Confounding Robust Deep Reinforcement Learning: A Causal Approach

Mingxuan Li[1*] **Junzhe Zhang**[2*] **Elias Bareinboim**[1]
[1] Columbia University, [2] Syracuse University
[1]{ml,eb}@cs.columbia.edu, [2]jzhan403@syr.edu

Abstract

A key task in Artificial Intelligence is learning effective policies for controlling agents in unknown environments to optimize performance measures. Off-policy learning methods, like Q-learning, allow learners to make optimal decisions based on past experiences. This paper studies off-policy learning from biased data in complex and high-dimensional domains where *unobserved confounding* cannot be ruled out a priori. Building on the well-celebrated Deep Q-Network (DQN), we propose a novel deep reinforcement learning algorithm robust to confounding biases in observed data. Specifically, our algorithm attempts to find a safe policy for the worst-case environment compatible with the observations. We apply our method to twelve confounded Atari games, and find that it consistently dominates the standard DQN in all games where the observed input to the behavioral and target policies mismatch and unobserved confounders exist.

1 Introduction

Over the last decade, reinforcement learning (RL) has gained significant popularity for solving complex sequential decision-making problems, primarily due to its integration with deep learning techniques [24, 52, 83]. This approach, known as deep reinforcement learning (deep RL), is particularly effective in high-dimensional state spaces [40, 61, 70, 84, 85, 88]. Deep RL addresses the challenges faced by earlier RL methods by extracting various abstractions from data in complex domains with minimal prior knowledge. For example, a classic algorithm known as Deep Q-Network (DQN) algorithm can efficiently learn from visual inputs containing thousands of pixels [61], enabling problem-solving capabilities comparable to humans in certain high-dimensional environments, such as Atari games for the first time. These achievements were followed by notable advancements in deep RL, including mastering the game of Go [89], defeating world-class professionals at the game of poker [64]. Deep RL has also shown potentials in various real-world applications like robotics [70, 88], autonomous driving [49] and protein design [40]. These advancements eventually culminated in Sutton and Barto receiving the Turing Award in 2025 [22].

This paper attempts to leverage the capabilities and insights of DQN, while identifying an important assumption embedded in this algorithm and its variants that does not necessarily hold in the real world. Particularly, we notice that it is often implicitly assumed through the (PO)MDPs [93] framework or explicitly enforced during the data-collection that no unmeasured confounder (NUC, [11, 78]) affects the observed action and the subsequent outcomes. When the NUC does not hold, the effect of the target policy is generally not *identifiable*, i.e., the model assumptions are insufficient to uniquely determine the value function from the offline data [72, 111]. On the other hand, partial identification is a line of methodologies that enable the derivation of informative bounds on target effects from confounded observations in non-identifiable settings [59]. It has been studied under the rubrics of causal inference [7, 115], econometrics [18, 35, 62, 75, 79, 92, 99], and dynamical systems [6, 15, 20, 63, 69]. More recently, researchers have been using partial identification methods to obtain

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

reliable off-policy evaluation in reinforcement learning [17, 39, 43, 44, 47, 48, 54, 67, 111, 113]. Despite these achievements, significant challenges still exist in applying partial identification for policy learning in complex and high-dimensional domains, including images and videos. We refer readers to App. A for a more detailed survey on partial identification and deep reinforcement learning.

This paper aims to address these challenges by investigating deep reinforcement learning algorithms from offline data over complex and high-dimensional domains, where the presence of unmeasured confounders could not be assumed away *a priori*. More specifically, our contributions are summarized as follows. (1) We introduce a novel DQN algorithm, which we call Causal DQN, capable of learning robust abstractions from confounded data over complex and high-dimensional domains with minimal prior knowledge. (2) We empirically demonstrate that our method significantly improves robustness and generalization under confounded observations and outperforms various DQN baselines across twelve popular Atari games. Due to space constraints, details of the experiment setup and additional experiments are provided in Apps. D and E. Videos of gameplay are included in the supplemental.

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

2 Challenges Due to Unobserved Confounders

We will focus on a sequential decision-making problem in the Markov Decision Process (MDP, [77]) where the agent intervenes on a sequence of actions to optimize subsequent rewards. Standard MDP models focus on the perspective of learners who could actively intervene in the environment. Consequently, confounding is generally assumed away a priori. On the other hand, when considering off-policy data collected from passive observations, the learner does not necessarily have the liberty to control how the behavioral policy generates the data, giving rise to unobserved confounders in decision-making tasks [26, 43, 51, 81, 114]. In this paper, we will consider a generalized family of confounded MDPs [14, 44, 54, 112, 113] explicitly modeling the presence of unobserved confounders in the off-policy data generation.

Definition 2.1. A Confounded Markov Decision Process (CMDP) $\mathcal{M}$ is a tuple of $\langle \mathcal{S}, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{F}, P \rangle$ where (1) $\mathcal{S}, \mathcal{X}, \mathcal{Y}$ are, respectively, the space of observed states, actions, and rewards; (2) $\mathcal{U}$ is the space of unobserved exogenous noise; (3) $\mathcal{F}$ is a set consisting of the transition function $f_S : \mathcal{S} \times \mathcal{X} \times \mathcal{U} \mapsto \mathcal{S}$, behavioral policy $f_X : \mathcal{S} \times \mathcal{U} \mapsto \mathcal{X}$, and reward function $f_Y : \mathcal{S} \times \mathcal{X} \times \mathcal{U} \mapsto \mathcal{Y}$; (4) P is an exogenous distribution over the domain $\mathcal{U}$.

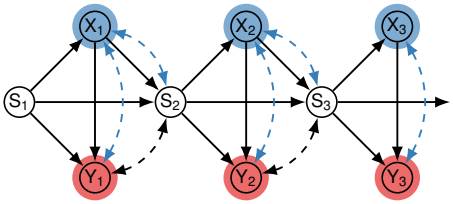


Figure 1: Causal diagram representing the data-generating mechanisms in a Confounded Markov Decision Process.

Throughout this paper, we will consistently assume the action domain $\mathcal{X}$ to be discrete and finite, while the state domain $\mathcal{S}$ could be complex and continuous; the reward domain $\mathcal{Y}$ is bounded in a real interval $[a, b] \subset \mathbb{R}$. Consider a demonstrator agent interacting with a CMDP $\mathcal{M}$, generating the off-policy data. For every time step $t = 1, \ldots, T$, the environment first draws an exogenous noise U_t from the distribution $P(\mathcal{U})$; the demonstrator then performs an action $X_t \leftarrow f_X(S_t, U_t)$, receives a subsequent reward $Y_t \leftarrow r_t(S_t, X_t, U_t)$, and moves to the next state $S_{t+1} \leftarrow f_S(S_t, X_t, U_t)$. The observed trajectories of the demonstrator (from the learner's perspective) are summarized as the observational distribution $P(\bar{\boldsymbol{X}}_{1:T}, \bar{\boldsymbol{S}}_{1:T}, \bar{\boldsymbol{Y}}_{1:T})$, i.e.,

$$P(\bar{\boldsymbol{x}}_{1:T}, \bar{\boldsymbol{s}}_{1:T}, \bar{\boldsymbol{y}}_{1:T}) = P(s_1) \prod_{t=1}^{T} \left(\int_{\mathcal{U}} \mathbf{1}_{s_{t+1}=f_S(s_t,x_t,u_t)} \mathbf{1}_{x_t=f_X(s_t,u_t)} \mathbf{1}_{y_h=f_Y(s_t,x_t,u_t)} P(u_t) \right)$$

Fig. 1 shows the causal diagram $\mathcal{G}$ [10] describing the generative process of the off-policy data in CMDPs. More specifically, solid nodes represent observed variables X_t, S_t, Y_t, and arrows represent the functional relationships f_X, f_S, f_Y among them. By convention, exogenous variables U_t are often not explicitly shown in the graph; bi-directed arrows $X_t \dashleftarrow\dashrightarrow Y_t$ and $X_t \dashleftarrow\dashrightarrow S_{t+1}$ indicate the presence of an unobserved confounder (UC) U_t affecting the action, state, and reward simultaneously. These bi-directed arrows (highlighted in blue) represent the unobserved confounders among action

X_t, reward Y_t, and state S_{t+1} in the off-policy data, violating the condition of NUC [11, 78]. Such violations could lead to challenges in off-policy learning.

Off-Policy Learning. A policy π in a CMDP $\mathcal{M}$ is a decision rule $\pi(x_t \mid s_t)$ mapping from state to a distribution over action domain $\mathcal{X}$. An intervention $\text{do}(\pi)$ is an operation that replaces the behavioral policy f_X in CMDP $\mathcal{M}$ with the policy π. Let $\mathcal{M}_\pi$ be the submodel induced by intervention $\text{do}(\pi)$. The interventional distribution $P_\pi(\bar{\boldsymbol{X}}_{1:T}, \bar{\boldsymbol{S}}_{1:T}, \bar{\boldsymbol{Y}}_{1:T})$ is defined as the joint distribution over observed variables in $\mathcal{M}_\pi$, i.e.,

$$P_\pi(\bar{\boldsymbol{x}}_{1:T}, \bar{\boldsymbol{s}}_{1:T}, \bar{\boldsymbol{y}}_{1:T}) = P(s_1) \prod_{t=1}^{T} \left(\pi(x_t \mid s_t) \mathcal{T}(s_t, x_t, s_{t+1}) \mathcal{R}(s_t, x_t, y_t) \right) \tag{1}$$

where the transition distribution $\mathcal{T}$ and the reward distribution $\mathcal{R}$ are given by, for $h = 1, \ldots, H$,

$$\mathcal{T}(s_t, x_t, s_{t+1}) = \int_{\mathcal{U}} \mathbf{1}_{s_{t+1} = f_S(s_t, x_t, u_t)} P(u_t), \quad \mathcal{R}(s_t, x_t, y_t) = \int_{\mathcal{U}} \mathbf{1}_{y_t = f_Y(s_t, x_t, u_t)} P(u_t) \tag{2}$$

For convenience, we write the reward function $\mathcal{R}(s, x)$ as the expected value $\sum_y y \mathcal{R}(s, x, y)$. Fix a discounted factor $\gamma \in [0, 1]$. A common objective for an agent is to optimize its cumulative return $R_t = \sum_{i=0}^{\infty} \gamma^i Y_{t+i}$. We define the optimal action-value function $Q_*(s, x)$ as the maximum expected return obtainable by following any policy π, after seeing a state s and taking an action x, $Q_*(s, x) = \max_\pi \mathbb{E}_{X_t \leftarrow x, \pi} [R_t \mid S_t = s]$. One could solve for an optimal policy by iteratively evaluating the action-value function using the *Bellman Optimality Equation* [12] given by,

$$Q_*(s, x) = \mathbb{E}_{X_t \leftarrow x} \left[Y_t + \gamma \max_{x'} Q_*(S_{t+1}, x') \mid S_t = s \right] \tag{3}$$

In off-policy evaluation, the agent (i.e., learner) attempts to learn an optimal policy by leveraging the observed data generated by a different behavior policy f_X (demonstrator). When there is no unmeasured confounder (NUC) introducing spurious correlations between action and subsequent outcomes, one could identify the parameterizations of the transition distribution $\mathcal{T}$ and reward function $\mathcal{R}$ from the observed data, i.e.,

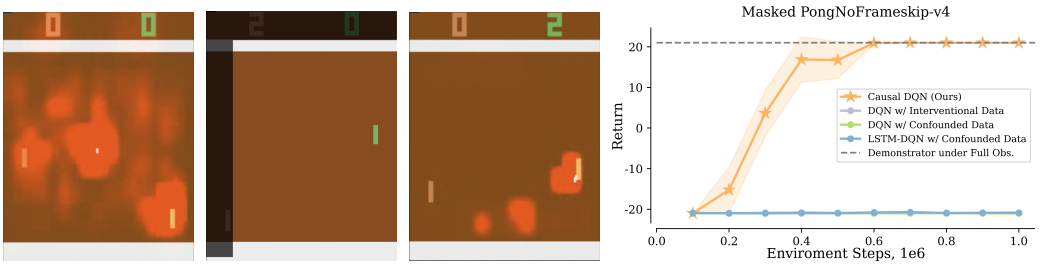
$$\mathcal{T}(s_t, x_t, s_{t+1}) = P\left(s_{t+1} \mid s_t, x_t\right), \qquad \mathcal{R}(s_t, x_t, y_t) = P\left(y_t \mid s_t, x_t\right) \tag{4}$$

When the above identification formula hold, several off-policy algorithms have been proposed to estimate the effect of candidate policies from finite observations [38, 41, 42, 65, 76, 94, 104, 105]. Together with the computational framework of deep learning, these methods could be further extended to complex domains [61, 70, 84, 85, 88]. However, NUC could be fragile in practice and does not necessarily hold due to some violations in the generative process. In these situations, applying standard off-policy methods may fail to converge to an optimal policy, despite using powerful deep learning models. The following example illustrates such challenges in a classic Atari game.

Example 1 (Confounded Pong)**.** Consider the Pong game in the classic Atari suite. As for the behavior policy, we use a pre-trained high-performing actor-critic agent [2] with residual blocks [29] and Long Short-Term Memory (LSTM, [32]) layers. Fig. 2a shows a saliency map visualizing the learned policy. Simulation results show that this policy is optimal. For example, this agent can deliver a "kill shot" by directing the ball to a location that the opponent is unlikely to intercept given its current location. We use this agent as the demonstrator in the off-policy learning task.

We now consider an alternative agent that learns to play Pong by observing the demonstrator's gameplay trajectories. This learning agent has a simpler neural network architecture and an impaired sensory capability: it can only observe movements in its nearby surroundings. Fig. 2b shows the learner's visual input; the board's left-hand side and the upper side, including the opponent's position and score, is now masked. In this case, the opponent's position becomes an unobserved confounder, introducing spurious correlations between the demonstrator's action and observed outcome. For example, the behavioral policy tends to hit the ball toward the center only when the opponent is positioned at either corner and unable to return it. As a result, center shots appear more effective than they truly are, due to confounding resulted from the masked opponent's position.

To validate whether the standard deep RL algorithms are robust to confounding biases, we train two DQN learners on masked trajectories from the demonstrator: one is the standard convolutional neural network based DQN (Nature DQN, [61]), the second one is an LSTM-based [28]. We also include a



(a) (b) (c) (d)

Figure 2: (a) A saliency map of the behavioral policy in Pong that tracks the opponent's location and score board; (b) a confounded Pong game where the opponent's location and score board is masked; (c) a saliency map of the conservative policy focusing on only itself and the ball; (d) the average return of our causal DQN and the standard DQN baselines. Baseline curves are overlapped.

standard DQN that learns directly in the masked Pong game without confounded demonstrations as a baseline. The simulation results, shown in Fig. 2d, indicate that none of those DQN variants is able to converge to an effective policy. Learning directly with impaired observation in Atari is challenging, while incorporating confounded demonstrations directly also does not enhance the convergence of DQN agents; instead, it negatively impacts their learning performance. ■

3 Confounding Robust Deep Q-Learning

In this section, we will introduce partial identification methods for off-policy learning that are robust to unobserved confounding. Recently, Zhang & Bareinboim [113] extended the well-celebrated Bellman equation to allow one to lower bound the state-action value function $Q_\pi(s,x)$ with a closed-form solution $\underline{Q_\pi}(s,x)$, which can be consistently estimated from the confounded observations. We extend this result to obtain a lower bound for the optimal value function.

Proposition 3.1 (Causal Bellman Optimality Equation)**.** *For a CMDP environment $\mathcal{M}$ with reward signals $Y_t \in [a,b] \subseteq \mathbb{R}$, its optimal state-action value function $Q_*(s,x) \geq \underline{Q_*}(s,x)$ for any state-action pair $(s,x) \in \mathcal{S} \times \mathcal{X}$, where the lower bound $\underline{Q_*}(s,x)$ is given by as follows,*

$$\underline{Q_*}(s,x) = P(x \mid s)\left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s') \max_{x'} \underline{Q_*}(s',x')\right) \tag{5}$$

$$+P(\neg x \mid s)\left(a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s',x')\right) \tag{6}$$

where $P(x \mid s) = P(X_t = x \mid S_t = s)$ and $P(\neg x \mid s) = 1 - P(x \mid s)$; $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{R}}$ are nominal transition distribution and reward function computed from the observational distribution, i.e.,

$$\widetilde{\mathcal{T}}(s,x,s') = P(S_{t+1} = s' \mid S_t = s, X_t = x), \qquad \widetilde{\mathcal{R}}(s,x) = \mathbb{E}[Y_t \mid S_t = s, X_t = x] \tag{7}$$

Prop. 3.1 lower bounds the expected return of an optimal policy π^* using the return of a pessimistic policy $\underline{\pi^*}$ that optimizes a worst-case CMDP instance $\underline{\mathcal{M}}$ compatible with the observational data. The lower bound is set as the expected return of the pessimistic policy $\underline{\pi^*}$ in the worst-case CMDP $\underline{\mathcal{M}}$, i.e., $\underline{Q_*}(s,x) \triangleq Q_{\underline{\pi^*}}(s,x;\underline{\mathcal{M}})$. Since π^* is optimal in the ground-truth CMDP environment $\mathcal{M}$, we must have $Q_*(s,x;\mathcal{M}) \geq Q_{\underline{\pi^*}}(s,x;\mathcal{M}) \geq Q_{\underline{\pi^*}}(s,x;\underline{\mathcal{M}})$. Optimizing the lower bound in Prop. 3.1 leads to a pessimistic policy with a performance guarantee in the ground-truth environment. Among quantities in the lower bound Prop. 3.1, nominal transition distribution $\widetilde{\mathcal{T}}$ and nominal reward function $\widetilde{\mathcal{R}}$ are functions of the observational distribution, and, at least in principle, are consistently estimable from the sampling process. The lower bound $\underline{Q_*}(s,x)$ can thus be further written as:

$$\underline{Q_*}(s,x) = \mathbb{E}\left[\mathbf{1}_{X_t=x}\left(Y_t + \max_{x'} \underline{Q_*}(S_{t+1},x')\right) + \mathbf{1}_{X_t\neq x}\left(a + \min_{s'} \max_{x'} \underline{Q_*}(s',x')\right) \mid S_t = s\right] \tag{8}$$

Algorithm 1 Causal Deep Q-Learning (`Causal-DQN`)

Initialize replay memory $\mathcal{D}$
Initialize action-value function $\underline{Q_*}(\cdot;\theta)$ with random weights θ
for episodes $= 1, \ldots, M$ **do**
  Sample initial state s_1
  for $t = 1, \ldots, T$ **do**
    Observe an action x_t taken by the demonstrator and subsequent reward y_t and state s_{t+1}
    Store transition (s_t, x_t, y_t, s_{t+1}) in $\mathcal{D}$
    Sample a minibatch of transitions $\{(s_i, x_i, y_i, s_{i+1})\}_{i=1}^{B}$ from $\mathcal{D}$
    Set value target $w_i(x)$ for every action $x \in \mathcal{X}$ w.r.t sample (s_i, x_i, y_i, s_{i+1}),

$$w_i(x) = \begin{cases} y_i + \gamma \max_{x'} \underline{Q_*}(s_{i+1}, x'; \theta) & \text{if } x = x_i \\ a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s', x'; \theta) & \text{if } x \neq x_i \end{cases} \tag{11}$$

    Perform a gradient descent step on $\sum_x \left(w_i(x) - \underline{Q_*}(s_i, x; \theta)\right)^2$ according to Eq. (10)
  end for
end for

In the above equation, Y_t and S_t are observed variables drawn from the nominal reward function $\widetilde{\mathcal{R}}$ and transition distribution $\widetilde{\mathcal{T}}$ in Eq. (7), respectively. Fig. 3 shows a backup diagram illustrating this update step. Like the standard Bellman optimality equation (Eq. (3)), Eq. (8) recursively updates the value function based on the current estimates of the optimal value function. On the other hand, Eq. (8) explicitly accounts for the off-poicy nature of the confounded observations: when the behavior policy takes the same action $x_t = x$ as the target action, the update follows standard Bellman equation and uses the next sampled state s_t; when the sampled action $x_t \neq x$ differs from the target, our algorithm updates, instead, using the value function associated with the next worst-case or best-case state s^*, corresponding to the estimation of the lower bound and upper bound respectively.

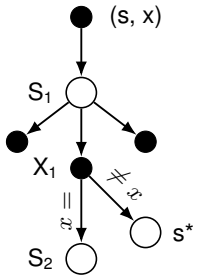


Figure 3: Backup diagram for causal deep Q-learning.

By using the causal Bellman equation of Eq. (8) as an iterative update, one could apply standard value iteration to obtain a robust policy against the confounding bias in the off-policy data [113]. In practice, however, this approach could be computationally challenging for complex and high-dimensional domains. Like many deep reinforcement learning algorithms [61], we will use a neural network with weights θ, called Q-network, to approximate the lower bound over the state-action value function, i.e., $\underline{Q_*}(s, x; \theta) \approx \underline{Q_*}(s, x)$. We will train a Q-network by minimizing a sequence of loss functions $L_i(\theta_i)$ at each iteration i. Formally,

$$L_i(\theta_i) = \mathbb{E}_{s \sim \rho(\cdot)} \left[\sum_x \left(W_i(x) - \underline{Q_*}(s, x; \theta_i)\right)^2 \right] \tag{9}$$

where function $W_i(x)$ is defined as the right-hand side of the update procedure Eq. (8); and $\rho(s)$ is the state occupancy distribution in the observed Markov chain under the behavioral policy. The parameters from the previous iteration θ_{i-1} are held fixed when optimizing the loss function. Note that the above loss function attempts to minimize the error of the Q-network bound over all actions. The reason is that, in the causal Bellman update (Eq. (8)), the next observed action contains information about the lower bound across all actions, regardless of whether it matches the actual action taken. Differentiating the loss function with respect to the weights, we arrive at the following gradient,

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s \sim \rho(\cdot)} \left[\mathbb{E} \left[\sum_x \left(W_t(x) - \underline{Q_*}(s, x; \theta_i)\right) \nabla_{\theta_i} \underline{Q_*}(s, x; \theta_i) \mid S_t = s \right] \right] \tag{10}$$

Details of our proposed algorithm, called Causal Deep Q-Learning (`Causal-DQN`), are provided in Algo. 1. Like the standard DQN [61], our algorithm utilizes experience replay [57]. Particularly, it stores trajectories observed at each time step, represented as (s_t, x_t, y_t, s_{t+1}), in a replay memory $\mathcal{D}$ that is pooled from many episodes. During the inner loop of the algorithm, we apply minibatch stochastic gradient descent to samples of experience $(s_i, x_i, y_i, s_{i+1}) \sim \mathcal{D}$, which are randomly

drawn from the pool of stored samples. On the other hand, our proposed causal algorithm made the following augmentations compared to the standard DQN. First, `Causal-DQN` is an off-policy learning algorithm and does not actively intervene in the environment. At step 6, instead of exploiting the Q-network being trained, it queries the demonstrator to generate a confounded transition sample. Second, at Step 9 during the experience replay, Causal-DQN utilizes the causal Q-learning updates of Eq. (10), which is robust to the potential presence of confounders. Particularly, when the observed action x_i is equal to the evaluated action x, the algorithm follows the standard Q-learning update. Otherwise, it performs the update using a lower-bound a over the immediate reward and the value function at the worst-case next state s'. The worst-case state s' is empirically estimated by repeatedly sampling the next possible states at random, and taking the one with the smallest value function estimate. These augmentations improve Causal-DQN over its non-causal counterpart in terms of robustness and sample efficiency, as it is able to utilize the abundant observational data to improve the evaluation of the state-action value function.

Example 2 (Confounded Pong continued)**.** Consider again the confounded Pong game described in Example 1. We train a Causal DQN agent with the masked observed trajectories. Fig. 2c shows the saliency map visualizing the learned policy. Our proposed method learns a conservative policy focusing on only tracking the ball location instead of opponent's location. Simulation results show that this conservative policy is able to achieve comparable performance to the optimal demonstrator using the full board information. Analyzing the gameplay video reveals that our causal DQN agent learns to proactively place the ball in either corner, where the hard-coded AI opponent is unable to return. See the gameplay video in the supplementary materials for more details. ■

4 Experiments

In this section, we aim to demonstrate the robustness and performance improvement of our proposed `Causal-DQN` under confounded settings. For a comprehensive evaluation of `Causal-DQN`, we choose twelve popular Atari games from the Gymnasium benchmark [100] and design the corresponding confounded versions. See below and also App. C for our detailed design of confounded Atari games. For a fair comparison, we also use vanilla DQN with little modifications as the baselines we test. More specifically, the baselines include (1) a CNN-based DQN with confounded demonstrator data (Conf. DQN), (2) an LSTM-based DQN with confounded demonstrator data (Conf. LSTM-DQN), and (3) a CNN-based DQN trained directly under masked observations without confounded data (Interv. DQN). For (1-2), the DQN agent will query the demonstrator for data samples as the `Causal-DQN` does. For (3), the DQN agent uses its own policy to sample environment transitions.

For each game, we train the agent for 1 million environment steps. We use 20 parallel environments to collect samples. At each parallel environment step, a minibatch is sampled to train the agents, equivalent to an update frequency of 20. We use a batch size of 512, a replay buffer of 100K in size, and a learning rate of $5\mathrm{e}{-4}$ to accelerate convergence. Other hyperparameters are the same as in [61]. All results presented in this section are evaluation performances where we test each trained agent in the Atari game with masked observations. Curves in Fig. 8 are generated by evaluating the agent periodically in a separate evaluation environment, not from training returns.

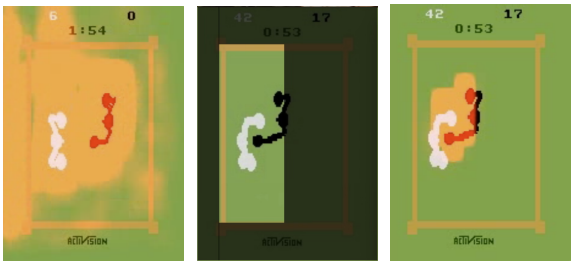
Data Preparation and Model Architecture. We use the standard Atari game preprocessing for the input to the agents except that we resize the input to be 64×64 to align with the size requirement of the demonstrator [2], a competitive actor-critic agent with deep residual blocks and LSTM layers. Other differences in input preprocessing for the demonstrator are that (1) the demonstrator takes a single colored frame as the input per each time step, and (2) the demonstrator has access to the original full screen observation while the learners only has access to a masked partial screen. For all DQN tested, we adopt Double DQN [101] to stabilize learning. The CNN version follows the set up of Nature DQN [61] and the LSTM version only replaces the second-to-last linear layer with an LSTM cell. See App. D for detailed model architectures.

Designing the Confounded Atari Games. In the confounded Atari games, we mask out certain areas in each game's observation to prevent the agent from using spurious correlated features. To find such spurious visual artifacts used by the demonstrators, we apply a perturbation-based approach [25] to visualize saliency maps of both the actor and the critic of the demonstrator [2]. As shown in Fig. 2a, in the Pong game, the demonstrator is constantly checking the score board and also the opponent's paddle locations, neither of which is necessary for winning since the opponent in Pong has a fixed policy regardless of the current score and as long as the agent shoot back the ball, there is a

chance to score. Thus, an intuitive optimal policy should only look at the ball location and the agent's own paddle location to decide the move. In Fig. 2b, we mask out those areas and use the masked out observation as the state input (s_t) to DQNs while the masked area becomes the confounder u_t which can only be observed by the demonstrator's policy, i.e., $x_t \leftarrow f_X(s_t, u_t)$.

For the remainder of this section, we will first present a few other notable confounded Atari games and discuss the performance of our proposed `Causal-DQN` in all 12 confounded Atari games. Specifically, despite confounding bias, our proposed causal agent is able to obtain an effective policy under masked observations from the demonstrator's trajectories. The learned policies demonstrate conservative behaviors aligned with human intuitions. Overall, `Causal-DQN` consistently dominates its non-causal vanilla counterparts in all 12 confounded Atari games in performance.

Confounded Boxing. In the original Boxing, the player controls the white agent to punch the black one to score. The party with the higher score when the time runs out, or any party hitting 100 first, wins the game. The demonstrator's policy picks up an aggressive "brawler" style which pressures the opponent and trades blows in the center of the arena. Fig. 4a shows the saliency map of the demonstrator's policy.

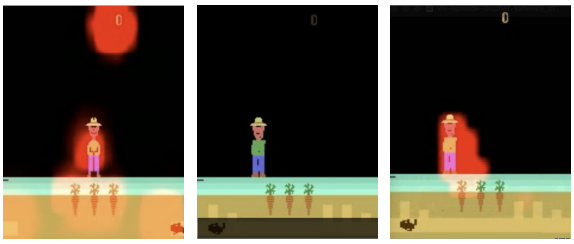


(a) (b) (c)

Figure 4: (a) A saliency map of the demonstrator's policy; (b) a confounded Boxing game where only the left half of the arena is visible; (c) a saliency map of `Causal-DQN`'s policy.

In the confounded Boxing, we mask out the score/remaining time, the outer area of the arena, and the right half of the arena. In words, the agent has impaired eyesight and cannot keep track of the current score and remaining time. Our `Causal-DQN` agent picks up a conservative "rope-the-dope" boxing style which focuses on defending its ground on the left-hand side of the arena. Fig. 4c shows the saliency map of such a conservative policy. Perhaps surprisingly, simulation results, shown in Table 1 and Fig. 8, reveal that despite the limited sensory capabilities, `Causal-DQN` agent can still achieve similar performance as the optimal demonstrator, defeating the hard-coded AI opponent.

Confounded Gopher. In the Gopher game, the player controls a farmer with a shovel, tasked with protecting a garden of carrots from a mischievous gopher. The gopher repeatedly attempts to tunnel underground to steal the carrots. The player must move horizontally across the screen to block the gopher's digging attempts by filling holes. A shortcut strategy is to follow the gopher's location underground closely so that the farmer is always close to those newly completed holes. Fig. 5a shows the saliency map of the demonstrator's policy. Our analysis reveals that it manages to pick up a "proactive" playing strategy, which actively tracks the gopher's location and uses this information to adjust the farmer's position.



(a) (b) (c)

Figure 5: (a) A saliency map of the demonstrator's policy; (b) a confounded Gopher game where the tunnel and score are masked; (c) a saliency map of `Causal-DQN`'s policy.

On the other hand, in the confounded Gopher game, the gophers' locations are now masked, and the agent no longer follows the same "proactive" strategy. Instead, our `Causal-DQN` picks up an alternative "reactive" strategy, which will reset the farmer's position around the center and only move when a gopher is digging out of the ground. We evaluate the learner's performance and provide them in Table 1 and Fig. 8. Interestingly, simulation results show that the "reactive" strategy is more effective than a "proactive" one, and `Causal-DQN` is able to outperform the demonstrator's policy.

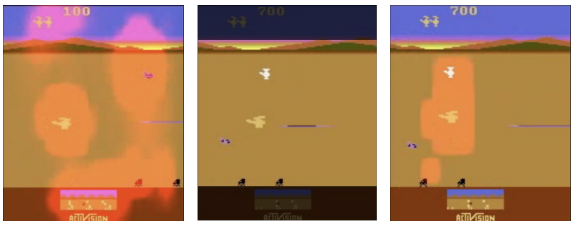
Confounded ChopperCommand. In the ChopperCommand game, the agent controls a helicopter tasked with defending a convoy of trucks from waves of enemy aircraft and helicopters. The agent must navigate across the desert landscape, shooting down enemies while avoiding incoming fire. Successfully protecting the convoy and eliminating threats increases the player's score, while

Table 1: Average evaluation returns of agents on the 12 confounded Atari games trained with 1M environment steps and aggregated normalized returns concerning the demonstrator's performance. Bold numbers indicate the best-performing methods. All results are averaged over 5 seeds except that column Random is from [2]. `Causal-DQN` significantly outperforms other DQN baselines.

Game	Demonstrator	Random	Interv. DQN	Conf. DQN	Conf. LSTM-DQN	`Causal-DQN` **(ours)**
Amidar	232.4	5.8	44.0	37.8	59.0	**282.6**
Asterix	3080.6	210.0	650.0	429.0	479.0	**2587.0**
Boxing	89.0	0.1	-0.62	-9.8	-6.9	**71.5**
Breakout	219.2	1.7	2.2	1.2	4.9	**131.2**
ChopperCommand	1280.0	811.0	1192.0	1076.0	1116.0	**1658.0**
Gopher	5480.6	257.6	288.8	752.0	485.6	**7327.2**
KungFuMaster	35400.0	258.5	12416.0	13674.0	6526.0	**44196.0**
MsPacman	2316.8	307.3	1191.6	881.8	787.4	**1747.6**
Pong	20.8	-20.7	-20.8	-20.8	-20.4	**21.0**
Qbert	4420.6	163.9	322.5	208.5	253.5	**4458.5**
RoadRunner	16560.6	11.5	1154.0	1168.0	484.0	**27414.0**
Seaquest	1412.4	68.4	237.2	281.6	164.8	**980.0**
Normalized Mean (↑)	1.00	0.00	0.13	0.10	0.09	**1.04**
Normalized Median (↑)	1.00	0.03	0.13	0.14	0.10	**1.01**
Normalized IQM (↑)	1.00	0.03	0.13	0.13	0.11	**1.02**

allowing enemy fire to destroy the trucks results in lost points or lives. At the bottom of the screen, a mini-map/radar is showing incoming trucks and enemies. Fig. 5a shows the saliency map for the demonstrator's policy. It learns to utilize the radar information to "look ahead."

We next consider a confounded ChopperCommand game where the chopper loses its radar and other sensor devices. As a result, the mini-map area, current score, and remaining lives are masked (as shown in Fig. 6b). By applying `Causal-DQN`, the agent picks up a more "spontaneous" playing style, focusing on staying alive and eliminating opponents upfront. Fig. 6c describes a saliency map of this "spontaneous" policy where only nearby opponents are highlighted. Simulation results in Table 1 and Fig. 8 reveal that `Causal-DQN` outperforms other baselines significantly and even surpasses the demonstrator's policy despite having a simpler neural network architecture.



(a) (b) (c)

Figure 6: (a) A saliency map of the demonstrator's policy; (b) a confounded ChopperCommand game with the minimap and score/lives masked; (c) a saliency map of `Causal-DQN`'s policy.

Overall Performance. Table 1 provides the best mean returns for all 12 confounded Atari games across trials, along with mean return normalized by demonstrator's performance and normalized interquantile mean (IQM). We see `Causal-DQN` consistently outperforming other non-causal baselines by a big margin. In 7/12 games, our proposed method even surpasses the demonstrator with full observations and a way more complex architecture [2]. This could be due to the reason that the demonstrator, though powerful in representation learning, may suffer from observational overfitting [90] and rely on spurious visual features to make decisions. Both prior work and our work have empirically verified this by using saliency maps [25]. Masking out those spurious features indeed poses a non-trivial challenge to the DQN. From Table 1, we see that the vanilla DQN with interventional data under masked observations (Interv. DQN) hardly achieves any meaningful scores in 1 million environment steps. Even with the help of LSTM cells or confounded data from a performing demonstrator, the DQN still cannot learn. Only with the causal bound, the same architecture (Nature DQN) can recover or even surpass the demonstrator's performance despite using masked observations and a shallow, small CNN as the feature extractor. In Fig. 7, we also

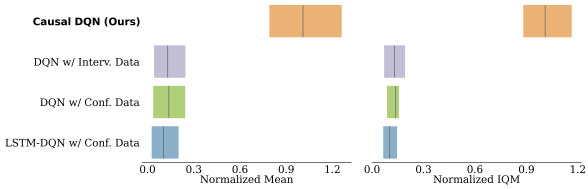


Figure 7: Normalized mean and normalized IQM scores. `Causal-DQN` achieves a normalized mean return of 1.04 and a normalized IQM of 1.02.

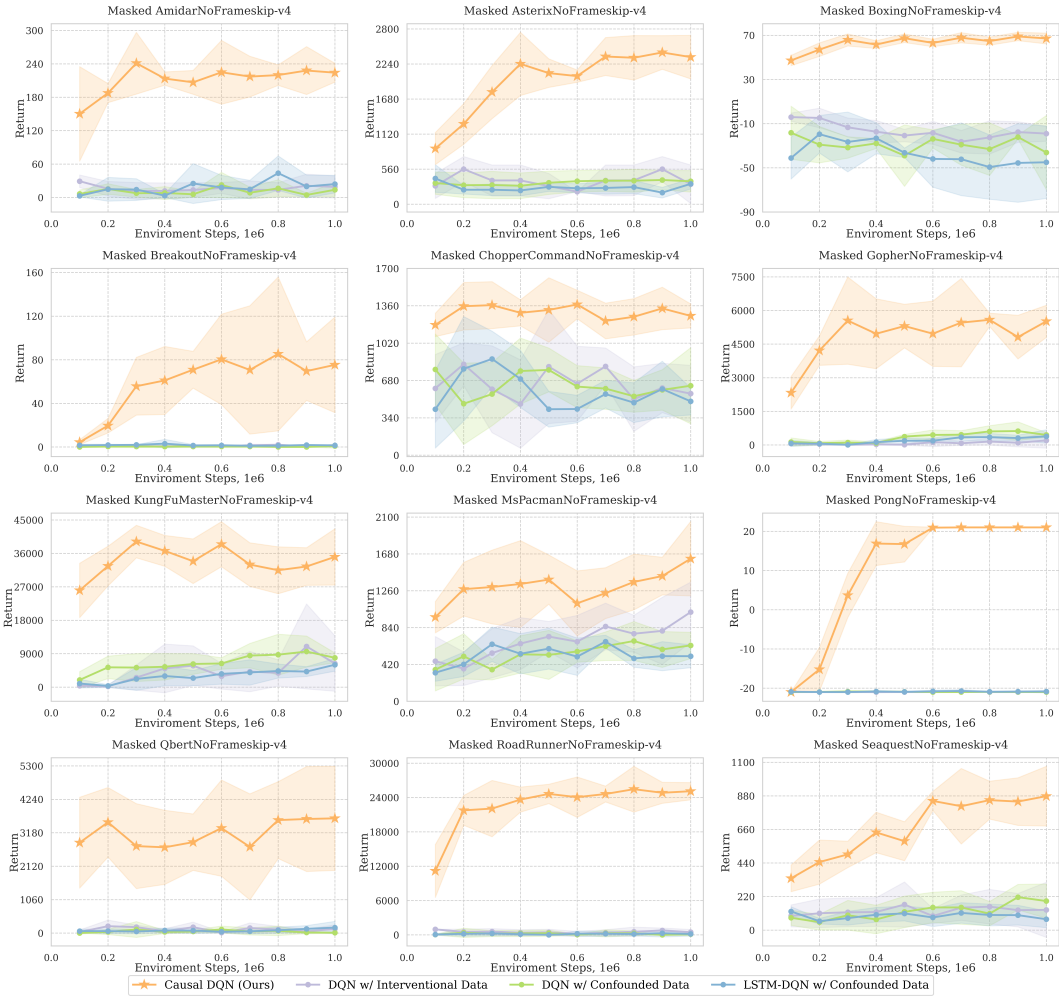


Figure 8: Average evaluation performance for 12 confounded Atari games. During training over the 1M environment steps, we evaluate the agent every 100K steps for 10 episodes each time. The curve is further averaged over 5 seeds with one standard deviation across trials as the shaded area.

provide stratified bootstrap confidence intervals for the normalized mean and normalized IQM scores as recommended by Agarwal et al. [1]. We can see clearly that our proposed `Causal-DQN` can recover the expert demonstrator's performance even under impaired sensors. And to get a sense of the sample efficiency of `Causal-DQN`, in Fig. 8, we report the evaluation performance during training of all 12 confounded Atari games. `Causal-DQN` converges uniformly in less than 1 million steps in all games. See App. E for more results.

Remark on the failure of DQN baselines. We have conducted thorough hyper-parameter tuning and used the recommended hyper-parameter settings from the original DQN work [61]. Due to limited time and computational resources, we are only able to finish 1M steps of training for all the environments and seeds. However, we do notice that the interventional DQN baseline can gradually learn the right policy with longer training time. The learning curve usually starts to grow after 3M steps. This corroborates the validity of our confounded environment design, that the learning task is now harder to solve, but not totally impossible. With the help of a causally aware learner (`Causal-DQN`), one would be able to learn more sample-efficiently (3x fewer steps in our experiments) than the pure interventional online regime or the biased causally unaware offline learners.

5 Conclusions

This paper investigates deep reinforcement learning from off-policy data collected by a different behavior policy through a causal lens. Particularly, we focus on a generalized setting where confounding biases cannot be ruled out *a priori*, which poses significant challenges to standard off-policy evaluation algorithms. We first extend the celebrated Bellman equation to the causal Bellman equation that lower bounds the agent's expected return from confounded observations. Building on this extension, we then propose a novel `Causal-DQN` algorithm that could obtain an effective policy from off-policy data even when unobserved confounders generally exist. Finally, we evaluate our proposed algorithm in twelve confounded Atari games, showing that the causal approach consistently dominates the standard DQN algorithm with different feature extractors or data sources.

Yet the implications of this work extend far beyond discrete control benchmarks. Unobserved confounding is not an anomaly but a pervasive property of real-world RL. It lurks beneath virtually all forms of observational or off-policy data, from robotic demonstrations to human feedback, silently distorting the mapping between actions, rewards, and outcomes. In an era when the field is increasingly guided by the scaling law, the belief that enlarging models and datasets will automatically yield better intelligence, our findings reveal a critical blind spot: scaling on confounded data does not scale performance. In fact, it may amplify biases, producing policies that are efficient yet misaligned.

Consider large-scale robotic pretraining from internet videos: behavioral policies (human or robot demonstrators) differ sharply from the learner's policy space, introducing systematic unobserved confounding between observations and intended actions. Similarly, in reinforcement learning from human feedback (RLHF) for aligning large language models (LLMs), preference datasets are deeply entangled with hidden factors like emotions, social norms, cultural context, temporal inconsistency, and individual baselines that no simple text prompt can fully encode. Without a causal understanding of these factors, LLM alignment merely approximates correlations within preference and prompts, not their causal origins. The same problem is exacerbated for healthcare, finance, and any domain where observational data conceals the determining forces driving decisions.

Toward the future, we envision causal reinforcement learning as a foundation for building confounding-robust, safely-aligned, and generalizable agents. Extending the causal Bellman framework to policy-gradient methods, continuous control, RLHF, and multi-agent systems will be crucial steps toward this goal. Ultimately, bridging causal inference and deep RL offers more than robustness. It paves the way for agents that reason about interventions and consequences, rather than merely fitting experience. In sum, this work marks an early yet essential stride toward causally grounded agents, a future where RL agents learn not only what to do, but why it works.

Acknowledgments

This research is supported in part by the NSF, ONR, AFOSR, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation.

References

[1] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[2] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[3] M. Andrychowicz, D. Crow, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5048–5058, 2017.

[4] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. D. Guo, and C. Blundell. Agent57: Outperforming the Atari Human Benchmark. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR, 2020.

[5] K. Badrinath and D. Kalathil. Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *Proceedings of the 38th International Conference on Machine Learning*, pages 465–474, 2021.

[6] P. Bajari, C. L. Benkard, and J. Levin. Estimating dynamic models of imperfect competition. *Econometrica*, 75(5):1331–1370, 2007.

[7] A. Balke and J. Pearl. Bounds on treatment effects from studies with imperfect compliance. *Journal of the American Statistical Association*, 92(439):1172–1176, September 1997.

[8] P. J. Ball, L. M. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 1577–1594. PMLR, 2023.

[9] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

[10] E. Bareinboim, J. D. Correa, D. Ibeling, and T. Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, pages 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.

[11] E. Bareinboim, J. Zhang, and S. Lee. An introduction to causal reinforcement learning. Technical Report R-65, Causal Artificial Intelligence Lab, Columbia University, December 2024.

[12] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[13] A. Bennett and N. Kallus. Proximal reinforcement learning: Efficient off-policy evaluation in partially observed markov decision processes. *Oper. Res.*, 72(3):1071–1086, 2024.

[14] A. Bennett, N. Kallus, L. Li, and A. Mousavi. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. In *International Conference on Artificial Intelligence and Statistics*, pages 1999–2007. PMLR, 2021.

[15] S. T. Berry and G. Compiani. An instrumental variable approach to dynamic models. *The Review of Economic Studies*, 90(4):1724–1758, 2023.

[16] P. Bhargava, R. Chitnis, A. Geramifard, S. Sodhani, and A. Zhang. Sequence modeling is a robust contender for offline reinforcement learning. arxiv, 2023.

[17] D. Bruns-Smith and A. Zhou. Robust fitted-q-evaluation and iteration under sequentially exogenous unobserved confounders. *arXiv preprint arXiv:2302.00662*, 2023.

[18] F. A. Bugni. Bootstrap inference in partially identified models defined by moment inequalities: Coverage of the identified set. *Econometrica*, 78(2):735–753, 2010.

[19] Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[20] M. J. Dickstein and E. Morales. What do exporters know? *The Quarterly Journal of Economics*, 133(4):1753–1801, 2018.

[21] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1406–1415. PMLR, 2018.

[22] T. A. for Computing Machinery. Acm a.m. turing award honors two researchers who led the development of cornerstone ai technology. https://awards.acm.org/about/2024-turing. Accessed: 2025-05-15.

[23] M. Ghavamzadeh, M. Petrik, and Y. Chow. Safe policy improvement by minimizing robust baseline regret. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2298–2306, 2016.

[24] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[25] S. Greydanus, A. Koul, J. Dodge, and A. Fern. Visualizing and understanding atari agents. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1787–1796. PMLR, 2018.

[26] H. Guo, Q. Cai, Y. Zhang, Z. Yang, and Z. Wang. Provably efficient offline reinforcement learning for partially observable markov decision processes. In *International Conference on Machine Learning*, pages 8016–8038. PMLR, 2022.

[27] H. Guo, Q. Cai, Y. Zhang, Z. Yang, and Z. Wang. Provably efficient offline reinforcement learning for partially observable markov decision processes. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8016–8038. PMLR, 2022.

[28] M. J. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015*, pages 29–37. AAAI Press, 2015.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[30] M. Hessel, M. Kroiss, A. Clark, I. Kemaev, J. Quan, T. Keck, F. Viola, and H. van Hasselt. Podracer architectures for scalable reinforcement learning. *CoRR*, abs/2104.06272, 2021.

[31] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3215–3222. AAAI Press, 2018.

[32] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[33] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

[34] G. W. Imbens and J. D. Angrist. Identification and estimation of local average treatment effects. *Econometrica*, 62(2):467–475, 1994.

[35] G. W. Imbens and D. B. Rubin. Bayesian inference for causal effects in randomized experiments with noncompliance. *The annals of statistics*, pages 305–327, 1997.

[36] G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[37] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[38] N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 652–661, New York, New York, USA, 20–22 Jun 2016. PMLR.

[39] S. Joshi, J. Zhang, and E. Bareinboim. Towards safe policy learning under partial identifiability: A causal approach. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 13004–13012. AAAI Press, 2024.

[40] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[41] Y. Jung, J. Tian, and E. Bareinboim. Learning causal effects via weighted empirical risk minimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[42] Y. Jung, J. Tian, and E. Bareinboim. Estimating identifiable causal effects through double machine learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12113–12122. AAAI Press, 2021.

[43] N. Kallus and A. Zhou. Confounding-robust policy improvement. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9289–9299, 2018.

[44] N. Kallus and A. Zhou. Confounding-robust policy evaluation in infinite-horizon reinforcement learning. *Advances in neural information processing systems*, 33:22293–22304, 2020.

[45] B. Kang, Z. Jie, and J. Feng. Policy optimization with demonstrations. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2474–2483. PMLR, 2018.

[46] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney. Recurrent experience replay in distributed reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[47] C. Kausik, Y. Lu, K. Tan, M. Makar, Y. Wang, and A. Tewari. Offline policy evaluation and optimization under confounding. In *International Conference on Artificial Intelligence and Statistics*, pages 1459–1467. PMLR, 2024.

[48] S. Khan, M. Saveski, and J. Ugander. Off-policy evaluation beyond overlap: partial identification through smoothness. *arXiv preprint arXiv:2305.11812*, 2023.

[49] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. K. Yogamani, and P. Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.*, 23(6):4909–4926, 2022.

[50] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[51] D. Kumor, J. Zhang, and E. Bareinboim. Sequential causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems*, 2021.

[52] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[53] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[54] M. Li, J. Zhang, and E. Bareinboim. Automatic reward shaping from confounded offline data. In *Forty-second International Conference on Machine Learning*, 2025.

[55] S. H. Lim, H. Xu, and S. Mannor. Reinforcement learning in robust markov decision processes. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 701–709, 2013.

[56] S. H. Lim, H. Xu, and S. Mannor. Reinforcement learning in robust markov decision processes. *Math. Oper. Res.*, 41(4):1325–1353, 2016.

[57] L.-J. Lin. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992.

[58] M. Lu, Y. Min, Z. Wang, and Z. Yang. 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*. OpenReview.net, 2023.

[59] C. F. Manski. Nonparametric bounds on treatment effects. *The American Economic Review*, 80:319–323, 1989.

[60] R. Miao, Z. Qi, and X. Zhang. Off-policy evaluation for episodic partially observable markov decision processes under non-parametric models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *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*, 2022.

[61] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[62] H. R. Moon and F. Schorfheide. Bayesian and frequentist inference in partially identified models. *Econometrica*, 80(2):755–782, 2012.

[63] E. Morales, G. Sheu, and A. Zahler. Extended gravity. *The Review of economic studies*, 86(6):2668–2712, 2019.

[64] M. Moravčík, M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

[65] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

[66] M. Nakamoto, S. Zhai, A. Singh, M. S. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine. Cal-ql: Calibrated offline RL pre-training for efficient online fine-tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *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*, 2023.

[67] H. Namkoong, R. Keramati, S. Yadlowsky, and E. Brunskill. Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33:18819–18831, 2020.

[68] A. Nilim and L. El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[69] A. Norets and X. Tang. Semiparametric inference in dynamic binary choice models. *Review of Economic Studies*, 81(3):1229–1262, 2014.

[70] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand. arxiv, 2019.

[71] K. Panaganti, Z. Xu, D. Kalathil, and M. Ghavamzadeh. Robust reinforcement learning using offline data. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[72] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

[73] M. Petrik and R. H. Russel. Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7047–7056, 2019.

[74] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial reinforcement learning. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2817–2826. PMLR, 2017.

[75] D. J. Poirier. Revising beliefs in nonidentified models. *Econometric theory*, 14(4):483–509, 1998.

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

[77] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1 edition, 1994.

[78] H. Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

[79] J. P. Romano and A. M. Shaikh. Inference for identifiable parameters in partially identified econometric models. *Journal of Statistical Planning and Inference*, 138(9):2786–2807, 2008.

[80] A. Roy, H. Xu, and S. Pokutta. Reinforcement learning under model mismatch. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3043–3052, 2017.

[81] K. Ruan, J. Zhang, X. Di, and E. Bareinboim. Causal imitation for markov decision processes: A partial identification approach. *Advances in Neural Information Processing Systems*, 37:87592–87620, 2024.

[82] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[83] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[84] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[85] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arxiv, 2017.

[86] M. Schwarzer, J. S. Obando-Ceron, A. C. Courville, M. G. Bellemare, R. Agarwal, and P. S. Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 30365–30380. PMLR, 2023.

[87] C. Shi, M. Uehara, J. Huang, and N. Jiang. A minimax learning approach to off-policy evaluation in confounded partially observable markov decision processes. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 20057–20094. PMLR, 2022.

[88] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. In J. Tan, M. Toussaint, and K. Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 642–660. PMLR, 2023.

[89] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[90] X. Song, Y. Jiang, S. Tu, Y. Du, and B. Neyshabur. Observational overfitting in reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[91] Y. Song, Y. Zhou, A. Sekhari, D. Bagnell, A. Krishnamurthy, and W. Sun. Hybrid RL: using both offline and online data can make RL efficient. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[92] J. Stoye. More on confidence intervals for partially identified parameters. *Econometrica*, 77(4):1299–1315, 2009.

[93] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, second edition, 2018.

[94] A. Swaminathan and T. Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823, 2015.

[95] A. Tamar, S. Mannor, and H. Xu. Scaling up robust mdps using function approximation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 181–189. JMLR.org, 2014.

[96] E. J. T. Tchetgen, A. Ying, Y. Cui, X. Shi, and W. Miao. An introduction to proximal causal learning, 2020.

[97] P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence policy improvement. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2380–2388, Lille, France, 07–09 Jul 2015. PMLR.

[98] J. Tian and J. Pearl. Probabilities of causation: Bounds and identification. *Ann. Math. Artif. Intell.*, 28(1-4):287–313, 2000.

[99] D. Todem, J. Fine, and L. Peng. A global sensitivity test for evaluating statistical hypotheses with nonidentifiable models. *Biometrics*, 66(2):558–566, 2010.

[100] M. Towers, A. Kwiatkowski, J. K. Terry, J. U. Balis, G. D. Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *CoRR*, abs/2407.17032, 2024.

[101] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In D. Schuurmans and M. 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.

[102] Y. Wang and S. Zou. Online robust reinforcement learning with model uncertainty. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7193–7206, 2021.

[103] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas. Dueling network architectures for deep reinforcement learning. In M. Balcan and K. Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1995–2003. JMLR.org, 2016.

[104] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[105] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge England, 1989.

[106] C. J. C. H. Watkins and P. Dayan. Technical note q-learning. *Mach. Learn.*, 8:279–292, 1992.

[107] W. Wiesemann, D. Kuhn, and B. Rustem. Robust markov decision processes. *Math. Oper. Res.*, 38(1):153–183, 2013.

[108] H. Xu and S. Mannor. Distributionally robust markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

[109] P. Yu and H. Xu. Distributionally robust counterpart in markov decision processes. *IEEE Transactions on Automatic Control*, 61(9):2538–2543, 2016.

[110] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. S. Boning, and C. Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[111] J. Zhang and E. Bareinboim. Near-optimal reinforcement learning in dynamic treatment regimes. In *Advances in Neural Information Processing Systems*, pages 13401–13411, 2019.

[112] J. Zhang and E. Bareinboim. Can humans be out of the loop? In *Conference on Causal Learning and Reasoning*, pages 1010–1025. PMLR, 2022.

[113] J. Zhang and E. Bareinboim. Eligibility traces for confounding robust off-policy evaluation: A causal approach. In *41st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2025.

[114] J. Zhang, D. Kumor, and E. Bareinboim. Causal imitation learning with unobserved confounders. *Advances in neural information processing systems*, 33:12263–12274, 2020.

[115] J. Zhang, J. Tian, and E. Bareinboim. Partial counterfactual identification from observational and experimental data. In *International Conference on Machine Learning*, pages 26548–26558. PMLR, 2022.

[116] Y. Zhu, Z. Wang, J. Merel, A. A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess. Reinforcement and imitation learning for diverse visuomotor skills. In H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims are supported by both theoretical proofs (Prop. 3.1) and empirical results on 12 confounded Atari games reported in Sec. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in a separate section in the appendix (App. G) and shed light on possible future works in the conclusion section (Sec. 5). For the experiment, we entail the computational requirements in App. D and the environment setup, data preparation in both Sec. 4 and App. D.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The detailed assumptions and proof setups are in Sec. 2 and Sec. 3. The proof detail for Prop. 3.1 is in App. B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all the necessary details for reproducing our work including preparing confounded Atari games, neural network architectures, training hyper-parameters and pseudo-code in Algo. 1 (and a more refined version Algo. 2) in both Sec. 4 and App. D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our experiments are based on a open benchmark (Atari from Gymnasium environments). And the detailed parameters and setups for generating confounded Atari games are also reported in Sec. 4. The demonstrator generating the confounded data is also an open sourced model, see https://github.com/eloialonso/diamond.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the full details of training/evaluation pipeline including model architectures, hyper-parameters, data preparation and evaluation metrics in Sec. 4 and App. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We follow the recommended evaluation protocol in [1]. We provide both mean and IQM on the returns (Table 1, Fig. 7). In Fig. 8, we also report standard deviation as the shaded area around curves.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computational resources used in each experiment in App. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not foresee any harms caused by the research process since no human subjects are involved in our work, and the environments we use are open-sourced Atari games. We also do not foresee any direct harmful societal impact of our work.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the motivation and contributions in the introduction section (Sec. 1) under a broad picture. We also discuss the social impacts further in App. F.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We will only release the trained confounding robust Atari game agents, which we believe won't pose such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: For the Atari games we cite the Gymnasium suite in Sec. 4 and for the demonstrator model we also cite the original paer in Sec. 4.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce a new set of confounded Atari games with demonstrator data. We have listed all details required to generate such games in Sec. 4 and App. D.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human objects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendices

Contents

A Related Work

Off-Policy Learning. Off-policy learning has a long history in RL dating back to the classic algorithms of Q-learning [105, 106], importance sampling [38, 94], and temporal difference [65, 76]. Recently, people also propose to utilize offline datasets to warm start the training [16, 50, 66], augmenting online training replay buffer [8, 91] or incorporating imitation loss with offline data [45, 116]. However, these work rely on a critical assumption that there is no unobserved confounders in the environment. While this assumption is generally true when the off-policy data is collected by an interventional agent, data generated by potentially unknown sources can easily break this assumption [53]. We will introduce more on the confounding robust off-policy learning next.

Causal Reinforcement Learning for Off-Policy Learning When the no unobserved confounding assumption does not hold, one would need to either identify the reward and transition distributions before evaluating policy values or bound the possible policy values. There is a rich line of literature in identifying policy values directly from confounded data [13, 27, 60, 87]. But they usually invoke other critical learning assumptions such as the existence of bridge functions in the line of proximal causal inference literature [96]. On the other hand, without further assumptions, one can utilize the bounding method to account for the whole range of possible policy values. Seminal work of Manski [59] developed the first bounds on causal effects in non-identifiable settings using observational data in the single-stage treatment model with contextual information (i.e., a contextual bandit model). These bounds were then expanded to the instrumental variable setting [7, 34], to partially identify counterfactual probabilities of causation [98], to construct reward shaping functions automatically [54]. This work is inspired by a recent work of Zhang & Bareinboim in partially identifying the policy values via bounding [113].

Robust Reinforcement Learning Unlike reinforcement learning in standard MDPs, robust reinforcement learning assumes that the parametrization of the transition probability function is contained in a set of model parameters which is called the uncertainty set [36, 56, 68, 73, 107–109]. The goal of the agent is to learn a robust policy that performs the best under the worst possible case in the uncertainty set. Similar problems have been studied under the rubrics of safe policy learning [23, 97] or pessimistic reinforcement learning [58]. Robust RL algorithms with provable guarantees have been proposed in tabular settings or under the assumptions of linear functions [5, 55, 80, 95, 102]. Combined with the computational framework of deep learning, robust RL algorithms have been extended to complex, high-dimensional domains [74, 110]. More recently, [71] proposed Robust Fitted Q-Iteration (RFQI) to learn the best possible robust policy from offline data with theoretical guarantees on the performance of the learned policy. Our work differs from robust RL methods since it does not require a pre-specified uncertainty set of model parameters. Instead, we construct the ignorance region over the underlying system dynamics from the confounded observational data

using partial causal identification. Based on the learned uncertainty set, we then derived closed-form bounds over the optimal interventional Q-value functions.

Model-Free Deep Reinforcement Learning Model-free deep reinforcement learning has seen substantial advancements through the evolution of value-based methods, particularly those building on the foundational Deep Q-Network (DQN)[61]. While DQN introduced the use of deep neural networks to approximate Q-values from high-dimensional sensory inputs, it suffered from issues such as overestimation bias, instability, and poor data efficiency. These challenges prompted a series of algorithmic improvements. Double DQN [101] addresses overestimation by separating action selection from action evaluation, while Dueling DQN [103] improves representational efficiency by decoupling value and advantage estimation. Prioritized Experience Replay [82] and Hindsight Experience Replay [3] enhances data efficiency by sampling more informative transitions. Rainbow DQN [31] effectively combines these ingredients, along with multi-step learning, distributional Q-learning, and noisy networks, resulting in one of the strongest baselines in Atari environments.

To scale value-based methods to more complex tasks and hardware infrastructures, subsequent works introduced distributed and more flexible architectures. IMPALA (Importance Weighted Actor-Learner Architectures) [21] tackles the inefficiency of distributed policy learning by decoupling acting and learning processes and correcting the resulting off-policy updates with the V-trace algorithm, enabling efficient parallel training at scale. R2D2 (Recurrent Experience Replay in Distributed Reinforcement Learning) [46] enables recurrent architectures to be trained off-policy in distributed settings with experience replay, supporting partial observability and long time horizons. Agent57 [4] further builds on R2D2, combining exploration bonuses from Random Network Distillation [19], meta-learning of exploration strategies, and population-based training, becoming the first agent to surpass human performance on all 57 Atari games. More recently, the Bigger, Better, Faster (BBF) framework [86] pushes the scalability frontier by optimizing infrastructure, neural network design, and training pipelines, enabling the training of large-scale agents with dramatically improved performance and sample efficiency. These innovations collectively mark a significant advancement toward more scalable and generalizable value-based deep reinforcement learning agents.

Note that our proposed Causal Bellman Optimality Equation does not require specific reinforcement learning algorithm implementations. In this work, we choose DQN as the base algorithm only for its simplicity given our goal of showcasing a practical implementation of the proposed result in a straightforward way. It is possible to extending the causal Q-learning update in Eq. (8) to be used with more advanced deep RL algorithms (e.g., Rainbow [31], IMPALA [21], and BBF [86]) so that one could enable more powerful agents in confounding settings, which we will leave for future work.

B Proof Details

This section entails the proof for Prop. 3.1. We also prove that our Causal Bellman Optimal Equation (lower bound) has a unique fixed point that is a valid lower bound.

Proposition B.1 (Causal Bellman Optimal Equation (Prop. 3.1))**.** *For a CMDP environment $\mathcal{M}$ with reward signals $Y_t \in [a, b] \subseteq \mathbb{R}$, its optimal state-action value function $Q_*(s, x) \geq \underline{Q_*}(s, x)$ for any state-action pair $(s, x) \in \mathcal{S} \times \mathcal{X}$, where the lower bound $\underline{Q_*}(s, x)$ is given by as follows,*

$$\underline{Q_*}(s,x) = P(x \mid s)\left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s') \max_{x'} \underline{Q_*}(s',x')\right) \tag{12}$$

$$+P(\neg x \mid s)\left(a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s',x')\right) \tag{13}$$

where $P(x \mid s) = P(X_t = x \mid S_t = s)$ and $P(\neg x \mid s) = 1 - P(x \mid s)$; $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{R}}$ are nominal transition distribution and reward function computed from the observational distribution, i.e.,

$$\widetilde{\mathcal{T}}(s,x,s') = P(S_{t+1} = s' \mid S_t = s, X_t = x), \qquad \widetilde{\mathcal{R}}(s,x) = \mathbb{E}[Y_t \mid S_t = s, X_t = x] \tag{14}$$

Proof. Starting from the Bellman Optimal Equation for Q-values, the optimal state action value function is given by,

$$Q^*(s,x) = R(s,x) + \sum_{s'} T(s,x,s') \max_{x'} Q^*(s',x') \tag{15}$$

Note that the actions in the reward and transition functions are done by an interventional agent, which is actually $\mathrm{do}(x)$ in the context of a CMDP. Due to the confounding nature of those two distributions, we can use the natural bounds to bound the interventional reward ($\mathcal{R}$) and transition distribution ($\mathcal{T}$) with observational data ($\widetilde{\mathcal{R}}$,$\widetilde{\mathcal{T}}$) [59].

$$\mathcal{R}(s,x) \geq \widetilde{R}(s,x)P(x|s) + aP(\neg x|s) \tag{16}$$

$$\sum_{s'} T(s,x,s') \max_{x'} Q^*(s',x') \geq \sum_{s'} \widetilde{T}(s,x,s')P(x|s) \max_{x'} Q^*(s',x') + P(\neg x|s) \min_{s'} \max_{x'} Q^*(s',x') \tag{17}$$

Then we have,

$$Q^*(s,x) \geq \widetilde{R}(s,x)P(x|s) + aP(\neg x|s) + \sum_{s'} \widetilde{T}(s,x,s')P(x|s) \max_{x'} Q^*(s',x') + P(\neg x|s) \min_{s'} \max_{x'} Q^*(s',x') \tag{18}$$

where $\widetilde{\mathcal{R}}_h(s,x) = \mathbb{E}[Y_h|S_h = s, X_h = x]$, $\widetilde{\mathcal{T}}_h$ is shorthand for $\widetilde{\mathcal{T}}_h(s,x,s') = P(S_{h+1} = s'|S_h = s, X_h = x)$ and $P(x|s) = P_h(X_h = x|S_h = s)$ are estimated from the offline dataset. And a is a known lower bound on the reward signal, $Y_h \leq b$. In this step, we lower bound the next state transition by assuming the worst case that for the action not taken with probability $P_h(\neg x|s)$, the agent transits with probability 1 to the worst possible next state, $\min_{s''} V^*_{h+1}(s'')$.

Then after rearranging terms, we have,

$$Q^*(s,x) \geq P(x \mid s)\left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s') \max_{x'} Q^*(s',x')\right) + P(\neg x \mid s)\left(a + \gamma \min_{s'} \max_{x'} Q^*(s',x')\right) \tag{19}$$

Optimizing the Q-value function w.r.t. this inequality gives us a lower bound on the optimal state value. Replace the symbol Q^* with $\underline{Q_*}$ and we have,

$$\underline{Q_*}(s,x) \geq P(x \mid s)\left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s') \max_{x'} \underline{Q_*}(s',x')\right) + P(\neg x \mid s)\left(a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s',x')\right) \tag{20}$$

□

Next, we will show in Prop. B.2 that this will converge to a unique fixed point, which is a valid lower bound of the optimal state-action value function.

Proposition B.2 (Convergence of Causal Bellman Optimal Equation in Stationary CMDPs)**.** *The Causal Bellman Optimality Equation converges to a unique fixed point, which is also a lower bound on the optimal interventional state values under the assumption that in the observational data $P(s,x) > 0, \forall s, x$ in the given CMDP.*

Proof. We will first show that the following Causal Bellman Optimality operator (will denote as "the operator" or T below for simplicity) is a contraction mapping with respect to a max norm. Then by Banach's fixed-point theorem [9], 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 T be,

$$T\underline{Q^*}(s,x) = P(x \mid s)\left(\widetilde{\mathcal{R}}(s,x) + \gamma \sum_{s',x'} \widetilde{\mathcal{T}}(s,x,s') \max_{x'} \underline{Q_*}(s',x')\right) + P(\neg x \mid s)\left(a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s',x')\right). \tag{21}$$

For arbitrary Q-value bound, $\underline{Q^1_*}, \underline{Q^2_*}$, let their initial difference under max-norm be $c = \max_{s,x} \left| \underline{Q^1_*}(s,x) - \underline{Q^2_*}(s,x) \right| \geq 0$. We can bound their difference after one step update by,

$$\max_{s,x} \left| T\underline{Q^1_*}(s,x) - T\underline{Q^2_*}(s,x) \right| \leq \gamma \max_{s,x} \Bigg| P(x|s) \sum_{s'} \widetilde{T}(s,x,s') \max_{x'} \left| \underline{Q^1_*}(s',x') - \underline{Q^2_*}(s',x') \right| + P(\neg x|s) \max_{s',x'} \left| \underline{Q^1_*}(s',x') - \underline{Q^2_*}(s',x') \right| \Bigg|. \tag{22}$$

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

$$\max_{s,x} \left| T\underline{Q^1_*}(s,x) - T\underline{Q^2_*}(s,x) \right| \leq \gamma \max_{s,x} \Bigg(P(x|s) \sum_{s'} \widetilde{T}(s,x,s') \max_{x'} \left| \underline{Q^1_*}(s',x') - \underline{Q^2_*}(s',x') \right| + P(\neg x|s) \max_{s',x'} \left| \underline{Q^1_*}(s',x') - \underline{Q^2_*}(s',x') \right| \Bigg), \tag{23}$$

$$\leq \gamma c \max_{s,x} \left(P(x|s) \sum_{s'} \widetilde{T}(s,x,s') + P(\neg x|s) \right), \tag{24}$$

$$= \gamma c. \tag{25}$$

for all $\underline{Q^1_*}, \underline{Q^2_*}$ satisfying $c \geq \max_{s,x} \left| \underline{Q^1_*}(s,x) - \underline{Q^2_*}(s,x) \right| \geq 0$. Thus, T is a contraction mapping with respect to the max norm. And there exists a unique fixed point $\underline{Q_*}$ when we apply this operator T iteratively to an arbitrary Q-value vector till convergence.

We then show that this fixed point is indeed a lower bound to the optimal interventional Q-value. By the update rule of T (Eq. (21)), $\forall Q(s,x), Q(s,x) \geq TQ(s,x)$. Thus, for the optimal Q-value, we can have $Q^*(s,x) \geq \lim_{k \to \infty} T^k Q^*(s,x) = \underline{Q_*}(s,x)$ where T^k denotes applying T iteratively for k times. This concludes the proof. □

C Confounded Atari Games Design

In this section, we present the detailed design of each confounded Atari games. The core design idea is that we would like to occlude the part of the screens that contains information useful for making decisions but is not a significant factor from human players' perspectives. For example, the remaining lives and current scores can be an indicator of the difficulty level or even a unique game level identifier. However, such information is not usually exploited intensively in human game plays. Thus, we decide to mask out such regions.

Below is a detailed list of confounders design for each game.

- **Amidar**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.
- **Asterix**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.
- **Boxing**: The original game screen shows score and remaining time which shouldn't be the major factor affecting the policies. The right half of the arena can also be excluded since there is still a wining strategy by staying on the left hand side (as long as the demonstrator has such demonstrations on the left hand side).
- **Breakout**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.
- **ChopperCommand**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies. The mini map can be helpful, but without it, there should also be a good policy.
- **Gopher**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies. The gopher location, while nice to have, can be removed to force the learning to focus more on the hole not the gopher.
- **KungFuMaster**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.

- **MsPacman**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.
- **Pong**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies. Moreover, one can simply win the game by shoot back every incoming ball. Thus, we can further block out the opponent's paddle location.
- **Qbert**: The color itself is already a good confounder. The demonstrator model [2] is observing three channel RGB images with LSTM cells. The student model only has grayscale image stacks. See more details in App. E.
- **RoadRunner**: The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies. Also the sky/desert part are mostly
- **Seaquest**:The original game screen shows score and remaining life which shouldn't be the major factor affecting the policies.

We show each masked atari games in Fig. 9.

D Implementation Details and Experiment Setups

The input to all the networks consists of a stack of four grayscale frames with a frame skipping of four (so the agent observes a stack of four out of sixteen consecutive actual frames), each down sampled to 64×64 resolution, allowing the agent to infer temporal dynamics such as ball velocity. To reduce the inherent flickering from the Atari game, we also apply max pooling over each two consecutive actual frames. We also downsize the input to 64x64 to align with the input size requirement of the demonstrator, diamond [2]. The only differences in terms of observations are that (1) the demonstrator takes a single colored frame (also max-pooled) as input per each time step, and (2) the demonstrator has access to the original full screen observation while the learners only has access to a masked partial screen. For the reward, we clip it between $[-1, +1]$ to stabilize training.

For the other demonstrator, we use the sebulba model [30] implemented by the CleanRL package [33]. Its backbone is from Impala [21] architecture and the training algorithm is PPO [85]. Its input image is in 81x81 but still are stacked grayscale images, the same as what the DQN agents use. Actions are selected as the argmax action with the biggest logits after applying the Gumbel softmax trick [37]. The training batch size is also increased to 2048 and is trained with cosine annealing learning rate scheduler. For all runs, the learning rate is set to 5e-4. For the consine annealing learning rate scheduler, the minimum learning rate is 1e-6.

For all CNN based DQN networks in this work, we adopt the nature DQN architecture introduced by Mnih et al. [61]. The network comprises three convolutional layers followed by two fully connected layers, outputting Q-values for each discrete action. While for LSTM based ones, we only replace the second to last linear layer in nature DQN with an LSTM cell. For both the linear layers and lstm cells, we use a hidden dimension of 512. To mitigate overestimation bias in Q-learning, we further incorporate the Double DQN modification [101], which decouples action selection and evaluation in the target update by using the online network to select actions and the target network to evaluate their value. This leads to more stable and accurate value estimates. We also use epsilon greedy for exploration as the standard DQN algorithm. See Algo. 2 for the full pseudo-code.

To train our model, we use an H100 GPU. On average, for each game and each seed, it takes around 2 hrs and a RAM space of less than 2 GB for using the diamond demonstrator [2]. While it takes up to 8 hrs for using the sebulba demonstrator [30] from CleanRL [33].

E More Experiment Results

Here we present the experiment results for agents trained with another even stronger demonstrator, sebulba [30]. From Table 2 and Fig. 10, we can draw the same conclusion that our `Causal-DQN` is robust to confounded demonstrator demonstrations and is able to extract useful policies out of such demonstrations. And we can see that in most of the games, the agent trained by sebulba demonstrator significantly outperforms the agent trained by diamond demonstrator [2]. This can be an empirical evidence that our proposed `Causal-DQN` is able to scale with the demonstrator's performance. But his also reveals two limitations of our current approach that when the demonstrator generated data has zero support in the unmasked area, it is not possible to learn any meaningful behaviors from such

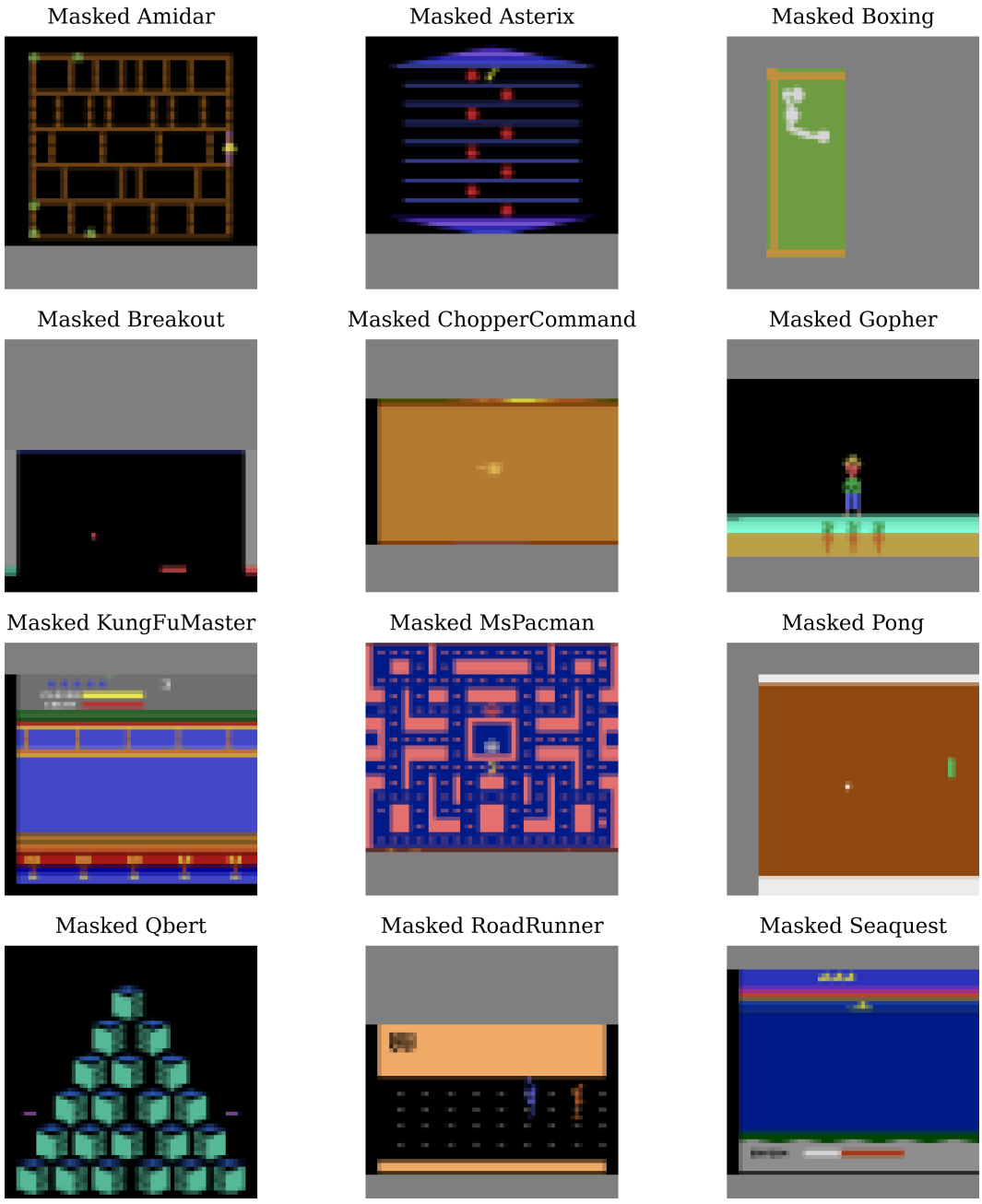


Figure 9: All 12 confounded Atari games. Masked areas are shown in grey.

demonstrations. In our current setup of the confounded Boxing game, we mask out the right half but the sebulba demonstrator has a policy of fighting in the right half. Thus, none of the algorithms we tested can learn. To further verify this intuition, we mask out the left hand side of the arena in the Boxing game (App. E) and rerun all baselines and `Causal-DQN`. As expected, as shown in App. E, our model is now able to converge to the optimal 100 score matching the demonstrator's performance while other baeslines still struggle to learn from the confounded demonstrations. Also, when the demonstrator's policy is distributional and multi-modal, i.e., there could be multiple best actions, the deterministic policy like DQN cannot capture such knowledge very well. In Asterix, due to the non-standard way of using Gumbel softmax implemented by CleanRL [33], there could be different optimal actions for the same state, posing a challenge to the deterministic learners.

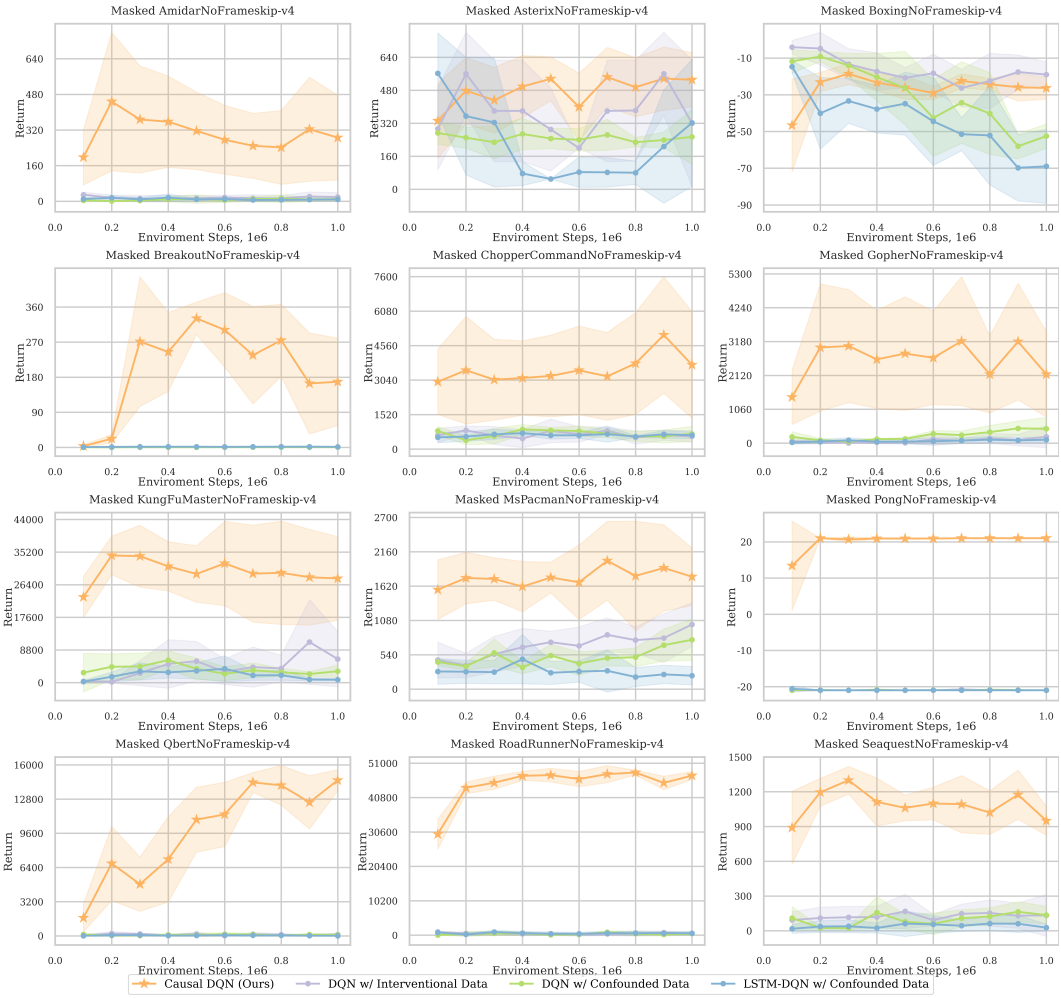


Figure 10: Average evaluation performance for 12 confounded Atari games with sebulba [30] as the demonstrator. During training over the 1M environment steps, we evaluate the agent every 100K steps for 10 episodes each time. The curve is further averaged over 5 seeds with one standard deviation across trials as the shaded area.

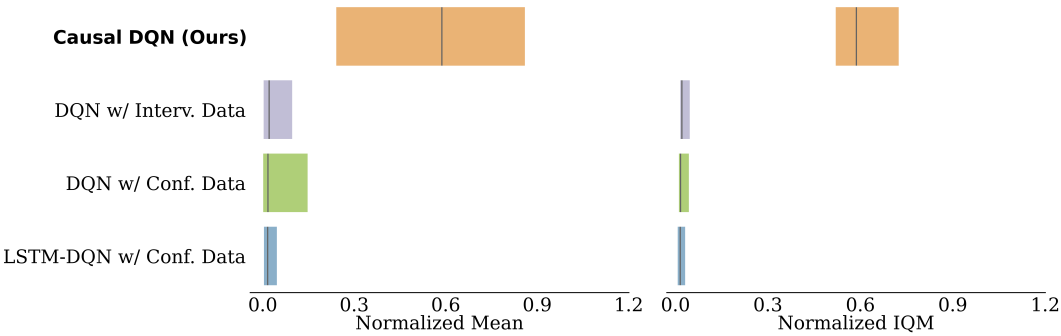


Figure 11: Normalized mean and normalized IQM scores. `Causal-DQN` achieves a normalized mean return of 0.55 and a normalized IQM of 0.59 with sebulba demonstrator.

Algorithm 2 Causal Deep Q-Learning (`Causal-DQN`)

Initialize replay memory $\mathcal{D}$
Initialize action-value function $\underline{Q_*}^{\theta}$ and a target network $\underline{Q_*}^{\theta^-}$, $\theta^- \leftarrow \theta$
for episodes $= 1, \ldots, M$ **do**
  Sample initial state s_1 and obtain preprocessed $\phi_1 = \phi(s_1)$
  for $t = 1, \ldots, T$ **do**
    With probability ϵ select a random action x_t
    Otherwise sample an action from demonstrator, $x_t \leftarrow f_X(s_t, u_t)$
    Execute action $\text{do}(x_t)$ in environment and observe reward y_t and state s_{t+1}
    Store transition (s_t, x_t, y_t, s_{t+1}) in $\mathcal{D}$
    Sample a minibatch of transitions $\{(s_i, x_i, y_i, s_{i+1})\}_{i=1}^{B}$ from $\mathcal{D}$
    Set value target $w_i(x)$ for every action $x \in \mathcal{X}$ w.r.t sample (s_i, x_i, y_i, s_{i+1}),

$$w_i(x) = \begin{cases} y_i + \gamma \max_{x'} \underline{Q_*}(s_{i+1}, x'; \theta^-) & \text{if } x = x_i \\ a + \gamma \min_{s'} \max_{x'} \underline{Q_*}(s', x'; \theta^-) & \text{if } x \neq x_i \end{cases} \tag{26}$$

    Perform a gradient descent step on $\sum_x \left(w_i(x) - \underline{Q_*}(s_i, x; \theta)\right)^2$ according to Eq. (10)
    Every T_{target} steps, update $\theta^- \leftarrow \theta$
  end for
end for

Table 2: Average evaluation returns of agents on the 12 confounded Atari games trained with 1M environment steps and aggregated normalized returns concerning the sebulba demonstrator's performance [30]. Bold numbers indicate the best-performing methods. All results are averaged over 5 seeds except that column Random is from [2]. `Causal-DQN` significantly outperforms others.

Game	Demonstrator (sebulba)	Random	Interv. DQN	Conf. DQN	Conf. LSTM-DQN	`Causal-DQN` (diamond)	`Causal-DQN` (sebulba)
Amidar	2148.2	5.8	44.0	22.6	24.6	282.6	**462.8**
Asterix	250182.0	210.0	650.0	369.0	662.0	**2587.0**	586.0
Boxing	100.0	0.1	-0.62	-7.6	-13.4	**71.5**	-16.86
Breakout	771.0	1.7	2.2	0.9	2.9	131.2	**408.2**
ChopperCommand	21682.0	811.0	1192.0	1096.0	918.0	1658.0	**5410.0**
Gopher	3719.2	257.6	288.8	646.4	132.0	**7327.2**	4008.0
KungFuMaster	46046.0	258.5	12416.0	8468.0	4844.0	**44196.0**	39222.0
MsPacman	4538.4	307.3	1191.6	963.6	561.4	1747.6	**2346.8**
Pong	21.0	-20.7	-20.8	-20.8	-20.6	**21.0**	**21.0**
Qbert	23484.0	163.9	322.5	283.5	136.5	4458.5	**15136.0**
RoadRunner	56056.0	11.5	1154.0	1182.0	1108.0	27414.0	**49482.0**
Seaquest	1797.2	68.4	237.2	247.6	101.6	980.0	**1350.0**
Normalized Mean (↑)	1.00	0.00	0.00	0.00	0.00	0.53	**0.55**
Normalized Median (↑)	1.00	0.01	0.03	0.04	0.02	0.40	**0.59**
Normalized IQM (↑)	1.00	0.0	0.02	0.02	0.02	0.47	**0.59**

F Broader Impact

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. One major reason is that our proposed algorithm aims at extracting knowledge from another demonstrator model solving Atari games of which we don't find any profound social impacts worth mentioning here.

G Limitations

Our current derivation applies to single step Q-value functions. For other objectives for critics like multi-step returns, eligibility traces and advantages, we need to further extend the Causal Bellman Equation to accommodate those. As we also show in App. E, the proposed `Causal-DQN` cannot learn useful policies when the demonstrator policy has no support in the unmasked area. For example, in the boxing game, we mask out the right half of the screen while the demonstrator agent's winning

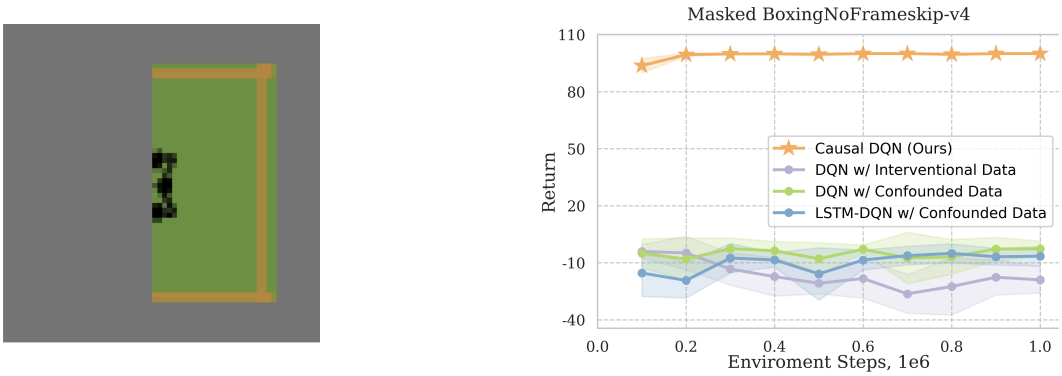


Figure 12: Left: The confounded Boxing game with left side arena masked. Right: Evaluation returns of `Causal-DQN` and other baselines. The curve is an average over five random seeds with one standard deviation as the shaded area. The orange curve is our `Causal-DQN` and other colors are baselines. Same legend following previous figures.

policy is to fight in the right half. In which case, none of the agent under masked observations is able to learn. Furthermore, in games like Asterix where a distributional demonstrator has a more stochastic behavior, our `Causal-DQN` also cannot outperform others. We hypothesis this to be an inherent representational limit of deterministic DQN policies.