Causal Deep Reinforcement Learning Using Observational Data

Wenxuan Zhu\textsuperscript{1}, Chao Yu\textsuperscript{2}\textsuperscript{*} and Qiang Zhang\textsuperscript{1}\textsuperscript{*}
\textsuperscript{1}Dalian University of Technology
\textsuperscript{2}Sun Yat-sen University
zhuwenxuan@mail.dlut.edu.cn, yuchao3@mail.sysu.edu.cn, zhangq@dlut.edu.cn

Abstract
Deep reinforcement learning (DRL) requires the collection of interventional data, which is sometimes expensive and even unethical in the real world, such as in the autonomous driving and the medical field. Offline reinforcement learning promises to alleviate this issue by exploiting the vast amount of observational data available in the real world. However, observational data may mislead the learning agent to undesirable outcomes if the behavior policy that generates the data depends on unobserved random variables (i.e., confounders). In this paper, we propose two deconfounding methods in DRL to address this problem. The methods first calculate the importance degree of different samples based on the causal inference technique, and then adjust the impact of different samples on the loss function by reweighting or resampling the offline dataset to ensure its unbiasedness. These deconfounding methods can be flexibly combined with existing model-free DRL algorithms such as soft actor-critic and deep Q-learning, provided that a weak condition can be satisfied by the loss functions of these algorithms. We prove the effectiveness of our deconfounding methods and validate them experimentally.

1 Introduction
Human beings can learn from observation (e.g., in astronomy) and experimentation (e.g., in physics). For example, people understand the laws of astronomy by observing the movements of celestial bodies and the laws of physics by doing physical experiments. Analogously, agents can learn in these two ways as well. In some cases, however, experimentation (i.e., the collection of interventional data) can be expensive and even unethical, while observational data are easy to obtain. For example, it is unsafe for an agent to learn to drive a car on a real road, but we can easily collect data from a human driving a car with sensors.

Reinforcement learning (RL) [Sutton and Barto, 2018] is generally regarded as an interactive learning process, which means that the agent learns from the interventional data generated by experimentation. As a type of RL methods, offline RL [Kumar et al., 2020; Levine et al., 2020; Siegel et al., 2020; Ernst et al., 2005; Fujimoto et al., 2019; Kumar et al., 2019; Agarwal et al., 2020; Jaques et al., 2019] has been proposed to study how to enable RL algorithms to learn strategies from observational data without interacting with the environment.

Most existing offline RL methods are based on an assumption that $O_1 = O_2 = O_3$, where $o_1 \in O_1$ denotes the observation of the environment in which the offline data is collected, $o_2 \in O_2$ denotes the observation in the offline data, and $o_3 \in O_3$ denotes the observation in the online data where the agent trained using the offline data is tested (In the driving car example, $o_1$ represents the environmental information perceived by the driver, $o_2$ represents the environmental information collected by the sensors, and $o_3$ represents the environmental information perceived by the agent in the testing environment.). This assumption, however, is often difficult to hold in real-world problems. The driver may be able to see more broadly than the sensors, to judge if the road is slippery based on weather conditions, or even to receive traffic conditions around them based on the radio, therefore, it is generally that $O_1 \neq O_2$. If the driver makes a decision based on the information not collected by the sensors, i.e., there are unobserved random variables (confounders) that affect the action and the next sensory observation at the same time, the observational data generated by the driver may be misleading. An agent then learns the wrong dynamics of the environment based on this misleading information, leading to a biased estimation of the value functions and the final policies.

Some researches [Sen et al., 2017; Kallus and Zhou, 2018; Wang et al., 2021; Gasse et al., 2021] in recent years study how to train an agent using offline data with the confounders based on the causal inference techniques. However, most of these researches [Sen et al., 2017; Kallus and Zhou, 2018] only target at the bandit problems. Other approaches propose deconfounding methods in RL settings but they are based on certain assumptions such as linear reward/transition functions [Wang et al., 2021], small state spaces [Gasse et al., 2021], or a complex correlation between the features [Wang et al., 2021]. Therefore, these approaches only work for specific types of offline RL algorithms, and cannot be applied to large and continuous environments (See Section 5 for more
details). To address these problems, we propose two kinds of deconfounding methods based on the importance sampling and causal inference techniques. Our deconfounding methods first estimate a conditional distribution density ratio through the least-squares conditional density estimation (LSCDE) [Sugiyama et al., 2010; Rothfuss et al., 2019], and then adjust the impact of different samples on the loss function to ensure the unbiasedness of the loss function. Specifically, we make the following contributions.

1) Unlike the existing importance sampling techniques in off-policy evaluation (OPE) [Kallus and Zhou, 2020; Gelada and Bellemare, 2019; Hallak and Mannor, 2017], we estimate a conditional distribution density ratio, which keeps constant in the learning process, and thus can be applied to the RL field.

2) In the proposed deconfounding methods, we decouple the deconfounding process from the RL algorithm, i.e., we can uniformly incorporate the conditional distribution density ratio into the loss functions of the offline RL algorithms. In other words, our plug-in deconfounding methods can be combined with existing offline RL algorithms provided that a weak condition is satisfied.

3) Furthermore, since we do not learn a latent-based transition model, and do not assume a complex correlation between the features, our deconfounding methods do not contain a complex computational process, and thus can be applied to large and continuous environments.

4) We prove theoretically that these two deconfounding methods can construct an unbiased loss function w.r.t. the online data, and thus improve the performance of the offline RL algorithms. The experimental results verify that the proposed deconfounding methods are effective: offline RL algorithms using deconfounding methods perform better on datasets with the confounders.

2 Background
In this section, we define a confounded Markov decision process (CMDP) (similar to [Wang et al., 2021]) and two corresponding structural causal models (SCMs) to describe the RL tasks, in which the offline data include unobserved confounders between the action, reward and next state, and the online data include no confounder between these variables. In specific, there are both continuous random variables $X$ and discrete random variables $Y$ in our problems. Without loss of generality, we assume that any discrete random variable belongs to the set of integers. The probability density function $P(x, y)$ of the discrete and continuous random variables is given as follows:

$$P(x, y) = \frac{\partial^p}{\partial x_1 \cdots \partial x_p} \Pr(X \leq x, Y = y), \quad (1)$$

where $X \in \mathbb{R}^p$ and $Y \in \mathbb{Z}^q$. The CMDP for unobserved confounders can be denoted by a nine-tuple $(S, M, A, W, R, P_1, P_2, P_3, \mu_0)$, where $S$ denotes the state space, $M$ denotes the intermediate state space, $A$ denotes the discrete action space, $W$ denotes the confounder space, $R$ denotes the reward space, $P_1(s', r|s, w, m)$ denotes the dynamics of the CMDP, $P_2(w|s)$ denotes the confounder transition distribution, $P_3(m|s, a)$ denotes the intermediate state transition distribution, and $\mu_0(s)$ denotes the initial state distribution. Note that the environment generates the intermediate state $m$ in the process of generating the next state $s'$. For example, a physician prescribes some drugs to a patient. The actual amount of drugs taken by the patient (i.e., $m$) may be different from that prescribed by the physician (i.e., $a$) due to compliance issues. Also note that we integrate the state transition distribution and the reward transition distribution into the dynamics of the CMDP for notational convenience.

The SCM in the offline setting as shown in the first column of Figure 1 can be defined as a four-tuple $(U, V, F, P_0)$, where $U$ is the exogenous variables, which are not visible in the experiment, $V$ is the endogenous variables, including $(s, m, a, w, s', r)$, $F$ is the set of structural functions, including the state-reward transition distribution $P_1(s', r|s, w, m)$, the confounder transition distribution $P_2(w|s)$, the intermediate state transition distribution $P_3(m|s, a)$, and the behavior policy $\pi_0(a|s, w)$, and $P_0$ is the distribution of exogenous variables. The positivity assumption here is that, for $m \in M, a \in A, s \in S, w \in W$ such that $P(s, w) > 0$, $P(m, a|s, w) > 0$. The SCM in the online setting where the agent can intervene on the variable $a$ as shown in the second column of Figure 1 can be defined as another four-tuple $(U, V, F, P_e)$, where the set of structural functions $F$ includes the state-reward transition distribution $P_1(s', r|s, w, m)$, the confounder transition distribution $P_2(w|s)$, the intermediate state transition distribution $P_3(m|s, a)$, and the policy $\pi(a|s)$. Note, that the policy $\pi(a|s)$ does not depend on $w$ since we assume that $w$ is unobserved to the agent.

3 Confounded Reinforcement Learning

![Diagram](image-url)

Figure 1: The left and right subfigures represent the SCM in the offline setting and online setting, respectively, which correspond to the CMDP for unobserved confounders. The behavior policy depends on $w$ in the offline setting, but not in the online setting.
unnecessary to utilize the offline dataset to train an agent that will be tested in the online environment. Specifically, we assume that the observation $o_1$ of a person/agent in the offline environment includes the state set $s$ and the confounder set $w$. This person interacts with the environment several times with some fixed strategy. Note that the data generation process in the offline environment conforms to the SCM in the offline setting. At the same time, there are other sensors that keep observing the process of interaction of this person/agent with the environment and gather it into an offline dataset. We assume that the observation $o_2$ of these sensors includes the state set $s$ and the intermediate state set $m$. The offline RL algorithms need to utilize the offline dataset to train an agent that will be tested in the online environment, where the observation $o_3$ includes only the state set $s$. Note that the online data generation process in the testing environment conforms to the SCM in the online setting. In summary, the confounders are unobserved in both the offline and the online data.

The above task setting leads to different dynamics $\hat{P}(s', r | s, a)$ and $P(s', r | s, a)$ in offline and online data (as proven in Appendix A), where $\hat{P}$ and $P$ denote the probability distributions corresponding to the SCMs in the offline and online settings respectively. Because the dynamics in the data determine the optimal value functions, the optimal value functions corresponding to offline data and online data are different. Therefore, if an agent is trained directly by the original deep RL algorithms using the offline data, the estimated optimal value function of this agent is optimal w.r.t. the offline data and suboptimal w.r.t. the online data. Thus, this agent will perform poorly in the online environment. We can also understand the above problem from another perspective, i.e., by analyzing the loss function of the original deep RL algorithms that satisfy Assumption 1.

**Assumption 1.** The loss function of the neural network (NN) of the deep RL algorithm only depends on the current state $s$, the action $a$, the next state $s'$, and the reward $r$.

The loss function of the original deep RL algorithms which satisfy Assumption 1 is shown as follows:

$$L_1 (\phi, D_{\text{off}}) \triangleq E_{s, a, s', r \sim D_{\text{off}}} [f_\phi + h_\phi]$$

$$= E_{s, a \sim D_{\text{off}}} \left[ E_{s', r \sim \hat{P}(.|s, a)} [f_\phi + h_\phi] \right],$$

where $D_{\text{off}}$ denotes all offline data collected through observation, $\phi$ denotes the NN parameters to be optimized, $f_\phi + h_\phi$ denotes the “loss” for a single training sample which differs between different NNs, $f_\phi$ denotes the part of the “loss” that depends on $(s, a, s', r)$, and $h_\phi$ denotes the part of the “loss” that depends only on $(s, a)$. Table 1 gives some examples of $f_\phi$ and $h_\phi$ for different NNs of the deep RL algorithms. It is obvious that the loss function in Equation 2 corresponds to the dynamics in the offline data, and thus the agent trained using this loss function performs poorly in the online environment.

To address this problem, based on importance sampling and do-calculus [Pearl, 2009; Pearl, 2012], we propose two deconfounding methods, namely, reweighting method and resampling method, which can both be combined with existing deep RL algorithms such as soft actor-critic (SAC) [Haarnoja et al., 2017], deep Q-learning (DQN) [Mnih et al., 2013], double deep Q-learning (DDQN) [van Hasselt et al., 2016] and conservative Q-learning (CQL) [Kumar et al., 2020], provided that a weak condition (i.e., Assumption 1) is satisfied. The deconfounding RL algorithms, i.e., the deep RL algorithms combined with the deconfounding methods, can estimate the optimal value function corresponding to the online data, and accordingly the agent trained by these deconfounding RL algorithms is expected to perform well in the online environment. In this section, we assume that the confounders are unobserved in the offline data. We also provide analysis of partially observed confounders in the offline data, and derive the reweighting and resampling methods in Appendix B accordingly.

### 3.1 Reweighting Method

As mentioned above, since the loss function $L_1 (\phi, D_{\text{off}})$ of the original deep RL algorithms corresponds to the dynamics in the offline data, the agent trained by the original deep RL algorithms performs poorly in the online environment. This inspires us to modify $L_1 (\phi, D_{\text{off}})$ to $L_2 (\phi, D_{\text{off}})$ as follows:

$$L_2 (\phi, D_{\text{off}}) \triangleq E_{s, a \sim D_{\text{off}}} \left[ E_{s', r \sim \hat{P}(.|s, a)} [f_\phi + h_\phi] \right].$$

Clearly, $L_2 (\phi, D_{\text{off}})$ corresponds to the dynamics in the online data. However we cannot directly estimate $L_2 (\phi, D_{\text{off}})$ in the form of Equation 3 from the offline data. Therefore, we transform $L_2 (\phi, D_{\text{off}})$ into a form that can be estimated from the offline data as follows:

**Proposition 1.** Under the definitions of the CMDP and SCMs in Section 2, it holds that

$$L_2 (\phi, D_{\text{off}}) \triangleq E_{s, a \sim D_{\text{off}}} \left[ E_{s', r \sim \hat{P}(.|s, a)} [f_\phi + h_\phi] \right]$$

$$= E_{s, a \sim D_{\text{off}}} \left[ E_{s', r, m \sim \hat{P}(.|s, a)} [d_1 (\tau) f_\phi + h_\phi] \right]$$

$$= E_{s, a, s', r, m \sim D_{\text{off}}} [d_1 (\tau) f_\phi + h_\phi],$$

where $\tau$ is a shorthand for the set of variables $(s, m, a, s', r)$ and $d_1 (\tau)$ is defined as follows:

$$d_1 (\tau) = \frac{\sum_{a'} \hat{P} (s', r|m, a', s) \hat{P} (a'|s)}{\hat{P} (s', r|m, a, s)}.$$

**Proof.** See Appendix A.
As shown in Equation 2 and Equation 4, the only difference between \( L_2 (\phi, D_{\text{off}}) \) in its new form and \( L_1 (\phi, D_{\text{off}}) \) is that \( f_o \) is multiplied by an extra weight \( d_1 (r) \). Therefore, we only need to estimate \( d_1 (r) \) and modify the loss function of the deep RL algorithms from \( L_1 (\phi, D_{\text{off}}) \) to \( L_2 (\phi, D_{\text{off}}) \). Note that \( d_1 (r) \) is composed of several conditional density functions. So, as long as these conditional density functions are estimated from the offline data, we can get an estimate of \( d_1 (r) \). To estimate these conditional density functions containing both discrete and continuous random variables, we adopt the LSCDE technique combined with a trick called adding jitter or jittering [Nagler, 2018] to add noises to all discrete random variables. It has been theoretically justified that this trick works well if all the noises are chosen from a specific class of noise distribution [Nagler, 2018]. The settings of choosing noises are given in Appendix C.

3.2 Resampling Method

The essence of the reweighting method is to adjust the impact of different samples on the loss function by reweighting the offline data. In other words, by incorporating weights into the loss function, we expand the impact of the samples that are less likely to occur in the offline data than in the online data, i.e., \( P(s', r | s, a) < P(s', r | s, a) \), and vice versa. However, we can also adjust the impact of different samples on the loss function through adjusting the probability that each sample occurs in the offline data. This idea inspires us to define a loss function as follows:

\[
L_3 (\phi, D_{\text{off}}) \triangleq E_{I \sim p_I} [f_{\phi,t} + h_{\phi,t}],
\]

where \( p_I (I = i) = d_{1,i} / \sum_{j=1}^N d_{1,j} \), \( d_{1,i} \) is a shorthand for \( d_1 (s_i, m_i, a_i, s'_i, r_i) \) where \( s_i, a_i, r_i, s'_i, m_i \) denotes the state, action, reward, next state and intermediate state in the offline dataset \( D_{\text{off}} \), respectively, \( f_{\phi,t} \) is a shorthand for \( f_\phi (s_t, a_t, s'_t, r_t) \) and \( h_{\phi,t} \) for \( h_\phi (s_t, a_t) \). In \( L_3 (\phi, D_{\text{off}}) \), we modify the probability that each sample occurs in the offline data according to the value of \( d_1 (r) \). By the definition of \( L_3 (\phi, D_{\text{off}}) \), based on the importance sampling technique and reparameterization trick, we can prove that the deep RL algorithms combined with the resampling method can also learn the optimal policy w.r.t. the online data from the offline data as follows:

Proposition 2. Under the definitions of the CMDP and SCMs in Section 2, the loss function of the resampling method is asymptotically equal to that of the reweighting method as in Equation 7 provided that the dataset is large enough.

\[
\lim_{N \to \infty} (L_3 (\phi, D_{\text{off}}) - L_2 (\phi, D_{\text{off}})) = 0
\]

Proof. See Appendix A.

According to Proposition 2, the resampling method is asymptotically equivalent to the reweighting method and can also deconfound the offline data provided that the offline dataset is large enough.

The advantages of the above two deconfounding methods are obvious. On the one hand, since both the reweighting and resampling methods decouple the deconfounding process from the RL algorithm, these methods can be easily combined with existing offline RL algorithms provided that the weak condition in Assumption 1 is satisfied. On the other hand, the implementation of these methods is quite straightforward, only requiring minor modification to the original RL algorithms, i.e., we only need to incorporate the estimated \( d_1 (r) \) into the loss function.

3.3 Simplification of the Causal Models

The convergence rate of conditional density estimation (CDE) decreases exponentially as the dimension of the variables increases due to the curse of dimensionality. Therefore, it is necessary to simplify the causal models by reducing the dimensions of the variables in the process of CDE, so that our deconfounding methods can be applied to problems with higher-dimensional variables. For example, if the confounder exists only between the action and reward, we can simplify the causal model correspondingly by ignoring the variable \( s' \) in the process of CDE. We formally define two types of simplified causal models for two specific cases and derive the corresponding simplified forms of \( d_1 (r) \) in Appendix D.

4 Results

Evaluation of the deconfounding method is a challenging issue due to the lack of benchmark tasks. In addition, there is little work in deep RL on learning from observational data with confounders. To this end, we design four benchmark tasks, namely, EmotionalPendulum*, WindyPendulum, EmotionalPendulum*, and WindyPendulum*, by modifying the Pendulum task in the OpenAI Gym [Brockman et al., 2016]. All the implementations of the offline RL algorithms in this paper follow d3rlpy, an offline RL library [Takuma Seno, 2021]. All the hyperparameters of the offline RL algorithms are set to the default values of d3rlpy. The rewards are tested over 20 episodes every 1000 learning steps, and averaged over 5 random seeds. Other hyperparameters and the implementation details are described in Appendix C.

4.1 Unobserved Confounders

![Causal graphs for EmotionalPendulum and WindyPendulum](image.png)

Figure 3: **Left**: the causal graphs depicting the offline and online data generating processes in EmotionalPendulum. **Right**: the causal graphs depicting the offline and online data generating processes in WindyPendulum. In both tasks, the confounders in the offline data are unobserved.
Figure 3 shows the graphical models for EmotionalPendulum and WindyPendulum, where the confounders are unobserved in the offline data. Obviously, these graphical models are special cases of the graphical models described in Section 2. In both tasks, we assume that there is an entertainment facility similar to a pendulum. One end of the pendulum is attached to a fixed point and a human sitting in a seat at the other end of the pendulum needs to swing the pendulum to an upright position. However, the controller of the pendulum fails intermittently, i.e., the action \(a \in \{-2, -1, 0, 1, 2\}\) that the human wants to take may not equal the actual action \(m \in \{-2, -1, 0, 1, 2\}\) executed by the machine. Specifically, in most cases, \(m\) is equal to \(a\), and there is a probability \(p_{\text{fail}}\) that \(m\) randomly selects an action in the action space, with each action equally likely to be selected. There are some sensors are used to collect the confounded offline data generated in the above human-environment interaction process. Note that the environmental information \(o_2\) in these offline data includes the state \(s\) and the actual action \(m\) executed by the machine. The offline RL algorithms need to use these confounded offline data to train an agent that substitutes the human to control the pendulum in the testing environment, where the confounders are unobserved, i.e., \(o_3 = \{s\}\). The main difference between the two tasks is that the confounders in EmotionalPendulum are between the action and reward while the confounders in WindyPendulum are between the action, reward and next state.

The design of EmotionalPendulum refers to the real-world phenomenon that humans may take some actions that are not rational when they are in negative emotions such as boredom and fear. Specifically, we assume that the human sitting in the entertainment facility may feel afraid and decide to slow down if the speed is too fast (i.e., if the speed \(|v|\) is above the threshold \(v_T\)), or feel boring and decide to speed up if the speed is too slow (i.e., if \(|v|\) \(\leq v_T\)), and will make rational decisions according to the trained agent if the emotions of the human are not negative, i.e., the behavior policy of the human depends on \(o_1 = \{s, w_1, w_2\}\), where \(w_1\) denotes whether the human has negative emotions and \(w_2\) denotes whether the expressions of the human are negative. Then, the environment will return a reward \(r = r_o + r_a\), where \(r_o\) denotes the original reward in the Pendulum task and \(r_a\) denotes an additional reward that is generated by the environment to encourage the human if the emotions are negative.

In WindyPendulum, we take into consideration the wind that may change the direction arbitrarily at each step. We assume that the wind force is 2.5 times larger than the largest force applied to the pendulum by the human/agent and the human will feel afraid if there is a strong wind. The human in a state of fear may choose the force opposite to the wind or decide to slow down, or choose the rational action, i.e., the behavior policy of the human depends on \(o_1 = \{s, w_1, w_2\}\), where \(w_1\) denotes whether the human is afraid due to the wind and \(w_2\) denotes the direction of the wind. See Appendix E for more details of the description of the two tasks. Similar to EmotionalPendulum, a reward \(r = r_o + r_a\) is then received, where the additional reward \(r_a\) is used to encourage the human if the wind exists.

As shown in Figure 4, we compare the deep RL algorithms DQN, DDQN, SAC and CQL with and without our deconfounding methods in EmotionalPendulum and WindyPendulum, respectively. DQN_RW denotes the DQN algorithm that uses the reweighting method and DQN_RS for the resampling method. Other algorithms are denoted similarly. It is clear that the deconfounding deep RL algorithms perform better than the original deep RL algorithms in both tasks.

We take EmotionalPendulum as an example to specifically illustrate the confounding problem and why our proposed methods could perform better than the original deep RL algorithms. As shown in the top left of Figure 3, the association flowing along the directed path \(a \rightarrow m \rightarrow r\) is causal association, while the association flowing along \(a \leftarrow w_1 \rightarrow w_2 \rightarrow r\) is non-causal association. In other words, because the human may take irrational actions when he/she is in negative emotions, and because the environment may return some additional reward if the emotions of the human are negative, there is non-causal association between the irrational action and the additional reward, which may mislead the human to believe that the irrational action will be well rewarded. Given \(s\), there is only causal association between \(a\) and \(r\) in the online data, because the non-causal association flowing along \(a \leftarrow s \rightarrow r\) is blocked by \(s\). In contrast, given \(s\), there are both causal association and non-causal association between \(a\) and \(r\) in the offline data. Therefore, the original deep RL algorithms trained using confounded offline data will be misled by non-causal association in the data, and thus perform poorly in the online environment. However, the deconfounding deep RL algorithms can learn causal association from confounded offline data, and thus perform well in the online environment.

So, what will happen if we enhance the non-causal association flowing along \(a \leftarrow w_1 \rightarrow w_2 \rightarrow r\) in the offline data? The values of the environmental hyperparameters of EmotionalPendulum, which correspond to the top row of Figure 4, are as follows: \(p_{\text{fail}} = 0.2\), \(odds_{s1} = 4\), \(v_T = 1\), and \(I_{p_1} = 0.7\), where \(I_{p_1}\) denotes the probability of the human choosing irrational actions when he/she has negative emotions and \(odds_{s1}\) denotes the odds that the human does not have negative emotions. Clearly, if we keep the other hyperparameters constant and increase the value of

| \(p_{\text{fail}}\) | \(odds_{s1}\) | \(v_T\) | \(I_{p_1}\) | BC | CQL | CQL_RW | CQL_RS |
|---|---|---|---|---|---|---|---|
| 0.1 | 0.5 | 0.7 | 0.1 | -1020.0 | -236.3 | 114.4 | 153.9 |
| 0.2 | 0.9 | -1145.9 | -552.5 | 59.1 | 100.6 |
| 0.5 | 1.0 | -1096.5 | -531.5 | 85.2 | 113.6 |
| 0.5 | 0.5 | -261.2 | 46.3 | 92.4 | 84.6 |
| 0.9 | -999.2 | -70.7 | 36.8 | 72.3 |
| 1.0 | 0.9 | -834.5 | -25.6 | 83.3 | 76.9 |
| 0.5 | 0.5 | -977.2 | -82.0 | 115.0 | 123.1 |
| 0.5 | 0.5 | -1126.7 | -523.1 | 97.2 | 44.5 |
| 0.9 | -999.2 | -70.7 | 36.8 | 72.3 |
| 1.0 | 0.9 | -1065.8 | -443.5 | 56.3 | 144.3 |
| 0.5 | 0.5 | -577.1 | 94.2 | 130.3 | 131.3 |
| 0.9 | -1022.0 | -10.0 | 81.6 | 115.3 |
| 1.0 | 0.9 | -899.5 | 80.4 | 110.3 | 112.1 |

Table 2: Comparison of BC, CQL, CQL_RW, and CQL_RS under different settings of EmotionalPendulum.
Moreover, CQL_RW and CQL_RS perform better than CQL under all settings in Table 2, which verifies the robustness of our deconfounding methods. In addition, as shown in Table 2, the behavior cloning (BC) algorithm performs much worse than CQL, CQL_RW and CQL_RS because the offline data are not expert data. In both EmotionalPendulum and WindyPendulum, the behavior policy of the human performs poorly, because the human takes many irrational actions based on his/her emotions. Therefore, imitation learning methods such as BC are not suitable for both tasks. In fact, in addition to Table 2, we compare BC, the original deep RL algorithms, and the deconfounding deep RL algorithms under different settings in Appendix E to verify the robustness of our deconfounding methods. Furthermore, we perform some ablation experiments in Appendix E.

4.2 Partially Observed Confounders

In the above two tasks, we assume that the confounders in the offline data are unobserved and identify the causal effect based on a variation of the frontdoor criterion. By contrast, in the two new tasks, EmotionalPendulum* and WindyPendulum*, we assume that the confounders in the offline data are partially observed and condition on a subset of the confounders to identify the causal effect. In the new tasks, the

\[ I_{p,1} \text{ from 0.7 to 0.9, the non-causal association flowing along } \alpha \leftarrow w^1 \rightarrow w^2 \rightarrow r \text{ in the offline data will be enhanced.} \]

As shown in Table 2, the enhancement of the non-causal association leads to increases in the gaps between the performance of CQL and CQL_RW, and between the performance of CQL and CQL_RS. Additionally, Table 2 shows that, no matter how much \( p_{fail}, odds_1, \) and \( v_T \) are, the gaps increase if we increase the value of \( I_{p,1} \) from 0.7 to 0.9. Similarly, if we keep the other hyperparameters constant, reducing the value of \( odds_1 \) leads to an enhancement of the non-causal association. As shown in Table 2, regardless of how much \( p_{fail}, v_T, \) and \( I_{p,1} \) are, the gaps increase if we reduce the value of \( odds_1 \) from 6 to 4.

Figure 4: Performance of the deep RL algorithms with and without our deconfounding methods in EmotionalPendulum and WindyPendulum.

Figure 5: Performance of the deep RL algorithms with and without our deconfounding methods in EmotionalPendulum* and WindyPendulum*.
controller of the pendulum never fails, i.e., the action $a$ that the human wants to take is the same as the actual action $m$ executed by the machine. The corresponding causal graphs and details of these two new tasks are given in Appendix E. To solve the new tasks, we define a new CMDP and two new SCMs and derive two new deconfounding methods under the new definitions in Appendix B. As shown in Figure 5, the deconfounding deep RL algorithms perform better than the original deep RL algorithms. Additionally, in Appendix E, we compare BC, the original deep RL algorithms, and the deconfounding deep RL algorithms under different settings.

5 Related Work

A number of studies [Lattimore et al., 2016; Nair and Jiang, 2021] seek to introduce causality into the field of RL in various settings, where the use of observational data [Wang et al., 2021; Lu et al., 2018; Gasse et al., 2021] is one of the key issues.

The Bandit Problem. The Causal bandit problem [Lattimore et al., 2016] seeks to learn the optimal intervention from the interventional data, which conform to a causal graph, to minimise a simple regret. Sen et al. [2017] propose the successive rejects algorithms and derive the gap-dependent bounds for causal bandit based on importance sampling. The Contextual bandit problem [Langford and Zhang, 2007] seeks to learn a policy which depends on the context, namely the environmental variables, to maximize the reward which depends on the action and the context. Kallus and Zhou [2018] propose algorithms for OPE and learning from the confounded observational data with continuous actions based on the inverse probability weighting. Our paper, however, studies the causal RL problem which is more difficult than the bandit problem with a longer horizon.

Causal RL. There is some work studying causal RL in the model-based RL settings. Lu et al. [2018] propose a model-based RL method that estimates an SCM from observational data with the time-invariant confounder between the action and reward based on the latent-variable model proposed by Louizos et al. [2017]. This model, however, cannot estimate the correct causal effect provided that the latent variable is misspecified or the data distribution is complex as shown by Rissanen and Marttinen [2021]. Gasse et al. [2021] combine interventional data with observational data to estimate the latent-based transition model, which can be used for model-based RL. However, since that the time consumed by their program increases rapidly as the size of the discrete latent space increases, and that they assume that the state space is smaller than the discrete latent space, their algorithm can only be used to address problems with small state spaces. On the contrary, our deconfounding methods can be applied to problems with both continuous and discrete variables and our assumptions do not restrict the size of the state and confounder spaces. The closest work is the one by Wang et al. [2021], where the authors focus on how to improve the sample efficiency of the online algorithm by incorporating large amounts of observational data in the model-free RL settings. However, they follow the assumption of Yang and Wang [2019; 2020]; Jin et al. [2020] that the transition kernels and reward functions are linear, so that the corresponding SCMs are also linear [Peters et al., 2017]. This strong assumption is hard to hold in the real world, where the dynamics of the environment are nonlinear. Furthermore, they assume that there is very complex correlation between the backdoor-adjusted or frontdoor-adjusted feature and another feature. For example, they assume that the backdoor-adjusted feature is the expectation of the state-action-confounder feature. These assumptions make their theoretical algorithms difficult to implement. In contrast, the transition kernels and reward functions in our assumptions can be nonlinear, and experiments are provided to verify the performance of the offline RL algorithms combined with our deconfounding methods.

Off-Policy Evaluation. A line of work in the OPE field uses confounded data to estimate the performance of the evaluation policy. There are several papers that design estimators for the partially observable Markov decision process in the tabular setting [Nair and Jiang, 2021; Tennenholtz et al., 2020], which cannot be applied to the large or continuous state space. Recently a lot of work [Kallus and Zhou, 2020; Gelada and Bellemare, 2019; Hallak and Mannor, 2017; Kallus and Uehara, 2019; Liu et al., 2018] proposes algorithms for OPE based on the importance sampling technique. However, these algorithms estimate the stationary distribution density ratio, which changes as the evaluation policy changes. This makes their work inapplicable to the RL field, where the evaluation policy is always changing. Instead, since we realize that SAC, DQN, and other off-policy RL algorithms utilize only the single-step transition in the data, our deconfounding methods are based on the conditional distribution density ratio, which remains constant in the learning process, so that our work can be applied to the RL field.

6 Conclusion

In this paper, we propose two plug-in deconfounding methods based on the importance sampling and causal inference techniques for model-free deep RL algorithms. These two methods can be applied to large and continuous environments. In addition, we prove that our deconfounding methods can construct an unbiased loss function w.r.t the online data and show that the deconfounding deep RL algorithms perform better than the original deep RL algorithms in the four benchmark tasks which are created by modifying the OpenAI Gym. A limitation of our work is the assumption that we already know the causal graph of the environment. In future work, we plan to build a causal model through causal discovery without prior knowledge of the environment and to deconfound the confounded data based on the discovered causal model. We hope that our deconfounding methods can bridge the gap between offline RL algorithms and real-world problems by leveraging large amounts of observational data.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (No. 2021ZD0112400), the NSFC-Liaoning Province United Foundation under Grant U1908214, the National Natural Science Foundation of China.
(No. 62076259), the 111 Project (No. D23006), the Fundamental Research Funds for the Central Universities under grant DUT21TD107, DUT22ZD214, the LiaoNing Revitalization Talents Program (No. XLYC2008017), the Fundamental and Applicational Research Funds of Guangdong province (No. 2023A151012946), and the Fundamental Research Funds for the Central Universities-Sun Yat-sen University. The authors would like to thank Zifan Wu for revising the author response to the reviewers.

References

[Agarwal et al., 2020] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In International Conference on Machine Learning, pages 104–114. PMLR, 2020.

[Brockman et al., 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. arXiv preprint arXiv:1606.01540, 2016.

[Gelada and Bellemare, 2019] Carles Gelada and Marc G Bellemare. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23), pages 3647–3655, 2019.

[Haarnoja et al., 2017] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In International Conference on Machine Learning, pages 1352–1361. PMLR, 2017.

[Hallak and Mannor, 2017] Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. In International Conference on Machine Learning, pages 1372–1383. PMLR, 2017.

[Jaques et al., 2019] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way offline policy batch deep reinforcement learning of implicit human preferences in dialog. arXiv preprint arXiv:1907.00456, 2019.

[Jin et al., 2020] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In Conference on Learning Theory, pages 2137–2143. PMLR, 2020.

[Kallus and Uehara, 2019] Nathan Kallus and Masatoshi Uehara. Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. arXiv preprint arXiv:1909.05850, 2019.

[Kallus and Zhou, 2018] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In International conference on artificial intelligence and statistics, pages 1243–1251. PMLR, 2018.

[Kallus and Zhou, 2020] Nathan Kallus and Angela Zhou. Confounding-robust policy evaluation in infinite-horizon reinforcement learning. arXiv preprint arXiv:2002.04518, 2020.

[Kumar et al., 2019] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. arXiv preprint arXiv:1906.00949, 2019.

[Kumar et al., 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. arXiv preprint arXiv:2006.04779, 2020.

[Langford and Zhang, 2007] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems, volume 20. Curran Associates, Inc., 2007.

[Lattimore et al., 2016] Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. arXiv preprint arXiv:1606.03203, 2016.

[Levine et al., 2020] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643, 2020.

[Liu et al., 2018] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. arXiv preprint arXiv:1810.12429, 2018.

[Louizos et al., 2017] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[Lu et al., 2018] Chaochao Lu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Deconfounding reinforcement learning in observational settings. arXiv preprint arXiv:1812.10576, 2018.

[Mnih et al., 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.
[Nagler, 2018] Thomas Nagler. A generic approach to non-parametric function estimation with mixed data. *Statistics & Probability Letters*, 137:326–330, 2018.

[Nair and Jiang, 2021] Yash Nair and Nan Jiang. A spectral approach to off-policy evaluation for pomdps. *arXiv preprint arXiv:2109.10502*, 2021.

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

[Pearl, 2012] Judea Pearl. The do-calculus revisited. *arXiv preprint arXiv:1210.4852*, 2012.

[Peters et al., 2017] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[Rissanen and Marttinen, 2021] Severi Rissanen and Pekka Marttinen. A critical look at the consistency of causal estimation with deep latent variable models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4207–4217. Curran Associates, Inc., 2021.

[Rothfuss et al., 2019] Jonas Rothfuss, Fabio Ferreira, Simon Walther, and Maxim Ulrich. Conditional density estimation with neural networks: Best practices and benchmarks. *arXiv:1903.00954*, 2019.

[Sen et al., 2017] Rajat Sen, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Identifying best interventions through online importance sampling. In *International Conference on Machine Learning*, pages 3057–3066. PMLR, 2017.

[Siegel et al., 2020] Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

[Sugiyama et al., 2010] Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 781–788, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Takuma Seno, 2021] Michita Imai Takuma Seno. d3rlpy: An offline deep reinforcement library. In *NeurIPS 2021 Offline Reinforcement Learning Workshop*, December 2021.

[Tennenholtz et al., 2020] Guy Tennenholtz, Uri Shalit, and Shie Mannor. Off-policy evaluation in partially observable environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10276–10283, 2020.

[van Hasselt et al., 2016] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press, 2016.

[Wang et al., 2021] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21164–21175. Curran Associates, Inc., 2021.

[Wang and Wang, 2019] Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

[Wang and Wang, 2020] Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.