Confounding Robust Continuous Control via Automatic Reward Shaping

Anonymous Author(s)
Submission Id: 222

ABSTRACT

Reward shaping has been applied widely to accelerate Reinforcement Learning (RL) agents' training. However, a principled way of designing effective reward shaping functions, especially for complex continuous control problems, remains largely under-explained. In this work, we propose to automatically learn a reward shaping function for continuous control problems from offline datasets, potentially contaminated by unobserved confounding. Specifically, our method builds upon the recently proposed causal Bellman equation to learn a tight upper bound on the optimal state values, which is then used as the potentials in the Potential-Based Reward Shaping (PBRS) framework. Our proposed reward shaping algorithm is tested with Soft-Actor-Critic (SAC) on multiple commonly used continuous control benchmarks and exhibits strong performance guarantees under unobserved confounders. More broadly, our work marks a solid first step towards confounding robust continuous control from a causal perspective.

CCS CONCEPTS

• **Theory of computation → Sequential decision making**.

KEYWORDS

Causal Inference, Reinforcement Learning, Unobserved Confounder

ACM Reference Format:
Anonymous Author(s). 2026. Confounding Robust Continuous Control via Automatic Reward Shaping. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 17 pages.

1 INTRODUCTION

Reinforcement learning (RL) has demonstrated impressive success in continuous control domains such as robotic manipulation, locomotion, and autonomous systems [24, 39, 40]. Despite this progress, learning effective policies in high-dimensional, complex environments remains challenging due to sample inefficiency and high sensitivity to reward design. When the original task reward is not efficient to learn from, reward shaping can significantly accelerate learning by injecting informative signals that guide exploration and policy improvement. However, designing effective shaping functions remains a persistent challenge, often requiring substantial domain expertise and manual effort to ensure they are helpful without hurting the performance [9, 30, 32, 37].

Potential-Based Reward Shaping (PBRS) [30] offers a principled framework for injecting additional reward signals while preserving the original task's optimal policy. However, the effectiveness of PBRS critically depends on the quality of the potential function used. Recent work has explored learning state potentials automatically from offline data [3, 21, 55], but such approaches typically assume no unobserved confounders (NUC) [27, 28], the access to fully observed (unconfounded) trajectories. Such an assumption can easily break down in many real-world settings. Unobserved confounders can arise from human demonstrations, legacy systems, or sensor capability differences in robotic platforms. When the NUC assumption is violated, the effects of candidate policies become generally unidentifiable. That is, the given model assumptions are insufficient to uniquely recover the value function from offline data, regardless of the sample size [33, 51]. As a result, standard RL methods relying implicitly on NUC can suffer from degraded learning performance in such settings. More recently, Li et al. [18] propose to use partial identification approach to learn confounding robust shaping functions. But their proposed method is limited to discrete and lower dimensional settings. A practical solution for continuous and higher dimensional environments is yet to be discussed.

In this work, we tackle the problem of automatic reward shaping in continuous control settings where offline data may be confounded. Specifically, we utilize offline trajectories collected from unknown, potentially biased behavioral policies to estimate causal upper bounds on the optimal interventional state values. These bounds are then employed as potential functions within the PBRS framework to construct shaping rewards. By incorporating these shaped rewards, we enable model-free RL agents to perform more informed exploration and policy learning, even in the presence of unobserved confounders. We empirically evaluate our framework in confounded MuJoCo environments with partial observability. Experiments on a suite of confounded continuous environments show that our method consistently outperforms unshaped and causally unaware shaping baselines (CQL-Shaping, [14]). These results highlight the robustness and practical effectiveness of our approach in real-world confounded RL settings.

Our main contributions are as follows:

- We derive a Causal Bellman Equation for the stationary infinite-horizon Confounded Markov Decision Process (CMDP), which converges to a tight upper bound of the optimal state values;
- We design a neural learning algorithm that approximates the Causal Bellman Equation in high-dimensional continuous state-action CMDPs;
- We use the learned state value bound as the reward shaping function and empirically demonstrate the superior performance boost when applying to Soft-Actor-Critic [5] in various challenging confounded continuous control environments.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

2 BACKGROUND

Confounding Robust Decision-making. We will focus on the sequential decision-making setting in an infinite horizon stationary Markov Decision Process (MDP, Puterman [35]) where the agent intervenes on a sequence of actions $X_1, \dots$ to optimize the cumulative return over reward signals $Y_1, \dots$ given state observations $S_1, S_2, \dots$ at each corresponding time step. Standard MDP formalism focuses on the perspective of the learners who could actively intervene in the environment. Consequently, the data collected from randomized experiments is free from the contamination of unobserved confounding bias and is generally assumed away in the model. However, when considering offline data collected by passive observation [14, 16, 20] where the learner may not necessarily have deliberate control over the behavioral policy generating the data, or when the state attributes are partially observed [10, 12, 45], unobserved confounding arises. Consequently, this could lead to biased estimation and safety/alignment issue in various reinforcement learning tasks, including off-policy learning [13, 18–20, 53], curriculum learning [17] and imitation learning [15, 54].

Continuous Control with Deep Reinforcement Learning. In continuous action space, the sample efficiency problem is exacerbated rendering commonly used on-policy learning solutions unfavorable, such as TRPO [38], PPO [40] or A3C [23]. At each time step, on-policy algorithms collect new trajectories from the environment only for updating the agents by a single gradient step. As task complexity grows, this procedure becomes increasingly expensive. Off-policy algorithms, on the other hand, reuse past experiences. The direct application of this idea is DQN and its variants [24]. For continuous policy learning, actor-critic based method is preferred for its stability and easy-to-tune hyper-parameters [5, 39]. In this work, we use Soft-Actor-Critic (SAC), a maximum entropy reinforcement learning framework which improves upon the traditional maximum reward framework with substantially better exploration and robustness [4, 56], as our base learning algorithm.

Potential-based reward shaping (PBRS). Reward shaping is a popular line of techniques for incorporating domain knowledge during policy learning. Common approaches such as Potential-Based Reward Shaping (PBRS, Ng et al. [30]) add supplemental signals to the reward function so that it would be easier to learn in future downstream tasks without affecting the optimality of the learned policy. PBRS modifies the reward function in the system by adding the discounted next state potential subtracted by current state potential. This encourages the learning agent to visit states with higher potentials while avoiding visiting states with low potentials. More importantly, optimal policies remain invariant across this shaping process, i.e., every optimal policy learned in the MDP under PBRS is guaranteed to be optimal in the original MDP, and vice versa.

Notations. We will consistently use capital letters (V) to denote random variables, lowercase letters (v) for their values, and cursive $\mathcal{V}$ to denote their domains. Fix indices $i, j \in \mathbb{N}$. We use bold capital letters ($\boldsymbol{V}$) to denote a set of random variables and let $|\boldsymbol{V}|$ denote its cardinality of the set $\boldsymbol{V}$. Finally, $\mathbf{1}_{Z=z}$ is an indicator function that returns 1 if event $Z = z$ holds true; otherwise, it returns 0.

3 THE CHALLENGE OF CONFOUNDED CONTINUOUS CONTROL

In real-world continuous control tasks, agents may have limited sensor information due to cost or environment constraints when deployed in online operation. This leads to poor sample efficiency if the agent is to be tuned fully online. Pre-training with offline data is a common remedy. But people generally expect the behavioral policy generating the offline datasets to have the same sensor capability as the online agent [14, 16, 29]. However, offline data might be collected by agents in a more controlled environment with privileged sensor capability [6, 7, 25]. This mismatch induces unobserved confounding at the decision level: offline decisions reflect privileged information unavailable to the online learning agent. As a result, directly learning from such data without addressing confounding can lead to biased policies and poor online performance. Tackling confounded continuous control is thus critical for reliable and sample-efficient RL under realistic sensory constraints.

In this paper, we consider an extended family of MDPs that explicitly models the presence of unobserved confounders when generating offline data.

Definition 3.1. A Confounded Markov Decision Process (CMDP) $\mathcal{M}$ is a tuple of $\langle \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathbb{F}, \mathbb{P} \rangle$ where,

- $\mathcal{S}, \mathcal{X}, \mathcal{Y}$ are, respectively, the space of observed states, actions, and rewards;
- $\mathcal{U}$ is the space of unobserved exogenous noise;
- $\mathbb{F}$ is a set consisting of the transition function $\tau : \mathcal{S} \times \mathcal{X} \times \mathcal{U} \mapsto \mathcal{S}$, behavioral policy $\beta : \mathcal{S} \times \mathcal{U} \mapsto \mathcal{X}$, and reward function $r : \mathcal{S} \times \mathcal{X} \times \mathcal{U} \mapsto \mathcal{Y}$;
- $\mathbb{P}$ is a set of distributions P over the unobserved domain $\mathcal{U}$.

To model general continuous control problems, we assume the space of states, $\mathcal{S}$, actions, $\mathcal{X}$, and unobserved exogenous noise, $\mathcal{U}$, to be multi-dimensional and continuous throughout the paper. Consider a demonstrator agent interacting with a CMDP. For every time step $h = 1, \dots,$ the nature draws an exogenous noise U_h from the distribution $P(\mathcal{U})$; the demonstrator performs an action $X_h \leftarrow f_X(S_h, U_h)$, receives a subsequent reward $Y_h \leftarrow f_Y(S_h, X_h, U_h)$, and moves to the next state $S_{h+1} \leftarrow f_S(S_h, X_h, U_h)$. The observed trajectories of the demonstrator (from the learner's perspective) are thus summarized as the observational distribution $P(\bar{\boldsymbol{X}}, \bar{\boldsymbol{S}}, \bar{\boldsymbol{Y}})$. [1] In the data-generating process described above, for every time step h, the exogenous noise U_h becomes an unobserved confounder affecting the action X_h, reward Y_h, and next state S_{h+1} simultaneously. Therefore, CMDP is also referred to MDP with Unobserved Confounders (MDPUC, [52]) and is a subclass of Confounded Partially Observed MDP [2, 22, 41] where Markov property holds.

The observed distribution of such an offline collected dataset with finite trajectories up to H steps can be written as,

$$P(\bar{\boldsymbol{X}}_{1:H}, \bar{\boldsymbol{S}}_{1:H}, \bar{\boldsymbol{Y}}_{1:H}) = P(s_1) \prod_{h=1}^{H} \left(\int_{\mathcal{U}} \mathbf{1}_{s_{h+1}=f_S(s_h,x_h,u_h)} \mathbf{1}_{x_h=f_X(s_h,u_h)} \mathbf{1}_{y_h=f_Y(s_h,x_h,u_h)} P(u_h) du_h \right) \tag{1}$$

[1] We will consistently use $\bar{\boldsymbol{X}}, \bar{\boldsymbol{S}}, \bar{\boldsymbol{Y}}$ to represent trajectory sequences.

(a) CMDP - Offline Data Collection, $\mathcal{M}$ **(b) CMDP - Online Learning, $\mathcal{M}_\pi$**

Figure 1: (a) Causal diagram of the CMDP modeling the offline data generating process; (b) Causal diagram of the CMDP modeling the online learning process under policy $\text{do}(\pi)$**.**

The causal diagram in Fig. 1a showcases how the confounders are affecting state transitions, reward, and the behavioral agent's policy while the online learning agent doesn't have such information when acting in the environment Fig. 1b. By convention [33], we use bi-directed arrows (e.g., $X_h \leftrightarrow S_h$) to indicate the presence of unobserved confounders, U_h, affecting actions, states and rewards.

During the online learning phase, as shown in the causal diagram in Fig. 1b, the agent intervenes on the action variable following a policy $\pi(x_h|s_h)$ that maps from state to a distribution over the action domain $\mathcal{X}$. This is denoted as policy intervention $\text{do}(\pi)$ replacing the behavioral policy f_X during the offline data collection phase. The online trajectory distribution in CMDP under π, $\mathcal{M}_\pi$ is,

$$P_\pi(\bar{\boldsymbol{X}}_{1:H}, \bar{\boldsymbol{S}}_{1:H}, \bar{\boldsymbol{Y}}_{1:H}) = P(s_1)\prod_{h=1}^{H}\left(\pi(x_h|s_h)\mathcal{T}(s_h, x_h, s_{h+1})\mathcal{R}(s_h, x_h, y_h)\right) \tag{2}$$

where the transition distribution $\mathcal{T}$ and the reward distribution $\mathcal{R}$ are given by,

$$\mathcal{T}(s_h, x_h, s_{h+1}) = \int_{\mathcal{U}} \mathbf{1}_{s_{h+1}=f_S(s_h,x_h,u_h)} P(u_h)du_h \tag{3}$$

$$\mathcal{R}(s_h, x_h, y_h) = \int_{\mathcal{U}} \mathbf{1}_{y_h=f_Y(s_h,x_h,u_h)} P(u_h)du_h \tag{4}$$

In CMDP, the learning goal is still to find the optimal policy π^* that maximizes the cumulative return (often discounted by $\gamma \in (0, 1)$). That is, $\pi^* = \arg\max_\pi \mathbb{E}[\sum_{h=1}^{\infty} \gamma^{h-1} y_h]$. This objective function can be solved iteratively using the Bellman Optimality Equation [35],

$$Q^*(s, x) = \mathbb{E}[Y_h + \gamma \max_{x'} Q^*(S_{h+1}, x')|S_h = s, X_h = x] \tag{5}$$

where $Q^*(s, x) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} y_{h+t}|S_h = s, X_h = x]$ by definition is the optimal state action value function denoting the best return after taking action x in state s.

In the offline to online learning setting, one would envision that learning from offline datasets generated by a competitive policy with a good action space coverage should yield near-optimal online policies [14, 29]. This is indeed true under the MDP definition where there are no unobserved confounders. The state transitions and reward functions can be easily identified by the observation distribution of the offline dataset,

$$\mathcal{T}(s_h, x_h, s_{h+1}) = P(s_{h+1}|s_h, x_h) \tag{6}$$

$$\mathcal{R}(s_h, x_h, y_h) = P(y_h|s_h, x_h). \tag{7}$$

When the above identification formula hold, several off-policy algorithms have been proposed to estimate the effect of candidate policies from finite observations [11, 14, 26, 29, 34, 44, 48, 49]. Together with deep learning, these methods could be further extended to complex domains [24, 31, 39, 40, 42]. However, NUC could be fragile in practice and does not necessarily hold due to violations like sensory mismatch in data generation process. In these situations, applying standard off-policy methods may fail to converge to optimal, despite using powerful deep learning models. [2]

To witness the challenge of confounded continuous control, we instantiate a confounded variant of Hopper [47] where the offline agent has access to the full state observation, a total of 11-dimension vectors while the online agent can observe all of the dimension except the angle of it thigh joint (comparable to the knee in humans). Not observing the position of its thigh joint presents a genuine challenge to the agent given it would be unable to know to how move its thigh joint to generate the power to hop. As shown in Fig. 2 the vanilla SAC [5] agent trained under partial observation cannot recover 1/2 of the performance of the agent trained with complete state observations.

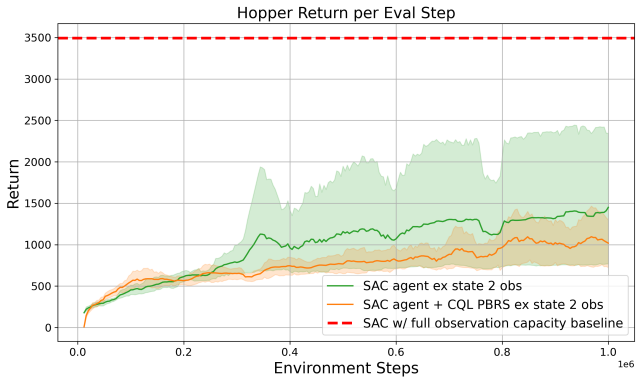


Figure 2: Performance of Hopper SAC Agent with full capacity and SAC agent unable to observe state 2.

These results expose a critical barrier to reliable learning under restricted sensory capabilities: without access to key hidden factors, even state-of-the-art off-policy RL methods will suffer. And adding

[2] NUC is orthogonal to the issue of action space coverage discussed by the line of research on conservative Q-learning (CQL) [14]. As shown in Fig. 2, CQL learned state values cannot handle the unobserved confounding issue in offline datasets.

good quality offline datasets which supposedly contain valuable information on the optimal state values does not help. As demonstrated in Fig. 2, commonly used algorithms also fall short under the confounded setting due to their fundamental limitation of assuming NUC implicitly followed from the (PO)MDP definition. Then, our central research question is: Can we extract and transfer such contaminated offline knowledge to guide online learning? The answer is yes. In the next section, we propose the causal bellman equation from which automatic reward shaping with confounded offline data is made possible to facilitate online learning.

4 CONFOUNDED CONTINUOUS CONTROL WITH SHAPED REWARDS

In this section, we will introduce how we learn an optimistic state potential from confounded continuous offline data, which is then used for online fine-tuning. See Sec. 7 in the appendix for proof details of theorems discussed in this section.

4.1 Learn Optimistic State Potentials via Confounding Robust Offline Pretraining

It is well acknowledged that a good state potential function is the optimal state value [30]. But without training the agent, one cannot have easy access to the optimal state values. The premise of using state values estimated from offline dataset as state potentials is that such values are close to optimal state values. However, without deliberate control on the quality of the behavioral policy nor the NUC condition, such offline learned values could be heavily biased and cannot be used for reward shaping as we have seen in Fig. 2. Recently Li et al. [18] proposes to use partial identification methods to upper bound the optimal state values for finite horizon non-stationary CMDPs from such confounded offline datasets robustly. Here we extend the results to stationary infinite-horizon CMDPs.

Theorem 4.1 (Causal Bellman Optimal Equation for Stationary Infinite-Horizon CMDPs)**.** *For a CMDP environment $\mathcal{M}$ with reward $Y_h \leq b, b \in \mathbb{R}$, the optimal value of interventional policies, $V^*(\mathbf{s}), \forall \mathbf{s} \in \mathcal{S}$, is upper bounded by $V^*(s) \leq \overline{V}(s)$ satisfying the Causal Bellman Optimality Equation,*

$$\overline{V}(s) = \max_x \Bigg[P(x|s) \left(\widetilde{\mathcal{R}}(s,x) + \gamma \mathbb{E}_{\widetilde{\mathcal{T}}}[\overline{V}(s')] \right) + P(\neg x|s) \left(b + \gamma \max_{s'} \overline{V}(s') \right) \Bigg] \tag{8}$$

where $\widetilde{\mathcal{R}}$ is offline estimated reward distribution and $\widetilde{\mathcal{T}}$ is the estimated transition distribution.

Compared with the original Bellman Optimal Equation, the Causal Bellman Equation accounts for the uncertainty brought by confounders in the offline dataset via an extra term,

$$\left(b + \gamma \max_{s'} \overline{V}_{h+1}(s') \right) \tag{9}$$

which represents the best return that the agent could have achieved from those "unselected" actions, i.e., $P(\neg x|s)$. With the Causal Bellman Optimal Equation, we can robustly upper bound the optimal state values from a confounded offline dataset generated by CMDP $\mathcal{M}$. Next we show that the extended Causal Bellman Optimal Equation converges to a unique fixed point, which is a valid upper bound on the optimal interventional state values for online agents (Fig. 1b).

Theorem 4.2 (Convergence of Causal Bellman Optimal Equation)**.** *The Causal Bellman Optimality Equation converges to a unique fixed point, which is also an upper bound on the optimal interventional state values under the assumption that $P(s,x) > 0, \forall s, x$ in the stationary infinite horizon CMDP $\mathcal{M}$.*

The proposed Causal Bellman Equation does not translate directly into a practical algorithm for high dimensional continuous control problems with function approximators, though. The propensity score of actions $P(x|s), P(\neg x|s)$ is ill-defined as any single point in a continuous distribution has zero probability. Furthermore, naively enumerating all the states to calculate $\max_{s'} \overline{V}(s')$ or enumerating actions to get values from Q-values is intractable in continuous state and action space. Thus, we make several practical approximations when implementing the Causal Bellman Equation.

Firstly, we parametrize the causal upper bound state potential to be $V_{\theta_1}(\cdot)$ and its corresponding target network $V_{\theta_1'}(\cdot)$ for smoother learning updates [24]. We also replace the outside $\max_x$ in Eq. (8) with an expectation over the observed action distribution in the offline dataset under the condition that the behavioral policy is competitive. Then, we restrict the observed policy distribution to be within a tractable class of Gaussian policies as P_{θ_2}. To train this policy distribution, we maximize the likelihood of the observed actions given states in the offline dataset $\mathcal{D}$,

$$J(\theta_2) = \mathbb{E}_{\mathcal{D}} \Big[\log P_{\theta_2}(x|s) \Big] \tag{10}$$

When training V_{θ_1}, we apply the reparametrization trick to make sampling from P_{θ_2} differentiable,

$$x = f_{\theta_2}(\epsilon, s) \tag{11}$$

where ϵ is an input noise vector sampled from a fixed standard Gaussian distribution. Lastly, instead of calculating the global best possible next state $\max_{s'} \overline{V}(s')$, we aim at finding the best possible nearby states if an action $x' \neq x$ had been taken. Thus, we choose to model the difference distribution between the current state s and next state s' given state action pair (s,x). Similarly, we restrict this distribution to be Gaussian, P_{θ_3}, and the training objective is to maximize the log likelihood of observed state differences in $\mathcal{D}$,

$$J(\theta_3) = \mathbb{E}_{\mathcal{D}} \Big[\log P_{\theta_3}(\Delta_s \mid s, x) \Big] \tag{12}$$

where Δ_s is the state difference between s and s'. And we apply the reparametrization trick again for differentiable sampling,

$$\Delta_s = f_{\theta_3}(\epsilon, s, x). \tag{13}$$

Now we can approximate the Causal Bellman Equation backup, $\mathcal{B}$, with parametrized neural network components as,

$$\hat{\mathcal{B}}\overline{V}_{\theta_1}(s) = \mathbb{E}_{\mathcal{D}} \Bigg[P_{\theta_2}(x|s) \left(y + \gamma \overline{V}_{\theta_1}(s') \right) + P_{\theta_2}(x'|s) \Big(\max_{y \in \mathcal{D}} y + \gamma \overline{V}_{\theta_1}(s + \Delta_s) \Big) \Bigg] \tag{14}$$

where $x' = \arg\max_{x'' \sim P_{\theta_2}(\cdot|s), x'' \neq x} V_{\theta_1}(s + \Delta_s'), \Delta_s' \sim P_{\theta_3}(\cdot|s, x')$ is the action that maximize the return of "the road not taken", and $\Delta_s \sim P_{\theta_3}(\cdot|s, x')$ is the state difference sampled given s, x'. The extra

Algorithm 1 Neural Causal Upper Bound State Potential

Initialize parameters $\theta_1, \theta_1', \theta_2, \theta_3$
while Not Converged **do**
  Sample an offline batch $\{(s_i, x_i, s_i', y_i)\}_{i=1}^{B}$
  Update observed policy, $\theta_2 \leftarrow \theta_2 + \lambda_2 \hat{\nabla} J(\theta_2)$ (Eq. (10))
  Update state difference, $\theta_3 \leftarrow \theta_3 + \lambda_3 \hat{\nabla} J(\theta_3)$ (Eq. (12))
end while
while Not Converged **do**
  Sample an offline batch $\{(s_i, x_i, s_i', y_i)\}_{i=1}^{B}$
  Update state potential, $\theta_1 \leftarrow \theta_1 - \lambda_1 \hat{\nabla} J(\theta_1)$ (Eq. (15))
  Update target networks, $\theta_1' \leftarrow \tau\theta_1 + (1-\tau)\theta_1'$
end while
return $V_{\theta_1}(s)$

compensation term in Eq. (9) is approximated by the maximum reward observed in the offline dataset plus the discounted best next state value as if the agent had taken x' and transited to a state s' that is Δ_s away from s. In implementation, we also approximate the action propensity score with its corresponding Gaussian density to avoid the zero probability issue. The state potential function $\overline{V}_{\theta_1}$ is then trained to minimize the squared residual error,

$$J(\theta_1) = \mathbb{E}_{\mathcal{D}}\left[\frac{1}{2}\left(\overline{V}_{\theta_1}(s) - \hat{\mathcal{B}}\overline{V}_{\theta_1'}(s)\right)^2\right] \tag{15}$$

See Algo. 1 for the full pseudo-code of learning the state potentials from continuous confounded offline data. In the next section, we will illustrate how to use this learned state potentials as reward shaping functions during online training with SAC.

4.2 Online Fine-tuning with Reward Shaping

We first re-establish the optimal policy invariance property of PBRS [30] in infinite-horizon stationary CMDPs.

Proposition 4.3. *For a CMDP under policy π, $\mathcal{M}_\pi$, let $\mathcal{M}_\pi'$ be the CMDP obtained from $\mathcal{M}_\pi$ by replacing the reward with the following function, for every time step $h = 1, \ldots, H$,*

$$y_h' := y_h + \gamma\phi_h(S_{h+1}) - \phi_h(S_h), \tag{16}$$

where y_h is the original reward returned by $\mathcal{M}_\pi$; $\phi_h(\cdot) : \mathcal{S} \mapsto \mathbb{R}$ is a real valued potential function. Then every optimal policy in $\mathcal{M}_\pi$ will also be an optimal policy in $\mathcal{M}_\pi'$, and vice versa.

This guarantees that the agent could still obtain the same optimal policy as the agent without using reward shaping.

For the learning algorithm, we choose to use a popular value-based maximum entropy learner, SAC [5], for its amenity to PBRS. Because for policy gradient based methods like PPO [40], multi-step returns are commonly adopted when calculating advantages [39] resulting in the intermediate shaping terms being canceled out without affecting learning process.

The original SAC algorithm learns a soft policy that minimizes the following objective function:

$$J(\phi) = \mathbb{E}_{\mathcal{D}}\left[\mathrm{D}_{\mathrm{KL}}\left(\pi_\phi(\cdot|s)\middle\|\frac{\exp(Q_\theta(s,\cdot))}{Z_\theta(s)}\right)\right] \tag{17}$$

where the parametrized policy π_ϕ is updated towards the exponential of the learned Q function Q_θ and $Z_\theta(s)$ normalizes the distribution. With the reward shaping function defined in Eq. (16), the Q value function Q_θ is now trained to track returns with the shaped reward y' instead of y while other parts of SAC stay unchanged.

5 EXPERIMENTS

In this section, we evaluate the efficacy of the state value upper bounds learned in Algo. 1 by applying them as potentials in Potential-Based Reward Shaping (PBRS, [30]) to augment SAC agents' training rewards in a suite of confounded, continuous control environments. We compare our results to 1) a baseline online SAC agent that does not use PBRS and 2) CQL PBRS [14] that uses offline learned conservative Q-values as shaping potentials in PBRS. We provide further commentary on which cases the causal reward shaping function can provide the most improvement to online training, and how offline data quality affects performance.

5.1 Experiment Design

Environments and Offline data: We selected six continuous control tasks from the Gymnasium [47] to evaluate our causal PBRS method. We selected four MuJoCo [46] tasks on robotic locomotion including 1) Hopper - control a single legged agent to hop in 2D space without falling, 2) HalfCheetah - control a two legged, Cheetah-like, agent to run forward in 2D space as fast as possible, 3) Walker2D - control a bipedal agent to walk forward in 2D space without falling over, and 4) Ant - control a quadruped robot to walk forward in 3D space. We also selected two tasks from Adroit [36] on controling a robotic hand to open a door (Door) and to move a ball to the target location (Relocate). We use the offline datasets from Minari [50]. For details, see Sec. 8 in the appendix.

Confounded Environment Setup: To simulate unobserved confounders in our experiments, we remove dimensions from the environment's observation space. As a result, we simulate trajectories generated by an agent with access to richer sensory information (those removed dimensions), which is not accessible to the online agent. Consequently, typical off-policy learners cannot identify the behavioral policy or the state / q-value function accurately, given the unobserved confounders in the offline datasets [33, 51]. In practice, removing some of the dimensions will render the task much more challenging than removing others. For example, removing the agent's velocity observation in the MuJoCo environments - which is a key input into the environment's reward function - would make it difficult to derive the useful state value function from the offline data. On the other hand, removing an observation dimension that is independent of the state / Q-value function would not affect standard off-policy learners significantly. Therefore, to understand which dimensions to remove to create meaningful unobserved confounders, we use the Randomized Conditional Independence Test (RCIT, [43]) to measure the independence degree between an observation dimension and the episode's reward-to-go, conditioned on the remaining observations and actions. For detailed analysis on the relations between the removal of an observation dimension and the online agent's performance, see Sec. 5.3.

Offline and Online Training: Once we remove selected state dimensions from the offline dataset, we train our state value upper

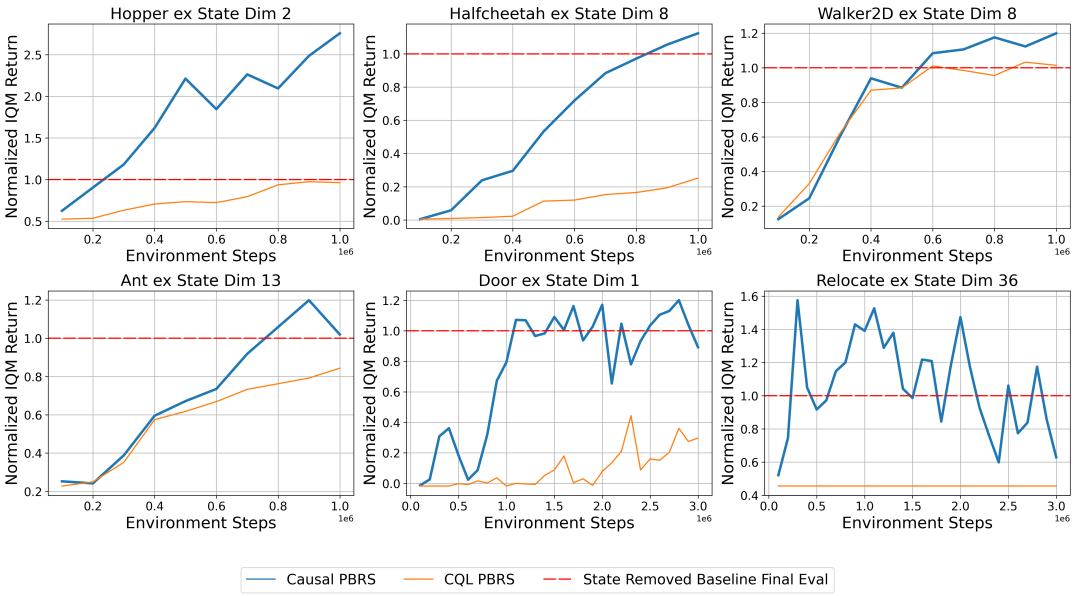


Figure 3: Normalized IQM returns w.r.t State Removed Baseline SAC agent in confounded continuous control benchmarks.

bounds following Algo. 1. We train the environment models for 50 epochs each, and then train the state value upper bounds for 200 epochs each. To prevent over-estimation (in particular from the "road not taken" estimate $\gamma \overline{V}_{\theta_1}(s + \Delta_s)$), we clip state value functions using $\frac{\max_{y \in D} y}{1-\gamma}$, the theoretical max value for the state value function given the maximum observed reward in the offline data. As a comparison to our causal shaping functions, we also train Q-value functions using the CQL algorithm [14].

After training the offline state value functions, we test three online learners: 1) a baseline SAC where we remove the selected state dimension from the observation space (State Removed), 2) a PBRS-based approach where we use the learned Q-value function from CQL (CQL PBRS), and 3) a PBRS-based approach where the potentials are the learned state value upper bounds (Causal PBRS, ours). For the Causal and CQL PBRS method, we apply a scaled version of the shaping rewards detailed in Prop. 4.3. Specifically, we use $y_{env} + \beta(\gamma\phi(s') - \phi(s))$ as the shaping reward, where y_{env} is the observed environment reward, β is a scaling factor, ϕ is the potential function, and s is the current state and s' is the next state. Empirically, we find the agent benefits the most from the potentials when $\beta \leq 1$ in those environments tested.

5.2 Causal Reward Shaping Performance

Fig. 3 shows the interquartile mean (IQM) returns normalized by the state removed baseline SAC best evaluation score in each environment. Table 1 provides the returns (including the best average return reached by the agent and the return at the final evaluation step), normalized mean, median, and inter-quartile mean return relative to the vanilla SAC baseline under partial state observation.

Overall, the Causal PBRS method outperforms the causally unaware baselines (CQL Shaping) consistently and exceeds the performance of the baseline method in 12 / 18 tests, for an average normalized mean score of 1.32 and normalized IQM of 1.10. In particular, the Causal PBRS method performs the best in cases where the removal of a certain state dimension leads to a larger decrease in the online baseline performance. For example, the Causal PBRS method outperforms the remove SAC baseline in the Hopper environment ex. State 1 and 2, where the online baseline does not surpass a score of 2000, versus Hopper ex. State 5, where the online baseline performance of around 3300 is comparable to the performance of the Hopper with full observational capabilities [5]. However, we note there is a limit, given that when removing states 1-4 in the Hopper environment (which are the dimensions related to joint positions), both the Causal PBRS and baselines are unable to learn a meaningful policy. Unsurprisingly, the causally unaware CQL PBRS method largely underperforms the online baseline (normalized mean of 0.86 at best eval), given that the unobserved confounding in the offline reward signal leads to highly biased Q-value estimations. In each of the following sections, we provide an overview of the observation dimensions removed in each environment and the agent's performance in those environments.

5.2.1 Hopper. We removed the following dimensions:

- State Dim 1: The Hopper's torso angle. One of the termination conditions in Hopper is if the torso angle is bounded between $[-0.2, 0.2]$. Without this dimension, the agent is unable to know how to adjust its torso angle to maintain healthy body positions. The Causal PBRS can improve on the baseline by around 20%, whereas CQL PBRS underperforms by 15%.
- State Dim 2: Angle of the thigh joint. Removing this dimension has a large impact on the Hopper's performance, reducing its baseline return by 50% vs the Hopper's performance with full obs capacity using SAC. The Causal PBRS method can drive a large improvement of almost 100% vs the baseline.
- State Dim 3: Angle of the leg joint. The leg joint is comparable to the knee, so without it, the Hopper has difficulties learning how to use its lower joints to propel itself forward. Interestingly, even with this dimension removed, the Hopper is able to get high returns, though it takes almost 900,000 steps. The

Table 1: Average evaluation returns of agents on the 18 confounded environments. All results are averaged over 5 seeds (except Door and Relocate, they are over 1 seed). "Best Eval" is the agent's best performance over all smoothed eval steps, and "Final Eval" is the performance at the end of training (1M steps for MuJoCo, 3M for Adroit). Bold Numbers indicate best best-performing methods. Our Causal PBRS method significantly outperforms the baselines.

Environment	State Dim Removed	Full State SAC	Best Eval			Final Eval		
			State Removed Baseline	CQL PBRS	**Causal PBRS (ours)**	State Removed Baseline	CQL PBRS	**Causal PBRS (ours)**
Hopper-v5	1	3500	2073.1	1777.5	**2251.8**	2005.1	1592.2	**2199.1**
Hopper-v5	2	3500	1450.5	1095.9	**2761.7**	1450.5	1018.5	**2729.2**
Hopper-v5	3	3500	3283.0	1478.0	**3581.6**	3280.7	1321.5	**3581.6**
Hopper-v5	5	3500	**3321.4**	1038.4	3183.5	**3297.9**	1035.0	3055.0
Hopper-v5	7	3500	2935.4	1245.5	**3011.0**	**2884.5**	1239.1	2793.0
Hopper-v5	5,6	3500	655.5	**1143.2**	1036.5	645.4	**1060.3**	902.5
Hopper-v5	1,2,3,4	3500	304.0	**556.8**	304.6	265.2	527.8	301.5
Hopper-v5	7,8,9,10	3500	371.7	421.9	**1442.4**	366.7	408.8	**1228.2**
Halfcheetah-v5	1	12400	2013.2	2038.8	**2054.7**	2013.2	2038.8	**2051.2**
Halfcheetah-v5	3	12400	8930.5	**9747.7**	8954.4	8930.5	**9747.7**	8948.9
Halfcheetah-v5	8	12400	896.3	372.8	**1103.3**	895.1	372.8	**1103.3**
Walker2D-v5	6	4050	**3980.9**	3242.6	3899.2	**3893.1**	3132.4	3883.9
Walker2D-v5	8	4050	3640.4	3765.7	**4295.0**	3580.1	**3708.1**	3632.3
Walker2D-v5	10	4050	3925.4	3600.3	**3950.3**	**3925.4**	3424.4	3813.6
Ant-v5	13	4000	3084.3	2791.8	**3388.9**	3081.6	2781.0	3093.1
Ant-v5	16	4000	**3749.9**	3190.4	2962.9	**3473.8**	3169.2	2818.3
AdroitHandDoor-v1	1	N/A	2517.6	1119.5	**3124.0**	44.0	749.4	**2247.1**
AdroitHandRelocate-v1	36	N/A	25.8	11.7	**43.9**	10.5	11.7	**16.2**
Normalized Mean (↑)			1.00	0.86	**1.32**	1.00	1.81	**4.03**
Normalized Median (↑)			1.00	0.85	**1.09**	1.00	0.91	**1.06**
Normalized IQM (↑)			1.00	0.81	**1.10**	1.00	0.92	**1.10**

Causal PBRS method is able to recover full state observation SAC's performance, and does so in only around 500,000 steps.

- State Dim 5: The Hopper's forward velocity. The Hopper's reward function is heavily dependent on its forward velocity. Therefore, not observing the forward velocity might decrease the critic's ability to estimate the Q-value function. Surprisingly, without this state dimension, the baseline agent can reach a return comparable to a Hopper agent with full observation capabilities. We note that the Causal PBRS agent reaches its highest return after only 400,000 time steps, whereas the baseline agent requires almost 900,000 steps. This highlights the Causal PBRS's ability to accelerate training.
- State Dim 7: The angular acceleration of the Hopper's torso. The Causal PBRS's average return is slightly higher than the baseline agent; however, we note the Causal PBRS reaches its performance peak at around 300,000 steps, whereas the baseline agent reaches its peak at 900,000 steps.
- State Dim 1,2,3,4: All of the angular positions of the Hopper's joints and Torso. Without knowing the position of its joints, the Hopper agent is unable to learn a meaningful policy. The Causal PBRS method is similarly unable to learn a meaningful policy, suggesting a limit to the Causal PBRS's ability to improve performance in highly confounded environments. Interestingly, the CQL method can generate a slightly better policy, although we note the overall still low return.
- State Dim 5,6: The forward and vertical velocity of the hopper. The Causal PBRS agent can learn a better policy than the baseline, achieving almost double the baseline's return. The CQL method can learn a slightly better policy.
- State Dim 7,8,9,10: The angular acceleration of the hopper's torso and joints. Interestingly, the baseline Hopper's performance is comparable to its performance when removing dimensions 1,2,3,4, however, the Causal PBRS agent can learn a more meaningful policy (return between 1200-1400).

5.2.2 HalfCheetah. We removed the following dimensions:

- State Dim 1: Angle of the front tip. Without the angle of the front tip (used to know the orientation of the Cheetah), the Half Cheetah agent's return drastically decreases from 12,000 to around 2,000. The baseline, the Causal PBRS methods, and the CQL PBRS methods all perform comparably.
- State Dim 3: Angle of the back shin. The back shin in the middle joint on the HalfCheetah's back leg. Removing this dimension from the observation dimension decreases performance slightly (10%). Similar to removing state dimension 1, the Causal PBRS method performs similarly to the baseline; however, the CQL PBRS method can learn the best policy.
- State Dim 8: Velocity of the x-coordinate of the front tip. Unlike the Hopper, removing the forward velocity greatly reduces the HalfCheetah's return. The Causal PBRS method achieves a higher return vs the baseline policy.

5.2.3 Walker2D. We removed the following dimensions:

- State Dim 6: Angle of the left leg joint. Removing the left leg joint has a limited effect on the Walker2d. Both the Causal PBRS and Baseline methods perform comparably, but the CQL method underperforms both by almost 20%.
- State Dim 8: Velocity of the x-coordinate of the torso. Removing the forward velocity dimension of the Walker2d slightly decreases performance. The Causal PBRS method achieves a 20% improvement over the baseline method.

- State Dim 10: Angular velocity of the angle of the torso. Removing the left leg joint observation dimensions has a limited effect on the Walker-2d. The Causal PBRS, Online Baseline, and CQL PBRS perform comparably.

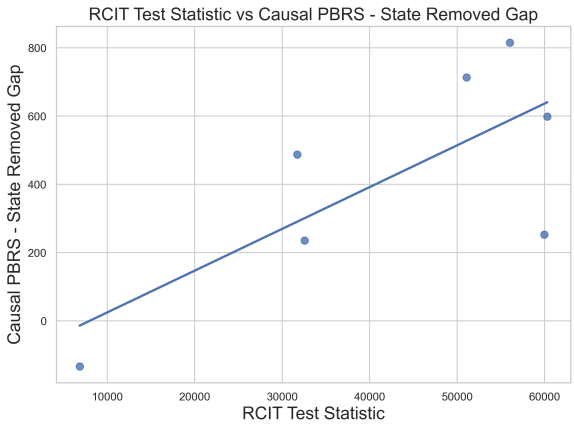


Figure 4: RCIT test statistic v.s. Causal PBRS improvements.

5.2.4 Ant. We removed the following dimensions:

- State Dim 13: Velocity of the x-coordinate of the torso. Similar to other environments, the Causal PBRS method helps the Ant recover some of the performance loss from the loss of its forward velocity observation, unlike the CQL PBRS method.
- State Dim 16: x-coordinate angular velocity of the torso. The Causal PBRS underperforms the baseline; however, the baseline performs much better than the baseline when removing dimension 13, suggesting less confounding bias in this setting and hence a potential reason for the poor performance. The CQL PBRS method slightly outperforms the Causal PBRS method, but still underperforms the baseline.

5.2.5 Adroit Door. We removed the following dimensions:

- State Dim 1: Angular position of the horizontal arm joint. The baseline agent and Causal PBRS agent perform comparably. The Causal PBRS method outperforms the baseline by 24%, while the CQL PBRS method lags behind.

5.2.6 Adroit Relocate. We removed the following dimensions:

- State Dim 36: x positional difference from the ball to the target. While the agent is still able to infer this position through other observations (namely dimensions 30 and 33), the overall performance is still low. The Causal PBRS method outperforms both the baseline and Causal PBRS method.

5.3 Relation Between State Return Independence and Performance Gap

To understand when the Causal PBRS method can out perform the online baseline without shaping, we plot the test statistic of the RCIT test statistic from the conditional independence of the selected state dimension and the remaining returns-to-go (conditioned on the remaining states and actions) from the Hopper-v5 Environment, and the average gap between the Causal PBRS method and the Online State Removed Baseline (Fig. 4). A higher test statistic implies that the observation dimension has a higher dependence on the returns-to-go, conditioned on the remaining observation dimensions and actions. In Fig. 4, as the test statistic increases, the gap between the Causal PBRS method and the Online Baseline increases, thus supporting our theory that the Causal PBRS method performs better when there is an increase in confounding bias. We also notice that when we remove dimensions necessary for the task (those with the highest test statistic), the learned causal shaping function may not be informative enough to help recover the performance. See Sec. 10 for a more detailed analysis.

5.4 Impact of Offline Data Quality on Online Training with Causal PBRS

For our experiments, we used the Minari offline datasets including three levels of expertise. These three levels of expertise are denoted as "simple", "medium", and "expert". The results in Table 1 are based on training the optimistic state value functions with all three datasets. To evaluate how the expertise of the agent generating the offline data affects the Causal PBRS method, we trained three different causal reward shaping function, each on one of the Minari datasets. Fig. 5 shows the results. The Causal PBRS trained solely on the expert data performed the best. However, the Causal PBRS trained with the medium and simple datasets is still able to improve the agent's policy over the baseline. Notably, the performance of the combined datasets (Combined) is comparable to the Expert performance, albeit slightly slower to converge, suggesting Algo. 1 can converge well in practice, despite the differing data quality.

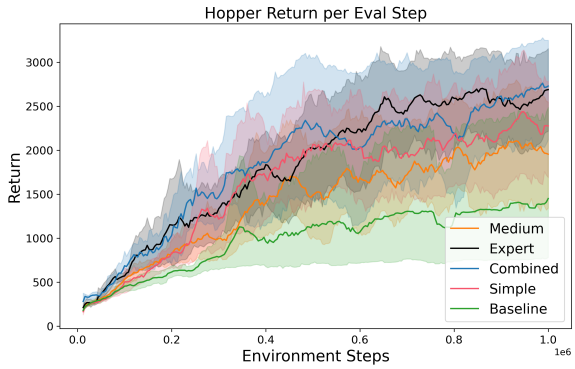


Figure 5: Causal PBRS performance by offline data quality.

6 CONCLUSION

We introduce a causal framework for automatic reward shaping in high-dimensional continuous control with unobserved confounders. By extending the Causal Bellman Equation to continuous settings, our method learns optimistic state potentials that serve as principled shaping functions within the PBRS framework, improving learning efficiency while preserving policy optimality. We evaluate our approach on confounded MuJoCo and Adroit benchmarks, where it consistently outperforms unshaped and causally unaware baselines such as CQL PBRS (achieving a normalized IQM score of 1.10 vs. the unshaped baseline). This work marks an important step toward confounding-robust reinforcement learning and causal reward design for real-world continuous control.

REFERENCES

[1] Stefan Banach. 1922. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae* 3 (1922), 133–181. https://api.semanticscholar.org/CorpusID:118543265

[2] Andrew Bennett and Nathan Kallus. 2024. Proximal Reinforcement Learning: Efficient Off-Policy Evaluation in Partially Observed Markov Decision Processes. *Oper. Res.* 72, 3 (2024), 1071–1086. https://doi.org/10.1287/OPRE.2021.0781

[3] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E. Taylor, and Ann Nowé. 2015. Reinforcement Learning from Demonstration through Shaping. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael J. Wooldridge (Eds.). 3352–3358. http://ijcai.org/Abstract/15/472

[4] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement Learning with Deep Energy-Based Policies. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). 1352–1361. http://proceedings.mlr.press/v70/haarnoja17a.html

[5] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.. In *ICML (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). 1856–1865. http://dblp.uni-trier.de/db/conf/icml/icml2018.html#HaarnojaZAL18

[6] Edward S. Hu, James Springer, Oleh Rybkin, and Dinesh Jayaraman. 2024. Privileged Sensing Scaffolds Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=EpVe8jAjdx

[7] Dongchi Huang, Jiaqi WANG, Yang Li, Chunhe Xia, Tianle Zhang, and Kaige Zhang. 2025. PIGDreamer: Privileged Information Guided World Models for Safe Partially Observable Reinforcement Learning. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=mtk8tTKWs0

[8] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. 2022. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research* 23, 274 (2022), 1–18. http://jmlr.org/papers/v23/21-1342.html

[9] Sinan Ibrahim, Mostafa Mostafa, Ali Jnadi, Hadi Salloum, and Pavel Osinenko. 2024. Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications. *IEEE Access* 12 (2024), 175473–175500. https://doi.org/10.1109/ACCESS.2024.3504735

[10] Tommi S. Jaakkola, Satinder Singh, and Michael I. Jordan. 1994. Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. In *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, Gerald Tesauro, David S. Touretzky, and Todd K. Leen (Eds.). 345–352. https://proceedings.neurips.cc/paper_files/paper/1994/hash/1c1d4df596d01da60385f0bb17a4a9e0-Abstract.html

[11] Nan Jiang and Lihong Li. 2016. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). 652–661. http://proceedings.mlr.press/v48/jiang16.html

[12] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

[13] Nathan Kallus and Angela Zhou. 2018. Confounding-Robust Policy Improvement. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 9289–9299. https://proceedings.neurips.cc/paper/2018/hash/3a09a524440d44d7f19870070a5ad42f-Abstract.html

[14] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html

[15] Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. 2021. Sequential Causal Imitation Learning with Unobserved Confounders. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 14669–14680. https://proceedings.neurips.cc/paper/2021/hash/7b670d553471ad0fd7491c75bad587ff-Abstract.html

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

[17] Mingxuan Li, Junzhe Zhang, and Elias Bareinboim. 2024. Causally Aligned Curriculum Learning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=hp4yOjhwTs

[18] Mingxuan Li, Junzhe Zhang, and Elias Bareinboim. 2025. Automatic Reward Shaping from Confounded Offline Data. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=Hu7hUjEMiW

[19] Mingxuan Li, Junzhe Zhang, and Elias Bareinboim. 2025. Confounding Robust Reinforcement Learning: A Causal Approach. In *Advances in Neural Information Processing Systems 39: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2 - 7, 2025*. https://neurips.cc/virtual/2025/poster/119520

[20] Miao Lu, Yifei Min, Zhaoran Wang, and Zhuoran Yang. 2023. Pessimism in the Face of Confounders: Provably Efficient Offline Reinforcement Learning in Partially Observable Markov Decision Processes. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. https://openreview.net/forum?id=PbkBDQ5_UbV

[21] Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. 2022. Learning Goal-Conditioned Policies Offline with Self-Supervised Reward Shaping. In *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand (Proceedings of Machine Learning Research, Vol. 205)*, Karen Liu, Dana Kulic, and Jeffrey Ichnowski (Eds.). 1401–1410. https://proceedings.mlr.press/v205/mezghani23a.html

[22] Rui Miao, Zhengling Qi, and Xiaoke Zhang. 2022. Off-Policy Evaluation for Episodic Partially Observable Markov Decision Processes under Non-Parametric Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/03dfa2a7755635f756b160e9f4c6b789-Abstract-Conference.html

[23] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). 1928–1937. http://proceedings.mlr.press/v48/mniha16.html

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533. https://doi.org/10.1038/nature14236

[25] Gianluca Monaci, Michel Aractingi, and Tomi Silander. 2022. DiPCAN: Distilling Privileged Information for Crowd-Aware Navigation. In *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July 1, 2022*, Kris Hauser, Dylan A. Shell, and Shoudong Huang (Eds.). https://doi.org/10.15607/RSS.2022.XVIII.045

[26] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. 2016. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*. 1054–1062.

[27] Susan A Murphy. 2003. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 65, 2 (2003), 331–355.

[28] Susan A Murphy. 2005. A Generalization Error for Q-Learning. *Journal of Machine Learning Research* 6 (2005), 1073–1097.

[29] Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. 2023. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/c44a04289beaf0a7d968a94066a1d696-Abstract-Conference.html

[30] Andrew Y. Ng, Daishi Harada, and Stuart Russell. 1999. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, Ivan Bratko and Saso Dzeroski (Eds.). 278–287.

[31] 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. 2019. Solving Rubik's Cube with a Robot Hand. (2019). arXiv:1910.07113 http://arxiv.org/abs/1910.07113 arxiv.

[32] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. https://openreview.net/forum?id=JYtwGwIL7ye

[33] Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference* (2 ed.). Cambridge University Press. https://doi.org/10.1017/CBO9780511803161

[34] Doina Precup, Richard S. Sutton, and Satinder P. Singh. 2000. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*. 759–766.

[35] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1 ed.). Wiley. https://doi.org/10.1002/9780470316887

[36] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations.

[37] Jette Randløv and Preben Alstrøm. 1998. Learning to Drive a Bicycle Using Reinforcement Learning and Shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, Jude W. Shavlik (Ed.). 463–471.

[38] John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. 1889–1897.

[39] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1506.02438

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347 arxiv.

[41] Chengchun Shi, Masatoshi Uehara, Jiawei Huang, and Nan Jiang. 2022. A Minimax Learning Approach to Off-Policy Evaluation in Confounded Partially Observable Markov Decision Processes. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). 20057–20094. https://proceedings.mlr.press/v162/shi22f.html

[42] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. 2023. RoboCook: Long-Horizon Elasto-Plastic Object Manipulation with Diverse Tools. In *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA (Proceedings of Machine Learning Research, Vol. 229)*, Jie Tan, Marc Toussaint, and Kourosh Darvish (Eds.). 642–660. https://proceedings.mlr.press/v229/shi23a.html

[43] Eric Strobl, Kun Zhang, and Shyam Visweswaran. 2018. Approximate Kernel-Based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. *Journal of Causal Inference* 7 (12 2018). https://doi.org/10.1515/jci-2018-0017

[44] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*. 814–823.

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

[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033. https://doi.org/10.1109/IROS.2012.6386109

[47] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments.

[48] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.

[49] Christopher John Cornish Hellaby Watkins. 1989. *Learning from delayed rewards*. Ph.D. Dissertation. University of Cambridge England.

[50] Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. 2024. *Minari*. Farama Foundation. https://doi.org/10.5281/zenodo.13767625

[51] Junzhe Zhang and Elias Bareinboim. 2019. Near-optimal reinforcement learning in dynamic treatment regimes. In *Advances in Neural Information Processing Systems*. 13401–13411.

[52] Junzhe Zhang and Elias Bareinboim. 2022. Can Humans Be out of the Loop?. In *1st Conference on Causal Learning and Reasoning, CLeaR 2022, Sequoia Conference Center, Eureka, CA, USA, 11-13 April, 2022 (Proceedings of Machine Learning Research, Vol. 177)*, Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (Eds.). 1010–1025. https://proceedings.mlr.press/v177/zhang22a.html

[53] J. Zhang and E. Bareinboim. 2024. *Eligibility Traces for Confounding Robust Off-Policy Evaluation*. Technical Report R-105. Causal Artificial Intelligence Lab, Columbia University.

[54] Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. 2020. Causal imitation learning with unobserved confounders. *Advances in neural information processing systems* 33 (2020), 12263–12274.

[55] Yi Zhang, Ruihong Qiu, Jiajun Liu, and Sen Wang. 2024. ROLeR: Effective Reward Shaping in Offline Reinforcement Learning for Recommender Systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, Edoardo Serra and Francesca Spezzano (Eds.). 3269–3278. https://doi.org/10.1145/3627673.3679633

[56] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum Entropy Inverse Reinforcement Learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, Dieter Fox and Carla P. Gomes (Eds.). 1433–1438. http://www.aaai.org/Library/AAAI/2008/aaai08-227.php

APPENDICES

Contents

7 PROOF DETAILS

Here we present the proof details of theorems and propositions in the main text.

7.1 Proof for Thm. 4.1 Causal Bellman Optimal Equation for Stationary Infinite-Horizon CMDPs

Proof. Starting from the Bellman Optimal Equation, the optimal state value function is given by,

$$V^*(s) = \max_x R(s,x) + \gamma \sum_{s'} T(s,x,s')V^*(s') \tag{18}$$

Note that the actions here are done by an interventional agent, which is actually $\text{do}(x)$ in the context of a CMDP. We swap in the causal bounds for interventional reward and transition distribution,

$$V^*(s) \leq \max_x \left[\widetilde{R}(s,x)P(x|s) + bP(\neg x|s) + \gamma \sum_{s'} \widetilde{T}(s,x,s')P(x|s)V^*(s') + P(\neg x|s) \max_{s''} V^*(s'') \right] \tag{19}$$

where $\widetilde{\mathcal{R}}(s,x) = \mathbb{E}[Y|S=s, X=x]$, $\widetilde{\mathcal{T}_h}$ is shorthand for $\widetilde{\mathcal{T}}(s,x,s') = P(S'=s'|S=s, X=x)$ and $P(x|s) = P(X=x|S=s)$ are estimated from the offline dataset. And b is a known upper bound on the reward signal, $Y \leq b$. In this step, we upper bound the next state transition by assuming the best case that for the action not taken with probability $P(\neg x|s)$, the agent transits with probability 1 the best possible next state, $\max_{s''} V^*(s'')$. After rearranging terms, we have,

$$V^*(s) \leq \max_x \left[P(x|s) \left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s'} \widetilde{T}(s,x,s')V^*(s') \right) + P(\neg x|s) \left(b + \gamma \max_{s''} V^*(s'') \right) \right] \tag{20}$$

And optimizing the value function w.r.t this inequality gives us an upper bound on the optimal state value,

$$\overline{V}(s) \leq \max_x \left[P(x|s) \left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s'} \widetilde{T}(s,x,s')\overline{V}(s') \right) + P(\neg x|s) \left(b + \gamma \max_{s''} \overline{V}(s'') \right) \right]. \tag{21}$$

□

7.2 Proof for Thm. 4.2 Convergence of Causal Bellman Optimal Equation

Proof. We will first show that the following Causal Bellman Optimality operator (will denote as "the operator" or B below for simplicity) is a contraction mapping with respect to a max norm. Then by Banach's fixed-point theorem [1], this operator has a unique fixed point, and updating any initial point iteratively will converge to it. Then we show that this unique fixed point is indeed a lower bound of the optimal interventional Q-value.

Let the operator B be,

$$B\overline{V}(s,x) = \max_x \left[P(x \mid s) \left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s')\,\overline{V}(s') \right) + P(\neg x \mid s) \left(b + \gamma \max_{s'} \overline{V}(s') \right) \right]. \tag{22}$$

For arbitrary value bounds, $\overline{V^1}, \overline{V^2}$, let their initial difference under max-norm be $c = \left\| \overline{V^1} - \overline{V^2} \right\|_\infty \geq 0$. We can bound their difference after one step update by,

$$\left\| B\overline{V^1} - B\overline{V^2} \right\|_\infty \leq \gamma \max_{s,x} \left[P(x|s) \sum_{s'} \widetilde{T}(s,x,s') \left(\overline{V^1}(s') - \overline{V^2}(s') \right) + P(\neg x|s) \max_{s'} \left(\overline{V^1}(s') - \overline{V^2}(s') \right) \right]. \tag{23}$$

Thus, under the operator T, we have non-expansion Q-value differences,

$$\left\|B\overline{V^1} - B\overline{V^2}\right\|_\infty \leq \gamma \max_{s,x}\left[P(x|s)\sum_{s'}\widetilde{T}(s,x,s')\left(\overline{V^1}(s') - \overline{V^2}(s')\right) + P(\neg x|s)\max_{s'}\left(\overline{V^1}(s') - \overline{V^2}(s')\right)\right] \tag{24}$$

$$\leq \gamma c \max_{s,x}\left(P(x|s)\sum_{s'}\widetilde{T}(s,x,s') + P(\neg x|s)\right), \tag{25}$$

$$= \gamma c. \tag{26}$$

for all $\overline{V^1}, \overline{V^2}$ satisfying $\left\|B\overline{V^1} - B\overline{V^2}\right\|_\infty \leq c, c \geq 0$. Thus, T is a contraction mapping with respect to the max norm. And there exists a unique fixed point $\overline{V^*}$ when we apply this operator B iteratively to an arbitrary state value vector till convergence.

We then show that this fixed point is indeed an upper bound to the optimal interventional state values. By the backup rule of B (Eq. (21)), $\forall V(s), V(s) \leq BV(s)$. Thus, for the optimal state value, we can have $V^*(s) \leq \lim_{k\to\infty} B^k V^*(s) = \overline{V^*}(s)$ where B^k denotes applying causal bellman backup B iteratively for k times. This concludes the proof. □

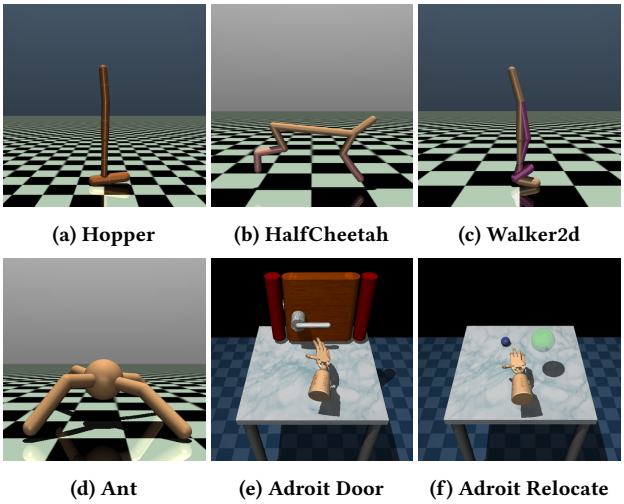
7.3 Proof for Prop. 4.3

Proof. Because CMDP also enjoys the Markov property, the overall proof procedure highly resembles the original one in Ng et al. [30]. Only to note that the optimal policy invariance is proved in the online learning sense, which is between CMDP $\mathcal{M}'_\pi$ after reward shaping under policy π and the original CMDP $\mathcal{M}_\pi$ under policy π. □

8 ENVIRONMENTS AND OFFLINE DATASET

Fig. 6 provides visualization of the six environments tested.

For offline datasets, the MuJoCo tasks use offline data generated by three different policies of varying expertise (simple-v0, medium-v0, expert-v0). The Adroit tasks use mixed offline data generated by human demonstrators (human-v2), an "expert" fine-tuned RL policy (expert-v2), and an imitation policy of the Human and Expert policies (cloned-v2).



(a) Hopper **(b) HalfCheetah** **(c) Walker2d**

(d) Ant **(e) Adroit Door** **(f) Adroit Relocate**

Figure 6: Visualizations of the Six Environments Tested.

Table 2: Offline Dataset Size

Environment	Trajectory Size
Hopper-v5	2,997,774
HalfCheetah-v5	3,000,000
Walker2d-v5	2,998,745
Ant-v5	3,998,498
AdroitHandDoor-v1	2,009,942
AdroitHandRelocate-v1	2,006,729

Data pulled from the Minari data repository [50]. The Hopper, HalfCheetah, and Walker2d offline datasets roughly have the same split of Simple, Medium, and Expert policies. The Ant dataset is 1/2 Expert data, with the remaining data evenly split between the Simple and Medium policies. The Door and Relocate datasets have 1m observations from the Expert and Cloned policies, with the remaining coming from Human demonstrators.

9 HYPER PARAMETERS

9.1 Offline Hyper Parameters

For CQL, we used the baseline implementation and training hyperparameters in [8], trained the Q-value function for 1M timesteps, and used a batch size of 1028. For the Causal Upper Bounded State Value functions, we trained an environment model for 50 epochs to estimate the parameters for θ_2 and θ_3 and then trained Algo. 1 for 200 epochs, with a batch size of 1028. This is roughly equivalent to 600k timesteps in environments with 3M trajectories. To facilitate convergence, we mean-normalized the offline rewards.

Table 3: Hyperparameters for Offline Training Causal Upper Bounded State Value Functions

Hyperparameter	Value
Environment Model Training Epochs	50
Causal Upper Bound Model Training Epochs	200
Optimizer	Adam
Batch Size	1028
Policy θ_2 Learning Rate	1e-4
State Transition θ_3 Learning Rate	1e-5
Q Learning Rate	1e-4
Discount Factor	0.99
Target Network Update τ	0.005
Target Update Interval	1
Policy Training Frequency	3
Num Hidden Dim	128
Num Residual Blocks	3

9.2 Online Hyper Parameters

To tune the hyperparameter β as detailed in Sec. 5.1, we tested the following values: 1, 0.1, 0.01, and 0.001, to find the β that resulted in the best performance at the overall environment level. Sec. 9.2 includes the β value used for each environment and method. Given that the agent's performance is sensitive to β, the performance of an agent using Causal or CQL PBRS methods might be improved by further hyper-tuning or optimizing β (instead of picking one β for the whole environment).

10 RELATION BETWEEN CONDITIONAL INDEPENDENCE AND PERFORMANCE

We looked at the relation between the RCIT test statistic (measure of dependence of selected state dimension and returns-to-go, conditioned on remaining states and actions) and performance of the State Removed Baseline and Causal PBRS method. Note, a higher RCIT test statistic implies a higher dependence between the two selected variables, conditioned on the remaining variables. We use results from the Hopper-v5 with the following re dimensions in the Hopper Environment - 1, 2, 3, 5, 6, 7, 1 - 4, 5 - 6, 7 - 10. State Removed Gap is defined as the average eval return gap between the State Removed SAC Baseline and a full state SAC Hopper agent. The Causal Gap is defined as the average eval return gap between the State Removed SAC Baseline and the Causal PBRS method. As seen in Fig. 7a, there is a negative correlation between an increase in conditional dependence, and Hopper performance degradation. Similarly, as seen in Fig. 7b, there is a positive relationship

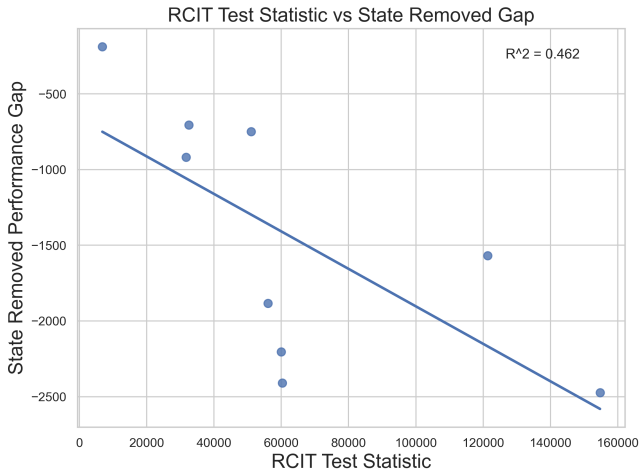
Table 4: Hyperparameters for Online Training SAC

Hyperparameter	Value
Training Steps	1e6 (MuJoCo), 3e6 (Adroit)
Optimizer	Adam
Batch Size	512
Policy Learning Rate	3e-4
Q Learning Rate	1e-3
Alpha	0.2
Discount Factor	0.99
PBRS Discount Factor	1
PBRS β	See Sec. 9.2
Target Network Update τ	0.005
Target Update Interval	1
Policy Training Frequency	2
Gradient Steps	1
Num Hidden Dim	256
Num Residual Blocks	2

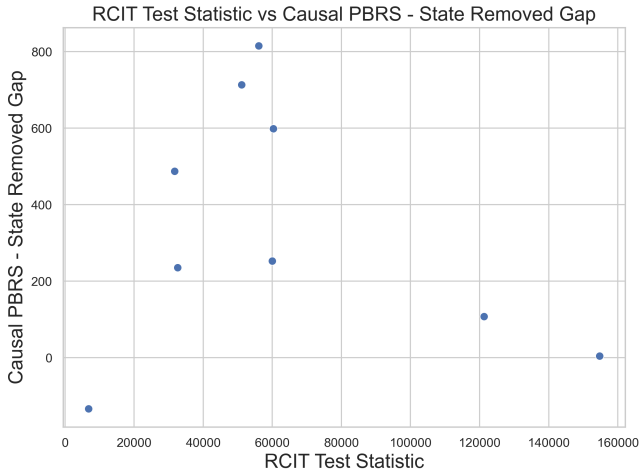
Table 5: SAC PBRS Scaling Factors

Environment	Causal Beta	CQL Beta
Hopper-v5	0.1	0.1
HalfCheetah-v5	0.001	0.001
Walker2d-v5	0.1	0.01
Ant-v5	0.001	0.001
AdroitHandDoor-v1	0.1	0.1
AdroitHandRelocate-v1	0.1	0.1

between an increase in the RCIT test statistic and gap between the Causal PBRS and State Removed Baseline - but only up until a point. For certain dimensions that have a very high dependence, the causal upper bounded state value function might not be as informative, and hence a less improvement when compared the to State Removed Baseline.



(a) Relation Between RCIT test statistic and State Removed Baseline Degradation in Hopper-v5.



(b) Relation Between RCIT test statistic and Causal PBRS Improvement vs State Removed Baseline.

Figure 7: Relations between RCIT test statistic, baseline degradation, and causal PBRS improvement in Hopper-v5.

11 FUTURE DIRECTIONS AND CHALLENGES

This work takes a step towards improving reinforcement learning in confounded, continuous control environments and causal reward design for real-world continuous control. In this section, we discuss current limitations, challenges, and consequently future directions. First, estimating the value of the "road-not-taken" to adjust for potential confounders in Thm. 4.1 is not trivial. In particular, deciding which actions $\neg X$ to take is challenging given that in some offline datasets, there may not be enough action policy variance to 1) determine reasonable $\neg X$ actions and 2) understand the state transition probability given state s and action $\neg X$. One potential solution and area of future work is to include datasets generated by agents with less sophisticated policies ([18] demonstrates, including datasets with trajectories from less sophisticated agents can create better causal reward shaping functions) in the offline dataset training. Adding and testing the Causal PBRS method's efficacy with a wider range of policy levels could also further the understanding of how much expert data is needed to create effective potentials. Finally, the other main challenge of this work was selecting the appropriate scaling factor, β, during the online training phase. Implementing methods to automatically tune or optimize β could lead to a large performance improvement in the Causal PBRS method.

12 ADDITIONAL EXPERIMENT RESULTS

For Door and Relocate experiments, we only run 1 seed with 3M training steps due to limited time and training resources.

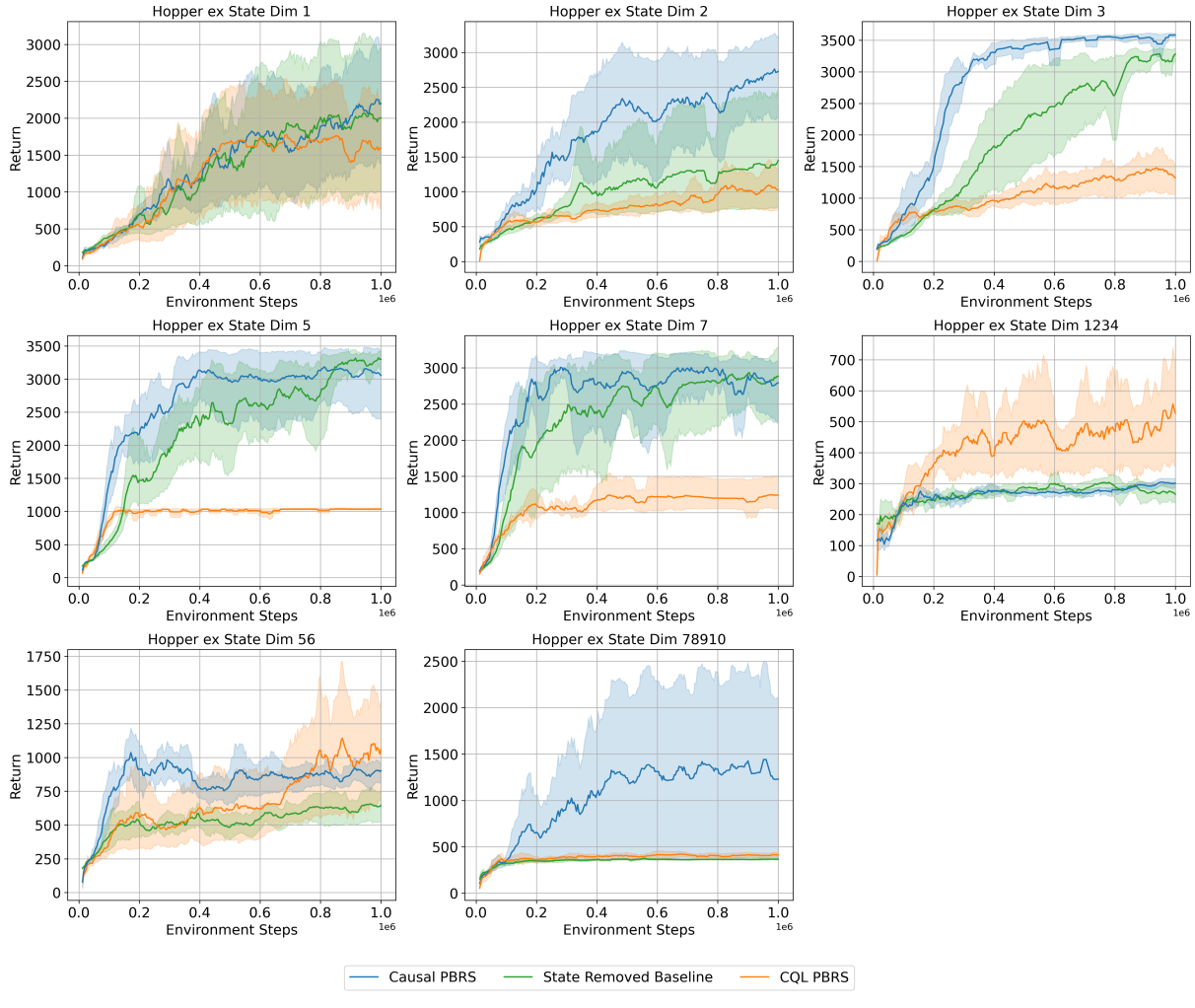


Figure 8: Confounded Hopper-v5 Experiment Results

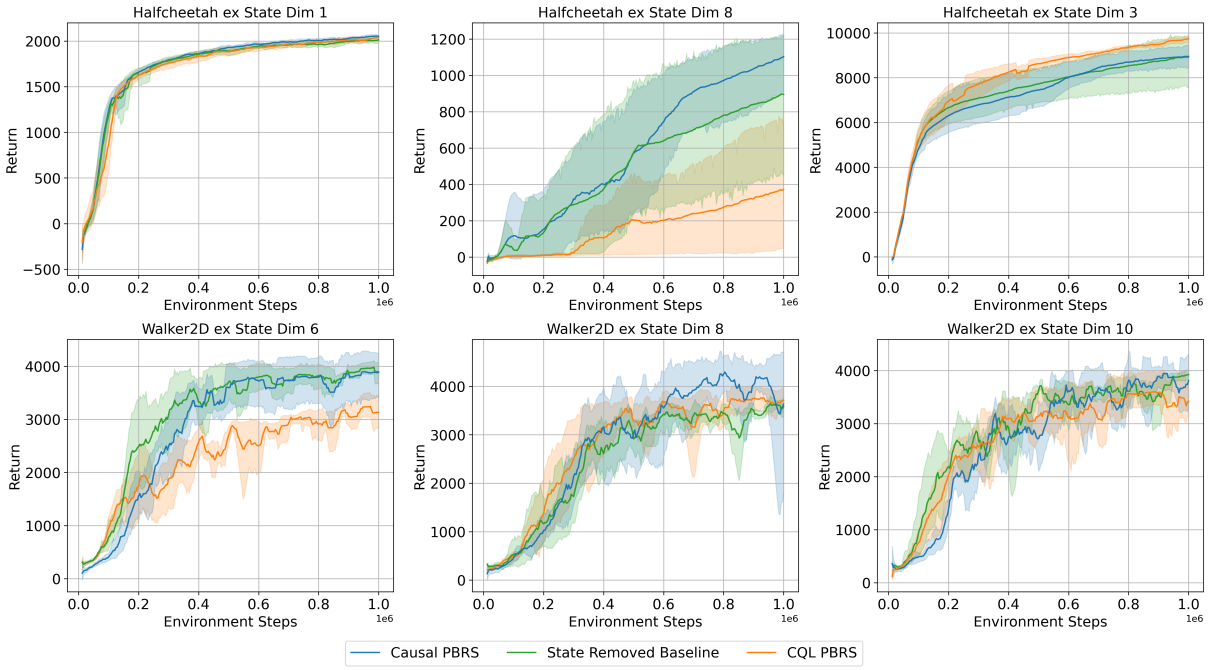


Figure 9: Confounded HalfCheetah-v5 and Walker2d-v5 Experiment Results

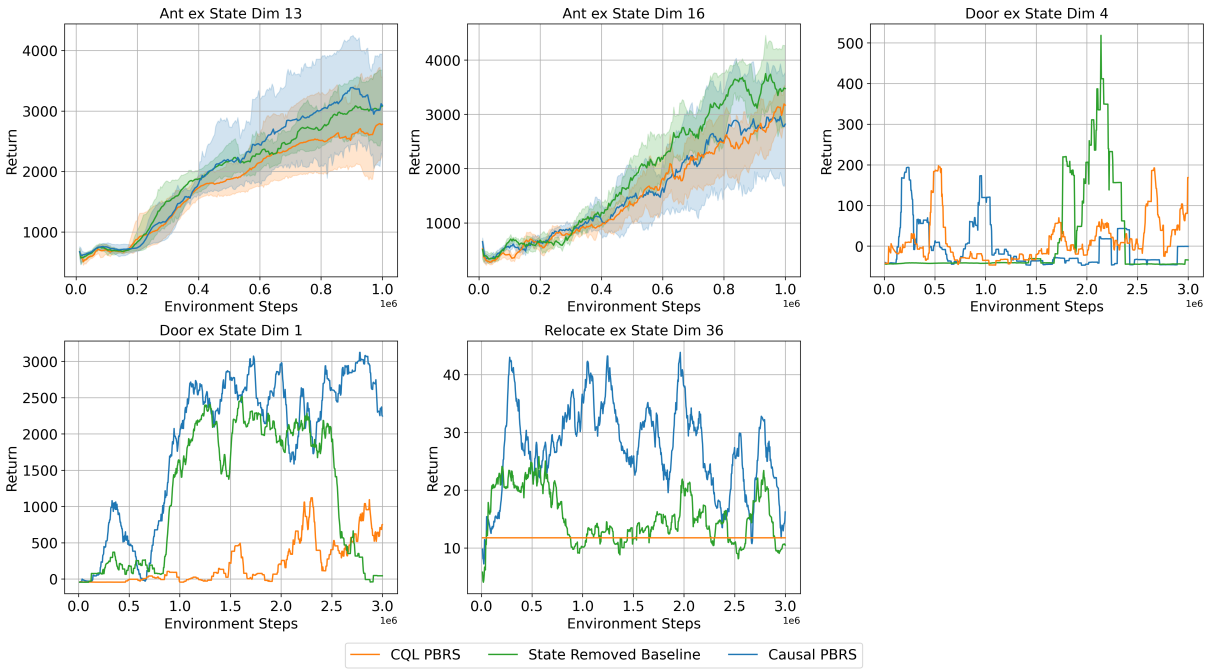


Figure 10: Confounded Ant-v5, AdroitHandDoor-v1, AdroitHandRelocate-v1 Experiment Results (Returns)

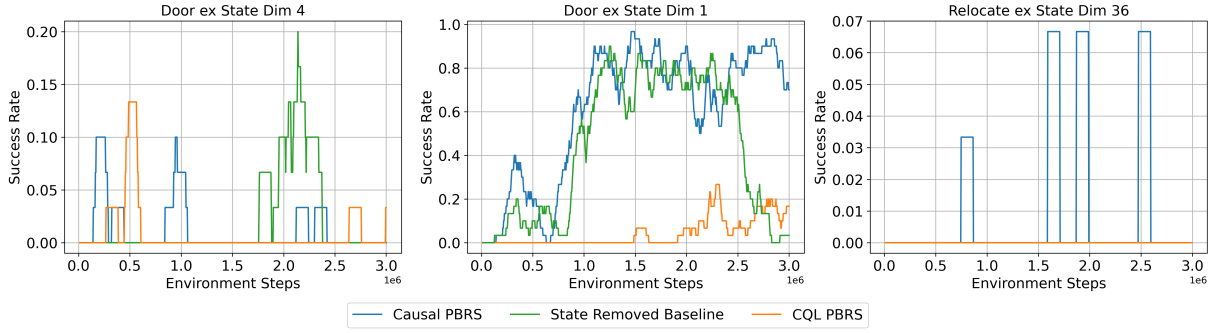


Figure 11: Confounded AdroitHandDoor-v1, AdroitHandRelocate-v1 Experiment Results (Success Rates)