Causal based Action Selection Policy for Reinforcement Learning

Anonymous

No Institute Given

Abstract. Reinforcement learning is the *de facto* learning by interaction paradigm within machine learning. One of the intrinsic challenges of reinforcement learning is the trade-off between exploration and exploitation. To solve this problem, an agent can limit its search space by leveraging the known properties of its environment or using previous knowledge. Specifically, in this paper, we propose to improve the reinforcement learning process of an agent that can exploit causal relationships of the world. A causal graphical model helps to restrict the search space by reducing the actions that an agent can take through graph queries that check what variables are direct causes of the variables of interest. Our main contribution is a framework to represent the causal information and an algorithm to guide the action selection process of a reinforcement learning agent, by querying the causal graph. We test our approach on discrete and continuous domains and show that using the information from a causal structure in the Q-learning action selection step, leads to higher jump-start reward and stability. Furthermore, it is also shown that a better performance is obtained even with partial and spurious relationships in the causal graphical model.

Keywords: Reinforcement learning · causal graphical models · action selection.

1 Introduction

One of the goals of artificial intelligence (AI) is to create autonomous agents that learn through interaction with their environment [2]. One framework that emerges for that purpose is *reinforcement learning* (RL), that studies how an agent can learn to choose actions that maximize its future rewards through interactions with its environment [35]. RL algorithms have been shown to be effective in various domains such as video games [25,36], robotics [29], and medical care [16]. However, most of the current reinforcement learning systems suffer from some shortcomings, in particular, they do not take advantage of high-level processes to exploit patterns beyond the associative ones [11,14]. Among these high-level processes is causal reasoning [31,34].

Causal inference (CI) is a learning paradigm concerned at uncovering the cause-effect relationships between different variables [31], [30]. CI addresses questions like: If I desire this outcome, what action do I need to take? So it can provide

the information needed for an intelligent system to predict what may happen next so that it can plan better for the future. Learning causal relations in the real world is a challenging task for which many algorithms have been proposed according to different set of constraints. However, once the causal structure is known 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 are never observed before. This paper follows the manipulationist causality theory [39], in which the fundamental idea about causality can be described intuitively as: if A is genuinely a cause of X, then if A is manipulated, this must be a way to manipulate or change X [6,38]; in particular, we use the manipulationist framework proposed by [31].

Both reinforcement learning (RL) and causal inference (CI) have evolved independently and practically with no interaction between them, despite the fact that both are directly relevant to problem solving processes. Nonetheless, recent work has focused on connecting both fields [13], [19], [8]. The goal of these works is to show how RL can be made more robust and general through causal mechanisms, known as *CausalRL* [21]. CausalRL attempts to mimic human behavior: learning causal effects from an agent communicating with the environment and then optimizing its policy based on the learned causal relations.

In this work, we focus on one way of combining both fields, that is to use causal inference to improve reinforcement learning. In particular, we focus in the trade-off dilemma in RL between trying new actions (exploration) or selecting the best action based on previous experience (exploitation). Traditional exploration and exploitation strategies are undirected and do not explicitly chase 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 There are tasks in which an expert or even the algorithm itself can learn the latent causal model: robotics, environments such as Animal AI [4], goal-directed tasks [28] and even some games [22].

We propose a method to guide action selection in RL tasks that have an underlying causal structure. The agent begins its search blindly, through trial-and-error interactions. However, the agent can, through interventions in a causal model, make queries of the type: *What if I do ...?* e.g., If I drop the passenger off here, will my goal be achieved? These interventions reduce the search space and allow the model to be used as an "oracle" to avoid performing actions that can lead to undesired states or to prefer actions that lead to a goal.

Experiments in different variants of the light switch scenario [28] show how an agent using the causal model achieves a higher reward in a shorter time compared to a traditional RL agent. Even an incomplete or partially incorrect causal model improves the performance.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the proposed method. In Section 4 the experimental set-up is described and the main results presented. Finally, in Section 6, conclusions and future research directions are given.

2 Related Work

A fundamental problem in reinforcement learning algorithms is the trade-off between exploration of the environment and exploitation of the information available to the agent, and there are several techniques to deal with this trade-off. Exploitation and exploration strategies are undirected and do not explicitly seek interesting transitions [24]. However, according to Hafner et al. [17], using prediction models seems a promising way to deal with this problem. There are several examples of using prior knowledge to guide an RL agent, as in [23,32,12,1,27]. However, there are several advantages on using causal information for agents attacking decision-making problems: (i) evaluate changes in consequences given interventions in causes, (ii) to know why a certain sequence of decisions was chosen, and (iii) to evaluate the potential impact of alternative actions (counterfactual).

The idea of using knowledge of causal models to avoid or reduce trial-and-error learning in RL is an area with little exploration but great promise. Commonly, the problem tackle by existing works is of the type *multi-armed bandit (MAB)*. Lattimore et al.[20] exploit causal information in the bandit problem and show how, through interventions, the rate at which actions with a higher reward are identified can be improved.

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.

It is shown in [20] that adding causal information in a fixed budget decision problem [1] allows the decision maker to learn faster than if it 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 [33] where the proposed system requires only that some part of the causal model is known and allow interventions over the unknown part. In [20] 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.

[1] 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

An interesting example focus on knowledge transfer in RL using causal inference tools [40]. 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. Simulations show that their strategy is consistently more efficient than the current (non-causal) state-of-the-art methods.

Another problem in RL that is being attacked with causal elements is reward tampering [10]. This problem arises when an agent focuses on collecting small rewards and avoids the behavior that leads to the larger reward. For example, an agent who has to collect diamonds, but there are rocks that also give it rewards, may be inclined to only look for the rocks and not the diamonds. The authors use a causal influence diagram to attack this problem.

In contrast to previous work, we address the general case in which a decision problem can be represented as a Markov decision process (MDP); and propose a way to represent and use causal knowledge to accelerate the learning process.

3 Proposed Method

Our work is focused on problems that can be posed as a goal-conditioned Markov decision processes. According to Nair et al. [28], this type of task has an underlying causal structure that describes the behavior of the environment. We define a goal-conditioned MDP as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{P}, \mathcal{G}, r, \gamma, \phi)$. The elements of the tuple are described as follows: $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the set of possible actions, $\mathcal{X}$ is the set of causal macro-variables[2] which describes the state of the environment at a high abstraction level (see [7]), $\mathcal{D}$ is a graph of the causal model ruling the agent's world, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the probability transition function between states given an action, $\mathcal{G}$ is the goal space, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathrm{R}$ is the reward function where $r(s, a, g)$ yields the immediate reward conditioned on the goal $g \in \mathcal{G}$, γ is the discount factor, and $\phi : \mathcal{S} \to \mathcal{X}$ is a function which associates the state space to the macro-variables space. The goal of the RL system is to learn an optimal policy $\pi^*_g : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$ such that it maximizes the expected total return $R = \sum_k^\infty \gamma^k r(s_k, a_k, g)$.

Often, an agent does not have enough information to define explicitly the transition dynamics of its environment, i.e., $\mathcal{P}$. However, there are tasks where some extra information can be given by a domain expert. This extra information can be encoded within the graph $\mathcal{D}$. An agent with a representation of the underlying causal model of the environment, e.g., $\mathcal{D}$, is able to pursue actions that lead to desired states or to avoid choosing actions that take it into unwanted states. On the other hand, in many of the current problems in which RL-based solutions are applied, the information received by the agent is hard to model.

[2] A high-level variable or macro variable is a function over a data structure, which in turn is defined from other variables [7]. These variables can be seen as a quantity that summarizes information about some aspect.

For example, some Gym [5] environments send to the agent multidimensional arrays of pixels describing RGB images. Generating a causal model that represents all the information of the environment with which the agent interacts can be intractable. However, we propose to take as an advantage the property of having a causal structure within goal-conditioned MDPs. Such causal structure describes the cause-effect relationships between actions and high-level variables that represent the observable states.

In this section, we describe our methodology to improve the performance of a classical reinforcement learning agent. Our aim is to decrease the agent's learning time guiding its actions using the extra information given by the causal model which rules its environment.

3.1 Assumptions and limitations

In this work, the following assumptions are considered to limit the scope of the proposed solution.

1. The agent knows the causal graph $\mathcal{D}$ with all or some of the causal relationships in the world. Therefore, a specification of how some variables are influenced by their parents in the causal structure $\mathcal{D}$ can be obtained.
2. It is assumed that a, possibly incomplete but partially correct, causal model is known which relates state variables or meta-variables and actions with goals, sub-goals, and undesirable states.
3. In tasks where states are continuous, the RL agent also has access to ϕ. The latter function relates states to high-level variables.
4. It is assumed that there exist a mapping between a state's description and the variables used in the causal model. This mapping may be trivial, when the causal model is described by the same state variables used by the RL agent or can include a mapping involving for instance deep networks when states are represented by images.
5. $\mathcal{D}$ is considered to be a causal graph; that is, the Markov, minimality and faithfulness conditions (described in [18]) are assumed to be satisfied.

3.2 Action selection

In the classical ϵ greedy action selection policy, the agent randomly selects, based on the ϵ value, between explore or exploit its current knowledge. Therefore, the agent balances the exploitation and exploration trade-off. With ϵ probability the agent selects a random action, and with $1 - \epsilon$ probability the agent takes the current best action. We extend the ϵ greedy strategy by adding it an extra step (ϵ is a value between 0 and 1 that weights the relationship between exploration and exploitation). In addition to the options of executing a random action or the one that seems to be the best so far, an agent can also choose to query the causal model. In this work, the proposed system is incorporated in the ϵ greedy policy. However, it is possible to use the method in other action selection frameworks.

There is a particular type of query that is exploited in the proposed approach; it allows an agent to ask: What action leads me to meet the current goal? We can define a set $E = \{x | x \in \mathcal{X}$ and x is not equal to the desired value given by $g\}$, where its elements are those variables that have not reached the desired value. The desired value of a variable is indicated by the goal g. With this query, we want to know which variables in the set of actions affect the variables of E. What this means is that performing an action, i.e., assigning a value to a $a \in \mathcal{A}$, causes one or more variables in E to change their value, bringing the agent closer to the goal. At the same time, the agent does not execute actions that would lead to an undesired change on one or more variables. By computing the set of predecessors of an effect $x \in \mathcal{X}$ we answer the previous query; so the agent prefers actions that are a parent of the variable of interest in $\mathcal{D}$.

In this work, the value of ϵ is linearly decremented throughout the agent's training. In this way, the agent starts with a high probability of using information from the causal model or exploring when there is not causal information, and eventually, the agent uses the learned policy, for instance, through the value function as in Q-learning.

A brief overview of the action selection policy is described as follows. With probability $1 - \epsilon$ the best action is chosen. For example, in the Q-learning algorithm the best action is given by that action which gives the largest value for the action value function, i.e., $\text{argmax}_a\, Q(s, a)$. Initially the probability of exploiting the best option is very low, i.e., ϵ starts with a value equal to or close to 1. If the best action is not chosen, then the causal structure is queried and if there is insufficient information in it, a random action is selected.

Because the causal structure is in terms of high-level variables, ϕ is needed to map the observation s to a vector of macro variables $\mathbf{x} = [x_1, \ldots, x_N]$, where $x_i \in \mathcal{X}$. Figure 1 shows an example of mapping a two-dimensional observation s into a vector of size N with variables describing the state at a high level.

Once the association between the state space and the high-level variables space is done, a list E of variables of interest is obtained, which will be consulted in the causal graph. For simplicity, the list E is obtained through a function f that calculates which variables have a different value between the goal g and the vector of macro variables $\mathbf{x}$. We can suppose that $\mathbf{x} \neq g$ at the first training step, however, since environments are dynamic, it cannot be assured that the initial configuration of an environment will not be the same as the desired one.

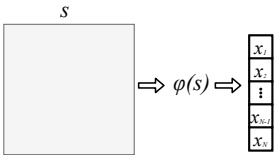


Fig. 1: The observation s of the agent's environment can be transformed by ϕ into a vector of high-level variables.

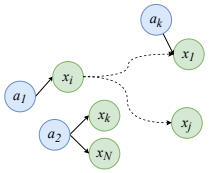


Fig. 2: Example of a causal graph $\mathcal{D}$. Arrows with curved and dotted lines denote causal paths.

In some cases, the variables of interest may follow a causal order, which can be interpreted to mean that one goal is dependent on other subgoals being met. Therefore, the list E can be stored in a data structure such that its elements are ordered by a priority function. Thus, the agent choose first those actions that lead to the final goal. Figure 3 describes an example of how the list E is produced with respect to the graph in Figure 2.

The next step corresponds to obtain an action $a \in \mathcal{A}$ for the agent to execute. For this, the agent makes a query to the causal structure. The query consists of traversing the list of elements of E. If for an element $x \in E$ we found one action variable a as a predecessor, such action is performed. We consider problems where it is not necessary to perform several actions at the same time to affect x.

4 Experimental set up

To evaluate the proposed method, we attack the light switch problem defined by Nair et al. [28] and the cab problem proposed by [9]. In this paper we include the experiments in the light switch task. In the supplementary material it is shown how the proposed method is applied to the cab problem, where the causal model is applicable only to a small subset of the state space, and still provides a clear advantage over the baseline. In summary, the experiments consist of integrating the causal graph to the classical ϵ greedy policy in the Q-learning algorithm.

For this proof of concept, we assume that the action space $\mathcal{A}$ is discrete and the values of the actions are binary, $\{0, 1\}$. This can be interpreted that the action $a \in \mathcal{A}$ is either performed or not. The elements of $\mathcal{X}$ are variables with binary values, $\{0, 1\}$. This can be seen as $x \in \mathcal{X}$ is either on or off, true or false, accomplished or incomplete.

In the ϵ greedy policy, instead of keeping the value of ϵ fixed, we propose the to start the learning by motivating the agent to use the causal model or

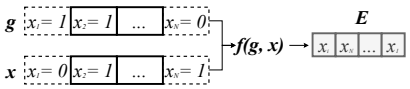


Fig. 3: E is computed finding those variables that have different values between $\mathbf{x}$ and g. x_i is stored in E if $|x_i - x'_i| = 1$ such that $x_i \in g$ and $x'_i \in \mathbf{x}$.

Fig. 4: A simple example of a trajectory of actions, from an initial lightning setting of the environment to the final lightning setting. A bird's eye view of the environment's state and the causal graph that rules the environment are shown. The blue and green vertices denote the switches and the lights, respectively. The bold blue node (on the left side) represents the manipulated variable to affect the darkest green node (on the right side). In the first case (at t_1), after manipulating the third blue node, the light is turned off in one room (represented by the fifth green node). In the second case (at t_2), after manipulating the second blue node the lights are turned on in another room (represented by the first green node).

explore. Then, we decrease ϵ to give more weight to exploitation (choose the best action according to the learned policy). Four algorithms are compared, where all of them follow the Q-learning approach. The main difference between them is the amount and quality of the information encoded in the causal graph. The following sections describe in detail the experiments performed and the results. All the software developed is available at `https://anonymous.4open.science/r/cbdbb0ba-d371-4e0b-97b6-24613aff69ac/`.

4.1 Environment

We conducted a series of experiments on the light switch control tasks testbed introduced by Nair et al. [28]. This is an episodic problem where in each episode an agent, which is provided with an initial lighting setting of the environment, aims to reach a previously specified configuration of the lights; i.e., which lights are on and which ones off. An example of an initial observation an the goal of the agent is shown in Figure 4. Specifically, an agent has control over N light switches which control N lights. The agent performs an action (flipping a switch) and the environment feeds back with a new state of the lights and switches. The relationship between switches and lights is given by a causal graphical model which defines how the former controls the latter.

The testbed involves three different types of causal relationships between the switches and lights shown in Figure 5. The three types of causal relationships are: *one to one*, where each switch controls only one light; *common cause*, where a single switch may control more than one light and each light is controlled by at most one switch; *common effect*, where each switch maps to one light, but more than one switch can control the same light.

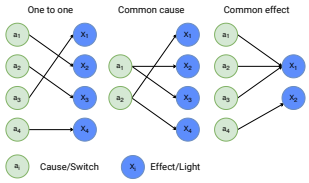


Fig. 5: Examples of the three types of latent causal structures on the light switches environment.

4.2 Implementation and compared approaches

The proposed method is incorporated into the Q-learning algorithm [37]. The training process of the Q-learning algorithm remains identical except for the guided action selection step. Four algorithms are compared, based on the Q-learning method, the difference lies in the quantity and quality of the additional information available:

- *Q-learning without additional information* (Q_1). It serves as a baseline to measure how much learning is improved. The algorithm, depending on the state space being worked on, is the classic Q-learning [37] or deep Q-learning with experience replay (DQN) [25], for discrete and continuous states, respectively. Action selection is carried out by an *epsilon* greedy policy.
- *Q-learning + complete causal structure* (Q_2). The agent has the complete and true causal structure of the environment $\mathcal{D}$.
- *Q-learning + partial causal structure* (Q_3). In this case, the agent has a subgraph $\mathcal{D}'$ of the graph $\mathcal{D}$. This subgraph is generated by randomly removing edges from $\mathcal{D}$.
- *Q-learning + incorrect causal structure* (Q_4). This algorithm queries a $\mathcal{D}''$ structure with spurious relationships and without some true links. The graph $\mathcal{D}''$ is obtained by generating a subgraph of $\mathcal{D}$ as in the previous case and randomly adding edges.

4.3 Evaluation metric and exploration rate decay

To measure the performance of the algorithms in each experiment, the average reward is evaluated over a series of simulations. Each simulation consists of running the learning algorithm for k episodes, in an environment with a fixed causal structure $\mathcal{D}$ and where the goal g is sought to be achieved. The average reward for the $i-$th. episode is given by $R^i = \frac{1}{H}\sum_{t=0}^{H} r(\mathbf{x}_t, g)$, where H corresponds to the size of the episode. The vector $\mathbf{R_i}$, of the $i-$th. experiment contains the average rewards for each episode, and is defined as $\mathbf{R_i} = (R^1, \ldots, R^k)$.

Thus, the comparison measure between algorithms is the average of the vectors $\mathbf{R_i}$, $i \in [1, M]$, obtained in M simulations. This measure, denoted as *average*, can be written as

$$average(\mathbf{R_1}, \ldots, \mathbf{R_M}) = \frac{1}{M}(\sum_{i}^{M} \mathbf{R_i^1}, \ldots, \sum_{i}^{M} \mathbf{R_i^k}), \tag{1}$$

where M is the number of simulations and $\mathbf{R_i^j}$ indicates the average reward obtained in the $j-$th. episode of the $i-$th. simulation.

The parameter ϵ decreases linearly, where at each action selection it decreases until a minimum value is reached. The ϵ update rule at time step t can be defined as $\epsilon = \max(\epsilon_{\min}, \epsilon_{\max} - \frac{|\epsilon_{\max} - \epsilon_{\min}|}{H \times k \times \delta} \times t)$, where $H = N$ and $0 < \delta \leq 1$, is a factor to control how fast the minimum ϵ value is reached, the closer to 0, the faster the exploration stage finishes.

5 Results

Three experiments are carried out, and for the first two experiments, we directly provide the agent with the set of high-level variables, such that it works with a tabular version of the Q-learning algorithm. In the third experiment, we do not have the aforementioned set, so we only have access to $s \in \mathcal{S}$. Thus, for the latter case, we use the DQN [26] algorithm to deal with images as inputs.

The first experiment aims to measure the performance when modifying the causal structure $\mathcal{D}$ at different percentages to obtain $\mathcal{D}'$ and $\mathcal{D}''$. In the second experiment we propose to change the decrease rate of ϵ to get faster or slower to the constant exploitation phase. In the last experiment, we want to test the algorithm when the high-level variables are not available as direct observations, therefore, it works on a continuous state space.

5.1 Modifying the causal graph

The goal of this experiment is to determine whether the information provided by an incomplete or partially incorrect model helps and does not negatively affect the performance of the RL algorithm. These subgraphs are generated from the $\mathcal{D}$, altering it to different levels. We modify $\mathcal{D}$ at three levels, to obtain the graphs $\mathcal{D}'$ and $\mathcal{D}''$. The percentage level of change is represented by the parameter p_{mod}. For each level, the $\mathcal{D}'$ subgraph is generated by removing a percentage p_{mod} of the edges of the $\mathcal{D}$ graph. To produce $\mathcal{D}''$, after removing edges in a similar way, from half of the missing connections, new ones different from the initial ones are created. We use three values to test the modification level: low, medium and high, with $p_{mod} = 25\%$, $p_{mod} = 50\%$, and $p_{mod} = 75\%$, respectively.

The experiment is run on environments with the three possible types of structures: one to one, common cause, and common effect. The exploration rate is given by a $\delta = 0.5$, indicating that approximately halfway through the training, the probability of exploitation reaches its maximum value.

The results are summarized in Figure 6. We can observe that in most cases where the algorithms use knowledge from the causal model ruling the agent's

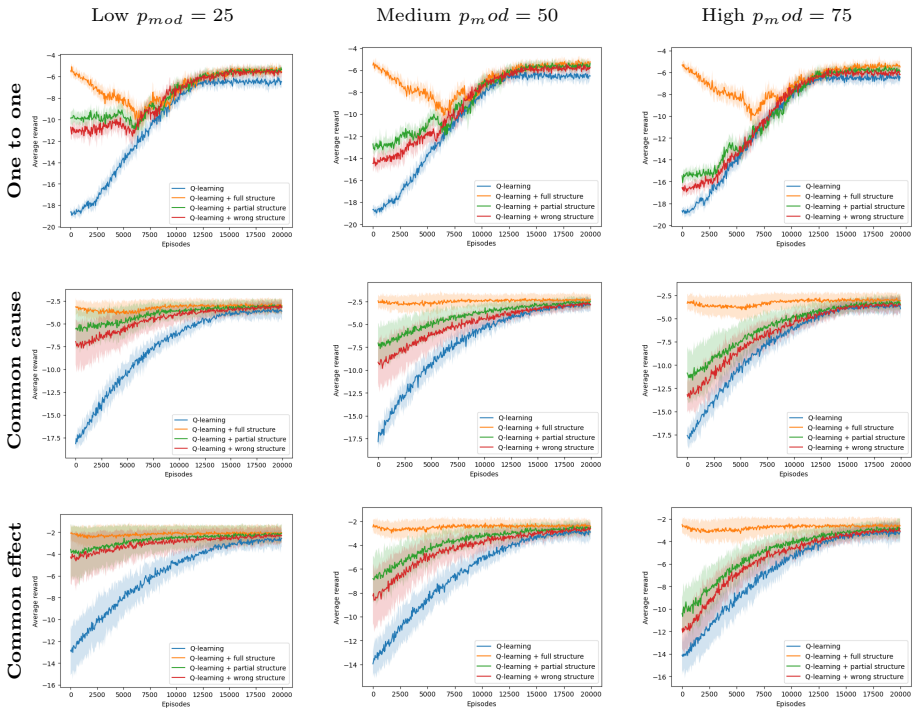


Fig. 6: The average value and standard deviation of the episodic reward over 10 simulations and $N = 9$, for the light switches scenario. The plots show the average reward for each of the three types of structures and different values of p_{mod}, to compare the four algorithms. The vertical and horizontal axis correspond to the value of the evaluation metric and the episodes, respectively.

environment, there is a higher jumpstart and a faster converge than the Q-learning algorithm without additional information. In general, it can be seen that the algorithms Q_3 and Q_4 behave similarly. For the case where $p_{mod} = 25$, the algorithms with incomplete and incorrect information have a similar performance as the algorithm with the complete causal model, and the difference with the basic Q-learning is significant. Something similar occurs for $p_{mod} = 50$. For $p_{mod} = 75$, despite having modified the causal graph by a fairly high percentage, the small amount of information that remains and is correct, is enough to reach a higher reward much faster. It is most noticeable for common cause and common effect type structures. There is a strange behavior in tasks with a one to one type structure, which might be related to the exploration strategy.

5.2 Exploit or keep exploring

This section shows experiments to compare the performance of the algorithms with two different values for the factor controlling the decrement rate of ϵ. We

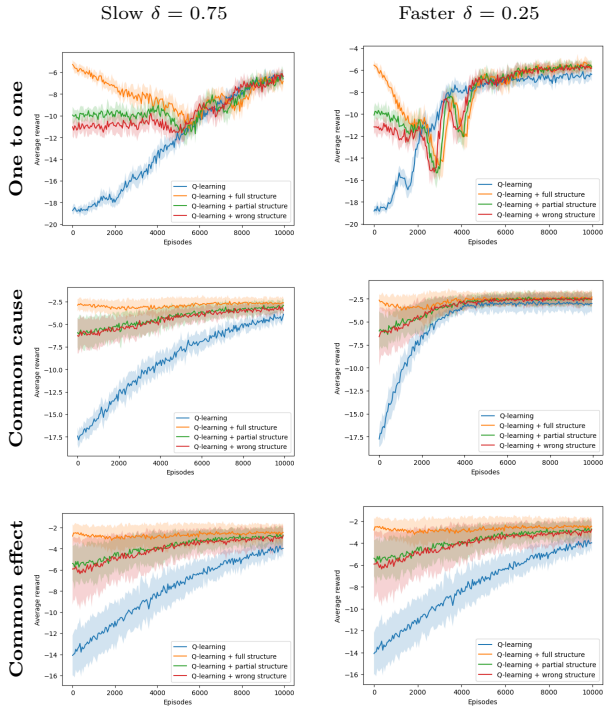


Fig. 7: Average value and standard deviation of the reward per episode over 10 simulation and $N = 9$, for the light switches scenario. The plots show the average reward for each of the three types of structures and different values of δ, to compare the four algorithms. The vertical and horizontal axis in each plot correspond to the value of the evaluation metric and the episodes, respectively.

control how fast we want to reach ϵ_{min} by varying δ. If the training time is divided into quarters, then the values of δ correspond to the quarter in which the minimum value for ϵ is reached. We can reach the ϵ_{min} value at different training times. To reach that value at the first, second, and third quarter of the training, we set δ to 0.25, 0.50, and 0.75, respectively Since in Section 5.1, $\delta = 0.5$, we do include the results for this value in this set of experiments. Our objective is to determine whether reducing or increasing causal graph queries throughout the learning process affects the performance of the algorithms. To get the graphs $\mathcal{D}'$ and $\mathcal{D}''$, the percentage of change is set to $p_{mod} = 25$.

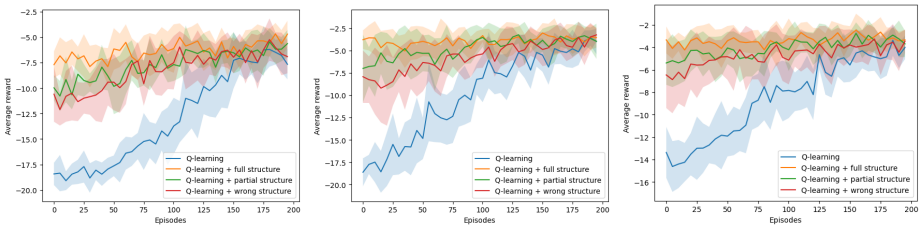
From the learning curves in Figure 7 it can be observed that the higher δ, the longer it takes to stabilize the learning algorithm without information to help it. This is expected, since it continues exploring for a longer time. In most cases, the Q_{2-4} algorithms are faster than Q_1, in the various environment configurations. The cases with almost no difference correspond to when it finishes the exploration stage earlier ($\delta = 0.25$).

Fig. 8: Example of a possible observation of the agent.

5.3 Using visual observations of the environment

In this experiment the agent does not have access to the macro variables $\mathcal{X}$ directly. However, it receives images of the state of the environment as observations. The objective is to determine whether the causal model with variables in another space still retains the capacity of accelerating learning as in the discrete cases.

The observations are images of 84×84 pixels in RGB color space, obtained from a eye's bird view of the environment (see Figure 8). To associate the images to the high-level state space $\mathcal{X}$, the ϕ function is a multi-label classifier parameterized by a convolutional neural network [15]. The agent is provided with the classifier already trained. The outputs of the network represent the probability p_i that the variable x_i takes the value 1, where $i \in \{1, \ldots, l\}$, with $l = 9$. The causal graph alteration parameter value is $p_{mod} = 25$. The rate of decrement of *epsilon* is controlled by the factor $delta = 0.75$. Since the observations are images, the DQN steady-state version of the Q-learning algorithm is used. The architecture and training hyperparameters are the same as those in the original DQN paper[26]. The results depicted in Figure 9 show that the algorithms using knowledge from the causal structures start with a higher reward (jumpstart) and stabilize faster than the DQN algorithm without additional information, in all three types of underlying structures.



(a) One to one (b) Common cause (c) Common effect

Fig. 9: Average reward and standard deviation per episode over 10 simulations for the continuous state space experiment. Each figure compares the performance of the four algorithms for the three types of structures. The vertical and horizontal axis indicates the value of the reward and the episodes number, respectively.

6 Conclusions

A methodology to guide the interaction step of an RL agent was presented. To test the proposed concept we use the light switch task proposed by [28]. For the experiments we integrated the causal graph governing the agent's world, or at least a part of it, into the action selection policy. The method was tested in different experimental settings: modifying the causal network at different levels to have cases where only little or partially incorrect information is available, varying the rate at which the causal model is no longer consulted, and using images of the state of the environment as observations. Based on the results of the experiments, the following conclusions were reached:

- Incorporating causal knowledge into RL accelerates the learning process. This is relevant, not because we are adding knowledge, which is expected to improve performance, but because the knowledge is expressed in terms of a causal model which can be used for explanations and reused in similar tasks.
- Providing an agent with a graph that preserves few true causal relationships continues to perform better than without guiding its choice of actions. This is important for learning at the same time the causal model.
- Reducing the probability of querying the causal graph at an early stage of training, i.e., giving more weight to exploitation than to exploration, does not affect the agent's performance too much. This could be because the little guided exploration experienced by the agent may have already biased its behavior.

The results indicate that the presented methodology is a suitable alternative for attacking interaction tasks, where the environment is governed by a causal model. Using the causal graph as a means of querying the information of the latent causal model can reduce the learning time with respect to a trial-and-error interaction with the environment. The proposed approach was implemented on commonly used reinforcement learning algorithms, Q-learning and DQN, however, it can be easily transfer to other algorithms. It is left as future work how to learn a causal model while using it to make decisions.

References

1. Abel, D., Hershkowitz, D., Barth-Maron, G., Brawner, S., O'Farrell, K., MacGlashan, J., Tellex, S.: Goal-based action priors. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 25 (2015)
2. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866 (2017)
3. Bareinboim, E., Forney, A., Pearl, J.: Bandits with unobserved confounders: A causal approach. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. pp. 1342–1350 (2015), http://papers.nips.cc/paper/5692-bandits-with-unobserved-confounders-a-causal-approach

4. Beyret, B., Hernández-Orallo, J., Cheke, L., Halina, M., Shanahan, M., Crosby, M.: The animal-ai environment: Training and testing animal-like artificial cognition (2019)
5. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym (2016)
6. Campbell, D.T., Cook, T.D.: Quasi-experimentation: Design & analysis issues for field settings. Rand McNally College Publishing Company Chicago (1979)
7. Chalupka, K., Perona, P., Eberhardt, F.: Visual causal feature learning. In: Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence. p. 181–190. UAI'15, AUAI Press, Arlington, Virginia, USA (2015)
8. Dasgupta, I., Wang, J.X., Chiappa, S., Mitrovic, J., Ortega, P.A., Raposo, D., Hughes, E., Battaglia, P., Botvinick, M., Kurth-Nelson, Z.: Causal reasoning from meta-reinforcement learning. CoRR **abs/1901.08162** (2019), `http://arxiv.org/abs/1901.08162`
9. Dietterich, T.G.: Hierarchical reinforcement learning with the maxq value function decomposition. J. Artif. Int. Res. **13**(1), 227–303 (Nov 2000), `http://dl.acm.org/citation.cfm?id=1622262.1622268`
10. Everitt, T., Hutter, M.: Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective (2019)
11. Garnelo, M., Arulkumaran, K., Shanahan, M.: Towards deep symbolic reinforcement learning (2016)
12. Geibel, P.: Reinforcement learning with bounded risk. In: In Proceedings of the Eighteenth International Conference on Machine Learning. pp. 162–169. Morgan Kaufmann (2001)
13. Gershman, S.J.: Reinforcement learning and causal models. The Oxford handbook of causal reasoning p. 295 (2017)
14. Gonzalez-Soto, M., Sucar, L.E., Escalante, H.J.: Playing against nature: causal discovery for decision making under uncertainty. CoRR **abs/1807.01268** (2018), `http://arxiv.org/abs/1807.01268`
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), `http://www.deeplearningbook.org`
16. Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, L., Ding, Y., Wihl, D., Peng, X., Yao, J., Lage, I., Mosch, C., wei H. Lehman, L., Komorowski, M., Komorowski, M., Faisal, A., Celi, L.A., Sontag, D., Doshi-Velez, F.: Evaluating reinforcement learning algorithms in observational health settings (2018)
17. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination (2019)
18. Hitchcock, C.: Causal models. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, summer 2019 edn. (2019)
19. Ho, S.: 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. AAAI Workshops, vol. WS-17. AAAI Press (2017), `http://aaai.org/ocs/index.php/WS/AAAIW17/paper/view/15182`
20. Lattimore, F., Lattimore, T., Reid, M.D.: Causal bandits: Learning good interventions via causal inference. In: Advances in Neural Information Processing Systems. pp. 1181–1189 (2016)

21. Lu, C., Schölkopf, B., Hernández-Lobato, J.M.: Deconfounding reinforcement learning in observational settings. CoRR **abs/1812.10576** (2018), http://arxiv.org/abs/1812.10576
22. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. arXiv preprint arXiv:1905.10958 (2019)
23. Mazumder, S., Liu, B., Wang, S., Zhu, Y., Yin, X., Liu, L., Li, J., Huang, Y.: Guided exploration in deep reinforcement learning (2019), https://openreview.net/forum?id=SJMeTo09YQ
24. McFarlane, R.: A survey of exploration strategies in reinforcement learning. McGill University, http://www. cs. mcgill. ca/cs526/roger. pdf, accessed: April (2018)
25. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
26. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
27. Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Overcoming exploration in reinforcement learning with demonstrations (2017)
28. Nair, S., Zhu, Y., Savarese, S., Fei-Fei, L.: Causal induction from visual observations for goal directed tasks. arXiv preprint arXiv:1910.01751 (2019)
29. OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., Zhang, L.: Solving rubik's cube with a robot hand (2019)
30. Pearl, J., Mackenzie, D.: The Book of Why: The New Science of Cause and Effect. Penguin Books Limited (2018)
31. Pearl, J.: Causality: models, reasoning, and interference. Cambridge University Press (2009)
32. Saunders, W., Sastry, G., Stuhlmueller, A., Evans, O.: Trial without error: Towards safe reinforcement learning via human intervention (2017)
33. Sen, R., Shanmugam, K., Dimakis, A.G., Shakkottai, S.: Identifying best interventions through online importance sampling. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3057–3066. JMLR. org (2017)
34. Spirtes, P., Glymour, C.N., Scheines, R., Heckerman, D.: Causation, prediction, and search. MIT press (2000)
35. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. The MIT Press. (2018)
36. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature pp. 1–5 (2019)
37. Watkins, C.J., Dayan, P.: Q-learning. Machine learning **8**(3-4), 279–292 (1992)
38. Woodward, J.: Making things happen: A theory of causal explanation. Oxford university press (2005)
39. Woodward, J.: Causation and manipulability. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, winter 2016 edn. (2016)
40. Zhang, J., Bareinboim, E.: Transfer learning in multi-armed bandit: a causal approach. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. pp. 1778–1780 (2017)