

RESOLVING SPURIOUS CORRELATIONS IN CAUSAL MODELS OF ENVIRONMENTS VIA INTERVENTIONS

Anonymous authors

Paper under double-blind review

ABSTRACT

Causal models could increase interpretability, robustness to distributional shift and sample efficiency of RL agents. In this vein, we address the question of learning a causal model of an RL environment. This problem is known to be difficult due to spurious correlations. We overcome this difficulty by rewarding an RL agent for designing and executing interventions to discover the true model. We compare rewarding the agent for disproving uncertain edges in the causal graph, rewarding the agent for activating a certain node, or rewarding the agent for increasing the causal graph loss. We show that our methods result in a better causal graph than one generated by following the random policy, or a policy trained on the environment’s reward. We find that rewarding for the causal graph loss works the best.

1 INTRODUCTION

Causality (Halpern & Pearl, 2005) is an important concept (Pearl, 2018) for Machine Learning, since it resolves many issues in performance and Artificial Intelligence (AI) safety (Amodei et al., 2016) such as interpretability (Madumal et al., 2019; Bengio, 2017), robustness to distributional shift (de Haan et al., 2019a) and sample-efficiency (Buesing et al., 2018). It is particularly well suited for Reinforcement Learning (RL), compared to supervised learning, because in RL there is an opportunity to take actions and influence the environment in a directed way. Since causality is a cornerstone in science, such an agent is expected to be superior to non-causal agents (Marino et al., 2019).

Spurious correlations are a major obstacle in learning causal models. If present, they make learning from purely observational data impossible (Pearl & Mackenzie, 2018). We take advantage of the fact that it is possible to uncover the causal graph by executing *interventions* (Halpern & Pearl, 2005) which change the data distribution. We design a method to automatically resolve spurious correlations when learning the causal graph of the environment. Since we are interested in learning the high-level dynamics of the environment, low-level actions do not directly represent interventions in the environment. As a solution, we reward an RL agent for either setting nodes or edges (based on uncertainty) to specific values, or by rewarding the agent for disproving the learned causal model.

Contributions. In this paper, we formulate the problem of learning the causal graph of an RL environment in an active way. We present ways to design interventions and execute them in an end-to-end fashion. We show that interventions outperform a random baseline policy and a policy trained on the environment’s reward. The main contribution is a framework to deal with spurious correlations in RL via interventions, from definitions to methods and experiments.

2 PROBLEM

We have an RL environment μ containing a set of observations \mathcal{O} , set of actions \mathcal{A} , transition probability $\mathbb{P}[o', r|s, o]$ (we use \mathbb{P} for probability) for $r \in \mathcal{R}$, initial observation distribution $\mathbb{P}[o_0]$. We call a tuple of an observation, action and a reward $x = (o, a, r)$ a *step*. We denote \mathcal{X} the space of steps. History $h = (x_0, \dots, x_T)$ contains the steps from one agent-environment interaction. $f: \mathcal{X} \rightarrow \mathbb{R}^d$ is the mapping from raw steps to high-level features.

Causal learning. We use the standard definition of a Structural Causal Model for time series data (Halpern & Pearl, 2005). We create a directed graph of d nodes, one for every feature. Edges represent dependencies between features of time-steps and are labelled with number of time-steps it takes for one node to influence another. The value of a node is determined by values of its parents at past time-steps. $L_{\pi,t}(G) \geq 0$, where π is the policy used to collect the data, denotes the loss of the causal learner (Runge et al., 2019). This means how well the graph fits the features of previous episodes.

Spurious correlations and interventions. We would like to find a graph G which fits the environment μ , without giving the agent the ground truth G^* . However, which policy should we execute in μ to learn G ? Our proposition shows that a random policy is sufficient to learn the graph in a realizable case defined by (Shalev-Shwartz & Ben-David, 2014) where some graph perfectly fits the data¹. In that case, interventions are not required to resolve spurious correlations.

Proposition 1. *For an environment μ and features f with a true causal graph G^* s.t. $L_{\pi}(G^*) = 0$ for all π and $L > 0$ for other G , a random policy π_r gives the true graph: $\arg \min_G \max_t L_{\pi_r,t}(G) = G^*$*

Intuitively, in the realizable case, more data is always better. The proof is based on two ideas: first, for a policy π^* giving the true G^* , a random policy π_r will take same actions as π^* with some probability: $\mathbb{P}[\pi_r = \pi^*] > 0$. Thus, data from π^* will be in the dataset. Next, since G^* fits any policy, the learner will find a graph G s.t. $L_{\pi_r,t}(G) = 0$. By linearity of L , loss on π_r equals a non-negative combination of losses over policies π_r equals to, including one for π^* . Now, since the non-negative combination is 0, one particular term $L_{\pi^*,t}(G) = 0$ as well, and $G = G^*$

In contrast, in cases where we cannot fit the data perfectly ($L_{\pi,t}(G^*) > 0$), it is possible that two policies produce different graphs, no matter the length of the history t or the method to learn the graph. Indeed, given data from one policy only, we can construct two environments with different correct ground truth graphs, which will match the existing data perfectly. Therefore, no learning method can uncover the graph because it is not fully determined by the data obtained. In some cases, this leads to learning spurious correlations: dependencies which work given one policy, but not the other. For example, noise in one feature might force the learner to rely on spurious features, which is irrelevant given a different policy.

We define the problem in a minimax² fashion: the graph should not be disproved even by the worst policy π , at any number of collected episodes t . This is similar to the scientific method in the real world: we want to learn causal relationships that are true no matter which experiments or actions we perform.³

$$G^* = \arg \min_G \max_{\pi} \max_t L_{\pi,t}(G) \tag{1}$$

In practice, in the equation above, we use a finite sequence of policies, instead of all policies in \max_{π} , and we consider a finite $t < \infty$. Executing the next policy π_{i+1} after π_i can be seen as doing an intervention $do(I_{i+1})$ in the causal model, since the policy sets nodes to specific values. In that sense, we sample from the interventional distribution $\mathbb{P}[\cdot | do(I_{i+1})]$.

In the next section we give concrete methods for intervention design.

3 SOLUTION

We compare different methods for intervention design:

Intervention design via edges. We test if selected edges in the graph are real or caused by spurious correlations. Edges where the causal graph learning algorithm is uncertain are selected more often.

¹Note that in practice, a random policy might take too much time to explore the environment

²Note also that the definition is dependent on the particular $L(\cdot)$. It might be meaningless if different policies generate drastically different graphs. In that case, the environment has an ill-defined minimax causal structure. This is similar to the problem of defining the general fitness of an RL agent (Legg, 2008), which does not have a silver-bullet solution (it is prior-dependent). A slightly better way is to create a soft prior over policies, like the complexity prior (Rathmanner & Hutter, 2011): $\arg \min_G \max_t \sum_{\pi} \mathbb{P}[\pi] L_{\pi,t}(G)$. The minimax version, though, is easier to define and evaluate.

³For example, the goal of Physics is to find laws which cannot be disproved by doing experiments.

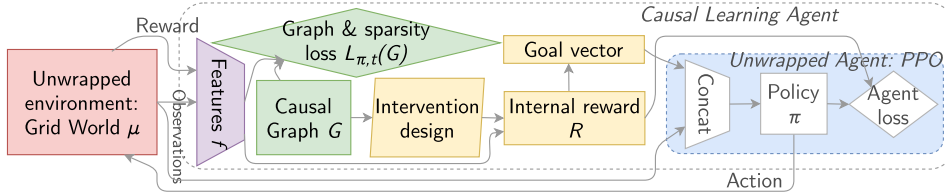


Figure 1: Learning the causal graph by actively interacting with the environment. Given a high-level set of features f_i and an environment μ , we collect data using a policy π to learn an initial causal model G (hypothesis) from features time series. Then, we design an intervention (an experiment), a new policy aimed at disproving the current causal graph to learn the true one. Then, the intervention is executed in the environment as a standard agent-environment interaction with intrinsic reward, and the process is repeated.

We measure uncertainty by how different the results are when trained on different subsets of the data. When selecting an edge $f_i \rightarrow f_j$ with a positive coefficient in a linear causal model⁴, we propose to test it by setting $do(f_i = \max, f_j = \min)$ (minimal and maximal feature values). To do so, we reward the agent for setting $f_i = \max$ and for $f_j = \min$. The total reward is $R = f_i - f_j$ ⁵. Note that in this approach, we do not need to explicitly learn the mapping between low-level actions and high-level features f_j , since we simply use RL for the high-level task.

Intervention design via nodes. We reward the agent for setting a target node f_i to a target value x . We also reward for keeping everything else the same by penalizing the difference d between the averages and variances of feature distributions from previous and current policies⁶: $R = -|f_i - x| - d(f(h)|\pi, f(h)|\pi_{old})$. Another method for the distance part is to reward the agent for the environment reward with some coefficient, because having a common reward will keep the behavior similar: $\gamma: R = -|f_i - x| + \gamma R_{old}$. Node values are selected uniformly at random, while nodes are selected based on the average uncertainty of the edges as explained above.

Intervention design via loss. We reward the agent for finding policies π which give high causal graph loss $R = L_\pi(G)$. This is similar to curiosity approaches (Pathak et al., 2017). The reasoning behind this is that we want to find data disproving our model, like in Eq. 1. Compared to previous methods, we do not select the node or an edge explicitly.

For the rest of the setup, we use the simplest methods (Granger 1-step causality (Granger, 1969) with hardcoded features and a sparsity loss). In this approach, we simply regress the current time-step x_t on the previous one x_{t-1} : $x_t = Wx_{t-1} + b$ using a linear relationship with l_1 regularization. It is trained in a supervised manner by aggregating data from different policies to approximate a solution to Eq. 1. We note that better methods (Runge et al., 2019) of learning causal graphs would still fail without interventions (as explained in the previous section). Other causality learning methods are compatible with our approach. Methods which discover the features end-to-end (Thomas et al., 2018; Kurutach et al., 2018; Ke et al., 2019; François-Lavet et al., 2018; Zhang et al., 2019) can be used to discover the nodes in the causal graph.

Now, all the discussed components are combined together into a causal agent, to discover the true causal graph of the environment, see Figure 1.

4 THE ENVIRONMENT

We use a simple Grid-World environment. The agent needs to eat food in the environment or the episode will end. In addition, it collects keys to open chests, with each chest giving a reward. There is a button which turns a light on and off and does not give reward. Figure 2a represents the causal model we want to discover. Appendix C contains more details about the environment.

⁴In the non-linear case, the coefficient might depend on the current value of features. In that case, this approach will still work, but the step has to be small enough.

⁵In case with a negative coefficient, we need $f_i + f_j$.

⁶We try set d as the difference between random variables' expected values and variances.

We use the following specific environments: (A) a 5x5 grid-world with randomly placed items. (B, Figure 2b) a grid-world with a fixed map, where the agent must collect the key before the food. (C, Figure 2c) 10x10 grid-world with randomly placed items where the food is close to the key, and the chest is far away.

We add noise to each of the environments. With some probability, food is visible at cells not containing food.

The environment we choose is characteristic of the real world, as it contains spurious correlations that we need to uncover by changing behavior.

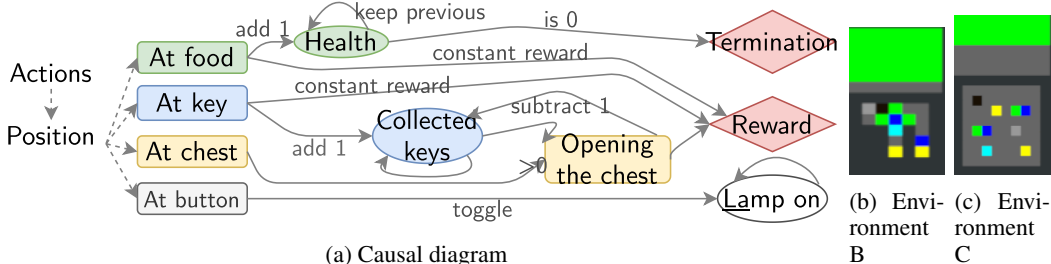


Figure 2: Left 2a: The causal diagram of the environment which the agent should learn. The player needs to collect food and keys. Keys are used to open chests and the number of keys is displayed above the first black line. Top row with health decreases at every time-step, and the episode ends if it is 0. The button toggles the lamp (black/white) which gives no reward. Right two 2b, 2c: layouts of environments B and C.

5 EXPERIMENTS

Hardcoded features. We augment the feature set with conjunctions of relevant features in order to keep the problem in the linear domain. This allowed us to keep the causal learning simple to focus on interventions. There are many techniques for learning nonlinear causal graphs that are compatible with our approach.

Baselines and methods. For all methods we first use a random policy for exploration of the environment. Next, we train a PPO (Schulman et al., 2017) agent to follow the intervention reward. We compare three methods for interventions: rewarding the agent proportional to the loss of the model (Loss), disproving edges (Edge), setting nodes to values (Node). We measure if the correct graph was learned using cosine similarity.

5.1 RESULTS

The random and environment reward policies discover the true causal graph in environment A, because the environment is randomly generated, so there aren’t any spurious correlations. The random and environment reward policies extremely rarely discover the true causal graph in environments B and C. This is because the food presence feature is noisy and a random policy often collects the food before the keys because they are close together. To predict the health increase, it is best to rely on the spurious feature “food and keys > 0”.

The intervention methods discover the true causal graph in environments B and C more often (results for C in the appendix). This is because they also include data from the intervention policy which collects the food when the agent doesn’t have a key. With data from the intervention policy, it is no longer optimal to rely on the spurious feature.

The Loss intervention method outperforms Node and Edge methods on environment (B), Figure 2b. The main problem with the Node and Edge methods is that they have to choose the correct node or edge to intervene on. Once the correct edge or node is chosen, the true graph is learned quickly. In contrast, the Loss method doesn’t have to choose the right thing to intervene on. We expected that for harder environments, we would have to explicitly specify what node or edge to intervene on, as they are less trivial to find by random exploration. This would lead to Node and Edge methods

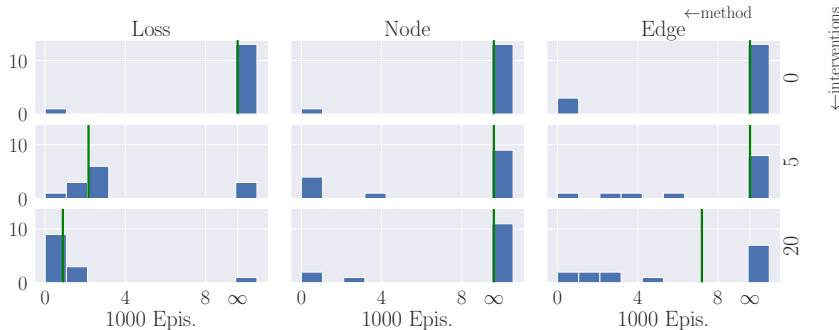


Figure 3: Experimental results on environment B for predicting the true causal graph. Horizontal axis shows the number of episodes (in 1000s), plots are arranged by intervention method (Loss, Node, Edge). Vertical axis represents the number of runs (out of 10) which have converged to the true graph G^* . Plots are arranged by the number of interventions (0, 5, 20). Green line represents the median. 0 interventions corresponds to training with reward. The random policy is evaluated in a separate experiment with spurious correlations as a result. ∞ means that the algorithm didn't find the correct graph during training.

outperforming the Loss method. However, the experimental results show the opposite: in simple environments hand-designed methods (edges and nodes) perform reasonably well, but if we increase the number of features or the complexity of the environment, they stop finding good policies. We didn't test the node and edge interventions in environment C extensively due to this reason. Sampling without replacement when selecting edges gives faster convergence to the true graph versus sampling with replacement. We also found that selecting edges based on uncertainty (by training on different subsets $s = 5$) gives better results than selecting random edges. Results for selecting nodes based on uncertainty are similar to selecting randomly. For the nodes method, we found that keeping the new policy close to the old one by using the reward from the environment works. However keeping the feature statistics the same doesn't work because the agent learns a new way to achieve the same statistics.

6 CONCLUSION

We design a method to learn the causal model of the environment by performing interventions, which helps prevent learning spurious correlations. This shows the potential of RL to improve causal graph learning and compares techniques to accomplish this. We state the problem of learning a causal graph in a RL setting so other work can build off of ours.

7 FUTURE WORK

We plan to combine our graph learning with one of the approaches for learning the features (Kurutach et al., 2018; Ke et al., 2019; François-Lavet et al., 2018; Zhang et al., 2019) and train the entire network end to end. The sparsity loss will help the features be disentangled because it will minimize dependencies between them (Thomas et al., 2018).

To make our method more general, we plan to use one of the advanced non-linear causality learners (Ke et al., 2019). Some of them are differentiable, which would allow to backpropagate from the graph to the features.

Finally, we can utilize the high-level graph as a hierarchical RL controller (Nachum et al., 2018). Specifically, we can run a traversal algorithm on the causal graph to find chains of nodes that lead to high reward. Then, we can reward the agent for activating the nodes in the correct sequence. This might increase the robustness to distributional shift, as we will rely on the correct features for acting.

REFERENCES

- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- Yoshua Bengio. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.
- Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.
- Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal Confusion in Imitation Learning. (NeurIPS):1–17, 2019a. URL <http://arxiv.org/abs/1905.11979>.
- Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pp. 11693–11704, 2019b.
- Tom Everitt, Pedro A Ortega, Elizabeth Barnes, and Shane Legg. Understanding agent incentives using causal influence diagrams, part i: single action settings. *arXiv preprint arXiv:1902.09980*, 2019.
- Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined Reinforcement Learning via Abstract Representations. sep 2018. URL <http://arxiv.org/abs/1809.04506>.
- Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.
- Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. *The British journal for the philosophy of science*, 56(4):843–887, 2005.
- Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Chris Pal, and Yoshua Bengio. Learning Neural Causal Models from Unknown Interventions. oct 2019. URL <http://arxiv.org/abs/1910.01075>.
- Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart Russell, and Pieter Abbeel. Learning Plannable Representations with Causal InfoGAN. jul 2018. URL <http://arxiv.org/abs/1807.09341>.
- Shane Legg. *Machine super intelligence*. PhD thesis, Università della Svizzera italiana, 2008.
- Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable Reinforcement Learning Through a Causal Lens. may 2019. URL <https://arxiv.org/abs/1905.10958>.
- Kenneth Marino, Rob Fergus, and Arthur Szlam. Toward a Scientist Agent : Learning to Verify Hypotheses. 2019.
- Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3303–3313, 2018.
- Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.
- Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.
- Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.
- Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Valentin Thomas, Emmanuel Bengio, William Fedus, Jules PONDARD, Philippe Beaudoin, Hugo Larochelle, Joelle Pineau, Doina Precup, and Yoshua Bengio. Disentangling the independently controllable factors of variation by interacting with the world. *arXiv preprint arXiv:1802.09484*, 2018.

Amy Zhang, Zachary C. Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, and Tommaso Furlanello. Learning Causal State Representations of Partially Observable Environments. (ii):1–16, 2019. URL <http://arxiv.org/abs/1906.10437>.

A RELEVANT WORK

Our method to perform an intervention on an edge is similar to the method used in Marino et al. (2019) to test hypotheses. Compared to that approach, we are interested in the true causal graph of the environment rather than in testing specific hypotheses. Interventions to learn the true graph can be seen in (de Haan et al., 2019b). Our approach is focused on learning the correct graph rather than acting well. We extend the Action-Influence model (Marino et al., 2019; Everitt et al., 2019) to understand the environment. Compared to (Madumal et al., 2019), we learn the graph rather than design it by hand. The idea to reward the agent for the loss of the causal graph is taken from (Pathak et al., 2017). However, here we are interested in a very low-dimensional causal graph rather than in a black-box model of the environment.

B HYPERPARAMETER SELECTION

Resources and parameters. Parameters were chosen with a hyperparameter search on the task of solving the environment using PPO (Schulman et al., 2017). We run the total of 8000 episodes for B and 50000 episodes for C. We vary the number of epochs to train the causal graph in 500-10000, number of interventions 0-50, number of training calls 5-100, intervention method Loss, Edge, maximal number of episodes in the buffer 10-5000, method to select the edge Constant, Weighted and Random. We learn the graph on evaluation data without noise, and update the reward for the trainer.

C THE ENVIRONMENT

We implement the environment using pycolab. All updates are delayed 1 time-step to give causal information.

D EXPERIMENTS

Figure 4 shows the results for environment C. Without interventions, the correct graph is never uncovered. In contrast, with interventions, the correct graph is learner, the more interventions the better.

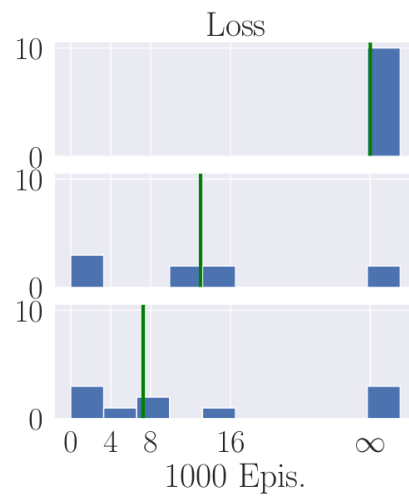


Figure 4: Experimental results on environment C for predicting health, reward, keys and lamp. The description matches that of Figure 3.