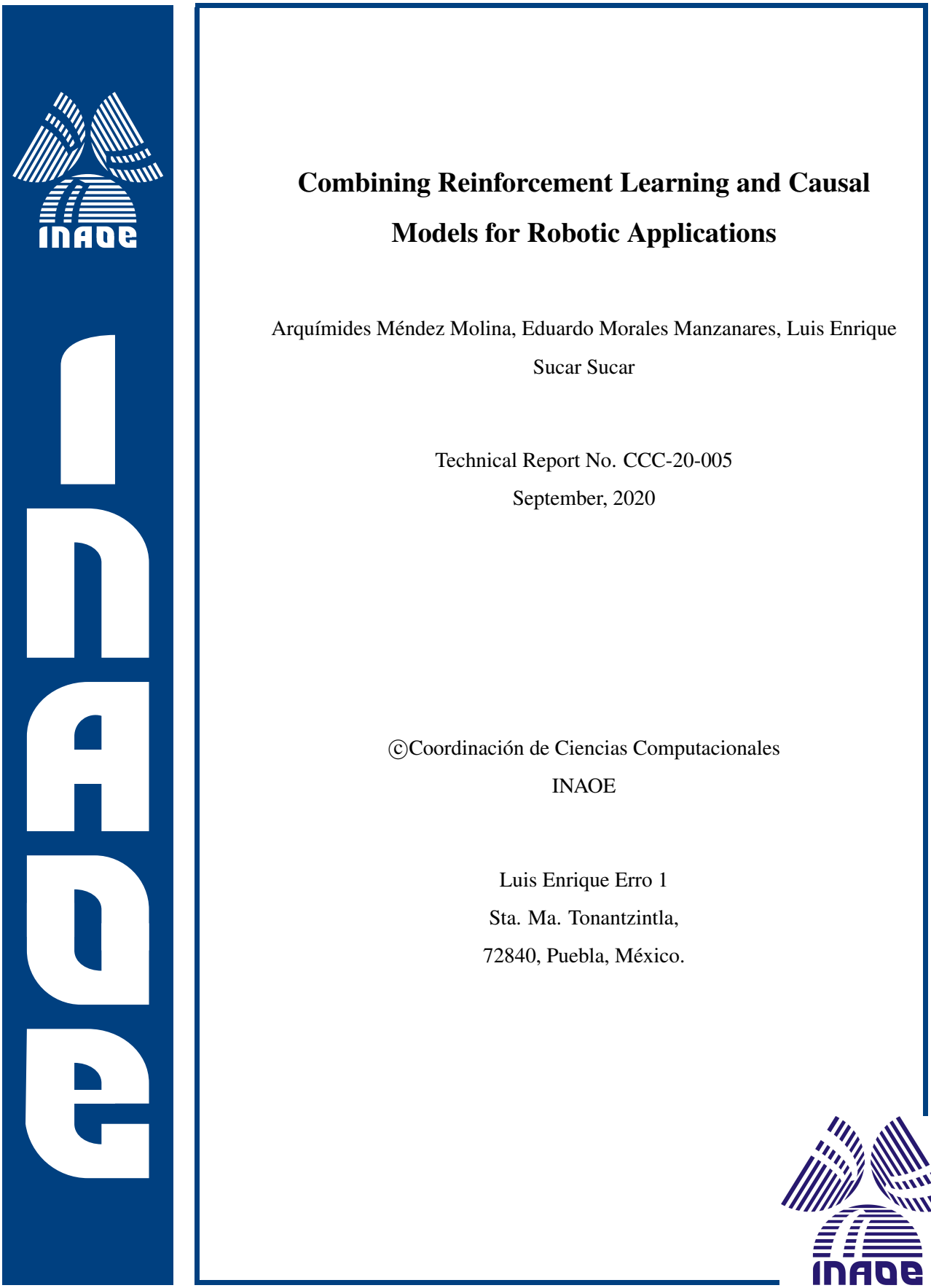

Combining Reinforcement Learning and Causal Models for Robotic Applications

Arquímides Méndez Molina, Eduardo Morales Manzanares, Luis Enrique Sucar Sucar

Technical Report No. CCC-20-005
September, 2020

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Abstract

Both reinforcement learning (RL) and Causal Modeling(CM) are indispensable part of machine learning and each plays an essential role in artificial intelligence, however, they are usually treated separately, despite the fact that both are directly relevant to problem solving processes. On the one hand, Reinforcement Learning has proven to be successful in many sequential decision problems (been robotics a prominent field of application); and on the other hand, causal models using Graphical Probabilistic Models is clearly a novel but a relevant and related area with untapped potential for any learning task. In this Ph.D. research proposal, we combine both areas to improve their respective learning processes, especially in the context of our application area (service robotics). The idea is to use observational and interventional data from a reinforcement learning agent to discover the underlying causal structure and simultaneously use this structure to learn better and faster a policy for a given task. The preliminary results obtained so far are a good starting point for thinking about the success of our research project, especially for part of our hypothesis which states that once the casual model is known, the learning time can be improved when compared with traditional reinforcement algorithms.

Keywords— Reinforcement Learning, Causal Models , Robotics

Contents

1 Introduction

Reinforcement Learning (RL) [48] is the study of how an agent (human, animal or machine) can learn to choose actions that maximize its future rewards. This approach is inspired by the way humans learn, and this is to let the agent to explore the environment to learn a task through rewards associated with each of the actions taken on each situation (labeled as state) along the way. Determining what action to take on each state is known as a policy. RL provides a promising technique to solve complex sequential decision making problems in several domains such as healthcare, economy, robotics (our area of interest), among others. However, existing studies apply RL algorithms in discovering optimal policies for a targeted problem, but ignores the abundant causal relationships present in the target domain.

Causal Modeling [39] (CM) is another learning paradigm concerned at uncovering the cause-effect relations between a set of variables. It provides the information for an intelligent system to predict what may happen next so that it can better plan for the future. In this paradigm, it is also possible to reason backwards: If I desire this outcome, what actions should I take? Given a causal structure of a system, it is possible to predict what would happen if some variables are intervened, estimate the effect of confounding factors that affect both an intervention and its outcome, and also, predict the outcomes of cases that were never observed before.

In recent years, the machine learning research community has expressed growing interest in both fields. This interest in Reinforcement Learning has been fueled by significant achievements in combining Deep Learning and Reinforcement Learning to create agents capable of defeating human experts. Prominent examples include the ancient strategy game Go [43] and Atari games [34].

Both reinforcement learning (RL) and Causal Modeling(CM) are an indispensable parts of machine learning and each plays an essential role in artificial intelligence, however, they are usually treated separately, despite the fact that both are directly relevant to problem solving processes. At present, the first works focusing on the relationship between these learning methods are beginning to be appeared [18, 21, 51, 8]. However, a growth in what some are beginning to call (CausalRL) [31, 32] is to be expected in order to become an indispensable part of General Artificial Inteligence. What CausalRL does, seems to mimic human behaviors, learn causal effects from an agent communicating with the environment, and then optimizing its policy based on the learned causal relationships. More specifically, humans summarize rules or experience from their interaction with nature and then exploit this to improve their adaptation in the next exploration. [31]

One area with great application possibilities is robotics. So far, the use of traditional RL techniques for learning task in robotics has been hampered by the following aspects: (i) problems for learning in continuous spaces, (ii) the inability to re-use previously learned policies in new, although related tasks, (iii)

difficulty in incorporating domain knowledge, (iv) long learning times and (v) many data samples. Our research efforts would be focused on the use of causal models in favor of reinforcement learning to mainly attack the problems raised in points (ii, iii, iv). On the other hand, we believe that it is possible to use the Reinforcement Learning process thought directed interventions to improve or discover new relationships of the underlying causal model for the given task or problem.

Through this document, a new methodology for learning and using Causal Models during Reinforcement Learning will be developed and hypothesized to be computationally feasible.

1.1 Motivation

One of the main motivations for this work is that in the area of robotics there are certain characteristics that facilitate causal discovery, such as the fact that interventions (experiments) can be made, which is prohibitive in other areas.

If an agent can know the possible consequences of its actions, it can then make a better selection of them. This is particularly relevant in RL because that knowledge, which can be given by a causal model, can significantly reduce the exploration process and therefore accelerate the learning process (as will be seen in the preliminary results). On the other hand, trying to learn causal models from observational data presents several problems which can be reduced if we can make interventions, so more reliable causal models can be learned. This is relevant for Robotics, because under certain circumstances, targeted interventions can be made to learn causal models. In this thesis we are going to see how we can link RL and Causal Discovery to learn faster policies and better causal models. [1]

In this way, a very attractive set of potential applications arises, such as: explanation (the agent can explain the reason for its actions using causal models), transfer (the learned causal models can be directly reused between similar tasks), and efficiency (reduce the long times of the learning process by reinforcement learning).

1.2 Justification

This research will address the following open problems reported in the literature: From the Reinforcement Learning perspective:

[1] It is worth mentioning that Machine Learning pioneers such as Yoshua Bengio (a computer scientist at the University of Montreal who shared the 2018 Turing Award for his work on deep learning) recently suggests that creating algorithms that can infer cause and effect is the key to avoiding another AI winter and unlocking new frontiers in machine intelligence.

1. Long training times and many training examples.
2. Difficult to re-use previously learned policies in new, although related tasks.
3. Difficult to incorporate causal relationships of the target task

From Causal Modeling perspective:

1. How to combine observational and interventional data from a robotics domain in an efficient way for causal discovery?
2. How to use and guide Reinforcement Learning experiences for causal discovery?

1.3 Problem Statement

How can we provide an intelligent agent with the ability to simultaneously learn causal relationships and efficient induction of task policies, based on the experiences obtained during the reinforcement learning process?

Formally: Let G be a causal graphical model and let $M = (S, A, T, R)$ a sequential Markov Decision Problem (MDP) whose actions (A), states (S), and rewards (R) are causally related and corresponds to variables in G. Let $(RL_{\pi/Q})$ denote the process of learning a policy π and a value function (Q) for M following a reinforcement learning algorithm. Let CL_G denote the learning process of G following a causal discovery algorithm. Consider a decision maker who does not know the parameters nor the structure of G which control the probabilities of observing a consequence (transition T or reward R) given an action $a \in A$ we want to solve how to integrate CL_G during $RL_{\pi/Q}$ such that the system can learn faster/better π and Q.

1.4 Research Questions

The following questions will guide this doctoral research:

1. How can we use partial causal models for learning more efficiently a policy for a given task?
2. How we can use data from an agent performing RL to learn a causal PGM model? And which variables, parameters and suppositions should be part of that model?
3. How to trade off exploration and exploitation when trying to learn about the causal structure of the environment while also trying to make good choices?
4. Having learned a causal PGM during RL, how can we use it to learn similar tasks of increasing complexity more efficiently?

1.5 Hypothesis

The general hypothesis of this research is:

Combining jointly Reinforcement Learning and Causal Discovery can produce faster policy learning and better causal models, where "faster" means that a nearly-optimal policy can be obtained in fewer episodes than traditional RL; and "better" that the structure of the causal model will be closer to the actual one, than if only observational data is used.

1.6 Objectives

1.6.1 General Objective

The general objective of this research is to develop and validate an algorithm to learn faster nearly-optimal policies and better causal models by combining reinforcement learning and causal discovery.

1.6.2 Specifics Objectives

1. To define a strategy for using a given causal model to speed up the Reinforcement Learning process.
2. To analyze the theoretical requirements for causal discovery in order to identify to what extent and under what assumption it is possible to use data generated by Reinforcement Learning agent to learn the underlying causal model.
3. To define a strategy for the combination of observational and interventional data in the causal discovery phase.
4. To integrate causal discovery, causal inference and reinforcement learning in a single algorithm.
5. To verify and validate the proposed algorithm in a suite of tasks of increasing complexity in the robotics domain.

1.7 Scope and Limitations

The proposed algorithm will not be limited to applications in robotics, but it is necessary that the addressed decision problem allows interventions. It is assumed that it will be possible to produce realistic simulations of the given task.

1.8 Expected Contributions

In general terms, the main contribution of the present research is a novel way to integrate these two important areas (Reinforcement Learning and Causal Models) in the context of sequential decision-making process. More specific contributions will be:

- An algorithm for learning causal models during a Reinforcement Learning process.
- An algorithm to use causal models in Reinforcement Learning.
- An algorithm that simultaneously learn/use Causal Models during Reinforcement Learning.

1.9 Outline

The rest of this document is structured as follows. In chapter 2 we present the main theoretical concepts of our area of study. Then, in chapter 3 we present the related works. In chapter 4 we present the methodology to carry out the research. In chapter 5 we present the preliminary results and the final remarks in chapter 6.

2 Background

In this section, concepts related with Reinforcement Leaning and Causality (both, the area of study of our work) are presented. We will start by explaining the Reinforcement Learning paradigm and its formal definition as a Markov Decision Process, followed by the main types of existing algorithms and a comparison between model-free and model-based approaches. In Causality we will start from the formal definition we will use in our work and the difference between causal and associative relationships, highlighting the advantages of the former over the latter. We will then move on to the definition of causal models using Probabilistic Graphic Models (PGMs), specifically Bayesian Causal Networks (CBNs). Finally, the main concepts to be taken into account in our work on Causal Discovery and Causal Inference are described. When necessary we will highlight the approach we intend to use in our work.

2.1 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning concerned with how software agents can learn to make good sequences of decisions. Reinforcement learning is one of the three basic machine learning paradigms, alongside *supervised learning* and *unsupervised learning*.

In the words of Sutton and Barto, *"Reinforcement Learning is learning what to do, how to map situations to actions so as to maximize a numerical reward signal. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics trial and error search and delayed reward are the two most important distinguishing features of reinforcement learning."* [48, p. 18]

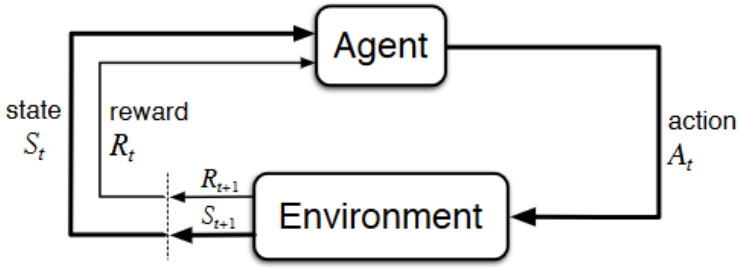

Figure 1: Agent-environment interaction in Reinforcement Learning (Taken from [48]). At a given state (S_t) the Agent executes an action (A_t) for which it receives a reward (R_t) and transfers to the state (S_{t+1}).

The agent must be able to sense the state of its environment to some extent and must be able to take actions that affect the state. The agent also must have a goal or goals relating to the state of the environment [48].

In mathematical terms, the learner (agent) and environment interact during a sequence of time steps, $t = 0, 1, 2, 3, ...$ At each time t, the agent receives some representation of the environment's state, $S_t \in S$, where S is the set of possible states, and based on its knowledge, selects an action, $A_t \in A(S_t)$, where $A(S_t)$ is the set of actions available in state S_t. One time step later, as a consequence of its action, the agent receives a numerical reward, $R(S_{t+1}) \in \mathbb{R}$, and finds itself in a new state, S_{t+1}. Figure 2.2 shows a diagram of the agent-environment interaction. At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted by π_t, where $\pi_t(a|s)$ is the probability of taking action a at time t if $S_t = s$. Reinforcement learning methods specify how the agent changes its policy as a result of its experience. The agent's goal, roughly speaking, is to maximize the total amount of reward it receives over the long run. There are different reward models. In this proposal we will used the expected discounted reward model. This reward is calculated using Equation 1.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{t=0}^{\infty} \gamma^t r_{t+1} \tag{1}$$

Where $0 \leq \gamma \leq 1$ is a discount rate based on the infinite horizon principle, which minimizes the influence of reinforcements received in the future.

The formal definition of reinforcement learning algorithms is based on the assumption that the environment has the Markov property[2]. A reinforcement learning task that satisfies the Markov property is called a Markov decision processes, or MDP. We explain MDPs in more detail in the next section.

2.1.1 Markov Decision Process

A Markov Decision Process (MDP) is a tuple $M =< S, A, T, R >$, where S is a set of states, $A(s) \in A$ is a set of actions for each state $s \in S$, T is the transition function $T : S \times A \times S \rightarrow [0, 1]$ and R is the reward function $R : S \times A \times S \rightarrow R$. A transition from state $s \in S$ to state $s' \in S$ caused by some action $a \in A(s)$ occurs with probability $P(s'|a, s)$ and receives a reward $R(s, a, s')$. A policy $\pi : S \rightarrow A$ for M specifies which action $a \in A(s)$ to execute when an agent is in some state $s \in S$, i.e., $\pi(s) = a$.

[2]A stochastic process has the Markov property if the conditional probability distribution of future states of the process (conditional on both past and present states) depends only upon the present state, not on the sequence of events that preceded it. A process with this property is called a Markov process.

A solution for a given MDP $M =< S, A, T, R >$ consists of finding a policy that maximizes the long-term expected total reward. A deterministic policy $\pi : S \rightarrow A$ specifies which actions $a \in A(s)$ to perform on each state $s \in S$. The policy is associated with a value function $V^{\pi} : S \rightarrow R$. For each state $s \in S$, $V^{\pi}(s)$ denotes the expected accumulated reward that will be obtained from state s and following the actions suggested by π. This can be expressed in a discounted infinite horizon by Equation 2.

$$V^{\pi}(s) = E_{\pi}\left(\sum_{t=0}^{\infty} \gamma^{t} R(s_t) | s_t = s\right) \tag{2}$$

Similary, the action-value function for policy π, denoted by $Q^{\pi}(a, s)$ is defined in Equation 3:

$$Q^{\pi}(s, a) = E_{\pi}\left(\sum_{t=0}^{\infty} \gamma^{t} R(s_t) | s_t = s, a_t = a\right) \tag{3}$$

and represent the expected accumulated reward that will be obtained by taking the action a in state s, as suggested by policy π.

The expression for V can be recursively defined in terms of the Bellman Equation in Equation 4.

$$V^{\pi}(s) = \sum_{s' \in S} P(s|\pi(s), s')(R(s) + \gamma V^{\pi}(s')) \tag{4}$$

The success of an agent is determined by how well it maximizes the total reward it receives in the long run while acting under some policy π. An optimal policy,π^*, is a policy which does maximize the expectation of this value.

2.1.2 Learning Algorithms

Any reasonable learning algorithm attempts to modify π over time so that the agent's performance approaches that of π^* in the limit. There are many possible approaches to learning such a policy, including:

- *Temporal difference(TD)* methods, such as Q-learning [49, 48] and Sarsa [40, 44], learn by backing up experienced rewards through time. An estimated action-value function, $Q : S \times A \rightarrow \mathbb{R}$ is learned, where $Q(s, a)$ is the expected return found when executing action a from states, and greedily following the current policy thereafter. The current best policy is generated from Q by simply selecting the action that has the highest value for the current state. Exploration, when the agent chooses an action to learn more about the environment, must be balanced with exploitation, when the agent selects what it believes to be the best action. One simple approach that balances the two is $\epsilon - greedy$ action

selection where the agent selects a random action with small probability ϵ, and the current best action is selected with probability $1 - \epsilon$.

- *Dynamic programming*[5] approaches assume that a full model of the environment is known (i.e.,S,A,T, and R are provided to the agent and are correct). No interaction with the environment is necessary, but the agent must iteratively compute approximations for the true value or action-value function, improving them over time.

- *Policy search methods*, such as policy iteration (dynamic programming), policy gradient [50, 4], and direct policy search [36],are in some sense simpler than TD methods because they directly modify a policy over time to increase the expected long-term reward by using search or other optimization techniques.

- *Batch learning methods* (e.g.,Least Squares Policy Iteration [27] and Fitted-Q Iteration [15] are offline and do not attempt to learn as the agent interacts with the environment. Batch methods are designed to be more sample efficient, as they can store a number of interactions with the environment and use the data multiple times for learning. Additionally, such methods allow a clear separation of the learning mechanism from the exploration mechanism (which much decide whether to attempt to gather more data about the environment or exploit the current best policy).

- *Relational reinforcement learning(RRL)* [13] uses a different learning algorithm as well as a different state representation. RRL may be appropriate if the state of an MDP can be described in a relational or first-order language. Such methods work by reasoning over individual objects (e.g., a single block in a Blocksworld task) and thus may be robust to changes in numbers of objects in a task.

- *Deep Reinforcement Learning(DRL)* In tasks with small, discrete state spaces, Q and π can be fully represented in a table. As the state space grows, using a table becomes impractical, or impossible if the state space is continuous. In such cases, RL learning methods use function approximators, such as artificial neural networks, which rely on concise, parameterized functions and use supervised learning methods to set these parameters. Function approximation is used in large or continuous tasks to better generalize experience. Parameters and biases in the approximator are used to abstract the state space so that observe data can influence a region of state space, rather than just a single state, and can substantially increase the speed of learning.

2.1.3 Model based RL vs Model Free RL

A commonly used distinction between RL algorithms is that of Model Based vs Model Free. The main difference lies in the presence or absence of the transition (T) and reward (R) functions.

A **model-based** algorithm is an algorithm that uses the transition function (and the reward function) in order to estimate the optimal policy. The agent might have access only to an approximation of the transition function and reward functions, which can be learned by the agent while it interacts with the environment or it can be given to the agent (e.g. by another agent). In general, in a model-based algorithm, the agent can potentially predict the dynamics of the environment (during or after the learning phase), because it has an estimate of the transition function (and reward function). However, note that the transition and reward functions that the agent uses in order to improve its estimate of the optimal policy might just be approximations of the "true" functions. Hence, the optimal policy might never be found (because of these approximations).

A **model-free** algorithm is an algorithm that estimates the optimal policy without using or estimating the dynamics (transition and reward functions) of the environment. In practice, a model-free algorithm either estimates a "value function" or the "policy" directly from experience (that is, the interaction between the agent and environment), without using neither the transition function nor the reward function. A value function can be thought of as a function which evaluates a state (or an action taken in a state), for all states. From this value function, a policy can then be derived.

In our work we intend to use both approaches in a certain way within the learning algorithm. For the model-free part we intend to use temporal differences (e.g. Q-Learning). Simultaneously, we intend to learn a model with the difference that this would be a causal model instead of a transition and/or recompilation function. A preliminary idea of this combination can be seem in Chapter 5.

2.2 Causality

Establishing causal (explicative) relations among variables is one of the main aims in several disciplines across science. The definition of causality considered in this thesis is the one given by Spirtes et al.:

"We understand causation to be a relation between particular events: something happens and causes something else to happen. Each cause is a particular event and each effect is a particular event. An event A can have more than one cause, none of which alone suffice to produce A. An event A can also be over determined: it can have more than one set of causes that suffice for A to occur. We assume that causation is (usually) transitive, irreflexive, and antisymmetric. That is, i) if A is a cause of B and B is a cause of C, then A is also a cause of C, ii) an event A cannot cause itself, and iii) if A is a cause of B then B is not a cause of A."[47, p. 42]

2.2.1 Difference between Causal Relations and Associative Relations

According to [38], there are three levels of causal reasoning: observing, manipulating and counterfactual reasoning. Associative relations (which are called zero-level causal relations) belong to the first level in which only correlations can be obtained from data. Correlations, or observational data by themselves offer a very simple tool to establish relations of dependence between variables. A wide variety of spurious (non-causal) correlations are known, and we can conclude from them that mere correlations do not suffice to establish anything beyond what is observable. The core of causal reasoning lies in the second and third levels, where it can be observed what happens after some variable is directly manipulated (interventions), and where one can ask what would happen if certain intervention had been done some other way (conunterfactuals).

Associative relations are based on correlation between events, or patterns of occurence to be found in data, while causal relations are based in cause-efect patterns, which result from some mechanism which is rooted in the nature of the observed events and can be obtained from manipulation. As an example, although a very simple one, consider the following: a dog that lives as a pet in some household has observed that every day it has food so it does not worry about eating or not, but any kid knows that whether he eats or not depends on certain grown-up being present. The dog uses only observed occurrences in order to obtain conclusions, while the kid can imagine what would happen if his parents arrive home late.

2.2.2 Advantages of Causal Models

Observational data has the limitation that any conclusion made is relative to the observed sample, and that the models used to extract knowledge from them are mainly based on correlations and samples information, which has severe limitations as mentioned above. Causal models allow to evaluate stronger claims, such as counterfactual reasoning. Counterfactuals and the ability to manipulate and observe are essentials when a model needs to be interpretable and explainable. Once a Causal Model is obtained, it can be used to predict the effect of certain interventions, as well as perform counterfactual reasoning in order to evaluate an intervention that has been made; i.e., what if another intervention had been performed? This allows for manipulation and planning of the environment. A Causal Model can allow for transferability of knowledge to similar domains, since similar causal relations can hold in other domains.

2.2.3 Causal Probabilistic Graphical Models

Information about causal relations between variables can be encoded in causal probabilistic graphical models[3]. In particular, Bayesian Networks are used for modeling causal relations, because they provide facilities to represent and infer effects of actions. In order to Bayesian networks encode reliably causal relations, a set of assumptions is required in their construction that is summarized in the following definition:

Definition 2.1 [39] Let $P(\mathbf{v})$ be a probability distribution over a set $\mathbf{V}$ of variables, and let a $P(\mathbf{v}|do(\mathbf{X} = \mathbf{x}))$ denote the resulting distribution from the intervention $do(\mathbf{X} = \mathbf{x})$ that sets a subset $\mathbf{X}$ of variables to constants $\mathbf{x}$, and delete all incoming edges to $\mathbf{X}$. Denote by $\mathbf{P}*$ the set of all interventional distributions $P(\mathbf{v}|do(\mathbf{X} = \mathbf{x}))$, $\mathbf{X} \subseteq \mathbf{V}$, including $P(\mathbf{v})$ that represents no intervention (i.e. $\mathbf{X} = \emptyset$). A DAG G is said to be a **causal Bayesian network** (CBN) compatible with $\mathbf{P}*$ if and only if the following three conditions hold for every distribution in $\mathbf{P}*$:

1. $P(\mathbf{v}|do(\mathbf{X} = \mathbf{x}))$ is Markov relative to G
2. $P(v_i|do(\mathbf{X} = \mathbf{x})) = 1\ \forall V_i \in \mathbf{X}$ whenever v_i is consistent with $\mathbf{x}$
3. $P(v_i|do(\mathbf{X} = \mathbf{x}, \mathbf{pa}(V_i)) = P(v_i|\mathbf{pa}(V_i))\ \forall V_i \notin \mathbf{X}$ whenever $\mathbf{pa}(V_i)$ is consistent with $\mathbf{x}$

The set of assumptions considered in this definition allow differentiating Bayesian networks from causal Bayesian networks. This definition assumes that the structure G of a causal Bayesian Network complies with the following rule:

Definition 2.2[47] $G = (V; E)$ represents a causal graph, when there is a directed edge $X \rightarrow Y$ in E iff X is a direct cause of Y relative to V.

Considering that a direct cause between variables is defined as follows:

Definition 2.3 [52] X is a direct cause of Y relative to V, if there exists $x1 \neq x2$ and z with $Z = V \setminus \{X, Y\}$, such that $P(y|do(X = x_1), Z = z) \neq P(y|do(X = x_2), Z = z)$

In this definition of direct cause, an intervention $do(X = x)$ is considered a mechanism that fixes X to value x, and delete all direct causes over X (the incoming edges to X). The reason for this graph surgery is due to these direct causes of X have no influence during the intervention. Moreover, the intervention makes the intervened variable independent of its direct causes. In Figure 2, some examples of interventions are shown.

[3]Although there are several types of causal models, in this research we consider the causal PGMs. Important concepts related to Probability Theory, Probabilistic Graphical Models, Graph Theory, etc are intentionally omitted here

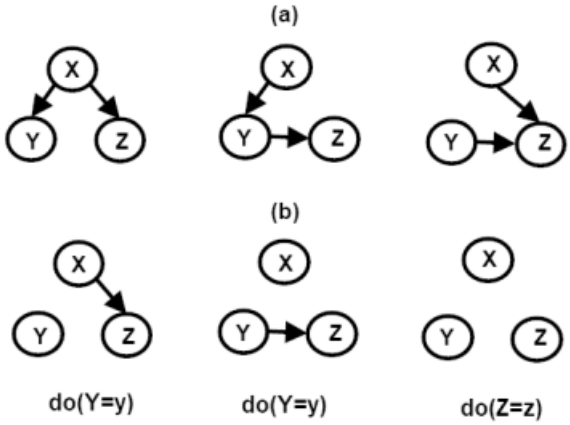

Figure 2: An example of (a) causal graph and (b) its corresponding modified version due to an intervention

2.2.4 Causal Discovery

The learning of causal Bayesian networks is often performed in two steps: learning of the causal structure, and estimation of parameters from data and the causal structure. In this section we separate the algorithms for causal discovery based on the nature of the data you have. The type of datasets used in the learning of the causal structure may be: **Observational**: Data corresponding to measurements made under natural conditions of a causal system.
Interventional: Data corresponding to measurements made under different disturbances of the system caused by external interventions.

Ideally, interventional data should be used in the structure learning of causal BNs. However, these data are difficult to collect, because it is complex, costly, or time demanding to perform experiments. Therefore, several algorithms have been developed to learn partial causal structures only from observational data. These algorithms, known as causal discovery algorithms, often relies on a number of assumptions including but not limited to the following:

1. **Causal Sufficiency (CS)** There are no common confounders of the observed variables in the model.
2. **Causal Markov Condition (CMC)** Each variable in the causal structure is independent of its non-effects given its directed causes (parents in the graph).
3. **Faithfulness Condition (FC)** Each true conditional independence between variables is entailed by the causal structure.
4. **Causal Minimality Condition (CLC)** No proper subgraph of the true causal G over V, with joint

distribution P, satisfies that P is Markov relative to G.

Three types of causal discovery algorithms have been proposed:

1. Constraint-based: These algorithms use statistical tests of conditional independence, which require a representative sample size to be reliable (Examples: PC[46], Fast Causal Inference (FCI) [45])
2. Score-based: Rely on various local heuristics to search for a directly acyclic graph according to a predefined score function (Examples: Greedy Equivalence Search (GES) [6], Greedy Fast Causal Inference (GFCI) [37])
3. Based on functional causal models: Learn causal relations assuming that those relations are in the form $Y = f(X, \epsilon)$, and in some cases, when some appropriate constraints are imposed to the functional model, a unique DAG within a Markov equivalence class can be identified (Linear Non-Gaussian Model (LiNGAm) [42], NonLinear Additive Noise Model (ANM)[22]).

The causal discovery algorithms, from the analysis of observational data and under several assumptions, can only recover a set of structures. Specifically, directed acyclic graphs equivalents to the true structure of a causal BN are recovered and grouped in a Markov equivalence class (MEC).

If the search method is limited to independence constraints, then in general, interventions are required to uniquely identify the true graph in a Markov equivalence class. For an intervention to be useful to discovery, it must place constraints on the system it intervenes on and bring exogenous influences into the system of variables under consideration [14]. The former is achieved by the manipulation of the marginal distribution of the intervened variables and by assumptions about parts of the causal structure surrounding the intervention variable. The latter is achieved by ensuring that the intervention is either exogenous to all other variables or has at least one cause that is not in the set of variables. Both aspects can be used for causal discovery since they distinguish causal structures which might otherwise appear indistinguishable.

Interventions which make the intervened variable independent of its normal causes are sometimes referred to as randomizations [17], surgical interventions [39], ideal interventions [47] or independent interventions [26]. In Fig 3 is presented an example of structural intervention.

Structural interventions take full control of the intervened variable, but an intervention does not have to be so strong. To qualify, an intervention only needs to influence the conditional distribution. This weaker form of an intervention is embodied in the notion of a parametric intervention, also sometimes called a partial, soft, conditional or dependent intervention. In Fig 4 is presented an example of parametric intervention.

In [14] are specified the strategies(pure or mixed) and computational bounds and for sequences of

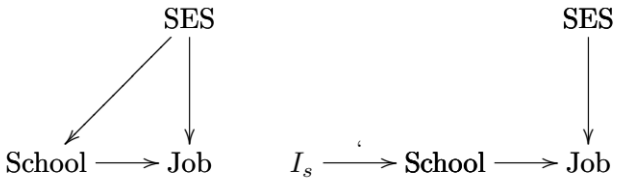

Figure 3: Example of structural intervention (taken form [14]): if the assignment of children to certain schools is random, then the socio-economic situation (SES) of the family, which under normal circumstances would influence the school district in which a child lives, becomes independent of the school assignment. If the social economic situation is also a cause of employment prospects, then randomness destroys any confusion between school attendance and employment prospects.

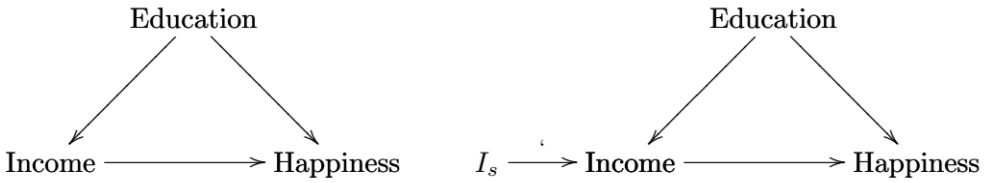

Figure 4: Example of parametric intervention (taken from [14]): Let us assume an intervention in the income of the participants in an experiment. Instead of establishing their income according to an independent probability distribution, thus determining it completely, a parametric intervention increases their income by $1,000. This would have the effect that people with high incomes would still have high incomes, determined largely by the root causes of their high income, but would have changed the conditional probability distribution, due to the influence of additional money.

experiments that are sufficient and for some worst case necessary to recover the causal structure among a set of N variables. That is, these sequences of experiments guarantee under the specified assumptions that sufficient constraints can be recovered to uniquely identify the true causal graph. These constraints come in the form of independence constraints or in terms of constraints on differences in correlations generated by the different experiments in a sequence. Given the constraints, the algorithms that perform the inference of the causal structure can be separated into algorithms based on independence constraints and those based on differences in correlations, and among those based on independence constraints there are posthoc and online algorithms. Posthoc algorithms are used for fixed strategies with pre-determined sequences of experiments. The algorithms are run after all experiments have been performed. Online algorithms are run after each experiment for in adaptive and mixed search strategies.

On a high level, structure search algorithms using independence relations works in three steps: (i) A stage that determines the information relevant to the structure search from the experimental set-up. (ii) A structure search that incorporates knowledge of the experimental set-up and searches for the manipulated causal structure in one experiment. (iii) A combining algorithm, that combines manipulated structures from different experiments to form one structure.

Although originally designed for structure search in passive observational data, algorithms like PC [47] and GES [6] can be supplemented with additional knowledge that allows search for structure search (step ii) in data that derives from a distribution subject to interventions. A set of algorithms in order of complexity of the combination algorithm (step iii) are also provided in [14].

In our work we intend to use observational and interventional data. So far we have only used algorithms that use observational data, but we know the significant contributions of targeted interventions to causal discovery.

2.2.5 Causal Inference

Through this document we use Casual Inference to refer to the techniques that allows us to answer queries, once the Casual Model is known. We make this distinction because in the literature the term Causal Inference was often used to refer simultaneously to the process of determining causal relationships (Causal Discovery) as well as to estimate the effect of intervening variables once they were known.

The most important type of queries that we will use in our research are intereventional queries of the form: *what is the probability of observe certain value in our target variable given that we set the value of other to a fixed value?* In the case that only observational data are available and the the causal structure is a known DAG and there are no hidden and selection variables, we can help with the Do-Calculus.

The Do-Calculus [39], summarized in Theorem 2.2, is a set of rules for manipulating probabilistic statements that involve interventions and, under certain conditions, allow them to be transformed into statements that do not involve interventional data.

Some notation is important to be mentioned: Consider a causal graphical model G and X, Y, Z disjoint sets of nodes of G. We denote by $G_{\overline{\mathrm{X}}}$ the graph that is obtained by deleting from G all of the edges that enter nodes in X. In the same way, $G_{\underline{\mathrm{X}}}$ is the graph obtained by deleting the edges that emerge from X. Finally, $G_{\underline{\mathrm{Z}}\overline{\mathrm{X}}}$ is the graph obtained by deleting edges incoming into X and outgoing from Z.

Theorem 2.2[39]: Let G a CGM and P_G the probability measure induced by the model; then, for disjoint sets of nodes X, Y, Z, W it holds:

- If for the graph $G_{\overline{\mathrm{X}}}$ it holds that Y is conditionally independent from Z given X and W, then

  $$P_G(Y = y|do(X = x), Z = z, W = w) = P_G(Y = y|do(X = x), W = w)$$

- If for the graph $G_{\underline{\mathrm{Z}}\overline{\mathrm{X}}}$ it holds that Y is conditionally independent from Z given X and W, then

  $$P(Y = y|do(X = x), do(Z = z), W = w) = P(Y = y|do(X = x), Z = z, W = w)$$

- Let $Z(W)$ the set of nodes in Z that are not ancestors of any node in W in the graph $G_{\overline{\mathrm{X}}}$. If Y is conditionally independent from Z given X and W in the graph $G_{\overline{\mathrm{X}},\overline{\mathrm{Z(W)}}}$, then

  $$P(Y = y|do(X = x), do(Z = z), W = w) = P(Y = y|do(X = x), W = w)$$

2.3 Summary

In this chapter we have presented the main concepts related to our research proposal, in which we intend to integrate several aspects of both areas (Reinforcement Learning and Causal Models) into a single algorithm. Within Reinforcement Learning we intend to use a combination of model-based and model-free methods with the fundamental difference that the model to be used will be a causal model. Within causal modeling, our work will address both causal discovery and causal inference, with greater emphasis on the former through the combination of observational and interventional data. In the next chapter we will see the main existing works related to our proposal.

3 Related work and State-of-the-art

This chapter is aimed at presenting some of the few existing works where the areas of Reinforcement Learning and Causality are related. These are grouped into three blocks: those who use Causal Inference ($CM \rightarrow RL$) as side information to improve RL, those who use Reinforcement Learning for Causal Discovery ($RL \rightarrow CM$), and finally and more important, those in which a causal model is simultaneously learned and used during a Reinforcement Learning task ($RL \leftrightarrow CM$).

3.1 RL and Causal Inference

This line of works is dedicated to improve RL models with causal knowledge as side information. Commonly, the problem tackle by these works is of the type *multi-armed bandit (MAB)*. In those problems a fixed limited set of resources must be allocated between competing (alternative) choices in a way that maximizes their expected gain, when each choice properties are only partially known at the time of allocation, and may become better understood as time passes or by allocating resources to the choice. This is a classic reinforcement learning problem that exemplifies the exploration–exploitation tradeoff dilemma. The name comes from imagining a gambler at a row of slot machines (sometimes known as "one-armed bandits"), who has to decide which machines to play, how many times to play each machine and in which order to play them, and whether to continue with the current machine or try a different machine.

In [3], the problem of unobserved confounders while trying to learn policies for RL models such as multi-armed bandits (MAB) is attacked. Without knowing the causal model, MAB algorithms can perform as badly as randomly taking an action at each time step. Specifically, the Causal Thompson Sampling algorithm is proposed to handle unobserved confounders in MAB problems. The reward distributions of the arms that are not preferred by the current policy can also be estimated through hypothetical interventions on the action (choice of arm). By doing this, it is possible to avoid confounding bias in estimating the causal effect of choosing an arm on the expected reward. To connect causality with RL, the authors view a strategy or a policy in RL as an intervention.

Another example is [30]. In this work, the problem is to identify the best action in a sequential decision-making setting when the reward distributions of the arms exhibit a non-trivial dependence structure, which is governed by the underlying causal model of the domain where the agent is deployed. In this setting, playing an arm corresponds to intervening on a set of variables and setting them to specific values. In the paper, it is shown that whenever the underlying causal model is not taken into account during the decision-making process, the standard strategies of simultaneously intervening on all variables or on all the subsets of the variables may, in general, lead to suboptimal policies, regardless of the number of interventions performed by the agent in the environment. An algorithm is proposed that takes as input a

causal structure and finds a minimal, sound, and complete set of qualified arms that an agent should play to maximize its expected reward. The authors empirically demonstrate that the new strategy learns an optimal policy and leads to orders of magnitude faster convergence rates when compared with its causal-insensitive counterparts. Later in [29], the same authors study a relaxed version of the structural causal bandit problem when not all variables are manipulable. Specifically, they develop a procedure that takes as argument partially specified causal knowledge and identifies the possibly-optimal arms in structural bandits with non-manipulable variables. They introduce an algorithm that uncovers non-trivial dependence structure among the possibly-optimal arms.

It is shown in [28] that adding causal information in a fixed budget decision problem [4] allows the decision maker to learn faster than if he does not considered causal information. Their work requires that the causal model is fully known to the decision maker, this requirement is relaxed later in [41] where the proposed system requires only that some part of the causal model is known and allow interventions over the unknown part. In [28] a causal graphical model G is assumed to be known and a number of learning rounds T is fixed. In round $t \in [1, ..., T]$ the decision maker chooses $a_t = do(X_t = x_t)$ and observes a reward Y_t . After the T learning rounds, the decision maker is expected to choose an optimal action a^* that minimizes the expected regret, which is defined as $R_T = \mu^* - \mathbb{E}[\mu_{a^*}]$ where $\mu^* = \max \mathbb{E}[a]$. They show that the achieved regret is smaller than the regret obtained by non-causal algorithms.

Note that the aforementioned papers assume the causal model is known to the decision makers so their work focuses on using causal information to make good choices, but the problem of acquiring this causal knowledge is left unatacked. Discovering the causal model itself while using current knowledge to make choices is left as future work in both [28] and [41].

An interesting example focus on transfer of knowledge in reinforcement learning (RL) settings using causal inference tools is presented in [53]. Here, the problem is how to transfer knowledge across bandit agents in settings where causal effects cannot be identified by Pearl's do-calculus nor standard off-policy learning techniques. A new identification strategy is proposed that combines two steps – first, deriving bounds over the arm's distribution based on structural knowledge; second, incorporating these bounds in a novel bandit algorithm, $B - kl - UCB$. Simulations shows that their strategy is consistently more efficient than the current (non-causal) state-of-the-art methods.

[4] In this setting, each action is associated with a cost and the agent cannot spend more than a fixed budget allocated for all the task

3.2 RL for Causal Discovery

Opposite to the works in the previous section, in this group the goal is to use Reinforcement Learning for Causal Discovery. There are few works so far that only focus on this direction, most of the works instead also use the Causal Model once learned to improve the policy for the task, as we will see in the next section.

In [33] is presented an approach that learns a structural causal model during reinforcement learning and encodes causal relationships between variables of interest. This model is then used to generate explanations about the agent's behavior based on counterfactual analysis of the causal model. The authors shows a feasibility of learning a set of structural equations, when given a graph of causal relations between variables. To this end, they assume that a DAG specifying causal direction between variables is given, and learn the structural equations relating (s_t, a_t, r_t, s_{t+1}) variables as multivariate regression models during the training phase of the RL agent.

Another interesting example that uses Reinforcement Learning for score-based Causal Discovery is presented in [54]. In this work, the authors uses deep reinforcement learning (DRL) to search for the DAG with the best scoring. An encoder-decoder model takes observable data as input and generates graph adjacency matrices that are used to compute corresponding rewards. The reward incorporates both a predefined score function and two penalty terms for enforcing acyclicity. In contrast with typical RL applications where the goal is to learn a policy, they use RL as a search strategy and the final output is the graph, among all graphs generated during training, that achieves the best reward.

3.3 Causal RL

The idea of combing RL and CM is fundamented in psychology works like [54]. This contrasts a model-free system that learns to repeat actions that lead to reward with a model-based system that learns a probabilistic causal model of the environment which it then uses to plan action sequences. Evidence suggests that these two systems coexist in the brain, both competing and cooperating with each other [1, 9, 12].

Recently, some authors have used the term Causal Reinforcement Learning to refer to "Learning causal effects from an agent communicating with the environment and then, optimizing its policy based on the learned causal relations" [31] and even to "putting these two disciplines under the same theoretical umbrella, in a way that several natural and pervasive classes of learning problems emerge" [2].

Some works such as [35, 25] has been focused on how to combine RL and CM to improve transfer between similar tasks.

In [35] is presented a method for Causal Induction using Visual Observations for Goal Directed

Tasks. During each training episode, the agent samples one of each K training environments and uses an interaction policy π_I to probe the environment and collect a trajectory of visual observations. Then, using supervised learning, they train the causal induction model F, which takes as input the trajectory of observational data and constructs C, which captures the underlying causal structure. Then, the predicted structure C is provided as input to the policy π_G conditioned on goal g, which learns to use the causal model to efficiently complete a specified goal in a given training environments. At test time, F and π_G are fixed and the agent is evaluated on new environments with unseen causal structures. The main limitation of this work is that the causal relations are just binary relations between lights and switches in a house.

Schema networks [25] is an example of how learning causal relationships and using them to plan can result in better transfer than model-free policies. In this work the authors introduce an object-oriented generative physics simulator capable of disentangling multiple causes of events and reasoning backward through causes to achieve goals. The richly structured architecture of the Schema Network can learn the dynamics of an environment directly from data. Compared with DRL methods like Asynchronous Advantage Actor-Critic and Progressive Networks on a suite of Breakout Game variations, Schema Networks reports better results on training efficiency and zero-shot generalization, consistently demonstrating faster, more robust learning and better transfer.

A closer approach to what we intend to do in our work can be seen in [19]. The authors propose a decision making procedure in which an agent holds beliefs about its environment which are used to make a choice and then are updated using the observed outcome. The agent, using its current beliefs (a Dirichlet Process is used to generate Dirichlet distributions of causal graphical models), generates a local causal model and chooses an action from it as if that model was the true one. Then, after it observes the consequences of its actions, its beliefs are updated according to the observed information in order to make a better choice the next time. The agent, besides learning a policy to choose actions will also learn a causal model from the environment since the causal model it forms will approximate the true model. In the experiments however, only the case where (i) the causal model is completely known and (ii) only the structure is known, are analyzed. The problem of discovering the variables itself and the connections between them is left as future work.

3.4 Summary

In spite of the progress in the areas of Reinforced Learning and Causal Models (both fundamental to the development of intelligent agents), today the first works focused on the combination of both areas are just beginning to appear. In Table 1 is presented a summary of these works is presented, grouping them by the sub-problem to be solved and highlighting the main existing limitations. The main differences with our proposal can be summarized as: (i) not limited to MAB environments, but to more general tasks whenever

it is possible to simulate interventions, (ii) the use of interventional data to obtain causal models closer to the real one, (iii) the causal structure is learned and used simultaneously with the learning of the task.

Topic	General Idea	Limitations of the existing works
$(CM \rightarrow RL)$	Use Causal Models as side information to improve traditional RL algorithms	Experiments only in Bandits environment [29, 30, 3, 28, 41]
$(RL \rightarrow CM)$	Use RL to learn causal relationships of the environment directly from data Use DeepRL to score-based causal discovery	The structure is given, so only parameters are learned [33]. The agent do not learn any policy for any task [54]
$(RL \leftrightarrow CM)$	Learning causal effects from an agent communicating with the environment and then, optimizing its policy based on the learned causal relations	Use observational data only. Structural assumptions according to the specific problem are made [19, 35, 25]

Table 1: Related Work Summarized

4 Research Proposal

Below is the proposed methodology to collect the evidence to accept or reject the hypothesis raised in Section 1.5.

4.1 Methodology

1. *Design and development of an algorithm to use causal models in RL* $(CM \rightarrow RL)$*:*

   In this stage, an algorithm to use a given Causal Model(s) in Reinforcement Learning will be designed. In order to do that it will be necessary:

   Study causal inference tools (mainly Pearl's do Calculus).

   Design a causal based action selection mechanism that exploit the advantages of causal inference to speed-up the learning process.

   Validation: The RL algorithm that used the causal model will be compared against a classical RL algorithm. The proposed algorithm will be considered successful if it converges faster or obtains higher rewards for a defined number of episodes.

2. *Study and evaluation of current solutions for learning the structure of causal PGMs using observational and interventional data:* In this step, the foundations of causal discovery will be studied through the analysis of the main existing algorithms for both observational and interventional data. Initially we will focus on learning the structure of the model and not on the parameters. For observational data, some constraint-based algorithms like (PC, FCI) and some score-based algorithms like (GES, GIES) will be analyzed. For interventional data we will explore structural, parametric, single and multiple simultaneous intervention. At the end of this step we hope to get an idea of the type of algorithms that would best suit the types of data generated during a Reinforcement Learning process.

3. *Design, analysis and selection of the variables, parameters and assumptions of the causal model:* In this step key aspects must be defined. For example, the variables of the causal model whose relations (direction and parameters of the relation function) are possible to infer using the results of the previous step and that at the same time can be useful in the process of inference to accelerate the learning of the task in question. In addition, the key assumptions (faithfulness, sufficiency, etc) of the model will be studied and selected. In this step, a set of candidate models could be obtained from which the definitive model will progressively emerge.

4. *Design and develop an algorithm for learning a causal model during RL* $(RL \rightarrow CM)$*:*

   In this stage, an algorithm for learning PGMs using data generated during Reinforcement Learning process will be developed.

A first stage could be based on algorithms using only observational data. By observational data we understand those data generated by the agent while performing actions focused on finding the optimal policy, following some reinforcement learning algorithm.

Later, given the equivalence class, it is possible to work with interventions (data generated by the agent while performing specific actions or by directly set some variables of the environment to a specific value) to try to orient the undefined edges and thus be closer to the true causal model.

Validation: The algorithm's ability for recovering skeleton and v-structures of the ground truth causal structures will be assessed. Here it is difficult to check the veracity of the model because we do not have a ground truth causal model a priori. One option may be to adapt some example from another domain where there is a ground truth, and validate the algorithm by making small modifications to the data from the other domain to be similar to the data generated during an RL process, and the other is to validate the model with experts in the target domain.

5. *Design an algorithm to integrate the algorithms from step 3 and 4 into one* $(RL \leftrightarrow CM)$*:*

   This stage aims to integrate the algorithms obtained from the previous two stages into one. In order to do that, it will be necessary:

   To define a strategy that combines the stages of causal discovery and causal inference with the stages of exploration and exploitation in Reinforcement Learning. Our first approach, based on Dyna-Q architecture [48] is presented in form of pseudo code in chapter 5.

   Validation: A set of tasks of increasing complexity in the field of robotics will be design in order to test the algorithm. The results obtained using the algorithm will be compared against traditional reinforcement learning algorithms. Also, we pretend to explore Transfer Learning ability of our algorithm by using the causal model generated for a single task as input to another more complex task.

The relation between stages in the methodology, objectives and research questions is presented in Table 2.

4.2 Work Plan

In Figure 4.2 is presented the schedule of activities for the realization of this research.

4.3 Publications Plan

The publications plan shown in Table 3 will be attempted:

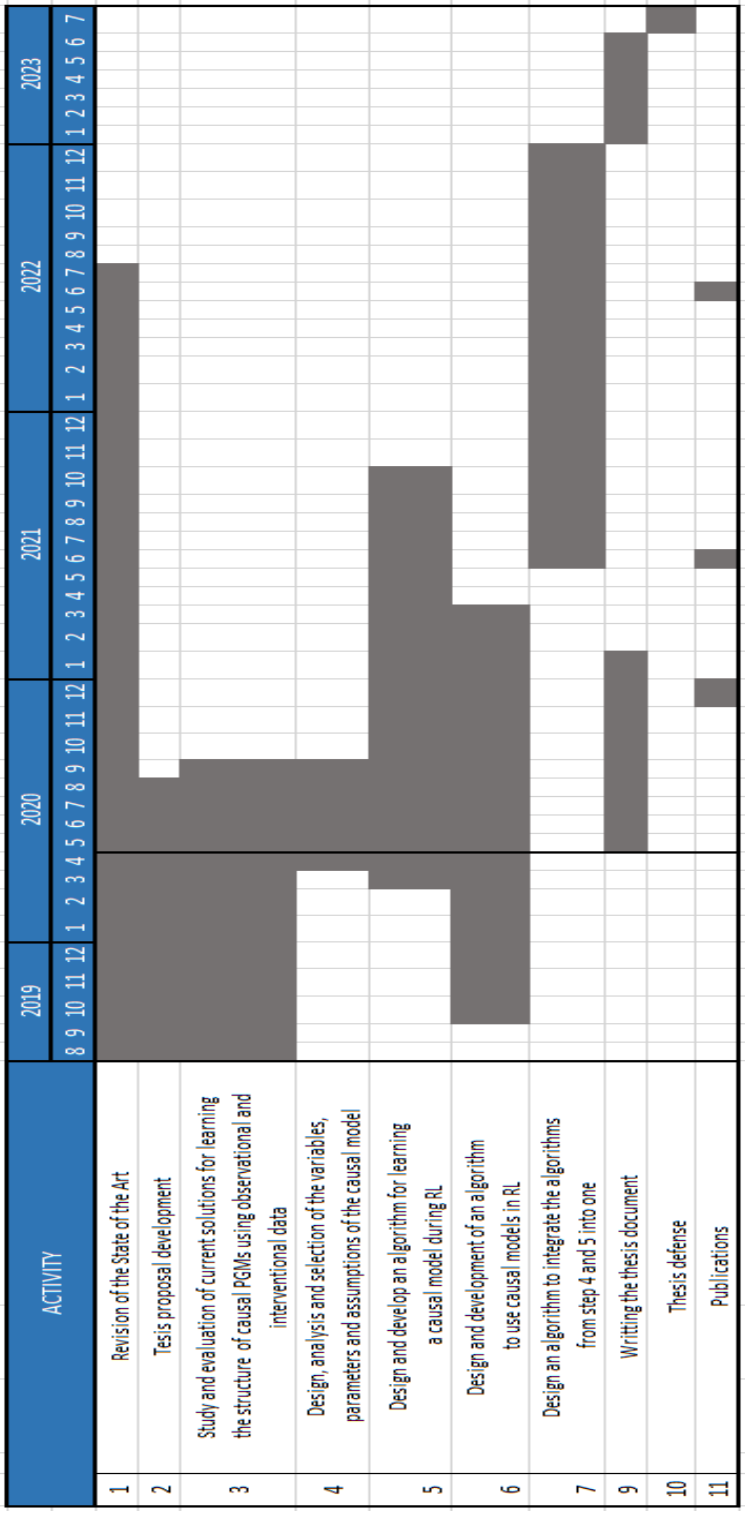

Figure 5: Calendar

Stage	Task	Objs	RQ
1	Algorithm to use causal models during RL $(CM \rightarrow RL)$	1	1
2	Study and evaluation of current solutions for learning the structure of causal PGMs using observational and interventional data	$2, 3, 4$	2
3	Design, analysis and selection of the variables, parameters and assumptions of the causal model	2	$2, 3$
4	Algorithm for learning a causal model in RL $(RL \rightarrow CM)$	2	$2, 3$
5	Algorithm to integrate the algorithms from step 4 and 5 into one $(CM \leftrightarrow RL)$	$4, 5$	4

Table 2: Relation between stages in the methodology, objectives (Objs) and research questions (RQ)

Name	Type	IF	Contribution	RQ	Obj	Submission
PGM or UAI	C	-	A method for causal based action selection in Reinforcement Learning	1	1	Dic 2020
International Journal of Approximate Reasoning	J	1.98	Survey in the interactions between Reinforcement Learning and Causal Models	2	2	Jun 2021
PGM or UAI	C	-	Algorithm to learn causal model using Reinforcement Learning	3	2,3,4	Dic 2021
Machine Learning	J	2.8	Algorithm to learn/use causal models in Reinforcement Learning	4	1,2,3,4,5	Jun 2022

Table 3: Publications plan, including journals (J) and conference (C) papers, impact factor (IF), relation with research questions (RQ) and objectives (Obj) and intended submission date

5 Preliminary Results

In this Chapter, preliminary results for this research are presented. Specifically, we address the first and second research questions. Guided by the estimated level of complexity involved we decided first to explore how a given causal knowledge could be used to learn new tasks more efficiently. For this we proposed and tested an action selection algorithm guided by causal knowledge [5] in a simple navigation task. In a second part, the first experiments focused on causal discovery were performed using observational data and already existing algorithms whose results support to some extent the feasibility of using the data collected during RL to obtain causal relationships. Finally, the first ideas on how Reinforcement Learning and Causal Discovery can be combined in a single algorithm are presented.

[5]This work was presented in the Causal Reasoning Workshop at the Mexican International Conference on Artificial Intelligence 2019

5.1 Using Causal Models to improve Reinforcement Learning

One of the challenges that emerge in Reinforcement Learning, is the trade-off between trying new actions (exploration) and selecting the best action based on previous experience (exploitation) in a given state. Traditional exploration and exploitation strategies are undirected and do not explicitly chose interesting transitions. Using predictive models is a promising way to cope with this problem. In particular, these models may hold causal knowledge, that is, causal relationships.

In this preliminary research we propose a method to guide the action selection in an RL algorithm using one or more causal models as oracles. The agent can consult those oracles to avoid actions that lead to unwanted states or choose the best option. This helps the agent learn faster since it will not act blindly. Through interventions in the causal model, we can make queries of the type What if I do ...?, e.g., If I drop the passenger off here, will my goal be achieved?

Our hypothesis is that causal inference can assist RL in learning value functions or policies more efficiently through the use of causal relations between state variables or between actions and state variables and therefore reducing the state or action space significantly.

To that end we proposed a method which consists of applying Algorithm 1 as a modification of the exploitation stage of the original Q-learning algorithm [49]. The difference (see lines 5 and 6 in Algorithm 1) is that the action selection in a given state it is first guided through interventional queries to one or more causal models, so the agent selects the action likely to allow it to meet a goal. If there is not such action, the algorithm continues exactly as the original Q-Learning.

Before explaining how the mechanism of action selection based on the causal model works, the characteristics and assumptions of the models used are presented. The variables of the each causal model are divided in three sets: state variables X, actions A and targets or subgoals Z where $x = f_x(Pa_x), x \in X$, $z = f_z(Pa_z), z \in Z$ where $Pa_x \subseteq X \cup A$ and $Pa_z \subseteq X \cup Z \cup A$. From the taxi example (explained in next section), X, A, Z can be set as follows:

$$
\begin{aligned}
X = \{&passengerPosition, onPassengerPosition, cabPosition,\\
&onDestinationPosition, destinationPosition\},\\
A = \{&pickUp, dropOff\},\\
Z = \{&inTheCab, goal\}.
\end{aligned}
$$

For our proposed method to work, the following assumptions must be satisfied:

- Non-empty set Z of target variables, can be ordered by a priority function.
- Non-empty set A of actions variables, constraints only boolean variables.

- The agent can select only one action in a given state.
- All parameters of each Causal Model are defined.

Given the knowledge describe above, the action selection is described in Algorithm 2. First we store in B the agent's observation to assign values to state variables from X. Then for each variable $z \in Z$, we explore each possible action in parents of z in the causal model, that by taking that action there is a high probability of meeting sub-goal z. We assume that interventionist and observation distributions are already given so we can query $P(z|do(a), B)$ to obtain the causal effect (line 4). Finally the selected action if exist is returned.

Algorithm 1: Causal Q-Learning

input : $< S, A, R >, G$

output: Table Q

1 Initialize $Q(s, a)$ arbitrarily
2 Repeat (for each episode):
3 Initialize s
4 Repeat (for each step of episode):
5 $a \leftarrow$ interventional based selection using (s, G)
6 If ($a = None$):
7 Choose a from s using policy derived from Q(e.g., ϵ - greedy)
8 Take action a, observe r, s'
9 $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a')) - Q(s, a))]$
10 $s \leftarrow s'$
11 until s is terminal or invalid
12 **return** Q

Algorithm 2: Action selection based on interventional queries.

Input : A state s sense by the agent, a set of causal models G, a set Z of target variables of every $g \in G$ ordered by a priority function

Output: An action a.

```
B ← get_state_observable_values(s)
foreach z ∈ Z do
    foreach a ∈ parents(z) where a is an action variable do
        p ← P(z = True|do(a = True), B)
            ▷ Here we get the causal effect on the target variable z through an intervention in the
          action variable a using the causal model g containing z.
        if p > 0.5 then
            return a
        end
    end
end
return None
```

5.1.1 Experiments

To show that our approach promises to be a way to improve RL we integrate it into the classical Q-learning algorithm. We replace the exploration step in ϵ-greedy method to choose the actions by our method that queries the model. The problem to solve is the classical taxi task described in [11]. Figure 5.1.1 graphically shows the problem. A 5×5 grid world is attended by a taxi agent. There are four locations in this world, marked as R, B, G, and Y. The taxi problem is episodic. In each episode, the taxi starts in a randomly-chosen square. There is a passenger at one of the four locations (chosen randomly), and that passenger wishes to be transported to one of the four locations (also chosen randomly). The taxi must go to the passenger's location, pick up the passenger, go to the destination location, and drop the passenger off. The episode ends when the passenger is deposited at the destination location.

There are six primitive actions in this domain: (a) four navigation actions that move the taxi one square North, South, East, or West; (b) a Pickup action; and (c) a Drop off action. The six actions are deterministic. There is a reward of -1 for each action and an additional reward of +20 for successfully delivering the passenger. There is also a 10 point penalty for illegal pick-up and drop-off actions [11]. There are 500 possible states: 25 squares, 5 locations for the passenger (including when the passenger is inside the cab), and 4 destinations.

The causal model used is presented in Figure (7). For ease, in the graphical model we got rid of the

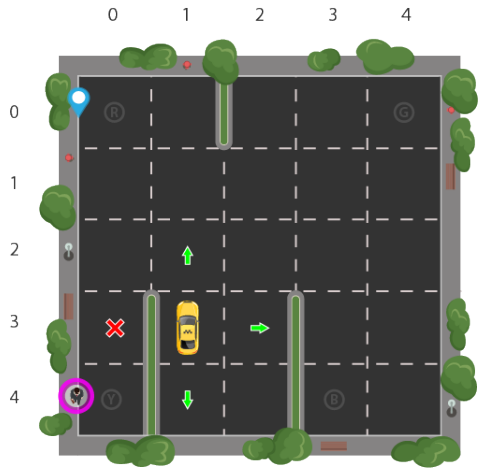

Figure 6: Sketch of the taxi environment [24].

noise variables u_i. The color of the nodes indicates to which set of variables corresponds: red for actions (A), yellow for target variables (Z) and blue for state variables (X).

$$
\begin{aligned}
pickup &= u_1 \\
dropoff &= u_2 \\
cabPosition &= u_3 \\
destinationPosition &= c_4 \\
passengerPosition &= c_5 \\
onDestinationPosition &= [(destinationPosition = cabPosition) \vee u_6] \wedge \neg u'_6 \\
onPassengerPosition &= [(passengerPosition = cabPosition) \vee u_7] \wedge \neg u'_7 \\
inTheCab &= [(pickup = True \wedge onPassengerLocation = True) \\
&\quad \vee u_8)] \wedge \neg u'_8 \\
goal &= [(dropoff = True \wedge inTheCab = True \wedge \\
&\quad onDestinationLocation = True) \vee u_9] \wedge \neg u'_9.
\end{aligned}
\tag{5}
$$

As our baseline we implement a vanilla version of the Q-learning algorithm and we compare it with our version which we denominate Q-learning + Causal Model (CM). We run 50 times each version of the algorithm and in each execution we compute the average reward per episode. Also, we set a qualifying mark based on the one established by Open AI Gym [6]. For this, we consider that the algorithm had reached an optimal reward once the average reward is equal to 9. So we assume that the algorithm that achieve it in a

[6] https://gym.openai.com/envs/Taxi-v1/

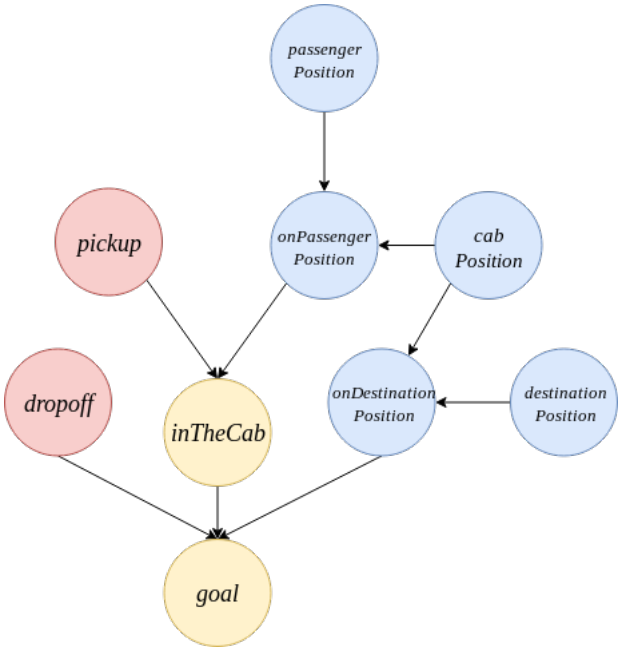

Figure 7: Causal structure D for set of equations 5. The color of the nodes indicates to which set of variables corresponds. Red for actions (A), Yellow for target variables (Z) and blue for state variables (X). (Best seen in color.)

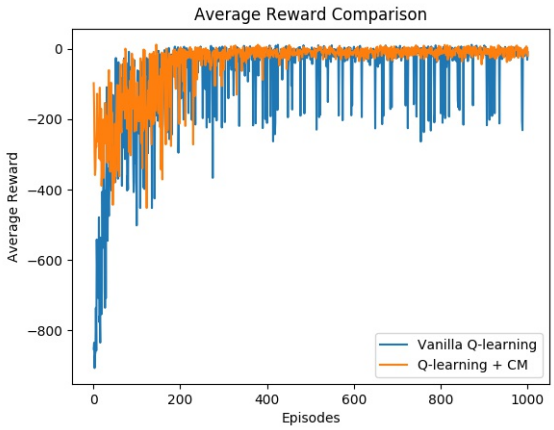

Figure 8: Average reward of Vanilla Q-learning and Q-learning guided by a causal model.

smaller number of episodes is faster.

5.1.2 Results

On average, vanilla Q-learning reaches the optimal reward in 95 episodes and Q-learning + CM in 65 episodes. From Figure 8 we can observe that our guided version starts with a higher reward. This is to be expected because, the agent doesn't start blindly. For a range of episodes there is no difference between the methods. However, after a couple of hundred episodes, the Q-learning guided by a causal model seems to converge and stays more stable. In order to validate the results that the guided Q-learning version of the algorithm performs better than the vanilla version, we use the Wilcoxon Mann-Whitney rank sum test[7] with $p < 0.001$ to find statistical significant differences. The test concluded that there are significant differences in favor of our implementation.

5.1.3 Conclusions of this experiment

Reinforcement Learning has proved to be successful in decision making problems. On the other hand, causal inference is clearly a novel but relevant and related area with untapped potential for any learning task. The use of causal models to provide auxiliary knowledge to an RL algorithm is a barely explored area. However, from the results obtained, we can see that this type of knowledge has the potential to accelerate RL. Although the problem attacked is simple because all the causes we have are direct and observable, the experimental results show that using causal models in the Q-learning action selection step leads to higher and faster jump-start reward and convergence, respectively. However, we still see how we can use partial

information (i.e. partial or incomplete casual models) to accelerate learning. This will be very important when integrating the causal discovery and learning stages of the task into a single algorithm.

5.2 Using Reinforcement Learning for Causal Discovery

As step 2 of our methodology suggests, a first stage for Causal Discovery using Reinforcement Learning data will be using observational data. By observational data we understand data generated by the agent while performing actions focused on finding the optimal policy, following some reinforcement learning algorithm, without any external manipulation of the system (for example, to force the agent to take a specific action at a given state or intentionally modify the environment at some time).

Based on that, we design a set of experiments to see, how existing causal discovery algorithms perform with data generated by an agent during a *pick and place* task learning. In the selection of the task we take into account that it is easy to validate the resulting causal model by an expert just checking if each of the suggested connections makes sense. We conduct experiments on two different models using two existing algorithms for casual discovery based on observational data. For now, each algorithm runs after all episodes of the given RL task.

5.2.1 The Reinforcement Learning task

The *pick and place* task is to move the end effector of the robot (in this case the arm) to the position of an object, pick it up and then move it to a target position and release it. It is a simple task but at the same time very similar to that of the taxi problem discussed in the previous section, so that the casual relationships obtained in this experiment could be easily compared with the input casual knowledge of the taxi task.

In the experiments, the agent was placed in a simple grid world of dimensions (3×3) with an object located in the position $(1, 1)$ and the goal is to take it to the position $(2, 2)$. The possible actions are $pick$, $move$ and $place$.

A state is represented by a tuple $< a, o, g, h >$ indicating the arm position (a), the object position (o), the gripper status (g) either open or close, and holding (h) indicating the presence or not of an object held by the arm.

A reward of 10.0 is given to the agent for grasping the object in the correct position, while 100.0 is given for placing it in the target position, -1 is given for each move, -50 for out of position pick or place and -100 is given for leaving the effective area.

100000 iterations of the Q-learning algorithm were executed using the values of $\alpha = 0.8$ and $\gamma = 0.2$

which are sufficient to learn the right policy to perform the task.

5.2.2 The candidates models

At the same time that the agent is learning the policy in an iterative way, on each episode we can choose which variables to measure to be part of the causal model we want to discover. Different aspects were taken into account for the selection of the variables of the candidate models: (i) that they were easily sensed by the agent, (ii) that they included actions and rewards, (iii) that they provided information about the stage of the task in the case of tasks that include more than one sub-goal. For now we only consider models with discrete variables. We have selected two cases with different variables that are explained below:

Case 1: In the first case we intend to capture the relationships between the value of the variables in an instant of time i with the same variables in next time j in addition to the action variables and the immediate reward obtained (positive or negative). The idea to capture the dynamics of the system between two consecutive time steps is similar to the Dynamic Bayesian Networks(DBNs) [10] but here we add a binary variable for each action and also for the reward. The following variables are kept: arm position (A_i, A_j), object position (O_i, O_j), gripper status (G_i, G_j) either open or closed, holding (H_i, H_j) indicating if the arm is holding the object at the given time, $(Move, Pick, Place)$ one for each possible action, and the immediate reward obtained either positive (R_p) or negative(R_n).

Assuming that the causal relation between the selected variables can be represented by a DAG, the expected model is presented in Figure 9.

Case 2: In this model, variables equivalent to those used in the taxi example were selected. Here we do not take into account the time but the relation between the state variables, the actions, the reward and the step of the task (before picking and after picking the object). The variables are used to indicate: (A) the arm position, (O) the object position, (G) the griper status (open or close), (H) if the arm is holding the object, (On_pick) if the arm is on the right pick position (that corresponding to the object), (On_place) if the arm is on the right place position (the one where the object must be released), $(Move)$, $(Pick)$ and $(Place)$ for actions, $(SGoal)$ that the agent reach the first subgoal (pick up the object), $(Goal)$ successful completion of the task, and (R_p) and (R_n) positive and negative reward respectively.

Assuming that the causal relation between the selected variables can be represented by a DAG, the expected model is presented in Figure 10.

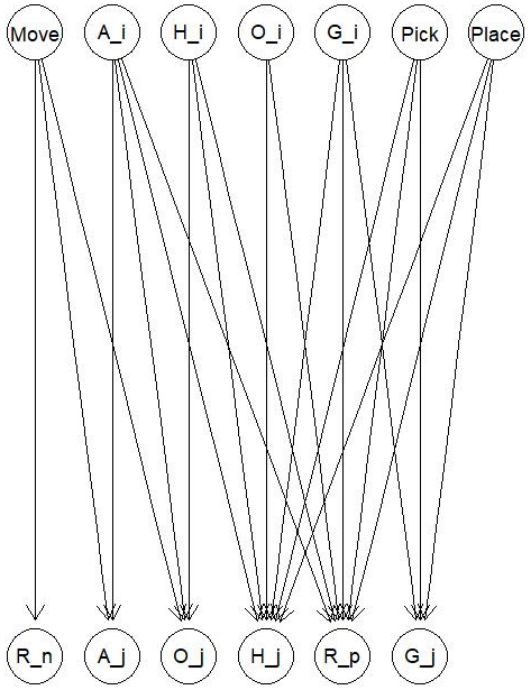

Figure 9: Expected model for Case 1

5.2.3 Causal Discovery Algorithms

We test two existing algorithms for causal discovery, PC [47] and GES [6].

PC-algorithm (named after its inventors Peter Spirtes and Clark Glymour) estimates a completed partially directed acyclic graph of the true causal structure, assuming no hidden confounders and i.i.d data. Specifically we use the pcalg package implementation in R language.

The algorithm works in three steps. The algorithm starts with a complete undirected graph. Then, for each edge (say, between a and c) the constraint is tested, whether there is any conditioning set s, so that a and c are conditional independent given s. If such a set (called a separation set or $sepset(a, c)$) is found, the edge between a and c is deleted. In the second step, unshielded triples are oriented. An unshielded triple are three nodes a, b and c with ab,bc but a and c are not connected. If node b is not in $sepset(a, c)$, the unshielded triples abc is oriented into an unshielded collider $a \rightarrow b \leftarrow c$. Otherwise b is marked as a non-collider on abc. In the third step, the partially directed graph from step two is checked using three rules to see if further edges can be oriented while avoiding new unshielded colliders (all of them were already found in step two) or cycles (which is forbidden in a DAG).

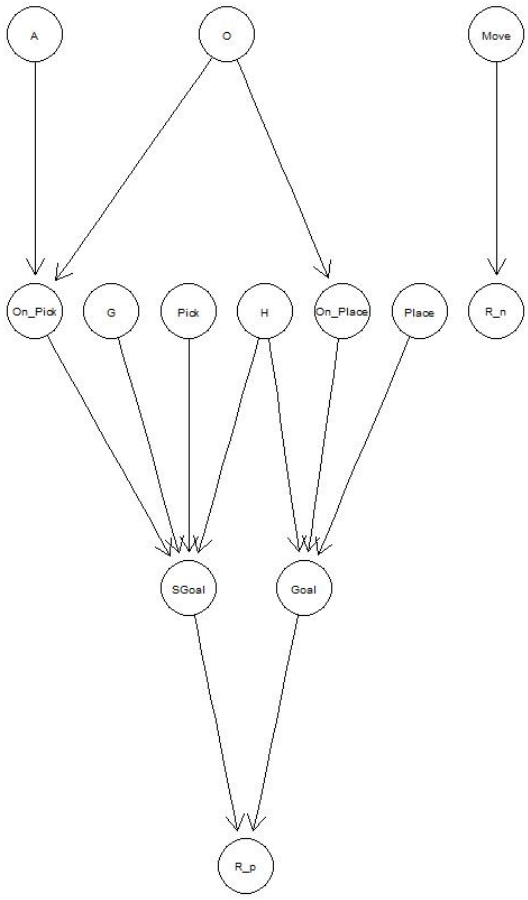

Figure 10: Expected model for Case 2

An important feature of PC-algorithm is that it allows us to define a set of constraints in the edges of the returning graph. This can be seem as to incorporate expert knowledge about the causal relations. For example connections are not possible between two action variables $(Move, Pick, Place)$, because it is impossible that an action cause another action, instead it can cause a new state or reward. As a result of this constraint, no edges between two action variables are present in the resulting graph.

The PC algorithm presented so far is based on conditional independence tests. Score-based methods form an alternative approach to causal inference. They try to find a completed partially directed graph (CPDAG) that maximizes a score, typically a model selection criterion, which is calculated from data. The greedy equivalence search (GES) makes the maximization of the Bayesian Information Criteria (BIC) computationally feasible for much larger graphs. As the name of the algorithm implies, GES maximizes the BIC in a greedy way, but still guarantees consistency in the large-sample limit. It still has exponential-time complexity in the worst case, but only polynomial complexity in the average case where the size of the

largest clique in a graph grows only logarithmically with the number of nodes [20].

GES greedily optimizes the BIC in two phases: (i) in the **forward** phase, the algorithm starts with the empty graph. It then sequentially moves to larger CPDAGs by operations that correspond to adding single arrows in the space of DAGs. This phase is aborted if no augmentation of the BIC is possible anymore. (ii) in the **backward** phase, the algorithm moves again into the direction of smaller graphs by operations that correspond to removing single arrows in the space of DAGs. The algorithm terminates as soon as no augmentation of the BIC is possible any more.

5.2.4 Experiments and Results

For our experiments we use the R package pcalg [23] implementation of PC algorithm and custom implementation in R of GES.

Figures 11 and 12 shows the resulting causal graph of our experiments compared with the expected model for cases 1 and 2 using PC and GES algorithms respectively.

As we can see, the performance of both algorithms (number of correct connections discovered) is quite low for both cases. Obtaining slightly higher results in case 2. This suggests the need for targeted interventions to improve the discovery but also indicates the difficulty in selecting the variables of the model. However, it has recently shown that even partial causal models can accelerate the reinforcement learning process [16].

5.3 Combining Reinforcement Learning and Causal Discovery

The main hypothesis of our work is that we can integrate the steps of causal discovery and a policy/value function learning into one. In Algorithm 3 we present our preliminary idea about how this combination may be possible [7]. We take inspiration in the Dyna-Q algorithm [48]. In this algorithm a model-based system was used to produce simulated experience from which a model-free system could learn. In our case the model would be a causal model rather than a deterministic transition and reward function as in Dyna-Q.

Initially (lines 2-3) the algorithm would collect a number of observations where the agent does not intervene in the environment (does not perform any action). The objective of this is to try to capture the natural dynamics of the environment. These observations can be useful to discern between causal effects caused by the action of the agent and effects of the environment. Later (lines 5-6) the agent would start to

[7] This is not a preliminary result strictly speaking because it has not yet been implemented. However, we think it is appropriate to mention the ideas we have about it.

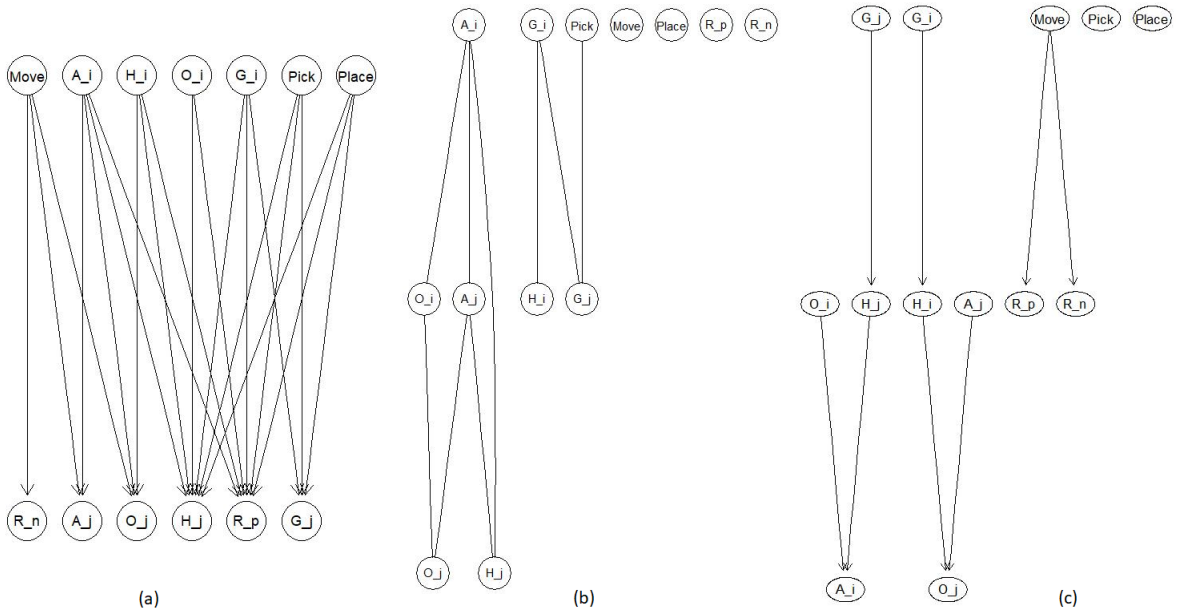

Figure 11: Obtained Causal DAG and expected model (a) for case 1 using PC (b) and GES (c)

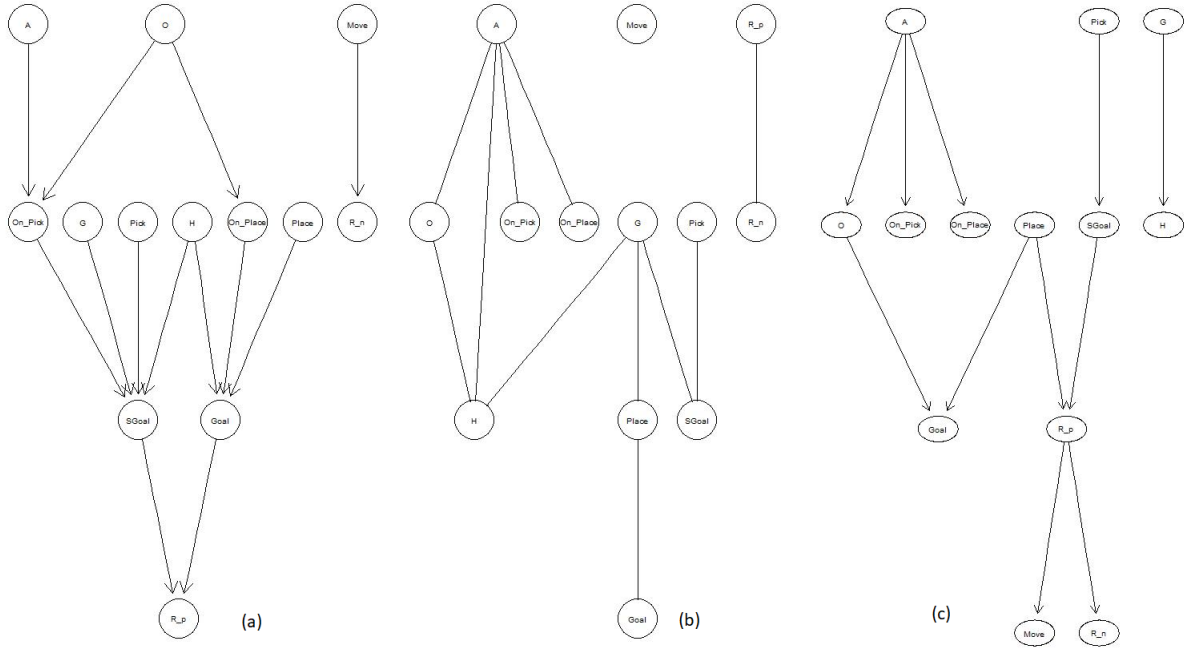

Figure 12: Obtained Causal DAG and Ground Truth (a) for model 2 using PC (b) and GES (c)

Algorithm 3: Possible algorithm for simultaneously learn a policy and causal model

input : Domain knowledge in form of constraints in the causal graph D (optional)

output: A value function Q, a causal model G

```
1  while True do
2      for i ← 0 to n_steps do
3          G ← observe_the_environment()
4      end
5      for i ← 0 to m_steps do
6          Q ← reinforcement_learning()
7      end
8      G ← causal_discovery()
9      for i ← 0 to l_steps do
10         G ← model_improvement()
11         Q ← rl_using_causal_model(G)
12     end
13 end
14 return Q, G
```

explore the environment using a classical strategy (e-greedy) while performing model-free reinforcement learning (e.g. Q-Learning). During these episodes, data on the variables of interest for the causal model would also be collected. Then (line 8) a first stage of causal discovery would be performed with the data collected from the previous stages. At this point we would have a partial and incomplete initial model but where it is likely to find some useful relationships. Finally we would proceed to improve the partial model obtained on the basis of interventions (line 10) and learning by reinforcement using the causal model (line 11). For the selection of these interventions, we would take into account those that contribute more to the discovery of the model but that also have good rewards for the agent. Then we move on to the stage of reinforcement learning based on the model obtained so far. The whole process would be repeated until we can not improve neither the policy nor the causal model.

It is important to mention that there are still several aspects to be resolved at this point. For example: how and when to intervene?, how and when to explore with a partial causal model?, what will be the set of assumptions and restrictions of the obtained model? and of course, there are still many experiments to be done.

6 Final Remarks

In spite of the progress in the areas of Reinforced Learning and Causal Models (both fundamental to the development of intelligent agents), today the first works focused on the combination of both areas are just beginning to appear. In this research we have proposed that an autonomous agent that faces an uncertain environment governed by an unknown casual mechanism can discover and use causal information while it learns a policy that maximized its reward using a reinforcement learning algorithm and at the same time, it can perform actions that focus on discovering the underlying causal structure. The preliminary results obtained so far (especially result 1) is a good evidence to validate that part of our hypothesis ($CM \rightarrow RL$) is probably true. Furthermore, we have been able to prove that using only observational data it is not possible to obtain much valuable information for causal discovery. In this sense we think that the use of guided interventions (natural in robotics) can help. Finally we have presented the preliminary ideas of how to do the combination of causal discovery and task learning. There are still several aspects to be solved, however, we consider that the continuity of the present research could lead to a novel results both in the area of reinforcement learning and in the area of causal modeling.

References

[1] Bernard W Balleine and Anthony Dickinson. "Goal-directed instrumental action: contingency and incentive learning and their cortical substrates." In: *Neuropharmacology* 37.4-5 (1998), pp. 407–419.

[2] Elias Bareinboim. *Causal Reinforcement Learning*. https://crl.causalai.net/. 2019.

[3] Elias Bareinboim, Andrew Forney, and Judea Pearl. "Bandits with Unobserved Confounders: A Causal Approach." In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes et al. 2015, pp. 1342–1350. URL: http://papers.nips.cc/paper/5692-bandits-with-unobserved-confounders-a-causal-approach.

[4] Jonathan Baxter and Peter L Bartlett. "Infinite-horizon policy-gradient estimation." In: *Journal of Artificial Intelligence Research* 15 (2001), pp. 319–350.

[5] R Bellman. "Dynamic programming: Princeton univ. press." In: *NJ* 95 (1957).

[6] David Maxwell Chickering. "Learning equivalence classes of Bayesian-network structures." In: *Journal of machine learning research* 2.Feb (2002), pp. 445–498.

[7] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. *A Hitchhiker's Guide to Statistical Comparisons of Reinforcement Learning Algorithms*. 2019. arXiv: 1904.06979 [stat.ME].

[8] Ishita Dasgupta et al. "Causal Reasoning from Meta-reinforcement Learning." In: *CoRR* abs/1901.08162 (2019). arXiv: 1901.08162. URL: http://arxiv.org/abs/1901.08162.

[9] Nathaniel D Daw, Yael Niv, and Peter Dayan. "Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control." In: *Nature neuroscience* 8.12 (2005), pp. 1704–1711.

[10] Thomas Dean and Keiji Kanazawa. "A model for reasoning about persistence and causation." In: *Computational intelligence* 5.2 (1989), pp. 142–150.

[11] Thomas G. Dietterich. “Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition.” In: *J. Artif. Int. Res.* 13.1 (Nov. 2000), pp. 227–303. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=1622262.1622268.

[12] Ray J Dolan and Peter Dayan. “Goals and habits in the brain.” In: *Neuron* 80.2 (2013), pp. 312–325.

[13] Sašo Džeroski, Luc De Raedt, and Kurt Driessens. “Relational reinforcement learning.” In: *Machine learning* 43.1-2 (2001), pp. 7–52.

[14] Frederick Eberhardt. “Causation and intervention.” In: (2007).

[15] Damien Ernst, Pierre Geurts, and Louis Wehenkel. “Tree-based batch mode reinforcement learning.” In: *Journal of Machine Learning Research* 6.Apr (2005), pp. 503–556.

[16] Ivan Feliciano Avelino. “Incorporando Conocimiento Causal en Aprendizaje por Refuerzo.” Instituto Nacional de Astrof ísica, Óptica y Electrónica (INAOE), 2020.

[17] RA Fisher. “The Design of Experiments (Hafner, New York).” In: (1935).

[18] Samuel J Gershman. “Reinforcement learning and causal models.” In: *The Oxford handbook of causal reasoning* 1 (2017), p. 295. DOI: 10.1093/oxfordhb/9780199399550.013.20.

[19] Mauricio Gonzalez-Soto, Luis Enrique Sucar, and Hugo Jair Escalante. “Playing against nature: causal discovery for decision making under uncertainty.” In: *arXiv preprint arXiv:1807.01268* (2018).

[20] Geoffrey R Grimmett and Colin JH McDiarmid. “On colouring random graphs.” In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 77. 2. Cambridge University Press. 1975, pp. 313–324.

[21] Seng-Beng Ho. “Causal Learning versus Reinforcement Learning for Knowledge Learning and Problem Solving.” In: *The Workshops of the The Thirty-First AAAI Conference on Artificial Intelligence, Saturday, February 4-9, 2017, San Francisco, California, USA*. Vol. WS-17. AAAI Workshops. AAAI Press, 2017. URL: http://aaai.org/ocs/index.php/WS/AAAIW17/paper/view/15182.

[22] Patrik O Hoyer et al. “Nonlinear causal discovery with additive noise models.” In: *Advances in neural information processing systems*. 2009, pp. 689–696.

[23] Markus Kalisch et al. “Causal inference using graphical models with the R package pcalg.” In: *Journal of Statistical Software* 47.11 (2012), pp. 1–26.

[24] Satwik Kansal. *Reinforcement Q-Learning from Scratch in Python with OpenAI Gym*. 2018. URL: https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/.

[25] Ken Kansky et al. “Schema networks: Zero-shot transfer with a generative causal model of intuitive physics.” In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1809–1818.

[26] Kevin B. Korb et al. “Varieties of Causal Intervention.” In: *PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004, Proceedings*. Ed. by Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap. Vol. 3157. Lecture Notes in Computer Science. Springer, 2004, pp. 322–331. DOI: 10.1007/978-3-540-28633-2_35. URL: https://doi.org/10.1007/978-3-540-28633-2%5C_35.

[27] Michail G Lagoudakis and Ronald Parr. “Least-squares policy iteration.” In: *Journal of machine learning research* 4.Dec (2003), pp. 1107–1149.

[28] Finnian Lattimore, Tor Lattimore, and Mark D Reid. “Causal bandits: Learning good interventions via causal inference.” In: *Advances in Neural Information Processing Systems*. 2016, pp. 1181–1189.

[29] Sanghack Lee and Elias Bareinboim. “Structural Causal Bandits with Non-Manipulable Variables.” In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 4164–4172. DOI: 10.1609/aaai.v33i01.33014164. URL: https://doi.org/10.1609/aaai.v33i01.33014164.

[30] Sanghack Lee and Elias Bareinboim. “Structural causal bandits: where to intervene?” In: *Advances in Neural Information Processing Systems*. 2018, pp. 2568–2578.

[31] Chaochao Lu. *Introduction to CausalRL*. 2019. URL: https://causallu.com/2018/12/31/introduction-to-causalrl/.

[32] Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. "Deconfounding Reinforcement Learning in Observational Settings." In: *CoRR* abs/1812.10576 (2018). arXiv: 1812.10576. URL: http://arxiv.org/abs/1812.10576.

[33] Prashan Madumal et al. "Explainable reinforcement learning through a causal lens." In: *arXiv preprint arXiv:1905.10958* (2019).

[34] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning." In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[35] Suraj Nair et al. "Causal Induction from Visual Observations for Goal Directed Tasks." In: *arXiv preprint arXiv:1910.01751* (2019).

[36] Andrew Y Ng and Michael I Jordan. "Approximate inference a lgorithms for two-layer bayesian networks." In: *Advances in neural information processing systems*. 2000, pp. 533–539.

[37] Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. "A hybrid causal search algorithm for latent variable models." In: *Conference on Probabilistic Graphical Models*. 2016, pp. 368–379.

[38] J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Penguin Books Limited, 2018. ISBN: 9780241242643.

[39] Judea Pearl. *Causality*. Cambridge university press, 2009.

[40] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*. Vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[41] Rajat Sen et al. "Identifying best interventions through online importance sampling." In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 3057–3066.

[42] Shohei Shimizu et al. "A linear non-Gaussian acyclic model for causal discovery." In: *Journal of Machine Learning Research* 7.Oct (2006), pp. 2003–2030.

[43] David Silver et al. "Mastering the game of go without human knowledge." In: *nature* 550.7676 (2017), pp. 354–359.

[44] Satinder P Singh and Richard S Sutton. "Reinforcement learning with replacing eligibility traces." In: *Machine learning* 22.1-3 (1996), pp. 123–158.

[45] Peter L Spirtes, Christopher Meek, and Thomas S Richardson. "Causal inference in the presence of latent variables and selection bias." In: *arXiv preprint arXiv:1302.4983* (2013).

[46] Peter Spirtes and Clark Glymour. "An algorithm for fast recovery of sparse causal graphs." In: *Social science computer review* 9.1 (1991), pp. 62–72.

[47] Peter Spirtes et al. *Causation, prediction, and search*. MIT press, 2000.

[48] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN: 0262193981. URL: http://www.worldcat.org/oclc/37293240.

[49] Christopher JCH Watkins and Peter Dayan. "Q-learning." In: *Machine learning* 8.3-4 (1992), pp. 279–292.

[50] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." In: *Machine learning* 8.3-4 (1992), pp. 229–256.

[51] Chao Yu et al. "Incorporating causal factors into reinforcement learning for dynamic treatment regimes in HIV." In: *BMC Med. Inf. & Decision Making* 19-S.2 (2019), pp. 19–29. DOI: 10.1186/s12911-019-0755-6. URL: https://doi.org/10.1186/s12911-019-0755-6.

[52] Jiji Zhang and Peter Spirtes. "Detection of unfaithfulness and robust causal inference." In: *Minds and Machines* 18.2 (2008), pp. 239–271.

[53] Junzhe Zhang and Elias Bareinboim. "Transfer learning in multi-armed bandit: a causal approach." In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 2017, pp. 1778–1780.

[54] Shengyu Zhu and Zhitang Chen. "Causal discovery with reinforcement learning." In: *arXiv preprint arXiv:1906.04477* (2019).