Tracing Actual Causes with Counterfactual Witness Maps

Tim Woydt
Department of Computer Science
Technical University of Darmstadt
tim(dot)woydt(at)tu-darmstadt(dot)de

Matej Zečević
Department of Computer Science
Technical University of Darmstadt

Kristian Kersting
Department of Computer Science
Technical University of Darmstadt
Hessian Center for AI (hessian.AI)
German Research Center for AI (DFKI)

Moritz Willig
Department of Computer Science
Technical University of Darmstadt

Abstract

Halpernian actual causation, the formal study of which events in a specific situation caused a specific outcome in acyclic structural causal models, underpins legal responsibility, moral blame, and the assessment of causal harm. Despite the rich logical framework, no prior sound and complete enumeration procedure is known for finding *all* actual causes in a given setting, blocking downstream tasks such as responsibility attribution, blame assessment, and causal-harm grading that aggregate over the full cause set. We close this gap by introducing *witness targets*: sets of joint configurations containing every counterfactual witness. In combination with an *ancestor intervention grammar* we derive a finite directed acyclic *witness map*. We prove that for every actual cause there exists a contingency set such that the joint intervention set appears as a node of this map, and present an algorithm for *Tracing Actual Causes (TrAC)* that is sound and complete for enumerating all actual causes of any target event. We compare TrAC empirically with existing single-cause baselines on random Boolean structural causal models, demonstrating practical feasibility for graphs of up to 15 nodes.

1 Introduction

What *actually* caused a specific outcome, such as a patient's death after medication, an accident, or a software failure? Actual causation [16] answers this question for singular, context-dependent cases, grounding legal responsibility [25], moral blame [11], and algorithmic accountability [9]. Downstream tasks that aggregate over the full cause set, notably responsibility and blame scores [9] and the grading of causal harm [6], are blocked without an efficient enumeration procedure. The dominant framework [17, 16] formalises actual causation in acyclic structural causal models (SCMs) [27] via three conditions (AC1)–(AC3) on a candidate variable assignment, requiring its actual co-occurrence with the effect, a *contingency set* of variables held at actual values exposing counterfactual dependence, and subset-minimality (formal definitions in Sec. 3). We adopt the *modified* HP definition [15, 16], one of several proposals in the literature, for its conceptual clarity and favourable complexity profile: verifying a single candidate is D_1^P-complete [1, 15], vs. Σ_2^P-complete in general under the original [12]. The modified definition is also the natural fit for our construction: holding contingencies at their actual values is exactly the membership constraint of Corollary 2, so condition AC2 collapses to

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

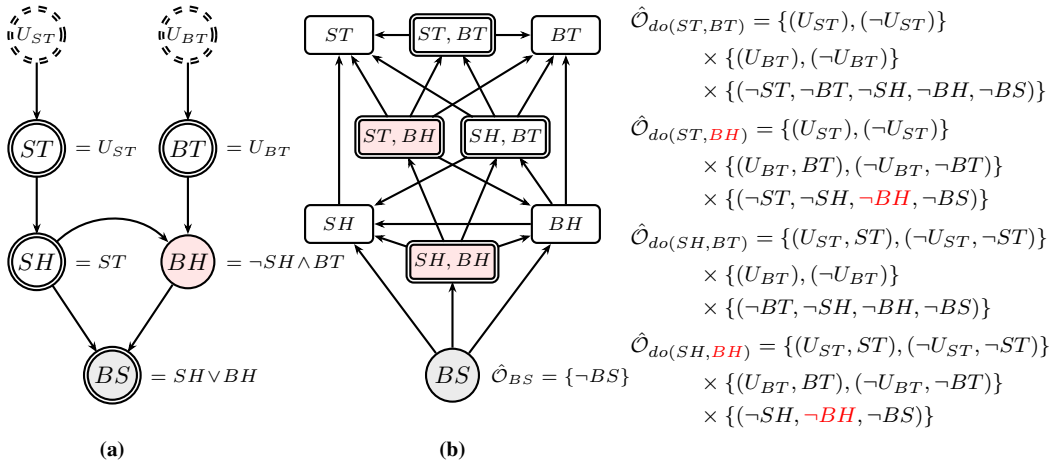


Figure 1: (a) The DAG of the Billy and Suzy example. The actual world is $(ST, BT, \neg BH, SH, BS)$ with environment (U_{ST}, U_{BT}) (True variables are double-bordered). (b) The witness map (Definition 6) for the counterfactual target $\neg BS$ with double-bordered intervention sets for global targets (Definition 3) allowing the observed environment. *As no singleton passes and none of these sets contains another, all four are but-for causes.* The nodes that allow for non-empty contingency ($\neg BH$) are highlighted in red. *Thus we have two actual causes $ST(=1)$ and $SH(=1)$.* See Example 3 for details. (Best viewed in color.)

a single query against a precomputed target set, whereas the original definition's additional universal clause over contingency subsets admits no such collapse in general.

Existing exact algorithms suffer from two limitations: Ibrahim and Pretschner [19] translate HP conditions into ILP clauses for a *single smallest* cause, restricted to Boolean SCMs and unable to enumerate the full cause set; Reyd et al. [29] extend coverage to non-Boolean and black-box SCMs via forward simulation with sampled alternative values, but offer no soundness or completeness guarantee. We present the first constructive, sound-and-complete algorithm for enumerating *all* actual causes of an event in an acyclic SCM. Rather than iterating over alternative values forward, we propagate the *complement of the target event* backward through the structural equations, yielding the *witness targets*: for each candidate intervention set, the exact set of joint variable configurations compatible with that complement. This converts the existential search in AC2 into a single set-membership query, eliminating the dependence on alternative-value cardinality from the per-candidate verification cost. A finite *witness map* over structurally admissible ancestor subsets then enumerates all candidate intervention sets; its worst-case cost remains exponential (Section 4.3), which pruning substantially reduces in practice. Figure 1 introduces the running rock-throwing preemption example (Suzy- and Billy throw rocks at a bottle with perfect accuracy (ST=1,BT=1): Suzy Hits (SH=1), but Billy does not hit (BH = 0) due to Suzy hitting first. The bottle shatters (BS=1) due to either of them hitting.) used throughout, with the formal machinery developed in Section 4.

Contributions. We introduce three components that together yield a sound and complete enumeration procedure for actual causes. First, *witness targets* (Section 4.1) provide an exact backward-propagation criterion for counterfactual satisfaction, converting AC2 into a set-membership query (Definitions 1–3, Theorem 1, Corollary 2). Second, the *witness map* (Section 4.2) builds on the ancestor intervention grammar and ancestral map to enumerate all structurally admissible candidate intervention sets (Definitions 4–6, Propositions 3–4). Third, the *Tracing Actual Causes (TrAC)* algorithm (Section 4.3, Algorithm 1) combines both with soundness and completeness guarantees (Theorem 7) and an explicit complexity analysis (Appendix E).

2 Related Work

HP causation: extensions and algorithms. Foundational work formulates responsibility and blame [9] (whose computation requires precisely the enumeration we provide), recasts HP via causal

sufficiency [2], develops actual causation in CP-logic with graded judgements [4], and extends to probabilistic [13] and nondeterministic [3] settings; downstream aggregations such as causal harm [5] all presuppose an enumeration step. Context-specific independence (CSI) [7, 33] provides an algebraic lens on overdetermination and preemption that the outer approximation of Appendix D realises in marginal form [28]; we discuss this connection further in Appendix H.

Explainability and attribution. Instance-level XAI sits close to but is distinct from actual causation. Shapley-value attributions, both associational [24, 21] and DAG-aware [18], quantify marginal contributions rather than identify singular causes. Counterfactual explanations [30, 26] and causally-augmented recourse [22] implement but-for reasoning (AC2 without contingency), missing overdetermination and preemption. Population relative unit-level SCM explanations are discussed in [32]. Galhotra et al. [14] apply probabilistic contrastive counterfactuals in databases; Chockler and Halpern [10] bridge HP causation to image-level XAI under an assumed SCM, and causal root-cause analysis [23, 8, 20, 31] addresses a related root-cause diagnosis problem under different assumptions and outside the HP framework; these methods neither target nor guarantee HP actual causes. We take the SCM as given and address the prerequisite enumeration step; SCM extraction is orthogonal.

Adopted and new components. We adopt the modified HP definition (AC1)–(AC3) [15, 16] with the responsibility measure used in the examples [9, 16], the complexity results for verifying a single candidate cause [1, 12], which motivate our choice of definition, and the baselines with their solution strategies [19, 29], used unchanged for comparison; context-specific independence [7, 33, 28] serves only to interpret the pruning (Appendix H). New are the witness targets (Definitions 1–3) with their exactness (Theorem 1) and the reduction of the AC2 test to a single set-membership query (Corollary 2); the ancestor intervention grammar, ancestral map and witness map (Definitions 4–6) with the closed-form child characterisation (Proposition 3) and the reachability criterion (Proposition 4); TrAC with the first soundness-and-completeness guarantee for enumerating all actual causes (Theorem 7), where prior exact methods return a single smallest cause [19] or lack guarantees [29]; and the three-parameterisation complexity analysis (Appendix E), the constraint-based outer approximation (Appendix D) and the formally justified pruning strategies (Appendix F).

3 Preliminaries

Structural causal models (SCM) are structures $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, \mathbb{P}_{\mathbf{U}} \rangle$ with a set of exogenous variables $\mathbf{U} = \{X_1, \ldots, X_o\}$ and a set of endogenous variables $\mathbf{V} = \{X_{o+1}, \ldots, X_{o+p}\}$ with *full configurations* $\mathbf{O} = \mathbf{U} \cup \mathbf{V}$ and values $\vec{o} \in \vec{\mathcal{O}} = \vec{\mathcal{U}} \times \vec{\mathcal{V}} = \bigtimes_{i=1}^{o+p} \mathcal{X}_i$, a set of structural equations $\mathcal{F} = \{X = F_X : X \in \mathbf{V}\}$ with functions $F_X : \vec{\mathcal{X}}_{pa(X)} \to \mathcal{X}_X$ and a distribution $\mathbb{P}_{\mathbf{U}}$ of the exogenous variables. Assignments[1] $\vec{u} : \mathbf{U} \to \bigcup_{X_i \in \mathbf{U}} \mathcal{X}_i$ of (all) exogenous variables are called **environments** of $\mathcal{M}$ and assignments $\mathtt{w} : \mathbf{V} \to \bigcup_{X_i \in \mathbf{V}} \mathcal{X}_i$ of (all) endogenous variables are called **worlds** of $\mathcal{M}$. A pair $(\mathcal{M}, \vec{u})$ is called a **setting** and the **actual world** $\mathtt{w}_{\vec{u}}$ of a specific setting $(\mathcal{M}, \vec{u})$ is induced by the evaluation of structural equations in $\mathcal{F}$. For an assignment $\vec{X} = \vec{x}$ with $\vec{X} \subseteq \mathbf{V}$, the **intervened SCM** $\mathcal{M}_{do(\vec{X}=\vec{x})}$ is obtained from $\mathcal{M}$ by replacing every structural equation F_{X_i} with the constant x_i (for $X_i \in \vec{X}$); its actual world under $\vec{u}$ is denoted $\mathtt{w}^*_{\vec{u}}$ and called *counterfactual* when $\mathtt{w}^*_{\vec{u}} \neq \mathtt{w}_{\vec{u}}$. The corresponding **modal operator** satisfies $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}]\varphi \;\Leftrightarrow\; \mathcal{M}_{do(\vec{X}=\vec{x})}, \vec{u} \models \varphi$.

The actual causes easiest to define are so-called *but-for causes*, which are *necessary conditions* of an event. In a given setting $(\mathcal{M}, \vec{u})$ we say that the event $(\vec{X} = \vec{x})$ is a **but-for cause** of the event φ if and only if $\mathcal{M}, \vec{u} \models (\vec{X} = \vec{x}) \wedge \varphi$ (AC1), there exists $\vec{x}' \in \vec{\mathcal{X}}$ with $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}']\neg\varphi$ (AC2*), and $\vec{X} = \vec{x}$ is a smallest conjunction of atomic events satisfying the previous conditions (i.e., there is no subevent $(\vec{X}' = \vec{x}') \subset (\vec{X} = \vec{x})$ satisfying both AC1 and AC2*) (AC3). Note here that a variable alone is never a cause; only the variable taking a specific value can be a cause. The atomic components $(X_i = x_i)$ for $X_i \in \vec{X}$ are referred to as *parts* of the cause $(\vec{X} = \vec{x})$. To also capture causes whose but-for dependence is masked by a backup process, the modified HP definition [15] relaxes the second condition as follows: if there exist $\vec{x}' \in \vec{\mathcal{X}}$ and $\vec{Z} \subseteq \mathbf{V} \setminus \vec{X}$ such that $\mathcal{M}, \vec{u} \models [\vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z}), \vec{X} = \vec{x}']\neg\varphi$ (AC2), and $\vec{X} = \vec{x}$ is a smallest conjunction of atomic events

[1] There is a canonical bijection between assignments $\vec{u} : \mathbf{U} \to \bigcup_{X_i \in \mathbf{U}} \mathcal{X}_i$ with $\vec{u}(X_i) \in \mathcal{X}_i$ and configurations $\vec{u} \in \vec{\mathcal{U}}$; we use both notations interchangeably and call either an *environment*.

satisfying AC1 and AC2, then $\vec{X} = \vec{x}$ is called an **actual cause** of the event φ. Preemption is the canonical example (Figure 1, Appendix C.2): Suzy's throw is not a but-for cause of the shattered bottle, but it is an actual cause, since holding Billy's miss at its actual value exposes the dependence.

As the but-for causes and actual causes are defined via the existence of counterfactual worlds where the target event φ does not hold, we call such a world $\mathtt{w}^*_{\vec{u}}$ a **counterfactual witness** for the actual cause or but-for cause $\vec{X} = \vec{x}$ of the event φ in the setting $(\mathcal{M}, \vec{u})$. For a given setting $(\mathcal{M}, \vec{u})$, we can identify such a witness for an actual cause $\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X})$ of an event φ by a tuple $(\vec{Z}, \vec{X}, \vec{x}', \varphi)$ with $\vec{Z} \subseteq \mathbf{V} \setminus \vec{X}$. We call $\vec{Z}$ the **contingency set**: it consists of variables held fixed at their actual values $\mathtt{w}_{\vec{u}}(\vec{Z})$ in order to reveal the counterfactual dependence of φ on $\vec{X}$. The causal claim $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ thus holds as a cause *contingent on* the context $\vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})$. If $\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X})$ is (also) a but-for cause of φ, then $\vec{Z} = \emptyset$, i.e., no contingency is required.

4 Tracing Actual Causes

We assume throughout that the SCM induces an acyclic causal graph. W.l.o.g., the target $Y \in \mathbf{V}$ is a single variable with value domain $\mathcal{Y}$; if not, introduce a fresh variable whose structural equation copies the joint configuration of the target set. We write $(Y \in \mathcal{Y}^*)$ with $\mathcal{Y}^* \subseteq \mathcal{Y}$ for the target event and $(Y \in \hat{\mathcal{Y}})$ for the induced counterfactual target with $\hat{\mathcal{Y}} = \mathcal{Y} \setminus \mathcal{Y}^*$. We characterise the exact set of joint variable configurations compatible with the counterfactual target via backward propagation through the structural equations. The construction proceeds through two building blocks: *parent targets* and *descendant targets*.

Our construction separates verification from enumeration. Section 4.1 develops witness targets, computed by a single backward pass through the structural equations; they reduce the AC2 test for a fixed candidate pair $(\vec{X}, \vec{Z})$ to one membership query in a precomputed set (Corollary 2), with no search over alternative values. Section 4.2 develops the ancestor intervention grammar and the witness map, a finite DAG whose nodes are the candidate intervention sets that have to be tested. Section 4.3 combines both into Algorithm 1 (TrAC) and proves it sound and complete (Theorem 7).

4.1 Witness Targets

Definition 1 (parent targets)**.** *For an endogenous variable $X \in \mathbf{V}$ and a set $\hat{\mathcal{O}} \subseteq \vec{\mathcal{O}}$, the **parent targets** are $\hat{\mathcal{O}}^{pa}_{\mathcal{M}}(X, \hat{\mathcal{O}}) := \{\vec{o} \in \hat{\mathcal{O}} \mid \vec{o}_{pa(X)} \in F_X^{-1}(\vec{o}_X)\}$, where $\vec{o}_{pa(X)}$ is the projection of $\vec{o}$ onto the parent components of X and $F_X^{-1}(\vec{o}_X) \subseteq \vec{\mathcal{X}}_{pa(X)}$ is the preimage of $\vec{o}_X$ under F_X.*

Note that the parent target operator is exact because a configuration $\vec{o}$ is retained if and only if its parent values are structurally consistent with producing $\vec{o}_X$, hence no feasible assignment is lost and no spurious one is introduced.

The parent target operator handles a single step of backward propagation. To accumulate constraints all the way from the target Y up to any ancestor X, we apply it recursively across the graph.

Definition 2 (descendant targets)**.** *For a given SCM $\mathcal{M}$, target variable $Y \in \mathbf{V}$, target values $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, variable $X \in \mathbf{V}$, and intervention set $\vec{S} \subseteq \mathbf{V}$, the **descendant targets** $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}})$ (written $\hat{\mathcal{O}}^{\downarrow}(X)$ when the context is fixed) are defined recursively by:*

$$\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}}) := \begin{cases} \{\vec{o} \in \vec{\mathcal{O}} \mid \vec{o}_Y \in \hat{\mathcal{Y}}\} & \text{if } X = Y, \\ \vec{\mathcal{O}} & \text{if } ch(X) = \emptyset, \\ \bigcap_{Z \in ch(X) \setminus \vec{S}} \hat{\mathcal{O}}^{pa}_{\mathcal{M}}\big(Z, \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(Z, \hat{\mathcal{Y}})\big) & \text{otherwise.} \end{cases} \tag{1}$$

This is the exact set of joint configurations of $\mathcal{M}_{do(\vec{S})}$ compatible with $(Y \in \hat{\mathcal{Y}})$, as constrained by the causal mechanisms between X and Y.

The recursion traverses the causal graph from Y upward. At the target Y, the set consists of all full configurations where Y takes a value in $\hat{\mathcal{Y}}$. For leaves with no children, the set is unconstrained. Otherwise, the descendant targets of all non-intervened children are intersected via parent targets, propagating the joint feasibility constraint upward.

Remark 1 (connection to causal sufficiency)**.** *The preimage $F_{X_i}^{-1}(\vec{o}_i)$ used by the parent target operator (Definition 1) is exactly the set of joint parent configurations* directly sufficient *for $\vec{o}_i$ in the sense of Beckers [2]; iterating across descendants (Definition 2) realises a graph-aware version of* strong sufficiency*, the transitive closure of direct sufficiency along the network.*

Remark 2 (target reuse under intervention)**.** *For any variable $X \in \mathbf{V}$ that is either (i) a strict descendant of some $S' \in \vec{S}$ (with $X \notin \vec{S}$), meaning S' lies* above *X in the DAG and hence not on any directed path from X toward Y, or (ii) an element of $\vec{S}$ with $desc(X) \cap \vec{S} = \emptyset$, the intervention on $\vec{S}$ does not alter the structural equations on any directed path from X to Y, so the descendant targets are invariant: $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}}) = \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\emptyset)}(X, \hat{\mathcal{Y}})$. This enables efficient recomputation when moving along the witness map introduced in the next subsection; only the ancestors of and within the intervention set need to be recomputed; all descendant targets below the intervention are reused from the observational pass.*

The descendant targets $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}})$ capture what is required at each individual variable X. To obtain the complete set of joint configurations compatible with the counterfactual target across the *entire* model, we aggregate over all roots of the intervened model.

Definition 3 (global targets)**.** *For a given SCM $\mathcal{M}$, target variable $Y \in \mathbf{V}$, target values $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, and intervention set $\vec{S} \subseteq \mathbf{V}$, the* ***global targets*** *$\hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}) := \bigcap_{R \in \mathbf{U} \cup \vec{S}} \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(R, \hat{\mathcal{Y}})$ are the set of all variable configurations $\vec{o} \in \vec{\mathcal{O}}$ compatible with the target event $(Y \in \hat{\mathcal{Y}})$ when constraints are propagated backward from Y through the structural equations of $\mathcal{M}_{do(\vec{S})}$[2], obtained by intersecting the descendant targets of all roots of $\mathcal{M}_{do(\vec{S})}$, that is, of all exogenous and all intervened variables. Concrete instances at the four pair-intervention sets of the Billy-and-Suzy example appear in the side panel of Figure 1.*

The global targets are constructed by a purely algebraic backward pass. The following theorem validates this construction for the variable configurations that are actually realised: for the unique joint configuration $\vec{o}_{\vec{u},\vec{s}}$ induced by $\mathcal{M}_{do(\vec{S}=\vec{s})}$ under environment $\vec{u}$, global-target membership is equivalent to counterfactual satisfaction of $(Y \in \hat{\mathcal{Y}})$.

Theorem 1 (witness target membership)**.** *Let $\mathcal{M}$ be an acyclic SCM, $Y \in \mathbf{V}$, $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, and $\vec{S} \subseteq \mathbf{V}$ with fixed values $\vec{s} \in \vec{\mathcal{S}}$. For the unique configuration $\vec{o}_{\vec{u},\vec{s}} := \big(\vec{u}(X_1), \ldots, \vec{u}(X_o), \mathtt{w}^*_{\vec{u}}(X_{o+1}), \ldots, \mathtt{w}^*_{\vec{u}}(X_{o+p})\big)$ of setting $(\mathcal{M}_{do(\vec{S}=\vec{s})}, \vec{u})$ we have:*

$$\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}) \Leftrightarrow \mathcal{M}_{do(\vec{S}=\vec{s})}, \vec{u} \models (Y \in \hat{\mathcal{Y}}). \tag{2}$$

Proof sketch. By induction on the reverse topological order of the DAG. The $(\Rightarrow)$ direction follows from the base case of Definition 2, propagated along a directed path from a root of $\mathcal{M}_{do(\vec{S})}$ to Y. For $(\Leftarrow)$, acyclicity guarantees that $\vec{o}_{\vec{u},\vec{s}}$ is the unique configuration of $(\mathcal{M}_{do(\vec{S}=\vec{s})}, \vec{u})$; membership in every descendant target set then follows by upward induction, using structural consistency of F_Z at each step. The full inductive proof is given in Appendix B. □

Theorem 1 gives a membership criterion for *fixed* values $\vec{s}$. For the AC2 condition, we need to check existence over *all* alternative values $\vec{x}' \in \vec{\mathcal{X}}$. The following corollary shows that this existential quantifier can be collapsed into a single projection of the global targets, converting AC2 from a search into a membership test.

Corollary 2 (witness existence)**.** *For an acyclic SCM $\mathcal{M}$, $Y \in \mathbf{V}$, $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, $\vec{X}, \vec{Z} \subseteq \mathbf{V}$ with $\vec{X} \cap \vec{Z} = \emptyset$, there exists $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X},\vec{Z})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ and $\vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$, the actual values of $\vec{Z}$ (the restriction of the actual world $\mathtt{w}_{\vec{u}}$ to $\vec{Z}$), if and only if there exists some $\vec{x}' \in \vec{\mathcal{X}}$ such that $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$. The membership condition thus collapses the existential quantifier over alternative values $\vec{x}' \in \vec{\mathcal{X}}$ in (AC2) into a set-membership query against the global targets.*

[2]The global target is a superset of the actual solutions of $\mathcal{M}_{do(\vec{S})}$: it may contain configurations that disagree with $\mathcal{M}_{do(\vec{S})}$ on variables outside the ancestor cone of Y, since the backward propagation only constrains the path leading to Y. These spurious configurations are projected out by the environment-conditional membership query of Corollary 2.

Proof. $\vec{S} := \vec{Z} \cup \vec{X}$. ($\Leftarrow$) Suppose there is $\vec{x}' \in \vec{\mathcal{X}}$ s.t. $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$. By Theorem 1 (with intervention set $\vec{S}$ fixed at $(\vec{x}', \mathtt{w}_{\vec{u}}(\vec{Z}))$), the configuration $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$. Since $(\vec{o}_{\vec{u},\vec{s}})_{\mathbf{U}} = \vec{u}$ and $(\vec{o}_{\vec{u},\vec{s}})_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$, the configuration $\vec{o}_{\vec{u},\vec{s}}$ witnesses the existence statement. ($\Rightarrow$) Suppose there exists $\vec{o}' \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ with $\vec{o}'_{\mathbf{U}} = \vec{u}$ and $\vec{o}'_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$. Let $\vec{x}' := \vec{o}'_{\vec{X}}$. Since $\mathcal{M}$ is acyclic, the joint configuration of $\mathcal{M}_{do(\vec{X}=\vec{x}',\vec{Z}=\mathtt{w}_{\vec{u}}(\vec{Z}))}$ under environment $\vec{u}$ is *unique* (each endogenous variable is determined by its parents and the structural equations); hence $\vec{o}' = \vec{o}_{\vec{u},\vec{x}',\mathtt{w}_{\vec{u}}(\vec{Z})}$. Theorem 1 then yields $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$. □

Corollary 2 solves the *verification* half of the enumeration problem: given a candidate pair $(\vec{X}, \vec{Z})$, a single membership query suffices to decide AC2. What remains is the *enumeration* half: which intervention sets $(\vec{X}, \vec{Z})$ need to be tested at all? Naively, one would search over all subsets of $\mathbf{V}$, a search space that is exponential in $|\mathbf{V}|$. The next subsection replaces this brute-force search with a grammar-based structure that generates only the *structurally admissible* candidate sets via backward unfolding of the causal graph.

4.2 Witness Maps

In the grammar, every variable of the current set either stays as it is (keep) or is replaced by its direct endogenous causes (unfold), moving the search one step further back through the graph. Starting from $\{Y\}$ and repeating these two local steps generates all candidate intervention sets; Figure 2 shows a complete example.

Definition 4 (ancestor intervention grammar)**.** *Let $\mathcal{M}$ have a DAG and $Y \in \mathbf{V}$. Write $\Sigma_{\mathcal{M}}(Y) := anc^{\mathbf{V}}_{\mathcal{M}}(Y) \cup \{Y\}$ for the endogenous ancestors of Y including Y itself. The **ancestor intervention grammar** $G_{\mathcal{M}}(Y)$ is the parallel set-rewriting system with **alphabet** $\Sigma_{\mathcal{M}}(Y)$, **start set** $\{Y\}$ and **productions** for every $X \in \Sigma_{\mathcal{M}}(Y)$:*

$$(P_{\text{keep}}) \quad X \longrightarrow \{X\}, \tag{3}$$

$$(P_{\text{unfold}}) \quad X \longrightarrow pa^{\mathbf{V}}_{\mathcal{M}}(X). \tag{4}$$

Both productions are applicable to every symbol (Remark 3 gives their causal reading); for X with $pa^{\mathbf{V}}_{\mathcal{M}}(X) = \emptyset$, applying (P_{unfold}) has the effect of removing *X from the current set.*

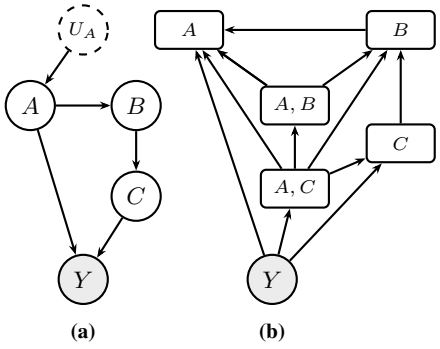


Figure 2: (a) Example DAG with $pa(Y) = \{A, C\}$, $pa(C) = \{B\}$, $pa(B) = \{A\}$. (b) Ancestral map $\mathcal{A}_{\mathcal{M}}(Y)$ (Definition 5): the root's children are the non-empty subsets of $pa^{\mathbf{V}}_{\mathcal{M}}(Y)$; the children of every other node follow from Proposition 3. Node $\{A, C\}$ has four children, $\{A\}$, $\{B\}$, $\{C\}$ and $\{A, B\}$ (Example 1).

*A **derivation step** $\vec{S} \Rightarrow_G \vec{S}'$ between non-empty sets $\vec{S}, \vec{S}' \subseteq \Sigma_{\mathcal{M}}(Y)$ applies a production to each element independently. Choose $\vec{S} = \vec{S}_{\text{unfold}} \dot{\cup} \vec{S}_{\text{keep}}$ and set:*

$$\vec{S}' \subseteq \bigcup_{X \in \vec{S}_{\text{unfold}}} pa^{\mathbf{V}}_{\mathcal{M}}(X) \cup \vec{S}_{\text{keep}}, \qquad \vec{S}' \neq \emptyset, \tag{5}$$

*where $\vec{S}_{\text{unfold}} \neq \emptyset$ (at least one element is unfolded or dropped). Note that for $X \in \vec{S}_{\text{unfold}}$ with $pa^{\mathbf{V}}_{\mathcal{M}}(X) = \emptyset$, applying (P_{unfold}) contributes nothing to $\vec{S}'$, so unfolding a source node is equivalent to dropping it. We write $\vec{S} \Rightarrow^+_G \vec{S}'$ if $\vec{S}'$ is obtained from $\vec{S}$ by one or more derivation steps (the transitive closure of $\Rightarrow_G$). The **generated language** is $\mathcal{L}\big(G_{\mathcal{M}}(Y)\big) := \big\{\vec{S} \subseteq \Sigma_{\mathcal{M}}(Y) \;\big|\; \vec{S} \neq \emptyset,\ \{Y\} \Rightarrow^+_G \vec{S}\big\}$.*

Remark 3 (productions and their roles)**.** *The two productions have distinct causal meanings.* Keep *($X \in \vec{S}_{\text{keep}}$): the variable is already at the right ancestral depth and will remain in the intervention set.* Unfold *($X \in \vec{S}_{\text{unfold}}$): the variable's causal influence is traced one step further back through its endogenous parents $pa^{\mathbf{V}}_{\mathcal{M}}(X)$. This allows the grammar to generate combinations of ancestors at various depths, which is crucial for identifying actual causes that may lie at different levels of the causal graph. In particular, the grammar allows to move from an intervention set to any of its non-empty proper subsets in one step, since unfolded elements need not be selected for $\vec{S}'$.*

Definition 5 (ancestral map)**.** *For an SCM $\mathcal{M}$ and target $Y \in \mathbf{V}$, the **ancestral map** $\mathcal{A}_{\mathcal{M}}(Y)$ is the directed graph whose nodes are the elements of $\mathcal{L}(G_{\mathcal{M}}(Y))$ together with the root $\{Y\}$, and whose edges are the derivation steps: $\vec{S} \to \vec{S}'$ in $\mathcal{A}_{\mathcal{M}}(Y)$ iff $\vec{S} \Rightarrow_G \vec{S}'$ in a single step.*

The ancestral map is defined implicitly via derivation steps of the grammar. For computational purposes, it is useful to have a closed-form expression for the children of any node $\vec{S}$ directly in terms of the graph structure of $\mathcal{G}_{\mathcal{M}}$, without simulating individual derivation steps.

Proposition 3 (grammar–graph equivalence)**.** *For a set $\vec{S}'$, write $pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S}') := \bigcup_{X \in \vec{S}'} pa^{\mathbf{V}}_{\mathcal{M}}(X)$, let $ch_{\mathcal{A}}(\vec{S})$ denote the set of children of a node $\vec{S}$ in $\mathcal{A}_{\mathcal{M}}(Y)$, and let $\mathcal{P}(\cdot)$ be the powerset operator. Then $ch_{\mathcal{A}}(\vec{S}) = \bigcup_{\emptyset \neq \vec{S}' \subseteq \vec{S}} \mathcal{P}\Big(pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S}') \cup (\vec{S} \setminus \vec{S}')\Big) \setminus \{\emptyset\}$. The proof is given in Appendix B.*

Example 1 (derivation steps)**.** In Figure 2, consider the node $\vec{S} = \{A, C\}$. Unfolding C and keeping A yields the candidate set $pa^{\mathbf{V}}_{\mathcal{M}}(C) \cup \{A\} = \{A, B\}$, whose non-empty subsets $\{A\}$, $\{B\}$ and $\{A, B\}$ are children. Unfolding A drops it, since $pa^{\mathbf{V}}_{\mathcal{M}}(A) = \emptyset$, and keeping C yields $\{C\}$; unfolding both yields $\{B\}$. Hence $ch_{\mathcal{A}}(\{A, C\}) = \{\{A\}, \{B\}, \{C\}, \{A, B\}\}$, as stated by Proposition 3.

Remark 4 (termination)**.** *Since the DAG is acyclic and finite, every derivation from $\{Y\}$ terminates: each application of (P_{unfold}) to X replaces it by elements at strictly smaller topological depth, so $\mathcal{A}_{\mathcal{M}}(Y)$ is a finite directed acyclic graph. See Figure 2 for an example.*

Finiteness bounds the search space; the next question is which ancestor sets actually appear as nodes. The following proposition gives a sufficient condition for reachability, a criterion we will invoke in the completeness proof of Theorem 6 to show that every actual cause's intervention set is represented in the map.

Proposition 4 (ancestor reachability)**.** *Let $\vec{S} \subseteq anc^{\mathbf{V}}_{\mathcal{M}}(Y)$, $\vec{S} \neq \emptyset$. If every $X \in \vec{S}$ has a directed path to Y in $\mathcal{G}_{\mathcal{M}}$ passing through no other element of $\vec{S}$, then $\vec{S} \in \mathcal{A}_{\mathcal{M}}(Y)$.*

Proof. Starting at the root node $\{Y\}$ in $\mathcal{A}_{\mathcal{M}}(Y)$, we iteratively unfold variables into their parents until some node contains an element $X \in \vec{S}$. We keep X and continue unfolding the remaining variables until a node also contains the next element of $\vec{S}$, and so on. If some $X \in \vec{S}$ were not reachable by this procedure, every directed path from X to Y would pass through another element of $\vec{S}$ already included, contradicting the assumption. □

Remark 5 (sufficiency only)**.** *The path-disjointness condition is sufficient but not necessary for $\vec{S} \in \mathcal{A}_{\mathcal{M}}(Y)$; Appendix B.1 exhibits a reachable ancestor set on a chain-into-diamond DAG whose path-disjointness fails. The proposition is sufficient for the completeness argument of Theorem 6.*

We now have both ingredients needed to identify actual causes: the global targets from Section 4.1, which capture counterfactual satisfaction algebraically, and the ancestral map, which enumerates all structurally admissible candidate intervention sets. Combining them yields the *witness map*: the ancestral map annotated at each node with the corresponding global targets.

Definition 6 (witness map)**.** *For an ancestral map $\mathcal{A}_{\mathcal{M}}(Y)$ and target values $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$, the **witness map** $\mathcal{W}_{\mathcal{M}}(Y \in \hat{\mathcal{Y}})$ is the directed graph obtained from $\mathcal{A}_{\mathcal{M}}(Y)$ by labelling each node $\vec{S}$ (a set of ancestors of Y) with $\hat{\mathcal{O}}_{\mathcal{M}, do(\vec{S})}(\hat{\mathcal{Y}})$, where the root node $\{Y\}$ is labelled $\hat{\mathcal{O}}_{\mathcal{M}, do(\emptyset)}(\hat{\mathcal{Y}})$. Each node thus encodes the exact set of joint variable configurations of $\mathcal{M}_{do(\vec{S})}$ compatible with the target event, and traversal of the map corresponds to systematically extending the intervention set along the causal ancestry of Y. Figure 1(b) displays the witness map for the Billy-and-Suzy preemption example.*

We now establish soundness (Lemma 5) and sufficiency of the witness map (Theorem 6), the latter directly inducing completeness of Algorithm 1.

Lemma 5 (AC-witness identification)**.** *Let $(\mathcal{M}, \vec{u})$ be a setting of an acyclic SCM with $\mathcal{M}, \vec{u} \models (Y \in \mathcal{Y}^*)$, and let $Y \in \mathbf{V}$, $\mathcal{Y}^* \subseteq \mathcal{Y}$, $\hat{\mathcal{Y}} := \mathcal{Y} \setminus \mathcal{Y}^*$. For disjoint $\vec{X}, \vec{Z} \subseteq \mathbf{V}$ the following two statements hold: **(But-for causes.)** If $\vec{X} \in \mathcal{L}(G_{\mathcal{M}}(Y))$ is a non-root node of $\mathcal{W}_{\mathcal{M}}(Y \in \hat{\mathcal{Y}})$ and there exists $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M}, do(\vec{X})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$, then there exists $\vec{x}' \in \vec{\mathcal{X}}$ with $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$ such that $(\emptyset, \vec{X}, \vec{x}', \mathcal{Y}^*)$ is a counterfactual witness for $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ satisfying (AC1) and (AC2*). In particular, $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ is a but-for cause of $(Y \in \mathcal{Y}^*)$ if additionally $\vec{X}$ is minimal. **(Actual causes.)** If $\vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$*

is a non-root node of $\mathcal{W}_{\mathcal{M}}(Y \in \hat{\mathcal{Y}})$ *and there exists* $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X},\vec{Z})}(\hat{\mathcal{Y}})$ *with* $\vec{o}_{\mathbf{U}} = \vec{u}$ *and* $\vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$*, then there exists* $\vec{x}' \in \vec{\mathcal{X}}$ *with* $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$ *such that* $(\vec{Z}, \vec{X}, \vec{x}', \mathcal{Y}^*)$ *is a counterfactual witness for* $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ *satisfying (AC1) and (AC2) with contingency set* $\vec{Z}$*. In particular,* $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ *is an actual cause of* $(Y \in \mathcal{Y}^*)$ *if additionally* $\vec{X}$ *is minimal.*

Proof sketch. (AC1) holds trivially since $\mathtt{w}_{\vec{u}}$ is the actual world. For (AC2), Corollary 2 yields an $\vec{x}' \in \vec{\mathcal{X}}$ with $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$; that $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$ follows from the consistency of structural equations (setting $\vec{x}' = \mathtt{w}_{\vec{u}}(\vec{X})$ would reproduce the actual world, contradicting $Y \in \hat{\mathcal{Y}}$). See Appendix B for the full proof. □

Lemma 5 handles the *soundness* direction: any node in the witness map that passes the membership test certifies a genuine cause. The converse, which states that no actual cause can be missed, requires showing that every actual cause is *represented* by some node in the map.

Theorem 6 (sufficiency of witness map)**.** *Let* $(\mathcal{M}, \vec{u})$ *be a setting of an acyclic SCM with* $\mathcal{M}, \vec{u} \models (Y \in \mathcal{Y}^*)$*, and let* $\hat{\mathcal{Y}} := \mathcal{Y} \setminus \mathcal{Y}^*$*. For every actual cause* $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ *of* $(Y \in \mathcal{Y}^*)$*, there exists a contingency set* $\vec{Z} \subseteq \mathbf{V} \setminus \vec{X}$ *such that* $\vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$ *and*

$$\exists\, \vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X},\vec{Z})}(\hat{\mathcal{Y}}) \quad \textit{with} \quad \vec{o}_{\mathbf{U}} = \vec{u} \;\; \textit{and} \;\; \vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z}). \tag{6}$$

In particular, but-for causes are the special case $\vec{Z} = \emptyset$*, for which the membership condition reduces to the existence of* $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X})}(\hat{\mathcal{Y}})$ *with* $\vec{o}_{\mathbf{U}} = \vec{u}$*.*

Proof. By (AC2), there exist $\vec{Z}_0 \subseteq \mathbf{V} \setminus \vec{X}$ and $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$ such that $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z}_0 = \mathtt{w}_{\vec{u}}(\vec{Z}_0)](Y \in \hat{\mathcal{Y}})$. (For but-for causes, (AC2*) gives this with $\vec{Z}_0 = \emptyset$ directly.) Let $\vec{Z} \subseteq \vec{Z}_0$ be inclusion-minimal such that $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$ still holds; such a minimum exists since the collection of valid $\vec{Z}' \subseteq \vec{Z}_0$ is non-empty and finite. For **node existence:** $\vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$**.** By Proposition 4, it suffices to show that every element of $\vec{X} \cup \vec{Z}$ has a directed path to Y in $\mathcal{G}_{\mathcal{M}}$ passing through no other element of $\vec{X} \cup \vec{Z}$. Suppose for contradiction that some $V' \in \vec{X} \cup \vec{Z}$ has all directed paths to Y blocked at some $V'' \in (\vec{X} \cup \vec{Z}) \setminus \{V'\}$. In $\mathcal{M}_{do(\vec{X}=\vec{x}',\, \vec{Z}=\mathtt{w}_{\vec{u}}(\vec{Z}))}$, the structural equation for V'' is overridden, so V' cannot influence $\mathtt{w}^*_{\vec{u}}(Y)$ through V''. Since every causal path from V' to Y passes through V'', the value of V' is causally inert for Y in this model; hence $\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}',\, (\vec{Z} \cup \vec{X} \setminus \{V'\}) = \mathtt{w}_{\vec{u}}(\vec{Z} \cup \vec{X} \setminus \{V'\})](Y \in \hat{\mathcal{Y}})$. If $V' \in \vec{X}$, this contradicts the minimality (AC3) of $\vec{X}$ (we could drop V' from the cause). If $V' \in \vec{Z}$, this contradicts the minimality of $\vec{Z}$. Hence every $V' \in \vec{X} \cup \vec{Z}$ has a path to Y through no other element of $\vec{X} \cup \vec{Z}$, and $\vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$ follows. The **membership condition** now directly follows from Corollary 2. □

4.3 Algorithm

We summarise the identification procedure in Algorithm 1. By Theorem 6, every actual cause admits a witness $(\vec{Z}, \vec{X}, \vec{x}', \mathcal{Y}^*)$ with $\vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$, so every relevant intervention set already appears as a node of the witness map. By Corollary 2, the AC2 condition for a pair $(\vec{X}, \vec{Z})$ then reduces to a single membership query against the precomputed global targets $\hat{\mathcal{O}}_{\mathcal{M},do(\vec{X}\cup\vec{Z})}(\hat{\mathcal{Y}})$. The algorithm therefore treats each node $\vec{S} \in \mathcal{A}_{\mathcal{M}}(Y)$ directly as the total intervention set and enumerates all partitions $\vec{S} = \vec{X} \cup \vec{Z}$, where $\vec{X}$ is the candidate cause and $\vec{Z} = \vec{S} \setminus \vec{X}$ is the contingency set, testing each against these targets (Phase 2).

Theorem 7 (soundness and completeness)**.** *Let* $(\mathcal{M}, \vec{u})$ *be a setting of an acyclic SCM with* $\mathcal{M}, \vec{u} \models (Y \in \mathcal{Y}^*)$*, and let* $\hat{\mathcal{Y}} := \mathcal{Y} \setminus \mathcal{Y}^*$*. Algorithm 1 returns exactly the collection* $\mathcal{C}$ *of variable sets* $\vec{X} \subseteq \mathbf{V}$ *such that* $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ *is an actual cause of* $(Y \in \mathcal{Y}^*)$*:* $\vec{X} \in \mathcal{C}$ *iff* $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ *is an actual cause of* $(Y \in \mathcal{Y}^*)$*.*

Proof. **Soundness (⇒).** Suppose $\vec{X} \in \mathcal{C}$ upon termination of Algorithm 1. By Phase 2, there exists a node $\vec{S} \in \mathcal{A}_{\mathcal{M}}(Y)$ such that $\vec{X} \subseteq \vec{S}$ and, with $\vec{Z} := \vec{S} \setminus \vec{X}$, the membership test $\exists\, \vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$

with $\vec{o}_{\mathbf{U}} = \vec{u}, \vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$ passed. Since $\vec{S} \in \mathcal{L}(G_{\mathcal{M}}(Y))$, $\vec{S}$ is a non-root node of $\mathcal{W}_{\mathcal{M}}(Y \in \hat{\mathcal{Y}})$. Lemma 5 then gives the existence of $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$ such that $(\vec{Z}, \vec{X}, \vec{x}', \mathcal{Y}^*)$ is a counterfactual witness satisfying (AC1) and (AC2). Phase 3 retains only minimal elements, so $\vec{X}$ satisfies (AC3). Hence $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ is an actual cause.

Completeness ($\Leftarrow$). Suppose $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ is an actual cause of $(Y \in \mathcal{Y}^*)$. By Theorem 6, there exists $\vec{Z} \subseteq \Sigma_{\mathcal{M}}(Y) \setminus \vec{X}$ such that $\vec{S} := \vec{X} \cup \vec{Z} \in \mathcal{L}(G_{\mathcal{M}}(Y))$ and some $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X},\vec{Z})}(\hat{\mathcal{Y}})$ satisfies $\vec{o}_{\mathbf{U}} = \vec{u}, \vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$. Hence the membership test for the partition $(\vec{X},\ \vec{Z})$ of $\vec{S}$ succeeds in Phase 2, so $\vec{X}$ is added to $\mathcal{C}$. Since $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ satisfies (AC3) by assumption, $\vec{X}$ is not removed in Phase 3. Therefore $\vec{X} \in \mathcal{C}$ upon termination. □

Relative to the theoretical framework of Section 4.2, correctness is established by Theorem 7 below: soundness rests on Lemma 5, which shows that every positive membership test corresponds to a genuine counterfactual witness; completeness rests on Theorem 6, which guarantees that every actual cause is represented by some node in the witness map. Phase 3 compares the collected candidates pairwise under set inclusion and removes every $\vec{X}$ for which some strict subset is also in $\mathcal{C}$. This filter is exact: by Lemma 5, every removed $\vec{X}$ has a strict subset satisfying AC1 and AC2 and hence violates AC3, and by Theorem 6, every actual cause is in $\mathcal{C}$, so every non-minimal $\vec{X}$ contains one as a strict subset and is removed.

Having established that Algorithm 1 is both sound and complete, we briefly comment on its computational cost. Crucially, Corollary 2 eliminates any enumeration over alternative values $\vec{x}'$, so the cost depends only on the causal graph and intervention-set sizes, not on the domain size. The number of membership queries is $O(3^a)$ in the worst case (with $a := |anc^{\mathbf{V}}_{\mathcal{M}}(Y)|$) and reduces to a polynomial bound under bounded intervention size; the full three-parameterisation analysis is deferred to Appendix E. The remainder of this section describes pruning strategies that substantially reduce the practical cost without affecting soundness or completeness.

Remark 6 (Phase 2 pruning strategies)**.** *Three strategies reduce the cost of Phase 2 without affecting soundness or completeness:* ***(i) size-first traversal*** *processes smaller nodes first, letting confirmed causes prune supersets early;* ***(ii) node-level pre-check*** *tests existence of $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ before enumerating the $2^{|\vec{S}|} - 1$ subsets of $\vec{S}$, skipping the node on failure;* ***(iii) superset pruning*** *skips any $\vec{X} \subseteq \vec{S}$ for which a confirmed $\vec{X}' \subsetneq \vec{X}$ already exists. The completeness proof of Theorem 6 only needs nodes satisfying the condition of Proposition 4, so nodes violating it never have to be tested. Justifications, the implication that (iii) makes Phase 3 redundant when combined with (i), and continuous-domain adaptations (interval, LP, and bisection-based pre-checks) are given in Appendix F.*

Algorithm 1: Tracing Actual Causes (TrAC)

Input: SCM $\mathcal{M}$, environment $\vec{u}$, target variable Y, target event $\mathcal{Y}^*$
Output: Set of (nodes of) actual causes $\mathcal{C} \subseteq \mathcal{P}(\mathbf{V})$

$\hat{\mathcal{Y}} \leftarrow \mathcal{Y} \setminus \mathcal{Y}^*$;
`/* Phase 1: Precompute observational global targets (single bottom-up pass) */`
foreach $X \in \mathbf{V} \cup \mathbf{U}$ *in reverse topological order* **do**
  Compute $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\emptyset)}(X, \hat{\mathcal{Y}})$;
$\hat{\mathcal{O}}_{\mathcal{M},do(\emptyset)}(\hat{\mathcal{Y}}) \leftarrow \bigcap_{U \in \mathbf{U}} \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\emptyset)}(U, \hat{\mathcal{Y}})$;
$\mathcal{C} \leftarrow \emptyset$;
`/* Check that the actual world is incompatible with the counterfactual target */`
if $\exists\, \vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\emptyset)}(\hat{\mathcal{Y}})$ *with* $\vec{o}_{\mathbf{U}} = \vec{u}$ **then**
  return $\mathcal{C}$
`/* Phase 2: Traverse witness map */`
foreach *node* $\vec{S}$ *in* $\mathcal{A}_{\mathcal{M}}(Y)$ **do**
  `/* Recompute targets above` $\vec{S}$ `only; reuse below (Remark 2) */`
  Compute $\hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$;
  `/* Enumerate all cause/contingency partitions of` $\vec{S}$ `*/`
  foreach *non-empty* $\vec{X} \subseteq \vec{S}$ **do**
    $\vec{Z} \leftarrow \vec{S} \setminus \vec{X}$;
    if $\exists\, \vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ *with* $\vec{o}_{\mathbf{U}} = \vec{u}$ *and* $\vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\vec{X}\}$;
`/* Phase 3: Apply minimality (AC3) */`
Remove each $\vec{X} \in \mathcal{C}$ with a strict subset in $\mathcal{C}$;
return $\mathcal{C}$

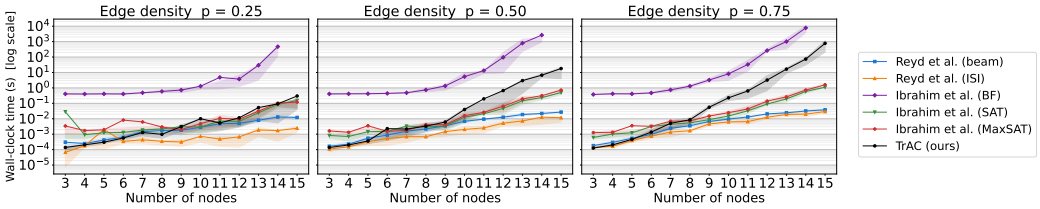


Figure 3: **Runtime comparisons to adjacent baselines.** Mean runtime (lines) with $[25, 75]$-IQR bands (shading) across 10 seeds for TrAC, Reyd et al. [29] (beam search, ISI), & Ibrahim and Pretschner [19] (HP2SAT BF, SAT, MaxSAT) on random Boolean SCMs. Each SCM is drawn from an acyclic Erdős-Rényi DAG with $n \in \{3, \ldots, 15\}$ nodes & edge density $d \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$; structural equations are random AND/OR gates with independently negated inputs. (Best viewed in color.)

Appendix C.1 works through the forest-fire scenario, contrasting overdetermination ($F_Y = L \vee A$, unique joint cause) with conjunctive causation ($F_Y = L \wedge A$, two singleton but-for causes). Appendix C.2 (Figure 1) covers the Billy-and-Suzy preemption example, illustrating the role of non-empty contingency sets and the derivation of responsibility scores.

4.4 Baseline Comparisons

We benchmark TrAC on random Boolean SCMs from acyclic Erdős-Rényi DAGs ($n \in \{3, \ldots, 15\}$, density $d \in \{0.25, 0.50, 0.75\}$, 10 seeds each), comparing against Reyd et al. [29] (beam search, ISI) and Ibrahim and Pretschner [19] (HP2SAT brute-force (BF), SAT, MaxSAT). Runtimes are shown in Fig. 3. The comparison is not perfectly direct: TrAC is the only method that provably enumerates *all* actual causes, while the baselines target a single cause without completeness guarantees. On these benchmarks, the combined pruning of Remark 6(ii+iii) removes on average over 97% of witness-map nodes at $n \geq 10$ for $d \geq 0.5$ and 73–94% for $d = 0.25$ (Fig. 6); the rate is instance-dependent and leaves the $O(3^a)$ worst case of Section 4.3 unchanged. As a result, TrAC is typically more than an order of magnitude faster than the brute-force search, while single-cause methods (SAT, MaxSAT, beam, ISI) are faster per query, as expected given they solve a strictly simpler task without completeness guarantees. Further plots and discussion are in Appendix G. All code is available at `https://github.com/MoritzWillig/tracingActualCauses`.

5 Conclusion

We presented a constructive procedure for enumerating all actual causes of an event in an acyclic SCM. The key insight is a change of direction: rather than searching forward over alternative values, we propagate the counterfactual target backward through the structural equations. This yields *witness targets* (Definitions 1–3, Theorem 1), a criterion for counterfactual satisfaction that converts the existential search in AC2 into a single set-membership query, eliminating any dependence on domain size from the per-candidate verification cost. An *ancestor intervention grammar* then generates all structurally admissible candidate intervention sets as nodes of a finite DAG and the *witness map* (Definitions 4–6, Propositions 3–4), whose traversal is guaranteed to cover every actual cause (Theorem 6). Algorithm 1 combines both ingredients and is proven sound and complete (Theorem 7) for the modified HP definition.

Limitations. TrAC works with fully specified acyclic SCM; latent confounders can be encoded as exogenous roots, but unknown or partially specified models are out of scope. The main computational limitation is storage: exact global targets require $O(D^{|\mathbf{O}|})$ space in the worst case, with D the largest domain size, plus preimage tables of size $O(|\mathbf{V}| \cdot D^{\Delta+1})$, with Δ the largest number of endogenous parents (Appendix E.1). All 390 benchmark instances up to n=15 completed on a single 32 GB machine (Appendix G), whereas our attempts at n=16 ran out of memory; since each additional Boolean variable doubles the number of configurations, added memory buys roughly one further node. The outer approximation of Appendix D needs only $O(|\mathbf{V}| \cdot D^{\Delta})$ storage, but it can only screen candidates: a failed check safely rules a candidate out. Compact exact target representations (e.g. BDDs, polytopes) for broader mechanism classes, full distributions $P(\mathbf{U})$ for probabilistic actual causation, and nondeterministic SCMs are the most pressing open directions (Appendix H).

Acknowledgments and Disclosure of Funding

We thank the reviewers for their constructive and thoughtful feedback, which helped to improve this work. This work has benefited from the German Research Foundation (DFG) research grant "Tractable Neuro-Causal Models" (KE 1686/8-1), from the clusters of excellence "Reasonable Artificial Intelligence" (RAI, EXC 3057) and "The Adaptive Mind" (TAM, EXC 3066) under Germany's Excellence Strategy, and from the LOEWE Center DYNAMIC of the State of Hesse. It was also funded by the European Union (Grant Agreement no. 101120763, TANGO). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

References

[1] G. Aleksandrowicz, H. Chockler, J. Y. Halpern, and A. Ivrii. The computational complexity of structure-based causality. *Journal of Artificial Intelligence Research*, 58:431–451, 2017.

[2] S. Beckers. Causal sufficiency and actual causation. *Journal of Philosophical Logic*, 50(6): 1341–1374, 2021.

[3] S. Beckers. Actual causation and nondeterministic causal models. In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pages 514–532. PMLR, 2025.

[4] S. Beckers and J. Vennekens. A general framework for defining and extending actual causation using cp-logic. *International Journal of Approximate Reasoning*, 77:105–126, 2016.

[5] S. Beckers, H. Chockler, and J. Y. Halpern. Quantifying harm. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 363–371, 2023. doi: 10.24963/ijcai.2023/41.

[6] S. Beckers, H. Chockler, and J. Y. Halpern. A causal analysis of harm. *Minds and Machines*, 34 (3):34, 2024. doi: 10.1007/s11023-024-09689-7.

[7] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Proc. of the 13th Conf. on Uncertainty in Artificial Intelligence*, 1996.

[8] K. Budhathoki, L. Minorics, P. Blöbaum, and D. Janzing. Causal structure-based root cause analysis of outliers. In *International conference on machine learning*, pages 2357–2369. PMLR, 2022.

[9] H. Chockler and J. Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[10] H. Chockler and J. Y. Halpern. Explaining image classifiers. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 264–272, 2024.

[11] F. Cushman, J. Knobe, and W. Sinnott-Armstrong. Moral appraisals affect doing/allowing judgments. *Cognition*, 108(1):281–289, 2008.

[12] T. Eiter and T. Lukasiewicz. Complexity results for structure-based causality. *Artificial Intelligence*, 142(1):53–89, 2002.

[13] L. Fenton-Glynn. A proposed probabilistic extension of the Halpern and Pearl definition of 'Actual Cause'. *The British Journal for the Philosophy of Science*, 68(4):1061–1124, 2017.

[14] S. Galhotra, R. Pradhan, and B. Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 international conference on management of data*, pages 577–590, 2021.

[15] J. Y. Halpern. A modification of the Halpern-Pearl definition of causality. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 3022–3033, 2015.

[16] J. Y. Halpern. *Actual Causality*. MIT Press, 2016.

[17] J. Y. Halpern and J. Pearl. Causes and explanations: A structural-model approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

[18] T. Heskes, E. Sijben, I. G. Bucur, and T. Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems*, 33:4778–4789, 2020.

[19] A. Ibrahim and A. Pretschner. From checking to inference: Actual causality computations as optimization problems. In *International Symposium on Automated Technology for Verification and Analysis*, pages 343–359. Springer, 2020.

[20] A. Ikram, S. Chakraborty, S. Mitra, S. Saini, S. Bagchi, and M. Kocaoglu. Root cause analysis of failures in microservices through causal discovery. *Advances in Neural Information Processing Systems*, 35:31158–31170, 2022.

[21] D. Janzing, L. Minorics, and P. Blöbaum. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on artificial intelligence and statistics*, pages 2907–2916. PMLR, 2020.

[22] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5): 1–29, 2022.

[23] M. Kim, R. Sumbaly, and S. Shah. Root cause detection in a service-oriented architecture. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):93–104, 2013.

[24] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[25] M. S. Moore. *Causation and Responsibility: An Essay in Law, Morals, and Metaphysics*. Oxford University Press, 2009.

[26] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[27] J. Pearl. *Causality*. Cambridge University Press, 2009.

[28] J. Pensar, H. Nyman, T. Koski, and J. Corander. Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. *Data mining and knowledge discovery*, 29(2):503–533, 2015.

[29] S. Reyd, A. Diaconescu, and J.-L. Dessalles. Searching for actual causes: Approximate algorithms with adjustable precision. In *NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science*, 2025.

[30] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2): 841–887, 2018.

[31] W. Yang, K. Zhang, and S. C. Hoi. A causal approach to detecting multivariate time-series anomalies and root causes. *arXiv preprint arXiv:2206.15033*, 2022.

[32] M. Zečević, D. S. Dhami, and K. Kersting. Causal articulation theory (cat): Articulating static and temporal causal models beyond llm rationalizations. In *Fifth Conference on Causal Learning and Reasoning*, 2026.

[33] N. L. Zhang and D. Poole. On the role of context-specific independence in probabilistic inference. In *IJCAI*, volume 99, pages 1288–1293, 1999.

Appendix for "Tracing Actual Causes with Counterfactual Witness Maps"

Contents

A Notation **14**

B Extended Proofs **15**

B.1 Counterexample to the Necessity of Proposition 4 16

C Examples **16**

C.1 Forest-Fire Example 16

C.2 Billy-and-Suzy Example 17

D Constraint-Based Outer Approximation **18**

E Full Complexity Analysis **19**

E.1 Setup: Base Costs 19

E.2 Enumeration Cost under Three Parameterisations 20

F Phase 2 Pruning Strategies **20**

G Extended Experimental Results **21**

G.1 Setup 21

G.2 Results 22

H Roadmap and Open Problems **23**

A Notation

Table 1: Notation summary.

Symbol	Meaning
Variables and subsets	
$\mathbf{V}, \mathbf{U}$	sets of endogenous, exogenous variables
$\mathbf{O} = \mathbf{U} \cup \mathbf{V}$	set of all variables (full configurations)
X, Y, X_i	single variables
$\vec{X}, \vec{Z}, \vec{S}$	arbitrary subsets of $\mathbf{V}$ ($\vec{Z}$ used as contingency, $\vec{S}$ as intervention/node)
$pa(X), ch(X), anc(X)$	parents, children, ancestors of X ($pa^{\mathbf{V}}_{\mathcal{M}}, anc^{\mathbf{V}}_{\mathcal{M}}$ restricted to endogenous)
$pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S})$	set form $\bigcup_{X \in \vec{S}} pa^{\mathbf{V}}_{\mathcal{M}}(X)$
$\mathcal{P}(\cdot)$	powerset
$\mathcal{G}_{\mathcal{M}}$	causal DAG induced by $\mathcal{M}$
Values and domains	
x, y, x_i	values of single variables
$\vec{x}, \vec{z}, \vec{s}$	joint value tuples for $\vec{X}, \vec{Z}, \vec{S}$
$\vec{u}$	environment (joint value of $\mathbf{U}$)
$\vec{o}$	joint outcome (joint value of $\mathbf{O}$)
$\vec{o}_{\vec{u},\vec{s}}$	unique joint configuration of $(\mathcal{M}_{do(\vec{S}=\vec{s})}, \vec{u})$
$\mathcal{X}_i$	domain of single variable X_i
$\vec{\mathcal{X}}, \vec{\mathcal{S}}$	joint-value domains for $\vec{X}, \vec{S}$ (canonical); $\vec{\mathcal{X}}_{pa(X)}$ analogously for $pa(X)$
$\vec{\mathcal{U}}, \vec{\mathcal{V}}, \vec{\mathcal{O}}$	joint-value domains for $\mathbf{U}, \mathbf{V}, \mathbf{O}$
$\vec{o}_{\mathbf{U}}, \vec{o}_{\vec{Z}}$	projection of $\vec{o}$ onto components $\mathbf{U}$, $\vec{Z}$
SCM, settings, intervention	
$\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, \mathbb{P}_{\mathbf{U}} \rangle$	structural causal model
$\mathcal{F} = \{F_X : X \in \mathbf{V}\}$	structural equations
$F_X : \vec{\mathcal{X}}_{pa(X)} \to \mathcal{X}_X$	structural equation for X
$F_X^{-1}(x)$	preimage of value x under F_X
$\mathtt{w} : \mathbf{V} \to \bigcup_{X_i \in \mathbf{V}} \mathcal{X}_i$	world (assignment of endogenous values)
$\mathtt{w}_{\vec{u}}, \mathtt{w}_{\vec{u}}(\vec{Z})$	actual world induced by $\vec{u}$; actual values of $\vec{Z}$ (restriction of $\mathtt{w}_{\vec{u}}$ to $\vec{Z}$)
$\mathtt{w}^*_{\vec{u}}$	counterfactual world (after intervention)
$(\mathcal{M}, \vec{u})$	setting
$\mathcal{M}_{do(\vec{X}=\vec{x})}$	intervened SCM
$\mathcal{M}, \vec{u} \models \varphi$	event φ holds in $(\mathcal{M}, \vec{u})$
$\mathcal{M}, \vec{u} \models [\vec{X} = \vec{x}]\varphi$	φ holds in $(\mathcal{M}_{do(\vec{X}=\vec{x})}, \vec{u})$
Actual causation	
$(Y \in \mathcal{Y}^*)$, $(Y \in \hat{\mathcal{Y}})$	target event, counterfactual target ($\hat{\mathcal{Y}} := \mathcal{Y} \setminus \mathcal{Y}^*$)
$(\vec{X} = \vec{x})$	candidate (actual) cause
$\vec{Z}$	contingency set ($\vec{Z} \subseteq \mathbf{V} \setminus \vec{X}$)
$(\vec{Z}, \vec{X}, \vec{x}', \mathcal{Y}^*)$	counterfactual witness tuple
AC1, AC2, AC2*, AC3	modified HP conditions for actual causation
Witness targets (Section 4.1)	
$\hat{\mathcal{O}}^{pa}_{\mathcal{M}}(X, \hat{\mathcal{O}})$	parent targets (Definition 1)
$\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}})$	descendant targets (Definition 2)
$\hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$	global targets (Definition 3)
Grammar and witness map (Section 4.2)	
$\Sigma_{\mathcal{M}}(Y)$	alphabet $anc^{\mathbf{V}}_{\mathcal{M}}(Y) \cup \{Y\}$ of the grammar
$G_{\mathcal{M}}(Y), \mathcal{L}(G_{\mathcal{M}}(Y))$	ancestor intervention grammar, language (Definition 4)
$(P_{\text{keep}}), (P_{\text{unfold}})$	keep / unfold productions
$\vec{S}_{\text{unfold}}, \vec{S}_{\text{keep}}$	unfold / keep partition of a node in a derivation step
$\Rightarrow_G, \Rightarrow^+_G$	one-step / multi-step derivation
$\mathcal{A}_{\mathcal{M}}(Y)$	ancestral map (Definition 5)
$ch_{\mathcal{A}}(\vec{S})$	children of node $\vec{S}$ in the ancestral map
$\mathcal{W}_{\mathcal{M}}(Y \in \hat{\mathcal{Y}})$	witness map (Definition 6)
$\mathcal{C}$	set of actual causes returned by Algorithm 1

B Extended Proofs

Proof of Theorem 1 (Witness Target Membership)

Proof. We proceed by induction on the reverse topological order of the DAG, traversing from the target Y upward to the roots of $\mathcal{M}_{do(\vec{S})}$.

(⇒) Suppose $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$. By Definition 3, $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(R,\hat{\mathcal{Y}})$ for every root $R \in \mathbf{U} \cup \vec{S}$. Since every endogenous variable has a parent, following parents upward from Y through non-intervened variables reaches such a root along a directed path $R \to X_1 \to \cdots \to X_k = Y$; since $\hat{\mathcal{O}}^{pa}(X_j,\hat{\mathcal{O}}) \subseteq \hat{\mathcal{O}}$, the recursion of Definition 2 gives $\hat{\mathcal{O}}^{\downarrow}(R) \subseteq \hat{\mathcal{O}}^{\downarrow}(X_1) \subseteq \cdots \subseteq \hat{\mathcal{O}}^{\downarrow}(Y)$. Hence the base case of Definition 2 at $X = Y$ requires $(\vec{o}_{\vec{u},\vec{s}})_Y \in \hat{\mathcal{Y}}$, i.e. $\mathtt{w}^*_{\vec{u}}(Y) \in \hat{\mathcal{Y}}$, which gives $\mathcal{M}_{do(\vec{S}=\vec{s})},\vec{u} \models (Y \in \hat{\mathcal{Y}})$.

(⇐) Suppose $\mathcal{M}_{do(\vec{S}=\vec{s})},\vec{u} \models (Y \in \hat{\mathcal{Y}})$, i.e. $\mathtt{w}^*_{\vec{u}}(Y) \in \hat{\mathcal{Y}}$. Since $\mathcal{M}$ is acyclic, $\vec{o}_{\vec{u},\vec{s}}$ is the unique joint configuration consistent with $\vec{u}$ and the structural equations of $\mathcal{M}_{do(\vec{S}=\vec{s})}$. We show $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X,\hat{\mathcal{Y}})$ for every $X \in \mathbf{V} \cup \mathbf{U}$ by induction on reverse topological order (from Y upward): **Base case** $X = Y$: By the first branch of Definition 2, $\hat{\mathcal{O}}^{\downarrow}(Y) = \{\vec{o} \in \vec{\mathcal{O}} \mid \vec{o}_Y \in \hat{\mathcal{Y}}\}$, and $(\vec{o}_{\vec{u},\vec{s}})_Y = \mathtt{w}^*_{\vec{u}}(Y) \in \hat{\mathcal{Y}}$ by assumption. **Inductive step**: Assume the claim holds for all children $Z \in ch(X) \setminus \vec{S}$. By definition, $\hat{\mathcal{O}}^{\downarrow}(X) = \bigcap_{Z \in ch(X)\setminus\vec{S}} \hat{\mathcal{O}}^{pa}(Z,\, \hat{\mathcal{O}}^{\downarrow}(Z))$. For each such Z, the inductive hypothesis gives $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}^{\downarrow}(Z)$. Furthermore, the structural equation F_Z evaluated at the actual parent configuration $(\vec{o}_{\vec{u},\vec{s}})_{pa(Z)}$ produces $(\vec{o}_{\vec{u},\vec{s}})_Z$ by construction, so $(\vec{o}_{\vec{u},\vec{s}})_{pa(Z)} \in F_Z^{-1}((\vec{o}_{\vec{u},\vec{s}})_Z)$ and hence $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}^{pa}(Z,\, \hat{\mathcal{O}}^{\downarrow}(Z))$ (Definition 1). The intersection over all non-intervened children yields $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}^{\downarrow}(X)$. **Leaves** $ch(X) = \emptyset$: The second branch of Definition 2 sets $\hat{\mathcal{O}}^{\downarrow}(X) = \vec{\mathcal{O}}$, which trivially contains $\vec{o}_{\vec{u},\vec{s}}$. Taking the intersection over all roots $R \in \mathbf{U} \cup \vec{S}$ (Definition 3) yields $\vec{o}_{\vec{u},\vec{s}} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$. □

Proof of Proposition 3 (Grammar–Graph Equivalence)

Proof. We show that the closed-form expression of Proposition 3 characterises exactly the one-step derivates of $\vec{S}$ under $G_{\mathcal{M}}(Y)$. ($\supseteq$) Let $\vec{S}'' \in \mathcal{P}(pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S}') \cup (\vec{S} \setminus \vec{S}')) \setminus \{\emptyset\}$ for some non-empty $\vec{S}' \subseteq \vec{S}$. Set $\vec{S}_{\text{unfold}} := \vec{S}'$ and $\vec{S}_{\text{keep}} := \vec{S} \setminus \vec{S}'$, so that the candidate set in (5) is $pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S}') \cup (\vec{S} \setminus \vec{S}')$ and $\vec{S}''$ is a non-empty subset of this candidate, hence $\vec{S} \Rightarrow_G \vec{S}''$, i.e. $\vec{S}'' \in ch_{\mathcal{A}}(\vec{S})$. ($\subseteq$) Let $\vec{S}'' \in ch_{\mathcal{A}}(\vec{S})$, i.e. $\vec{S} \Rightarrow_G \vec{S}''$ by Definition 5, via partition $\vec{S}_{\text{unfold}} \dot\cup \vec{S}_{\text{keep}}$ with $\vec{S}_{\text{unfold}} \neq \emptyset$. Setting $\vec{S}' := \vec{S}_{\text{unfold}}$ in the closed-form union implies $\vec{S}_{\text{keep}} = \vec{S} \setminus \vec{S}'$ and $\vec{S}''$ being a non-empty subset of $pa^{\mathbf{V}}_{\mathcal{M}}(\vec{S}') \cup (\vec{S} \setminus \vec{S}')$. Hence, $\vec{S}''$ lies in the right-hand side. □

Proof of Lemma 5 (AC-Witness Identification)

Proof. **(AC1):** By assumption, $\mathcal{M},\vec{u} \models (Y \in \mathcal{Y}^*)$. Moreover, $\mathcal{M},\vec{u} \models (\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ holds trivially, since $\mathtt{w}_{\vec{u}}$ is the actual world induced by $\vec{u}$. Hence $\mathcal{M},\vec{u} \models (\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X})) \wedge (Y \in \mathcal{Y}^*)$, which is $(AC1)$. **(AC2)/(AC2*):** By Corollary 2, the existence of $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{X},\vec{Z})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ and $\vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$ gives the existence of $\vec{x}' \in \vec{\mathcal{X}}$ such that $\mathcal{M},\vec{u} \models [\vec{X} = \vec{x}', \vec{Z} = \mathtt{w}_{\vec{u}}(\vec{Z})](Y \in \hat{\mathcal{Y}})$. It remains to verify $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$. If $\vec{x}' = \mathtt{w}_{\vec{u}}(\vec{X})$, then $\mathcal{M}_{do(\vec{X}=\vec{x}',\vec{Z}=\mathtt{w}_{\vec{u}}(\vec{Z}))},\vec{u}$ agrees with the actual setting $(\mathcal{M},\vec{u})$ by the *consistency* of structural equations, so Y takes its actual value $\mathtt{w}_{\vec{u}}(Y) \in \mathcal{Y}^*$, contradicting $Y \in \hat{\mathcal{Y}} = \mathcal{Y} \setminus \mathcal{Y}^*$. Therefore $\vec{x}' \neq \mathtt{w}_{\vec{u}}(\vec{X})$, and $(\vec{Z},\vec{X},\vec{x}',\mathcal{Y}^*)$ constitutes a valid counterfactual witness for (AC2). For (AC2*), the same argument applies with $\vec{Z} = \emptyset$. In both cases, $(\vec{X} = \mathtt{w}_{\vec{u}}(\vec{X}))$ is a but-for or actual cause, respectively, upon applying the minimality filter $(AC3)$. □

B.1 Counterexample to the Necessity of Proposition 4

The path-disjointness condition of Proposition 4 is sufficient but not necessary for $\vec{S} \in \mathcal{A}_{\mathcal{M}}(Y)$. Consider the DAG with edges $A \to B$, $B \to X_1$, $B \to X_2$, $X_1 \to Y$, $X_2 \to Y$. The ancestor set $\vec{S} = \{A, B\}$ is reachable in $\mathcal{A}_{\mathcal{M}}(Y)$ via the derivation

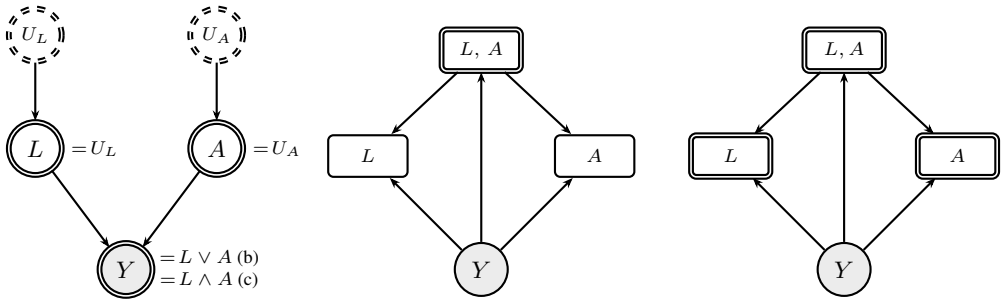
$$\{Y\} \;\Rightarrow\; \{X_1, X_2\} \;\Rightarrow\; \{B, X_2\} \;\Rightarrow\; \{A, B\},$$

where the diamond structure allows B to re-enter the set after being unfolded (the last step unfolds both B and X_2, with $pa(B) \cup pa(X_2) = \{A\} \cup \{B\} = \{A, B\}$). Yet every directed path from A to Y passes through B, so the path-disjointness condition fails on A while $\{A, B\} \in \mathcal{A}_{\mathcal{M}}(Y)$. Reachability of an ancestor set is therefore generated by a non-local sequence of grammar steps that can re-introduce intermediate variables; it is not captured by any local property of cause-to-effect path structure.

C Examples

C.1 Forest-Fire Example

We use the forest-fire scenario to contrast two qualitatively different causal structures that share the same DAG but differ in the structural equation for Y. Both cases use the DAG $L \to Y \leftarrow A$ with binary variables $\mathcal{X}_i = \{0, 1\}$, independent exogenous roots U_L, U_A, and equations $F_L = U_L$, $F_A = U_A$. The setting $\vec{u}$ has $U_L = U_A = 1$, so $\mathtt{w}_{\vec{u}} = (L, A, Y) = (1, 1, 1)$. We explain $\mathcal{Y}^* = \{1\}$, hence $\hat{\mathcal{Y}} = \{0\}$.



(a) **(b)** $F_Y = L \vee A$ **(c)** $F_Y = L \wedge A$

Figure 4: (a) Shared DAG for the forest-fire scenario. All variables in the actual world are double-bordered as they are true $(U_L, U_A, L, A, Y) = (1, 1, 1, 1, 1)$. (b) Witness map for Case I ($F_Y = L \vee A$, symmetric overdetermination): singleton nodes $\{L\}$ and $\{A\}$ fail the node-level pre-check; only the joint node $\{L, A\}$ passes. (c) Witness map for Case II ($F_Y = L \wedge A$, conjunctive causation): all non-root nodes pass but the joint node is non-minimal.

Example 2 (Forest fire: overdetermination and joint causes)**.** The DAG, actual world, and witness maps for both cases are shown in Figure 4.

Case I: Overdetermination ($F_Y = L \vee A$). Both a lightning strike ($L = 1$) and an arsonist ($A = 1$) simultaneously ignite fires, each alone sufficient to burn down the forest. Backward propagation gives $\hat{\mathcal{O}}_{do(\emptyset)}(\{0\}) = \{(0, 0, 0)\}$; the actual world $(1, 1, 1)$ is absent.

No singleton but-for causes. At node $\vec{S} = \{L\}$: $do(L = 0)$ gives $F_Y = 0 \vee 1 = 1$, so $\mathtt{w}^*_{\vec{u}} = (0, 1, 1) \notin \hat{\mathcal{Y}}$: $\times$. The node-level pre-check (Remark 6(ii)) fails because every $\vec{o} \in \hat{\mathcal{O}}_{do(\{L\})}(\{0\})$ requires $U_A = 0$ (since $A = U_A$ keeps $F_Y = 0 \vee 1 = 1$ unless $A = 0$), whereas $\vec{u}$ has $U_A = 1$. By symmetry, $\vec{S} = \{A\}$ fails identically.

One joint actual cause. At the joint node $\vec{S} = \{L, A\}$, both L and A are intervened, so U_L and U_A are disconnected from Y: the global target leaves the environment unconstrained and, by propagation from the intervened roots L and A, contains exactly the configurations with $L = A = Y = 0$. The node-level pre-check therefore passes for any $\vec{u}$, whereas the partition $\vec{X} = \{L\}$, $\vec{Z} = \{A\}$ fails,

since it would require $A = \mathtt{w}_{\vec{u}}(A) = 1$. Partition $\vec{X} = \{L, A\}$, $\vec{Z} = \emptyset$: $do(L = 0, A = 0) \Rightarrow Y = 0 \vee 0 = 0 \in \hat{\mathcal{Y}}$: ✓. No proper subset of $\{L, A\}$ satisfies AC2, so minimality holds: $(L{=}1, A{=}1)$ *is the unique joint actual cause of* $(Y{=}1)$. Following Halpern [16] (Definition 6.1.6), responsibility under the modified HP definition is $\rho = 1/(|\vec{Z}| + |\vec{X}|)$ for a cause $\vec{X}$ with contingency $\vec{Z}$; here $|\vec{X}| = 2$, $\vec{Z} = \emptyset$, giving $\rho = 1/2$ per conjunct.

Case II: Conjunctive causation ($F_Y = L \wedge A$). Both ignition sources are now necessary. $\hat{\mathcal{O}}_{do(\emptyset)}(\{0\}) = \{(0, 0, 0), (1, 0, 0), (0, 1, 0)\}$; the actual world $(1, 1, 1)$ is again absent.

Two singleton but-for causes. For $\vec{S} = \{L\}$: $do(L = 0)$ gives $F_Y = 0 \wedge 1 = 0 \in \hat{\mathcal{Y}}$: ✓. Symmetrically, $\vec{S} = \{A\}$: ✓. Both singletons are minimal, so $(L{=}1)$ and $(A{=}1)$ are the *two distinct but-for causes* of $(Y{=}1)$, each with $\vec{Z} = \emptyset$ and responsibility $\rho = 1$.

Joint node pruned by minimality. The node $\vec{S} = \{L, A\}$ also passes the pre-check, and $do(L = 0, A = 0) \Rightarrow Y = 0$: ✓. However, $\{L\} \subsetneq \{L, A\}$ and $\{L\} \in \mathcal{C}$, so Phase 3 removes $\{L, A\}$ as non-minimal.

Summary. In Case I, the unique cause is the joint conjunction $(L{=}1, A{=}1)$. In Case II, each variable is individually necessary, yielding two singleton but-for causes. In both cases the result is read off the witness map without any forward simulation over alternative values.

C.2 Billy-and-Suzy Example

Example 3 (Billy and Suzy: preemption and contingency)**.** The Billy-and-Suzy scenario is the canonical illustration of *preemption*: an actual cause whose counterfactual dependence is masked by a backup process, so that a non-empty contingency set is required to reveal it. Suzy ($ST = 1$) and Billy ($BT = 1$) both throw rocks at a bottle. Suzy's rock arrives first: her hit ($SH = 1$) shatters the bottle ($BS = 1$), preempting Billy's rock from ever connecting ($BH = 0$).

The DAG and witness map are shown in Figure 1. The SCM has binary variables $\mathcal{X}_i = \{0, 1\}$, independent exogenous roots U_{ST}, U_{BT}, and structural equations

$$F_{SH} = ST,\ F_{BH} = BT \wedge \neg SH,\ F_{BS} = SH \vee BH.$$

In the setting $\vec{u}$ with $U_{ST} = U_{BT} = 1$ the actual world is

$$\mathtt{w}_{\vec{u}} = (ST, BT, SH, BH, BS) = (1, 1, 1, 0, 1).$$

We explain $\mathcal{Y}^* = \{1\}$, so $\hat{\mathcal{Y}} = \{0\}$.

Observational global targets. Backward propagation of $\hat{\mathcal{Y}} = \{0\}$ through $F_{BS} = SH \vee BH$ requires $SH = BH = 0$. The set $\hat{\mathcal{O}}_{do(\emptyset)}(\{0\})$ therefore contains exactly those worlds with $SH = BH = 0$; since $\mathtt{w}_{\vec{u}}$ has $SH = 1$, the actual world is absent, confirming that AC1 holds and the search is non-trivial.

No singleton but-for causes. All four singleton but-for tests fail: $do(ST{=}0)$ gives $BH = 1 \wedge \neg 0 = 1$, so $BS = 0 \vee 1 = 1$: ×. $do(BT{=}0)$ gives $BH = 0 \wedge \neg 1 = 0$, so $BS = 1 \vee 0 = 1$: ×. $do(SH{=}0)$ gives $BH = 1 \wedge \neg 0 = 1$, so $BS = 0 \vee 1 = 1$: ×. $do(BH{=}0)$ leaves $BS = SH \vee 0 = 1$: ×. Suzy's preemption of Billy's path, and Billy's backup of Suzy's, each mask the singleton counterfactual dependencies.

Four joint but-for causes. Moving to pairs in $\mathcal{A}_{\mathcal{M}}(BS)$, the node-level pre-check (Remark 6(ii)) passes for exactly the four double-bordered nodes visible in Figure 1(b). Setting both members to 0 in each case witnesses $\hat{\mathcal{Y}} = \{0\}$:

- $do(ST{=}0, BT{=}0) \Rightarrow SH = 0,\ BH = 0,\ BS = 0$. ✓
- $do(ST{=}0, BH{=}0) \Rightarrow SH = 0,\ BS = 0$. ✓
- $do(SH{=}0, BT{=}0) \Rightarrow BH = 0,\ BS = 0$. ✓
- $do(SH{=}0, BH{=}0) \Rightarrow BS = 0$. ✓

No pair is a strict subset of another, so all four are minimal: they are the *four joint but-for causes* of $(BS = 1)$.

Two actual causes via non-empty contingency. The two red-highlighted nodes in Figure 1(b), which are $\{ST, BH\}$ and $\{SH, BH\}$, additionally admit singleton-cause partitions with contingency $\vec{Z} = \{BH\}$, where $\mathtt{w}_{\vec{u}}(BH) = 0$.

Actual cause test for $ST = 1$, node $\vec{S} = \{ST, BH\}$, partition $\vec{X} = \{ST\}$, $\vec{Z} = \{BH\}$: In $\mathcal{M}_{do(ST=0,\ BH=0)}$: $SH = 0$, $BH = 0$ (fixed by contingency), $BS = 0 \vee 0 = 0 \in \hat{\mathcal{Y}}$. Hence there exists $\vec{o} \in \hat{\mathcal{O}}_{do(\{ST,BH\})}(\{0\})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ and $\vec{o}_{BH} = \mathtt{w}_{\vec{u}}(BH)$: ✓. Since $\{ST\}$ is minimal, $(ST = 1)$ is an *actual cause* of $(BS = 1)$ with responsibility $\rho = 1/(|\vec{Z}| + |\vec{X}|) = 1/2$.

Actual cause test for $SH = 1$, node $\vec{S} = \{SH, BH\}$, partition $\vec{X} = \{SH\}$, $\vec{Z} = \{BH\}$: In $\mathcal{M}_{do(SH=0,\ BH=0)}$: $BH = 0$ (fixed by contingency), $BS = 0 \vee 0 = 0 \in \hat{\mathcal{Y}}$. Hence there exists $\vec{o} \in \hat{\mathcal{O}}_{do(\{SH,BH\})}(\{0\})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ and $\vec{o}_{BH} = \mathtt{w}_{\vec{u}}(BH)$: ✓. Since $\{SH\}$ is minimal, $(SH = 1)$ is an *actual cause* of $(BS = 1)$ with responsibility $\rho = 1/(|\vec{Z}| + |\vec{X}|) = 1/2$.

Intuitively, Suzy's throw $(ST = 1)$ and her hit $(SH = 1)$ are both actual causes: each alone, with Billy's miss held fixed at its actual value, suffices to counterfactually explain the broken bottle. Their equal responsibilities reflect that the contingency $\{BH\}$ is the same for both, a consequence of the preemption structure $F_{BH} = BT \wedge \neg SH$.

Why $(BT = 1)$ **is not an actual cause.** For every node $\vec{S} \in \mathcal{A}_{\mathcal{M}}(BS)$ containing BT and every subset $\vec{X}$ including BT with contingency $\vec{Z} = \vec{S} \setminus \vec{X}$, holding $\vec{Z}$ at actual values and setting $do(BT = 0)$ leaves Suzy's path $ST \to SH \to BS$ intact. Since $SH = 1$ in the actual world, $BS = SH \vee BH = 1$ regardless of BT's value, and no contingency over non-Suzy variables can block this route without altering actual values. No membership test passes for any node containing BT: $(BT = 1)$ is *not* an actual cause, correctly reflecting that Billy's rock never reached the bottle.

Summary. Algorithm 1 returns two actual causes, $(ST = 1)$ and $(SH = 1)$, each with contingency $\vec{Z} = \{BH\}$ and responsibility $\rho = 1/2$. The four joint but-for causes and the two actual causes are read off directly from Figure 1(b) without any forward simulation over alternative values: the contingency set $\vec{Z} = \{BH\}$ is *discovered automatically* by the witness map traversal.

D Constraint-Based Outer Approximation

As an efficient alternative to computing the exact global targets, we can track *per-variable marginal subsets* $\hat{\mathcal{O}} = \{\hat{\mathcal{X}}_i \mid X_i \in \mathbf{U} \cup \mathbf{V},\ \hat{\mathcal{X}}_i \subseteq \mathcal{X}_i\}$, which is a collection of individual domain subsets rather than joint configurations. This yields an *outer approximation* of the global targets that is cheaper to compute and store, at the cost of possible false positives. We re-use the operator names $\hat{\mathcal{O}}^{pa}_{\mathcal{M}}$ and $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}$ from Definitions 1–2 for their constraint analogues throughout this appendix; the marginal nature of the codomain is implicit in the input being a constraint collection rather than a joint set.

Definition 7 (parent constraints)**.** *For SCM* $\mathcal{M}$*, variable* $X \in \mathbf{V}$*, and constraint collection* $\hat{\mathcal{O}}$*, the* ***parent constraints*** $\hat{\mathcal{O}}^{pa}_{\mathcal{M}}(X, \hat{\mathcal{O}})$ *are defined by marginal preimage projection:*

$$\hat{\mathcal{O}}^{pa}_{\mathcal{M}}(X, \hat{\mathcal{O}}) := \Big(\big\{ \big(F_X^{-1}(\hat{\mathcal{O}}_X)\big)_i \mid X_i \in pa(X) \big\} \cup \big\{ \mathcal{X}_j \mid X_j \in (\mathbf{U} \cup \mathbf{V}) \setminus (X \cup pa(X)) \big\} \Big) \sqcap \hat{\mathcal{O}}, \tag{7}$$

where $(F_X^{-1}(\hat{\mathcal{O}}_X))_i := \{x_i \in \mathcal{X}_i \mid \exists\, \vec{o}_{pa} \in F_X^{-1}(\hat{\mathcal{O}}_X)$ *with* $(\vec{o}_{pa})_i = x_i\}$ *and* $\sqcap$ *denotes component-wise intersection.*

Definition 8 (descendant constraints)**.** *For SCM* $\mathcal{M}$*, target variable* $Y \in \mathbf{V}$*, target* $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$*, variable* $X \in \mathbf{V}$*, and intervention set* $\vec{S} \subseteq \mathbf{V}$*, the* ***descendant constraints*** $\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}})$ *are:*

$$\hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(X, \hat{\mathcal{Y}}) := \begin{cases} \{\mathcal{X}_i \mid X_i \in (\mathbf{U} \cup \mathbf{V}) \setminus Y\} \cup \{\hat{\mathcal{Y}}\} & \text{if } X = Y, \\ \{\mathcal{X}_i \mid X_i \in \mathbf{U} \cup \mathbf{V}\} & \text{if } ch(X) = \emptyset, \\ \displaystyle\bigsqcap_{Z \in ch(X) \setminus \vec{S}} \hat{\mathcal{O}}^{pa}_{\mathcal{M}}(Z, \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(Z, \hat{\mathcal{Y}})) & \text{otherwise.} \end{cases} \tag{8}$$

Definition 9 (global constraints)**.** *The* ***global constraints*** $\hat{\mathcal{O}}^{\text{con}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}) := \bigcap_{R\in\mathbf{U}\cup\vec{S}} \hat{\mathcal{O}}^{\downarrow}_{\mathcal{M},do(\vec{S})}(R,\hat{\mathcal{Y}})$ *are the component-wise intersection of descendant constraints across all roots of* $\mathcal{M}_{do(\vec{S})}$ *(Definition 3).*

Remark 7 (outer approximation)**.** *The projection* $(F_X^{-1}(\hat{\mathcal{O}}_X))_i$ *preserves* marginal *feasibility but not* joint *feasibility: every* $x_i \in (F_X^{-1}(\hat{\mathcal{O}}_X))_i$ *admits some complete parent configuration* $\vec{o}_{pa}$ *with* $(\vec{o}_{pa})_i = x_i$*, but the valid configurations for different parent components may be incompatible. As a consequence, the global constraints satisfy* $\hat{\mathcal{O}}^{\text{con}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}) \supseteq \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$*: they are an outer approximation of the exact global targets. False positives arise precisely when some mechanism on the path from* X *to* Y *has a non-product-structured preimage (e.g., XOR:* $F_C(A,B) = A \oplus B$ *with* $\hat{\mathcal{C}} = \{1\}$ *gives projections* $\hat{\mathcal{A}} = \hat{\mathcal{B}} = \{0,1\}$*, yet* $(0,0)$ *passes both checks but violates the target). For mechanisms with product-structured preimages (e.g., AND, OR, or any parent-separable function), the approximation is exact.*

The global constraints can replace the global targets in Algorithm 1 as an efficient *screening* step for a partition $(\vec{X},\vec{Z})$ of a node $\vec{S}$: if $\vec{u}(U) \notin (\hat{\mathcal{O}}^{\text{con}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}))_U$ for some $U \in \mathbf{U}$, or $\mathtt{w}_{\vec{u}}(Z_j) \notin (\hat{\mathcal{O}}^{\text{con}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}}))_{Z_j}$ for some $Z_j \in \vec{Z}$, then no witness exists (sound exclusion). If all component checks pass, a forward-simulation verification $\mathcal{M}_{do(\vec{X}=\vec{x}',\vec{Z}=\mathtt{w}_{\vec{u}}(\vec{Z}))}, \vec{u} \models (Y \in \hat{\mathcal{Y}})$ confirms or refutes the candidate; this step needs a candidate value $\vec{x}'$ and therefore, in general, reintroduces a search over alternative values. The screen thus trades exactness for space: with $D := \max_i |\mathcal{X}_i|$, the constraint collection needs $O((|\mathbf{U}| + |\mathbf{V}|)\cdot D)$ space, and a backward pass over the mechanism tables takes $O(|\mathbf{V}|\cdot D^{\Delta+1})$ time, with Δ the largest number of endogenous parents (Appendix E.1), avoiding the potentially exponential joint target representation.

E Full Complexity Analysis

We give the complete cost analysis of Algorithm 1 across three parameterisations of the search space. Throughout, write $a := |anc^{\mathbf{V}}_{\mathcal{M}}(Y)|$, $D := \max_i |\mathcal{X}_i|$, $\Delta := \max_X |pa^{\mathbf{V}}_{\mathcal{M}}(X)|$, $p := |\mathbf{V}|$, and $o := |\mathbf{U}|$.

E.1 Setup: Base Costs

Two components determine the per-node cost in Phase 2 and are shared by all three parameterisations below.

Target representation. The global targets $\hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ are subsets of the joint domain $\vec{\mathcal{O}} = \prod_i \mathcal{X}_i$, which has cardinality $|\vec{\mathcal{O}}| \le D^{o+p}$. With an explicit (enumerated) set representation, storing a single global target requires $O(D^{o+p})$ space; this is the fundamental bottleneck of exact computation and motivates the factored outer approximation of Appendix D. The required preimage projections $(F^{-1}_{X_i}(x_i))_j$ can be precomputed once in $O(p\cdot D^{\Delta+1})$ time and stored as lookup tables of total size $O(p\cdot D^{\Delta+1})$. Phase 1 performs one bottom-up pass over the p endogenous variables at worst-case cost $O(p\cdot D^{o+p}\cdot\Delta)$.

Per-node target recomputation. Per-node recomputation in Phase 2 (Remark 2) costs

$$T_{\text{target}} = O\Big(|anc_{\mathcal{M}}(\vec{S})|\cdot|\vec{\mathcal{O}}|\cdot\Delta\Big) \le O\big(a\cdot D^{o+p}\cdot\Delta\big)\,.$$

The mechanism class modulates T_{target} substantially: for mechanisms with product-structured preimages (AND, OR, linear, monotone), the global targets factorise and $T_{\text{target}} = O(a\cdot D^{\Delta})$; for continuous variables $\mathcal{X}_i \subseteq \mathbb{R}$, targets are represented implicitly (polytopes, intervals, semialgebraic sets) at cost $T_{\text{target}} = O(a\cdot\rho)$, where ρ is the cost of a single preimage computation: $\rho = O(p)$ for linear mechanisms (preimage is a hyperplane), $\rho = O(1)$ for monotone mechanisms (marginal interval arithmetic, exact via Appendix D), and $\rho = O(\log(1/\varepsilon))$ for general smooth mechanisms (bisection). In all cases the logical structure of Algorithm 1 and the guarantees of Theorem 7 are unaffected; only the membership subroutine requires adaptation to the mechanism class.

E.2 Enumeration Cost under Three Parameterisations

The three parameterisations restrict the search space of the ancestral map in qualitatively different ways; the per-node cost T_{target} from Section E.1 applies uniformly in each case.

(i) Unrestricted. The ancestral map has at most 2^a nodes. At a node $\vec{S}$ of size s, the algorithm tests $2^s - 1$ subsets with exactly one membership query each (Corollary 2). The total number of membership queries is bounded by

$$\sum_{s=1}^{a} \binom{a}{s} \cdot (2^s - 1) < \sum_{s=0}^{a} \binom{a}{s} \cdot 2^s = 3^a.$$

The total enumeration cost is therefore exponential in a in the worst case.

(ii) Bounded depth d. Restricting traversal to nodes reachable within d derivation steps from $\{Y\}$, the analysis of case (i) applies with a replaced by $a_d \leq 2\Delta^d$ (the number of ancestors within d hops for bounded in-degree Δ), yielding a membership-query bound of $3^{2\Delta^d}$. At depth $d = 1$ the search is restricted to direct parents of Y and performs at most 3^Δ membership queries.

(iii) Bounded intervention size $|\vec{S}| \leq s$. Restricting to nodes of size at most s yields at most $\sum_{k=1}^{s} \binom{a}{k} = O(a^s)$ nodes, with at most $2^s - 1$ membership queries per node. For product-structured mechanisms the total cost is

$$O\big(a^s \cdot 2^s \cdot p \cdot D^\Delta\big)\,,$$

which is polynomial in p, a, and D for fixed s and Δ. For continuous or general mechanism classes, substituting $T_{\text{target}} = O(a \cdot \rho)$ gives the unified bound

$$O(a^s \cdot 2^s \cdot (p \cdot \rho + T_{\text{mem}}))\,,$$

where T_{mem} is the cost of a single membership query. Phase 3 verification requires checking at most 2^s strict subsets per candidate.

Expected-case behaviour. The worst-case bound $|\mathcal{A}_\mathcal{M}(Y)| \leq 2^a$ is loose in practice: ancestor subsets violating the path-disjointness condition of Proposition 4 are typically not reached, and those that are tend to fail the node-level pre-check (Appendix F(ii)) in overdetermined regimes. Empirically (Appendix G, Figure 6), the combined pruning of Remark 6(ii) and (iii) eliminates on average over 97% of witness-map nodes for $n \geq 10$ at densities $d \geq 0.5$ and 73–94% at $d = 0.25$. Under bounded in-degree Δ and bounded tree-width τ of $\mathcal{G}_\mathcal{M}$, paths from ancestors to Y are more constrained, so fewer subsets satisfy the path-disjointness condition; while this does not change the $O(a^s)$ asymptotic for case (iii), the implicit constant decreases with τ. A formal expected-case analysis under random-DAG models is beyond the present scope.

F Phase 2 Pruning Strategies

The three strategies are stated in Remark 6; we give the formal justifications here, including how each strategy behaves under continuous variable domains.

(i) Size-first traversal. Because the ancestral map $\mathcal{A}_\mathcal{M}(Y)$ is a finite DAG and actual causes form an antichain under $\subsetneq$ (AC3), processing nodes in non-decreasing order of $|\vec{S}|$ means that whenever a cause $\vec{X}' \in \mathcal{C}$ is confirmed, every node $\vec{S}'$ with $\vec{X}' \subsetneq \vec{S}'$ has not yet been processed, so strategy (iii) achieves its maximal pruning effect. This strategy operates purely on the lattice structure of the ancestral map and carries over unchanged to continuous variable domains. *Cost saving.* Combined with strategy (iii), size-first traversal makes Phase 3 redundant: the antichain invariant of $\mathcal{C}$ is maintained *during* Phase 2, eliminating the $O(|\mathcal{C}|^2 \cdot s_{\max})$ post-filter pairwise antichain comparisons (where $s_{\max} = \max_{\vec{X} \in \mathcal{C}} |\vec{X}|$) at zero per-query overhead.

(ii) Node-level pre-check. For any non-empty partition $\vec{X} \cup \vec{Z} = \vec{S}$, the existence of $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ and $\vec{o}_{\vec{Z}} = \mathtt{w}_{\vec{u}}(\vec{Z})$ implies in particular the existence of $\vec{o}' \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ with $\vec{o}'_{\mathbf{U}} = \vec{u}$ (taking $\vec{o}' := \vec{o}$). Contrapositively, failure of the latter weaker condition is a *sufficient* reason to skip the entire node, since no partition $(\vec{X}, \vec{Z})$ of $\vec{S}$ can pass the membership test. In redundantly overdetermined settings (e.g. $F_Y = A \vee C$ with independent $A = C = 1$), this eliminates every root-adjacent node whose intervention set is a strict subset of $pa^{\mathbf{V}}_{\mathcal{M}}(Y)$, while $pa^{\mathbf{V}}_{\mathcal{M}}(Y)$ itself typically passes the check. *Cost saving.* Each failed node $\vec{S}$ skips all $2^{|\vec{S}|} - 1$ partition queries at the cost of one membership query, a net saving of $2^{|\vec{S}|} - 2$ queries. Across the witness map, the Phase 2 query count drops from $\sum_{\vec{S}}(2^{|\vec{S}|} - 1)$ to $\mathcal{N} + \sum_{\vec{S}\text{ passes}}(2^{|\vec{S}|} - 1)$, where $\mathcal{N}$ is the number of pre-check failures. The savings dominate in overdetermined regimes (e.g. Forest-fire Case I in Appendix C.1, where two of three non-root nodes fail the pre-check).

The pre-check remains formally valid in continuous domains; only its *implementation* changes, since deciding the existence of $\vec{o} \in \hat{\mathcal{O}}_{\mathcal{M},do(\vec{S})}(\hat{\mathcal{Y}})$ with $\vec{o}_{\mathbf{U}} = \vec{u}$ becomes a real-valued feasibility problem rather than a hash-table lookup. The constraint-based outer approximation of Appendix D provides an efficient and often exact screening step in all three mechanism classes: for *monotone mechanisms*, marginal preimage projections are closed intervals and the check reduces to component-wise interval comparisons, which are exact (Remark 7); for *linear-Gaussian mechanisms*, the check reduces to a single LP feasibility query; for *general smooth mechanisms*, bisection or gradient-based root-finding locates a witness $\vec{x}'$ or certifies non-existence.

(iii) Superset pruning. If $\vec{X}' \subsetneq \vec{X}$ and $\vec{X}' \in \mathcal{C}$, then by (AC3) $\vec{X}$ cannot be minimal and would be discarded in Phase 3 regardless. Maintaining $\mathcal{C}$ as an antichain under $\subsetneq$ throughout Phase 2 therefore makes Phase 3 redundant when combined with strategy (i):

$$\vec{X} \in \big\{\emptyset \neq \vec{X} \subseteq \vec{S} \;\big|\; \nexists\, \vec{X}' \in \mathcal{C} \colon \vec{X}' \subsetneq \vec{X}\big\}.$$

Like strategy (i), superset pruning operates purely on the antichain structure of $\mathcal{C}$ and carries over unchanged to continuous variable domains. *Cost saving.* For each confirmed cause $\vec{X}' \in \mathcal{C}$ of size k and each later-processed node $\vec{S} \supseteq \vec{X}'$, the $2^{|\vec{S}|-k} - 1$ proper supersets of $\vec{X}'$ within $\vec{S}$ (every set extending $\vec{X}'$ by a non-empty subset of $\vec{S} \setminus \vec{X}'$) are skipped before the membership query. The savings compound multiplicatively in regimes with many small confirmed causes.

G Extended Experimental Results

G.1 Setup

All experiments use randomly generated Boolean SCMs. The causal DAG is sampled from an acyclic Erdős-Rényi model: each ordered pair of nodes is connected by a directed edge independently with probability d. Every structural equation F_X is a random AND/OR gate applied to the parents of X, with each parent input independently negated with probability 0.5. We vary the number of endogenous nodes $n \in \{3, 4, \ldots, 15\}$ and edge density $d \in \{0.25, 0.50, 0.75\}$, drawing 10 independent seeds per (n, d) configuration (390 instances in total). The target variable Y is chosen as the deepest node in the graph; the environment $\vec{u}$ is drawn uniformly at random, and target event $\mathcal{Y}^* = \{1\}$ is used throughout. TrAC is run with all three pruning strategies of Remark 6 implemented (size-first traversal, node-level pre-check, superset pruning). Individual runs are sequentially benchmarked on a single CPU core of an AMD Ryzen Threadripper 1900X 8-Core Processor with 32GB RAM.

Baselines. Reyd et al. [29] identify a single cause via forward simulation with *beam search* and *ISI* by sampling alternative variable values. This method supports non-Boolean and black-box SCMs but provides no soundness or completeness guarantee. Ibrahim and Pretschner [19] encode the HP conditions as propositional formulae and apply three solving strategies: Brute Force (BF), SAT, and MaxSAT. These methods are restricted to Boolean SCMs. ILP/ILP-WHY strategies are excluded as they require a Gurobi licence. Both baselines target a single (minimal) cause and are not designed for full enumeration; the runtime comparison is therefore indicative rather than strictly analogous to TrAC.

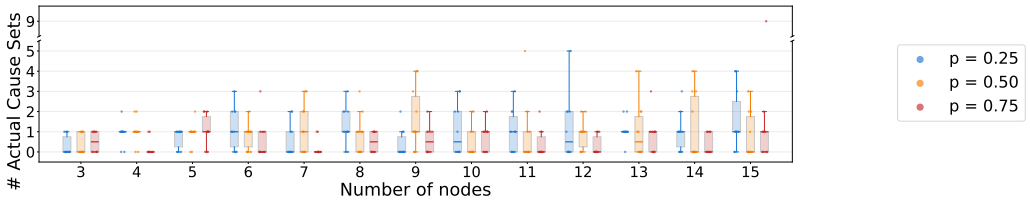


Figure 5: **Number of actual cause sets found by TrAC.** Box-and-whisker across 10 seeds (median line, $[25, 75]$-IQR box, whiskers, outlier markers). The number of causes grows moderately with graph size and edge density.

G.2 Results

Cause counts. Figure 5 shows the number of actual cause sets found by TrAC per instance. The number of causes grows moderately with both n and d. Even at moderate cause counts, a principled enumeration algorithm is indispensable for downstream aggregation tasks such as responsibility and blame, as the combinatorial structure of the witness map, not the number of causes alone, drives the difficulty of complete enumeration.

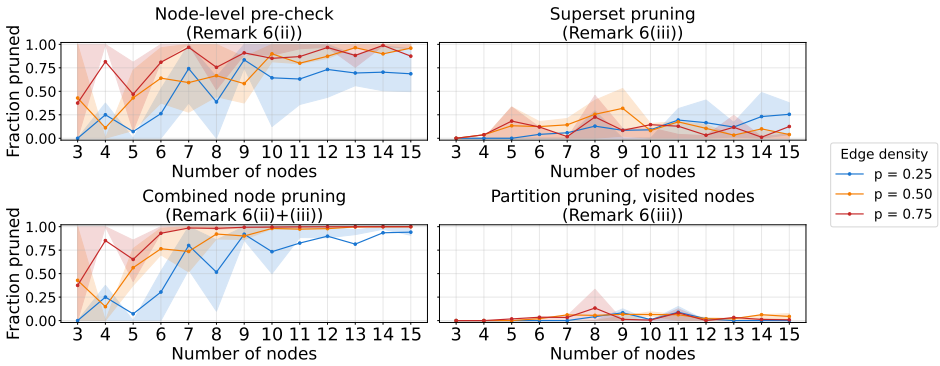


Figure 6: **Pruning rates.** Fraction of witness-map nodes eliminated by each strategy (mean $\pm$ $[25, 75]$-IQR), broken down by node count and density. The node-level pre-check dominates; combined with superset pruning, on average over 97% of nodes are eliminated for $n \geq 10$ at $d \geq 0.5$, and 73–94% at $d = 0.25$.

Pruning effectiveness. Figure 6 reports the fraction of work eliminated by each pruning strategy of Remark 6. The node-level pre-check (strategy (ii)) is by far the most effective: at $n \geq 10$ it prunes more than 96% of all witness-map nodes before any partition enumeration, reaching 99.98% at n=15, d=0.75. Superset pruning (strategy (iii)) provides additional savings at the partition level, while size-first traversal (strategy (i)) amplifies both by ensuring confirmed causes are discovered as early as possible. Together, these three strategies reduce the practical cost of Phase 2 by several orders of magnitude relative to exhaustive traversal.

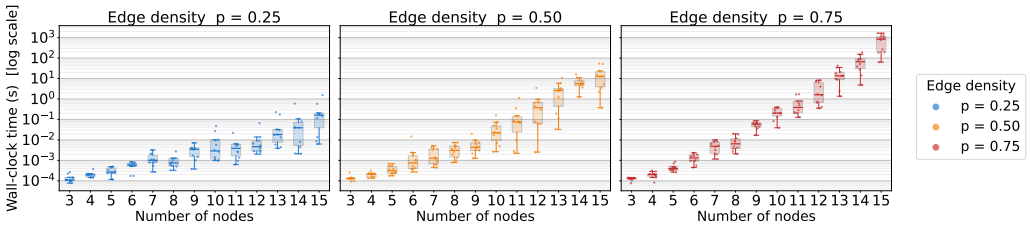


Figure 7: **TrAC wall-clock runtime on random Boolean SCMs.** Box-and-whisker across 10 seeds per (n, d) cell (median line, $[25, 75]$-IQR box, whiskers, outlier markers). Runtime scales steeply at high density (d=0.75) due to the larger witness maps induced by denser graphs.

TrAC standalone runtime. Figure 7 presents TrAC's wall-clock runtime for each (n, d) configuration. Runtime grows with both n and d: at n=15 the mean ranges from 0.30 s (d=0.25) to 18 s (d=0.50) and 773 s (d=0.75), with substantial inter-seed variance at high density visible in the widening IQR bands.

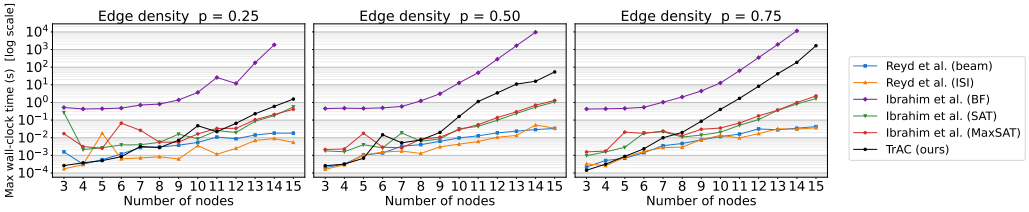


Figure 8: **Worst-case runtime comparison.** Maximum observed runtime per (n, d) cell. HP2SAT BF was excluded at n=15 due to exploding runtime and memory requirements; TrAC completes every instance, with a worst-case runtime of approximately 1,600 s at n=15, d=0.75.

Worst-case runtime. Figure 8 shows the maximum runtime per (n, d) cell, highlighting the tail behaviour of each method. HP2SAT BF exhibits the most extreme worst cases and was excluded at n=15 due to exploding runtime and memory requirements. TrAC's worst-case runtime grows with n and d but remains finite across all tested configurations.

H Roadmap and Open Problems

Context-specific independence and pruning. The node-level pre-check (Remark 6(ii)) has a precise algebraic interpretation that connects to context-specific independence (CSI) [7, 33]. Specifically, the parent target operator (Definition 1) filters joint configurations by checking, for each child X_i along the backward path, whether the parent tuple of a configuration lies in the joint preimage $F_{X_i}^{-1}(\vec{o}_i)$ at that configuration's child value $\vec{o}_i$. The *outer approximation* of Appendix D relaxes this joint check to a set-level operation that tracks per-variable marginal subsets via $(F_X^{-1}(\hat{\mathcal{O}}_X))_i$, and thereby already realises the marginal (component-wise) version of CSI-based pruning [28], making the precision gap between the exact and approximate algorithms transparent (see Remark 7). A formal characterisation of the conditions under which pre-check failure coincides with graph-theoretic CSI criteria remains an open problem.

Compact representations of global targets. Global targets are subsets of the joint domain $\vec{\mathcal{O}}$, requiring $O(D^{o+p})$ space in the worst case. For product-structured mechanisms (AND, OR, linear, monotone), targets factorise and storage reduces to $O(p \cdot D^{\Delta})$; for Boolean mechanisms, binary decision diagrams (BDDs) support polynomial-time set operations (intersection, union, complement) on the target sets, though BDD size can itself be exponential in the number of variables in the worst case; for linear mechanisms, polytopes in half-space representation reduce membership to LP feasibility. A systematic characterisation of the broadest mechanism classes admitting compact representations would eliminate the exponential storage bottleneck and enable the algorithm to scale to large acyclic models.

Responsibility, blame, harm, and risk. Algorithm 1 returns all actual causes for a fixed environment $\vec{u}$; responsibility and blame scores [9] and causal harm assessments [6] are defined as weighted aggregates over actual causes and contingency sets, so the witness map provides precisely the missing computational substrate for these downstream quantities. Extending the algorithm to a full distribution $P(\mathbf{U})$ amounts to marginalising over environments: for a fixed variable set $\vec{X}$ and value $\vec{x}$, the *probability that* $(\vec{X}{=}\vec{x})$ *is an actual cause of* $(Y \in \mathcal{Y}^*)$ equals the $P(\mathbf{U})$-measure of environments $\vec{u}$ satisfying $\mathtt{w}_{\vec{u}}(\vec{X}) = \vec{x}$, $\mathcal{M}, \vec{u} \models (Y \in \mathcal{Y}^*)$, and passing the membership test for $\vec{X}$ (with some contingency $\vec{Z}$) in Phase 2 with $\vec{X}$ retained after the minimality filter of Phase 3. This probabilistic lifting connects directly to the framework of probabilistic actual causation [13] and to nondeterministic extensions [3]: in both cases the witness map structure is preserved, with the binary membership predicate replaced by a measure over the set of witnessing environments.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper presents a sound and complete algorithm for tracing all actual causes. Individual contributions regarding witness targets are given in Section 4.1, specifically Definitions 1–3, Theorem 1, Corollary 2. Witness maps are outlined in Section 4.2, specifically Definitions 4-6, Propositions 3-4. The Tracing Actual Causes (TrAC) algorithm is introduced in Section 4.3 and Algorithm 1 with soundness and completeness proof (Theorem 7) and complexity analysis (Sec. E). Finally, experiments and baseline comparisons are presented in Sec. 4.4 and Appendix G.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Verifying a single candidate cause under the modified Halpern–Pearl definition is D_1^P-complete [1, 15], so exponential worst-case cost is inherent to exact enumeration. Runtime and storage limitations of TrAC are analysed in Sec. 4.3, Appendix E and the limitations paragraph of Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper presents a sound and complete algorithm for tracing all actual causes under the modified Halpern-Pearl definition of actual causality on acyclic SCM. (Stated in the abstract, Sec. 1 and Sec. 4). Individual definitions regarding witness targets are given in Section 4.1, Definitions 1–3, Theorem 1, Corollary 2. Witness maps are outlined in Section 4.2, specifically Definitions 4-6, Propositions 3-4. The complete Tracing Actual Causes (TrAC) algorithm is introduced in Section 4.3 and Algorithm 1 with soundness and completeness proof (Theorem 7) and complexity analysis (Sec. E).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details on tested parameter configurations, graph and data sampling, as well as repetitions are described in Sec. 4.4 and Appendix G. All code is available at `https://github.com/MoritzWillig/tracingActualCauses`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code and recorded runtime results are available at `https://github.com/MoritzWillig/tracingActualCauses`. The included README contains all instructions to run the benchmark and plot the figures.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental details on tested parameter configurations, graph and data sampling, as well as repetitions are described in Sec. 4.4 and Appendix G. Baselines algorithms are tested as provided by their respective code repositories.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Each (n, d) configuration is evaluated on 10 independently sampled instances (random graph, structural equations and environment), so the reported variability is across instances. Figs. 3 and 6 show the mean with $[25, 75]$-IQR bands, Figs. 7 and 5 box-and-whisker plots (median, IQR, whiskers, outliers), and Fig. 8 the worst case per configuration.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Sec. G describes the hardware of a single AMD Ryzen Threadripper 1900X 8-Core PC with 32GB RAM (no GPU required). Runtime comparisons to baselines, individual repetition runtimes and worst-case runtimes for TrAC are reported in Figures 3, 7 and 8.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research is purely theoretical/methodological. It involves no human subjects, no personally identifiable information or sensitive data. Experiments are conducted exclusively on synthetic data (randomly sampled causal graphs), and illustrative examples are

standard, well-established toy scenarios from the causal inference literature (rock-throwing, forest fires). We anticipate no direct ethical concerns, risks of discrimination or privacy violation arising from this work.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Actual causation underpins definitions of legal responsibility, moral blame, and the assessment of causal harm. Previous works in this area primarily focused on finding single smallest cause sets, blocking these tasks that either depend on finding *all* actual causes in a given setting or assume a particular case was already established a-priori. By enumerating all actual causes, TrAC may thus advance reasoning in all of these areas. Conversely, automated cause enumeration could be misused to attribute blame or responsibility mechanically; its output is only as reliable as the specified causal model, and conclusions drawn from a misspecified model can be misleading. TrAC should therefore support, not replace, human judgement. Relevant discussions are included in the abstract, introduction and Appendix H.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: Experiments are on purely synthetic data.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The baselines of Reyd et al. [29] (actualcauses, MIT licence) and Ibrahim and Pretschner [19] (HP2SAT1.0, MIT licence) are cited in the paper and used unchanged under their licences; their code is referenced in our repository. All other code was written by the authors with the help of coding agents.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The structure and license of the accompanying code repository, as well as the required commands for running experiments and plotting results are described in the README of the repository. All code is released under MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs have only been used for writing, editing, and formatting of the paper and code in a way that does not impact the originality, rigor or core methodology of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.