Local Causal Reasoning in Multiagent Systems (Extended Abstract)

Pinaki Chakraborty[1*], Tristan Caulfield[1*], and David Pym[2,1*]

[1] University College London, England, UK pinaki.chakraborty.22@ucl.ac.uk, t.caulfield@ucl.ac.uk, david.pym@sas.ac.uk
[2] Institute of Philosophy, University of London England, UK david.pym@sas.ac.uk
[*] corresponding author

Abstract. Causal reasoning is essential for the design, audit, and interpretation of decision-making in multi-agent systems. Recent developments have brought this need to the fore in multi-agent LLM systems, notably in retrieval-augmented generation (RAG), where techniques from information retrieval are used to augment model inference within modular workflows. We propose a behaviour-centric model of system configurations and a unified language for reasoning about such systems. Our framework introduces an intervention operator that captures the notion of *mechanism change*, reflecting interventionist views of causation, while a separation-logic-style conjunction supports local reasoning via explicit system interfaces, consistent with mechanistic accounts that explain phenomena through organized and modifiable parts. Agent policy changes are treated as interventions on the components they control, enabling counterfactual analysis and attribution of responsibility within the same logic. We define actual causation directly in this language and show, via time-unfolding of finite system runs, that our notion aligns with the Halpern-Pearl account of actual causation in the acyclic structural model induced by the run. We establish van Benthem-Hennessy-Milner-style correspondence results: a bisimulation that respects both transitions and interventions characterizes logical equivalence under finiteness assumptions. Thus, we integrate system evolution with modular decomposition within a single language: its modalities refer directly to configuration transitions, interventions on mechanisms, and interface-indexed decompositions. We apply the framework to a retrieval-augmented generation (RAG) workflow for LLM-based systems to specify explicit interfaces, model mechanism changes as interventions, and answer design-time causal queries such as, whether some admissible mechanism change guarantees a stated safety constraint while preserving invariants modularly across an interface.

Keywords: Logic · Transition systems · Distributed systems · Behaviour · Agency · Decision-making · Strategic reasoning · Causality · Interventions · Separation · Large language models · Retrieval-augmented generation

1 Introduction

Causal reasoning sits at the heart of explanation, control, and design in systems composed of many interacting parts. In contemporary deployments — from software microservice ecosystems [11,28] and cyber-physical controllers to multi-agent LLM workflows and increasingly the mechanistic interpretability of neural networks [19]. Practitioners do not only ask *'did X cause Y?'*; instead, they routinely ask *'what* change of mechanism or policy *will* guarantee *or* prevent Y*, while preserving invariants elsewhere?'*. This paper develops a logical framework that addresses that design-oriented question with minimal structural assumptions and with an explicit account of *mechanism change* as a first-class intervention. We integrate the logic with system dynamics and modular structure: its modalities speak directly about system transitions, interventions, and it respects a form of van Benthem-Hennessy-Milner correspondence [7].

The Halpern-Pearl (HP) programme [20,33] axiomatizes actual causation using structural equations and *value-setting* interventions. It underpins a large body of work in causal analysis. In many engineered systems, however, the natural unit of intervention is not a fixed value but a *rule*: a policy switch, a controller update, a threshold change, or a mode transition. Standard structural causal models (SCMs) can encode such régime changes by adding latent 'switch' variables or by moving to a family of models. However, such encodings obscure locality and make it difficult to reason *within a single logic* about *which* mechanism changes are admissible and which invariants they preserve.

Large systems are reasoned about *modularly* on the principle that well-designed interfaces allow change and verification of parts in isolation while preserving global behaviour. Motivated by this, we adopt a view that a system is a collection of components with observable behaviours; each component updates according to a local rule (mechanism) that depends only on a declared 'influence context'. Agents supply policies that feed into these local mechanisms; a policy profile parametrizes the overall dynamics. Most importantly, we treat *interventions* as first-class *mechanism changes*: an intervention rewrites some local rules (policy changes are just a special case). The logic features a single dynamic operator to represent the effect of such an edit and the behaviours reachable afterwards (Section 4).

A system may be carved into two regions separated by a named interface that makes dependencies explicit: components on the left may depend on the interface, and components on the right may depend on the interface, but neither side reaches through the interface to the other. This 'boundary' lets us reason locally: a statement about the left side can be checked using only the left-side model (plus the shared interface), and likewise for the right. If we make a mechanism change that is confined to the right side and does not alter the interface, all facts established about the left side at the current configuration continue to hold after the update. The interface functions as a locus of manipulability and control, and provides a means of stable, targeted change emphasized by interventionist accounts of causation [20,38]. We use an *interface-indexed separating conjunction* operator: a statement of the form 'φ on the left and ψ on the right'

asserts that φ holds in the model restricted to the left-hand components and ψ holds in the model restricted to the right-hand components, with the two sides sharing only the declared interface. This construction mirrors the locality discipline of logic of bunched implications which enables compositional reasoning about independent parts [32].

We define actual causation (in the sense of Halpern and Pearl [20,21]) directly in our logical language, and by an 'unfolding' construction show that in any finite run of our system models, this notion conforms with Halpern-Pearl actual causes in an acyclic structural model built from that run, with mechanism changes represented as updates to the model itself. The notion of an interface also echoes Causal Influence Diagrams [15,23]: changing a policy corresponds to replacing a decision node's policy. Unlike do-calculus [33] with fixed mechanisms, we reason over a set of admissible mechanism changes: if a change is confined to one side of a declared interface, facts established about the other side still hold. Agent policies fit into this framework as first-class interventions, and coordinated policy changes by multiple agents are just sets of such edits. Practically, this lets causality practitioners pose the design-time questions such as 'is there an admissible mechanism change after which the desired property holds on all continuations?', and obtain a modular safety guarantee.

In Section 1, we motivate the design goals and relate them to engineering practice and needs. Section 2 discusses the behaviour-centric system model, interfaces, interventions, and system decompositions. Section 3 introduces agents, and policy profiles, and shows how policy changes are represented as interventions. Section 4 presents $\mathcal{L}(\langle\theta\rangle,*_\Lambda)$ and its semantics on restricted models. Section 5 defines actual causation and establishes alignment with HP-framework via a time-unfolding construction. Section 6 instantiates the framework on representative systems (agentic RAG-LLM workflows). Section 7 develops the intervention-preserving bisimulation and establishes soundness and completeness. Section 8 reflects on philosophical motivations, summarizes limitations, and sketches quantitative extensions.

2 The system modelling framework

In this section, we adopt a deliberately minimalist, behaviour-centric view of systems: instead of enumerating internal state, a component is specified by the *behaviours* an external observer can witness and by how those behaviours *influence* other components. This draws a line between intensional state (irrelevant here) and extensional behaviour (observable facts). This point of view, introduced in [17], represents a more abstract view of models of distributed systems than that based on process calculus and process logic as introduced in, for example, [3,9,10], building on a body of earlier work cited therein. Also refer to [13] for a related perspective, and to [36] for a historical background.

This section is devoted to the base behavioural model of system evolution; in Section 3 we discuss agents and policies. Formally, let $\mathcal{C}$ be the set of components and $\mathcal{B}$ the set of all behaviours. A mapping $\mathbb{B} : \mathcal{C} \to 2^{\mathcal{B}}$ assigns to each $c \in \mathcal{C}$

its allowable behaviours $\mathbb{B}(c)$. A *configuration* specifies the current behaviour of every component (cf. [13] for a similar theme).

Definition 1 (Configuration). *A* configuration *over $\mathcal{C}$ is a total function $f : \mathcal{C} \to \mathcal{B}$ with $f(c) \in \mathbb{B}(c)$ for all $c \in \mathcal{C}$. The set of all configurations is $F_{\mathcal{C}}$; when $\mathcal{C}$ is clear we write F.* □

To model evolution we use *influence mechanisms*: for each component, a function that, given its current behaviour and the behaviours of selected others, returns its next behaviour.

Definition 2 (Influence mechanisms and contexts). *For each $c \in \mathcal{C}$, the* influence context $\mathsf{Inf}(c) \subseteq \mathcal{C} \setminus \{c\}$ *lists those components whose behaviours are relevant for updating c. An* influence mechanism *for c is a function $\mathcal{I}_c : \mathbb{B}(c) \times \prod_{d \in \mathsf{Inf}(c)} \mathbb{B}(d) \to \mathbb{B}(c)$. The set $\mathcal{I} = \{\mathcal{I}_c\}_{c \in \mathcal{C}}$ denotes all such mechanisms.* □

Definition 3 (Transition relation). *Given $(\mathcal{C}, \mathbb{B}, \mathcal{I})$, the* transition relation *$\Delta_{\mathcal{I}} \subseteq F \times F$ contains (f, f') iff there exists exactly one $c \in \mathcal{C}$ such that $f'(c) = \mathcal{I}_c(f(c), (f(d))_{d \in \mathsf{Inf}(c)})$ and for all, $d \neq c : \ f'(d) = f(d)$. Thus, each step updates precisely one component according to its corresponding mechanism.* □

Definition 4 (System model). *A* system model *is $\mathcal{M} = (\mathcal{C}, \mathcal{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma)$, where F and $\Delta_{\mathcal{I}}$ are as above, and $\Gamma : \mathcal{P} \to 2^F$ is a valuation assigning to each atomic proposition the set of configurations where it holds. For brevity we often write $\mathcal{M} = (F, \Delta_{\mathcal{I}}, \Gamma)$.* □

Example 1. Let $\mathcal{C} = \{c_1, c_2, c_3\}$ with $\mathbb{B}(c_1) = \{b_{11}, b_{12}, b_{13}\}$, $\mathbb{B}(c_2) = \{b_{21}, b_{22}\}$, $\mathbb{B}(c_3) = \{b_{31}\}$. Take $\mathsf{Inf}(c_1) = \varnothing$, $\mathsf{Inf}(c_2) = \{c_1\}$, $\mathsf{Inf}(c_3) = \varnothing$, and mechanisms $\mathcal{I}_{c_1}(b_{11}) = b_{12}$, $\mathcal{I}_{c_1}(b_{12}) = b_{13}$, $\mathcal{I}_{c_1}(b_{13}) = b_{11}$, $\mathcal{I}_{c_2}(b_{21}, b_{12}) = b_{22}$, $\mathcal{I}_{c_2}(b_{21}, \_) = b_{21}$, $\mathcal{I}_{c_2}(b_{22}, \_) = b_{22}$, $\mathcal{I}_{c_3}(b_{31}) = b_{31}$. Let f_1 be a configuration such that $f_1(c_1) = b_{11}, f_1(c_2) = b_{21}, f_1(c_3) = b_{31}$. Then updating c_1 yields f_2 with $f_2(c_1) = b_{12}$ and $f_2(c_2) = b_{21}$, $f_2(c_3) = b_{31}$, so $(f_1, f_2) \in \Delta_{\mathcal{I}}$. □

To analyse subsystems, we use partial configurations and an *interface* that mediates dependencies across a decomposition.

Definition 5 (Partial configuration). *If $\mathcal{C}' \subseteq \mathcal{C}$, a* partial configuration *over $\mathcal{C}'$ is a function $f' : \mathcal{C}' \to \bigcup_{c \in \mathcal{C}'} \mathbb{B}(c)$ with $f'(c) \in \mathbb{B}(c)$. The restriction of $f \in F$ to $\mathcal{C}'$ is $f{\restriction}_{\mathcal{C}}$.* □

Definition 6 (Admissible interface-cut). *Let $\mathcal{M}$ have component set $\mathcal{C}$ and influence map $\mathsf{Inf}(\cdot)$. A triple (C_1, I, C_2) with $C_1 \cup C_2 = \mathcal{C}$ and $I = C_1 \cap C_2$ is an* admissible interface-cut *iff the following conditions are satisfied:*

1. *(**Left locality**) For all $c \in C_1 \setminus I$, $\mathsf{Inf}(c) \subseteq C_1 \cup I$.*
2. *(**Right locality**) For all $c \in C_2 \setminus I$, $\mathsf{Inf}(c) \subseteq C_2 \cup I$.*
3. *(**Interface closure**) For all $c \in I$, $\mathsf{Inf}(c) \subseteq I$.*

A model is interface-admitting *if it admits at least one interface-cut.* □

Remark 1. For each $c \in C_i$, one may view $\mathcal{I}_c$ as a local mechanism $\mathcal{I}_c^i$ over $C_i \cup I$ (the two coincide on shared inputs). □

Example 2 (Interface). With the contexts and mechanisms from Example 1, take $C_1 = \{c_1, c_2\}$, $C_2 = \{c_1, c_3\}$, and $I = C_1 \cap C_2 = \{c_1\}$. Then the three interface conditions hold, so $\{c_1\}$ is an interface. □

Definition 7 (Restriction along a cut). *Fix an admissible interface-cut $\lambda = (C_1, I, C_2)$ in a system model $\mathcal{M}$. For $j \in \{1, 2\}$, the* C_j-restriction $\mathcal{M}\restriction_{C_j}$ *is the system model obtained by: keeping only components in C_j; and, restricting behaviours and configurations to C_j; and, inducing a transition relation on C_j from $\Delta_{\mathcal{I}}$; that is, $f_j \, \Delta^{\lambda}_{C_j} \, f'_j$ iff there exist f, f' in $\mathcal{M}$ with $f \Delta_{\mathcal{I}} f'$, $f\restriction_{C_j} = f_j$, $f'\restriction_{C_j} = f'_j$, and $f'\restriction_I = f\restriction_I$. By locality and interface-closure in Definition. 6, $\mathcal{M}\restriction_{C_j}$ is well-defined. Moreover, if $f \Delta_{\mathcal{I}} f'$ in $\mathcal{M}$, then for $j = 1, 2$, $f\restriction_{C_j} \, \Delta^{\lambda}_{C_j} f'\restriction_{C_j}$ and $f'\restriction_I \, f' = f\restriction_I$.* □

Interventions replace (some of) the right-hand sides of the influence mechanisms while preserving the input contexts $\mathsf{Inf}(c)$.

Definition 8 (Intervention payload). *Given a model $\mathcal{M}$, for a component c, let* $\mathrm{Inp}(c) \coloneqq \prod_{d \in \mathsf{Inf}(c)} \mathbb{B}(d)$. *An* intervention payload *is a pair $(T, \mathcal{J}_T)$ where $T \subseteq \mathcal{C}$ and $\mathcal{J}_T = \{\mathcal{J}_c : \mathbb{B}(c) \times \mathrm{Inp}(c) \to \mathbb{B}(c)_{c \in T}$ provides replacement mechanisms for exactly the targets in T. Applying $(T, \mathcal{J}_T)$ to an augmented model $\mathcal{M}$ yields $\mathcal{M}[T \mapsto \mathcal{J}_T]$, which keeps $\mathcal{C}$, $\mathbb{B}$, $\mathsf{Inf}(\cdot)$, F, and Γ unchanged; and replaces $\mathcal{I}_c$ by $\mathcal{J}_c$ for $c \in T$, leaving $\mathcal{I}_c$ unchanged for $c \notin T$, with the transition relation recomputed from the modified family of mechanisms. If θ denotes an intervention, we write $\mathcal{M}_\theta = \mathcal{M}[T \mapsto \mathcal{J}_T]$.* □

Example 3. In Example 1, let $\theta = (\{c_1\}, \{\mathcal{I}'_{c_1}\})$ be an such that $\mathcal{I}'_{c_1}(b_{11}) = \mathcal{I}'_{c_1}(b_{12}) = \mathcal{I}'_{c_1}(b_{13}) = b_{11}$. In the intervened model $\mathcal{M}_\theta$, every reachable configuration f satisfies $f(c_1) = b_{11}$; consequently c_2 remains at b_{21}. □

Remark 2. If $\mathcal{M}$ is interface-admitting and θ is *cut-preserving* i.e., it leaves all influence contexts unchanged and does not rewrite the mechanisms of any interface component of any admissible interface-cut, then every interface-cut (C_1, I, C_2) admissible in $\mathcal{M}$ remains admissible in $\mathcal{M}_\theta$; in particular, $\mathcal{M}_\theta$ admits the same (C_1, I, C_2). □

Definition 9 (Augmented model). *An* augmented model *is a tuple $\widehat{\mathcal{M}} = (\mathcal{C}, \mathbb{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma, \ \mathsf{cut}, \ [\![\cdot]\!])$, where the first six components form a system model, $\mathsf{cut} : \Lambda \to \{(C_1, I, C_2)\}$ assigns admissible interface-cuts (Definition 6), and $[\![\cdot]\!]$ interprets each label θ by installing the corresponding intervention payload (Definition 8) and updating the induced transition relation.* □

3 Agents and agency

Many systems are steered by decision makers whose choices are best viewed as *policy changes* to parts of the mechanism. Our goal is to reason about questions

such as: what if an actor switches policy, which edits guarantee a safety property, and how should responsibility be attributed? We keep agency lightweight and compatible with the existing intervention modality, rather than introducing a separate strategic calculus (see [1] for an alternative approach).

Agents and policies: Let $\widehat{\mathcal{M}} = (\mathcal{C}, \mathbb{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma, \mathsf{cut}, [\![\cdot]\!])$ be an augmented model, and $\mathcal{A}$ be a finite set of agents. For each component $c \in \mathcal{C}$, write $A_c \subseteq \mathcal{A}$ for the agents that are permitted to change c's mechanism. A *policy choice* by an agent can be viewed as, from a set of admissible replacements, selecting new mechanisms for $c \in A_c$. Concretely, for a coalition $A \subseteq \mathcal{A}$, it is an intervention payload $(T, \mathcal{J}_T)$ with $T \subseteq \bigcup_{a \in A} \{ c \in \mathcal{C} \mid a \in A_c \}$, and, $\mathcal{J}_T = \{\mathcal{J}_c : \mathbb{B}(c) \times \mathrm{Inp}(c) \to \mathbb{B}(c)\}_{c \in T}$. For each c, let $\mathrm{Adm}(c)$ be the set of admissible local mechanisms $\mathcal{J}_c : \mathbb{B}(c) \times \prod_{d \in \mathsf{Inf}(c)} \mathbb{B}(d) \to \mathbb{B}(c)$, required to respect locality (depend only on $\mathsf{Inf}(c)$). We require that these mechanisms respects the fixed influence boundary: $\mathrm{Inp}(c) = \prod_{d \in \mathsf{Inf}(c)} \mathbb{B}(d)$ and $\mathcal{J}_c$ depends only on $\mathsf{Inf}(c)$.

Definition 10 (Policy profile). *A* policy profile *is a mapping $\Pi = \{\Pi_a \mid a \in \mathcal{A}\}$ such that each Π_a assigns, to every $c \in a_c$, a mechanism $\Pi_a(c) \in \mathrm{Adm}(c)$. The mechanism in force at c under Π is $\mathcal{I}_c^{\Pi} = \Pi_a(c)$ if there is a unique a with $c \in a_c$; and otherwise, $\mathcal{I}_c^{\Pi} = \mathcal{I}_c$.* □

Definition 11. *Given a coalition $A \subseteq \mathcal{A}$ and a finite target set $T \subseteq \bigcup_{a \in A} a_c$ with replacement mechanisms $\mathcal{J}_T = \{\mathcal{J}_c \in \mathrm{Adm}(c) \mid c \in T\}$, the* policy intervention *$\theta_{A,\mathcal{J}_T}$ has interpretation $[\![\theta_{A,\mathcal{J}_T}]\!](\widehat{\mathcal{M}}, \Pi) = (\widehat{\mathcal{M}}[T \mapsto \mathcal{J}_T], \Pi_\theta)$. where $\widehat{\mathcal{M}}[T \mapsto \mathcal{J}_T]$ rewrites the local mechanisms for $c \in T$ to $\mathcal{J}_c$, and the* updated policy profile *Π_θ is obtained by overriding the sub-profile of A so that for each $a \in A$ and $c \in a_c$, $\Pi_\theta(a)(c) = \mathcal{J}_c$ if $c \in T$; and otherwise, $\Pi_\theta(a)(c) = \Pi_a(c)$ For non-policy (pure) interventions we set $\Pi_\theta \coloneqq \Pi$.* □

Definition 12. *Fix an augmented model $\widehat{\mathcal{M}} = (\mathcal{C}, \mathbb{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma, \mathsf{cut}, [\![\cdot]\!])$ and a policy profile Π. Let $\mathcal{I}^{\Pi} = \{\mathcal{I}_c^{\Pi}\}_{c \in \mathcal{C}}$ be the family of local mechanisms* in force *under Π (Definition 10). The transition relation* induced by *Π is $\Delta_{\mathcal{I}}^{\Pi} \subseteq F \times F$ defined by $f \, \Delta_{\mathcal{I}}^{\Pi} \, f'$ iff there exists $c \in \mathcal{C}$ such that $f'(c) = \mathcal{I}_c^{\Pi}(f(c), (f(d))_{d \in \mathsf{Inf}(c)})$. Here, for all $d \neq c : f'(d) = f(d)$. If an intervention θ has interpretation $[\![\theta]\!](\widehat{\mathcal{M}}, \Pi) = (\widehat{\mathcal{M}}_\theta, \Pi_\theta)$ with updated mechanisms $\mathcal{I}^\theta$ in $\widehat{\mathcal{M}}_\theta$ (Definition 11), then the post-intervention transition steps are taken with respect to $\Delta_{\mathcal{I}^\theta}^{\Pi_\theta}$.* □

4 Syntax and semantics

4.1 Syntax

Let $\mathcal{P}$ be a denumerable set of atomic propositions. Fix two denumerable sets of *labels*, Θ (symbols naming admissible interventions), ranged over by θ, and Λ (symbols naming admissible interface-cuts), ranged over by λ. The language $\mathcal{L}(\langle\theta\rangle, *_\Lambda)$ is generated by:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \langle\theta\rangle\varphi \mid \varphi *_\lambda \psi,$$

where $p \in \mathcal{P}$, $\theta \in \Theta$, and $\lambda \in \Lambda$. As usual, $\varphi \vee \psi$, $\varphi \to \psi$, and $\top, \bot$ are defined classically. We write $\Diamond^+\varphi$ (resp. $\Box^+\varphi$) as shorthand for the existential (resp. universal) modality over the transitive closure of the transition relation (see semantics).

4.2 Semantics

We work with augmented models (Definition. 9) $\widehat{\mathcal{M}} = (\mathcal{C}, \mathbb{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma, \mathsf{cut}, [\![\cdot]\!])$, where $\mathsf{cut} : \Lambda \to \{(C_1, I, C_2) \mid C_1 \cup C_2 = \mathcal{C},\ I = C_1 \cap C_2\}$ selects admissible interface-cuts, and $[\![\cdot]\!]$ interprets each intervention label $\theta \in \Theta$ as a mechanism-change operator with $\widehat{\mathcal{M}}_\theta = (\mathcal{C}, \mathbb{B}, \mathcal{I}^\theta, F, \Delta^\theta_{\mathcal{I}}, \Gamma, \mathsf{cut}, [\![\cdot]\!])$, leaving F,Γ, cut, and $[\![\cdot]\!]$ unchanged; but inducing $\Delta^\theta_{\mathcal{I}}$ from the modified influence mechanisms $\mathcal{I}^\theta$ (Definition 8).

Definition 13. *Satisfaction $(\widehat{\mathcal{M}}, f, \Pi) \models \varphi$ is defined by*

$$
\begin{array}{rll}
(\widehat{\mathcal{M}}, f, \Pi) \models p & \textit{iff} & f \in \Gamma(p) \\
(\widehat{\mathcal{M}}, f, \Pi) \models \neg\varphi & \textit{iff} & (\widehat{\mathcal{M}}, f, \Pi) \not\models \varphi \\
(\widehat{\mathcal{M}}, f, \Pi) \models \varphi \wedge \psi & \textit{iff} & (\widehat{\mathcal{M}}, f, \Pi) \models \varphi \textit{ and } (\widehat{\mathcal{M}}, f, \Pi) \models \psi \\
(\widehat{\mathcal{M}}, f, \Pi) \models \Diamond\varphi & \textit{iff} & \textit{for some } f' \in F \textit{ s.t. } (f\ \Delta^\Pi_{\mathcal{I}}\ f' \textit{ and } (\widehat{\mathcal{M}}, f', \Pi) \models \varphi) \\
(\widehat{\mathcal{M}}, f, \Pi) \models \Box\varphi & \textit{iff} & \textit{for all } f' \in F \textit{ s.t. } (f\ \Delta^\Pi_{\mathcal{I}}\ f' \textit{ implies } (\widehat{\mathcal{M}}, f', \Pi) \models \varphi) \\
(\widehat{\mathcal{M}}, f, \Pi) \models \langle\theta\rangle\varphi & \textit{iff} & \textit{for some } f' \in F \textit{ s.t. } (f\ (\Delta^{\Pi_\theta}_{\mathcal{I}_\theta})^\star\ f' \textit{ and } (\widehat{\mathcal{M}}_\theta, f', \Pi_\theta) \models \varphi) \\
(\widehat{\mathcal{M}}, f, \Pi) \models \varphi *_\lambda \psi & \textit{iff} & (C_1, I, C_2) \textit{ is interface-admitting in } \widehat{\mathcal{M}} \textit{ and} \\
 & & (\widehat{\mathcal{M}}{\restriction_{C_1}}, f{\restriction_{C_1}}, \Pi) \models \varphi \textit{ and } (\widehat{\mathcal{M}}{\restriction_{C_2}}, f{\restriction_{C_2}}, \Pi) \models \psi
\end{array}
$$

Here $(\Delta^\theta_{\mathcal{I}})^*$ is the reflexive and transitive closure of $\Delta^\theta_{\mathcal{I}}$ (zero or more steps). We also use the shorthands $\Diamond^+\varphi$ (resp. $\Box^+\varphi$) for existential (resp. universal) modalities over the transitive *closure $\Delta^+_{\mathcal{I}}$.* □

For $C' \subseteq \mathcal{C}$, we write $\widehat{\mathcal{M}}{\restriction_{C'}} = (C', \mathbb{B}{\restriction_{C'}}, \mathcal{I}{\restriction_{C'}}, F_{C'}, \Delta^{C'}, \Gamma{\restriction_{C'}}, \mathsf{cut}{\restriction_{C'}}, [\![\cdot]\!])$ where $F_{C'}$ is the set of partial configurations $g : C' \to \bigcup_{c \in C'} \mathbb{B}(c)$ with $g(c) \in \mathbb{B}(c)$; and, $\Gamma{\restriction_{C'}}(p) = \{g \in F_{C'} \mid \text{for some } f \in \Gamma(p) \text{ with } f{\restriction_{C'}} = g\}$ (valuation restricted to C'); and, $\Delta^{C'} \subseteq F_{C'} \times F_{C'}$ is given by $(g, g') \in \Delta^{C'}$ iff, for some $f, f' \in F$, $f{\restriction_{C'}} = g$, $f'{\restriction_{C'}} = g'$, $(f, f') \in \Delta_{\mathcal{I}}$.

When $\lambda = (C_1, I, C_2)$ is admissible, the restricted relation $\Delta^\lambda_{C_i}$ coincides with the transition obtained by executing only the local mechanisms on C_i while keeping the interface I fixed; this follows from left/right locality and interface-closure (Definition 6), and therefore satisfaction in $\widehat{\mathcal{M}}{\restriction_{C_i}}$ is well-defined. Under agent policy interventions (Section 3), replacement influence mechanisms for any c depend only on $\mathsf{Inf}(c)$ and no mechanism for components in I is altered.

5 Actual causation system models

Halpern-Pearl structural causal models (SCMs) [20,33] represent a scenario by endogenous variables V, an exogenous context U, and structural equations F

that determine each V_i from its parents. Interventions are value-setting changes $\mathrm{do}(X = x)$ that replace the equation for X and induce counterfactual worlds. HP define *actual cause* of an effect ψ_E via three clauses: (AC1) *actuality* ($X = x$ and ψ_E both hold), (AC2) *counterfactual dependence under a contingency* (there exists W such that $[X \leftarrow x', W \leftarrow w]\ \neg\psi_E$), and (AC3) *minimality*. SCMs excel at post-hoc analysis, but design-time reasoning about *which* mechanism changes are admissible and *where* they apply often forces one to move across families of models or to introduce ad-hoc switch variables.

Philosophically, our choice to treat *mechanism change* as the core intervention aligns with manipulability accounts of causation: causes are what can be purposefully changed to control effects. In engineered, multiagent settings the natural unit of manipulation is rarely a variable-value assignment but a *rule edit*. Making such edits first-class situates the logic at the level of design and control, not merely post-hoc explanation. Viewing agent's policies as interventions matches the agency-as-capacity-to-intervene view: agents have an admissible policy set over its observations. A *choice* by the agent is represented by an intervention whose interpretation updates exactly the mechanisms for those components it can control. This matches interventionist accounts of causation, where control is analysed via counterfactual changes one can *bring about* [38], and it is conformant with mechanistic views that explain phenomena by operations of organized parts and their modifiable activities [30,36].

Another closely related formalism is that of Causal Influence Diagrams which extend classical influence diagrams with a causal interpretation of arcs, combining chance, decision, and utility nodes within a single directed acyclic graph and supporting counterfactual reasoning via Pearl's do-calculus [23,33,35]. Recent work extends CIDs to agent modelling, sequential decision-making, and safety analysis in multi-agent systems [15,27,22]. In contrast, our framework treats both *mechanistic* and *policy-induced* changes uniformly through a single intervention modality $\langle\theta\rangle\varphi$, interpreted directly in the transition semantics.

We define an actual cause as follows:

Definition 14 (Actual cause). *Let $f_0, f_1 \in F$ be configurations, and let the* effect *be a formula ψ_E whose atoms lie in a designated set $C_E \subseteq \mathcal{C}$. A non-empty set of components $C^\star \subseteq \mathcal{C}$ is an* actual cause of $(\widehat{\mathcal{M}}, f_1, \Pi) \models \psi_E$ from f_0 *(written* $\mathrm{Cause}_\Pi(f_0, f_1; C^\star, \psi_E)$*) iff the following hold:*

(AC1) Actuality: $(\widehat{\mathcal{M}}, f_0, \Pi) \models \Diamond^+(\psi_E \ \wedge\ \chi_{C^\star}(f_0))$ *and* $f_0(c) = f_1(c)$*, for all* $c \in C^\star$

(AC2^m) Counterfactual dependence with witness: *There exists a witness set $W \subseteq \mathcal{C}$, an admissible interface-cut $\lambda = (C_1, I, C_2)$,[3] and an intervention label $\theta \in \Theta$ such that $(\widehat{\mathcal{M}}, f_0, \Pi) \models \langle\theta\rangle\,(\,\chi_W(f_0) \ast_\lambda \delta_{C^\star \setminus W}(f_0)\,)$ implies $(\widehat{\mathcal{M}}, f_0, \Pi) \models \langle\theta\rangle\ \Box^+\neg(\psi_E \ \wedge\ \chi_{C^\star}(f_0))$. Intuition: holding the witness W fixed as in f_0 while altering at least one coordinate in $C^\star \setminus W$ (via some admissible θ) prevents the effect from arising along all post-intervention continuations.*

[3] For example, take $W \subseteq C_1$ and $(C^\star \setminus W) \subseteq C_2$ so that $\chi_W(f_0)$ and $\delta_{C^\star \setminus W}(f_0)$ can be conjoined via $\ast_\lambda$.

(AC3) Minimality *No proper subset $C' \subsetneq C^\star$ satisfies* ***AC1*** *and* $\mathbf{AC2}^m$ *with the same f_0, f_1 and some (possibly different) W, λ, θ.* □

5.1 HP-alignment

Our notions (Definition 14) are aligned with the Halpern-Pearl framework in the following sense: for any finite evolution, a *time-unfolded* structural causal model (SCM) whose variables are the components in our set-up indexed by discrete steps can be built. This yields an acyclic SCM suitable for defining standard HP actual causation.

We can associate a 'time-unfolding' of an SCM to a finite run of the system models. Fix a policy profile Π, a finite sequence of configurations $(f_0, \ldots, f_n)$ with $f_i \, (\Delta_{\mathcal{I}}^{\Pi})^+ \, f_{i+1}$, and a (deterministic) choice of an updating component $s_i \in \mathcal{C}$ for each unit step along the run (so that s_i is the unique component updated at the i-th micro-step). Define the SCM $M_{\mathrm{unf}}(n, \Pi) = \langle U,\ V,\ F \rangle$ as follows:

1. Endogenous variables $V = \{\, V_c^i \mid c \in \mathcal{C},\ i = 0, \ldots, n \,\}$, with $\mathrm{dom}(V_c^i) = \mathbb{B}(c)$.
2. Exogenous variables $U = \{\, U_0,\ S^1, \ldots, S^n \,\}$, where U_0 fixes the initial configuration and S^i is a scheduler selecting s_i.
3. Structural functions F:

$$V_c^0 = f_0(c) \qquad \text{(determined by } U_0\text{)},$$
$$V_c^i = \begin{cases} \mathcal{I}_c^{\Pi}(V_c^{i-1},\ (V_d^{i-1})_{d \in \mathsf{Inf}(c)}) & \text{if } S^i = c, \\ V_c^{i-1} & \text{if } S^i \neq c, \end{cases} \qquad i = 1, \ldots, n.$$

This graph is acyclic (edges point from time $i{-}1$ to i) and has a unique solution for any context $\vec{u} = (U_0 = f_0, S^i = s_i)$, namely $V_c^i = f_i(c)$.

Interventions can be viewed as change in model: a mechanism or policy intervention $\theta \in \Theta$ that (possibly) rewrites the influence mechanisms for a subset $S_\theta \subseteq \mathcal{C}$ induces a *modified* unfolded SCM $M_{\mathrm{unf}}^{\theta}(n, \Pi)$ by replacing, for all $c \in S_\theta$ and all $i \geq 1$, the right-hand side $\mathcal{I}_c^{\Pi}(\cdots)$ with the post-intervention mechanism $\mathcal{I}_c^{\Pi_\theta}(\cdots)$; all other equations remain unchanged. This mirrors our semantics where $(\widehat{\mathcal{M}}, \Pi)$ is mapped to $(\widehat{\mathcal{M}}_\theta, \Pi_\theta)$ before taking (reflexive-transitive) steps.

Theorem 1 (HP-alignment). *Let $(f_0, \ldots, f_n)$ be a causal chain in $(\widehat{\mathcal{M}}, \Pi)$ with outcome f_n and let ψ_E be an effect formula whose atoms lie in $C_E \subseteq \mathcal{C}$. Suppose $C^\star \subseteq \mathcal{C}$ is an actual cause of ψ_E from f_0 to f_n in the sense of Definition 14, witnessed by (W, λ, θ) for* $\mathbf{AC2}^m$*. Define the unfolded SCM $M_{\mathrm{unf}}(n, \Pi)$ and the effect formula*

$$\varphi_E^n \coloneqq \bigwedge_{c \in C_E} (V_c^n = f_n(c))$$

Let $X \coloneqq \{V_c^0 \mid c \in C^\star\}$ and $x \coloneqq (f_0(c))_{c \in C^\star}$. Then there exists a context $\vec{u}$ (namely $U_0 = f_0$ and the schedule realizing the chain) such that, in the HP sense:

1. ***AC1.*** $(M_{\mathrm{unf}}(n, \Pi), \vec{u}) \models (X{=}x) \wedge \varphi_E^n$.

2. ***AC2.*** *There is a (possibly empty) set $W' \subseteq V$ (encoding the witness W initially) such that, in the* modified *model $M^{\theta}_{\mathrm{unf}}(n, \Pi)$,*

$$(M^{\theta}_{\mathrm{unf}}(n, \Pi), \vec{u}) \models \big[X \leftarrow x',\ W' \leftarrow w^*\big]\neg\varphi^n_E$$

for some x' differing from x on at least one coordinate (corresponding to $\delta_{C^\star \setminus W}(f_0)$) while w^* fixes W' to their values in $\vec{u}$.

3. ***AC3.*** *X is minimal for* AC1–AC2 *above.*

Hence $X{=}x$ is an HP actual cause of φ^n_E in the unfolded model.

Proof (Proof sketch). AC1: By construction of $M_{\mathrm{unf}}(n, \Pi)$, the unique solution under $\vec{u}$ is $V^i_c = f_i(c)$, hence φ^n_E holds and $X{=}x$ holds.

AC2: Definition 14**(AC2^m)** asserts that there is an interface-cut λ, a witness set W, and an intervention θ such that keeping W fixed as in f_0 while changing at least one coordinate in $C^\star \setminus W$ (effected via θ) forces $\neg(\psi_E \wedge \chi_{C^\star}(f_0))$ along *all* post-intervention continuations. In the unfolded SCM, we $W' \coloneqq \{\, V^0_c \mid c \in W\}$ to their f_0-values, fix X to some $x' \neq x$ and taking the *modified* model $M^{\theta}_{\mathrm{unf}}(n, \Pi)$. The conclusion is $\neg\varphi^n_E$ under $[X \leftarrow x', W' \leftarrow w^*]$ in $M^{\theta}_{\mathrm{unf}}(n, \Pi)$.

AC3: Minimality of $C^\star$ in Definition 14 transfers to minimality of X because X is just the time-0 copy of $C^\star$. If a proper $X' \subset X$ sufficed in the SCM, its projection onto components would contradict **AC3** for $C^\star$. □

Remark 3. The use of $M^{\theta}_{\mathrm{unf}}(n, \Pi)$ in **AC2** corresponds to HP's contingency on a *modified model* when mechanisms are changed. For policy-labelled θ, the modification is exactly the replacement $\mathcal{I}^{\Pi} \to \mathcal{I}^{\Pi_\theta}$; for mechanistic θ, it rewrites the affected $\mathcal{I}_c$ in the equations. Both are admitted in our semantics and align with the 'model change + counterfactual' reading in HP. □

Corollary 1. *If, for a given policy profile Π, the* one-step *dependency graph $c \to \mathsf{Inf}(c)$ is acyclic and each $\mathcal{I}^{\Pi}_c$ is total, then there exists a (non-time-indexed) SCM with endogenous variables $\{V_c\}_{c\in\mathcal{C}}$ and equations $V_c = \mathcal{I}^{\Pi}_c(V_c, (V_d)_{d\in\mathsf{Inf}(c)})$ that matches the unique fixed point reached from f_0 (under any schedule). In this case, Theorem 1 holds with $X = \{V_c \mid c \in C^\star\}$ and $\varphi_E = \bigwedge_{c\in C_E} p_{c=f_n(c)}$.* □

6 Application: LLM-RAG workflows

Retrieval-augmented generation (RAG) couples a *retriever* that, given a query q, returns a small set of evidence items, with a *generator*, a Large Language Model, (LLM) that produces the final text conditioned on a context assembled from those items [18,29]. The architectural specifics (whether retrieval, re-ranking, or tools are neural, neuro-symbolic, or heuristic) do not matter here: our framework is implementation-agnostic. Operationally, retrieval and re-ranking produce a *context* that is handed off to the generator. The generator reads this context and emits a draft which is then filtered by a guard module. Tools may be invoked and their effects folded back into the context. Policies over these modules are

supplied by *agents* (for example, an orchestrator for retrieval, re-ranking, and tool calls). This picture aligns with DSPy, a declarative framework that compiles structured modules into effective prompts and weights across base models, inference schemes, and learning algorithms [26].

In this view, LLM workflows are *text transformation graphs*: imperative computational graphs where LLMs are invoked through declarative, parametrized modules [26]. Our formal treatment complements this systematic development by giving a semantics for mechanism and policy edits, and connects these to Halpern-Pearl style causal claims. There is longstanding Information Retrieval work using counterfactual reasoning for evaluation and learning-to-rank [25], and recent strands inject explicit causal structure into retrieval and conditioning [37].

In parallel, a growing line connects formal verification with Halpern-Pearl actual causation to explain why properties hold or fail and to attribute responsibility within system models [6]. We lift this perspective to modular RAG workflows by making *interfaces* and *mechanism edits* first-class. This aligns with emerging assurance guidance: the NIST AI Risk Management Framework emphasizes lifecycle validation across context and outputs [31]; the UK Financial Conduct Authority advocates technology-agnostic oversight oriented to firms' systems and processes [16]. Cloud-operations guidance such as AWS's MLOps white-paper treats reliability and governance as end-to-end workflow properties with auditable, reproducible edits and controls [2].

In particular, we model a (stylized) RAG software-workflow (with tools and guards) as an augmented system $\widehat{\mathcal{M}} = (\mathcal{C}, \mathcal{B}, \mathcal{I}, F, \Delta_{\mathcal{I}}, \Gamma, \mathsf{cut}, [\![\cdot]\!])$ with agents as in Section 3. The goal is to reason modularly about edits to mechanisms and policies while preserving invariants across named interfaces. In particular, inputs/outputs are not components in our framework; but are represented as *behaviours of interface components*. Thus the context passed from retrieval to generation is the behaviour of a dedicated component CtxOut, and decoder requests to tools are behaviours of Dec.

The component set is $\mathcal{C} = \{\mathsf{Tok}, \mathsf{Retr}, \mathsf{Rank}, \mathsf{CtxOut}, \mathsf{Dec}, \mathsf{Guard}, \mathsf{Tool}, \mathsf{Mem}, \mathsf{PortCtx}, \mathsf{PortDecReq}\}$:

1. Tok (*tokenizer*): parses the user input into tokens, and is the source of retrieval queries.
2. Retr (*retriever*): fetches candidate documents given Tok (and possibly Mem).
3. Rank (*ranker/re-ranker*): orders Retr's candidates.
4. CtxOut (*context builder*): composes the prompt(context) from Rank (and Mem) for the LLM instance.
5. Dec (*decoder*): produces model outputs conditioned on the interface context and tool state.
6. Guard (*guardrail*): vets drafts produced by Dec and either blocks or permits responses.
7. Tool (*tool executor*): It executes external tool calls (search, code, DataBase connector, etc.).

8. Mem (*memory cache*): It stores and retrieves auxiliary state (e.g., past interactions, embeddings) influencing Retr and observed by Dec.
9. PortCtx (*context port*): It holds the hand-off value written by a designated intervention (e.g., $\theta_{\mathrm{pushCtx}}$) summarizing CtxOut. It is influenced by no other component.
10. PortDecReq (*tool-request port*): This component is toggled by a designated intervention (e.g., $\theta_{\mathrm{callTool}}$) to signal tool invocation. It is influenced by no other component.

The Model Context Protocol (MCP) [4] is an open standard that lets instances of LLM agents connect in a standardized, two-way, client-server way to external tools and data sources enabling them to discover and to invoke external tools and pull context. In practice, CtxOut, PortCtx, and PortDecReq correspond directly to the hand-off points defined by MCP. CtxOut implements client-side assembly from MCP prompts, writing PortCtx is the commit of the constructed model input to the decoder, and toggling PortDecReq mirrors MCP tool invocations. We model each component with only finitely many behaviours, and dependencies are expressed through influence contexts:

$$\begin{aligned}
\mathbb{B}(\mathsf{Tok}) &= \{\mathsf{ok}, \mathsf{err}\} & \mathbb{B}(\mathsf{Retr}) &= \{\mathsf{none}, \mathsf{rel}, \mathsf{spur}\} \\
\mathbb{B}(\mathsf{Rank}) &= \{\mathsf{good}, \mathsf{bad}\} & \mathbb{B}(\mathsf{CtxOut}) &= \{\mathsf{short}, \mathsf{long}, \mathsf{noctx}\} \\
\mathbb{B}(\mathsf{Dec}) &= \{\mathsf{safe}, \mathsf{unsafe}, \mathsf{abort}\} & \mathbb{B}(\mathsf{Guard}) &= \{\mathsf{pass}, \mathsf{block}\} \\
\mathbb{B}(\mathsf{Tool}) &= \{\mathsf{off}, \mathsf{on}, \mathsf{fail}\} & \mathbb{B}(\mathsf{Mem}) &= \{\mathsf{hit}, \mathsf{miss}\} \\
\mathbb{B}(\mathsf{PortCtx}) &= \{\mathsf{short}, \mathsf{long}, \mathsf{noctx}\} & \mathbb{B}(\mathsf{PortDecReq}) &= \{\mathsf{off}, \mathsf{on}\}
\end{aligned}$$

$$\begin{aligned}
&\mathsf{Inf}(\mathsf{Tok}) = \varnothing && \mathsf{Inf}(\mathsf{Retr}) = \{\mathsf{Tok}, \mathsf{Mem}\} && \mathsf{Inf}(\mathsf{Rank}) = \{\mathsf{Retr}\} \\
&\mathsf{Inf}(\mathsf{CtxOut}) = \{\mathsf{Rank}, \mathsf{Mem}\} && \mathsf{Inf}(\mathsf{PortCtx}) = \varnothing && \mathsf{Inf}(\mathsf{Dec}) = \{\mathsf{PortCtx}\} \\
&\mathsf{Inf}(\mathsf{Guard}) = \{\mathsf{Dec}\} && \mathsf{Inf}(\mathsf{PortDecReq}) = \varnothing \\
&\mathsf{Inf}(\mathsf{Mem}) = \{\mathsf{Retr}\} && \mathsf{Inf}(\mathsf{Tool}) = \{\mathsf{PortDecReq}\}
\end{aligned}$$

We set $\mathsf{Inf}(\mathsf{Tok}) = \varnothing$ because Tok is the RAG workflow's exogenous source: it updates from user input, not from any modelled component. Let $\mathcal{A} = \{\mathsf{Orch}, \mathsf{Safety}\}$ be a set denoting agents: orchestrator and safety owner. A relation $\mathrm{Ctrl} \subseteq \mathcal{A} \times \mathcal{C}$ with $\mathrm{Ctrl}(\mathsf{Orch}) = \{\mathsf{Retr}, \mathsf{Rank}, \mathsf{CtxOut}, \mathsf{Dec}, \mathsf{Tool}\}$ and $\mathrm{Ctrl}(\mathsf{Safety}) = \{\mathsf{Guard}\}$ specifies which agent controls which components. A policy profile $\Pi = (\pi_{\mathsf{Orch}}, \pi_{\mathsf{Safety}})$ parametrizes the local rules (cf. Section 3). Thus each component c consumes only $(f(d))_{d \in \mathsf{Inf}(c)}$ (e.g., Dec sees only PortCtx; Tool sees only PortDecReq). We use interventions $\theta[\mathsf{PortCtx}]$ and $\theta[\mathsf{PortDecReq}]$ for port writes. We use two admissible interface cuts $(C_1^{\mathrm{retr}}, I^{\mathrm{retr}}, C_2^{\mathrm{gen}})$ and $(C_1^{\mathrm{tool}}, I^{\mathrm{tool}}, C_2^{\mathrm{rest}})$ (cf. Definition 6) with

$$\begin{aligned}
I^{\mathrm{tool}} &= \{\mathsf{PortDecReq}\} & C_1^{\mathrm{retr}} &= \{\mathsf{Tok}, \mathsf{Retr}, \mathsf{Rank}, \mathsf{Mem}, \mathsf{CtxOut}\} \\
I^{\mathrm{retr}} &= \{\mathsf{PortCtx}\} & C_2^{\mathrm{gen}} &= \{\mathsf{Dec}, \mathsf{Guard}, \mathsf{Tool}, \mathsf{PortDecReq}\} \\
C_1^{\mathrm{tool}} &= \{\mathsf{Tool}\} & C_2^{\mathrm{rest}} &= \{\mathsf{Tok}, \mathsf{Retr}, \mathsf{Rank}, \mathsf{Mem}, \mathsf{CtxOut}, \mathsf{PortCtx}, \mathsf{Dec}, \\
& & & \quad \mathsf{Guard}\}
\end{aligned}$$

Admissibility holds since $\mathsf{Inf}(c) \subseteq C_1^{\mathrm{retr}} \cup I^{\mathrm{retr}}$ for $c \in C_1^{\mathrm{retr}} \setminus I^{\mathrm{retr}}$, and similarly for C_2^{gen}. Similarly, $\mathsf{Inf}(c) \subseteq C_1^{\mathrm{tool}} \cup I^{\mathrm{tool}}$ for $c \in C_1^{\mathrm{tool}} \setminus I^{\mathrm{tool}}$, and so for C_2^{rest} Also, $\mathsf{Inf}(\mathsf{PortCtx}) = \varnothing \subseteq I^{\mathrm{retr}}$

Representative queries. Atomic propositions are of the form $p_{X=\sigma}$ ('X exhibits behaviour σ'). Let $\mathsf{bad} \equiv (p_{\mathsf{Dec=unsafe}} \wedge p_{\mathsf{Guard=pass}})$.

1. **Guarded recovery.** Assume the policy-labelled edit $\theta_{\mathsf{policySaf}}$ rewrites only Guard and enforces $\mathcal{I}_{\mathsf{Guard}}^{\theta_{\mathsf{policySaf}}}(f) = \mathsf{block}$ whenever $f(\mathsf{Dec}) = \mathsf{unsafe}$. Then $(\widehat{\mathcal{M}}, f, \Pi) \models \langle\theta_{\mathsf{policySaf}}\rangle\Box\neg\mathsf{bad}$ Moreover, if the schedule re-evaluates Guard on every step that can change Dec, then the guarantee strengthens to $(\widehat{\mathcal{M}}, f, \Pi) \models \langle\theta_{\mathsf{policySaf}}\rangle\Box^{+}\neg\mathsf{bad}$
2. Let $\lambda = \lambda_{\text{retr-gen}}$ with interface $\mathsf{PortCtx}$ and write φ_{C_1} for any C_1-local property over $\{\mathsf{Tok}, \mathsf{Retr}, \mathsf{Rank}, \mathsf{Mem}, \mathsf{CtxOut}\}$ and ψ_{C_2} for any C_2-local property over $\{\mathsf{Dec}, \mathsf{Guard}, \mathsf{Tool}, \mathsf{PortDecReq}\}$. If θ rewrites only Retr (hence only C_1) and preserves the cut, then $(\widehat{\mathcal{M}}, f, \Pi) \models \varphi_{C_1} *_\lambda \psi_{C_2}$ implies $(\widehat{\mathcal{M}}, f, \Pi) \models \langle\theta\rangle\,(\varphi'_{C_1} *_\lambda \Diamond^{\star}_{\theta}\psi_{C_2})$ for some C_1-local φ'_{C_1}. In particular, C_1-facts are preserved and C_2-facts can be re-established after applying θ.
3. **Actual-cause.** Let $f \rightsquigarrow f'$ denote a non-empty path (one or more steps) under the current policy profile Π. Let the effect be $\psi_E \coloneqq p_{\mathsf{Dec=unsafe}} \wedge p_{\mathsf{Guard=pass}}$ (decoder unsafe and guard passes at f'). We test $C^\star = \{\mathsf{Retr}, \mathsf{Rank}\}$ as the candidate cause-set and use $W = \{\mathsf{PortCtx}\}$ as the witness held fixed at the retrieval-generation interface $\lambda_{\text{retr-gen}}$. ψ_E holds at f' and the components in $C^\star$ are unchanged along $f \rightsquigarrow f'$. For an admissible intervention θ that alters some element of $C^\star \setminus W$ (e.g. $\theta_{\mathsf{top\text{-}k}(k)}$ or $\theta_{\mathsf{tok\text{-}swap}}$) while preserving $\lambda_{\text{retr-gen}}$, $\langle\theta\rangle\big(\chi_W(f) *_{\lambda_{\text{retr-gen}}} \delta_{C^\star\setminus W}(f)\big)$ *if* $\langle\theta\rangle\,\Box^{+}\neg\big(\psi_E \wedge \chi_{C^\star}(f)\big)$. No proper subset of $C^\star$ satisfies the above conditions.

7 Metatheory: Soundness and Completeness

We state and prove the formal results. Under some finiteness conditions (Definition 15) our completeness result (Theorem 2) ensures logical equivalence implies a cut-preserving bisimulation under intervention (cf. [5,8] for related definitions).

Definition 15 (Image-finiteness). *An augmented model $\widehat{\mathcal{M}}$ is* image-finite *if every $f \in F$ has finitely many $\Delta_{\mathcal{I}}$-successors. We say Θ and Λ are* operationally finite *for $\widehat{\mathcal{M}}$ if: (i) only finitely many interventions $\theta \in \Theta$ have $\widehat{\mathcal{M}}_\theta$ well-defined and pairwise distinct; (ii) only finitely many $\lambda \in \Lambda$ are admissible.* □

Definition 16 (Bisimulation Relation). *Let $\widehat{\mathcal{M}}_i$ be interface-admitting augmented models with configuration sets F_i and policy profiles Π_i ($i = 1, 2$). A relation $\mathscr{R}$ between pointed states $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ and $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ is a* cut-preserving bisimulation under intervention *if the following hold:*

1. ***Atoms.*** *For all atoms $p \in \mathcal{P}$, $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models p$ iff $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models p$.*
2. ***(Back and Forth).*** *If $f_1 \Delta_{\mathcal{I}_1}^{\Pi_1} y_1$, then there is $y_2 \in F_2$ with $f_2 \Delta_{\mathcal{I}_2}^{\Pi_2} y_2$ and $((\widehat{\mathcal{M}}_1, y_1, \Pi_1), (\widehat{\mathcal{M}}_2, y_2, \Pi_2)) \in \mathscr{R}$. Symmetrically for steps from f_2.*

3. ***Interventions.*** *For every $\theta \in \Theta$, we write $\widehat{\mathcal{M}}_{i,\theta} \coloneqq [\![\theta]\!](\widehat{\mathcal{M}}_i)$ and $\Pi_{i,\theta}$ for the updated profile (Definition 9). If there exists y_1 with $f_1\,(\Delta_{\mathcal{I}_1}^{\Pi_{1,\theta}})^+\,y_1$ in $\widehat{\mathcal{M}}_{1,\theta}$, then there exists a y_2 with $f_2\,(\Delta_{\mathcal{I}_2}^{\Pi_{2,\theta}})^+\,y_2$ in $\widehat{\mathcal{M}}_{2,\theta}$ and $((\widehat{\mathcal{M}}_{1,\theta}, y_1, \Pi_{1,\theta}), (\widehat{\mathcal{M}}_{2,\theta}, y_2, \Pi_{2,\theta})) \in \mathscr{R}$. Symmetrically from f_2.*
4. ***Indexed separation.*** *If $\lambda \in \Lambda$ is admissible at $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ with $\mathsf{cut}_1(\lambda) = (C_1^1, I^1, C_2^1)$, then there exists $\lambda' \in \Lambda$ admissible at $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ with $\mathsf{cut}_2(\lambda') = (C_1^2, I^2, C_2^2)$ such that: $(\widehat{\mathcal{M}}_1 \restriction_{C_j^1}, f_1 \restriction_{C_j^1}, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2 \restriction_{C_j^2}, f_2 \restriction_{C_j^2}, \Pi_2)$ where $j = 1, 2$. The symmetric condition holds exchanging 1 and 2.*

We write $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2, f_2, \Pi_2)$. □

Remark 4. A cut-preserving bisimulation relation is called *only intervention preserving* when condition 4 in Definition 16 does not hold.

Theorem 2 (Completeness). *Let $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ and $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ be interface-admitting, image-finite pointed models. Assume that for all $\varphi \in \mathcal{L}(\langle\Theta\rangle, *_\Lambda)$, $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \varphi$ implies $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \varphi$ and $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \varphi$ implies $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \varphi$. Then there is a relation $\mathscr{R}$ s.t. $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ and $\mathscr{R}$ is a intervention-preserving bisimulation.* □

Proof. Let $\mathsf{Th}(x) \coloneqq \{\ \varphi \in \mathcal{L}(\langle\Theta\rangle, *_\Lambda) \mid (\widehat{\mathcal{M}}, f, \Pi) \models \varphi\ \}$, and let $\mathscr{R} \coloneqq \big\{\ ((\widehat{\mathcal{M}}_1, f_1, \Pi_1), (\widehat{\mathcal{M}}_2, f_2, \Pi_2)) \mid \mathsf{Th}(\widehat{\mathcal{M}}_1, f_1, \Pi_1) = \mathsf{Th}(\widehat{\mathcal{M}}_2, f_2, \Pi_2)\ \big\}$. We verify the bisimulation clauses of Definition 16.

Atoms. If $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2, f_2, \Pi_2)$, then for every $p \in \mathcal{P}$, $p \in \mathsf{Th}(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ implies $p \in \mathsf{Th}(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$, and conversely; hence both satisfy the same atoms.

Steps (forth/back). Fix $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2, f_2, \Pi_2)$. By image-finiteness, the successor set $\mathrm{Succ}_i(f_i) \coloneqq \{g \mid f_i\, \Delta_{\mathcal{I}_i}^{\Pi_i}\, g\}$ is finite. Let $\{\chi_j^1\}_{j \in J}$ be a finite *characteristic family* for $\mathrm{Succ}_1(f_1)$, i.e., for each $g \in \mathrm{Succ}_1(f_1)$ there is a unique j with $(\widehat{\mathcal{M}}_1, g, \Pi_1) \models \chi_j^1$, and distinct successors satisfy distinct χ_j^1. Then $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \Diamond\Big(\bigvee_{j \in J} \chi_j^1\Big)$ *if* $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \Diamond\Big(\bigvee_{j \in J} \chi_j^1\Big)$, so there exists g_2 with $f_2\, \Delta_{\mathcal{I}_2}^{\Pi_2}\, g_2$ and $(\widehat{\mathcal{M}}_2, g_2, \Pi_2) \models \chi_{j^*}^1$ for some $j^* \in J$. By construction of characteristic families, there is $g_1 \in \mathrm{Succ}_1(f_1)$ with $(\widehat{\mathcal{M}}_1, g_1, \Pi_1) \models \chi_{j^*}^1$ and $\mathsf{Th}(\widehat{\mathcal{M}}_1, g_1, \Pi_1) = \mathsf{Th}(\widehat{\mathcal{M}}_2, g_2, \Pi_2)$; hence $(\widehat{\mathcal{M}}_1, g_1, \Pi_1) \mathscr{R} (\widehat{\mathcal{M}}_2, g_2, \Pi_2)$. The back direction is symmetric, using a characteristic family for $\mathrm{Succ}_2(f_2)$.

Interventions. Fix $\theta \in \Theta$. Write $\widehat{\mathcal{M}}_{i,\theta} \coloneqq [\![\theta]\!](\widehat{\mathcal{M}}_i)$ and $\Pi_{i,\theta}$ for the updated profile (Definition 9). At $(\widehat{\mathcal{M}}_i, f_i, \Pi_i)$, by operational finiteness of Θ and image-finiteness, the set of *reachable* post-intervention points

$$R_i(\theta) \coloneqq \{\, g \mid f_i\, (\Delta_{\mathcal{I}_i}^{\Pi_{i,\theta}})^+\, g \text{ in } \widehat{\mathcal{M}}_{i,\theta} \,\}$$

admits a finite characteristic family $\{\psi_k^i\}_{k \in K_i}$ that distinguishes its members. Assume $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \langle\theta\rangle\psi_k^1$ for some k. Then $\langle\theta\rangle\psi_k^1 \in \mathsf{Th}(\widehat{\mathcal{M}}_1, f_1, \Pi_1) =$

$\mathsf{Th}(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$, so $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \langle\theta\rangle\psi_k^1$. Hence there exists $g_2 \in R_2(\theta)$ with $(\widehat{\mathcal{M}}_{2,\theta}, g_2, \Pi_{2,\theta}) \models \psi_k^1$. By characteristic completeness of the family for $R_1(\theta)$, there is some $g_1 \in R_1(\theta)$ with $(\widehat{\mathcal{M}}_{1,\theta}, g_1, \Pi_{1,\theta}) \models \psi_k^1$ and $\mathsf{Th}(\widehat{\mathcal{M}}_{1,\theta}, g_1, \Pi_{1,\theta}) = \mathsf{Th}(\widehat{\mathcal{M}}_{2,\theta}, g_2, \Pi_{2,\theta})$. Therefore $(\widehat{\mathcal{M}}_{1,\theta}, g_1, \Pi_{1,\theta})\,\mathscr{R}\,(\widehat{\mathcal{M}}_{2,\theta}, g_2, \Pi_{2,\theta})$. The back direction is symmetric.

Interface-cuts. By operational finiteness of Λ, at each pointed state only finitely many cut labels are admissible. Suppose λ is admissible at the pointed augmented model $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ with $\lambda = (C_1^1, I^1, C_2^1)$. Let $\{\alpha_p\}$ be a finite characteristic family for the restricted pointed model $(\widehat{\mathcal{M}}_1{\restriction}_{C_1^1}, f_1{\restriction}_{C_1^1}, \Pi_1)$, and $\{\beta_q\}$ a finite characteristic family for $(\widehat{\mathcal{M}}_1{\restriction}_{C_2^1}, f_1{\restriction}_{C_2^1}, \Pi_1)$. Consider the (finite) set of indexed separation formulas $\mathcal{S}_\lambda := \{\alpha_p *_\lambda \beta_q \mid p, q\}$. For every $\sigma \in \mathcal{S}_\lambda$ we have $\sigma \in \mathsf{Th}(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ implies $\sigma \in \mathsf{Th}(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$, and conversely. Therefore,

(a) There exists some $\lambda' = (C_1^2, I^2, C_2^2)$ admissible at $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ such that each $\alpha_p *_\lambda \beta_q$ is true at $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ witnessed by the corresponding restrictions to C_1^2 and C_2^2. This gives *cut existence (forth).*

(b) Moreover, by the way α_p and β_q characterize the local theories, we have

$$(\widehat{\mathcal{M}}_1{\restriction}_{C_j^1},\, f_1{\restriction}_{C_j^1},\, \Pi_1)\;\mathscr{R}\;(\widehat{\mathcal{M}}_2{\restriction}_{C_j^2},\, f_2{\restriction}_{C_j^2},\, \Pi_2), \qquad j \in \{1, 2\},$$

i.e., *local* forth/back links on both sides of the interface.

The symmetric argument (starting from an interface-cut at $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$) yields the back condition.

We have shown that $\mathscr{R}$ satisfies atoms, step-forth (or back), intervention-forth (or back), and the cut-preserving clause with local links. Therefore $\mathscr{R}$ is a cut-preserving bisimulation under intervention relating $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ and $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$, as required. □

Soundness is established for the *restricted* language $\mathcal{L}(\langle\Theta\rangle)$ without the indexed separating connective $*_\lambda$ (cf. [3], which solves a similar problem):

Theorem 3 (Soundness). *If two image-finite pointed models $(\widehat{\mathcal{M}}_1, f_1, \Pi_1)$ and $(\widehat{\mathcal{M}}_2, f_2, \Pi_2)$ are intervention-preserving bisimilar, then, for all $\varphi \in \mathcal{L}(\langle\Theta\rangle)$, $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \varphi$ iff $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \varphi$.* □

Proof. By structural induction on φ.

1. *Atoms and Boolean formulae:* Immediate from Definition 16 and the induction hypothesis.
2. *$\Diamond$ and $\Box$:* Suppose $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \Diamond\psi$. Then there exists f_1' with $f_1, \Delta_{\mathcal{I}_1}^{\Pi_1}, f_1'$ and $(\widehat{\mathcal{M}}_1, f_1', \Pi_1) \models \psi$. By the step *forth* clause in Definition 16, there is f_2' with $f_2, \Delta_{\mathcal{I}_2}^{\Pi_2}, f_2'$ and $(\widehat{\mathcal{M}}_1, f_1', \Pi_1)\mathscr{R}(\widehat{\mathcal{M}}_2, f_2', \Pi_2)$. By the induction hypothesis, $(\widehat{\mathcal{M}}_2, f_2', \Pi_2) \models \psi$, hence $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \Diamond\psi$. The converse follows from the *back* clause, and the $\Box$ case is analogous.

3. $\langle\theta\rangle$*:* Suppose $(\widehat{\mathcal{M}}_1, f_1, \Pi_1) \models \langle\theta\rangle\psi$. Then there is f_1' with $f_1(\Delta^{\Pi_{1,\theta}}_{\mathcal{I}_1^\theta})f_1'$ and $(\widehat{\mathcal{M}}_{1\theta}, f_1', \Pi_{1,\theta}) \models \psi$, where $(\widehat{\mathcal{M}}_{1\theta}, \Pi_{1,\theta}) = [\![\theta]\!](\widehat{\mathcal{M}}_1, \Pi_1)$. By the *intervention* clause of the bisimulation, the intervened structure $(\widehat{\mathcal{M}}_{2\theta}, \Pi_{2,\theta}) = [\![\theta]\!](\widehat{\mathcal{M}}_2, \Pi_2)$ exists and $(\widehat{\mathcal{M}}_{1\theta}, f_1, \Pi_{1,\theta})\ \mathscr{R}_\theta\ (\widehat{\mathcal{M}}_{2\theta}, f_2, \Pi_{2,\theta})$, with $\mathscr{R}_\theta$ a bisimulation between the intervened models. By the *forth* clause there exists f_2' with $f_2(\Delta^{\Pi_{2,\theta}}_{\mathcal{I}_2^\theta})f_2'$ and $(\widehat{\mathcal{M}}_{1\theta}, f_1', \Pi_{1,\theta})\mathscr{R}_\theta(\widehat{\mathcal{M}}_{2\theta}, f_2', \Pi_{2,\theta})$. By the induction hypothesis (applied in the intervened models), $(\widehat{\mathcal{M}}_{2\theta}, f_2', \Pi_{2,\theta}) \models \psi$. Therefore $(\widehat{\mathcal{M}}_2, f_2, \Pi_2) \models \langle\theta\rangle\psi$. The converse implication is symmetric and uses the *back* clause. □

8 Conclusion

The language $\mathcal{L}(\langle\Theta\rangle, *_\Lambda)$ combines a single intervention modality $\langle\theta\rangle$ with *indexed* separating conjunction $*_\lambda$, enabling modular reasoning across interfaces, and agent policies are captured as interventions. We have defined actual causation directly in this logic and established alignment with the Halpern-Pearl account of actual causation via a time-unfolding construction. We have also established a soundness and completeness property in the form of a Hennessy-Milner-van Benthem-Bergstra 'bisimulation invariance', under the necessary finiteness assumptions [7].

Our present account has two main limitations: we assume fixed interfaces, and our reasoning is qualitative. These constraints suggest concrete extensions. Drawing on quantitative model checking [12], probabilities on system configuration transitions can be placed.

Enriching the logic with quantitative modalities so that optimization claims, such as 'the minimum accumulated cost of changing mechanisms ensuring φ is c', where costs range over budgets, compute quotas, and permission/approval requirements, in line with emerging governance standards [24]. Interfaces and influence contexts may be allowed to evolve, capturing how mechanisms and dependencies are re-learned or reconfigured as systems adapt. This connects to work on causal structure learning and dynamic causal discovery [14,34].

Taken together, and expressible within our single language for configuration transitions, interventions, and modular decompositions, these extensions point toward a unified, certifiable logic of causal design, supporting probabilistic guarantees, cost-aware mechanism change, and adaptive boundaries for evolving multiagent systems.

Acknowledgements

Chakraborty is supported by a studentship from UCL's EPSRC-funded Centre for Doctoral Training in Cybersecurity (EP/S022503/1).

References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. J. ACM p. 672–713 (2002). https://doi.org/10.1145/585265.585270
2. Amazon Web Services: MLOps: Continuous Delivery for Machine Learning on AWS (2020), https://d1.awsstatic.com/whitepapers/mlops-continuous-delivery-machine-learning-on-aws.pdf
3. Anderson, G., Pym, D.: A calculus and logic of bunched resources and processes. Theor. Comput. Sci. **614**(C), 63–96 (2016). https://doi.org/10.1016/j.tcs.2015.11.035
4. Anthropic: Model Context Protocol, https://modelcontextprotocol.io/specification/2025-06-18
5. Aucher, G., van Benthem, J., Grossi, D.: Modal logics of sabotage revisited. J. Log. Computat. **28**(2), 269–303 (2017). https://doi.org/10.1093/logcom/exx034
6. Baier, Christel et al. : From verification to causality-based explications. In: LIPIcs 198: 48th Int. Colloq. Automata, Languages, and Programming (ICALP 2021). pp. 1:1–1:20 (2021). https://doi.org/10.4230/LIPIcs.ICALP.2021.1
7. van Benthem, J., Bergstra, J.: Logic of transition systems. J. Log. Lang. Inf. **3**(4), 247–283 (1994). https://doi.org/10.1007/bf01160018
8. Blackburn, P., de Rijke, M., Venema, Y.: Modal logic. CUP (2001)
9. Bujorianu, M., Caulfield, T., Ilau, M.C., Pym, D.: Interfaces in ecosystems: Concepts, form, and implement. In: Simulation Tools and Techniques. pp. 27–47. Springer (2025)
10. Caulfield, T., Ilau, M.C., Pym, D.: Engineering Ecosystem Models: Semantics and Pragmatics. In: Simulation Tools and Techniques. pp. 236–258. Springer (2022)
11. Chakraborty, P., Caulfield, T., Pym, D.: Causality and decision-making: A logical framework for systems and security modelling (2025), https://arxiv.org/abs/2508.01758
12. Chen, T., Forejt, V., Kwiatkowska, M., Parker, D., Simaitis, A.: Prism-games: A model checker for stochastic multi-player games. In: Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. p. 185–191 (2013). https://doi.org/10.1007/978-3-642-36742-7_13
13. Dubslaff, C.e.a.: Causality in configurable software systems. In: Proceedings of the 44th International Conference on Software Engineering. pp. 325–337. Association for Computing Machinery (2022). https://doi.org/10.1145/3510003.3510200
14. Eberhardt, F.: Introduction to the Epistemology of Causation. Philosophy Compass pp. 913–925 (2009). https://doi.org/10.1111/j.1747-9991.2009.00243.x
15. Everitt, T., Carey, R., Langlois, E.D., Ortega, P.A., Legg, S.: Agent incentives: A causal perspective. Proceedings of the AAAI Conference on Artificial Intelligence **35**(13), 11487–11495 (May 2021). https://doi.org/10.1609/aaai.v35i13.17368
16. Financial Conduct Authority, United Kingdom: Artificial Intelligence (AI) update (2024), https://www.fca.org.uk/publication/corporate/ai-update.pdf
17. Galmiche, D., Lang, T., Pym, D.: Minimalistic System Modelling: Behaviours, Interfaces, and Local Reasoning. In: Proc 16th EAI SIMUtools, Springer, 2024 (2024), https://doi.org/10.48550/arXiv.2401.16109, Accessed 9 June 2025
18. Gao, Y. et al.: Retrieval-augmented generation for large language models: A survey (2024), https://arxiv.org/abs/2312.10997
19. Geiger, A., Lu, H., Icard, T., Potts, C.: Causal abstractions of neural networks. In: Proc. 35th Int. Conf. on Neural Information Processing Systems (2021)

20. Halpern, J.Y.: Actual Causality. The MIT Press (2016). https://doi.org/10.7551/mitpress/10809.001.0001
21. Halpern, J.Y., Pearl, J.: Causes and Explanations: A Structural-Model Approach. Part I: Causes. Brit. J. Phil. Sci. **56**(4), 843–887 (2005)
22. Hammond, L., Fox, J., Everitt, T., Carey, R., Abate, A., Wooldridge, M.: Reasoning about causality in games. Artificial Intelligence **320**, 103919 (2023). https://doi.org/https://doi.org/10.1016/j.artint.2023.103919
23. Howard, R.A., Matheson, J.E.: Influence diagrams. Decision Analysis **2**(3), 127–143 (2005). https://doi.org/10.1287/deca.1050.0020
24. International Organization for Standardization: ISO/IEC 42001:2023 — information technology — artificial intelligence management systems. https://www.iso.org/standard/81230.html (2023)
25. Joachims, T., Swaminathan, A.: Counterfactual evaluation and learning for search, recommendation and ad placement. In: Proc. 39th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. pp. 1199–1201 (2016). https://doi.org/10.1145/2911451.2914803
26. Khattab, O. et al.: DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines (2023), https://arxiv.org/abs/2310.03714
27. Koller, D., Milch, B.: Multi-agent influence diagrams for representing and solving games. Games and Economic Behavior **45**(1), 181–221 (2003). https://doi.org/doi.org/10.1016/S0899-8256(02)00544-4
28. Krishna, R., Iqbal, M.S., Javidian, M.A., Ray, B., Jamshidi, P.: CADET: Debugging and Fixing Misconfigurations using Counterfactual Reasoning (2021), https://arxiv.org/abs/2010.06061
29. Lewis, Patrick et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Proc. 34th Int. Conf. on Neural Information Processing Systems (2020)
30. Machamer, P., Darden, L., Craver, C.F.: Thinking About Mechanisms. Philosophy of Science **67**(1), 1–25 (2000). https://doi.org/10.1086/392759
31. National Institute of Standards and Technology, U.S. Department of Commerce: Artificial intelligence risk management framework (2024), https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
32. O'Hearn, P.W., Pym, D.J.: The Logic of Bunched Implications. Bulletin of Symbolic Logic **5**(2), 215–244 (1999). https://doi.org/10.2307/421090
33. Pearl, J.: Causality: Models, Reasoning and Inference. CUP, 2nd edn. (2009)
34. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward Causal Representation Learning. Proceedings of the IEEE **109**(5), 612–634 (2021)
35. Shachter, R.D.: Evaluating influence diagrams. Operations Research **34**(6), 871–882 (1986). https://doi.org/10.1287/opre.34.6.871
36. Simon, H.A., Barnard, C.I.: Administrative Behavior: A study of Decision-making Processes in Administrative Organization. Macmillan Co. (1947)
37. Wang, N., Han, X., Singh, J., Ma, J., Chaudhary, V.: CausalRAG: Integrating causal graphs into retrieval-augmented generation. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 22680–22693 (2025). https://doi.org/10.18653/v1/2025.findings-acl.1165
38. Woodward, J.: What is a Mechanism? A Counterfactual Account. Philosophy of Science **69**(S3), 366–377 (2002). https://doi.org/10.1086/341859