of lasso lambda. Other differences might be attributable to the differences in domains and data volume compared to NCD.

K Baseline Details

We compare against two baselines from the RL literature and one from the graphical models literature, as well as one baseline that enforces the necessity property based on counterfactual methods from the RL literature.

K.1 Gradient Baseline

This method is explored in some prior work (Pitis et al., 2020; Wang et al., 2023), and has shown promising results, especially in quasi-static domains where there is a significant change that occurs as a result of interactions. It takes the magnitude of the gradient of the inputs with respect to the output ($g_{i,\mathbf{y}}$) for each variable in a particular state, given by $g_{i,\mathbf{y}} = \left\| \frac{\delta \mathbf{y}}{\delta \mathbf{s}_i} \right\|_1$. In practice, when learning from observational data this gradient is of a learned forward model $f(\mathbf{s};\phi) \to \mathbf{y}$. For a given threshold τ_g, this method assigns an actual cause based on the magnitude of $g_{i,\mathbf{y}}$ for causal variable i.

$$\mathbf{b}_i := \mathbb{1}\left(\left\| \frac{\delta \mathbf{y}}{\delta \mathbf{s}_i} \right\|_1 > \tau \right) \tag{6}$$

However, tuning the magnitude of τ_g can be very challenging as it is domain and sometimes interaction-specific. This observation is corroborated by our results. In this work, we tune the threshold magnitude using oracle knowledge of the actual causes, by choosing the midpoint between the mean $g_{i,y}$ for the true actual causes and the mean $g_{i,y}$ for the true non-causes. This gives good results for cases with a single actual cause since τ_g can be perfectly tuned for the dynamics of that cause. However, the performance declines as the number of causes increases, since finding a threshold that holds for all of these dynamics becomes increasingly challenging. We experimented with regularized gradients as suggested in Wang et al. (2023), which reduces the size of the gradients by applying a cost function to the model $f(\mathbf{s};\phi) \to \mathbf{y}$:

$$\mathcal{L}_{\text{grad}}(f, \mathbf{s}; \phi) := \lambda_{\text{ grad}} \left\| \frac{\delta f(\mathbf{s};\phi)}{\delta \mathbf{s}_i} \right\|_1 \tag{7}$$

However, this regularization did not significantly improve results. This is not unexpected; the error rates we found are also comparable with those observed in these works. This baseline does have some additional advantages when compared with other baselines. While neural networks in high dimensions appear to overfit to some extent, as shown by the performance drop in the **d-20** environment in Table 2), this dropoff is not as significant when using gradient-based inference. On top of this, the gradients can be computed without

making any significant changes to the model or training procedure. However, the loss in performance makes its application limited for downstream tasks.

K.2 Attention Baseline

With the success of attention-based networks such as Transformers Vaswani et al. (2017), the question naturally arises whether attention weights inherently capture actual causes. Prior work has used attention to identify relevant tokens Hafiz et al. (2021). Using each of the causal variables as tokens, we can train an attention model and determine if the attention provides a meaningful signal of the actual causes.

For this baseline, we implemented an attention network that uses multi-head attention over the input variables when predicting the outcome variable. For each attention head $a_i^k \in \{1, \ldots, H\}$, this method takes the average of the length N vector where every index corresponds to one of the causal variables (which we represent with a_i^k in abuse of notation). These are then incorporated in a multi-head attention strategy to infer the state of the outcome variable. To determine whether a variable is an actual cause, for each variable X_i we compute

$$\mathbf{b}_i \coloneqq \mathbb{1}\left(\frac{\sum_{k=1}^{N} a_i^k}{N} > \tau\right) \tag{8}$$

If the ratio of heads τ is greater than $1/N$, that variable is considered an actual cause. This represents the attention weight on that particular input being greater than some amount. We performed some tuning to determine if a better threshold value for the ratio of heads could be determined but found this method to struggle regardless of what ratio was used. If we just used an "at least one head" instead of a soft measure, this methoud would infer every variable as an actual cause. This is analogous to the idea of $\alpha-$necessity introduced in Appendix G. In practice, this baseline still performs poorly because the heads rarely differentiate between the inputs enough to provide meaningful assessments of the actual cause. This lack of differentiation we observed could be due to training on only hundreds of thousands of tokens instead of trillions of tokens used in many other works using attention networks.

To improve the performance, we altered the training procedure of the attention model by introducing an entropy loss over the attention of the input on the heads, so that the model would learn to differentiate between the inputs at least to some extent. This loss has the form

$$\mathcal{L}_{\text{attn}}(a_i) \coloneqq -a_i \log(a_i) \tag{9}$$

In practice, this prevented the network from converging to flat attention heads that put equal attention on all inputs.

K.3 Counterfactual Baseline

This baseline takes a generative approach to enforcing the necessity property and identifying the actual causes, and is corroborated in counterfactual-based techniques Pitis et al. (2020); Choi et al. (2023). This is done by first learning a forward model $f(\mathbf{s}; \phi) \to \mathbf{y}$. Then for each causal variable X_i, it compares the observed outcome with the output of the forward model for random interventions on X_i. We can represent the distribution of counterfactuals on variable X_i in state $\mathbf{s}$ as $C_i(\mathbf{s})$. Then the comparison is given by $E_{\mathbf{v}' \sim C_i(\mathbf{s})} \| f(\mathbf{v}'; \phi) - y \|_1$. In practice, we use samples from a uniform distribution over the normalized state space of the causal variable. If the expected outcomes differ by more than a threshold τ_{CF}, then the value is considered an actual cause. In essence, this is analogous to α-necessity, where at least τ_{CF} differences should be observed to consider this a cause. Since the values of random vectors are constrained between $[-1, 1]$, generating the intervention can be done by replacing the state of a variable X_i with a length 4 random vector drawn from that range. The resulting binary vector is computed as:

$$\mathbf{b}_i \coloneqq \mathbb{1}\left(\frac{1}{N} \sum_{n=1}^{N} \| f(\mathbf{s}_i^{(n)}; \theta) \|_1 > \tau_{\text{CF}}\right) \tag{10}$$

In practice, this method does not often get useful results and often overestimates actual causes. Part of this could be difficulty with determining a good value for τ_{CF}, the threshold parameter since this value can fluctuate over different training seeds. Additionally, it is difficult to generate meaningful counterfactuals in domains like Breakout, where certain states are not actually possible. However, the primary flaw in this method is that networks trained with observational values can often struggle with counterfactual values due to overfitting. The poor performance of this method is also what dissuaded us from explicitly enforcing $\alpha-$necessity in JOA.

K.4 Neural Contextual Decomposition

Neural Contextual Decomposition (NCD) Hwang et al. (2023) is a method for discovering context-specific independence, and uses a similar architecture of a binary mapping model $h_{\mathbf{Y}} : \mathcal{R}(\mathbf{S}) \to \{0,1\}^n$ parameterized by θ (where n is the number of endogenous variables) feeding into a prediction module $f : \mathcal{R}(\mathbf{S}) \times \{0,1\}^n \to \mathcal{R}(\mathbf{Y})$ parameterized by ϕ, where the output of $h_{\mathbf{Y}}$ is used to mask the input. Unlike JOA, however, NCD simply trains this architecture with the prediction loss, using the following objective:

$$\theta, \phi := \underset{\theta,\phi]}{\arg\min} \sum_{i=1}^{N} \left\| f(\mathbf{s}^{(i)}, h_{\mathbf{Y}}(\mathbf{s}^{(i)}; \theta); \phi) - \mathbf{y}^{(i)} \right\| . \tag{11}$$

This regression does not utilize the necessity-inducing 1D-convolution structure used by JOA (Figure 2b), or the explicit joint optimization using the regularized mask. Instead, NCD simply uses a fully connected network with the mask alongside the input.

In our domains, NCD performs poorly. Part of this is because in many domains, NCD will tend towards the local minima of predicting $\mathbf{1}$, the binary vector of all ones. This is a result we observed when investigating JOA as well, and it is not clear that without joint optimization an alternative outcome is likely to be produced. To further investigate NCD, we included joint optimization by adding $\|h_{\mathbf{Y}}(\mathbf{s})\|_1$ to the objective, which leads to improved performance. However, Hwang et al. (2023) explicitly mentions regularization was not used besides weight decay.

L Complex Domains

To evaluate how well JOA scales to more realistic domains with complex dynamics, we use the *Mini-Breakout, 2D Robot Pushing, and Spriteworld* environments. In each domain, the simulator indicates when objects interact and can be used to obtain ground truth actual-cause judgments.

L.1 2D Robot Pushing Domain

The causal variables in this domain are Action, Pusher, Block, Target, Obstacle1, Obstacle2, and Obstacle3, where Block is the outcome variable. Each variable is represented by a 3-dimensional continuous vector, where the first two dimensions encode the midpoint of the object, and the third dimension is used to indicate when the block is on the target; this value is fixed to 0 for non-target objects. Each of the objects occupies a 1x1 axis-aligned bounding box.

Actions in this domain move the pusher $[-1, 1]^2$ units from its current position, as long as the pusher remains within the $[5, 5]$ bounds. If the pusher collides with an obstacle, it will be stopped at the first edge of contact. Motion perpendicular to the obstacle will slide the pusher along its edge. If the pusher collides with the block, it will push the block along the first edge of contact. If the block is pushed into an obstacle, the block is stopped at the first edge of contact with the obstacle, and the push is blocked along the edge of the contacting edge of the block in the axis of the obstacle. The block and pusher are not impeded by the target. The causal graph is illustrated in Figure 5 (right).

To generate data for this domain, the object positions are initialized to random, non-intersecting locations. In each episode, the pusher takes random actions for 20 timesteps, after which the positions of the obstacles, agent, target, and block are re-initialized to random locations. At each timestep, the current state of the block

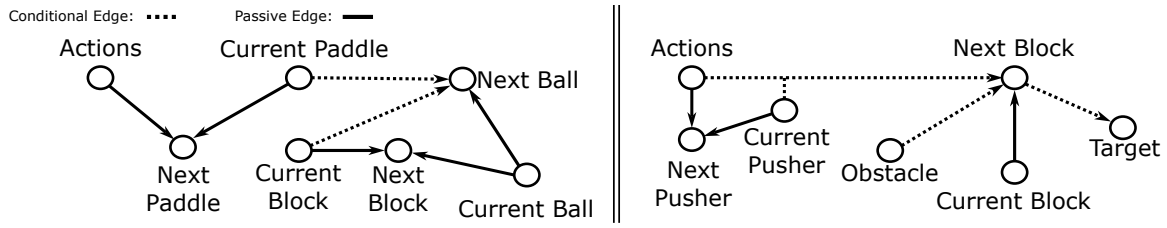


Figure 5: Causal graphs for environment dynamics in Mini Breakout (left) and Robot Pushing (right), where transitions are from current positions to next positions. Note objects that do not move still exhibit dynamics (next position = current position), they are just omitted from the graph. A conditional edge is one that exhibits context-specific independence.

is an actual-cause judgment of its next state. However, the action and the pusher state are also actual-cause judgments if the action had put the pusher in collision with the block. Similarly, the obstacle state is an actual-cause judgment if it impedes the block. This domain is challenging because the behavior of the block remains the same whether there are no actual-cause judgments or if the block is completely impeded by the obstacle.

Since interactions are infrequent in this domain, the previous state of the block is the most frequent actual-cause judgment and other non-trivial actual-cause judgments are rare. By default, the 2D pusher interacts with the block less than 10% of the time. To get meaningful statistics for the baselines, we resample states at test time such that non-trivial actual-cause judgments occur $\sim 40\%$ of the time. Since JOA does not assume access to the true actual-cause judgments, in training, we simulate the upsampling effect using a *passive model*, which uses only the previous state of the block as input. Since we used the passive model for sampling, JOA training never uses the true actual-cause judgment. We then upsample states where the passive model performs poorly. This method is inspired by Chuck et al. (2023).

L.2 Mini Breakout Domain

The causal variables in this domain are Action, Paddle, Ball, Block1, Block2, and Block3, where Ball is the outcome variable. Each variable is represented with a 5-dimensional continuous vector where the first two dimensions encode midpoint position, the next two encode velocity, and the last dimension encodes additional information; for blocks, this value would indicate whether the block was hit or not. The size of the domain is 30x50, where the paddle is 7x3, the ball is 2x2 and the blocks are 5x2.

Actions in this domain are discrete, where the paddle can move 2 units to the right or left, or remain stationary. The ball has a velocity in $\{-1, 1\} \times \{-2, 2\}$, and depending on which position it strikes, it will receive an upward velocity of $(1, 1), (1, 2), (-1, 2), (-1, 1)$. To make the ball bounce independently of the current action, the angle is computed before the paddle moves. Bouncing off of the blocks will reverse the y velocity of the ball, and bouncing off of a wall will flip the ball velocity perpendicular to the wall. Otherwise, the ball will be updated by adding the velocity to the position every timestep. If a block is hit, the last dimension of its state vector will be set to 0 and it can no longer affect the ball. The causal graph for this domain is illustrated in Figure 5 (left).

Data for this domain is generated by randomly initializing the ball with a y-velocity of -1, a random x-position and x-velocity in $[-1, 1]$, and a y-position randomly in $[22, 28]$ (below the blocks). The blocks are initialized to random integer positions in $[30, 45]$, and the paddle is initialized at the position $(15, 10)$. The environment resets if the ball hits the bottom or all the blocks are hit, and the agent takes random actions until one of the reset criteria is met. As in the 2D Robot Pushing domain, interactions occur less than 1% of the time and non-trivial actual-cause judgments are rare. As a result, we enforce the same upsampling methods to get meaningful performance. At test time, we once again sample states such that non-trivial actual-cause judgments occur $\sim 35\%$ of the time.

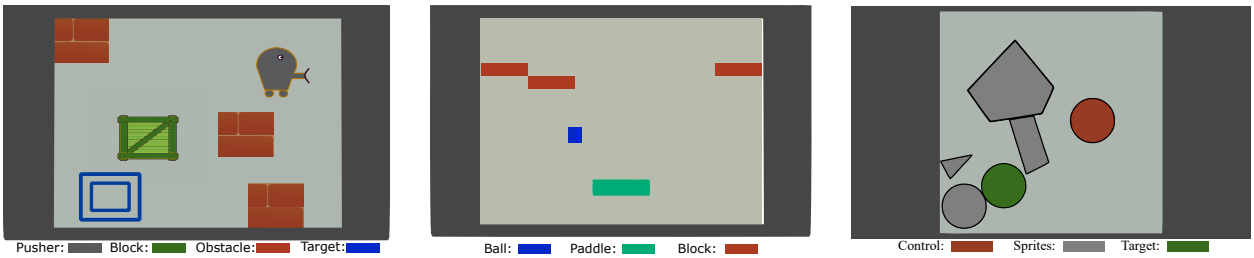


Figure 6: **(a) 2D Pushing**: continuous actions in $[-1, -1]^2$ move the pusher in a 5×5 area. **(b) Mini-Breakout**: the actions move the paddle to the right or left. (c)**SpriteWorld**: Actions apply forces to the control sprite.

L.3 Spriteworld

The Spriteworld domain Watters et al. (2019) was introduced in the context of Reinforcement Learning and typically involves sprites that can be individually perturbed using specific actions. In this work, we modify the Spriteworld domain by allowing for object collisions using a 2D physics engine Catto (2020) in a similar fashion to (Hwang et al., 2023). However, because the code for Hwang et al. (2023) was not available at the time of writing this paper, our domain is not exactly the same as that work.

The causal variables in this domain are Action, Control, Sprite1,..., SpriteK, and Target, where we vary the number and characteristics of the sprites. Target is the outcome variable, which is a green circle as seen in Figure 6. Control is a red circle that receives actions as a 2D force vector between 0N and 100N. Each variable is represented with a 5-dimensional continuous vector, where the first two dimensions encode the midpoint position, the next two encode velocity, and the last dimension encodes the radius. The size of the domain is $10m \times 10m$, where the target and control objects have radius $1m$. At the boundaries, the objects will collide with the walls and stop moving.

We test three different settings of this environment. **Spriteworld-2** has only two objects, Control and Target, and no other sprites. **Spriteworld-3** includes one other Sprite, Sprite0. Finally, **Spriteworld-6** includes 4 other sprites, totaling 6 objects. A state from this world is rendered in Figure 6.

Data for this domain is generated by randomly initializing the positions and velocities of the Control, Sprite and Target objects, then rolling out for 50 time steps. While this induces a large number of overall interactions, interactions between the target object and other objects are still fairly rare, especially in settings with fewer objects. occur around 5% of the time. Therefore at test time, we once again sample states such that non-trivial actual-cause judgments occur $\sim 30\%$ of the time.

M Exhaustive Search

We include a more in-depth description of our exhaustive search algorithm, previously described in Section 5. While this algorithm simply enumerates all possible subset-binary pairs and tests them for the desired properties, its implementation can still be involved. To provide additional context to the 1D mover example in Section 5, Figure 7b illustrates the distribution of valid partitions of the state for different states. While many valid binary-subset pairs exist, only a tiny subset of pairs minimizes cost. These correspond to the few assignments of actual-cause judgments that capture intuitions such as normality and necessity.

N Results of Algorithm 0 on AC Examples

In this section, we compare the results of Algorithm 0 with existing verdicts of actual causation in the literature. In prior work on actual causality, proposed definitions are usually compared by assessing their verdicts on a set of examples from the philosophy literature, for which there are commonly accepted answers to serve as ground truth. The examples we consider are:

Algorithm 0 Exhaustive Search

Input: SCM M, range function $\mathcal{R}$, ratios (α_0, α_1)
Output: Set of min-cost state-binary mappings $h_{\mathbf{Y}}$ and binary-subset pairs $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$
for all states $\mathbf{s} \in \mathcal{R}(\mathbf{S})$:
  let $g_{\mathbf{b}}(\mathbf{s}) :=$ set of all binaries for which $\mathbf{s}$ satisfies necessity under any witness
for all subsets of states $\mathcal{S} \in 2^{\mathcal{R}(\mathbf{S})}$:
  let $g_s(\mathcal{S}) := \bigcap_{\mathbf{s} \in \mathcal{S}} g_{\mathbf{b}}(\mathbf{s})$ ▷ Set of binaries for which all states in $\mathcal{S}$ satisfy necessity under any witness
  remove all binaries from $g_s(\mathcal{S})$ that do not satisfy invariance
let $\mathcal{L}_{min} = \infty$ and $\mathcal{H} = \emptyset$
for all possible partitionings of the state space $\mathcal{P} \in \mathcal{P}(\mathbf{S})$
  for all binary-partition pairings $(\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ from $\mathcal{P}$ and $g_s(.)$
    if all binaries in $\mathcal{B}(\mathbf{Y})$ are unique:
      for all $(\mathcal{I}_j, \mathbf{b}_j) \in (\mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))$ **let** $h_{\mathbf{Y}}(\mathbf{s}) := \mathbf{b}_j \ \forall \mathbf{s} \in \mathcal{I}_j$ ▷ Get state-binary mapping
      let $\mathcal{L} := \sum_j |\mathbf{b}_j||\mathcal{I}_j|$
      if $\mathcal{L} < \mathcal{L}_{min}$ **set** $\mathcal{L} = \mathcal{L}_{min}$ and $\mathcal{H} = \{(h_{\mathbf{Y}}, \mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))\}$ ▷ New min cost solution
      else if $\mathcal{L} = \mathcal{L}_{min}$ set $\mathcal{H} = \mathcal{H} \cup \{(h_{\mathbf{Y}}, \mathcal{B}(\mathbf{Y}), \mathcal{I}(\mathbf{Y}))\}$ ▷ Append to min cost solution set
return set of solutions $\mathcal{H}$

m	o	m'	$\mathbf{b}$	m	o	m'	$\mathbf{b}$
0	1	0	11	2	0	3	10
1	2	1	11	2	1	3	10
0	0	1	10	2	2	3	10
0	2	1	10	3	0	4	10
1	0	2	10	3	1	4	10
1	1	2	10	3	2	4	10

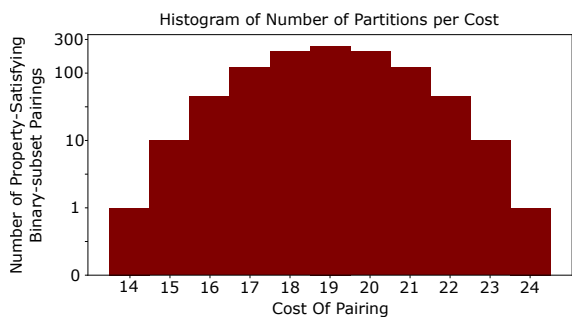


Figure 7: Results of Algorithm 0 in 1D mover environment. Abbreviations: $\mathbf{b}$: binary [mover, obstacle], m: mover, o: obstacle, m': next mover position. 1D mover dynamics: $m' = m + 1$ if $o \neq m + 1$, otherwise $m' = m$. **(a)** Table of binary vector, state and outcome for the mover environment. The obstacle is only identified as a cause when it blocks the mover. **(b)**: Histogram of costs and number of valid binary-subset pairs.

1. 1D Mover
2. Forest Fire (Example 7.1.4, Halpern (2016))
3. Gang Execution (Section 4, Rosenberg & Glymour (2018))
4. Halt and Charge (Example 2.3.7, Halpern (2016))
5. Switching Railroad Tracks (Example 2.3.6, Halpern (2016))
6. Binary AND ($C = A \wedge B$)
7. Binary OR ($C = A \vee B$)
8. Binary XOR ($C = A \oplus B$)
9. Rock Throwing (Example 2.3.3, Halpern (2016))

Unlike existing methods, we evaluate not only the actual causes of a particular state and outcome but also the actual causes for every combination of states and outcomes. This is by construction, since verifying that the invariant-set actual-cause judgment definition (Definition 4.7) is satisfied requires examining the full state space and constructing a binary-subset pair. Each table corresponds to a minimal binary-subset pair, and each row indicates the inferred actual-cause judgment for a particular state and outcome. To compare against existing definitions, we consider the observed state that corresponds to what happened in the example from the literature and look at its corresponding row in the tables to see if the agreed-upon answer matches with the binary vector returned by Algorithm 0.

Note that because Algorithm 0 returns *all* minimal cost subset-binary pairings, there may be multiple tables for a single example. This ambiguity appears to be inherent to the actual cause problem: If there are multiple ways of arriving at a dependency, then any one way could be an actual cause, though the combination of all of them would not be a minimal actual cause. This property is not unique to IAJ—prior definitions of actual cause infer multiple minimal actual causes of a particular event. The implications of this ambiguity are particularly pronounced in examples like Rock Throwing, and we can see this reflected in the number of tables returned.

We also provide the α_0, α_1 values used in Algorithm 0 to generate the subsequent tables. The meaning of these terms is discussed in detail in Appendix G. For the 1D mover example, we used $\alpha_0 = 0.4, \alpha_1 = 0.01$.

N.1 1D Mover

In the mover example, the mover m will move to position $m' = m + 1$ unless obstructed by obstacle o, in which case it will remain at $m' = m$. In this setting, m, o are always HP actual causes of the next mover position individually, since both can counterfactually placed at position $m + 1$ (or counterfactually moved away). This example exhibits a distinction between the HP actual cause and the IAJ actual-cause judgments. The mover can be in locations 0-2, the obstacle in positions 1-3 and the next mover in positions 0-3.

Binary	Mover	Obstacle	Mover-Next
10	0	2	1
10	0	3	1
10	1	1	2
10	1	3	2
10	2	1	3
10	2	2	3
01	0	1	0
01	1	2	1
01	2	3	2

Binary	Mover	Obstacle	Mover-Next
10	0	3	1
10	1	1	2
10	1	3	2
10	2	1	3
10	2	2	3
01	0	1	0
01	0	2	1
01	1	2	1
01	2	3	2

Binary	Mover	Obstacle	Mover-Next
10	0	2	1
10	0	3	1
10	1	1	2
10	2	1	3
10	2	2	3
01	0	1	0
01	1	2	1
01	1	3	2
01	2	3	2

Binary	Mover	Obstacle	Mover-Next
10	0	3	1
10	1	1	2
10	2	1	3
10	2	2	3
01	0	1	0
01	0	2	1
01	1	2	1
01	1	3	2
01	2	3	2

N.2 Forest Fire

The binary variables *April*, *May*, and *June* indicate whether or not there was rain in April, electrical storms in May, and electrical storms in June. The outcome $Fire = 1$ indicates a forest fire in May, and $Fire = 2$ indicates a forest fire in June. All tables indicate that when there was rain in April as well as electrical storms in May and June, the storm in May was not an actual cause. However, the rain in April was not an actual cause on its own, and the storms in June are also included in the actual cause. These tables are generated using $\alpha_0 = 0.2, \alpha_1 = 0.01$, where 0.01 is used as a default α_1 value to represent contrastive necessity without change.

Binary	April	May	June	Fire
011	0	0	0	0
011	1	0	0	0
011	0	0	1	2
110	0	1	0	1
110	0	1	1	1
101	1	1	0	0
101	1	0	1	2
101	1	1	1	2

Binary	April	May	June	Fire
110	0	1	0	1
110	0	1	1	1
011	0	0	0	0
011	1	0	0	0
011	0	0	1	2
011	1	0	1	2
101	1	1	0	0
101	1	1	1	2

Binary	April	May	June	Fire
011	0	0	0	0
011	0	0	1	2
110	0	1	0	1
110	0	1	1	1
101	1	0	0	0
101	1	1	0	0
101	1	0	1	2
101	1	1	1	2

Binary	April	May	June	Fire
110	0	1	0	1
110	0	1	1	1
011	0	0	0	0
011	0	0	1	2
011	1	0	1	2
101	1	0	0	0
101	1	1	0	0
101	1	1	1	2

N.3 Gang Execution

The binary variables *Gang* and *Leader* indicate whether or not the gang member or the gang leader respectively shot the victim, and the outcome variable *Death* indicates whether the victim died. In all cases, the gang leader is declared as the actual cause, which is in agreement with the conclusions of Rosenberg & Glymour (2018). This table is generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	Gang	Leader	Death
01	0	0	0
01	1	1	1

N.4 Halt and Charge

In this example, orders from higher-ranking officers could trump those of lower-ranking officers. The $Major$ can order the corporal to halt or charge (these are indicated by values 0 and 1 respectively), or choose not to make an order (indicated by a value of 2). If the $Major$ makes no order, the corporal will defer to the order of the $Sergeant$, where 1 indicates an order to charge and 0 indicates an order to halt. The outcome $Corporal = 1$ indicates that the corporal charged.

The results show that any time the major gives the order to halt or charge, the major is the sole actual cause since its order trumps the sergeant. However, when the major chooses to abstain and the corporal follows the order of the sergeant, the actions of the major and the sergeant are jointly responsible for the outcome. These tables are generated using $\alpha_0 = 0.3, \alpha_1 = 0.01$.

Binary	Major	Sergeant	Corporal
10	0	1	0
10	1	0	1
10	0	0	0
10	1	1	1
11	2	0	0
11	2	1	1

N.5 Binary And

The outcome is zero when either A or B is zero. When only one of them is zero, the variable that is equal to zero is declared as the actual cause. When both of them are zero, they are jointly considered the actual cause, since they both have to change for the outcome to change to 1.

The outcome is equal to 1 only when both A and B are equal to 1. In this case, we see that there are two minimal binary-subset pairs; one declares A as the actual cause and the other declares B as the actual cause. This reflects the ambiguity over which of the two is responsible when the outcome is determined by both jointly. Prior definitions of actual cause would also declare either of them to be actual causes by putting the other in the witness set. These tables are generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	A	B	C
11	0	0	0
10	0	1	0
01	1	0	0
01	1	1	1

Binary	A	B	C
11	0	0	0
01	1	0	0
10	0	1	0
10	1	1	1

N.6 Binary Or

The outcome is equal to zero only when both A and B are equal to zero. Similar to Binary AND, we have two minimal binary-subset pairs; one declares A as the actual cause and the other declares B as the actual cause. These tables are generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	A	B	C
01	0	0	0
01	0	1	1
10	1	0	1
11	1	1	1

Binary	A	B	C
01	0	1	1
10	0	0	0
10	1	0	1
11	1	1	1

N.7 Binary XOR

The outcome is 1 when only one of A or B is equal to 1, and the other is zero. When both A and B have the same value, the outcome is zero. In all cases, both A and B are individually minimal actual causes of the outcome, and similar to Binary AND and OR we have two minimal binary-subset pairs such that either A or B is the actual cause. These tables are generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	A	B	C
01	0	1	1
01	1	0	1
10	0	0	0
10	1	1	0

Binary	A	B	C
10	0	1	1
10	1	0	1
01	0	0	0
01	1	1	0

N.8 Rock Throwing

The variables in the model are Suzy Hits (SH), Suzy Throws (ST), Billy Hits (BH), Billy Throws (BT), and Bottle Shatters (BoS). The results show that whenever Suzy throws, her throw is always the sole actual cause whether or not Billy throws. However, the equations in the original example are such that whenever Suzy throws her rock, it will always hit the bottle (i.e. SH = ST). As a result, there is an ambiguity between $SH = 1$ and $ST = 1$ as candidate actual causes. Our method returns all minimal-binary subset pairs, which encompasses both of these valid answers. Similarly, when Billy is the only one who throws and his rock hits the bottle, both $BH = 1$ and $BT = 1$ are candidate actual causes.

However, when neither of them throws and the bottle does not shatter, any of $ST = 0, SH = 0, BT = 0$ and $BH = 0$ could be actual causes, since any one of them throwing a rock at the bottle would have caused it to shatter. Since there are more ambiguities and valid answers in this example, the number of minimal binary-subset pairs increases, and this is reflected in the number of tables returned below. These tables are generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	SH	ST	BH	BT	BoS
0010	0	0	0	0	0
0010	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	0	0	0
0001	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0010	0	0	0	0	0
0010	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0010	0	0	1	1	1
0100	0	0	0	0	0
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	0	0	0
0001	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0010	0	0	1	1	1
1000	0	0	0	0	0
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	1	1	1
0100	0	0	0	0	0
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	1	1	1
1000	0	0	0	0	0
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	0	0	0	0	0
0010	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	0	0	0	0	0
0001	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	0	0	0	0	0
0010	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	0	0	0	0	0
0001	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0010	0	0	0	0	0
0001	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0010	0	0	0	0	0
0001	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	0	0	0
0010	0	0	1	1	1
0100	1	1	0	0	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0001	0	0	0	0	0
0010	0	0	1	1	1
1000	1	1	0	0	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	0	0	0	0	0
0100	1	1	0	0	1
0010	0	0	1	1	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	0	0	0	0	0
0100	1	1	0	0	1
0001	0	0	1	1	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	0	0	0	0	0
1000	1	1	0	0	1
0010	0	0	1	1	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	0	0	0	0	0
1000	1	1	0	0	1
0001	0	0	1	1	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	1	1	0	0	1
0010	0	0	0	0	0
0010	0	0	1	1	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	1	1	0	0	1
0001	0	0	0	0	0
0001	0	0	1	1	1
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	1	1	0	0	1
0010	0	0	0	0	0
0010	0	0	1	1	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	1	1	0	0	1
0001	0	0	1	1	1
1000	0	0	0	0	0
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	1	1	0	0	1
0001	0	0	0	0	0
0001	0	0	1	1	1
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	1	1	0	0	1
0010	0	0	1	1	1
0100	0	0	0	0	0
0100	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
0100	1	1	0	0	1
0010	0	0	1	1	1
1000	0	0	0	0	0
1000	1	1	0	1	1

Binary	SH	ST	BH	BT	BoS
1000	1	1	0	0	1
0001	0	0	1	1	1
0100	0	0	0	0	0
0100	1	1	0	1	1

N.9 Switching Railroad Tracks

The model in the original example asserts that the train will reach its destination if there is no breakdown ($T \neq 2$), but there is no affordance in the model for breakdowns to actually occur. For this reason, we add another binary variable $Break$ to the model and set $Track$ to 2 if $Break = 1$.

The results show that whenever there is a breakdown, both $Break = 1$ and $Track = 2$ are returned as valid actual causes since as shown above, we always have $Track = 2$ when $Break = 1$. When there is no breakdown, the train reaches the station regardless of which track it arrived on, so as before we have the assignments to $Break$ and $Track$ as valid actual causes. In all tables below, the switch is never an actual cause. These tables are generated using $\alpha_0 = 1.0, \alpha_1 = 0.01$.

Binary	Break	Switch	Track	Arrive
100	0	0	0	1
100	0	1	1	1
100	1	0	2	0
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	0	0	1
100	0	1	1	1
100	1	0	2	0
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	0	0	1
001	0	1	1	1
001	1	0	2	0
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	0	0	1
100	0	1	1	1
001	1	0	2	0
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	0	0	1
001	0	1	1	1
001	1	0	2	0
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	0	0	1
001	0	1	1	1
100	1	0	2	0
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	1	1	1
001	0	0	0	1
001	1	0	2	0
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	1	1	1
100	0	0	0	1
100	1	0	2	0
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	0	0	1
100	0	1	1	1
100	1	0	2	0
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	0	0	1
001	0	1	1	1
001	1	0	2	0
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	1	1	1
100	1	0	2	0
001	0	0	0	1
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	1	1	1
001	1	0	2	0
100	0	0	0	1
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	0	0	0	1
100	1	0	2	0
001	0	1	1	1
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	0	0	0	1
001	1	0	2	0
100	0	1	1	1
100	1	1	2	0

Binary	Break	Switch	Track	Arrive
100	1	0	2	0
001	0	0	0	1
001	0	1	1	1
001	1	1	2	0

Binary	Break	Switch	Track	Arrive
001	1	0	2	0
100	0	0	0	1
100	0	1	1	1
100	1	1	2	0