OFFLINE MODEL-BASED REINFORCEMENT LEARNING WITH CAUSAL STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-based methods have recently been shown promising for offline reinforcement learning (RL), which aims at learning good policies from historical data without interacting with the environment. Previous model-based offline RL methods employ a straightforward prediction method that maps the states and actions directly to the next-step states. However, such a prediction method tends to capture spurious relations caused by the sampling policy preference behind the offline data. It is sensible that the environment model should focus on causal influences, which can facilitate learning an effective policy that can generalize well to unseen states. In this paper, we first provide theoretical results that causal environment models can outperform plain environment models in offline RL by incorporating the causal structure into the generalization error bound. We also propose a practical algorithm, o**F**fline m**O**del-based reinforcement learning with **C**a**U**sal **S**tructure (FOCUS), to illustrate the feasibility of learning and leveraging causal structure in offline RL. Experimental results on two benchmarks show that FOCUS reconstructs the underlying causal structure accurately and robustly, and, as a result, outperforms both model-based offline RL algorithms and causal model-based offline RL algorithms.

1 INTRODUCTION

Offline Reinforcement Learning (RL) refers to the problem of learning policies entirely from previously collected data. Offline RL is gaining popularity since it enables RL algorithms to scale to several real-world applications, e.g., autonomous driving (Yu et al., 2018) and healthcare (Gottesman et al., 2019), where trial-and-error is too expensive. In the offline setting, Model-Based Reinforcement Learning (MBRL) is a popular framework that learns a predictive environment model for policy optimization (Yu et al., 2020), which relies on the environment model being learned accurately.

However, current offline MBRL approaches usually have poor generalization because the environment models tend to capture spurious correlations that only exist in collected data, resulting in erroneous predictions. For instance, in autonomous driving, if offline data is acquired from a driver who always turns on the wiper and brake pedals on rainy days, such a preference will result in a spurious correlation between "the wiper is turned on" and "the speed is dropped" in the data, which will also be captured by the environment model. Once we employ this environment model for policy learning, the agent will likely urge the driver to switch on the windshield wiper when the vehicle's speed is too high because it believes that "the wiper is turned on" has an effect on "the speed is dropped", which is not sensible and potentially hazardous. Similarly, the distinction between offline data and testing data is influenced by sampling policies with varying preferences. Intuitively, leveraging the causal structure of observed variables can avoid considering spurious correlations as causal influences and thus facilitate the learning of an environment model with enhanced generalizability. Recent empirical evidence also indicates that inducing the causal structure is important to improve the generalization (Edmonds et al., 2018; Tenenbaum, 2018; Bengio et al., 2020; de Haan et al., 2019). Despite such evidence, it is still unknown whether and how the causal structure improves model generalization in offline RL.

For this purpose, we first provide theoretical support for the aforementioned intuition: we show that a causal environment model can outperform a plain environment model on generalization for offline RL. From the causal perspective, we divide the variables in states and actions into two categories, namely, causal variables and spurious variables, and then formalize the process that learns an environment model with both categories of variables. On the basis of the formalization, we quantify the effect of spurious dependencies on the generalization error bound and thereby demonstrate that integrating causal structures can assist in minimizing this bound.

We also propose a practical offline causal MBRL algorithm, FOCUS, to illustrate the feasibility of learning causal structure in offline RL. Learning the causal structure from data, also known as causal discovery (Spirtes et al., 2000b), is a crucial phase of FOCUS. The offline RL properties of sequential information and latent policy preference create certain difficulties but also provide some advantages for implementing causal discovery methods. Specifically, we modified the PC algorithm (Spirtes et al., 2000b), which seeks to uncover causal relationships based on inferred conditional independence relations, to incorporate the constraint that the future cannot cause the past. Consequently, we can reduce the number of conditional independence tests and determine the causal direction. In addition, we employ kernel-based conditional independence tests (Zhang et al., 2011), which can be applied to continuous variables without assuming a specific functional form between the variables or particular data distribution.

In conclusion, this paper makes the following key contributions:

- It theoretically demonstrates that a causal environment model outperforms a plain environment model in offline RL with respect to the generalization error bound.
- It proposes a practical algorithm, FOCUS, and illustrates the feasibility of learning and employing a causal environment model for offline MBRL.
- Our experimental results validate the theoretical claims, showing that FOCUS outperforms baseline models and other online causal MBRL algorithms in the offline setting.

2 Related Work

The RL algorithms with causal structure learning can be roughly divided by the type of their causal discovery methods. First, we will discuss relevant causal discovery methods, followed by related RL algorithms.

Causal Discovery Methods. On the basis of whether we can do interventions or randomized experiments, causal discovery methods can be divided into two groups. In cases where interventions are not possible and only observational data is available, the methods broadly fall into two categories: constraint-based methods and score-based methods. Constraint-based methods use statistic tests (conditional independent tests) to find the causal skeleton and determine the causal directions up to the Markov equivalence class. Score-based methods evaluate the quality of candidate causal models with some score functions and output one or multiple graphs having the optimal score (Heckerman et al., 2006).

RL Algorithms With Causal Structure Learning. de Haan et al. (2019) proposes an imitation learning algorithm in RL, which learns the causal structure between states and actions. It assumes that we can query experts for actions and uses interventioned data to do causal discovery. In model learning for RL, such causal discovery methods with interventioned data are not available because querying experts for the next states is not practical. For MBRL, Ke et al. (2019) (LNCM) views data sampled from different policies as data with soft interventions and use score-based methods with the log-likelihood on "interventional" data as the score function. Its implicit assumption that data is sampled from multiple policies and data has been labeled by its sampling policy is not a general assumption in offline RL, which only holds true in online RL. Given the properties of offline RL that the sampling policy has unknown preferences and interactions with the environment are not available, the above methods are not practical to learn the causal structure in offline RL. By contrast, FOCUS proposes a practical algorithm that learns the causal structure with offline data, which utilizes constraint-based methods and further reduces the testing number with the properties of RL environments.

3 Preliminaries

Markov Decision Process (MDP). We describe the RL environment as an MDP with five-tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$ (Bellman, 1957), where $\mathcal{S}$ is a finite set of states; $\mathcal{A}$ is a finite set of actions; P is the transition function with $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ denoting the next-state distribution after taking action $\mathbf{a}$ in state $\mathbf{s}$; R is a reward function with $R(\mathbf{s}, \mathbf{a})$ denoting the expected immediate reward gained by taking action a in state s; and $\gamma \in [0, 1]$ is a discount factor. An agent chooses actions $\mathbf{a}$ according to a policy $\mathbf{a} \sim \pi(\mathbf{s})$, which updates the system state $\mathbf{s}' \sim P(\mathbf{s}, \mathbf{a})$, yielding a reward $r \sim R(\mathbf{s}, \mathbf{a})$. The agent's goal is to maximize the the expected cumulative return by learning a good policy $\max_{\pi} \mathbb{E}_{\pi,P} \sum \gamma^t R(s_t, a_t)$. The state-action value Q^{π} of a policy π is the expected discounted reward of executing action a from state s and subsequently following policy π: $Q^{\pi}(s,a) := R(s,a) + \gamma \mathbb{E}_{s' \sim P, a' \sim \pi} Q^{\pi}(s', a')$.

Offline Model-based Reinforcement Learning. In the offline RL setting, the algorithm only has access to a static dataset $D = \{(s, a, r, s')\}$ collected by one or a mixture of behavior policies π_β, and further interactions with the environment is not available. When we use model-based approaches to solve offline RL problems, we will learn a virtual environment model $\hat{P}$ for transition prediction from offline data. With the learned environment model $\hat{P}$, we can define a new MDP $(S, A, \hat{P}, R, \gamma, \rho)$. Similarly, we can also define the value function $\hat{Q}^{\pi}$ with $\hat{P}$. A standard model-based RL algorithm (in an online setting) learns a virtual model by fitting it using a maximum-likelihood estimate of the trajectory-based data collected by running the latest policy, which guarantees that the virtual model can always be accurate when the policy keeps exploring (Williams et al., 2017; Kurutach et al., 2018). In the offline RL setting, where we only have access to the data collected by previous policies, the accuracy of the virtual model in exploring policy cannot be guaranteed. Therefore recent techniques all build on the idea of pessimism that regularizes the original problem based on how confident the agent is about the learned model (Kidambi et al., 2020; Yu et al., 2020). Specifically, the policy only visits the states where the learned model is confident in predictions.

4 THEORY

In this section, we provide theoretical evidence that a causal environment model outperforms a plain environment model in offline RL, which shows that utilizing a good causal structure can reduce the generalization error bounds for offline MBRL algorithms. Specifically, we formalize the process that learns the environment model with spurious relations induced by the offline setting and quantify the influence of the spurious relations. We quantify the impact by assuming the causal structure is known and factoring it into the generalization error bounds, which include the model prediction error bound and policy evaluation error bound in RL. In this section, it is assumed that the causal relations are linear and all causal variables are observed. The complete proof can be found in Appendix A.

4.1 MODEL PREDICTION ERROR BOUND

In this subsection, we assume a causal structure of the RL environment and spurious relations in offline data. We point out that the spurious relations lead the model learning problem to an ill-posed problem with multiple optimal solutions in offline data, hence increasing the model prediction error bound. With the aforementioned statement, we present a model prediction error bound that combines key properties of spurious relations, which is a quantitative assessment of the impact on model learning. We provide the model prediction error bound in a supervised learning framework, as model learning can be considered as a supervised learning problem.

Preliminary. Let D denote the data distribution where we have samples $(X, Y) \sim D$, $X \in \mathbb{R}^n$. The goal is to learn a linear function f to predict Y given X. From the causal perspective, Y is generated from only its causal parent variables rather than all the variables in X. Therefore we can split the variables in X into two categories $X = (X_{causal}, X_{spurious})$:

- X_{causal} represents the causal parent variables of Y, that is, $Y = f^*(X_{causal}) + \epsilon_{causal}$, where f^* is the ground truth and ϵ_{causal} is a zero mean noise variable that $X_{causal} \perp\!\!\!\perp \epsilon_{causal}$.
- $X_{spurious}$ represents the spurious variables that $X_{spurious} \perp\!\!\!\perp Y \mid X_{causal}$, but in some biased data sets $X_{spurious}$ and X_{causal} have strong relatedness. In other words, $X_{spurious}$ can be predicted by X_{causal} with small error, i.e. $X_{spurious} = X_{causal}\theta_{spurious} + \epsilon_{spurious}$, where $\epsilon_{spurious}$ is the regression error with zero mean and small variance.

For clearly representation, we use $X_{cau} = X \odot \omega_{cau}$ ($\odot$ represents element-wise multiplication) to replace X_{causal}, where cau records the indices of X_{causal} in X and $(\omega_{cau})_i = \mathbb{I}(i \in cau)$. Correspondingly, we also use $X_{spu} = X \odot \omega_{spu}$ to replace $X_{spurious}$. According to the definition of X_{cau}, we have $Y = (X \odot \omega_{cau})\theta^* + \epsilon_{cau}$, where $\omega_{cau} \odot \theta^*$ is the global optimal solution of the optimization problem

$$\min_{\theta} \mathbb{E}_{(X,Y) \sim D} \|X\theta - Y\|^2. \quad (1)$$

The above problem is easy if the data is uniformly sampled from D. However, in the offline setting, we only have biased data D_{train} sampled by given policy $_{train}$, where the optimization objective is

$$\min E_{\hat{}X;Y \sim D_{train}} \quad X \quad Y^{2}: \tag{2}$$

The Problem 2 has multiple optimal solutions due to the strong linear relatedness of X_{spu} and X_{cau} in D_{train}, which is proved in Lemma 4.1.

Lemma 4.1. Given that $!_{cau} \; X^{\ddagger}$ is the optimal solution of Problem 1, suppose that in D_{train}, $X_{spu} \; \hat{} X \; X! \;_{cau} \bullet \;_{spu} \;_{spu}$ where $E_{D_{train}} \;_{spu} \; 0$ and $_{spu} \; x \; 0$, we have that $\hat{}_{spu} < !_{cau} \; X \; \hat{}^{\ddagger}_{spu} \bullet \; !_{spu}$ is also an optimal solution of Problem 2 for any :

$$E_{\hat{}X;Y \sim D_{train}} \check{S}\check{S}X \hat{}!_{cau} X^{\ddagger} \bullet \; Y \check{S}_{2}^{2} \bullet SX \quad E_{\hat{}X;Y \sim D_{train}} \check{S}\check{S}X \hat{}_{spu} \; Y \check{S}_{2}^{2} \bullet SX$$

The most popular method for solving such ill-posed problem is to add a regularization term for parameters (OpenAI et al., 2019):

$$\min E_{\hat{}X;Y \sim D_{train}} \quad X \quad Y^{2} \quad k Y Y^{2}; \tag{3}$$

where k is a coefficient. The form of Problem 3 corresponds to the form of the ridge regression, which provides a closed-form solution of k by Hoerl-Kennard formula (Hoerl & Kennard, 2000).

In the following, we will first introduce the solution of under Problem 3 in Lemma 4.2, and then introduce the model prediction error bound with in Theorem 4.4. For ease of understanding, we provide a simple version where the dimensions of X_{cau} and X_{spu} are both one ($SX_{cau}S \quad SX_{spu}S \quad 1$).

Lemma 4.2 (Lemma). Given as the coefficient in Lemma 4.1, and k in Problem 3 chosen by Hoerl-Kennard formula, we have the solution of in Problem 3 that:

$$\frac{\ddagger \quad _{spu}}{\ddagger^{2} \quad ^{2}_{spu} \quad 1 \quad \frac{^{2}_{spu}}{^{2}_{cau}} \hat{}1 \quad \frac{1}{\hat{} \; \ddagger \bullet 2} \bullet} \tag{4}$$

Based on Lemma 4.2, we can find that the smaller $^{2}_{spu}$ (it means that X_{spu} and X_{cau} have stronger relatedness in the training dataset D_{train}), the larger . And we also have its bound:

Proposition 4.3. Given as Formula 4, the bound of is that $\frac{1}{2}$ B B $\frac{1}{2}$:

Theorem 4.4 (Spurious Theorem). Let D $\tilde{}\hat{} X; Y \bullet\bullet$ denote the data distribution, $\hat{}_{spu}$ denote the solution in Lemma 4.1 with in Lemma 4.2, and $\hat{Y}_{spu} \quad X \hat{}_{spu}$ denote the prediction. Suppose that the data value is bounded: $SX_{i}S_{1}$ B X_{max} ; i 1; ; n and the error of optimal solution $_{cau}$ is also bounded: $S_{cau} S_{1}$ B $_{c}$, we have the model prediction error bound:

$$E_{\hat{}X;Y \sim D} \hat{}S\hat{Y}_{spu} \quad Y S_{1} \bullet SX \quad B \; X_{max} \; S S_{1} \hat{}S_{spu} S_{1} \quad 1 \bullet \quad _{c}: \tag{5}$$

Theorem 4.4 shows that

- The upper bound of the model prediction error $S\hat{Y}_{spu} \quad Y S_{1}$ increases by $X_{max} \; S S_{1} \hat{}S_{spu} S_{1} \quad 1 \bullet$ for each induced spurious variable X_{spu} in the model.
- When X_{spu} and X_{cau} have stronger relatedness (which means a bigger), the increment of the prediction model error bound led by X_{spu} is bigger.

4.2 Policy Evaluation Error Bound

Although in most cases, an accurate model ensures a good performance in MBRL, the model error bound is still an indirect evaluation compared to the policy evaluation error bound for MBRL. In this subsection, we apply the spurious theorem (Theorem 4.4) to offline MBRL and provide the policy evaluation error bound with the number of spurious variables.

Suppose that the state value and reward are bounded that $SS_{t;i}S_{1}$ B S_{max} ; R_{t} B R_{max} , let $_{max}$ denote the maximum of and $_{max}$ denote the maximum of $S_{spu} S_{1}$, we have the policy evaluation error bound in Theorem 4.5.

Figure 1: The architecture of FOCUS. Given offline data, FOCUS learns a p value matrix by KCI test and then gets the causal structure by choosing a p threshold. After combining the learned causal structure with the neural network, FOCUS learns the policy through an offline MBRL algorithm.

Theorem 4.5 (RL Spurious Theorem). Given an MDP with the state dimension n_s and the action dimension n_a, a data-collecting policy π_D, let M^* denote the true transition model, M denote the learned model that M^i predicts the i^{th} dimension with spurious variable set spu_i and causal variables cau_i, i.e., $\hat{S}_{t+1,i} = M^i((S_t, A_t) \times \mathbb{1}_{cau_i \cup spu_i})$. Let V^M_π denote the policy value of the policy π in model M and correspondingly $V^{M^*}_\pi$. For an arbitrary bounded divergence policy π, i.e. $\max_S D_{KL}(\pi(\cdot|S), \pi_D(\cdot|S)) \le \epsilon$, we have the policy evaluation error bound:

$$|V^M_\pi - V^{M^*}_\pi| \le \frac{2\sqrt{2}R_{max}}{(1-\gamma)^2}\sqrt{\epsilon} + \frac{\gamma R_{max}}{2(1-\gamma)^2}|S|_{max}\, n_s\, c\,((1-\lambda_{max}) + \lambda_{max}\, n_s(n_s+n_a) R_{spu})$$

where $R_{spu} = \frac{\sum_{i=1}^{n_s}|spu_i|}{n_s(n_s+n_a)}$, which represents the spurious variable density, that is, the ratio of spurious variables in all input variables.

Theorem 4.5 shows the relation between the policy evaluation error bound and the spurious variable density, which indicates that:

- When we use a non-causal model that all the spurious variables are input, R_{spu} reaches its maximum value $R_{spu} = 1$. By contrast, in the optimal causal structure, R_{spu} reaches its minimum value of 0.
- The density of spurious variables R_{spu} and the correlation strength of spurious variables λ_{max} both influence the policy evaluation error bound. However, if we exclude all the spurious variables, i.e., $R_{spu} = 0$, the correlation strength of spurious variables will have no effect.

5 ALGORITHM

After demonstrating the necessity of a causal environment model in offline RL, in this section we propose a practical offline MBRL algorithm, FOCUS, to illustrate the feasibility of learning and using causal structure in offline RL. First, we assume the Causal Markov condition and Faithfulness in the environment transition, with which we can use conditional independence tests to infer the causal graph (Spirtes et al., 2000b). Second, we claim that the offline data and the data obtained through the learned policy share the same causal structure, through which the learned structure can be applied in unseen states. Specifically, policy preference affects the relations between variables by causing quantitative changes in causal relations and spurious relations in independent relations. Thanks to the ability to distinguish between spurious correlations and causal influences, the policy preference will not result in qualitative changes in the causal structure. Consequently, the above statement holds true in offline RL.

Our algorithm consists of two steps, namely, discovering the causal structure from offline data and properly merging the discovered structure with an offline MBRL algorithm. In the first step, offline RL features restrict the selection of causal discovery methods. Specifically, the methods with intervention or randomized experiments are not available because interactions with environments are prohibited. In the approaches for observational data, score-based methods rely on the data from distinct sampling policies and the form of causal mechanisms, neither of which is present in offline data gathered from a single policy. Constraint-based methods do not presuppose any particular form of causal mechanisms, but cannot distinguish structures in one Markov equivalence class and may be inefficient due to many independence tests. Therefore FOCUS extends the PC algorithm (derived from constraint-based methods) and addresses its flaws by incorporating sequential information. In the second step, FOCUS initializes the environment model with the learned causal structure and then learns the environment model as well as the policy.

5.1 PRELIMINARY

Conditional Independence Test. Independence and conditional independence (CI) play a central role in causal discovery (Pearl et al., 2000; Spirtes et al., 2000a; Koller & Friedman, 2009). Generally speaking, the CI relationship $X \perp Y \mid Z$ allows us to drop Y when constructing a probabilistic model for X with (Y, Z). There are multiple CI testing methods for various conditions, which provide the correct conclusion only given the corresponding condition. The kernel-based Conditional Independence test (KCI-test) (Zhang et al., 2011) is proposed for continuous variables without assuming a specific functional form between the variables as well as the data distributions.

Conditional Variables. Besides the specific CI test method, the conclusion of conditional independence testing also depends on the conditional variable Z, that is, different conditional variables can lead to different conclusions. Taking the triple (X, Y, Z) as an example, there are three typical structures, namely, *Chain*, *Fork*, and *Collider* as shown in Fig 2. *Chain*: There exists causation between X and Y but conditioning on Z leads to independence. *Fork*: There does not exist causation between X and Y but not conditioning on Z leads to non-independence. *Collider*: There does not exist causation between X and Y but conditioning on Z leads to non-independence.

Figure 2: The three basic structures for (X, Y, Z).

5.2 CAUSAL STRUCTURE LEARNING

Applying the Independence Test in RL. Based on the preliminaries, given the two target variables X, Y and the condition variable Z, the KCI test returns a probability value $p = f_{KCI}(X, Y, Z) \in [0, 1]$, which measures the probability that X and Y are conditionally independent given the condition Z. To transform a probability into a binary conclusion of whether the causation exists, we design a threshold p^* that:

$$\text{Causation}(X, Y) = \mathbb{I}(f_{KCI}(X, Y, Z) \le p^*) \in \{0, 1\},$$

where $\text{Causation}(X, Y) = 1$ represents independence and 0 represents that causation exists. Details of choosing p^* can be found in Appendix B.1.

In model learning of RL, variables are composed of states and actions of the current and next timesteps and the causal structure refers to whether a variable in t timestep (e.g., the i^{th} dimension, X_t^i) causes another variable in $t+1$ timestep (e.g., the j^{th} dimension, X_{t+1}^j). With the KCI test, we get the causal relation through the function $\text{Causation}(\cdot, \cdot)$ for each variable pair (X_t^i, X_{t+1}^j) and then form the causal structure matrix G:

$$G_{i,j} = \text{Causation}(X_t^i, X_{t+1}^j),$$

where $G_{i,j}$ is the element in row i and column j of G.

Choosing the Conditional Variable in RL. As stated in preliminaries, unsuitable conditional variables can reverse the conclusion of independence testing. The conditional variable set must include the intermediate variable of *Chain* and the common parent variable of *Fork*, but not the common son variable of *Collider*. Traditionally, the independence test traverses all possible combinations of the conditional variables and then reaches a conclusion, which is inefficient. However, in RL we can reduce the number of conditional independence tests by imposing the restriction that the future cannot cause the past. Actually, this constraint restricts the possible conditional variable sets to a tiny number. Consequently, we can have a classified discussion for every feasible collection of conditional variables. For simplicity, we eliminate two types of scenarios from the discussion:

- **Impossible situations.** We exclude some impossible situations as Fig 3 (i) (bottom left) by the temporal property of data in RL. Specifically, the direction of the causation cannot be from the variable of $t+1$ time step to that of t time step because the effect cannot happen before the cause.
- **Compound situations.** We only discuss the basic situations and exclude the compound situations, e.g., Fig 3 (j) (bottom right), which is a compound of (a) and (c). It is because in such compound situations, the target variables X_t^i and X_{t+1}^j have direct causation (or it can not be a compound situation) and the independence testing only misjudges independence as non-independence but not non-independence as independence.

As seen in Fig 3, we list all conceivable circumstances involving target variables X_t^i, X_{t+1}^j and condition variable $X_{t\sim t+1}^k$ in the environment model. Based on the preliminary knowledge of causal discovery, we investigate the following fundamental situations:

Top Line: In (a)(b), whether X_t^k is included in the conditional variable set does not affects the conclusion of causation; In (c), although X_t^k is an intermediate variable in a Chain and conditioning on X_t^k leads to the conclusion of independence of X_t^i and X_{t+1}^j, the causal parent set of X_{t+1}^j will include X_t^k when testing the causal relation between X_t^k and X_{t+1}^j, which can offset the influence of excluding X_t^i. In (d), conditioning on Z is necessary for getting the correct conclusion of causation since X_t^k is the common causal parent in a Fork structure.

Bottom Line: In (e)(f), whether X_{t+1}^k is included in the conditional variable set does not affects the conclusion of causation; In (g), not conditioning on X_{t+1}^k is necessary to get the correct conclusion of causation since X_{t+1}^k is the common son in a Collider structure; In (h), although X_{t+1}^k is an intermediate variable in a Chain and not conditioning on X_{t+1}^k leads to the conclusion of non-independence of X_t^i and X_{t+1}^j, including X_t^i in the causal parent set of X_{t+1}^j will not induce any problem since X_t^i does indirectly cause X_{t+1}^j.

Figure 3: The basic, impossible and compound situations of the causation between target variables and condition variables. In the basic situations, Top Line: (a)-(d) list the situations that the condition variable X^k is in the t time step. Bottom Line: Similarly, (e)-(h) list the situations that the condition variable X^k is in the $t+1$ time step.

Based on the classified discussion above, we can conclude our principle for choosing conditional variables in RL that:

- Condition on the other variables in t time step.
- Do not condition on the other variables in $t+1$ time step.

5.3 Combining Learned Causal Structure with An Offline MBRL Algorithm

We combine the learned causal structure with an offline MBRL algorithm, MOPO (Yu et al., 2020), to create a causal offline MBRL algorithm as in Fig 1. The entire learning procedure can be found in Algorithm 1 and Algorithm 2 (Appendix B.2). Notice that our causal model learning method can be combined with any offline MBRL algorithm theoretically. More implementation details and hyperparameter values are summarized in Appendix B.1.

6 Experiments

In order to demonstrate that (1) FOCUS enables learning a causal environment model in offline RL and (2) a causal environment model can outperform a plain environment model and other related methods in offline RL, we evaluate (1) causal structure learning and (2) policy learning on the Toy Car Driving and MuJoCo benchmarks. We evaluate FOCUS on the following indexes: (1) The accuracy, efficiency and robustness of causal structure learning. (2) The policy return and generalization ability in offline MBRL.

Baselines. We compare FOCUS with the sota offline MBRL algorithm, MOPO, and other online RL algorithms that also learn causal structure. (1) MOPO (Yu et al., 2020) is a popular and well-known offline MBRL algorithm that outperforms standard model-based RL algorithms and prior sota model-free offline RL algorithms on existing offline RL benchmarks. The central idea of MOPO is to artificially penalize rewards by the uncertainty of model predictions, hence avoiding erroneous predictions in unseen states. MOPO can be seen as the blank control with a plain environment model. (2) Learning Neural Causal Models from Unknown Interventions (LNCM) (Ke et al., 2019) is an online MBRL algorithm, in which the causal structure learning method can be transformed to the offline setting with a simple adjustment. We take LNCM as an example to show that an online method cannot be directly converted into offline RL algorithms.

Environment. **Toy Car Driving.** Toy Car driving is a typical RL environment where the agent can control its direction and velocity to finish various tasks including avoiding obstacles and navigating. In this paper, we use a 2D Toy Car driving as the RL environment where the task of the car is to arrive at the destination (The visualization can be found in Appendix C.1). The state includes the direction d, the velocity v, the velocity on the x-axis v_x, the velocity on the y-axis v_y and the position (p_x, p_y). The action is the steering angle a. We design the underlying causal structure to better demonstrate how spurious relations appear and highlight their influence in model learning (The structure can be found in Appendix C.1). **MuJoCo.** The MuJoCo (Todorov et al., 2012) is the most popular benchmark for evaluating performance in continuous controlling, where the variables of the state represent the positions, angles, and velocity of the agent. Each dimension in MuJoCo of the state has a specific meaning and is highly abstract, which provides the convenience of causal structure learning.

Offline Data. We prepare three offline data sets, *Random*, *Medium*, and *Replay* for the Car Driving and MuJoCo. *Random* represents that data is collected by random policies. *Medium* represents that data is collected by a fixed but not well-trained policy, which is the least diverse. *Medium-Replay* is a collection of data that is sampled during training of the *Medium* policy, which is the most diverse. The heat map of the data diversity is shown in Appendix C.1.

Table 1: The results on causal structure learning of our model and the baselines. Both the accuracy and the variance are calculated by five times experiments. *FOCUS (-KCI)* represents FOCUS with a linear independence test. *FOCUS (-CONDITION)* represents FOCUS with choosing all other variables as conditional variables.

INDEX	FOCUS	LNCM	FOCUS(-KCI)	FOCUS(-CONDITION)
ACCURACY	0.993	0.52	0.62	0.65
ROBUSTNESS	0.001	0.025	0.173	0.212
EFFICIENCY(SAMPLES)	1×10^{6}	1×10^{7}	1×10^{6}	1×10^{6}

6.1 CAUSAL STRUCTURE LEARNING

We compare FOCUS with baselines on the causal structure learning with the indexes of *accuracy*, *efficiency*, and *robustness*. The accuracy is evaluated by viewing the structure learning as a classification problem, where causation represents the positive example and independence represents the negative example. The efficiency is evaluated by measuring the samples for getting a stable structure. The robustness is evaluated by calculating the variance in multiple experiments. The results in Table 1 show that FOCUS surpasses LNCM in accuracy, robustness, and efficiency in causal structure learning. Noticed that LNCM also has a low variance because it predicts the probability of existing causation between any variable pairs with around **50%**, which means that its robustness is meaningless.

Table 2: The comparison on converged policy return in the two benchmarks. The detailed training curves are in Appendix C.1.

ENV	CAR DRIVING			MUJOCO(INVERTED PENDULUM)		
	RANDOM	MEDIUM	REPLAY	RANDOM	MEDIUM	REPLAY
FOCUS	68.1 ± 20.9	58.9 ± 41.3	86.2 ± 18.2	23.5 ± 17.9	24.9 ± 14.1	49.2 ± 19.0
MOPO	30.3 ± 49.9	50.1 ± 34.2	46.2 ± 28.1	8.5 ± 6.2	2.5 ± 0.08	43.4 ± 7.7
LNCM	9.9 ± 42.5	5.4 ± 32.5	11.4 ± 24.0	13.3 ± 0.9	3.1 ± 0.7	16.3 ± 6.4

6.2 POLICY LEARNING

Policy Return. We evaluate the performance of FOCUS and baselines in the two benchmarks on three typical offline data sets. The results in Table 2 show that FOCUS outperforms baselines by a significant margin in most data sets. In *Random*, FOCUS has the most significant performance gains to the baselines in both benchmarks because of the accuracy of causal structure learning in FOCUS.

By contrast, in Medium-Replay, the performance gains of FOCUS are least since the high data diversity in Medium-Replay leads to weak relatedness of spurious variables (corresponds to small), which veri es our theory. In Medium, the results in the two benchmarks are different. In Car Driving, the relatively high score of LNCM does not mean that LNCM is the best but all three fail. The failure indicates that extremely biased data makes even the causal model fail to generalize. However, the success of FOCUS in the Inverted Pendulum indicates that causal environment models depend less on the data diversity since FOCUS can still reach high scores in such a biased dataset where the baselines fail. Here we only provide the results in Inverted Pendulum but not all the environments in MuJoCo due to the characteristics of the robot control, speci cally the frequency of observations, which we present a detailed description in Appendix C.1.

Generalization Ability. The generalization ability of FOCUS refers to whether it can learn a good policy from the data with limited data size and low data diversity. Therefore we designed datasets from 1% to 100% of the original data size and datasets with a mix of 20% to 80% other datasets, where we can compare FOCUS and baselines in datasets with different sizes and diversities. The results in Fig 4 (Top) show that the advantage of FOCUS over MOPO is much more signi cant in small data size. In the dataset of 1% size, the advantage of FOCUS is relatively not signi cant because the size is too small. The results in Fig 4 (Bottom) show that FOCUS can performs well with a small ratio of Medium-Replay data while the baseline performs well only with a big ratio, which indicates that FOCUS is less dependent on the diversity of data. Related experiments on more environments can be found in Appendix C.2.

Figure 4: Top: The comparison for data size. The X % in the x-axis represents that the data size is X % of the original size. The ratio Y % in the y-axis represents the score ratio of FOCUS over the baseline MOPO. Bottom: The comparison for data diversity. The dataset is produced by mixing up Medium-Replay and Medium with different ratios. The X % in the x-axis represents that the data is mixed by $(100 - X)$% of the Medium and X % of the Medium-Replay.

6.3 Ablation Study

To evaluate the contribution of each component, we perform an ablation study for FOCUS. The results in Table 1 show that the KCI test and our principle of choosing conditional variables contribute to the causal structure learning of both accuracy and robustness.

7 Conclusion

In this paper, we point out that the spurious correlations hinder the generalization ability of current of ine MBRL algorithms, and that incorporating the causal structure into the model can improve generalization by removing spurious correlations. We provide theoretical support for the statement that utilizing a causal environment model reduces the generalization error bound in of ine RL. We also propose a practical algorithm, FOCUS, to address the problem of learning causal structure in of ine RL. The main idea of FOCUS is to leverage conditional independence tests for causal discovery, which does not need further assumptions on the causal mechanism. In FOCUS, we address the dif culties of extending the PC algorithm in of ine RL, particularly to reduce the number of independence tests by leveraging sequential information. Extensive experiments on the typical benchmarks demonstrate that FOCUS performs accurate and robust causal structure learning, surpassing of ine RL baselines by a signi cant margin.

We would like to note that: In our theoretical results (Theorem 4.4 and 4.5), we assume that the true causal structure is already known. However, in practice, we must learn it from data before applying it (section 5), which will introduce additional theoretical errors. As it is recognized that quantifying the uncertainty in the learned causal structure from data is a dif cult task, we will derive the generalization error bound with the learned causal structure as part of our future study.

References

Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5): 679–684, 1957.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS'19)*, pp. 11693–11704, 2019.

Mark Edmonds, James Kubricht, Colin Summers, Yixin Zhu, Brandon Rothrock, Song-Chun Zhu, and Hongjing Lu. Human causal transfer: Challenges for deep reinforcement learning. In *Proceedings of the 40th Annual Meeting of the Cognitive Science Society (CogSci'18)*, 2018.

O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, David Sontag, Finale Doshi-Velez, and L. A. Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, 2019.

David Heckerman, Christopher Meek, and Gregory Cooper. A bayesian approach to causal discovery. In *Innovations in Machine Learning*, pp. 1–28. Springer, 2006.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000. ISSN 00401706.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *CoRR*, 1910.01075, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)* 2020.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009. ISBN 978-0-262-01319-2.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *Proceedings of the 6th International Conference on Learning Representations, (ICLR'18)*, 2018.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, 1910.07113, 2019.

Judea Pearl et al. Models, reasoning and inference. *Cambridge University Press*, 19:2, 2000.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000a. ISBN 978-0-262-19440-2.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2000b.

Josh Tenenbaum. Building machines that learn and think like people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'18)*, pp. 5, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 5026–5033, 2012.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA'17)*, pp. 1714–1721, 2017. doi: 10.1109/ICRA.2017.7989202.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, pp. 15737–15749, 2020.

Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *CoRR*, 1805.04687, 2018.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In *Advances in Neural Information Processing Systems 33 (NeurIPS'20)*, 2020.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI'11)*, pp. 804–813, 2011.

A THEORY

Definition A.1 (Optimization objective in data distribution $\mathcal{D}$).

$$\min \mathbb{E}_{X,Y\sim\mathcal{D}} \left(X - Y \right)^2. \tag{6}$$

Definition A.2 (Optimization objective in data $\mathcal{D}_{train}$).

$$\min \mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left(X - Y \right)^2. \tag{7}$$

Definition A.3 (Optimization objective in data $\mathcal{D}_{train}$ with regularization).

$$\min \mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left(X - Y \right)^2 + k\|Y\|^2. \tag{8}$$

Lemma A.4. Given that $\omega_{cau}X^*$ is the optimal solution of Problem 1, suppose that in $\mathcal{D}_{train}$, $X_{spu} = \hat{}X\omega_{cau} + \epsilon_{spu}$ where $\mathbb{E}_{\mathcal{D}_{train}}\epsilon_{spu} = 0$ and $\epsilon_{spu} \perp X$, 0, we have that $\hat{\omega}_{spu} < \omega_{cau} X^{*}_{spu} + \omega_{spu}$ is also an optimal solution of Problem 2 for any.

$$\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\omega_{cau} X^{*} - Y)^2 \right] \geq \mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\hat{\omega}_{spu} - Y)^2 \right]$$

Proof.

$$
\begin{aligned}
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\omega_{cau} X^{*} - Y)^2 \right] \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\omega_{cau} X^{*} \quad {}_{spu} \quad {}_{spu}) - Y)^2 \right] \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\omega_{cau} X^{*} \quad {}_{spu}(\hat{}X\omega_{cau} \quad {}_{spu}) - Y)^2 \right] \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\omega_{cau} X^{*} \quad {}_{spu}(\hat{}X_{spu} \quad {}_{spu}) - Y)^2 \right] \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X(\omega_{cau} X^{*} \quad {}_{spu})) \quad \hat{}X\omega_{spu}) - Y)^2 \right] \\
&\qquad (\text{Since } \mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \; {}_{spu} = 0) \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X(\omega_{cau} X^{*} \quad {}_{spu}) \quad \omega_{spu}) - Y)^2 \right] \\
&\mathbb{E}_{X,Y\sim\mathcal{D}_{train}} \left[(X\hat{\omega}_{spu} - Y)^2 \right] \\
&(\text{Let } \hat{\omega}_{spu} \text{ denote } \omega_{cau} X^{*} \quad {}_{spu}) \quad \omega_{spu})
\end{aligned}
$$

□

Lemma A.5 (Lemma). Given as the coefficient in Lemma 4.1, and k in Problem 3 chosen by Hoerl-Kennard formula, we have the solution of in Problem 3 that:

$$\frac{{}^{*} \quad {}_{spu}}{{}^{*2} \quad {}^{2}_{spu} \quad 1 \quad \frac{{}^{2}_{spu}}{{}^{2}_{cau}}\hat{}1 \quad \frac{1}{\hat{}\,{}^{*}(\,)^{2}})} \tag{9}$$

Proof. Since the solution of the ridge regression is

$$\hat{}(k) = (X^T X + kI)^{-1} X^T Y;$$

we take $\hat{\omega}_{spu}$ into this solution and get:

$$\frac{{}^{2}_{cau} \quad {}^{*} \quad {}_{spu}\, k}{{}^{2}_{cau} \quad {}^{2}_{spu} \quad {}^{2}_{cau} \quad {}^{2}_{spu}\, k \quad {}^{2}_{cau}\, k \quad {}^{2}_{spu}\, k \quad k^2} \tag{10}$$

Since k is chosen by Hoerl-Kennard formula that $k = \frac{\sigma^2_{spu}}{\hat{\ } \ddagger \bullet 2}$, we have:

$$
\begin{aligned}
& \frac{\sigma^2_{cau} \ \ddagger \ \ \sigma_{spu}}{\sigma^2_{cau} \ \sigma^2_{spu} \ k \ \ \sigma^2_{cau} \ \sigma^2_{spu} \ \ \sigma^2_{cau} \ \ \sigma^2_{spu} \ \ k} \\
& \frac{\sigma^2_{cau} \ \ddagger \ \ \sigma_{spu}}{\sigma^2_{cau} \ \sigma^2_{spu} \ \hat{\ } \frac{\sigma^2_{spu}}{\hat{\ } \ddagger \bullet 2} \bullet \ \ \sigma^2_{cau} \ \sigma^2_{spu} \ \ \sigma^2_{cau} \ \ \sigma^2_{spu} \ \ \frac{\sigma^2_{spu}}{\hat{\ } \ddagger \bullet 2}} \\
& \frac{\sigma^2_{cau} \ \ddagger \ \ \sigma_{spu}}{\sigma^2_{cau} \ \ddagger^2 \ \ \sigma^2_{cau} \ \sigma^2_{spu} \ \ \sigma^2_{cau} \ \ \sigma^2_{spu} \ \ \frac{\sigma^2_{spu}}{\hat{\ } \ddagger \bullet 2}} \\
& \frac{\sigma^2_{cau} \ \ddagger \ \ \sigma_{spu}}{\sigma^2_{cau} \ \hat{\ } \ \ddagger^2 \ \ \sigma^2_{spu} \ \ 1 \bullet \ \ \sigma^2_{spu} \ \ \frac{\sigma^2_{spu}}{\hat{\ } \ddagger \bullet 2}} \\
& \frac{\ddagger \ \ \sigma_{spu}}{\ddagger^2 \ \ \sigma^2_{spu} \ \ 1 \ \ \frac{\sigma^2_{spu}}{\sigma^2_{cau}} \hat{\ } 1 \ \ \frac{1}{\hat{\ } \ddagger \bullet 2} \bullet}
\end{aligned}
$$

□

Proposition A.6. Given as Formula 4, we have

$$\frac{1}{2} B \quad B \frac{1}{2}:$$

Proof.

$$
\begin{aligned}
S \ S & \frac{S \ddagger \ \ \sigma_{spu} S}{S \ddagger^2 \ \ \sigma^2_{spu} \ \ 1 \ \ \frac{\sigma^2_{spu}}{\sigma^2_{cau}} \hat{\ } 1 \ \ \frac{1}{\hat{\ } \ddagger \bullet 2} \bullet S} \\
B & \frac{S \ddagger \ \ \sigma_{spu} S}{S \ddagger^2 \ \ \sigma^2_{spu} \ \ 1 S \ \ S \frac{\sigma^2_{spu}}{\sigma^2_{cau}} \hat{\ } 1 \ \ \frac{1}{\hat{\ } \ddagger \bullet 2} \bullet S} \\
B & \frac{S \ddagger \ \ \sigma_{spu} S}{S \ddagger^2 \ \ \sigma^2_{spu} \ \ 1 S} \\
B & \frac{S \ddagger \ \ \sigma_{spu} S}{S 2 \ \ddagger \ \ \sigma_{spu} \ \ 1 S} \\
B & \frac{1}{2}
\end{aligned}
$$

So we have : $\frac{1}{2}$ B B $\frac{1}{2}$. □

Theorem A.7 (Spurious Theorem). Let D ~^ X; Y •• denote the data distribution, $\hat{\ }_{spu}$ denote the solution in Lemma 4.1 with in Lemma 4.2, and $\hat{Y}_{spu}$ X $\hat{\ }_{spu}$ denote the prediction. Suppose that the data value is bounded: $S X_i S$ B X_{max} ; i 1; ; n and the error of optimal solution $_{cau}$ is also bounded: $S_{cau} S$ B $_c$, we have the model prediction error bound:

$$E_{\hat{\ } X;Y \bullet \ D} \ \hat{\ } S \hat{Y}_{spu} \quad Y S \bullet \ S X \ \ B X_{max} \ S S \hat{\ } S_{spu} S \quad 1 \bullet \quad _c: \tag{11}$$

Proof. Let $\hat{Y}_{cau}$ denote $f(X \odot \omega_{cau}^{*})$, we have

$$
\begin{aligned}
&\mathbb{E}_{(X,Y)\sim D_{test}}\left[\|\hat{Y}_{spu} - Y\|_1 \mid X\right] \\
&= \mathbb{E}_{(X,Y)\sim D_{test}}\left[\|\hat{Y}_{spu} - \hat{Y}_{cau} + (\hat{Y}_{cau} - Y)\|_1 \mid X\right] \\
&\le \mathbb{E}_{(X,Y)\sim D_{test}}\left[\|\hat{Y}_{spu} - \hat{Y}_{cau}\|_1 \mid X\right] + \mathbb{E}_{(X,Y)\sim D_{test}}\left[\|\hat{Y}_{cau} - Y\|_1 \mid X\right] \\
&\le \mathbb{E}_{(X,Y)\sim D_{test}}\left[\|X \odot (\omega_{cau} + \omega_{spu}) - \omega_{spu}\|_1 \mid X\right] + \epsilon_c \\
&= \mathbb{E}_{(X,Y)\sim D_{test}}\left[\|X \odot (\omega_{spu} - \omega_{spu})\|_1 \mid X\right] + \epsilon_c \\
&\le \mathbb{E}_{(X,Y)\sim D_{test}}\left[X_{max}\,\|\mathbf{1}^{\dagger}\|\,\|\omega_{spu} - \omega_{spu}\|_1 \mid X\right] + \epsilon_c \\
&\le \mathbb{E}_{(X,Y)\sim D_{test}}\left[X_{max}\,\|\mathbf{1}^{\dagger}(S_{spu}\|_1 (1-\delta)) \mid X\right] + \epsilon_c \\
&= X_{max}\,\|S_{spu}\|_1(1-\delta) + \epsilon_c
\end{aligned}
$$

∎

Theorem A.8 (RL Spurious Theorem). Given an MDP with the state dimension n_s and the action dimension n_a, a data-collecting policy π_D, let M^* denote the true transition model, M denote the learned model that M^i predicts the i^{th} dimension with spurious variable sets spu_i and causal variables cau_i, i.e., $\hat{S}_{t+1;i} = M^i((S_t; A_t) \odot \omega_{cau_i \cup spu_i})$. Let V_π^M denote the policy value of the policy π in model M and correspondingly $V_\pi^{M^*}$. For any bounded divergence policy π, i.e. $\max_S D_{KL}(\pi(\cdot|S), \pi_D(\cdot|S)) \le \epsilon_\pi$, we have the policy evaluation error bound:

$$
|V_\pi^M - V_\pi^{M^*}| \le \frac{2\sqrt{2}R_{max}\gamma}{(1-\gamma)^2}\sqrt{\epsilon_\pi} + \frac{R_{max}\gamma}{2(1-\gamma)^2}S_{max}\left(n_s\epsilon_c + (1-\delta_{max})\phi_{max}\,n_s(n_s+n_a)R_{spu}\right) \tag{12}
$$

where $R_{spu} = \frac{\sum_{i=1}^{n_s}|spu_i|}{n_s(n_s+n_a)}$, which represents the spurious variable density, that is, the ratio of spurious variables in all input variables .

Proof. Before proving, we first introduce three lemmas:

Lemma A.9.

$$
\begin{aligned}
|V_\pi^M - V_\pi^{M^*}| &\le |V_\pi^{M^*} - V_{\pi_D}^{M^*}| + |V_\pi^M - V_{\pi_D}^{M}| + |V_{\pi_D}^{M^*} - V_{\pi_D}^{M}| \\
&\le \frac{2\sqrt{2}R_{max}\gamma}{(1-\gamma)^2}\sqrt{\epsilon_\pi} + |V_{\pi_D}^M - V_{\pi_D}^{M^*}|
\end{aligned}
$$

Lemma A.10.

$$
|V_{\pi_D}^M - V_{\pi_D}^{M^*}| \le \frac{R_{max}}{1-\gamma}\sum_s |d_{\pi_D}^M(s) - d_{\pi_D}^{M^*}(s)|\sum_a \pi_D(a|s)
$$

Lemma A.11.

$$
|d_{\pi_D}^M(s) - d_{\pi_D}^{M^*}(s)| \le \frac{\gamma}{(1-\gamma)}\sum_{s,a,s'}|M(S_t;A_t) - M^*(S_t;A_t)|\,\pi_D(a|s)\,d_{\pi_D}^{M^*}(s)
$$

The detailed proof of these lemmas can be found in (Xu et al., 2020), which is omitted in this paper. Based on the model prediction error bound in Theorem 4.4, we have:

$$
\begin{aligned}
\|M(S_t;A_t) - M^*(S_t;A_t)\| &= \sum_{i=1}^{n_s}|M^i(S_t;A_t) - M^{*,i}(S_t;A_t)| \\
&\le \sum_{i=1}^{n_s} S_{max}\left(\epsilon_c + (1-\delta_{max})\phi_{max}|spu_i|\right) \\
&= S_{max}\left(n_s\epsilon_c + (1-\delta_{max})\phi_{max}\sum_{i=1}^{n_s}|spu_i|\right) \\
&= S_{max}\left(n_s\epsilon_c + (1-\delta_{max})\phi_{max}\,n_s(n_s+n_a)R_{spu}\right)
\end{aligned}
$$

With above lemmas, we have:

$$S V^M - V^{M^*} S B \frac{2\gamma\sqrt{2}R_{max}}{(1-\gamma)^2} - \frac{R_{max}}{2(1-\gamma)^2} S_{max} \; n_s \; c \; (\; max \; 1) \; max \; n_s(n_s \; n_a) R_{spu}$$

□

B ALGORITHM

B.1 CHOOSING THE THRESHOLD OF P-VALUE

To be fair, we share a common p^* for the testing between any two variables. The choice of p^* significantly influences the accuracy of causal discovery that too small and too big both lead to causal misspecification. The intuition behind our choosing principle is that there is a significant gap in p value between the causal relation and non-causal relation. Based on this intuition, we partition the probability range $[0, 1]$ into several intervals $[0, p_1), [p_1, p_2), \ldots, [p_n, 1]$ according to the sorted p values $(p_i)_{i=1}^n$ and design p^* by the formula:

$$p^* = \arg\max_{p_i} \frac{p_{i-1}}{i-1} - \frac{p_i}{i}. \qquad (13)$$

If we only consider the biggest gap between p_i, then we will easily choose a big but improper p^* due to the distribution of p_i in some intervals (e.g., [0.5,1]) may be very sparse and thus leads to a big gap.

(a) Random (b) Medium (c) Medium-Replay

Figure 5: The heat map of the three offline data sets. The high brightness represents high data density.

B.2 CAUSAL STRUCTURE NETWORK

The complete process is shown in Algorithm 1, where the details of Causal Structure Network is shown in Algorithm 2.

Algorithm 1 Causal Model Framework for Offline MBRL

Input: offline data set $D = \{(s_t, a_t, s_{t+1}, r_t)\}$, model $M(\cdot;\theta)$;
Stage 1: Causal Structure Learning
Get p value matrix G_p by KCI testing.
Get the threshold p^* by G_p.
Get causal structure mask matrix G by the threshold p^*.
Stage 2: Offline Reinforcement Learning
Choose an offline model-based reinforcement learning algorithm $Algo(\,)$ and replace its model $M(\cdot)$ by $M_{Causal}(\cdot; G; M)$ (Algorithm 2 in Appendix).
Obtain the optimal policy $\pi^* = Algo(D)$.
Return π^*

Algorithm 2 Causal Structure Network $M_{Causal}(\cdot)$

Input: states $s_t \in R^{n_s}$, action $a_t \in R^{n_a}$,
causal structure mask matrix $G \in \{0, 1\}^{(n_s+n_a) \times n_s}$,
Make $M_i(\cdot; \theta_i)$ as the copy of the basic model $M(\cdot; \theta)$, where $i = 1, \dots, n_s$.
for $i = 1$ to n_s **do**
  Let $G_{:,i}$ denote the i^{th} column of G
  Get the masked input $X = (s_t, a_t) \cdot G_{:,i}$
  Get prediction $\tilde{Y} = M_i(X; \theta_i) \in R^{n_s}$
  Let Y_i denote the i^{th} element of $\tilde{Y}$.
end for
Return $Y = (Y_i)_{i=1}^{n_s}$.

C EXPERIMENTS

C.1 ENVIRONMENT DETAILS

The heat map of the data diversity is shown in Fig 5. In *Random*, the data is clustered around the origin. In *Medium*, the data is gathered on a fixed trajectory from the origin to the destination. In *Medium-Replay*, the data is much more diverse where a lot of unseen data in above data sets is also sampled.

The visualization of the state in Car Driving and the ground truth of its causal graph are shown in Fig 9.

For example, when the velocity v_{t-1} maintains stationary due to an imperfect sample policy, $(v_x)_t$ and $(v_y)_t$ have strong relatedness that $(v_x)_t^2 + (v_y)_t^2 = v_{t-1}^2$ and one can represent the other. Since we design that $(p_y)_{t+1} = (p_y)_t + (v_y)_t$, $(v_x)_t$ and $(p_y)_{t+1} - (p_y)_t$ also have strong relatedness, which leads to that $(v_x)_t$ becomes a spurious variable of $(p_y)_{t+1}$ given $(p_y)_t$, despite that $(v_x)_t$ is not the causal parent of y_{t+1}. By contrast, when the data is uniformly sampled with various velocities, this spuriousness will not exist.

MuJoCo formulates robot control into MDPs with discrete timestep via equal interval sampling of the continuous-time. Therefore, for each timestep, s_{t+1} is the result of numerous times of simulation based on s_t with repeated action a_t. Even if spurious variables are existed in one time of simulation, after numerous simulations, the causal effect will be propagated to almost variables, which leads to a full-connection causal graph ($R_{spu} = 0$). Therefore FOCUS degrades into vanilla MOPO in this scenario, which is meaningless to test. Fortunately, after analyzing the propagate progress of the dynamics, we found that the *Inverted Pendulum* is a special case where the causal graph will keep sparse after numerous simulations.

Figure 6: The visualization of the example. The red dotted arrow presents that $(v_x)_t$ is a spurious variable for $(p_y)_{t+1}$.

MuJoCo formulates robot control into MDPs with discrete timestep via equal interval sampling of the continuous-time. Therefore, for each timestep, s_{t+1} is the result of numerous times of simulation based on s_t with repeated action a_t. Even if spurious variables are existed in one time of simulation, after numerous simulations, the causal effect will be propagated to almost variables, which leads to a full-connection causal graph ($R_{spu} = 0$). Therefore FOCUS degrades into vanilla MOPO in this scenario, which is meaningless to test. Fortunately, after analyzing the propagate progress of the dynamics, we found that the *Inverted Pendulum* is a special case where the causal graph will keep sparse after numerous simulations.

C.2 EXPERIMENT RESULT DETAILS

The detailed training curves are shown in Fig 7. The detailed comparisons on data size are shown in Fig 8.

(a) Random (b) Medium (c) Medium-Replay

(d) Random (e) Medium (f) Medium-Replay

Figure 7: Comparison of FOCUS and the baselines in the two benchmarks. (a)-(c): The comparison in the Car Driving on the three datasets. (d)-(f): The comparison in the Inverted Pendulum of MuJoCo on the three datasets.

(a) Inverted Pendulum (b) Inverted Doubled Pendulum (c) pendulum Swing

Figure 8: Comparison of FOCUS and the baselines in three of ine datasets of three environments.

Figure 9: The visualization of the state and the causal structure for the Car Driving benchmark. **Left:** the Toy Car Driving. The goal of the agent is to arrive at the star-shape destination. **Right:** The ground truth of the causal structure in Toy Car Driving. The state is vector-based and its value is continuous.