A Meta-Reinforcement Learning Algorithm for Causal Discovery

Author names withheld

Editor: Under Review for CLeaR 2023

Abstract

Causal models have great potential when it comes to enabling machine learning models to go beyond pure correlation-based inference. Unfortunately though, estimating the causal structure of an environment poses a significant challenge both in computational effort and in accuracy, let alone its impossibility without interventions in general. In this work, we show that it is possible to meta-learn an active learning algorithm for causal discovery (Meta-Causal Discovery, MCD) in synthetic environments. We learn a policy that learns to perform interventions and update its structure estimate simultaneously. The learned policy can be used to perform causal discovery in a matter of milliseconds. By limiting the episode length, we put an upper bound on the number of interventions that can be performed by MCD, making it more suitable for applications where post-interventional samples are hard to obtain. We show empirically that our algorithm estimates a good graph compared to SOTA approaches and that interventions contribute significantly to MCD's performance.

Keywords: Causal Discovery, Reinforcement Learning, Meta-Learning

1. Context and Contribution

Many scientific questions, from "Why did this apple fall on my head?" to "Does more physical activity reduce the risk of cardiovascular diseases?", aiming at answering questions about causal effects. Within the field of causality, these questions can be mathematically formalized. Although causality has been researched for decades (Glymour et al., 1991; Spirtes et al., 2000; Pearl and Mackenzie, 2018), it has recently gained new momentum in the context of machine learning (ML) (Schölkopf et al., 2021) and, more specifically, reinforcement learning (RL).

Many of the capabilities that causality brings for ML are due to the inference power of causal models, such as reasoning about actions and counterfactuals, which enable ML to go beyond pure correlation-based inference. While the inference power of causal models is impressive, estimating their cause-effect structure from data has been posing several challenges. The task of estimating causal structures from data is referred to as *causal discovery*. The core challenge of causal discovery lies in the fact that some causal structures cannot be distinguished from observational data alone (Hauser and Bühlmann, 2012). This issue can be mitigated by assigning values to variables independently from their causes (Pearl, 1993; Hauser and Bühlmann, 2012; Bareinboim et al., 2020), a process called *intervention*. Unfortunately, when confronted with real-world environments, performing interventions such as randomized controlled trials can be resource intense. Therefore, a large body of research exists on intervention design, or put in different words, on how to minimize the number of interventions needed to estimate the causal model.

With the successful application of RL algorithms to many domains (Moerland et al., 2020; Plaat et al., 2021; Wang et al., 2022), the opportunity to use RL as a tool for causal discovery has opened as well. RL methods allow for actively sampling data as opposed to learning from a static data set of pre-collected observations. This active learning setting for data collection allows for estimating

causal structures edge-by-edge. This can be beneficial for online decisions e.g. on which variable to intervene based on how informative an intervention is for estimating a causal structure. Furthermore, an RL setting allows us to sample data beyond a data set of fixed size and, thus, potentially improve generalization.

In this work, we show that it is possible to meta-learn an algorithm for causal discovery (MCD). We sketch an active meta-reinforcement learning model that estimates the causal structure of an environment with a given set of variables. The model is allowed to perform interventions with a limited budget to aid this process. The model simultaneously learns to perform informative interventions and to infer the updates to the structural model based on the resulting observations. During estimation, the weights of the model are frozen and, therefore, the model learns the causal structure only utilizing the current activations in the model. This work contributes to common challenges in causal discovery through the following capabilities:

- Providing good estimates of the ground-truth causal structure compared to the SOTA.
- Performing causal discovery in a matter of milliseconds.
- Integrating observational and interventional data for causal discovery.
- Limiting the number of interventions through a hyper-parameter.

We will start by providing an overview of the relevant literature on causal discovery leading to a discussion of the common challenges of the task. We will then proceed to introduce necessary notations, followed by an in-depth description of our approach and our model. We will conclude with experiments on the accuracy and runtime of our model w.r.t. SOTA approaches and a rigorous discussion thereof. An overview of our approach is presented in Figure 1. The implementation and data can be found at https://anonymous.4open.science/r/5D56/

2. Preliminaries and Notation

Causal relationships can formally be expressed in terms of a *structural causal model* (SCM). We define an SCM S as a tuple $(\mathcal{X}, \mathcal{U}, \mathcal{F}, \mathcal{P})$ where $\mathcal{X} = \{X_1, \ldots, X_{|\mathcal{X}|}\}$ is the set of *endogenous* variables; $\mathcal{U} = \{U_1, \ldots, U_{|\mathcal{U}|}\}$ is the set of *exogenous* variables; $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{X}|}\}$ is the set of functions whose elements are defined as *structural equations* in the form of $X_i \leftarrow f_i(.)$; $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{U}|}\}$ is a set of pairwise independent distributions where $U_i \sim P_i$. Every SCM induces a *graph structure* G in which each node represents a random variable. $\forall W_i \in \{\mathcal{X} \cup \mathcal{U}\}, X_j \in \mathcal{X} : G$ has a directed edge (W_i, X_j) iff W_i is an input of f_j. This implies that every exogenous variable U_i is a root node in G. In this work, we restrict ourselves to SCMs that induce a *directed acyclic graph* (DAG).

An intervention[1] on a variable $X_i \in \mathcal{X}$ is defined as replacing the corresponding structural equation $X_i \leftarrow f_i(.)$ with $X_i \leftarrow x$ for some value x, which we denote as $do(X_i = x)$. Intervening makes the variable independent of its parents, changing the causal mechanism of the data-generating process. The model is causal in the sense that one can derive the distribution of a subset $\mathcal{X}' \subseteq \mathcal{X}$ of variables following an intervention on a set of variables, called *intervention target*, $\mathcal{I} \subseteq \mathcal{X} \setminus \mathcal{X}'$. We call the resulting distribution over $\mathcal{X}$ *post-interventional*. When no intervention is performed ($\mathcal{I} = \emptyset$) we call the resulting distribution an *observational* distribution.

1. In this work we only consider perfect, hard interventions.

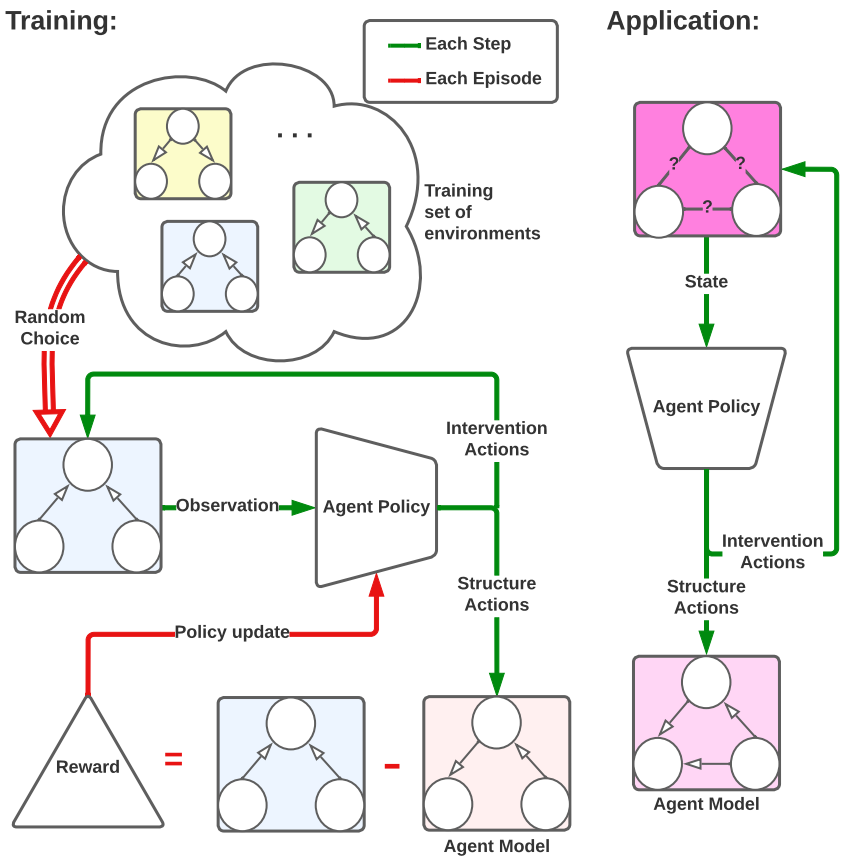

Figure 1: Shows the learning setup of our approach. Given a training set of environments with known causal structures, our model learns to perform interventions and to update its estimation of the structure. The training is guided through a reward that reflects the structural difference between the predicted structure and the ground truth structure. After training the learned causal discovery algorithm can be applied to environments, even if their ground truth structure is unknown.

An RL learning algorithm is characterized by a *state-space* $\mathcal{S}$, an *action-space* $\mathcal{A}$, a *reward* function $r(s) : \mathcal{S} \mapsto \mathbb{R}$, and a *policy* $\pi(s) : \mathcal{S} \mapsto \mathcal{A}$. We define an *episode* e as the state-action sequence from the beginning to the end of the estimation. We will refer to the length of the episode as *horizon* H. The *value function* $V_\pi(s) : \mathcal{S} \mapsto \mathbb{R}$ defines the expected, discounted cumulative reward of a state s, following a policy π, with *discount factor* γ. The objective of the RL agent is to find the optimal policy π^* that maximizes the value function of all states which can be expressed as $\pi^*(s) = \text{argmax}_\pi V_\pi(s), \forall s \in \mathcal{S}$.

3. Related Work

Due to its relevance in many applications, causal discovery research has gained further momentum in the last years leading to an impressive body of work (Vowels et al., 2021). Score-based causal

discovery approaches search the space of (partially) directed acyclic graphs (DAG) via metrics that indicate how well the graph fits the data. This is often done greedily over the space of Markov equivalence classes[2] (Meek, 1997; Chickering, 2002; Hauser and Bühlmann, 2012; Ramsey et al., 2017) or over permutations of node orderings (Solus et al., 2017; Wang et al., 2017a; Yang et al., 2018). Constraint-based approaches leverage the statistical independence patterns in the data to constrain the possible output graphs (Glymour et al., 1991; Spirtes et al., 2000). These constraints can even be expressed as propositional formulas and then solved with answer-set programming (Hyttinen et al., 2014). RL offers an alternative way of searching the space of DAGs by using the reward to navigate toward good graph generators (Zhu et al., 2019). Note that many algorithms rely on strong assumptions on the class of causal relations e.g. linear additive noise models (Bühlmann et al., 2014; Peters et al., 2014; Shimizu et al., 2006). This makes these algorithms interesting for theoretical analysis but it also restricts their application potential in practice.

Since the number of possible DAGs grows super-exponentially in the number of nodes (Robinson, 1977), most score- and constraint-based approaches suffer from long run times. A recent line of research tackles this problem by deploying optimization-based algorithms. These algorithms work e.g. with constraint optimization (Zheng et al., 2018; Brouillard et al., 2020) but also by learning causal graph neural networks (Goudet et al., 2018; Yu et al., 2019; Ton et al., 2021) or variational auto-encoders (Yang et al., 2021). For neuro-causal models, advances are also made in the theoretical analysis of their identifiability (Xia et al., 2021). A similar approach is taken by works that sample both the graph structures and the functional parameters from posterior distributions (Ke et al., 2019; Lippe et al., 2021; Scherrer et al., 2022). This improves learning efficiency, not only of the structures but also of the functional relations of the causal mechanism. While optimization-based approaches can reduce the run-time for structure learning by avoiding a combinatorial explosion, but they can still take a significant amount of time to learn the causal structure.

Another common challenge amongst most causal discovery algorithms is the integration of observational and interventional data. Although integrating frameworks exist (Mooij et al., 2020), only a fraction of causal discovery algorithms successfully jointly consider interventional and observational data (Vowels et al., 2021). A promising direction for the seamless integration of interventional data is by means of RL and active learning. We argue that this is partly due to the implicit connections between interventions and actions in any RL framework, and partly because RL can easily be combined with deep-learning models. Our work distinguishes itself from these closely related works in different ways. While Dasgupta et al. (2019) developed an algorithm that is similar to ours, their primary task was not causal discovery. Nair et al. (2019), Gasse et al. (2021), and Méndez-Molina et al. (2022) put a strong focus on using causal structures to aid RL while learning the structures is done in a supervised manner. Similarly, Scherrer et al. (2022) and Tigas et al. (2022) develop an active learning algorithm that chooses interventions more efficiently to estimate the structure from this data. Amirinezhad et al. (2022) have a similar setup and task but restrict RL to learn a heuristic function for choosing the next intervention target. Furthermore, they do not take into account the values and distributions of the random variables. Their graph-updating procedure is pre-defined, whereas in our approach the update rules can be learned.

2. Roughly speaking, a Markov equivalence class is a set of DAGs which cannot be distinguished by means of their observational distributions alone.

4. Reinforcement Learning Setup

4.1. Actions

We implement two types of discrete actions. The first type performs an intervention on SCM and observes the resulting values of the random variables. This enables the policy to choose a (post-interventional) distribution and to sample from it. We will refer to this kind of action as *listening action*. All, except for one, of the listening actions are *intervention actions* that intervene on exactly one variable (i.e., $|\mathcal{I}| = 1$). For each endogenous variable $X \in \mathcal{X}$, we provide an action $do(X = c)$ for a constant c. We argue that c should be chosen in a way that makes it easy to distinguish the post-interventional distribution from the observational distribution i.e. it should be unlikely that samples from the post-interventional distribution come from the observational distribution. A future expansion of our work could include learning a good c. The intervention actions amount to a total of n actions for n nodes. There is one additional listening action which we call the *non-action*. When the non-action is taken, the agent observes the current values of the observable variables without intervening (i.e., $\mathcal{I} = \emptyset$). This action accounts for the collection of purely observational data.

The second type of action is responsible for constructing the *epistemic model* of the agent. The epistemic model is the current directed graph estimate of the structure of the environment. We will refer to these actions as *structure-actions*. Each structure action can either *add*, *delete* or *reverse* an edge of the epistemic model. Whenever a delete or reverse action is applied to an edge that is not present in the current model, the action is ignored. This is effectively equivalent to performing the non-action. The same holds when the add action is applied to an edge that is already in the epistemic model. We do not make any further restrictions, for instance, w.r.t. acyclicity for the structure actions.

For a graph with n nodes, there are $n(n-1)$ possible edges, and hence there are $3n(n-1)$ structure-actions. Together with the listening-actions we have $n+1+3n(n-1)$ actions. Therefore, the size of the action space is quadratic in the size of nodes.

4.2. State Space

The state s of the environment consists of a concatenation of three vectors and one additional value. The first vector s^V contains the current values of the n endogenous variables where s^V_i is the value of X_i. The second vector s^O is a one-hot vector that indicates which variable is currently being intervened on. So if the i-th element of s^O is 1, then there is an intervention on X_i. The third vector s^G encodes the current epistemic model as a vector. Each value of this vector represents an undirected edge in the graph. The edges in the vector are ordered lexicographically. The value 0 encodes that there is no edge between the two nodes. The value 0.5 encodes that there is an edge going from the lexicographically smaller node to the bigger node of the undirected edge. And the value 1 encodes that there is an edge in the opposite direction. For example, a 3-node graph $X_0 \to X_2 \to X_1$ would be encoded as $s^G = [0, 0.5, 1]$. As the last element in the state vector s^T, we encode the time until the end of an episode normalized to 1 as $s^T = \frac{t}{H}$, where t is the number of steps taken in the current episode and H, is the horizon. Taken together, the size of the state is $2n + n(n-1)/2 + 1$ with n endogenous variable and hence quadratic in the size of the graph. [3]

3. For clarity we obfuscate that the hidden state of the LSTM (see Section 4.4) must also be considered part of the state to fully define the Markov decision process.

4.3. Rewards and Episodes

Our task is to find the causal structure of the environment, i.e., the DAG that corresponds to the graph induced by the SCM. Therefore, we compare the epistemic graph to the true causal structure of the environment. The quantification of this comparison serves as the reward for our algorithm. We count the edge differences between the two graphs. This ensures that generating a model that has more edges in common with the true DAG will be preferred over one which has fewer edges in common. It further gives a strong focus on causal discovery as opposed to scores based on causal inference. Specifically, we use a variant of the *Structural Hamming Distance* (SHD) (Tsamardinos et al., 2006). In this variant, we take two directed graphs and count how many of the edges need to be removed or added to transform the first graph into the second graph. This results in a metric that simply counts the distinguishing edges of two directed graphs. We will refer to this metric as *directed SHD* or *dSHD*. Given a predicted directed graph $G_P = (V, E_P)$ and a target, directed graph $G_T = (V, E_T)$, we define the dSHD as $dSHD(E_P, E_T) =\mid E_P \setminus E_T \mid + \mid E_T \setminus E_P \mid$.

For each episode, we set a finite horizon H. The estimation of the epistemic model is complete when $H-1$ actions were taken. Dynamically determining the end of the estimations is left for future research. Note that when a small episode length is chosen, fewer samples can be collected by the agent. This might impact how well the agent is informed on which updates to make to the epistemic model. At the same time, H should not be set too large since additional learning complexity might be introduced. At the beginning of each episode, an SCM is sampled from the training set and the epistemic model of the agent is reset to a random DAG, to further introduce randomness. The reward is calculated by taking the negative dSHD between the generated DAG and the true causal graph at the end of each episode. Every other step receives a reward of 0.1 if an intervention action is performed, and 0 otherwise. The resulting value function for a state s and a policy π is then defined as

$$V_\pi(s) = \mathbb{E}_{s_t \sim \pi}\left[-\gamma^{H-t}\text{dSHD}(E^H_{Epi}, E_{Env}) \mid s_0 = s\right] + E_{s_t \sim \pi}\left[\gamma^t 0.1 \mathbb{1}_I(s_t) \mid s_0 = s\right] \quad (1)$$

where E^H_{Epi} are the edges of the epistemic model at the end of an episode, E_{Env} are the edges of the current target graph and $\mathbb{1}_I(s_t)$ is the function that indicates whether there is an intervention in s_t.

4.4. Learning Algorithm and Policy Network

We use the Actor-Critic with Experience Replay (ACER) (Wang et al., 2017b) algorithm to solve this RL problem. We choose this algorithm because of its sample-efficient off-policy method and its (potentially) easy extension to continuous action spaces. We use a discount factor $\gamma = 0.99$, a buffer size of 500000, and a constant learning rate. All other parameters are according to the standard values of Stable-Baselines (Hill et al., 2018, version 2.10.1).

The architecture of our policy network is sketched in Figure 2. Both, the actor-network and the critic-network are fully-connected multi-layer perceptrons (MLP). They are preceded by a shared network that has fully-connected feed-forward layers followed by a single LSTM Hochreiter and Schmidhuber (1997) layer. The exact amounts of layers and their sizes are specified for each experiment. We want to emphasize the recurrent LSTM layer. It enables the policy to memorize past observations. More specifically, it enables the policy to remember samples from the (post-interventional) distributions induced by the data-generating SCM earlier in that episode and ideally build a representation of their distribution.

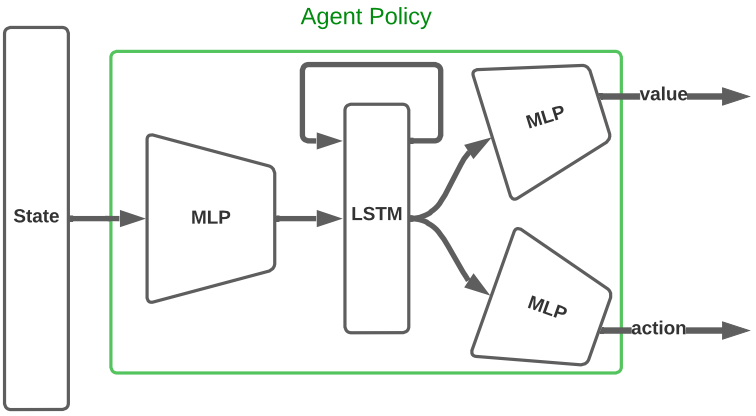

Figure 2: Shows the general architecture of our policy network. The actor and the critic share a fully connected feed-forward network with an additional LSTM layer.

5. Learning to Intervene

First, we develop a toy example to test whether our approach can learn to perform the right interventions to identify causal models under optimal conditions. To this end, we construct a simple experiment in which two observationally equivalent, yet interventionally different environments have to be distinguished. This can only be achieved with the help of interventions (Bareinboim et al., 2020). For this experiment, we disable the additional reward for performing interventions. Thus, if our policy learns to distinguish the two environments, it has to learn that interventions are needed and to infer the right graph from these interventions.

The two environments are governed by the fully observable, 3-variable SCMs with structures $G_1 : X_1 \leftarrow X_0 \rightarrow X_2$ and $G_2 : X_0 \rightarrow X_1 \rightarrow X_2$. In both environments, the root node X_0 follows a normal distribution with $X_0 \sim N(\mu = 0, \sigma = 0.1)$. The nodes X_1 and X_2 take the values of their parents in the corresponding graph. The resulting observational distributions $P_{G_1}(X_0, X_1, X_2)$ and $P_{G_2}(X_0, X_1, X_2)$ are equivalent and so are the post-interventional distributions after interventions on X_0 or X_2. For an intervention on X_1, $P_{G_1}(X_0, X_2 \mid do(X_1 = x)) \neq P_{G_2}(X_0, X_2 \mid do(X_1 = x))$. Hence the two SCMs can only be distinguished by intervening on X_1. The details for the training setup can be found in Appendix A.1. The algorithm is trained in both environments. This allows us to investigate whether, given enough training time and data, our approach *can* learn to distinguish the environments.

After training, we observe that the mean dSHD of the produced graphs is 0.0 with a standard derivation of 0.0. This is a perfect reproduction of the two environments in all cases. This indicates that our policy has indeed learned to use the right intervention to find the true causal structure. For further testing, we apply the converged policy 10 times to each of the environments and qualitatively analyze the behavior. What the resulting 20 episodes have in common is that, towards the beginning of each episode, they tend to delete edges that do not overlap in the two environments. Then an intervention on X_1 is performed. Depending on the outcome of the intervention, either G_1 or G_2 is ultimately generated. This can also be seen in the two example episodes in Figure 3.

This shows that our learned policy learns to use the intervention on X_1 to distinguish between the two environments. Thus, our approach is capable of learning to use interventions in an active

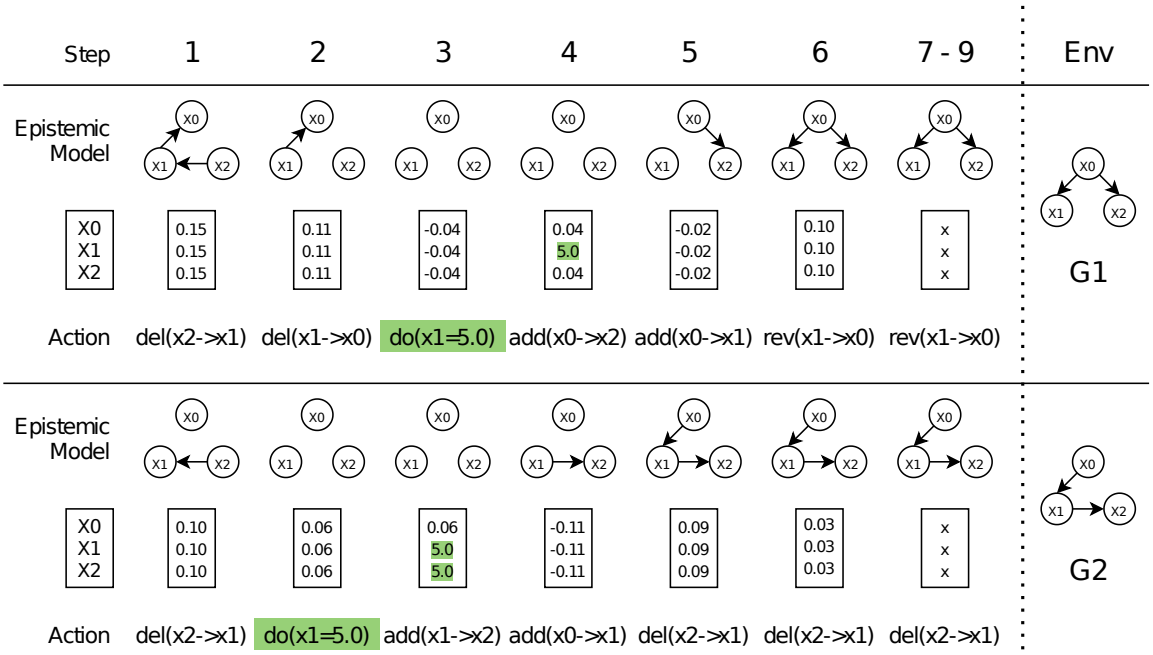

Figure 3: Illustration of two sample episodes after training with the respective causal environments G_1 and G_2. Each step shows the current estimate of the causal structure, the current values of the three random variables, and the action which is chosen by the policy based on those observations. Interventions and their effects are highlighted in green. In steps 7-9 neither the epistemic model nor the resulting action changes.

manner and to generate the appropriate graph from the resulting observations. If this can be successfully learned in more complex environments, our learned policy could potentially be used to discover new rules of causal structure estimation. Furthermore, these results suggest that the model has learned to only perform interventions that are relevant as opposed to random interventions.

6. Learning a Causal Discovery Algorithm

In this section, we investigate whether we can learn a causal discovery algorithm with the setup described in Section 4. If learning such an algorithm is successful, the algorithm should be able to identify structures that it has not encountered during training. To test this, we compare our learned algorithm to the SOTA approaches ENCO (Lippe et al., 2021) DCDI (Brouillard et al., 2020), NOTEARS (Zheng et al., 2018), and a baseline that generates a random DAG. ENCO and DCDI can both integrate observational and interventional data while NOTEARS only uses observational data.

Following the widely adopted practice, we test our approach on SCMs that have an additive linear causal model with independent Gaussian noise. Although this choice limits the applicability to real-world environments, it provides a good means for comparison to other approaches. It is known that these kinds of environments can suffer from varsortability where good results can be achieved by ordering the variables by the variance of their observational distribution (Reisach et al.,

2021; Kaiser and Sipos, 2021). To make our approach less prone to this error, we randomly sample the variance of each Gaussian noise we use.

Given a structure $G = (V, E)$ and $\mathcal{X} = V$, we model our SCM environments as follows. For every endogenous variable X_i we add a parent exogenous variable U_i with distribution $P_i = N(\mu = 0, \sigma = \Sigma)$ where Σ is sampled from $Uniform([0; 0.5])$. For each endogenous variable X_i, we model f_i as

$$f_i(Pa_X^G, U_i) = \left(\sum_{Y \in Pa_X^G} WY \right) + U_i \tag{2}$$

where Pa_X^G are the parents of X in G, and $W \sim Uniform([-1; 1])$ represents a random weight for each causal effect of a parent to a child. We randomly generate a test set of 7 DAGs with 3 variables and 200 DAGs with 4 variables. For each of these graphs, we generate 10 SCMs as described above for evaluation. During training, we sample a random ground truth DAG at the beginning of each episode. If this random DAG is in the test set, we discard it and sample a new random DAG. This process is repeated until the sampled DAG is not in the test set to ensure that our model has never seen the test set. When a DAG is found, we generate an SCM as described above as our current environment. The training details for this experiment can be found in Appendix A.2. We will refer to the best model that is found during training as *best model*. The setup for the benchmarks can be found in Appendix B.1. For each of the algorithms we computed the dSHD between the predicted DAG and the ground truth DAG. Table 1 shows the results of running the algorithms on the first 50SCMs in the test set.

	3 Variables			4 Variables		
	mean	median	std	mean	median	std
Random	4.43	4.0	0.90	4.80	5.0	1.72
DCDI	2.94	3.0	0.70	4.44	**4.0**	1.77
ENCO	3.18	3.0	1.09	3.74	**4.0**	1.73
NOTEARS	2.50	3.0	0.92	3.72	**4.0**	1.77
MCD (ours)	**1.28**	**1.0**	**0.66**	**3.60**	**4.0**	**1.62**

Table 1: Statistics over the dSHDs resulting from running the algorithms on the first 50 SCMs in the test set.

Firstly, Table 1 shows that our approach outperforms the random baseline, suggesting that MCD learns to estimate the environment's causal structure beyond randomly orienting edges. The means over the resulting dSHDs suggest that our approach compares favorably to the benchmarks. To investigate this difference in more detail, we performed a one-sided Wilcoxon signed-rank test on each estimate from our policy and from DCDI, ENCO, and NOTEARS. To correct for performing 3 comparisons, we consider a significance level of 1.7%. In the 3 variable case as well as the 4 variable case we can conclude that the dSHDs from our method are significantly lower than the ones from any of the other algorithms[4]. We also note that each run of MCD takes an average of 23ms in the

4. We note that the results for ENCO seem significantly worse than those proposed in the original paper Li et al. (2020). We could not find a definitive answer to why this issue occurs but we suspect either the specific setup of our environments or an undiscovered implementation issue.

3 variable case and 30ms in the 4 variable case on a consumer-grade notebook as opposed to the SOTA, which can take minutes for one estimation. We attribute this performance to the fact that one estimation of MCD only takes H forward passes through the policy network.

We conclude that with our approach a causal discovery algorithm can be learned that, in an active setting, performs interventions and updates its structure estimate. Our algorithm not only compares favorably to the SOTA w.r.t. to the dSHD to the ground truth graph but is also computationally quick in deriving the estimate making it interesting for a variety of applications.

7. Contribution of Interventions

To empirically investigate the effect of interventions on the performance of our algorithm, we perform an ablation study. To this end, we train a variant of our policy (MCD-O) which is based on purely observational data , i.e. we disallow the use of interventions, and compare it to the model which uses interventions. We then compare our results to the results of MCD and NOTEARS, which also works on purely observational data and the random baseline of the previous section.

We train our model with the same parameters as in Section 6 and measure the dSHD on the first 50 3-variable SCMs in the test set with the best model of the training run. We perform a Wilcoxon signed-rank test to evaluate whether there is a significant difference between the model that uses interventions and the one that does not. We also test whether there is a difference between NOTEARS and our approach when no interventions are allowed.

The statistics of MCD-O applied once on the first 50 test SCMs are as follows: $mean = 2.6, median = 3.0, std = 1.44$. When comparing this to the version which uses interventions ($mean = 1.28, median = 1.0, std = 0.66$, see Table 1), we can see the importance that interventions have on the overall performance of MCD. This is confirmed by performing a Wilcoxon signed-rank test between the results of MCD and MCD-O indicating that MCD is significantly better (with $p << 0.025$). When comparing MCD-O with NOTEARS, we do not observe any significant difference in a two-sided Wilcoxon signed-rank test ($p \sim 0.4$). In other words, while MCD-O does not provide an improvement over NOTEARS, it still constitutes a valid alternative approach. These results lead to the conclusion that introducing interventions results in the hypothesized edge over the purely observational version of our model.

8. Aspects of Intervention Design

As argued in Section 1, MCD provides an approach to restrict the number of interventions needed for causal discovery. The upper bound of interventions that the learned policy will perform is the horizon of an episode (20 in the current setup). Compared to the interventions used in the benchmarks (up to 10000 samples from the observational distribution and up to 3333 samples from the interventional distributions), this is a significant improvement considering the comparably good performance of MCD. More specifically, in the application of our best models to the 3-variable test SCMs we found an average of 17 samples from interventional distributions. The low number of interventions needed for MCD promises to make it more applicable than SOTA, especially in scenarios where samples from post-interventional distributions are expensive to obtain.

On average, 64% of the interventions were on the first variable and 36% on the third variable. No interventions were performed on the second in any of the runs. To investigate this behavior, we ran checkpoints of the model of earlier steps of the training and found that the model is performing

interventions on the second variable in those checkpoints. We hypothesize that during learning the model learns to not do this intervention because of the structures in the test SCMs.

To have a better comparison of the performance of MCD in a context where interventional samples are hard to obtain, we re-evaluate our approach w.r.t. to SOTA approaches using a similar amount of samples. For DCDI, we take 17 samples from each post-interventional distribution. For ENCO we take 17 samples from each post-interventional distribution and 4 samples from the observational distribution. For NOTEARS we take 20 observational samples.

	mean	median	std
DCDI	3.46	3.0	1.00
ENCO	2.40	3.0	0.80
NOTEARS	2.84	3.0	1.02
MCD (ours)	**1.28**	**1.0**	**0.66**

Table 2: Statistics over the dSHD obtained from predicting the causal structure of the 3 variable test SCMs by algorithm.

Table 2 shows the statistics over the dSHD obtained from running the corresponding algorithms on the 3 variable test SCMs. As expected, we see an increase in the performance gap between MCD and DCDI and MCD and NOTEARS indicating that their ability to perform well when few interventions are provided is limited. Interestingly, ENCO seems to estimate better structures when less data is provided. This suggests that ENCO has fewer issues with lower sample sizes. Overall, we conclude that MCD uses interventions in an efficient way which makes it perform well even when the budget for interventions is low.

9. Conclusion

This paper presents an approach to learning a causal discovery algorithm. In our RL setting, we learn a policy that simultaneously learns to actively perform informative interventions and update its structure estimate. Once the policy is learned, it can be used to perform causal discovery in a matter of milliseconds, even on SCMs whose structure it has not encountered during training. In doing so, it manages to integrate interventional and observational data. Lastly, by limiting the episode length, we put an upper bound on the number of interventions that can be performed by MCD, making it more suitable for applications where post-interventional samples are hard to obtain.

We acknowledge that our approach needs modifications to scale to realistic environments with more variables. The explosion of the action- and state-space that this would imply prompts considerations about better encodings. A further problem in a potential real-world setting is the availability of a large amount of data-generating models for training. To perform well on all the possible causal relations in the real world, the class of training SCMs would need to be significantly expanded. An alternative approach would be to make MCD transferable to SCM classes other than linear-additive SCMs. We argue that also an extension to a scenario in which the variables are learned from raw input would lead to even better applicability since hand-crafted variables often introduce sub-optimalities w.r.t. task performance.

References

Amir Amirinezhad, Saber Salehkaleybar, and Matin Hashemi. Active learning of causal structures with deep reinforcement learning. *Neural Networks*, 154:22–30, 2022.

Elias Bareinboim, JD Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. *ACM Special Volume in Honor of Judea Pearl (provisional title)*, 2020.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems*, 33:21865–21877, 2020.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019.

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

Clark Glymour, Peter Spirtes, and Richard Scheines. Causal inference. *Erkenntnis*, 35(1-3):151–189, 1991.

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. In *Explainable and interpretable models in computer vision and machine learning*, pages 39–80. Springer, 2018.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13 (1):2409–2464, 2012.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. https://github.com/hill-a/stable-baselines, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In *UAI*, pages 340–349, 2014.

Marcus Kaiser and Maksim Sipos. Unsuitability of notears for causal graph discovery, 2021.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Minne Li, Mengyue Yang, Furui Liu, Xu Chen, Zhitang Chen, and Jun Wang. Causal world models by unsupervised deconfounding of physical dynamics. *arXiv preprint arXiv:2012.14228*, 2020.

Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclicity constraints. In *International Conference on Learning Representations*, 2021.

Christopher Meek. *Graphical Models: Selecting causal and statistical models*. PhD thesis, PhD thesis, Carnegie Mellon University, 1997.

Arquımides Méndez-Molina, Eduardo F Morales, and L Enrique Sucar. Causal discovery and reinforcement learning: A synergistic integration. In *International Conference on Probabilistic Graphical Models*, pages 421–432. PMLR, 2022.

Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*, 2020.

Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21:1–108, 2020.

Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

Judea Pearl. Bayesian analysis in expert systems: Comment: graphical models, causality and intervention. *Statistical Science*, 8(3):266–269, 1993.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition, 2018. ISBN 046509760X.

J Peters, JM Mooij, D Janzing, and B Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

Aske Plaat, Walter Kosters, and Mike Preuss. High-accuracy model-based reinforcement learning, a survey, 2021. URL https://arxiv.org/abs/2107.08241.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2):121–129, 2017.

Alexander G. Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game, 2021.

Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*, pages 28–43. Springer, 1977.

Nino Scherrer, Olexa Bilaniuk, Yashas Annadani, Anirudh Goyal, Patrick Schwab, Bernhard Schölkopf, Michael C Mozer, Yoshua Bengio, Stefan Bauer, and Nan Rosemary Ke. Learning neural causal models with active interventions. *stat*, 1050:5, 2022.

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

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Liam Solus, Yuhao Wang, Lenka Matejovicova, and Caroline Uhler. Consistency guarantees for permutation-based causal inference algorithms. *arXiv preprint arXiv:1702.03530*, 2017.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale. *arXiv preprint arXiv:2203.02016*, 2022.

Jean-François Ton, Dino Sejdinovic, and Kenji Fukumizu. Meta learning for causal direction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9897–9905, 2021.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys (CSUR)*, 2021.

Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2022. doi: 10.1109/TNNLS.2022.3207346.

Yuhao Wang, Liam Solus, Karren Dai Yang, and Caroline Uhler. Permutation-based causal inference algorithms with interventions. *arXiv preprint arXiv:1705.10220*, 2017a.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay, 2017b.

Kevin Muyuan Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=hGmrNwR8qQP.

Karren Yang, Abigail Katcoff, and Caroline Uhler. Characterizing and learning equivalence classes of causal DAGs under interventions. In Jennifer Dy and Andreas Krause, editors,

Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 5541–5550. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/yang18a.html.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9593–9602, June 2021.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2019.

Appendix A. Training Details

A.1. Learning to Intervene

For the experiment in section 5 the policy network has a fully connected layer of size 30, followed by an LSTM layer of size 30. The actor-network has one fully connected layer of size 30, the critic-network one fully connected layer of size 10. The length of each episode was set to 10 and the model trained for 5 million training steps. As intervention actions we provide $do(X_i = 0)$ and $do(X_i = 5)$ for each $X_i \in \mathcal{X}$. For all other parameters, the default values were used.

A.2. Learning a Causal Discovery Algorithm

The following configuration for the policy network of the experiment in Section 6 worked best after preliminary experiments for the 3-variable (4-variable) environments: One (two) fully connected layer(s) of size 30 (64) followed by an LSTM layer of size 30 (128). Its outputs are fed into a fully connected layer of size 30 (32) for the actor-network and one of size 10 (32) for the critic-network. As intervention actions we provide $do(X_i = 5)$ for each $X_i \in \mathcal{X}$. We chose this value since it is unlikely to come from any of the noise distributions. For this experiment, we set the horizon to 20.

Appendix B. Setup of Benchmarks

B.1. Samples for Optimistic Experiment

For the evaluation in Section 6 we apply our best models and the benchmarks described above on the first 50 SCMs from the test set. For NOTEARS we sampled 10000 samples from the observational distribution of each SCM. For ENCO we sampled 10000 samples from the observational distribution and 3333 samples from each post-interventional distribution (one per variable) and trained for 50 epochs. For DCDI we took 3333 samples from each post-interventional distribution as well and trained the deep sigmoidal flow model version of the algorithm for 50000 iterations.