When Should Reinforcement Learning Use Causal Reasoning?

Anonymous authors
Paper under double-blind review

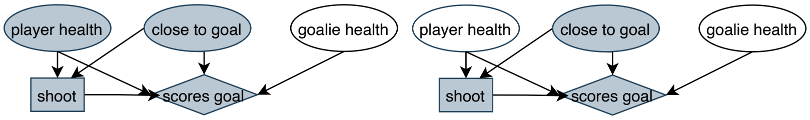
Abstract

Reinforcement learning (RL) and causal reasoning naturally complement each other. The goal of causal reasoning is to predict the effects of interventions in an environment, while the goal of reinforcement learning is to select interventions that maximize the rewards the agent receives from the environment. Reinforcement learning includes the two most powerful sources of information for estimating causal relationships: temporal ordering and the ability to act on an environment. This paper provides a theoretical study examining which reinforcement learning settings we can expect to benefit from causal reasoning, and how. According to our analysis, the key factor is *whether the behavioral policy—which generates the data—can be executed by the learning agent*, meaning that the observation signal available to the learning agent comprises all observations used by the behavioral policy. Common RL settings with behavioral policies that are executable by the learning agent include on-policy learning and online exploration, where the learning agent uses a behavioral policy to explore the environment. Common RL settings with behavioral policies that are not executable by the learning agent include offline learning with a partially observable state space and asymmetric imitation learning where the demonstrator has access to more observations than the imitator. Using causal graph theory, we show formally that when the behavioral policy is executable by the learning agent, conditional probabilities are causal, and can therefore be used to estimate expected rewards as done in traditional RL. However, when the behavioral policy is not executable by the learning agent, conditional probabilities may be confounded and provide misleading estimates of expected rewards. For confounded settings, we describe previous and new methods for leveraging causal reasoning.

1 Introduction: Causal Probabilities in Reinforcement Learning

The goal of decision-making in a Markov Decision Process (MDP) is to intervene in the environment to maximize the agent's cumulative reward. A key insight of causal decision theory is that the impact of an action should be estimated as a *causal effect*, not a correlation. Visits to the doctor correlate with illnesses, but avoiding seeing a doctor does not make a patient healthier (Pearl, 2000, Ch.4.1.1). Several causality researchers have therefore argued that reinforcement learning can benefit from adopting causal models to predict the effect of actions. This article is directed towards reinforcement learning researchers who want to explore the use of causal models. We provide conceptual and theoretical foundations to facilitate the adoption of causal models by reinforcement learning researchers. We use as much as possible terminology, notation, and examples from reinforcement learning. A running example gives explicit computations that illustrate causal concepts. This paper can therefore serve as a short tutorial on causal modeling for RL researchers. An excellent long tutorial is provided by Bareinboim (2020), and a recent survey by Deng et al. (2023).

Overview. Conditional probabilities measure the strength of associations or correlations, but not necessarily the causal effect of an action. Using Pearl's do operator, the **causal effect** of setting variable A to the value a given evidence covariates $\boldsymbol{X}$ can be written as a conditional probability of the form $P(Y|do(A=a), \boldsymbol{X}=\boldsymbol{x}))$. (The formal semantics for the *do* operator is defined in Section 2 below.) Since



(a) Online View. (b) Offline View.

Figure 1: Causal Graphs for a sports scenario like hockey or soccer to illustrate executability. We follow the conventions of influence diagrams to distinguish state variables, actions, and rewards. All variables are binary. Variables observed by the learning agent are gray, latent variables white. Whether a player takes a shot depends on their location and whether they are injured. Likewise, the chance of their shot leading to a goal depends on their location and health. Thus Player Health is a common cause of the action and reward. Figure 1a: In the online setting, the athlete learns from their own experience, which includes their health. Therefore a behavioral policy that depends on the agent's health is executable. Figure 1b: In the offline setting, the learner is different from the athlete, for example a coach, and does not observe the health of the behavioral agent. Therefore a behavioral policy that depends on the agent's health is not executable. Player health is an unobserved confounder of action and reward.

causal effects are based on interventions, we refer them also as **interventional probabilities**. In the medical visit example, the strong correlation means that $P(\mathit{Illness}|\mathit{Visit})$ is high. However, making a person visit the doctor has no causal effect on their illness, so we have $P(\mathit{Illness}|\mathit{Visit}) >> P(\mathit{Illness}|do(\mathit{Visit})) = P(\mathit{Illness})$.

Summary of Main Conclusion. The question we address is *under what conditions causal probabilities provide a different approach to reinforcement learning than conditional probabilities.* For example, we consider the conditions under which the conditional reward probability equals the interventional reward probability:

$$P(R_{t+1}|do(A_t = a_t), S_t) = P(R_{t+1}|A_t = a_t, S_t)$$

where $P(R_{t+1}|A_t = a_t, S_t)$ is the conditional probability of receiving reward R at time $t+1$ given action A_t and state S_t at time t.

Our answer is that the key factor for the equivalence is *whether the behavioral policy—which generates the data—can be executed by the learning agent.* By "executable" we mean that the observation signal available to the learning agent comprises all observations used by the behavioral policy. Figure 1 illustrates the concept in a simple sports setting (such as hockey or soccer). Player health is a common cause of the player's decision (e.g., to shoot) and the player's success (score a goal). In the first-person online setting, the athlete is aware of their own health, and can execute a policy that depends on the athlete's health (e.g., "shoot only if healthy"). In the third-person offline setting, an observer such as a coach, does not have access to the athlete's health. They can execute only policies that depend on whether a player is close to the goal, but not those that depend on whether the player is healthy.

Our main theorem states that *when the behavioral policy that generates the data is executable by the learning agent, conditional probabilities estimated from online data are also causal probabilities* (i.e., they represent the causal effect of interventions). For non-executable behavioral policies, that depend on observations not available to the learning agent, causal probabilities provide an alternative to conditional probabilities. In this case, the variables that are observable for the behavioral agent but not the learning agent can be, and often are, confounders that influence both the actions selected by and the rewards obtain by the behavioral agent. We illustrate computationally in our running example how in the presence of such confounders, *conditional probabilities lead to wrong estimates of the values of learned policies.* For example, in the offline sports example Figure 1b, policy evaluation based on conditional probabilities will lead to an overestimate of the value of the policy "shoot if you get close to the goal" if the observer does not take into account that the behavioral agent shoots only if they are healthy (which is not observable). This is true even in the infinite sample setting where probabilities can be estimated perfectly.

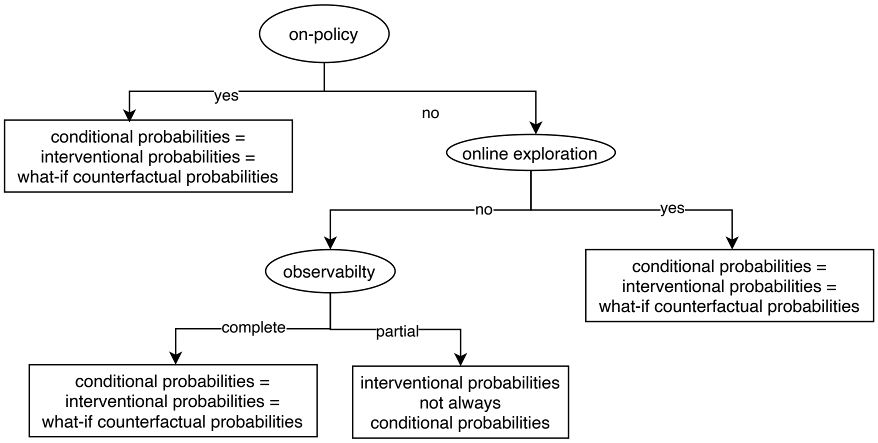

on-policy
yes
no
conditional probabilities = interventional probabilities = what-if counterfactual probabilities
online exploration
no
yes
observabilty
conditional probabilities = interventional probabilities = what-if counterfactual probabilities
complete
partial
conditional probabilities = interventional probabilities = what-if counterfactual probabilities
interventional probabilities not always conditional probabilities

Figure 2: Reinforcement learning settings in which we can expect conditional probabilities to be equivalent to interventional probabilities and to what-if counterfactual probabilities. Online exploration refers to a setting in which the learning agent executes a behavioral policy to explore the environment. Hindsight counterfactuals cannot be reduced to conditional probabilities in any setting.

Next we discuss which RL settings satisfy the executability condition. Our overall conclusion, illustrated in Figure 2, is that *interventional probabilities differ from conditional probabilities in the offline off-policy RL setting with partial observability* (the POOO setting).

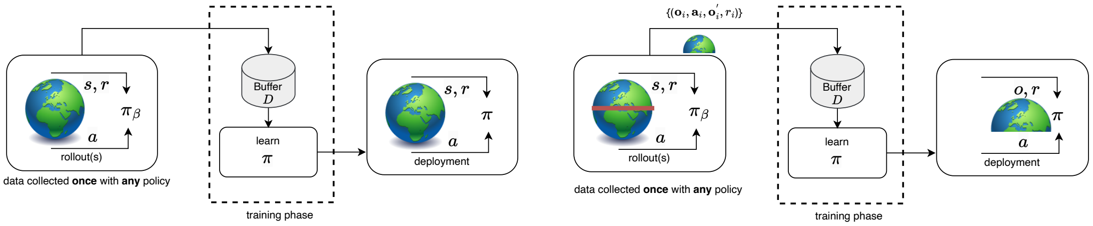
RL settings with executable behavioral policies. Common RL settings with behavioral policies that are executable by the learning agent include on-policy learning and online exploration, where the learning agent uses a behavioral policy to explore the environment (e.g., ϵ-greedy learning); see Figure 18. In *on-policy* learning the behavioral and the learned policy coincide so they clearly share the same observation signal. (For more explanation, please see Sutton & Barto (1998) and Figure 18 below.) Accordingly, Schölkopf et al. (2021) note that "[Reinforcement learning] sometimes effectively directly estimates do-probabilities. E.g., on-policy learning estimates do-probabilities for the interventions specified by the policy". Our work formalizes their observation that the on-policy setting is sufficient for conditional and interventional probabilities to coincide, and goes beyond it to consider other RL settings, such as the following.

In *online exploration*, the learning agent directly interacts with its environment to generate data and learn from their *own* experience exploring the environment (see Figure 18 below). For example a video game playing system can execute actions in the game and observe their effects (Mnih et al., 2015). Since the behavioral policy is executed by the learning agent to explore the environment, it must be executable. In the video game example, if the game playing agent employs an exploration policy, the policy must depend only on the information available in the video game.

Another sufficient condition for executability is *complete observability*, where the environment is completely observable for both the behavioral and the learning agent. For instance, the first phase of training the AlphaGo system was based on an offline dataset of games of Go masters (Silver et al., 2016). Go is a completely observable board game with no hidden information. Under complete observability, the learning agent has access to the same observations as the behavioral agent, as required for executability.

The Go example also illustrates how imitation learning can satisfy the executability condition if both the demonstrator and the learning agent (trainee), who aims to imitate them, share the same observation space. For an illustration in online imitation learning, in the DAgger approach (Ross et al., 2011), the trainee explores the state space using a mixture of the demonstrator's policy and the current learned policy. Using a mixture implies that both the expert and the trainee's policies operate with the same input observations.

RL settings without executable behavioral policies. As we show in worked-out examples below, *when the behavioral policy is not executable by the learning agent, conditional probabilities may be confounded and* *provide misleading estimates of expected rewards.* In the *offline* setting, the learning agent cannot directly interact with its environment, and instead relies on an offline dataset; see Figure 3. In this case, the learning agent may receive a different observation signal than the behavioral agent (Figure 3b). When the learning agent does not have access to all the causes of the decisions made by the behavioral agent, *actions and rewards may be confounded* from the learner's perspective. In the example of Figure 1b, Player Health is not observable by the learning agent, and thus confounds the decisions of the behavioral agent (i.e., whether the athlete shoots) with the rewards obtained by the behavioral agent (i.e., whether the athlete scores).



(a) Offline RL: executable behavioral policies. (b) Offline RL: non-executable behavioral policies.

Figure 3: **Offline** RL-learning employs a dataset D collected by some (potentially unknown) behavior policy π_β. The dataset is collected once, and is not altered during training. The training process does not interact with the environment directly, and the policy is only deployed after being fully trained. Figure 3a (Levine et al., 2020): offline RL with observation equivalence where the behavioral policy π_β and the learned policy π are based on the same observation signal. Figure 3b: offline RL without observation equivalence where the behavior policy π_β has access to more observations than the learned policy π.

Several algorithms have been developed for *imitation learning in a non-executable setting* where the expert has access to more observations than the trainee. Ha et al. (2024) refer to this type of setting as privileged learning, whereas Warrington et al. (2021) use the term asymmetric imitation learning.

All results described so far hold even in the infinite sample setting with perfect probability estimates. With limited samples, confounding can become an issue even when the behavioral agent and the learning agent are one and the same. This is due to the implicit partial observability phenomenon described by Ghosh et al. (2021): when generalizing a policy from training contexts to a new test-time context, the agent may have less information at test time than at training time.[1] An example given by Ghosh et al. is navigating to the location of a target object using a map. The training data may allow the agent to train an image classifier with near-perfect accuracy to extract the location, whereas when tested in a new context, the image classifier may provide insufficiently accurate information. During training time, the output of the image classifier is a common cause of the agent's actions (route chosen) and the reward (target found). Ghosh et al. suggest that because of limited samples, at test time the agent has only partial observability of the image information, which in contrast is perfectly observable during training. In our terminology, the test time agent is in a POOO setting, and policy estimates based on conditional probabilities from the training data will be confounded.

Paper Overview The paper is organized following a "ladder of causation" as described by (Pearl, 2000): A hierarchy of probabilistic statements that require causal reasoning of increasing complexity. The levels correspond to associations, interventions, and counterfactuals. Table 1 illustrates these concepts in the RL setting. A *formal semantics* for each type of probability can be defined in terms of a generative model that is based on a causal graph. Among counterfactuals, we distinguish between *what-if* queries and *hindsight* queries. A what-if query concerns the results of deviating from an action taken; an example from a sports domain would be "What if I had taken a shot instead of making a pass?". A hindsight query conditions on an observed outcome. An example of a hindsight query would be "I failed to score. What if I had taken a shot instead of making a pass?". We analyze the relationship between RL and each level of the causal hierarchy in three separate independent sections, one for interventional probabilities, one for what-if counterfactuals, and one for hindsight counterfactuals.

---

[1] We owe this point to an anonymous reviewer.

Table 1: A four-level causal hierarchy, which refines Pearl's three-level hierarchy Association-Intervention-Counterfactual (Pearl, 2000). *Our analysis shows that with executable behavioral policies, queries of the first three types can be computed from conditional probabilities.*

Level	Notation	Typical Question	Example
Association	$P(R\|\mathbf{S}, A)$	What reward follows after an agent chooses A?	How often does a shot lead to a goal?
Intervention	$P(R\|\mathbf{S}, do(A))$	If I chose A, what will my reward be?	If I take a shot, will I score a goal?
What-if Counterfactual	$P(R_A\|\mathbf{S}, B)$	What if I had chosen A instead of B?	What if I had taken a shot instead of making a pass?
Hindsight Counterfactual	$P(R_A\|\mathbf{S}, B, R)$	How would my reward change if I had chosen A instead of B?	I failed to score. What if I had taken a shot instead of making a pass?

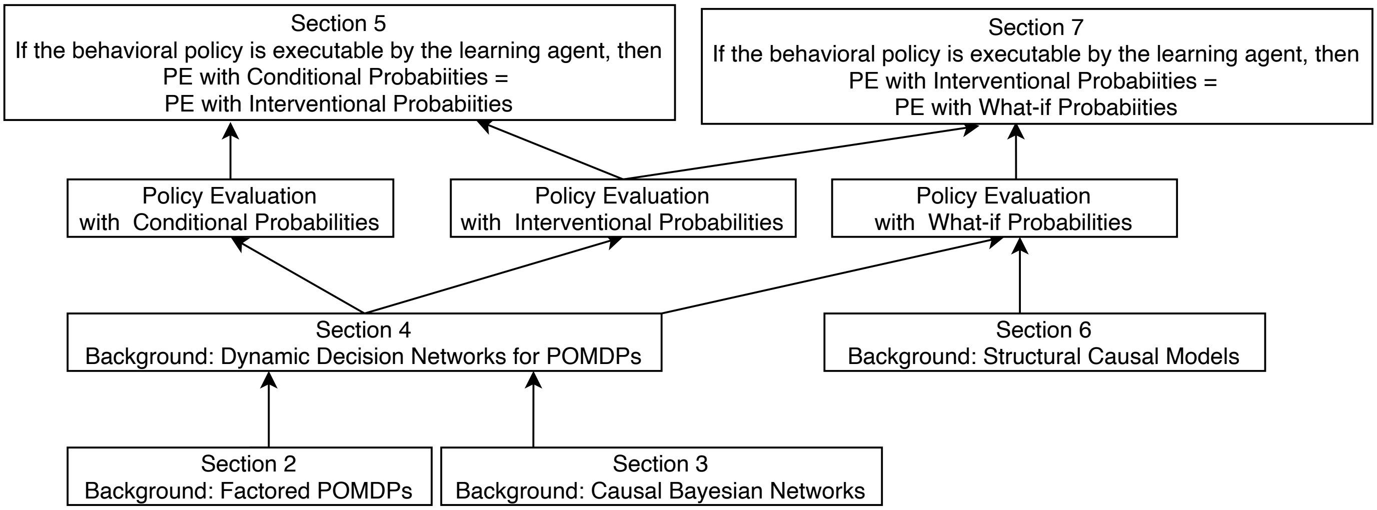

Section 5
If the behavioral policy is executable by the learning agent, then
PE with Conditional Probabiities =
PE with Interventional Probabiities
Section 7
If the behavioral policy is executable by the learning agent, then
PE with Interventional Probabiities =
PE with What-if Probabiities
Policy Evaluation with Conditional Probabilities
Policy Evaluation with Interventional Probabilities
Policy Evaluation with What-if Probabilities
Section 4
Background: Dynamic Decision Networks for POMDPs
Section 6
Background: Structural Causal Models
Section 2
Background: Factored POMDPs
Section 3
Background: Causal Bayesian Networks

Figure 4: Topics by paper section and connections between them. POMDP = Partially Observable Markov Decision Process. PE=Policy Evaluation.

Figure 4 illustrates our paper organization. We begin by reviewing the theory of partially observable Markov Decision Processes (POMDPs). The next background section presents causal Bayesian networks, a class of graphical models that provide an intuitive semantics for interventional probabilities. We then describe a causal graphs that represents POMDPs, which we refer to as *dynamic decision networks* (DDNs) following (Russell & Norvig, 2010, Ch.17). A DDN includes causes of both the agent's actions and the environment's responses.

We show how a DDN supports model-based policy evaluation in three different ways: based on conditional probabilities, interventional probabilities, or what-if probabilities. Our main formal proposition states that when the behavioral policy is executable by the learning agent, policy evaluation with conditional probabilities is equivalent to policy evaluation with interventional probabilities and what-if counterfactuals coincide with conditional probabilities. Moving to the next level of the causal hierarchy, we first review background on structural causal models and how they provide a formal semantics for counterfactual probabilities. We extend our previous equivalence proposition to what-if probabilities. In contrast, hindsight counterfactuals cannot be reduced to conditional probabilities, even under executability. Although interventional and counterfactual probabilities are defined with reference to a causal model, our results do not assume that the learning agent is given a true causal model of the behavioral policy and the environmental process. Rather, the import is that as long as the learning agent's observation signal includes the causes of the behavioral agent's actions, conditional probabilities inferred from the action data are equivalent to interventional and what-if probabilities, no matter what the true causal model is.

Our final section utilizes the concepts and insights developed in the main body to i) review related work on causal models in RL, and ii) describe several directions for future research from the perspective of confounding and executability.

2 Background: Causal Bayesian Networks

Notation Table 2 summarizes the notation used in this paper, and previews the concepts introduced in the remainder.

Notation	Meaning
$\boldsymbol{X}$	Generic Random Variables. Nodes in a Causal Graph. $\boldsymbol{X} = \boldsymbol{O} \cup \boldsymbol{Z}$
$\boldsymbol{O}$	Observed Variables.
$\boldsymbol{Z}$	Latent Variables
$\boldsymbol{U}$	Source Variables in a causal graph.
$\underline{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{U}$	Non-source variables with positive indegree
G	Directed Acyclic Graph (DAG); causal graph
Pa_X resp. Pa_i	Parents/direct causes of variable X resp. $\rv_{i}
B	Causal Bayesian Network
C	Probabilistic Structural Causal Model
f_X resp. f_i	Deterministic local function for variable X resp. $\rv_{i}
S	State Space in POMDP
$\boldsymbol{S}$	State Variables in factored POMDP; $\boldsymbol{S} = \boldsymbol{O} \cup \boldsymbol{Z} - \{A\}$
A	Intervention Target; Decision Variable in Influence Diagram; Action in POMDP
$do(A = \hat{a})$	Selecting action $\hat{a}$ as an intervention
R	Reward
$b(\boldsymbol{Z})$	Belief State
π	Policy
$\pi(a \mid \langle \boldsymbol{o}, b \rangle)$	Probability of action given current observation and belief state
$Q^{\pi}(\langle \boldsymbol{o}, b \rangle, a)$	Expected Return given current observation, belief state, and action
$Q^{\pi}(\langle \boldsymbol{o}, b \rangle, do(\hat{a}))$	Expected Return given current observation, belief state, and action/intervention

Table 2: Notation used in this paper

Our goal in this section is to define the semantics for the first two levels of causal hierarchy in Table 1, observational associations and intervention probabilities, based on *causal Bayesian networks* (CBNs). CBNs specify intervention probabilities through a truncation semantics (Pearl, 2000)[Ch.1.3]. Their parameters are conditional probability parameters of the form $P(child\_value|parent\_values)$. While causal Bayesian networks are easy to interpret, recent research has focused on *structural causal models* (Pearl, 2000)[Ch.2.2],(Schölkopf et al., 2021), which combine latent variables with Bayesian networks. Latent variables enhance the expressive power of causal graphs to define a formal semantics for counterfactual probabilities, which we describe in Section 6 below. In the next Section 3, we show how POMDPs can be represented using causal Bayesian networks.

2.1 Causal Bayesian Networks

A **causal graph** is a directed acyclic graph (DAG) whose nodes are a set of random variables $\boldsymbol{X} = \{X_1, \ldots, X_n\}$. Throughout the paper we assume that random variables are discrete. The definitions can easily be extended to continuous random variables. A **causal Bayesian network** (Pearl, 2000, Ch.1.3),(Cooper

& Yoo, 1999; Bareinboim et al., 2022), or **causal network** for short, is a causal graph G parametrized by conditional probabilities $P(x_i|pa_i)$ for each possible child value X_i and joint parent state pa_i.

A causal network B defines a **joint distribution** over random variables $\boldsymbol{X}$ through the product formula

$$P^B(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{n} P(x_i|pa_i) \tag{1}$$

where x_i resp. pa_i are the values given to node X_i resp. its parents Pa_i by the assignment $\boldsymbol{X} = \boldsymbol{x}$. Here and below we often omit the model index when the model is fixed by context.

A causal network B also defines a joint **interventional distribution** through the *truncation semantics* as follows. Write $do(A = \hat{a})$ to denote an intervention that sets variable A to value $\hat{a}$. In the RL context, A represents an action; in this section, it represents a generic intervention target. The effect of the intervention is to *change the causal network* B to a **truncated network** $B_{do(A=\hat{a})}$, in which node A has no parents and with probability 1 takes on the value $\hat{a}$. Removing the parents of A indicates that the parents of A no longer influence its value after the intervention. Given an intervention on variable A, the truncated network $B_{do(A=\hat{a})}$ defines a joint distribution as follows:

$$P^B_{do(A=\hat{a})}(\boldsymbol{X} = \boldsymbol{x}, A = a) = \prod_{X_i \neq A} P(x_i|pa_i)\delta(a, \hat{a}) \tag{2}$$

where x_i resp. pa_i are the values given to node/random variable X_i resp. its parents Pa_i by the assignment $(\boldsymbol{X} = \boldsymbol{x}, A = a)$ and $\delta(a, \hat{a}) = 1$ if $a = \hat{a}$, 0 otherwise. For interventional probabilities, Pearl (2000, Dfn.3.2.1.), uses conditional notation such as

$$P^B(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, do(A = \hat{a})) \equiv P^B_{do(A=\hat{a})}(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}) \text{ for disjoint } A, \boldsymbol{Y}, \boldsymbol{X}$$

to denote the causal effect on a list of outcome variables $\boldsymbol{Y}$ due to setting variable A to value $\hat{a}$ after observing evidence $\boldsymbol{X} = \boldsymbol{x}$. The truncation semantics easily extends to intervening on multiple variables by removing all their links.

Remarks on Notation. The $\hat{a}$ notation does not indicate a quantity estimated from data, but an intervention. We sometimes use the syntactic sugar $\hat{A}$ to highlight a context where A is intervened upon. In our applications to RL, we consider intervening only on a special variable A that represents the agent's actions. The truncation semantics Equation (2) shows how an action changes the distribution over environment states and rewards. However, the intervention semantics is well-defined for manipulating *any* random variable in a causal model, not only a designated special action/decision variable.

The next lemma states that if the parent values of a manipulated variable are given, interventional probabilities are equivalent to conditional probabilities.

Lemma 1. *Let B be a causal Bayesian network and let $\boldsymbol{Y}, A, \boldsymbol{X}$ be a disjoint set of random variables such that $\boldsymbol{X} \supseteq Pa_A$. Then $P^B(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, do(A = \hat{a})) = P^B(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, A = \hat{a})$. That is, a target probability given an intervention on variable A is the same as the target probability given an observation of variable A.*

The lemma is easily derived from the do-calculus (Pearl, 2000); Appendix B gives a proof directly from the truncation semantics. The significance of the lemma is that if we can observe all the direct causes of an agent's actions, which we argue is the case when the behavioral policy is executable, then the causal effect of an action can be estimated by a conditional probability.

Let $\boldsymbol{X} = (\boldsymbol{O} \cup \boldsymbol{Z})$ be a partition of the random variables into a nonempty set of observed variables $\boldsymbol{O}$ and a set of latent variables $\boldsymbol{Z}$. A set of observed variables $\boldsymbol{O}$ is **causally sufficient** for variable X in a causal graph G if all parents of X are observed; that is, if $pa_X \subseteq \boldsymbol{O}$.[2] A set of observable variables $\boldsymbol{O}$ is

---

[2] In Section 6.4 below we discuss a weaker notion of causal sufficiency that is commonly used in causal discovery (Spirtes et al., 2000).

action sufficient in a causal graph G if it is causally sufficient for the intervention variable A. Lemma 1 implies that if an observation signal $\boldsymbol{O}$ is action sufficient in (the graph of) a CBN B, then conditional on the observations, interventional probabilities of A are equivalent to conditional probabilities:

Lemma 2. *Let $\boldsymbol{O} \subseteq \boldsymbol{X}$ be an action sufficient set of observable variables in a causal Bayesian network B. Then $P^B(\boldsymbol{Y}|\boldsymbol{O} = \boldsymbol{o}, do(A = \hat{a})) = P^B(\boldsymbol{Y}|\boldsymbol{O} = \boldsymbol{o}, A = \hat{a})$ for any list of target outcomes $\boldsymbol{Y}$. That is, a target probability given an intervention on action variable A is the same as the target probability given an observation of action variable A.*

2.2 Examples of Causal Bayesian Networks.

We compute observational and interventional probabilities for taking a shot in the two causal graphs of Figure 1. All nodes are binary. We specify the following conditional probability parameters for the graph structures. For nodes without parents we assume a uniform prior:

$$P(PH = 1) = P(CG = 1) = P(GH = 1) = 1/2$$

where variables are abbreviated with their initials (e.g. $PlayerHealth = PH$).

For illustration, we assume an unrealistically simple noise-free causal mechanism governing the player's behavior: The player shoots if and only if they are healthy and close to the goal. In symbols, we have

$$P(SH = 1|PH, CG) = \begin{cases} 1, & \text{if } PH = CG = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

A player scores if and only if they shoot, are healthy, close to the goal, and the goalie is not healthy. In symbols, we have

$$P(SC = 1|SH, CG, GH) = \begin{cases} 1, & \text{if } SH = CG = 1 \text{ and } GH = 0 \\ 0, & \text{otherwise} \end{cases}$$

This parametrization implies the following *joint probability that the goalie is not healthy and all other variables are true*:

$$P(1 = PH = CG = SH = SC, GH = 0) = 1/2 \cdot 1/2 \cdot 1 \cdot 1 \cdot 1/2 = 1/8. \tag{4}$$

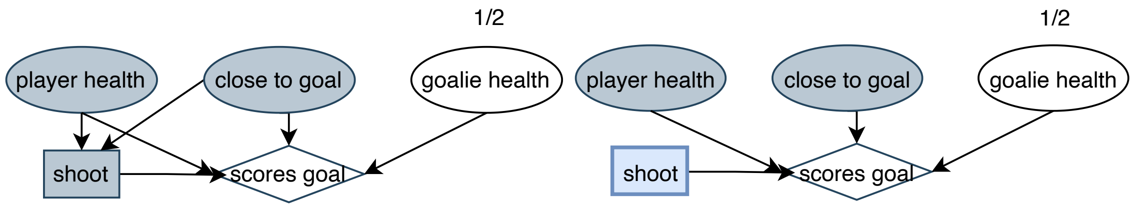
We next compute the *causal effect of shooting on goal scoring, given that all other observable variables are true.* Table 3 gives the resulting scoring probabilities for seeing a player take a shot (observation) vs. intervening to take a shot (action). Figures 5 and 6 show how these probabilities are computed.

Table 3: Goal scoring probabilities derived from the models of Figure 1, given the evidence that all variables observable in the model are true. For observational probabilities, the player is observed to take a shot. For interventional probabilities, the shot is the result of an intervention. Note that observational and interventional probabilities differ only in the confounded offline model Figure 1b.

	Observation $P(SC = 1|\boldsymbol{O} = \boldsymbol{1}, SH = 1)$	Intervention $P(SC = 1|\boldsymbol{O} = \boldsymbol{1}, do(SH = 1))$
Online Model 1*a*	1/2	1/2
Offline Model 1*b*	1/2	1/4

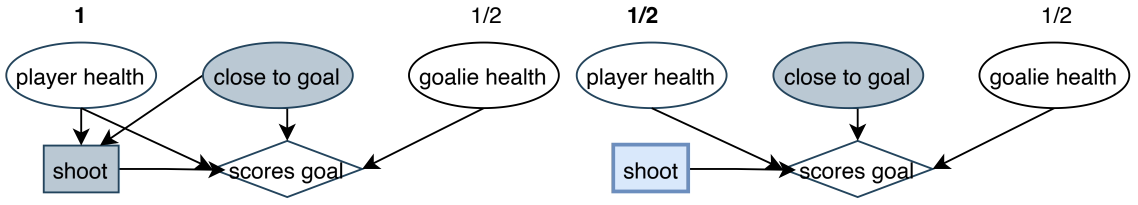
In the online model 1*a*, Player Health (PH) is observed, so the shooting effect queries compared are

$$P(SC = 1|CG = 1, PH = 1, SH = 1) = 1/2 \text{ and } P(SC = 1|CG = 1, PH = 1, do(SH = 1)) = 1/2.$$



(a) Observation (b) Intervention

Figure 5: Observational and intervention probabilities in the online model of Figure 1a. Gray indicates observed variables whose values are specified in the query. Numbers indicate posteriors over latent variables, given the observations. Light blue indicates an intervention on a variable. Figure 5a: The observational scoring probability $P(SC = 1|CG = 1, SH = 1, PH = 1)$ is 1/2, the same as the probability that the goalie is healthy. Figure 5b: The query $P(SC = 1|CG = 1, PH = 1, do(SH = 1))$ is evaluated in the intervention model. The scoring probability remains 1/2, because both player health and shooting are observed, so breaking the causal link between them has no effect on the scoring probability.



(a) Observation (b) Intervention

Figure 6: Observational and intervention probabilities in the confounded offline model of Figure 1b. The query $P(SC = 1|CG = 1, SH = 1)$ is evaluated in the observation model Figure 6a. *If we see a player shooting, we can infer that they are healthy.* Therefore the player scores if and only if the goalie is not healthy, so the scoring probability is 1/2. The query $P(SC = 1|CG = 1, do(SH = 1))$ is evaluated in the intervention model Figure 6b. Without a link between player health and shooting, the probability of player and goalie health are both 1/2, which means that the scoring probability is 1/4.

Since both parents of the manipulated variable SH are observed, Lemma 1 implies that both the observational and interventional probabilities should be the same. The first row of Table 3 confirms that both probabilities are 1/2; see Figure 5.

In the offline model 1*b*, Player Health (PH) is *not* observed, so the shooting effect queries compared are

$$P(SC = 1|CG = 1, SH = 1) = 1/2 \text{ and } P(SC = 1|CG = 1, do(SH = 1)) = 1/4,$$

which are different according to the second row of Table 3. As shown in Figure 6, the fundamental reason is that for observational probabilities, we can infer from the given observations that $PH = 1$; formally $P(PH = 1|CG = 1, SH = 1)) = 1$. The truncation semantics removes the causal link $PH \rightarrow SC$, thereby *blocking the inference from effect to cause.* This example illustrates the general reason for the importance of observing all common causes of the behavioral agent's actions: If the common cause between action and reward is observed, it induces a non-causal correlation between action and reward, but conditioning on the common cause eliminates the non-causal correlation.

The example of Figure 6 illustrates that when actions and rewards are confounded, *conditional probabilities can give wrong estimates of the effects of interventions*, even with an infinite sample and perfect probability estimates. In a confounded setting, policy evaluation based on conditional probabilities can lead to wrong recommendations. In our sports example, the offline model suggests that shooting whenever a player is close to the goal, regardless of their (unobserved) health has a success rate of 50%. If a coach were to adopt this policy for their team, their true success rate would in fact be 25%. In this example, the *conditional probabilities lead the learning agent to overestimate the value of shooting whenever close to the goal.* Further examples of confounded policy evaluation are provided by Gasse et al. (2021).

This concludes our definition of observational and interventional probabilities given a causal Bayesian network. In the next section we review background on Markov decision processes.

3 Background: Markov Decision Processes

For temporal data the difference between causation and correlation stems from the possible presence of confounders—unobserved common causes of the agent's actions and other environment variables. The appropriate setting for studying causality in RL is therefore a setting in which parts of the environment may be unobserved, which is known as a *partially observable Markov decision process* (POMDP). Our goal in this section is review the basic theory of POMDPs. We develop the main POMDP concepts with respect to a new factored POMDP variant that facilitates applying causal concepts. The two key concepts defined in this section for use in the remainder of the paper are the following.

1. The notion of a (behavioral) policy that is executable for a given observation signal.
2. A policy evaluation Bellman equation for factored POMDPs.

Table 4: Correspondence between Causal and RL terminology.

Reinforcement Learning	**Causality**
action	treatment
reward	response
observed state O	observed co-variates $\boldsymbol{O}$
state S	co-variates $\boldsymbol{O} \cup \boldsymbol{Z}$
belief state $b(S)$	latent variable distribution $b(\boldsymbol{Z})$
belief state update $b(S\|O)$	abduction $b(\boldsymbol{Z}\|\boldsymbol{O})$
complete observability	causal sufficiency

POMDP theory and causal concepts share a common formal structure, despite differences in terminology for describing interventions and their effects. Table 4 shows translations between analogous concepts. Latent variable concepts are described in Section 6. Key differences include the following.

1. In causal models, the response is a variable to be predicted, not a reward to be maximized.
2. A reinforcement learning, the policy guides *sequential* decisions, not a single one-time treatment.
3. RL concepts are usually defined in terms of a single state s; causal concepts are defined in terms of values for a list of variables. Using the terminology of (Russell & Norvig, 2010, Ch.2.4.7), RL uses an *atomic* environment representation, whereas causal models use a *factored* representation.

Bandit problems are the RL setting for one-time decisions, so causal models are closely related to contextual bandit problems (Lattimore et al., 2016; Lee & Bareinboim, 2018). Recent work has explored dynamic treatment regimes for applications in medicine, which brings causal modeling closer to the sequential setting of RL (Zhang & Bareinboim, 2020). While RL theory and notation utilizes an atomic state representation s, factored representations are familiar in practice. For example in a grid world, a location is described as a coordinate pair (x, y). Following previous causal models for RL (Bareinboim, 2020),(Russell & Norvig, 2010, Ch.17.4.3), we use a factored representation for the state space of a POMDP, as we describe next. We begin with the definition of a Markov Decision Process (MDP), then generalize it to POMDPs.

3.1 Factored Partially Observable Markov Decision Processes

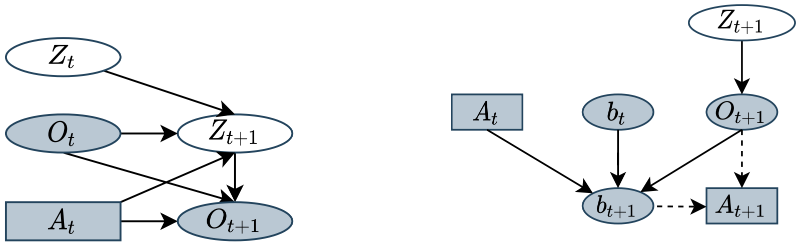
The state space of a **factored POMDP** is defined by a finite set of **state variables** $\boldsymbol{S}$. An environment **state** is an assignment $\boldsymbol{s}$ to the state variables. The state variables are partitioned as

$$\boldsymbol{S} = \boldsymbol{O} \cup \boldsymbol{Z}$$

where $\boldsymbol{O}$ represents the set of observable state variables, also called **the observation signal**, and $\boldsymbol{Z}$ represents the latent state variables, which we sometimes refer to simply as the latent state. If $\boldsymbol{Z}$ is empty, the state is **completely observable**; otherwise it is **partially observable**. Since a pair $(\boldsymbol{z}, \boldsymbol{o}) \equiv \boldsymbol{s}$ describes a state, we freely apply MDP notation to both $\boldsymbol{s}$ and pairs $(\boldsymbol{z}, \boldsymbol{o})$. In addition, we have an **action variable** A ranging over a finite set of actions available to the agent, and a real-valued **reward variable** R.

A POMDP comprises the following components.

- An initial state distribution $P_E(\boldsymbol{s}_0)$, which can be factored into a distribution over latent and observed variables:
$$P_E(\boldsymbol{s}_0) \equiv P_E(\boldsymbol{z}_0, \boldsymbol{o}_0) = P_E(\boldsymbol{z}_0) \times P_E(\boldsymbol{o}_0|\boldsymbol{z}_0)$$
where $P_E(\boldsymbol{z}_0)$ is an *initial latent distribution* and $P_E(\boldsymbol{o}_0|\boldsymbol{z}_0)$ is the *initial observation model.*
- A *transition model*, which factors into a *latent update model* and a *dynamic observation model* (see Figure 7a):
$$P_E(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t) \equiv P_E(\boldsymbol{z}_{t+1}, \boldsymbol{o}_{t+1}|\boldsymbol{s}_t, a_t) = P_E(\boldsymbol{z}_{t+1}|\boldsymbol{s}_t, a_t) \times P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1}, \boldsymbol{o}_t, a_t).$$
We assume that the current observations are Markovian in that they depend only on the current latent state, and the most recent observation and action:
$$P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1}, \boldsymbol{o}_t, a_t) = P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1}, \boldsymbol{o}_{\leqslant t}, a_{\leqslant t})$$



(a) Transition model in factored POMDP.

(b) Transition model and executable policy in belief MDP.

Figure 7: Generic causal graphs for POMDPs. Dashed lines represent a generic policy. Figure 7a: The environment transitions to a new latent state and generates an observation signal based on the current state and most recent action. Figure 7b: The agent selects an action through their policy based on their current belief state b. Their updated belief state depends on the previous belief state, previous action and new observation.

3.2 POMDP Example.

In the example of Figure 1, the action and reward variables are $A = SH$ and $R = SC$. The state variables are $\boldsymbol{S} = \{CG, GH, PH\}$. There are therefore 2^3 environment states. For example, the assignment $\boldsymbol{S} =$

$\boldsymbol{s} \equiv \langle CG = 1, GH = 1, PH = 1\rangle$ represents the state where all variables are true. The *reward model* $P_E((SH|\boldsymbol{S}, A))$ was defined by the CBN of Section 2.2. Informally, a player scores if and only if they shoot, are healthy, close to the goal, and the goalie is not healthy. Similarly, the *initial state distribution* is given by the CBN distribution over the $\boldsymbol{X}$ variables. For instance, for the initial state $\boldsymbol{s}_0 = (PH = 1, CG = 1, GH = \mathit{1})$, we have $P_E(\boldsymbol{s_0}) = 1/2 \cdot 1/2 \cdot 1/2 = 1/8$.

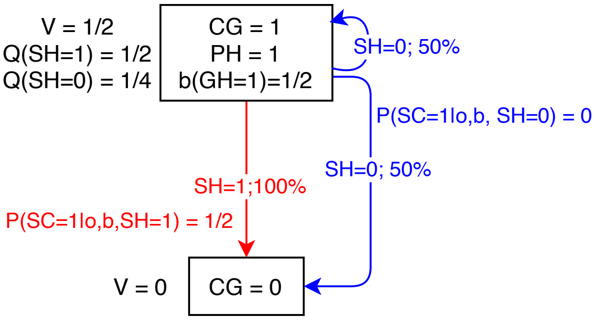


Figure 8: State transitions and policy evaluation in sports example Figure 1a. The policy evaluated is the standard behavioral policy of Equation (3). An epistemic state comprises values for the observable variables CG and PH and a belief over the values of the latent variable GH. We use state abstraction so that for example the abstract state labelled $CG = 0$ represents all states where the attacking team is not close to the goal. The policy evaluated is the standard behavioral policy of Equation (3), which chooses to shoot if and only if the player is healthy and close to the goal. The diagram shows the V value and Q action values for the epistemic state where the agent is observed to be healthy and close to the goal, and their belief is uniform over the latent variable GH. Transitions are labelled with probabilities. State-action pairs are annotated with expected rewards. The discount factor $\gamma = 1$.

Figure 8 illustrates the *state transition model*. We treat the player and goalie health as persistent time-invariant features (Lu et al., 2018): $P_E(PH' = PH) = 1$ and $P_E(GH' = GH) = 1$.

For the Close-to-Goal state variable, we adopt the following transition model

$$P(CG' = 0|SH, CG) = \begin{cases} 1, & \text{if } CG = 0 \text{ or } SH = 1 \\ 1/2, & \text{if } CG = 1 \text{ and } SH = 0 \end{cases}$$

This model can be interpreted as follows. (1) We make the simplifying assumption that the attacking team stays close to the goal if and only if they maintain possession. Any state with $CG = 0$ is therefore an absorbing state: $P_E(CG' = 0|CG = 0) = 1$. (2) If the attacking team is close to the goal, our scenario works as follows.

- If the player shoots, the team loses possession, either because they scored, or because the shot was blocked and the defending team took over (no rebounds): $P_E(CG' = 0|SH = 1) = 1.$).
- If the player does not shoot (e.g., they pass instead), there is a 50% change that the attacking team retains possession.

Same Environment with Different Observation Signals. In the online model of Figure 1a, the observable state variables are $\boldsymbol{O} = \{CG, PH\}$. Therefore the assignment $(\boldsymbol{O} = \boldsymbol{o}) = \langle CG = 1, PH = 1\rangle$ is the observation signal $\boldsymbol{o}$ received by the learning agent in the state $\boldsymbol{s}$ where all variables are true. In the offline model of Figure 1b, the observation signal does not include the player health. Thus $\boldsymbol{O} = \{CG\}$ and $\boldsymbol{o} = \langle CG = 1\rangle$ is the observation signal received by the learning agent in the state $\boldsymbol{s}$ where all variables are true. The difference illustrates the fundamental fact, highlighted by Zhang et al. (2020); Zhang & Bareinboim (2016); Bruns-Smith (2021), that *different agents can receive different observation signals in the same environment state.* A restriction on the agent's perceptual abilities translates into a restriction on the space of policies that are executable for them.

3.3 Factored Belief MDPs and Executable Policies

To evaluate and optimize a policy π under partial observability, the most common approach is to transform the POMDP into a belief MDP. We describe this transformation for a factored POMDP.

Even if the environment dynamics is Markovian in the state space, it may not be Markovian in observation space, because past observations can and typically do carry information about the current latent state. In order to apply MDP techniques to a POMDP, it is common to transform the POMD into an equivalent MDP whose states represent the agent's current beliefs. A **belief state** is a distribution $b(\boldsymbol{Z})$ over the latent environment state. The basic idea is to transform a POMDP into an MDP by replacing latent states with the agent's *beliefs* about latent states. As (Russell & Norvig, 2010, Ch.17.4.1.) write in their standard textbook: "The fundamental insight...is this: *the optimal action depends only on the agent's current belief state*" (emphasis Russell & Norvig). In a factored POMDP, observations are separate from the latent space, and the agent's decisions depend not only on their current belief state, but also on their current observation. Accordingly, an **epistemic state** $\langle \boldsymbol{o}, b\rangle$ comprises a current observation $\boldsymbol{o}$ and a current belief state. We use the tuple notation $\langle\rangle$ to make longer formulas easier to parse, and to emphasize the analogy between epistemic states and MDP states in a traditional atomic representation. A **executable policy** maps an epistemic state to a distribution over actions:

$$\pi : \boldsymbol{O}^{\pi} \times B^{\pi} \rightarrow \Delta(A)$$

where $\boldsymbol{O}^{\pi}$ represents the **observation space** of the agent executing policy π and B^{π} the space of belief states, i.e. distributions $\Delta(\boldsymbol{Z}^{\pi})$ over the unobserved variables. We also use the conditional probability notation $\pi(a|\langle \boldsymbol{o}, b\rangle)$.

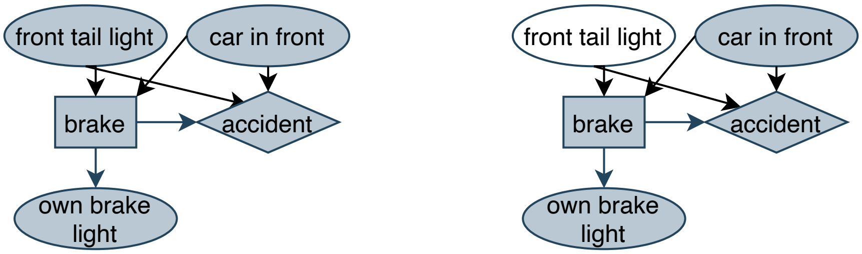
Since we can view belief states as summarizing the observation-action history (as in (Hausknecht & Stone, 2015; Liu et al., 2020; Warrington et al., 2021)), much as hidden states summarize past sequences in a recurrent neural network (Figure 7b), our notion of an executable policy is essentially equivalent to Russell and Norvig's concept of an *agent function*, which maps a history of the agent's percepts and actions to a distribution over current actions (Russell & Norvig, 2010, Ch.2.1). Executability is also equivalent to the act-state independence condition discussed by Gasse et al. (2021).

Examples. Because of the importance of the executability concept in the remainder of our analysis, we give several exampes. In general any behavioral policy that represents *online exploration* by the learning agent themselves is executable. For example, any strategy for exploring video game states is executable by the learning agent as long it is based on the same video game frames. The following three examples of non-executable behavioral policies include our running sports example and two examples from the literature.

Sports Example. In our running example, the behavioral policy Equation (3) of shooting if and only if the player is close to the goal and healthy is executable in the online model of Figure 1a with Player Health being observable, but not executable in the offline model of Figure 1b, where player health is not observable.

Driving Scenario. (Zhang et al., 2020) consider a self-driving car scenario similar to that illustrated in Figure 9. In the online setting, the agent learns by driving themselves; their observation signal includes the tail light of the car in front of them. In the offline setting, the agent learns from a dataset collected by drone surveillance, which does not include the front tail lights. In the driving example of (Zhang et al., 2020), the expert's behavioral policy may include a rule such as "brake if I see a tail light in front of me". If the learning agent does not observe the tail lights, this policy cannot be executed by them.

The Frozen Lake. Warrington et al. (2021) describe imitation learning in a frozen lake scenario where the expert has access to image data that show the locations of weak ice, but the trainee does not. In their simulations, the expert agent easily learns to cross the lake safely by taking into account where the ice is weak. Lacking this information, the expert's strategy is not executable by the trainee.



(a) Online view (b) Offline view

Figure 9: An illustration of different observation signals in the same environment, based on an example by Zhang et al. (2020). Figure 9a: In the online setting, the driving agent learns about traffic by driving themselves. Their braking decisions are influenced by observing the taillight of the car in front of them. Figure 9b: In the offline setting, the driving agent learns from a driving demonstration dataset. The dataset includes drone data specifying the location of other cards, but not the status of their tail lights.

3.4 Policy Evaluation for Belief MDPs

The **policy evaluation task** is to compute the value function for a **policy** π in a given environment. In a typical RL setting, we evaluate a policy π learned from data generated by a **behavioral** policy π_β; see Figure 3.[3] The Bellman equation to evaluate a policy based on epistemic states is as follows.

$$Q^\pi(\langle \boldsymbol{o}, b\rangle, a) = R(\langle \boldsymbol{o}, b\rangle, a) + \gamma \sum_{\boldsymbol{o}'} P_E(\boldsymbol{o}'|\langle \boldsymbol{o}, b\rangle, a) V^\pi(\boldsymbol{o}', b') \tag{5}$$

$$V^\pi(\langle \boldsymbol{o}, b\rangle) = \sum_a \pi(a|\langle \boldsymbol{o}, b\rangle) Q(\langle \boldsymbol{o}, b\rangle, a)$$

$$R(\langle \boldsymbol{o}, b\rangle, a) = E_{\boldsymbol{z}\sim b(\boldsymbol{z})} R(\boldsymbol{z}, \boldsymbol{o}, a) \tag{6}$$

$$R(\boldsymbol{z}, \boldsymbol{o}, a) = \sum_r P_E(r|\boldsymbol{z}, \boldsymbol{o}, a)$$

$$P(\boldsymbol{o}'|\langle \boldsymbol{o}, b\rangle, a) = \sum_{\boldsymbol{z}'} P_E(\boldsymbol{o}'|\boldsymbol{z}', \boldsymbol{o}, a) E_{\boldsymbol{z}\sim b(\boldsymbol{z})}[P_E(\boldsymbol{z}'|\boldsymbol{z}, \boldsymbol{o}, a)] \tag{7}$$

where $\gamma \in (0, 1]$ is a discount factor and the P_E terms are defined by the POMDP (Section 3.1).

According to recurrent Equation (5), given a new observation $\boldsymbol{o}'$ and the current action a, the expected policy value can be computed in two steps:

Posterior Update Compute the new belief state b' by conditioning the current beliefs b on observations $\boldsymbol{o}, \boldsymbol{o}'$ and action a. The posterior belief b' can be computed by the well-known belief update formula, which we derive for a factored POMDP in Appendix C (Equation (23)).

Prediction Estimate the expected return $V^\pi(\boldsymbol{o}', b')$ given the new observation $\boldsymbol{o}'$ and new belief state b'.

Updating a posterior to predict the outcome of an action a is a key part of the formal semantics of counterfactuals that we present in Section 6. Figure 8 illustrates how the Bellman equation can be used to evaluate our behavioral policy in the online POMDP setting where Goalie Health is not observable.

---

[3] The learned policy is called the estimation policy in (Sutton & Barto, 1998, Ch.5.6), the target policy by Wan et al. (2021), and the evaluation policy by Bruns-Smith (2021).

This completes our discussion of executable behavioral policies and policy evaluation in a factored POMDP. Next we describe a causal model for factored POMDPs.

4 Dynamic Decision Networks for POMDPS

This section describes a causal Bayesian network for POMDPs. We show that the concept of an executable behavioral policy can be represented in a causal graph as the condition that the causes of the behavioral agent's decisions are observable.

Following (Russell & Norvig, 2010, Ch.17) and Cooper & Yoo (1999); Boutilier et al. (1999), we utilize a *dynamic causal Bayesian network*. Dynamic Bayesian networks extend BNs to temporal data. The basic idea is to make a copy $\boldsymbol{X}'$ for the random variables in the BN, to represent successor variables. The dynamic BN is then a BN over the current and successor variables (i.e., over $\boldsymbol{X} \cup \boldsymbol{X}'$), such that there are no edges from the successor variables to the current variables. A dynamic BN satisfies the Markov condition in that the successor variables depend only on variables at a previous time. Adding action and reward variables to a dynamic CBN defines a *dynamic influence diagram*, a widely adopted graphical formalism for Markov decision processes (Polich & Gmytrasiewicz, 2007). We follow Russell & Norvig (2010) and use the term **dynamic decision network** (DDN) instead of the term dynamic influence diagram. Their terminology emphasizes that a DDN is special kind of CBN, so the concepts and results of Section 2.1 apply.

Definition 1. *A* ***dynamic decision network*** *(DDN) D for state variables $\boldsymbol{S}$ comprises the following random variables.*

1. *Current time slice:* $\boldsymbol{X} = \boldsymbol{S} \cup \{A\} \cup \{R\} \cup \{B\}$
2. *Next time slice:* $\boldsymbol{X}' = \boldsymbol{S}' \cup \{A'\} \cup \{R'\} \cup \{B'\}$.

where B resp. B' represent the agent's current resp. successor beliefs.

A DDN D satisfies the following causal assumptions.

1. *There are no edges from $\boldsymbol{X}'$ to $\boldsymbol{X}$.*
2. *There are no edges from R to nodes in $\boldsymbol{X}$, and no edges from R' to nodes in $\boldsymbol{X}'$.*
3. *There are no edges from A to nodes in $\boldsymbol{X} - \{R\}$, and no edges from A' to nodes in $\boldsymbol{X}' - \{R'\}$.*

These assumptions state that (1) causal relationships respect the temporal ordering, (2) rewards may causally depend on the current state and action, but not vice versa, (3) actions may causally depend on the current state, but not vice versa, and not on the current reward. Figure 7 illustrates these assumptions using generic causal graphs. The driving example in Figure 9 does not satisfy Assumption 3, because the state variable "own brake light" causally depends on the action of braking. De Haan et al. (2019) argue that this is possible when the temporal resolution of events is low enough that "braking" and "brake light" are assigned the same discrete time index. In that case conditioning on the state signal entails conditioning on an effect of the action, leading to "causal confusion" (De Haan et al., 2019). We discuss this scenario further under related work. Our main conclusions do not depend on Assumption 3, but we use it to simplify formal arguments.

For simplicity, in our eaxmples, we consider only Markovian behavioral policies, where the behavioral agent's decision A depends only on their current observations, not on their current beliefs B. Our general conclusions do not depend on this assumption. (Russell & Norvig, 2010, Ch.17.4) provide a worked out example of a DDN with belief states. We next give an extended example of a DDN for our sports scenario.

4.1 Dynamic Decision Network Example

A fully specified influence diagram, or DDN, defines a joint distribution $P^D(\boldsymbol{X}, \boldsymbol{X}')$ over both time slices. Therefore *a DDN specifies both an environment MDP and a behavioral policy.* We illustrate this in the example DNN structure of Figure 10. As with any CBN, the parameters of a DDN are conditional probabilities

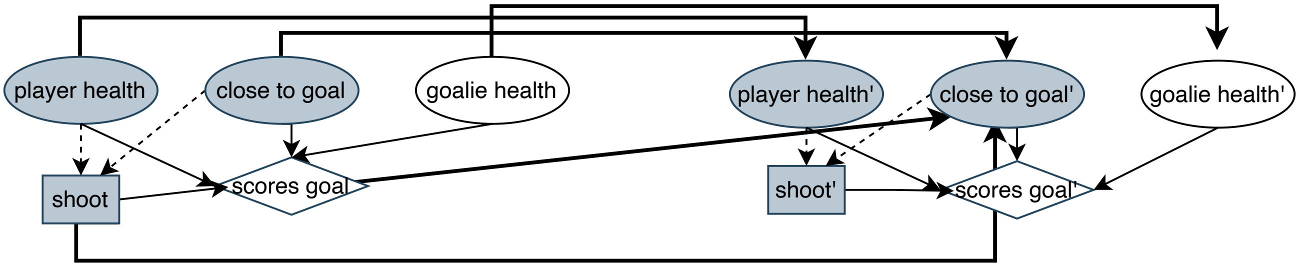


Figure 10: A dynamic decision network graph for our sports scenario. The reward model is indicated by thin lines, and the state transition model by thick lines. Dashed lines represent the agent's policy. (The agent's belief state is not shown, see text).

of the form $P(x_i|pa_i)$. It is straightforward to parametrize the DDN graph to match the POMDP model of Section 3. For example, the *reward model* $P^D(SH|\boldsymbol{S}, A)$ was defined in Section 2.2.

For the *state transition model*, we have as in Section 3.1

$$P^D(PH' = PH) = 1 \text{ and } P^D(GH' = GH) = 1$$

$$P^D(CG' = 0|SH, CG) = \begin{cases} 1, & \text{if } CG = 0 \text{ or } SH = 1 \\ 1/2, & \text{if } CG = 1 \text{ and } SH = 0 \end{cases}$$

The **network policy** π^D is defined by the conditional distributions $\pi^D(A|Pa_A)$ and $\pi^D(A'|Pa_{A'})$. In our example, we have

$$P^D(SH = 1|CG, PH) = \begin{cases} 1, & \text{if } CG = 1 \text{ and } PH = 1 \\ 0, & \text{otherwise} \end{cases}$$

4.2 Policy Executability and Action Sufficiency

A DDN represents the learning agent's current model of the transition data, which may or may not be accurate. We say that a DDN D **matches an environment model** P_E if the network reward/state transition/initial models agree with P_E. For example, matching requires that $P^D(R|\boldsymbol{S}_0) = P_E(R|\boldsymbol{S}_0)$ for every reward value R and initial state value $\boldsymbol{S}_0$. A DDN policy π^D **matches a behavioral policy** π_β : $\boldsymbol{S} \to \Delta(A)$ if $\pi(A|\boldsymbol{S}) = P^D(A|\boldsymbol{S} \cap Pa_A)$ and $\pi(A'|\boldsymbol{S'}) = P^D(A'|\boldsymbol{S'} \cap Pa_{A'})$.

Intuitively, a network policy is executable if all causes of the actions are observable, which means that the network is action sufficient. Extending the definition of action sufficiency from Lemma 1 to DDNs, we say that a set $\boldsymbol{O}$ of observable state variables is **action sufficient** in a DDN D if it includes all causes of decisions other than the agent's beliefs; that is, $Pa_A \subseteq \boldsymbol{O}$ and $Pa_{A'} \subseteq \boldsymbol{O'}$. The next observation states that a DDN satisfies action sufficiency if and only if actions are independent of latent variables. It requires a minor technical definition that rules out redundant parents: We say that a CBN is *action-minimal* if for every parent X of action variable A and every set $\boldsymbol{U}$ of variables disjoint from A and X, we have $P(A|X, \boldsymbol{U}) \neq P(A|\boldsymbol{U})$. That is, there is no variable set $\boldsymbol{U}$ such that conditioning on $\boldsymbol{U}$ makes X independent of its child A. Local minimality is entailed by the well-known stability/faithfulness conditions (Pearl, 2000, Ch.2.4).

Observation 1. *Let D be a locally minimal dynamic decision network. Then a set $\boldsymbol{O}$ of observable variables is action sufficient in D if and only if the network policy π^D is executable given $\boldsymbol{O}$.*

Proof. ($\Rightarrow$): Suppose that every parent of A is observed (i.e., $Pa_A \subseteq \boldsymbol{O}$). By Definition 1(3), the only potential descendant of A, except for successor variables, is the reward variable R. Thus the set of observed variables $\boldsymbol{O}$ contains no descendant of A. By the Markov condition, A is independent of all non-descendants given the parents of A. So given $\boldsymbol{O}$, A is independent of all contemporaneous latent environment variables $\boldsymbol{Z}$, which is the definition of an executable policy in Section 3.1. The same argument applies to the successor action variable A'.

($\Leftarrow$): Suppose that A is independent of the latent variables $\boldsymbol{Z}$ given the observed variables $\boldsymbol{O}$. Then action-minimality requires that no latent variable is a parent of A, which is the definition of action sufficiency. ☐

The upshot is that an *executable policy can be represented (matched) by an action sufficient DDN, where all parents of the action variable are observable.* This concludes our description of a dynamic causal model for POMDPs, and a causal graph formulation of the concept of an executable behavioral policy.

5 Policy Evaluation, Interventional Probabilities, and Dynamic Decision Networks

A DDN model of the behavioral policy can be used to evaluate another executable policy, that is, for *model-based off-policy evaluation.* Our goal in this section is to show how policy values can be computed based either on conditional probabilities or on interventional probabilities. We demonstrate that conditional and interventional policy values coincide under executability, but can differ when the behavioral policy is not executable by the learning agent.

Interventional policy evaluation views a policy as selecting an intervention $do(\hat{a})$, not selecting an action a (Wang et al., 2021; Zhang & Bareinboim, 2020)). The fundamental question of interventional policy evaluation is when interventional Q-values can be inferred from conditional Q-values. In this section we prove that *they are equivalent for any action sufficient dynamic influence model.* Since action sufficiency is equivalent to the executability of the behavioral policy (Observation 1) this result implies that *under executability, policy values based on conditional and interventional probabilities coincide.*

Our first result asserts the equivalence for the reward and transition models under executability.

Proposition 1. *Suppose that an observation signal $\boldsymbol{O}$ is action sufficient in a dynamic decision network D. Then $P^D(R|\boldsymbol{O}, \hat{A}) = P^D(R|\boldsymbol{O}, do(\hat{A}))$ and $P^D(\boldsymbol{S}'|\boldsymbol{O}, \hat{A}) = P^D(\boldsymbol{S}'|\boldsymbol{O}, do(\hat{A}))$. That is, the conditional and interventional reward probabilities are the same, as are the conditional and interventional state transition probabilities.*

Proof. Follows immediately from Lemma 2. ☐

Table 3 illustrates how conditional and interventional reward probabilities are the same for the action sufficient model of Figure 1a and different from the action insufficient model of Figure 1b.

5.1 Conditional and Interventional Policy Evaluation

Since a DDN defines an environment process, we can use it to perform *model-based evaluation of a policy*, by computing the required reward and transition probabilities from the DDN. The resulting **DDN Bellman equation** is derived from Equation (5) as follows.

$$\begin{aligned} Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, a) &= R^D(\langle \boldsymbol{o}, b\rangle, a) + \gamma \sum_{\boldsymbol{o}'} P^D(\boldsymbol{o}'|\langle \boldsymbol{o}, b\rangle, a) V^{\pi,D}(\boldsymbol{o}', b') \\ V^{\pi,D}(\langle \boldsymbol{o}, b\rangle) &= \sum_a \pi(a|\langle \boldsymbol{o}, b\rangle) Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, a) \\ R^D(\langle \boldsymbol{o}, b\rangle, a) &= \sum_r r \cdot P^D(R = r|\boldsymbol{o}, b, a) \end{aligned} \tag{8}$$

Since a DDN defines a joint distribution over both current and successor states, the required conditional probabilities are also specified by the model.

The **Bellman equation for the interventional Q-function** is obtained by replacing in the observational Bellman equation conditioning on an action a by conditioning on the intervention $do(\hat{a})$ (cf. Wang et al. (2021); Zhang & Bareinboim (2020)). Thus the interventional version of the DDN Equation (8) is as follows:

$$\begin{aligned}
Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, do(\hat{a})) &= R^{D}(\langle \boldsymbol{o}, b\rangle, do(\hat{a})) + \gamma \sum_{\boldsymbol{o}'} P^{D}(\boldsymbol{o}'|\langle \boldsymbol{o}, b\rangle, do(\hat{a}))V^{\pi,D}(\boldsymbol{o}', b') \\
V^{\pi,D}(\langle \boldsymbol{o}, b\rangle) &= \sum_{a} \pi(a|\langle \boldsymbol{o}, b\rangle)Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, do(\hat{a})) \\
R^{D}(\langle \boldsymbol{o}, b\rangle, do(\hat{a})) &= E_{\boldsymbol{z}\sim b(\boldsymbol{z}|do(\hat{a}))} \sum_{r} r \cdot P^{D}(R = r|\boldsymbol{z}, \boldsymbol{o}, do(\hat{a}))
\end{aligned} \tag{9}$$

The next proposition asserts that for action sufficient models, both conditional and interventional action value functions are the same.

Proposition 2. *Suppose that set of variables $\boldsymbol{O}$ is action sufficient in a dynamic decision network D. Then for any policy π with observation signal $\boldsymbol{O}$, the observational Q-value equals the interventional Q-value for every epistemic state and action:*

$$Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, \hat{a}) = Q^{\pi,D}(\langle \boldsymbol{o}, b\rangle, do(\hat{a})).$$

Proof. The basic insight is that conditional and interventional probabilities agree on rewards and transitions by Proposition 1. The resulting value functions then agree as well because they are defined recursively by reward and transition probabilities. Formally, consider

$$E_{\boldsymbol{z}\sim b(\boldsymbol{z})} R(\boldsymbol{z}, \boldsymbol{o}, do(\hat{a})) = \sum_{r} r \cdot P^{D}(r|\boldsymbol{o}, do(\hat{a})) = \sum_{r} r \cdot P^{D}(r|\boldsymbol{o}, \hat{a}) = E_{\boldsymbol{z}\sim b(\boldsymbol{z})} R(\boldsymbol{z}, \boldsymbol{o}, \hat{a})$$

where the penultimate equality follows from Proposition 1. The same proposition implies that the state transition probabilities are the same for conditional and interventional probabilities. Since reward and transition probabilities are the same, so are the value functions. ☐

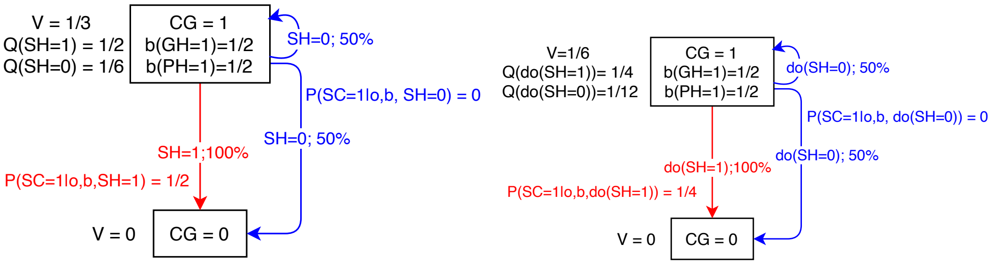
Although Proposition 2 is stated in terms of a causal model, it does *not* assume that the learning agent is given a true causal model of the behavioral policy and the environmental process. Rather, the import is that as long as the learning agent's observation signal includes the causes of the behavioral agent's actions, conditional probabilities inferred from the action data are equivalent to interventional probabilities, no matter what the true dynamic causal model is.

5.2 Policy Evaluation Examples

We give worked-out computations to illustrate how a causal model can be used for off-policy policy evaluation, both conditional and interventional.

Policy Evaluation with Executability Figure 8 shows the evaluation of our standard behavioral policy (Equation (3)) in the sports DDN, which is executable in the online setting. To illustrate the proposition for the standard behavioral policy π, we have from Figure 8 that $Q^{\pi,D}(CG = 1, PH = 1, b, SH = 1) = 1/2$; we leave it to the reader to verify that also $Q^{\pi,D}(CG = 1, PH = 1, b, do(SH = 1)) = 1/2$.

Policy Evaluation with Non-Executability To illustrate the DDN Bellman equation in the offline model of Figure 1b, note that our standard behavioral policy is not executable in this model. Instead we use Equation (8) to evaluate the **marginal** policy μ derived from the behavioral policy by averaging over latent states. The marginal policy is an important concept in offline policy evaluation (Kausik et al., 2024; Bruns-Smith, 2021). It can be viewed as a baseline form of behavioral cloning where we estimate the agent's action probability from frequencies based on the observation signals. In our example, if the player is close



(a) Using conditional probabilities.

(b) Using interventional probabilities.

Figure 11: Policy evaluation in the offline setting of Figure 1b. An epistemic state comprises values for the observable variable CG and a belief over the values of the latent variables GH and PH. The policy evaluated is the marginal behavioral policy of Equation (10), which chooses to shoot with probability 1/2 if the player is close to the goal. The evaluation uses conditional resp. interventional reward and transition probabilities derived from the DDN of Section 4.1. The diagram shows the V value and Q action values for the epistemic state where the agent is observed to be close to the goal, and their belief is uniform over the latent variables. Transitions are labelled with probabilities. State-action pairs are annotated with expected rewards. Figure 11b: Policy evaluation using *conditional probabilities* as in Equation (8). Figure 11a: Policy evaluation using *interventional probabilities* as in Equation (9) .

to the goal and we do not know their health status, the probability that they take a shot is 1/2. So we have the following standard marginal policy for our sports example:

$$P(SH = 1|CG) = \begin{cases} 1/2, & \text{if } CG = 1 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Figure 11 gives the Q and V values for the marginal policy Equation (10) based on (i) conditional probabilities (Figure 11a) and ii) based on interventional probabilities (Figure 11b). Due to non-executability, *they are different for the epistemic state where* $CG = 1$. For conditional probabilities, the epistemic state value V satisfies the equation $V = 1/4 + 1/4V$, so $V = 1/3$. For interventional probabilities, the epistemic state value V satisfies the equation $V = 1/2 \cdot 1/4 + 1/4V$, so $V = 1/6$.

For the conditional action values, we have

$$Q^D(CG = 1, b, SH = 1) = 1/2.$$

In contrast, for the interventional action values we have

$$Q^D(CG = 1, b, do(SH = 1)) = 1/4.$$

Intuitively, the reasoning for the difference is as follows. If we *observe* that a player is close to the goal and takes a shot, the DDN model entails that they are healthy. The conditional action value of a state where the player is healthy, close to the goal, and shoots is therefore 1/2, as shown in Figure 8. However, if we *intervene* to make the player take a shot, the association between player health and shooting is broken, and the updated belief that the player is healthy remains 1/2 in the DDN model. Since the player has a chance of scoring only if they are healthy, the interventional action value is only $1/2 \cdot 1/2 = 1/4$.

5.3 Summary

Before we move to the next ladder of the causal hierarchy and consider policy evaluation with counterfactual probabilities, we summarize, informally and intuitively, our conclusions regarding the first two layers, conditional and interventional probabilities.

RL settings With Executable Behavioral Policies We investigated in which general RL settings we can expect conditional probabilities to give correct policy values (unlike in our scenario). The key factor is whether the behavioral policy is executable by the learning agent. We showed that in causal terms, executability is equivalent to the condition that the learning agent can observe the causes of the behavioral agent's actions (Observation 1). Using causal graph theory, we proved that given executability, policy values based on conditional probabilities are equivalent to policy values based on interventional probabilities (Proposition 2). As we explained in the introduction, various common RL settings satisfy executability, so we may summarize our argument as follows:

> On-policy learning OR online exploration OR complete observability ⇒ executability
> ⇒ conditional probability policy values = interventional probability policy values

where online exploration refers to any setting where the learning agent uses the behavioral policy to generate observations.

Non-Executability and Confounding: Informal Examples First our running example (Figure 11) shows how, in the presence of confounding, policy evaluation based on the traditional conditional probabilities can differ from policy evaluation based on interventional probabilities. The following fictitious scenario paraphrases the example in an informal way.

> Vancouver Canucks coach Rick Tocchet watches the Edmonton Oilers to learn from the season finalists. He notices that whenever the Oilers shoot close to the goal, they score 50% of the time. So he directs the Canucks players to shoot whenever they get close. Tocchet is disappointed to find that the Canucks score only 25% of the time. "It must be that my players are worse than theirs" he thinks. Question: Is the coach right to blame his players?

The answer is no: Because Tocchet did not observe the health of the Oilers players, he did not realize that they shoot only when they are healthy. His policy directs the Canucks players to shoot whether they are healthy or not, which leads to a lower success rate. In this example, *using conditional probabilities leads to an overestimate of the policy's value* (namely 50%). The correct estimate is given by the interventional probabilities (namely 25%).

In the frozen lake example of Warrington et al. (2021), the trainee observes that the expert never falls into the lake (since they have access to image data showing the locations of weak ice), and wrongly concludes that crossing a lake is a safe policy. In policy evaluation terms, they overestimate the value of crossing the lake compared to the policy of walking around it. In the simulations of Warrington et al. (2021), this policy mis-estimation leads to a catastrophically poor return for a naive imitation learning algorithm that ignores confounding.

6 Structural Causal Models and Counterfactuals

Table 5: Two counterfactual probabilities relevant to RL. The random variable R' denotes the actual reward received, whereas $R'_{\hat{A}}$ denotes the potential reward following the decision to perform action $\hat{A}$.

Notation	Reading	Query Type
$P(R'_{\hat{A}}\|\boldsymbol{S}, A)$ or $P(R'\|\boldsymbol{S}, A, do(\hat{A})$	What is the reward if we were to choose $\hat{A}$ instead of A?	what-if
$P(R'_{\hat{A}}\|R', \boldsymbol{S}, A))$	If we receive reward R' after choosing action A, what is the reward after choosing $\hat{A}$ instead?	hindsight

We return to counterfactuals, the most advanced kind of causal reasoning according to Pearl. Our goal in this section is to define the class of structural causal models, which give a precise semantics for counterfactual queries. Table 5 gives two generic examples of counterfactual reward queries relevant to RL. In a *hindsight query*, we observe an outcome, and ask what the *potential outcome* is from an alternative course of action. In a *what-if query*, we do not observe the actual outcome, and ask how an alternative course of action might change what outcome is likely. What-if queries are closely related to policy optimization: Suppose that an agent's current policy recommends action a in state s (i.e., $\pi(s) = a$). To ascertain whether an alternative action a' would improve the agent's current policy, we can ask what the reward distribution would be if the agent chose a' instead in state s. We show that under action sufficiency, what-if counterfactuals are equivalent to conditional probabilities. Thus in the online setting, what-if queries can be answered using conditional probabilities, as is done in traditional RL.

While causal theory has focused on hindsight counterfactuals for evaluating potential outcomes as the most complex type of counterfactual query (Pearl, 2000), hindsight counterfactuals have not played a major role in reinforcement learning. One reason for this is that traditional reinforcement learning is based on conditional probabilities, and hindsight counterfactuals cannot be reduced to conditional probabilities, even in online learning, as we show in Section 7.2 below. However, the seminal work on hindsight experience replay (Andrychowicz et al., 2017) demonstrated that information about observed outcomes can be leveraged to improve RL. Recent work has proposed augmenting transition data with virtual experiences derived from hindsight counterfactuals (Sun et al., 2024). We discuss use cases for hindsight counterfactuals further in our related work Section 8.

In this section we introduce the formal semantics of counterfactual reasoning. We show how a *structural causal model* (SCM) can be used to evaluate observational, interventional, and counterfactual probabilities from a single model. SCMs combine a causal graph with local functions that map parent values to child values. They achieve greater expressive power than causal Bayesian networks by including latent variables as parents. The posterior over latent variables carries over information from observed outcomes to infer potential outcomes.

6.1 Structural Causal Models

Structural causal models (SCMs) combine a causal graph with latent variables and deterministic local functions that map parent values to child values (Pearl, 2000). While the use of latent variables typically makes them less interpretable than a Bayesian network with observable variables only, SCMs have two important advantages as causal models. First, the latent variables support a formal semantics for counterfactuals. Second, SCMs are compatible with deep learning, in that the local functions are essentially decoders in the sense of deep generative models. Deep generative models with latent variables are therefore a powerful architecture for implementing and learning with counterfactual reasoning (Geffner et al., 2022). For our purpose of relating causal models and reinforcement learning, SCMs contain two important elements that match standard POMDP theory: unobserved components and a distribution over latent variables, which corresponds to an agent's beliefs in POMDP theory. Our presentation of SCMs follows those of Pearl (2000) and of Schölkopf et al. (2021), and highlights connections with other generative models in machine learning.

A **structural causal model** (SCM) is a pair $\mathcal{S} = \langle G, \boldsymbol{F} \rangle$ meeting the following conditions.

- G is a DAG over random variables $\boldsymbol{X}$. Let $\boldsymbol{U}$ be the set of **source nodes** with indegree 0 in G. We write $\underline{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{U}$ for the set of non-source variables.
- $\boldsymbol{F} = \{f_1, \ldots, f_n\}$ is a set of local functions such that each f_i deterministically maps the parents of non-source variable $X_i \in \underline{\boldsymbol{X}}$ to a value of X_i. The local functions are often written in the form of a **structural equation**:
$$X_i = f_i(pa_i). \tag{11}$$

The source variables are also called *background variables* or *exogenous variables*. They can be, and often are, latent variables. For example, in a linear structural equation

$$f(X, \varepsilon) \equiv Y = aX + b + \varepsilon$$

the parent X of Y represents an observed cause and the noise term ε can be modeled as a latent parent summarizing unobserved causal influences. The set of functions $\boldsymbol{F}$ takes the place of the conditional probability parameters in a causal Bayesian network. Given an assignment $\boldsymbol{U} = \boldsymbol{u}$ of values to the source variables, we can compute values for the non-source variables $\underline{\boldsymbol{X}}$ by using the local functions to assign values to the children of the source nodes, then to the children of the children, etc. This recursive evaluation procedure defines a **solution function**

$$F^{\mathcal{S}}(\boldsymbol{u}) = \underline{\boldsymbol{x}} \tag{12}$$

where $\underline{\boldsymbol{x}}$ is an assignment of values to the non-source nodes $\underline{\boldsymbol{X}}$. A **probabilistic SCM** is a pair $C = (\mathcal{S}, b)$ where b is a joint **prior distribution** over the source variables $\boldsymbol{U}$ (Pearl, 2000, Eq.7.2). Since the source variables $\boldsymbol{U}$ typically are latent, we use the POMDP notation $b(\boldsymbol{U})$ rather than the more usual $p(\boldsymbol{U})$ to emphasize the similarity between a POMDP belief and a distribution over source variables. Like a causal Bayesian network, a probabilistic SCM C defines a **joint distribution** over the variables $\boldsymbol{X} = \boldsymbol{U} \cup \underline{\boldsymbol{X}}$:

$$P^{C}(\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}, \boldsymbol{U} = \boldsymbol{u}) = P^{C}(\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}} | \boldsymbol{U} = \boldsymbol{u}) \times b(\boldsymbol{U} = \boldsymbol{u}) \tag{13}$$

$$b(\boldsymbol{U} = \boldsymbol{u}) = \prod_{U \in \boldsymbol{U}} b(U = u) \tag{14}$$

$$P^{C}(\underline{\boldsymbol{X}} = \underline{\boldsymbol{x}} | \boldsymbol{U} = \boldsymbol{u}) = \begin{cases} 1, & \text{if } F^{\mathcal{S}}(\boldsymbol{u}) = \underline{\boldsymbol{x}} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $F^{\mathcal{S}}(\boldsymbol{u})$ is the solution function Equation (12). Equation (14) says that the unconditional prior distribution over source variables factors into individual priors over each source variable, which is true in any Bayesian network. Equation (15) applies the deterministic solution function as a deterministic decoder that maps the source variables to a unique assignment for non-source variables.

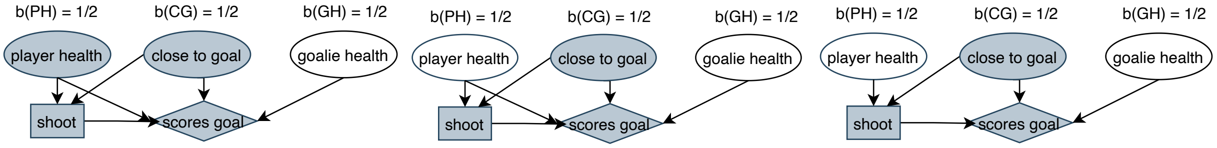
Relationship to Causal Bayesian Networks A well-known result in graphical model theory states that every Bayesian network B over n variables $\boldsymbol{X}$ can be represented by an equivalent SCM over variables $\boldsymbol{X} \cup \{U_i, i = 1, \ldots, n\}$ such that $P(x_i | pa_i) = \sum_{u_i : f_i(pa_i, u_i) = x_i} b(u_i)$ where f_i is the local SCM function, and $b(u_i)$ is the prior probability assigned to the u_i value (Druzdzel & Simon, 1993). This parametrization of Bayesian networks associates a scalar latent variable U_i with each observed variable X_i. The corresponding structural equation is $X_i = f_i(pa_i, u_i)$. The latent scalar U_i is often called the *error* or *noise* term for variable X_i, with common alternative notations being Z_i or ϵ_i. The resulting SCM is equivalent to the Bayesian network B in the sense that the joint SCM distribution defined by Equation (13) is equivalent to the joint Bayesian network distribution defined by the product formula Equation (1). SCMs and causal Bayesian networks therefore have equivalent expressive power in terms of the joint distributions that they can represent. As Figure 12 illustrates, the CBNs of Figure 1 are also SCMs because their conditional probabilities are deterministic.

Table 6: A set of structural equations for the causal graphs of Figures 12a and 12b.

Variable	Equation
Shoot(SH)	$SH = PH \cdot CG$
Scores(SC)	$SC = PH \cdot CG \cdot SH \cdot (1 - GH)$

Structural Causal Model Examples Table 6 shows structural equations for the graphs of Figures 12a and 12b, that represent the same causal mechanisms with deterministic functions that were described in Section 2.1 with conditional probabilities. For the SCM of Figure 12c, we use the same equation for SH but replace the scoring equation by

$$SC = CG \cdot SH \cdot (1 - GH)$$



(a) Online View. (b) Offline View with confounder. (c) Offline view without confounder.

Figure 12: Structural Causal Models to illustrate what-if counterfactuals in our sports scenario. Number labels specify the prior distribution over source variables. Table 6 specifies the deterministic local functions. Figure 12a: The SCM version of the online model Figure 1a. Figure 12b: The SCM version of the offline model Figure 1b. Figure 12c: An offline view where Player Health is not observed and does not affect goal scoring chances.

since in this model Player Health does not affect scoring chances. The sharing of structural equations across scenarios illustrates how they represent local independent mechanism that we can expect to be stable across different contexts and domains. This is an important advantage of causal modeling for machine learning (Schölkopf et al., 2021).

Together Figure 12a and Table 6 define a probabilistic structural causal model C. Adding a prior distribution over the binary source variables GH, PH, GH specifies a probabilistic SCM. With a uniform distribution over the source variables, the observational joint distribution assigns

$$P^C(1 = PH = CG = SH = SC, GH = 0) = 1/2 \cdot 1/2 \cdot 1 \cdot 1 \cdot 1/2 = 1/8. \tag{16}$$

which agrees with the corresponding result for the causal Bayesian network from Equation (4).

Interventional Distributions Similar to causal Bayesian networks, for a structural causal model $C = (\mathcal{S}, b)$ the distribution that results from intervening on a variable A is computed by removing the parents of A from the graph and replacing the local function f_A by a constant function. The **submodel** $\mathcal{S}_{\hat{a}} = \langle G_A, \boldsymbol{F}_{\hat{a}} \rangle$ is the causal model where G_A contains all edges in G except those pointing into variable A, and $\boldsymbol{F}_{\hat{a}} = \{f_i : X_i \neq A\} \cup \{A = \hat{a}\}$. Here $\{f_i : X_i \neq A\}$ is the set of all local functions for unmanipulated variables, and $A = \hat{a}$ is the constant function that assigns variable A its manipulated value. Similarly let $b_{\hat{a}}$ be the prior distribution over source node variables that assigns probability 1 to $\hat{a}$ and agrees with b on all other variables. Formally, $b_{\hat{a}}(A = \hat{a}) = 1$, and $b_{\hat{a}}(U = u) = b(U = u)$ for $U \neq A$. We compute the intervention distribution as the joint probability in the truncated submodel:

$$P^{(\mathcal{S},b)}_{do(A=\hat{a})}(\boldsymbol{X} = \boldsymbol{x}) = P^{(\mathcal{S}_{\hat{a}}, b_{\hat{a}})}(\boldsymbol{X} = \boldsymbol{x}) \tag{17}$$

where $P^{(\mathcal{S},b)}_{do(A=\hat{a})}$ is the interventional distribution of the causal model $(\mathcal{S}, b)$ and $P^{(\mathcal{S}_{\hat{a}}, b_{\hat{a}})}$ is the joint distribution of the truncated submodel $(\mathcal{S}_{\hat{a}}, b_{\hat{a}})$ defined by the intervention. Figure 13 below illustrates the truncation semantics. We next show how the interventional distribution can be used to define a formal semantics for counterfactuals.

6.2 Counterfactual Probabilities

A **counterfactual probability** $P(\boldsymbol{Y}_{\hat{a}} = \boldsymbol{y}' | \boldsymbol{X} = \boldsymbol{x}, A = a, \boldsymbol{Y} = \boldsymbol{y})$ can be read as follows: "Given that we observed action $A = a$, and state variables $\boldsymbol{X} = \boldsymbol{x}$, followed by outcome $\boldsymbol{Y} = \boldsymbol{y}$, what is the probability of obtaining an alternative outcome $\boldsymbol{y}'$, if we were to instead select the action $\hat{a}$ as an intervention?" Here $\boldsymbol{Y}_{\hat{a}}$ is a list of **potential outcome** random variables, distinct from the actual outcomes $\boldsymbol{Y}$, and $\boldsymbol{X}, A, \boldsymbol{Y}$ are disjoint. For a given probabilistic SCM $C = (\mathcal{S}, b)$, we can compute the counterfactual probability as follows (Pearl, 2000, Th.7.1.7).

Abduction/Posterior Update Condition on the observations $\boldsymbol{X} = \boldsymbol{x}, A = a, \boldsymbol{Y} = \boldsymbol{y}$ to compute a source variable posterior

$$b' \equiv b(\boldsymbol{U} | \boldsymbol{X} = \boldsymbol{x}, A = a, \boldsymbol{Y} = \boldsymbol{y}).$$

Intervention Apply the intervention $do(A = \hat{a})$ to compute the submodel $\mathcal{S}_{\hat{a}}$ and the SCM $C' = (\mathcal{S}_{\hat{a}}, b'_{\hat{a}})$.

Prediction Return the conditional probability $P(\boldsymbol{Y} = \boldsymbol{y}' | \boldsymbol{X} = \boldsymbol{x})$ computed in the updated SCM:

$$P^{C}(\boldsymbol{Y}_{\hat{a}} = \boldsymbol{y}' | \boldsymbol{X} = \boldsymbol{x}, A = a, \boldsymbol{Y} = \boldsymbol{y}) = P^{C'}(\boldsymbol{Y} = \boldsymbol{y}' | \boldsymbol{X} = \boldsymbol{x}) \tag{18}$$

A posterior update is often called abduction in causal modeling. Through their posterior, the source variables carry information from the observed configuration $\boldsymbol{X} = \boldsymbol{x}, A = a, \boldsymbol{Y} = y$ to the counterfactual configuration where $\boldsymbol{X} = \boldsymbol{x}, A = \hat{a}, \boldsymbol{Y} = \boldsymbol{y}'$. Counterfactual probabilities form a natural hierarchy that generalizes interventional probabilities as follows.

- We refer to the most general counterfactual of the form $P(\boldsymbol{Y}_{\hat{a}} | \boldsymbol{X}, A, \boldsymbol{Y})$ as a **hindsight** counterfactual query because it specifies the actual outcomes $\boldsymbol{Y}$.
- If the actual outcomes $\boldsymbol{Y}$ are *not* included in the evidence, we have a **what-if** counterfactual query $P(\boldsymbol{Y}_{\hat{a}} | \boldsymbol{X}, A)$ that asks what the likely outcome is after deviating from the actual choice A. For what-if queries, we use the causal effect notation $P(\boldsymbol{Y} | \boldsymbol{X}, A, do(A = \hat{a})) \equiv P(\boldsymbol{Y}_{\hat{a}} | \boldsymbol{X}, A)$.
- If neither an observed outcome nor an observed action are specified, a counterfactual probability reduces to an interventional probability $P(\boldsymbol{Y} | \boldsymbol{X}, do(A = \hat{a}))$.
- If neither an observed outcome nor an intervention are specified, a counterfactual probability reduces to a conditional probability $P(\boldsymbol{Y} | \boldsymbol{X}, A)$.

We refer to probabilities of the first three types as **causal probabilities**. The update-intervention-prediction procedure can be used for computing any causal probability.

6.3 Examples for What-if Counterfactual Probabilities.

We illustrate the computation of causal probabilities for the causal models of Figure 12. In the online model Figure 12a, the scoring probability due to an intervention not to shoot is unaffected by observing the agent shoot, and therefore equal to just the interventional scoring probability, which is 0 (see Section 2.2). Formally, we have

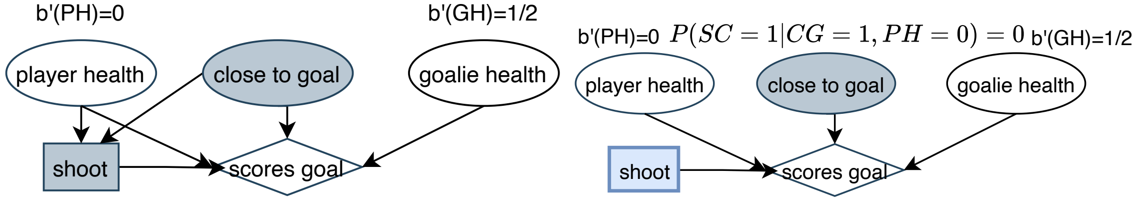
$$P(SC = 1 | CG = 1, PH = 1, SH = 1, do(SH = 0)) = P(SC = 1 | CG = 1, PH = 1, do(SH = 0)) = 0.$$

For the offline models in Figure 12, our example what-if counterfactual asks for the *probability of scoring a goal if the player were to shoot, given that they actually did not shoot.* Table 7 compares the corresponding reward probabilities given state and action information for observations, interventions, and counterfactuals. Figure 13 illustrates the observation-intervention-prediction steps for the offline models. For the confounded model of Figure 12b, the observational and interventional probabilities agree with our computations for the corresponding causal Bayesian network (see Figure 6). Because the player did not shoot, we can infer that he/she is not healthy (abduction). Since he/she is not healthy, the what-if probability of scoring is 0 given the scoring model.

In the non-confounded model Figure 12c, observational, interventional, and counterfactual probabilities are the same because the posterior over the player health variable does not affect the goal scoring probability. Figure 14 shows the computation of the what-if counterfactual. We leave the computation of the conditional and interventional probabilities as an exercise. In the next subsection we show that the equivalences of Figures 12a and 12c are an instance of a general pattern: *under action sufficiency, what-if counterfactuals are equivalent to interventional and to conditional probabilities.*

Table 7: Scoring probabilities for the reward model in the soccer examples of Figures 12b and 12c.

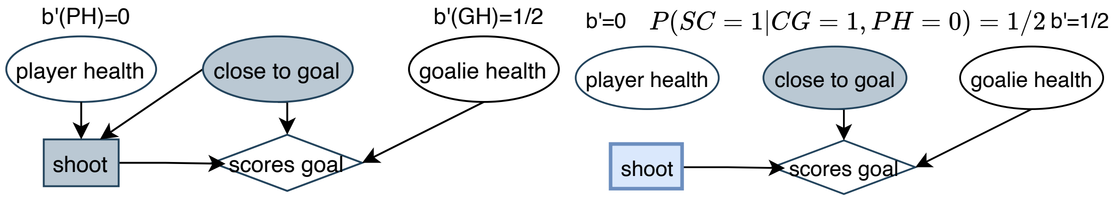
	Conditional	Intervention	What-If Counterfactual
Query	$P(SC = 1 \mid CG = 1, SH = 1)$	$P(SC = 1 \mid CG = 1, do(SH) = 1))$	$P(SC = 1 \mid CG = 1, SH = 0, do(SH) = 1))$
Model C_{12b}	1/2	1/4	0
Model C_{12c}	1/2	1/2	1/2



(a) Abduction

(b) Intervention and Prediction

Figure 13: Evaluating the what-if counterfactual query $P(SC = 1|CG = 1, SH = 0, do(SH) = 1))$ for the confounded offline model of Figure 12b. Numbers indicate posterior probabilities of latent source variables given the query observations. Figure 13a, Abduction: The posterior probability of the player being healthy is 0, given that they did not shoot. Figure 13b, Intervention and Prediction: The truncated model removes the link between Player Health and shooting and uses the posterior distribution over source variables. In the truncated model, the scoring probability is 0, given that we have inferred that the player is not healthy.



(a) Abduction

(b) Prediction

Figure 14: Evaluating the what-if counterfactual query $P(SC = 1|CG = 1, SH = 0, do(SH) = 1))$ for the unconfounded model of Figure 12c. Numbers indicate posterior probabilities of latent source variables given the query observations. Figure 14a, Abduction: The posterior probability of the player being healthy is 0, given that they did not shoot. Figure 14b, Intervention and Prediction: The truncated model removes the link between player health and shooting and uses the posterior distribution over source variables. The probability of scoring is independent of the player health, and thus equal to the probability of the goalie not being healthy, which is 1/2.

6.4 Action Sufficiency and What-if Counterfactuals

In this section we prove an analog of Lemma 2 for SCMs: Under action sufficiency, what-if, interventional, and conditional probabilities are equivalent.

The condition of action sufficiency introduced for causal Bayesian networks required that all causes of the agent's actions should be observed (Section 2.1). This definition needs to be modified for structural causal models, because non-deterministic variables require latent parents to generate variance. For example, in the linear structural equation $Y = aX + b + \varepsilon$, the noise term ε is a latent variable. For an RL example, consider a probabilistic policy $\pi(A|S)$ where S is a completely observable state. In an SCM, such a policy is represented by a structural equation $A = f(S, U)$ where U is a latent cause that generates the distribution over actions given an observed state. The causal modeling literature therefore utilizes a more general causal sufficiency condition that allows a variable to have a latent cause, but not a *shared* latent cause (Spirtes et al., 2000). The insight is that what matters for causal modeling is not the presence of latent causes, but of latent *confounders*. Formally we say that source variable U is a noise variable for X if X is the only child of U. A set of observed variables $\boldsymbol{O}$ is **causally sufficient** for variable X in a causal SCM graph G if every latent parent of X is a noise variable for X. An SCM C is **action sufficient** if its graph is causally sufficient for the action variable A.[4] The graph of Figure 12b is not action sufficient because Player Health is a latent common cause of Shooting and Scoring. The graph of Figure 12c is action sufficient because the latent cause Player Health of the shooting action is not a cause of any other variable, hence a noise variable for shooting.

The next lemma states that under action sufficiency, what-if counterfactuals reduce to conditional probabilities. The proof is in Appendix B.

Lemma 3. *Let C be a probabilistic SCM and let $\boldsymbol{Y}, A, \boldsymbol{X}$ be a disjoint set of random variables such that $\boldsymbol{X}$ includes all parents of A except for possibly a noise variable U_A of A (i.e., $\boldsymbol{X} \supseteq Pa_A - U_A$), and none of the descendants of A. Then for any actions $a, \hat{a}$ we have*

$$P^C(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, A = a, do(A = \hat{a})) = P^C(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, do(A = \hat{a})).$$

That is, the target probability given an intervention $\hat{a}$ instead of the actual action a is independent of the actual action selected.

Lemma 3 implies that under action sufficiency, a what-if counterfactual "what would happen if I selected action $\hat{a}$ instead of action a" can be evaluated by the conditional probability given that $\hat{a}$ is observed, without taking into account the actual action choice a. The next lemma states this observation formally.

Lemma 4. *Let $\boldsymbol{O} \subseteq \boldsymbol{X}$ be an action sufficient set of observable variables in a probabilistic SCM C that contains no effects (descendants) of the action variable A. Then*

$$P^C(\boldsymbol{Y}|\boldsymbol{O} = \boldsymbol{o}, A = a, do(A = \hat{a})) = P^C(\boldsymbol{Y}|\boldsymbol{O} = \boldsymbol{o}, do(A = \hat{a})).$$

for any actions $a, \hat{a}$ and any list of target outcomes $\boldsymbol{Y}$. That is, the target probability given an intervention $\hat{a}$ instead of the actual action a is independent of the actual action selected, given that the direct causes of the action have been observed.

This concludes our discussion of structural causal models and the formal semantics of counterfactual queries. We showed how under action sufficiency, interventional and what-if counterfactual probabilities are equivalent (Lemma 4). Recall from Observation 1 that the action sufficiency of a causal model is equivalent to the executability of the behavioral policy represented in the model. Thus we have shown that executability implies that interventional and what-if counterfactual probabilities are equivalent. The next section extends the equivalence to policy evaluation.

7 Counterfactuals and Policy Evaluation

The goal of this section is to investigate policy evaluation based on counterfactuals. We first consider what-if counterfactuals, then hindsight counterfactuals. We show that with an executable behavioral policy, what-if policy values are equivalent to interventional policy values, but hindsight counterfactuals are not. Proposition 2 showed that under executability, interventional policy values are equivalent to conditional policy values. Thus we conclude that *executability renders conditional, interventional, and what-if policy values equivalent.*

7.1 What-if Counterfactuals

In RL based on what-if counterfactual decisions, the learning agent seeks to learn an optimal policy that may deviate from the actual decisions taken. Following Pearl's suggestion in a similar context (Pearl, 2000, Ch.4.1.1), we refer to observed decisions by the behavioral agent as "acts", with associated random variable A and active decisions by the learning agent as "actions", with associated random variable $\hat{A}$. For what-if counterfactuals, as with interventional probabilities, we show that under action sufficiency counterfactual

---

[4] Our concept of action-sufficiency differs from the notion introduced by Huang et al. (2022). For them, a latent state is action sufficient if it is powerful enough to support an optimal policy based on the latent state space. Our concept could be called "causal action sufficiency" to disambiguate.

rewards, transitions, and value functions are equivalent to observational rewards, transitions, and value functions. As with causal Bayesian networks, probabilistic SCMs can be straightforwardly extended to a dynamic causal model by specifying them with respect to both current variables $\boldsymbol{S}, A, R$, and successor variables $\boldsymbol{S}', A', R'$. For the remainder of the paper, we use the D notation to refer to a **dynamic SCM**.

Proposition 3. *Suppose that an observation signal* $\boldsymbol{O}$ *is action sufficient in a dynamic probabilistic SCM* D*. Then* $P^D(R|\boldsymbol{O}, do(\hat{A})) = P^D(R|\boldsymbol{O}, A, do(\hat{A}))$ *and* $P^D(\boldsymbol{S}'|\boldsymbol{O}, do(\hat{A})) = P^D(\boldsymbol{S}'|\boldsymbol{O}, A, do(\hat{A}))$*. That is, the reward and state transition probabilities given an intervention* $\hat{a}$ *instead of the actual action* a *are independent of the actual action selected, given that the direct causes of the action have been observed.*

Proof. Follows immediately from Lemma 4. ☐

The importance of the proposition is that under action sufficiency, observing the acts of an agent does not impact reward and state transition probabilities. Table 7 illustrates how conditional and interventional reward probabilities are the same for the action sufficient model of Figure 12c and different from the confounded model of Figure 12b.

The next proposition asserts that for action sufficient models, both interventional and what-if action value functions are the same. We model what-if counterfactuals by including in the learning agent's observation signal the acts of the behavioral agent. In symbols, we have $\boldsymbol{O}^{\pi} = \boldsymbol{O} \cup A$ where $\boldsymbol{O}$ is the set of state variables observable by the learning agent. The **Bellman equation for the what-if counterfactual Q-function** is obtained by replacing the original observation signal with the expanded observation signal in the interventional Bellman Equation (9). The resulting equation is as follows.

$$\begin{aligned}
Q^{\pi,D}(\langle \boldsymbol{o}, a, b\rangle, do(\hat{a})) &= R^D(\langle \boldsymbol{o}, a, b\rangle, do(\hat{a})) + \gamma \sum_{\boldsymbol{o}'} P^D(\boldsymbol{o}'|\langle \boldsymbol{o}, a, b\rangle, do(\hat{a}))V^{\pi,D}(\boldsymbol{o}', a, b') \qquad (19)\\
V^{\pi,D}(\langle \boldsymbol{o}, a, b\rangle) &= \sum_{a} \pi(a|\langle \boldsymbol{o}, a, b\rangle) Q^{\pi,D}(\langle \boldsymbol{o}, a, b\rangle, do(\hat{a}))\\
R^D(\langle \boldsymbol{o}, a, b\rangle, do(\hat{a})) &= E_{\boldsymbol{z}\sim b(\boldsymbol{z}|do(\hat{a}))} \sum_{r} r \cdot P^D(R = r|\boldsymbol{z}, \boldsymbol{o}, a, do(\hat{a}))
\end{aligned}$$

Proposition 4. *Suppose that an observation signal* $\boldsymbol{O}$ *is action sufficient in a dynamic probabilistic SCM* D*. The what-if Q-value equals the interventional Q-value for every epistemic state and observed act* a*:*

$$Q^D(\langle \boldsymbol{o}, a, b\rangle, do(\hat{a})) = Q^D(\langle \boldsymbol{o}, b\rangle, do(\hat{a}))$$

That is, the Q-value associated with an intervention $\hat{a}$ *replacing the actual act* a *is independent of the act, given that the direct causes of the act have been observed.*

The proof proceeds as in Proposition 2: By Proposition 3, what-if and interventional probabilities agree on rewards and transitions by Proposition 1. The resulting value functions then agree as well because they are defined recursively by reward and transition probabilities.

Example Figure 15 shows the computation of values for the marginal policy μ, given that we observe a player not taking a shot when they are close to the goal. The confounded model of Figure 12b implies that the player is not healthy. Since they score only if they are healthy, it follows that their expected reward is 0 regardless of what action we direct them to perform. In contrast, from Figure 11b we have that conditional only on being close to the goal, the scoring chance of the marginal policy is 1/6.

The action sufficient model of Figure 12c also implies that the player is not healthy from the same observations. However, in this model player health does not affect scoring, which depends only on goalie health. This means that what-if and interventional Q-values are the same, given closeness to goal and an agent's actions, as entailed by Proposition 4.

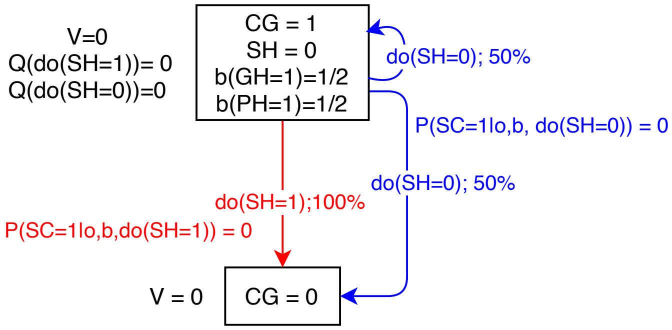


Figure 15: Off-policy policy evaluation based on what-if counterfactuals in the confounded offline model of Figure 12b. The policy evaluated is the marginal behavioral policy of Equation (10), which chooses to shoot with probability 1/2 if the player is close to the goal. The evaluation uses interventional *what-if* reward and transition probabilities derived from the DDN of Section 4.1. The diagram shows the V value and Q action values for the epistemic state where the agent is observed to be close to the goal, their belief is uniform over the latent variables, and *they do not take a shot.*

7.2 Hindsight Counterfactuals and Policy Evaluation.

A remarkable feature of counterfactual hindsight probabilities is that they can differ from conditional probabilities even in action sufficient settings, such as online learning. This subsection gives examples to illustrate the phenomenon, two examples for the reward model and one for policy evaluation. The general insight is that while it has long been noted in RL that *past* observations allow us to infer current latent states (Kaelbling et al., 1998; Hausknecht & Stone, 2015), *future* information also allows us to infer current latent states.

Suppose that we observe that in state $\boldsymbol{s}$, an act a was followed by a reward r. We can then ask "what would the reward have been if the agent had chosen the action $do(\hat{a})$" instead? The corresponding **hindsight reward probability** is given by counterfactual queries of the form $P(R_{do(\hat{a})}|\boldsymbol{O}, A, R)$.

A simple reward hindsight query in our sports example would be

$$P^D(SC_{SH=1} = 1|CG = 1, PH = 1, SH = 1, SC = 1),$$

where $SC_{SH=1} = 1$ denotes the potential outcome variable "scoring a goal after intervening to shoot".

To evaluate the hindsight reward probability in the online model of Section 4.1, first we update the initial belief given the observations:

$$b(GH = 0|CG = 1, PH = 1, SC = 1) = 1.$$

Informally, since the player scores only if the goalie is not healthy, we can infer from their scoring that the goalie is not healthy. Given that the player is healthy and the goalie is not, the player is certain to score, so the hindsight reward probability is 1:

$$P^D(SC_{SH=1} = 1|CG = 1, PH = 1, SC = 1) = P(SC = 1|CG = 1, PH = 1, GH = 0, do(SH = 1)) = 1$$

Without hindsight, the chance of scoring is only 1/2, since the goalie has a 50% chance of being healthy. A more interesting example is to consider not only immediate rewards, but hindsight based on future rewards (as in hindsight credit assignment (Harutyunyan et al., 2019)). Suppose that a player does not take a shot, and then their team scores at the next time instant. Since this implies that the goalie is not healthy, we can infer that they would have scored if they had taken a shot earlier. Using counterfactual notation, we have

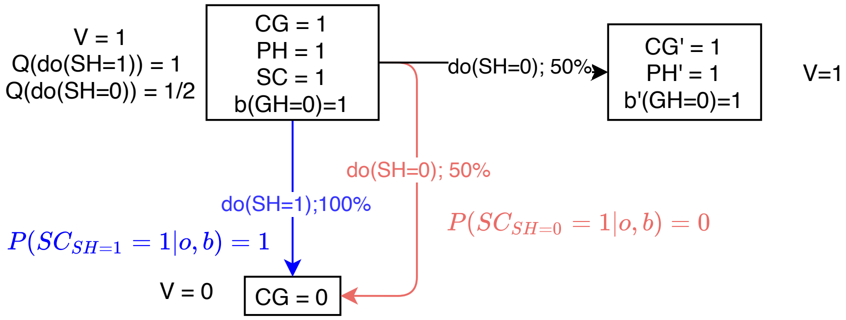


Figure 16: On-policy policy evaluation in the online setting with hindsight. The policy evaluated is the standard behavioral policy of Equation (3). The evaluation uses interventional reward and transition probabilities derived from the online DDN of of Section 4.1. The diagram shows the V value and Q action values for the epistemic state where the agent is observed to be close to the goal, their belief is uniform over the latent variables, and *the observation signal includes the current reward (goal scored).*

$$P^D(SC_{SH=1} = 1|CG = 1, PH = 1, SH = 0, SC' = 1) = P(SC = 1|CG = 1, PH = 1, GH = 0, do(SH = 1)) = 1,$$

where as before $SC_{SH=1} = 1$ denotes the potential outcome variable "scoring a goal after intervening to shoot", and $SC' = 1$ denotes that the observed outcome at the *next* time step is that the player scored.

Hindsight Q-values can be defined by including the observed outcomes as part of the agent's observation signal. Figure 16 shows how Q-values change with hindsight. If the player is made not to shoot, they have a 50% chance of maintaining possession. If they maintain possession, they will shoot at the next step according to the behavioral policy, so they are certain to score then because the goalie is not healthy. Hence their expected return after not shooting is 1/2; in symbols

$$Q^D(\langle CG = 1, PH = 1, SC = 1, b\rangle, do(SH = 0)) = 1/2.$$

These examples illustrate how observing outcomes can be a powerful source of information about the latent environment state (goalie health in our example).

8 Related Work: Current and New Research Directions

This section selectively describes current RL research involving causal models with an emphasis on the distinction between observational, interventional, and counterfactual probabilities. We describe several directions for future research from the perspective of confounding and executability. For comprehensive surveys of causal RL, please see Bareinboim (2020), (Deng et al., 2023) and (Schölkopf et al., 2021, Section E). We organize our discussion in different subsections corresponding to online RL, offline RL, and approaches that combine both online and offline learning. We focus on the causal RL tasks listed in the previous surveys.

A common goal in previous research is leveraging a given causal model. Such approaches can be categorized as *causal model-based RL*. Causal model-based RL inherits the challenges and benefits of model-based RL in general (Levine et al., 2020, Sec.5). Our discussion focuses on the following special features of causal model-based RL:

1. The ability to correctly evaluate the effects of interventions when confounders are present.
2. The greater expressive power of causal models, which define not only conditional reward-transition models, but also interventional and counterfactual reward-transition models.
3. The causal graph structure, which decomposes joint distributions into local mechanisms.

8.1 Online Causal RL

The term online RL in this section refers to single-agent online exploration, where the learning agent is the only agent executing a behavioral strategy to learn about the environment. We thus exclude asymmetric online imitation learning as considered by Warrington et al. (2021) where online exploration is combined with learning from a demonstrator. Our analysis so far implies that in single-agent online learning, we can expect conditional probabilities to be unconfounded, which reduces the importance of advantage 1. However, the last two advantages (graphical structure and greater expressive power) apply to online learning as well, and have been leveraged in previous work on online RL.

Eliminating Irrelevant Variables: Model-based State Abstraction. Lemma 1 implies that conditioning on a superset $\boldsymbol{X} \supseteq pa_A$ of observable causes (parents) of A suffices to ensure that conditional probabilities are causal. This means that the conditioning set need not include *all* observed variables, that is, it need not include the *entire* state observed by the agent. For example in the car driving model of Figure 9a, for the braking decision, it suffices to condition on *FrontTaillight* and *CarinFront*; the *OwnBrakeLightVariable* can be ignored during the decision process. In general, a state variable S is **conditionally irrelevant** to a decision if $S \perp R, \boldsymbol{S}'|A, Pa_A$. Thus a causal model supports variable selection as a form of state abstraction/simplification (Peters et al., 2017, Sec.8.2.2). The paper by Sen et al. (2017) is one of the first to leverage a given causal graph to reduce the effective state space. They prove that this reduction improves regret bounds in online bandit problems, when the bound is a function of the size of the state space. Zhang & Bareinboim (2020) provide an algorithm for reducing the state space by eliminating irrelevant variables given a causal model, which leads to substantive improvement in regret bounds. Wang et al. (2022; 2024) relate the elimination of irrelevant variables from a dynamic causal model to state abstraction.

Data Augmentation and Hindsight Counterfactuals. One of the traditional uses of models in RL, going back to the classic Dyna system (Sutton, 1990), is to augment the observed transition data with virtual experiences simulated from the model. Sun et al. (2024) utilize hindsight counterfactuals to generate virtual state transitions that specify the next state that would have occurred in the same scenario. These counterfactual state transitions take the form $P(\boldsymbol{S}'_{\hat{A}}|\boldsymbol{S}, \boldsymbol{S}', A)$, where we observe a next state transition from $\boldsymbol{S}$ to $\boldsymbol{S}'$ and ask what the next state would have been if the agent had selected action $\hat{A}$ instead of A.

Compared to traditional state-transition models of the form $P(\boldsymbol{S}'|\boldsymbol{S}, \hat{A})$, hindsight counterfactuals condition on more information and thus are potentially more accurate in generating virtual transitions. Sun et al. (2024) provide empirical evidence that hindsight state transitions speed up learning an optimal policy. Generating hindsight state transitions requires a causal model and is not possible with a traditional RL transition model that is based on conditional probabilities only. A topic for future research is using hindsight counterfactuals to generate roll-outs that are longer than single transitions, as is common in other model-based approaches (Janner et al., 2019; Sutton, 1990).

Another open topic for future research is leveraging for data augmentation hindsight *rewards*, which take the form $P(\boldsymbol{R}'_{\hat{A}}|\boldsymbol{S}, \boldsymbol{R}', A)$. A data augmentation example using hindsight reward counterfactuals would be as follows: Having observed the transition

$$(CG_t = 1, PH_t = 1, SH_t = 0; CG_{t+1} = 1, PH_{t+1} = 1, SH_{t+1} = 1, SC_{t+1} = 1),$$

we can augment the data with the counterfactual outcome

$$(CG_t = 1, PH_t = 1, SH_t = 1, SC_t = 1)$$

following the reasoning of Section 7.2. Hindsight rewards are considered in the well-known hindsight experience replay approach (Andrychowicz et al., 2017). Hindsight experience replay is based on evaluating multiple goals, corresponding to different reward signals. The dynamic causal models considered in this paper assume a single reward function. Dynamic causal models of hindsight rewards offer a promising approach to *hindsight credit assignment* (Harutyunyan et al., 2019).

Learning A Causal Model from Online Data. Causal model discovery methods that are applicable to online RL learning include deep models based on auto-encoders (Lu et al., 2018) and GANs (Sun et al., 2024).

Such deep generative models generate observations $\boldsymbol{x}$ from latent variables $\boldsymbol{z}$ (cf. Equation (12)), but they are not *structural* causal models based on a causal graph that represents local causal mechanisms. Huang et al. (2022) show how constraints from a given causal graph can be leveraged to learn latent state representations. Learning an influence diagram over state variables from online RL data seems to be a new research topic. An exciting new possibility for online learning is that the agent's exploration can include experimentation in order to ascertain the causal structure among the state variables.

Counterfactual Regret. Bareinboim (2020) describes an online counterfactual regret optimization procedure where the agent conditions on their intending to perform action A *before* they actually execute action A. For example, their policy may recommend $A = \pi(\boldsymbol{s})$, the agent conditions on this information, but considers alternative actions A'. Bareinboim gives the example of a gambler in a casino whose choices are influenced by how drunk they are; therefore intending a risky gamble should give them pause to reconsider. While it is possible for human agents to be unaware of what causes their intentions, for an agent to be opaque to themselves in this way raises deep philosophical and psychological questions about free will and intentionality, as Barenboim notes. In the case of a *computational agent* implemented by a computer program, the program's interface defines the possible inputs and hence the causes of its outputs. Thus when the behavioral agent is implemented by a computer program to which the learning agent has access, as they do in online learning, the causes of the behavioral agent's decisions are unlikely to be opaque to the learning agent.

8.2 Offline Causal RL

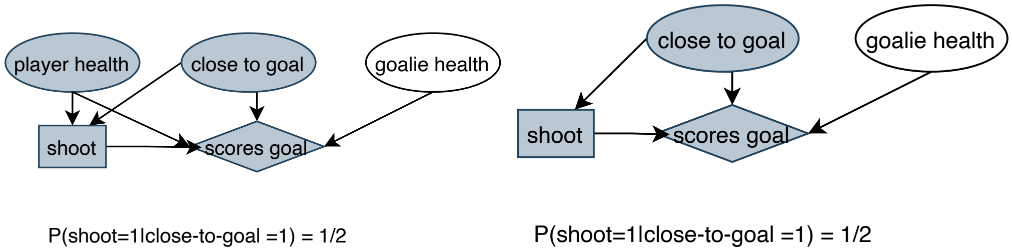
We discuss causal offline policy evaluation and imitation learning.

8.2.1 Offline Off-policy Evaluation

The off-policy evaluation (OPE) problem is to estimate the value function of a learned policy that is different from the behavioral policy. OPE is one of the major approaches to offline RL, where information about the behavioral policy is recorded in a previously collected dataset (Levine et al., 2020). The examples in this paper illustrated OPE based on a causal model. We selectively discuss OPE work related to causality and causal modeling.

Offline RL and Distribution Shift. As we mentioned in the introduction, Levine et al. (2020) assert that "offline reinforcement learning is about making and answering counterfactual queries." The key counterfactual for offline RL is about what might happen if an agent followed a policy different from the one observed in the data. In a car driving scenario (see Figure 9), we can ask based on expert driver data, what might happen if a beginner drives the car. As our examples show, the causal counterfactual semantics presented in Section 7 works well when the causal model supports inferring latent features of the same agent/environment, which facilitates counterfactual predictions. How causal models can be leveraged to answer counterfactual questions that involve changing agents is a valuable research topic for causal RL. It may be possible to address this in the transportability framework, which studies the extent to which causal mechanisms valid in one environment can be applied in ("transported to") another Bareinboim & Pearl (2014); Correa et al. (2022); Bareinboim (2020).

Another approach is to view offline RL as an instance of *distribution shift*: (Levine et al., 2020): while the dataset distribution over trajectories is based on the behavioral policy, the learned policy needs to be evaluated on the distribution reflecting its choices. Distribution shift departs from the i.i.d. assumption made in much machine learning work, because the training distribution differs from the test distribution (Levine et al., 2020, Sec.7). If causal mechanisms are shared between training and test distribution, as is often the case due to their local scope, they constrain the extent of distribution shift (Schölkopf et al., 2021). To illustrate this point, consider how can we leverage the causal model of Figure 1a to evaluate the marginal policy μ that directs a player to shoot with probability $1/2$ if they are close to the goal; see Figure 17a. Changing the policy affects only the parents of the shooting action variable. In causal terminology, this means that *the causal mechanisms governing the reward variable (Scores Goal) is the same even if the policy changes.* In MDP terminology, the reward model is invariant; the same reward model can be used to evaluate



(a) A new policy in the online model of Figure 1a

(b) A new policy in the offline model of Figure 1b

Figure 17: Leveraging a causal model to evaluate a policy different from the behavioral policy. The evaluation policy selects shooting with probability 1/2 if the player is close to the goal, and does not shoot if the player is far from the goal. Figure 17a: A causal graph for the new policy in the online setting. Player Health no longer is a cause of shooting. The graph remains the same for all other variables. Figure 17b: A causal graph for the new policy in the offline setting, where Player Health is unobserved. The graph agrees with the online graph for all observable variables. Note that the conditional probability $P(SC|SH, CG)$ is confounded because Player Health is not observed.

both the behavioral and the new policy. Similarly, the next state transitions are invariant under a policy change.

In causal modeling, the difficulty in modeling distribution shift arises from the presence of unobserved confounders. In the offline setting, Player Health is an unobserved confounder. If we use the same causal relations over the observable variables, as shown in Figure 17b, the Bayesian network requires estimating conditional reward probabilities of the form $P(SC|SH, CG)$ to evaluate the new policy. However, as we saw in Section 2.2, without observing Player Health, such conditional probabilities are confounded and do not correctly estimate interventional probabilities. On-going research in causal OPE addresses this issue.

Causal OPE. Most work is based on confounded MDPs (Zhang & Bareinboim, 2016; Kausik et al., 2024; Bruns-Smith, 2021): States are decomposed into an observed part and an unobserved part. The behavioral policy is assumed to depend on the complete state, whereas the policy to be evaluated depends on the observation signal only. We can view the belief POMDP of Section 3.1 as a confounded belief MDP. A current line of research gives bounds on the extent of the bias due to spurious correlations, based on assumptions about the confounders (Kausik et al., 2024; Bruns-Smith, 2021). For example, "memoryless confounders" are sampled independently at each time instant. At the other extreme, time-invariant latent variables such as Goalie Health or Player Health in our example are known as "global confounders" (Kausik et al., 2024). Some of the causal OPE methods are model-based in the sense of estimating transition probabilities, but not in the sense of utilizing a dynamic SCM. Another difference is that the evaluation policies considered are Markovian in that they depend on current observations only (cf. Section 4). In contrast, our belief MDP framework allows for evaluation policies that depend on past observations/current beliefs.

Warrington et al. (2021) and Ghosh et al. (2021) propose effective Bayesian approaches to confounding, where the learning agent maintains a distribution over information that was used by the behavioral agent during training, but cannot be directly observed by the learning agent at test time (e.g., location of ice patches, target locations on a map). They do not use the confounded MDP framework. A valuable topic for future research is whether causal modeling would further improve the Bayesian approach.

Leveraging a Causal Model for OPE Wang et al. (2021) propose policy optimization based on interventional probabilities, such as $P(SC|do(SH), CG)$ rather than conditional probabilities, such as $P(SC|SH, CG)$. To estimate the interventional probabilities, they assume that a causal model over the entire state space is available. The problem is then to compute a marginal interventional probability such as $P(SC|do(SH), CG)$ from conditional probabilities over the entire state space, such as $P(SC|SH, CG, PH)$. Pearl's do-calculus provides powerful techniques for inferring marginal interventional probabilities from conditional probabilities, using what are known as *adjustment formulas*. Two well-known types of adjustment formula are the backdoor and the frontdoor criterion. Wang et al. (2021) utilize both to compute causal reward and state-transition probabilities from a given causal model, and show how to use the causal probabilities in an interventional Bellman equation. To illustrate the idea in our sports example, consider offline learning in the confounded model of Figure 1b. Since in this setting, only closeness-to-goal is observable, an executable policy would be based on this variable only (e.g. "shoot whenever you are close to the goal"). Finding interventional values for such a policy involves computing interventional probabilities such as $P(SC = 1|CG = 1, do(SH = 1))$. According to the backdoor criterion, such probabilities can be computed by marginalizing over the unobserved values of player health as follows:

$$\begin{aligned} &P(SC = 1|CG = 1, do(SH = 1)) = \\ P(PH = 1)P(SC = 1|PH = 1, CG = 1, SH = 1)) + &P(PH = 0)P(SC = 1|PH = 0, CG = 1, SH = 1)) \\ &= 1/2 \cdot 1/2 + 1/2 \cdot 0 = 1/4 \end{aligned}$$

which agrees with the result of Table 3. While adjustment formulas provide an elegant approach to addressing spurious correlations in OPE, it is not entirely clear what their use case is when the behavioral policy is not executable, as the learning agent does not have access to the latent variables that appear in the adjustment formulas. A possibility is that the behavioral agent uses their access to latent variables (e.g., the athlete has access to their health) to compute the marginal interventional probability and communicate it to the offline learner.

Learning a Causal Model from Offline Data. Model-based offline RL is a major approach to offline RL (Levine et al., 2020, Sec.5.2). Many of the approaches designed for model-based offline RL can also be applied for offline RL based on causal models.

A specific possibility for causal models is to learn a fully specified latent variable model, such as the model of Figure 1b, and use the backdoor adjustment with latent variables to deconfound interventional probabilities (Lu et al., 2018). For learning a causal model offline, it is likely that the extensive work on learning Bayesian networks for temporal data can be leveraged, including recent approaches based on deep learning and continuous optimization (Sun et al., 2023). Latent variable models related to deep generative models are a promising direction for learning structural causal models (Geffner et al., 2022; Sun & Schulte, 2023; Mooij et al., 2016; Hyvärinen et al., 2010). Much of the research on learning a causal model assumes causal sufficiency (no confounders), but not all of it (Schölkopf et al., 2021). Learning an influence diagram with a causal graph over state variables from offline RL data seems to be a new research topic. Pearl (2018) emphasizes that domain knowledge is important in constructing a causal graph precisely because causal probabilities go beyond directly observable correlations. Most causal graph learning packages allow users to specify as background knowledge causal connections that should be present or absent (Spirtes et al., 2000).

8.2.2 Imitation Learning and Behavioral Cloning.

Imitation learning is a major approach to RL when the agent does not have the ability to interact directly with the environment online. Rather than learning an optimal policy from offline data, the goal is to learn a policy that matches an expert demonstrator. Zhang et al. (2020) introduce the concept of different observation spaces for the imitator and demonstrator, and state necessary and sufficient graphical conditions on a given causal domain model for when an imitating policy can match the expected return of a demonstrator, even when the observation signals of the imitator and demonstrator are different. They show that when imitator and demonstrator share the same observation signal, imitation is always possible (Zhang et al., 2020, Thm.1). This theory is the closest previous work to our results on executability and policy value equivalence.

The notion that imitation consists in matching the demonstrator's return is not the standard concept of imitation learning in RL; for example if the observation spaces are different, matching returns may require the imitator to follow a policy that is quite different from that of the demonstrator (Zhang et al., 2020, Fig.3b). A more usual concept of imitation in RL, known as *behavioral cloning*, is that the learned policy should be similar to that of the behavior policy that generated the data. Using our notation, the goal is to learn a policy π such that

$$b(A|\boldsymbol{S}) \approx \pi(A|\boldsymbol{S}) \tag{20}$$

where b is the behavior policy to be imitated. De Haan et al. (2019) in a paper on "causal confusion" point out that the conditional probability behavioral cloning objective eq. (20) can be problematic, even assuming executability. For example, in a self-driving car example like that of Figure 9, the observation signal includes the driver's own brake light. Since the brake light comes on only when the driver brakes, we have a very high correlation between braking and the brake light being on:

$$P(\mathit{Brake} = 1|\mathit{BrakeLight} = 1) \approx 1 \text{ and } P(\mathit{Brake} = 1|\mathit{BrakeLight} = 0) \approx 0.$$

Therefore if the imitator matches conditional probabilities according to the criterion eq. (20), they will brake only if they observe the brake light coming on. In this scenario, the imitator will not respond to the location of the other cars and their tail lights, which fails to match the expert driver's behavior (and fails to avoid accidents).

A causal diagnosis of the problem is that the imitator confused causes of acts with correlates of acts (cf. (Pearl, 2000, Ch.4.1.1)). As we discussed in Section 1, a high conditional probability such as $P(\mathit{Brake} = 1|\mathit{BrakeLight} = 1)$ is not causal if we condition on an effect (light) to predict a cause (brake). One remedy is to remove from the imitator's observation space effects of their actions, which also has the benefit of reducing the effective state space (see Section 8.1 above). A more fundamental approach is to redefine the behavioral cloning objective in terms of interventions:

$$b(A|do(\boldsymbol{S})) \approx \pi(A|do(\boldsymbol{S})) \tag{21}$$

This **causal behavioral cloning** objective can be interpreted as the query "what would the agent do if we were to put them into a state $\boldsymbol{S}$"? The causal objective eliminates the potential confusion due to non-causal correlations. To illustrate this point in our running example, in Figure 9 we have

$$P(\mathit{Brake} = 1|do(\mathit{BrakeLight} = 1)) = P(\mathit{Brake} = 1) << 1.$$

Causal behavioral cloning is a new topic for future research in causal RL.

8.3 Offline + online RL

In this hybrid setting, the learning agent interacts with the environment to collect more data, but a prior dataset is also utilized (Sutton, 1990; Janner et al., 2019). We can view this as an off-policy setting (see Figure 3a). A successful example is the AlphaGo system, which used an offline dataset of master games to find a good initial policy through imitation learning, then fine-tuned the policy with self-play (Silver et al., 2016). An active topic of research is how causal models can leverage offline datasets for online learning (Gasse et al., 2021; Zhang & Bareinboim, 2020), sometimes called generalized policy learning (Bareinboim, 2020).

Off-policy model-based RL approaches (Levine et al., 2020, Sec.5.2) can be applied with causal models to leverage the offline dataset. The techniques we outlined for causal models in the online and offline settings can be utilized in the hybrid offline/online setting as well. For example, we can use learn a causal graph from the offline dataset, and fine-tune it during online interactions with the environment. The ability to intervene in an environment in the online setting is potentially a powerful tool for learning a dynamic influence model. For instance, performing experiments can resolve the causal direction between two variables which may not be possible from observational data alone, at least not without assuming causal sufficiency.

Zhang & Bareinboim (2020) describe a hybrid approach for joint exploration and policy learning that is not based on a model. The offline dataset is used to estimate conditional state-reward transition probabilities. These estimates may not be correct interventional probabilities if confounders are present. Causal inference theory has established theoretical bounds on how far a conditional probability can differ from the interventional probability. Zhang and Barenboim use the resulting interval estimates for interventional probabilities as an input to optimistic exploration for online learning, where policies are evaluated according to their maximum possible value. Optimism under uncertainty is a well-known approach in RL for ensuring extensive exploration of the state space (Osband & Van Roy, 2014). The offline dataset provides tighter bounds on interventional probabilities than learning from online data only, thereby speeding up optimistic exploration.

9 Conclusion

We believe that many RL researchers share the intuition that in common traditional RL settings, conditional probabilities correctly estimate the causal effects of actions. Our paper spelled out the conditions where we can expect conditional probabilities to correctly measure causal effects, defined by interventional probabilities. We provided a rigorous argument, using the formal semantics of causal models, for why these conditions lead to correct estimates. Our conclusion is that it is only in partially observable environments with offline off-policy learning (the POOO setting) that conditional probabilities can diverge from observed conditional probabilities. The reason is that in this learning setting, the environment may contain confounders, not observed by the learning agent, that influence both the decisions of the behavioral agent and the states and rewards that follow these decisions. Such confounders can introduce spurious correlations between decisions and states/rewards that do not correctly estimate the causal impact of the agent's decisions.

In contrast, in an online or completely observable environment, such confounders are not present, and therefore conditional probabilities coincide with interventional probabilities. Our argument for this conclusion involves two steps. (1) In such environments, the set of variables that the learning agent can observe is causally sufficient for the behavioral agent's actions, in the sense that it includes all common causes of the behavioral agent's decisions and other variables. (2) We prove formally, using causal theory (2000), that if a set of observable variables is causally sufficient for actions, then actions are not confounded with states or rewards. Thus our analysis relies on the important distinction between the observation signals available to the learning and behavioral agents, which has been highlighted by previous work in causal RL (Zhang et al., 2020; Zhang & Bareinboim, 2016; Kausik et al., 2024; Gasse et al., 2021).

In addition to the causal effects of interventions, causal models provide a rigorous specification of counterfactual probabilities through a formal semantics. Causality researchers have recently proposed using counterfactuals to enhance reinforcement learning (Bareinboim, 2020; Deng et al., 2023). We therefore extend our analysis to counterfactuals, distinguishing two kinds of counterfactuals: what-if queries (e.g. if I choose action a' in state s instead of action a, what is the likely reward?) and hindsight counterfactuals that condition on an observed outcome (e.g. given that I received reward r after choosing action a in state s, what is the likely reward if I choose action a' instead?). We show that in an online or completely observable environment, what-if queries can be correctly estimated from conditional probabilities, but hindsight counterfactuals go beyond conditional probabilities (cf. Sun et al. (2024)).

Based on our analysis, we discussed the potential benefits of causal models in different reinforcement learning settings, such as online, offline, and off-policy. The most straightforward, though not the only, approach is to follow a model-based RL framework, replacing the traditional models involving conditional probabilities with a causal model. Structural causal models offer three main benefits: (1) They distinguish interventional and conditional probabilities, and therefore causation from correlation. (2) They factor the dynamics of a complex environment into local causal mechanisms represented in a causal graph. Local mechanisms are typically invariant under interventions (Schölkopf et al., 2021), which means that a causal graph can help address the challenge of distribution shift (Levine et al., 2020), where a learned policy is evaluated against data gathered by another policy. (3) Causal models have greater expressive power than conditional probability models, since they can also evaluate interventional and counterfactual probabilities. We described existing work and promising future directions for how the benefits of causal models can be leveraged for reinforcement learning.

Reinforcement learning and causality are areas of AI and machine learning that naturally complement each other. The analysis in this paper provides a guide for reinforcement learning researchers as to when and how they can make use of causal concepts and techniques to advance reinforcement learning.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, NIPS'17, pp. 5055–5065, 2017.

Elias Bareinboim. Towards causal reinforcement learning. International Conference on Machine Learning Tutorial, 2020. URL `https://crl.causalai.net`.

Elias Bareinboim and Judea Pearl. Transportability from multiple environments with limited experiments: Completeness results. *Advances in neural information processing systems*, 27, 2014.

Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and causal inference: the works of Judea Pearl*, pp. 507–556. ACM, 2022.

Craig Boutilier, Thomas L. Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research (JAIR)*, 11:1–94, 1999.

David A Bruns-Smith. Model-free and model-based policy evaluation when causality is uncertain. In *International Conference on Machine Learning*, pp. 1116–1126. PMLR, 2021.

Gregory F. Cooper and Changwon Yoo. Causal discovery from a mixture of experimental and observational data. In *Conference on Uncertainty in Artificial Intelligence*, UAI'99, pp. 116–125. Morgan Kaufmann Publishers Inc., 1999.

Juan D Correa, Sanghack Lee, and Elias Bareinboim. Counterfactual transportability: A formal approach. In *International Conference on Machine Learning*, pp. 4370–4390. PMLR, 2022.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhihong Deng, Jing Jiang, Guodong Long, and Chengqi Zhang. Causal reinforcement learning: A survey. *arXiv preprint arXiv:2307.01452*, 2023.

Marek J Druzdzel and Herbert A Simon. Causality in Bayesian belief networks. In *Conference on Uncertainty in Artificial Intelligence*, pp. 3–11. Elsevier, 1993.

Maxime Gasse, Damien Grasset, Guillaume Gaudron, and Pierre-Yves Oudeyer. Causal reinforcement learning using observational and interventional data. *arXiv preprint arXiv:2106.14421*, 2021.

Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in RL is difficult: Epistemic POMDPs and implicit partial observability. *Advances in neural information processing systems*, 34:25502–25515, 2021.

Sehoon Ha, Joonho Lee, Michiel van de Panne, Zhaoming Xie, Wenhao Yu, and Majid Khadiv. Learning-based legged locomotion; state of the art and future perspectives. *arXiv preprint arXiv:2406.01152*, 2024.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

Biwei Huang, Chaochao Lu, Liu Leqi, José Miguel Hernández-Lobato, Clark Glymour, Bernhard Schölkopf, and Kun Zhang. Action-sufficient state representation learning for control with structural constraints. In *International Conference on Machine Learning*, pp. 9260–9279. PMLR, 2022.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-Gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1–2):99–134, may 1998. ISSN 0004-3702.

Chinmaya Kausik, Yangyi Lu, Kevin Tan, Maggie Makar, Yixin Wang, and Ambuj Tewari. Offline policy evaluation and optimization under confounding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1459–1467. PMLR, 2024.

Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. *Advances in Neural Information Processing Systems*, 29, 2016.

Sanghack Lee and Elias Bareinboim. Structural causal bandits: Where to intervene? *Advances in Neural Information Processing Systems*, 31, 2018.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Guiliang Liu, Oliver Schulte, Pascal Poupart, Mike Rudd, and Mehrsan Javan. Learning agent representations for ice hockey. In *Advances in Neural Information Processing Systems*, 2020.

Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Deconfounding reinforcement learning in observational settings. *arXiv preprint arXiv:1812.10576*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored MDPs. *Advances in Neural Information Processing Systems*, 27, 2014.

J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge university press, 2000.

Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms.* The MIT Press, 2017.

Kyle Polich and Piotr Gmytrasiewicz. Interactive dynamic influence diagrams. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–3, 2007.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2010.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Rajat Sen, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Identifying best interventions through online importance sampling. In *International Conference on Machine Learning*, pp. 3057–3066. PMLR, 2017.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, 2000.

Xiangyu Sun and Oliver Schulte. Cause-effect inference in location-scale noise models: Maximum likelihood vs. independence testing. In *Advances in Neural Information Processing Systems*, 2023.

Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. NTS-NOTEARS: Learning nonparametric DBNs with prior knowledge. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

Yuewen Sun, Erli Wang, Biwei Huang, Chaochao Lu, Lu Feng, Changyin Sun, and Kun Zhang. ACAMDA: improving data efficiency in reinforcement learning through guided counterfactual data augmentation. In *AAAI*, pp. 15193–15201. AAAI Press, 2024.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *International Conference on Machine Learning*, pp. 216–224. Morgan Kaufmann, 1990.

Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning.* MIT Press Cambridge, 1998.

Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward Markov decision processes. In *International Conference on Machine Learning*, pp. 10653–10662. PMLR, 2021.

Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. *Advances in Neural Information Processing Systems*, 34:21164–21175, 2021.

Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. In *International Conference on Machine Learning*, July 2022.

Zizhao Wang, Caroline Wang, Xuesu Xiao, Yuke Zhu, and Peter Stone. Building minimal and reusable causal state abstractions for reinforcement learning. In *Proceedings AAAI*, 2024.

Andrew Warrington, Jonathan W Lavington, Adam Scibior, Mark Schmidt, and Frank Wood. Robust asymmetric learning in pomdps. In *International Conference on Machine Learning*, pp. 11013–11023. PMLR, 2021.

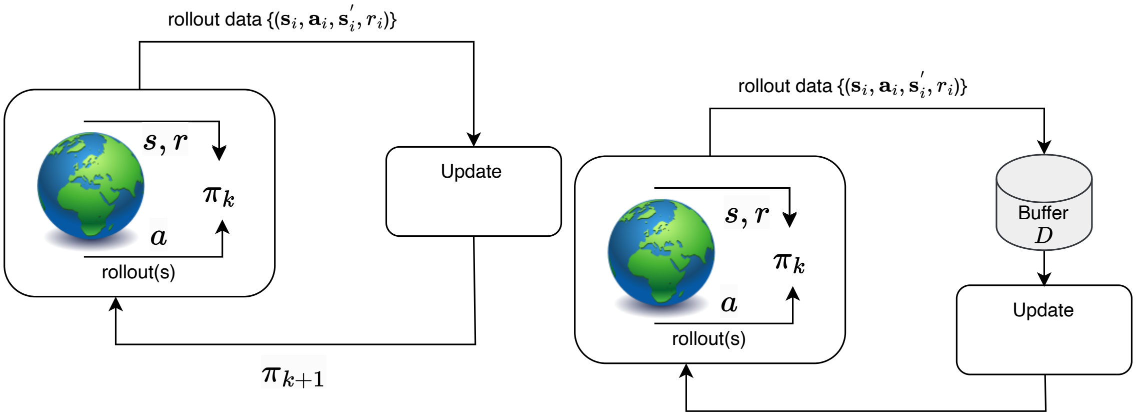
Junzhe Zhang and Elias Bareinboim. Markov decision processes with unobserved confounders: A causal approach. *Purdue AI Lab, West Lafayette, IN, USA, Tech. Rep*, 2016.

Junzhe Zhang and Elias Bareinboim. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In *International Conference on Machine Learning*, pp. 11012–11022. PMLR, 2020.

Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. Causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems*, 33:12263–12274, 2020.

A Online and Off-Policy Reinforcement Learning

In online reinforcement learning, the agent interacts with their environment through a behavioral policy to gather data. In the on-policy setting, the learned policy is evaluated using data generated by the learned policy itself. Figure 18 illustrates these learning settings with a diagram.


rollout data $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}_i^{'}, r_i)\}$
s, r
π_k
a
rollout(s)
Update
π_{k+1}
rollout data $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}_i^{'}, r_i)\}$
s, r
π_k
a
rollout(s)
Buffer
D
Update

(a) Online RL.

(b) Off-policy RL.

Figure 18: Online RL settings; figures adapted from Levine et al. (2020). Figure 18a: In classic **online** RL, the policy π_k is updated with streaming data collected by π_k itself. Figure 18b: In classic **off-policy** RL, the agent's online experience is appended to a data buffer (also called a replay buffer) D, each new policy π_k is used by the agent to collect additional data, such that D comprises samples from $\pi_0, \ldots, \pi_k$, and all of this data is used to train an updated new policy π_{k+1}. *In both online settings the behavioral policy is executable*, because the policies used to generate the data are based on the same observations as the learned policy.

B Proof of Lemmas 1 and 3

Lemma 1. *Let B be a causal Bayesian network and let $\boldsymbol{Y}, A, \boldsymbol{X}$ be a disjoint set of random variables such that $\boldsymbol{X} \supseteq Pa_A$. Then $P^B(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, do(A = \hat{a})) = P^B(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, A = \hat{a})$.*

Proof. Let $\boldsymbol{V}$ denote the set of variables distinct from A. Consider disjoint variables $\boldsymbol{X}, \boldsymbol{Y}, A$. Let $U = V - \boldsymbol{X} \cup \boldsymbol{Y} \cup A$ be the set of remaining random variables. If the manipulated variable A has no parents, the truncated and non-truncated model are the same, and the result follows immediately. Otherwise write $pa(\boldsymbol{v})$ for the assignment of values to the parents of A defined by the values $\boldsymbol{v}$. Similarly, we write $pa(\boldsymbol{x})$ to denote the assignment of values to the parents of A specified by the values $\boldsymbol{X} = \boldsymbol{x}$. Since $\boldsymbol{X}$ includes all parents of A, we have $pa(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{x}) = pa(\boldsymbol{x})$.

The truncation semantics implies that for any assignment $\boldsymbol{v}, a$, we have

$$P(\boldsymbol{V} = \boldsymbol{v}, A = \hat{a}) = P_{do(A=\hat{a})}(\boldsymbol{V} = \boldsymbol{v}, A = \hat{a})P(\hat{a}|pa(\boldsymbol{v}))$$

That is, the joint distribution differs only by including a term for the conditional probability of A given its parents. Now we have

$$\begin{aligned}
&P(\boldsymbol{y}|\boldsymbol{x},\hat{a}) \\
&= \frac{\sum_{\boldsymbol{u}} P(\boldsymbol{u},\boldsymbol{y},\boldsymbol{x},\hat{a})}{\sum_{\boldsymbol{u}',\boldsymbol{y}'} P(\boldsymbol{u}',\boldsymbol{y}',\boldsymbol{x},\hat{a})} \\
&= \frac{\sum_{\boldsymbol{u}} P_{do(\hat{a})}(\boldsymbol{u},\boldsymbol{y},\boldsymbol{x},\hat{a})P(\hat{a}|pa(\boldsymbol{u},\boldsymbol{y},\boldsymbol{x}))}{\sum_{\boldsymbol{u}',\boldsymbol{y}'} P_{do(\hat{a})}(\boldsymbol{u}',\boldsymbol{y}',\boldsymbol{x},\hat{a})P(\hat{a}|pa(\boldsymbol{u}',\boldsymbol{y}',\boldsymbol{x}))} \\
&= \frac{\sum_{\boldsymbol{u}} P_{do(\hat{a})}(\boldsymbol{u},\boldsymbol{y},\boldsymbol{x},\hat{a})P(\hat{a}|pa(\boldsymbol{x}))}{\sum_{\boldsymbol{u}',\boldsymbol{y}'} P_{do(\hat{a})}(\boldsymbol{u}',\boldsymbol{y}',\boldsymbol{x},\hat{a})P(\hat{a}|pa(\boldsymbol{x}))} \\
&= \frac{\sum_{\boldsymbol{u}} P_{do(\hat{a})}(\boldsymbol{u},\boldsymbol{y},\boldsymbol{x},\hat{a})}{\sum_{\boldsymbol{u}',\boldsymbol{y}'} P_{do(\hat{a})}(\boldsymbol{u}',\boldsymbol{y}',\boldsymbol{x},\hat{a})} \\
&= P(\boldsymbol{y}|\boldsymbol{x},do(\hat{a}))
\end{aligned}$$

□

Lemma 3. *Let C be a probabilistic SCM and let $\boldsymbol{Y}, A, \boldsymbol{X}$ be a disjoint set of random variables such that $\boldsymbol{X}$ includes all parents of A except for possibly a noise variable U_A of A (i.e., $\boldsymbol{X} \supseteq Pa_A - U_A$), and none of the descendants of A. Then for any actions $a, \hat{a}$ we have*

$$P^C(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}, A=a, do(A=\hat{a})) = P^C(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}, do(A=\hat{a})) = P^C(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}, A=\hat{a}).$$

Proof. The source variable posterior satisfies the following independence conditions:

$$\begin{aligned}
b(\boldsymbol{u}, u_A|\boldsymbol{X}=\boldsymbol{x}, A=a) = b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x}, A=a)\cdot b(u_A|\boldsymbol{X}=\boldsymbol{x}, A=a) \\
b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x}, A=a) = b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x}) \\
b(\boldsymbol{u}, u_A|\boldsymbol{X}=\boldsymbol{x}) = b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x})\cdot b(u_A|\boldsymbol{X}=\boldsymbol{x})
\end{aligned}$$

The first independence holds because $\boldsymbol{X}$ contains A, the only neighbor of u_A, and all the parents of A (i.e., the entire Markov blanket of u_A). The second independence holds because contains all the parents of $\boldsymbol{X}$ and none of its descendants, so by the Markov condition, A is independent of all its non-descendants. Since $\boldsymbol{u}$ contains only source variables, it contains no descendent of A. Similarly, the third independence holds because u_A is independent of all its non-descendants, and $\boldsymbol{X}$ contains no descendent of A and hence no descendant of u_A.

Now consider the evaluation of the counterfactuals $P^C(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}, A=a, do(A=\hat{a}))$ and $P^C(\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}, do(A=\hat{a}))$. Each probability is calculated in the same submodel $\mathcal{S}_{\hat{a}}$ but with different posteriors. Let C_1 be the submodel with source posterior distribution $b(\boldsymbol{u}, u_A|\boldsymbol{X}=\boldsymbol{x}, A=a)$ and let C_2 be the submodel with source posterior distribution $b(\boldsymbol{u}, u_A|\boldsymbol{X}=\boldsymbol{x})$. In each submodel, A is not generated by source variables but manipulated to the value $\hat{a}$. Let $\boldsymbol{U}$ be the set of source variables other than U_A. For an assignment of values to variables $\boldsymbol{W}=\boldsymbol{w}$, where $A \notin \boldsymbol{W}$, let $U_{\boldsymbol{w}|a}$ be the set of assignments to the source variables $\boldsymbol{U}$ such that the recursive solution procedure generates the assignment $\boldsymbol{W}=\boldsymbol{w}$ if variable $A=a$. Together with the independence conditions above, we therefore have the following:

$$\begin{aligned}
&P^{C_1}(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{X}=\boldsymbol{x})\\
&=\sum_{\boldsymbol{u}\in U_{\boldsymbol{x},\boldsymbol{y}|\hat{a}}}\sum_{u_a} b(\boldsymbol{u},u_A|\boldsymbol{X}=\boldsymbol{x},A=a)\\
&=\sum_{\boldsymbol{u}\in U_{\boldsymbol{x},\boldsymbol{y}|\hat{a}}} b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x})\sum_{u_a} b(u_A|\boldsymbol{X}=\boldsymbol{x},A=a)\\
&=\sum_{\boldsymbol{u}\in U_{\boldsymbol{x},\boldsymbol{y}|\hat{a}}} b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x})\cdot 1\\
&=\sum_{\boldsymbol{u}\in U_{\boldsymbol{x},\boldsymbol{y}|\hat{a}}} b(\boldsymbol{u}|\boldsymbol{X}=\boldsymbol{x})\sum_{u_a} b(u_A|\boldsymbol{X}=\boldsymbol{x})\\
&=P^{C_2}(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{X}=\boldsymbol{x})
\end{aligned}$$

Since the joint probabilities are the same for each posterior, so are the conditional counterfactual probabilities, which establishes the first equality of the Lemma.

The second equality follows as in Lemma 1: Given a parent assignment $pa_A=\boldsymbol{x}$, we can define a conditional probability over actions by $P(a|\boldsymbol{x})=\sum_{u_A:f_A(\boldsymbol{x},u)=a} b(u_A)$, that is, summing over the set of noise variable variables that generate the observed action. The conditional and interventional distributions differ only by this term, which does not depend on the target $\boldsymbol{Y}$ and therefore cancels out in the conditional probability.

□

C Belief Dynamics in a POMDP

Belief Dynamics Given a current action a_t and observation $\boldsymbol{o}_{t+1}$, the agent's beliefs move from current beliefs b_t to updated beliefs b_{t+1} through posterior updates; see Figure 7b. The observation signal provides an agent with information about the latent environment state through the posterior distribution $P(\boldsymbol{z}_{t+1}|\boldsymbol{o}_{\leqslant t+1},a_{\leqslant t})$ where $\boldsymbol{z}_{t+1}$ is the current latent state, $\boldsymbol{o}_{\leqslant t+1}$ is the current observation sequence, and $a_{\leqslant t}$ is the previous observation sequence before taking the current decision. We derive the well-known POMDP formula for recursively updating the latent state posterior (Russell & Norvig, 2010, Ch.17.4.2) for a factored POMDP and an executable policy. For the purposes of the derivation, we define executability in terms of observation sequences, meaning simply that a policy is executable if it depends only on observations, not the latent state component. This allows us to establish a more general result and simplifies our notation compared to the epistemic state notion. Formally, a policy π is **executable** if

$$\pi(a_t|\boldsymbol{z}_t,\boldsymbol{o}_{\leqslant t},\boldsymbol{a}_{<t})=\pi(a_t|\boldsymbol{o}_{\leqslant t},\boldsymbol{a}_{<t})$$

Observation 2. *If the transition data are generated by a POMDP P_E and an executable policy, the latent posterior update is given by*

$$P(\boldsymbol{z}_{t+1}|\boldsymbol{o}_{\leqslant t+1},a_{\leqslant t})=\alpha P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1},\boldsymbol{o}_t,a_t)\times\sum_{\boldsymbol{z}_t}P_E(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{o}_t,a_t)P(\boldsymbol{z}_t|\boldsymbol{o}_{\leqslant t},a_{<t}) \tag{22}$$

where α is a normalization constant.

Proof.

$$\begin{aligned}
P(\boldsymbol{z}_{t+1}|\boldsymbol{o}_{\leqslant t+1},a_{\leqslant t})&\propto P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1},\boldsymbol{o}_{\leqslant t},a_{\leqslant t})\sum_{\boldsymbol{z}_t}P_E(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{o}_{\leqslant t},a_{\leqslant t})P(\boldsymbol{z}_t|\boldsymbol{o}_{\leqslant t},a_{\leqslant t}) \text{ by Bayes' theorem}\\
&=P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1},\boldsymbol{o}_t,a_t)\sum_{\boldsymbol{z}_t}P_E(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{o}_t,a_t)P(\boldsymbol{z}_t|\boldsymbol{o}_{\leqslant t},a_{\leqslant t}) \text{ applying the Markov property}\\
&=P_E(\boldsymbol{o}_{t+1}|\boldsymbol{z}_{t+1},\boldsymbol{o}_t,a_t)\sum_{\boldsymbol{z}_t}P_E(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t,\boldsymbol{o}_t,a_t)P(\boldsymbol{z}_t|\boldsymbol{o}_{\leqslant t},a_{<t}) \text{ because } \pi \text{ is executable}
\end{aligned}$$

□

Equation (22) shows how the new belief state b_{t+1} can be computed from the previous belief state b_t. The standard notation for an agent's posterior **belief state** at time t is $b_t(\boldsymbol{z}_{t+1}) \equiv P(\boldsymbol{z}_{t+1}|\boldsymbol{o}_{\leqslant t+1}, a_{\leqslant t})$. We adopt the standard POMDP notation for an agent's current belief state b and b' for a successor belief state. Similarly, we write $\boldsymbol{o}, \boldsymbol{o}'$ for an observation signal and its successor, and $\boldsymbol{z}, \boldsymbol{z}'$ for a latent state component and its successor. With these conventions, the belief state update Equation (22) becomes

$$b'(\boldsymbol{z}') = \alpha P_E(\boldsymbol{o}'|\boldsymbol{z}', \boldsymbol{o}, a) \times E_{\boldsymbol{z}\sim b(\boldsymbol{z})}[P_E(\boldsymbol{z}'|\boldsymbol{z}, \boldsymbol{o}, a)] \tag{23}$$

Equation (23) is analogous to the standard POMDP for atomic POMDPs, with the latent state variables $\boldsymbol{z}$ replacing the latent state s.