# Appendix

## Table of Contents



## A Broader Impact

Incorporating causality into reinforcement learning methods increases the interpretability of artificial intelligence, which helps humans understand the underlying mechanism of algorithms and check the source of failures. However, the learned causal transition model may contain human-readable private information about the environment, which could raise privacy issues. To mitigate this potential negative societal impact, the causal transition model needs to be encrypted and only accessible to algorithms and trustworthy users.

## B Additional Related Works

In this section, besides the most related formulation, robust RL introduced in Sec 3.3, we also introduce some other related RL problem formulations partially shown in Figure 3. Then, we limit our discussion to mainly two lines of work that are related to ours: (1) promoting robustness in RL; (2) concerning the spurious correlation issues in RL.

### B.1 Related RL formulations

**Robustness to noisy state: POMDPs and SA-MDPs.** State-noisy MDPs refer to the RL problem that the agent can only access and choose the action based on a noisy observation rather than the true state at each step, including two existing types of problems: Partially observable MDPs (POMDPs) and state-adversarial MDPs (SA-MDPs), shown in Figure 3(b). In particular, at each step $t$, in POMDPs, the observation $o_t$ is generated by a fixed probability transition $\mathcal{O}(\cdot \mid s_t)$ (we refer to the case that $o_t$ only depends on the state $s_t$ but not action); for state-adversarial MDPs, the observation is an adversary $\nu(s_t)$ against and thus determined by the conducted policy, leading to the worst performance by perturbing the state in a small set around itself. To against the state perturbation, both POMDPs, and SA-MDPs are indeed robust to the noisy observation, or called agent-observed state, but not the real state that transitions to the environment and next steps. In contrast, our RSC-MDPs propose the robustness to the real state shift that will directly transition to the next state in the environment, involving additional challenges induced by the appearance of out-of-distribution states.

**Robustness to unobserved confounder: MDPUC and confounded MDPs.** To address the misleading spurious correlations hidden in components of RL, people formulate RL problems as MDPs with some additional components – unobserved confounders. In particular, the Markov decision process with unobserved confounders (MDPUC) [35] serves as a general framework to concern all types of possible spurious correlations in RL problems – at each step, the state, action, and reward are all possibly influenced by some unobserved confounder, shown in Figure 2(d); confounded MDPs [19] mainly concerns the misleading correlation between the current action and the next state, illustrated in Figure 3(e). The proposed state-confounded MDPs (SC-MDPs) can be seen as a specified type of MDPUC that focus on breaking the spurious correlation between different parts of the state space itself (different from confounded MDPs which consider the correlation between action and next state), motivated by various real-world applications in self-driving and control tasks. In addition, the proposed formulation is more flexible and can work in both online and offline RL settings.

**Contexual MDPs (CDMPs).** A contextual MDP (CMDP) [36] is basically a set of standard MDPs sharing the same state and action space but specified by different contexts within a context space. In particular, the transition kernel, reward, and action of a CMDP are all determined by a (possibly unknown) fixed context. The proposed robust state-confounded MDPs (RSC-MDPs) are similar to CMDPs if we cast the unobserved confounder as the context in CMDPs, while different in two aspects: (1) In a CMDP, the context is fixed throughout an episode, while the unobserved confounder in RSC-MDPs can vary as $\{c_t\}_{1\leq t\leq T}$; (2) In the online setting, the goal of CMDP is to beat the optimal policy depending on the context, while RSC-MDPs seek to learn the optimal policy that does not depend on the confounder $\{c_t\}_{1\leq t\leq T}$.

### B.2 Related literature of robustness in RL

**Robust RL (robust MDPs).** Concerning the robust issues in RL, a large portion of works focus on robust RL with explicit uncertainty of the transition kernel, which is well-posed and a natural way to consider the uncertainty of the environment [13, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]. However, to define the uncertainty set for the environment, most existing works use task structure-agnostic and heuristic 'distance' such as KL divergence and total variation [14, 47, 48, 15, 49, 50, 51, 52] to measure the shift between the training and test transition kernel, leading to a homogeneous (almost structure-free) uncertainty set around the state space. In contrast, we consider a more general uncertainty set that enables the robustness to a task-dependent heterogeneous uncertainty set shaped by unobserved confounder and causal structure, in order to break the spurious correlation hidden in different parts of the state space.

**Robustness in RL** Despite the remarkable success that standard RL has achieved, current RL algorithms are still limited since the agent is vulnerable if the deployed environment is subject to uncertainty and even structural changes. To address these challenges, a recent line of RL works begins to concern robustness to the uncertainty or changes over different components of MDPs – state, action, reward, and transition kernel, where a review [8] can be referred to. Besides robust RL framework concerning the shift of the transition kernel and reward, to promote robustness in RL, there exist various works [11, 12] that consider the robustness to action uncertainty, i.e., the deployed action in the environment is distorted by an adversarial agent smoothly or circumstantially; some works [9, 6, 10, 53, 54, 55] investigate the robustness to the state uncertainty including but not limited to the introduced POMDPs and SA-MDPs in Appendix B.1, where the agent chooses the action based on observation – the perturbed state determined by some restricted noise or adversarial attack. The proposed RSC-MDPs can be regarded as addressing the state uncertainty since the shift of the unobserved confounder leads to state perturbation. In contrast, RSC-MDPs consider the out-of-distribution of the real state that will directly influence the subsequent transition in the environment, but not the observation in POMDPs and SA-MDPs that will not directly influence the environment.

### B.3 Related literature of spurious correlation in RL

**Confounder in RL.** These works mainly focus on the confounder between action (treatment) and state (effect), which is a long-standing problem that exists in the causal inference area. However, we find that the confounder may cause problems from another perspective, where the confounder is built upon different dimensions of the state variable. Some people focus on the confounder between action and state, which is common in offline settings since the dataset is fixed and intervention is not allowed. But in the online setting, actions are controlled by an agent and intervention is available to eliminate spurious correlation. [56] reduces the spurious correlation between action and state in the offline setting. [57] deal with environment-irrelevant white noise; possible shift + causal [58]. The confounder problem is usually easy to solve since agents can interact with the environment to do interventions. However, different from most existing settings, we find that even with the capability of intervention, the confounding between dimensions in states cannot be fully eliminated. Then the learned policy is heavily influenced if these confounder change during testing.

**Invariant Feature learning.** The problem of spurious correlation has attracted attention in the supervised learning area for a long time and many solutions are proposed to learn invariant features to eliminate spurious correlations. A general framework to remedy the ignorance of spurious correlation in empirical risk minimization (ERM) is invariant risk minimization (IRM) [59]. Other works tackle this problem with group distributional robustness [60], adversarial robustness [61], and contrastive learning [62]. These methods are also adapted to sequential settings. The idea of increasing the robustness of RL agents by training agents on multiple environments has been shown in previous works [63, 30, 30]. However, a shared assumption among these methods is that multiple environments with different values of confounder are accessible, which is not always true in the real world.

**Counterfactual Data Augmentation in RL.** One way to simulate multiple environments is data augmentation. However, most data augmentation works [24, 64, 25, 65, 66, 67, 68] apply image transformation to raw inputs, which requires strong domain knowledge for image manipulation and cannot be applied to other types of inputs. In RL, the dynamic model and reward model follow certain causal structures, which allow counterfactual generation of new transitions based on the collected samples. This line of work, named counterfactual data augmentation, is very close to this work. Deep generative models [69] and adversarial examples [70] are considered for the generation to improve sample efficiency in model-based RL. CoDA [71] and MocoDA [32] leverage the concept of locally factored dynamics to randomly stitch components from different trajectories. However, the assumption of local causality may be limited.

**Domain Randomization.** If we are allowed to control the data generation process, e.g., the underlying mechanism of the simulator, we can apply the golden rule in causality – Randomized Controlled Trial (RCT). The well-known technic, domain randomization [72], exactly follows the idea of RCT, which randomly perturb the internal state of the experiment in simulators. Later literature follows this direction and develops variants including randomization guided by downstream tasks in the target domain [73, 74], randomization to match real-world distributions [75, 76], and randomization to minimize data divergence [77]. However, it is usually impossible to randomly manipulate internal states in most situations in the real world. In addition, determining which variables to randomize is even harder given so many factors in complex systems.

**Discovering Spurious Correlations** Detecting spurious correlations helps models remove features that are harmful to generalization. Usually, domain knowledge is required to find such correlations [78, 79, 80]. However, when prior knowledge is accessible, technics such as clustering can also be used to reveal spurious attributes [35, 81, 82]. When human inspection is available, recent works [83, 84, 85] also use explainability techniques to find spurious correlations. Another area for discovery is concept-level and interactive debugging [86, 87], which leverage concepts or human feedback to perform debugging.

# C Theoretical Analyses

## C.1 Proof of Theorem 1

The proof follows the pipeline of proving the existence of the optimal policy for standard MDPs but tailored for RSC-MDPs since the additional components confounder $C_{\rm s}$ and the infimum operator. To begin with, recall that the goal is to find a policy $\widetilde{\pi} = \{\widetilde{\pi}_t\}_{1\leq t\leq T}$ such that:

$$\widetilde{V}_t^{\widetilde{\pi},\sigma}(s) = \widetilde{V}_t^{\star,\sigma}(s) \coloneqq \sup_{\pi\in\Pi} \widetilde{V}_t^{\pi,\sigma}(s) \quad \text{and} \quad \widetilde{Q}_t^{\widetilde{\pi},\sigma}(s,a) = \widetilde{Q}_t^{\star,\sigma}(s,a) \coloneqq \sup_{\pi\in\Pi} \widetilde{Q}_t^{\pi,\sigma}(s,a). \tag{8}$$

Towards this, we start from the first claim in equation 8. Before proceeding, we let $\{S_t, A_t, R_t, C_t\}$ denote the random variables at time step $t$ for all $1 \le t \le T$. Then due to the Markov properties, we know that conditioned on current state $s_t$, the future state, action, and reward are all independent from the previous $s_1, a_1, r_1, c_1, \cdots, s_{t-1}, a_{t-1}, r_{t-1}, c_{t-1}$. For convenience, we introduce the following notation:

$$\forall 1 \le t \le T: \quad P_{+t} \coloneqq \{P_k\}_{t \le k \le T} \quad \text{and} \quad \mathcal{U}^\sigma(P^c_{+t}) \coloneqq \{\mathcal{U}^\sigma(P^c_k)\}_{t \le k \le T} \tag{9}$$

to represent the collection of variables from time step $t$ to the end of the episode, and choose $\widetilde{\pi}$ to obey

$$\forall 1 \le t \le T: \quad \pi_t(s) \coloneqq \mathrm{argmax}_{a \in \mathcal{A}} \mathbb{E}\left[ r_t(s,a) + \inf_{P_t \in \mathcal{U}^\sigma(P^c_{t,s,a})} \mathbb{E}_{c_t \sim P_t}\left[ \widetilde{V}^{\star,\sigma}_{t+1}(s_{t+1}) \right] \right] \tag{10}$$

With the above preparation in mind, for any $(t,s) \in \{1, 2, \cdots, T\} \times \mathcal{S}$, one has

$$\begin{aligned}
&\widetilde{V}^{\star,\sigma}_t(s) \\
&\overset{\text{(i)}}{=} \sup_{\pi \in \Pi} \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} \widetilde{V}^{\pi,P}_t(s) \overset{\text{(ii)}}{=} \sup_{\pi \in \Pi} \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} \mathbb{E}_{\pi, P_{+t}}\left[ \sum_{k=t}^{T} r_k(s_k, a_k) \right] \\
&\overset{\text{(iii)}}{=} \sup_{\pi \in \Pi} \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} \mathbb{E}_{\pi_t}\Bigg[ r_t(s, a_t) \\
&\quad + \mathbb{E}_{c_t \sim P_t} \mathbb{E}\left[ \sum_{k=t+1}^{T} r_k(s_k, a_k) \,|\, \pi, P_{+(t+1)}, (S_t, A_t, R_t, C_t, S_{t+1}) = (s, a_t, r_t, c_t, s_{t+1}) \right] \Bigg] \\
&= \sup_{\pi \in \Pi} \mathbb{E}_{\pi_t}\Bigg[ \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} r_t(s, a_t) + \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} \mathbb{E}_{c_t \sim P_t} \\
&\quad \mathbb{E}\left[ \sum_{k=t+1}^{T} r_k(s_k, a_k) \,|\, \pi, P_{+(t+1)}, (S_t, A_t, R_t, C_t, S_{t+1}) = (s, a_t, r_t, c_t, s_{t+1}) \right] \Bigg]
\end{aligned}$$

where (i) and (ii) holds by the definitions in equation 5 and equation 3 respectively, and (iii) follows from expressing the term of interest by moving one step ahead and $\mathbb{E}_{\pi_t}$ is taken with respect to $a_t \sim \pi_t(\cdot \,|\, S_1 = s_1, A_1 = a_1, \cdots, S_t = s)$, and the last equality arises from we can exchange the operators $\mathbb{E}_{\pi_t}$ and $\inf_{P \in \mathcal{U}^\sigma(P^c)}$ since they are independent.

To continue, we observe that the above equation can be rewritten and controlled as follows:

$$\begin{aligned}
&\widetilde{V}^{\star,\sigma}_t(s) \\
&= \sup_{\pi \in \Pi} \mathbb{E}_{\pi_t}\Bigg[ \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} r_t(s, a_t) + \inf_{P_t \in \mathcal{U}^\sigma(P^c_t)} \mathbb{E}_{c_t \sim P_t} \inf_{P_{+(t+1)} \in \mathcal{U}^\sigma(P^c_{+(t+1)})} \\
&\quad \mathbb{E}\left[ \sum_{k=t+1}^{T} r_k(s_k, a_k) \,|\, \pi', P_{+(t+1)}, (S_t, A_t, R_t, C_t, S_{t+1}) = (s, a_t, r_t, c_t, s_{t+1}) \right] \Bigg] \\
&\le \sup_{\pi \in \Pi} \mathbb{E}_{\pi_t}\Bigg[ \inf_{P_{+t} \in \mathcal{U}^\sigma(P^c_{+t})} r_t(s, a_t) + \inf_{P_t \in \mathcal{U}^\sigma(P^c_t)} \mathbb{E}_{c_t \sim P_t} \sup_{\pi' \in \Pi} \inf_{P_{+(t+1)} \in \mathcal{U}^\sigma(P^c_{+(t+1)})} \\
&\quad \mathbb{E}\left[ \sum_{k=t+1}^{T} r_k(s_k, a_k) \,|\, \pi', P_{+(t+1)}, (S_t, A_t, R_t, C_t, S_{t+1}) = (s, a_t, r_t, c_t, s_{t+1}) \right] \Bigg] \\
&\overset{\text{(i)}}{=} \sup_{\pi \in \Pi} \mathbb{E}_{\pi_t}\left[ r_t(s, a_t) + \inf_{P_t \in \mathcal{U}^\sigma(P^c_t)} \mathbb{E}_{c_t \sim P_t}\left[ \sup_{\pi' \in \Pi} \inf_{P_{+(t+1)} \in \mathcal{U}^\sigma(P^c_{+(t+1)})} \mathbb{E}_{\pi', P_{+(t+1)}}\left[ \sum_{k=t+1}^{T} r_k(s_k, a_k) \right] \right] \right] \\
&= \sup_{\pi \in \Pi} \mathbb{E}_{\pi_t}\left[ r_t(s, a_t) + \inf_{P_t \in \mathcal{U}^\sigma(P^c_t)} \mathbb{E}_{c_t \sim P_t}\left[ \widetilde{V}^{\star,\sigma}_{t+1}(s_{t+1}) \right] \right]
\end{aligned}$$

$$
\begin{aligned}
&= \sup_{a_t \in \mathcal{A}} \mathbb{E}_{a_t} \left[ r_t(s, a_t) + \inf_{P_t \in \mathcal{U}^\sigma(P_t^c)} \mathbb{E}_{c_t \sim P_t} \left[ \widetilde{V}_{t+1}^{\star,\sigma}(s_{t+1}) \right] \right] \\
&= \inf_{P_t \in \mathcal{U}^\sigma(P_t^c)} \mathbb{E} \left[ r_t(s, a_t) + \mathbb{E}_{c_t \sim P_t} \left[ \widetilde{V}_{t+1}^{\star,\sigma}(s_{t+1}) \right] \mid a_t = \widetilde{\pi}_t(s) \right], \qquad (11)
\end{aligned}
$$

where (i) holds by the Markov decision such that the rewards $\{r_k(s_k, a_k)\}_{t+1 \le k \le T}$ conditioned on determined $(S_t, A_t, R_t, C_t, S_{t+1})$ or $S_{t+1}$ are the same, and the last equality follows from the definition of $\widetilde{\pi}$ in equation 10 and the exchangeability of $\inf_{P_t \in \mathcal{U}^\sigma(P_t^c)}$ and $\mathbb{E}_{a_t}[\cdot]$.

Applying equation 11 recursively for $t+1, \cdots T$, we arrive at

$$
\begin{aligned}
&\widetilde{V}_t^{\star,\sigma}(s) \le \inf_{P_t \in \mathcal{U}^\sigma(P_t^c)} \mathbb{E} \left[ r_t(s, a_t) + \mathbb{E}_{c_t \sim P_t} \left[ \widetilde{V}_{t+1}^{\star,\sigma}(s_{t+1}) \right] \mid a_t = \widetilde{\pi}_t(s) \right] \\
&\le \inf_{P_t \in \mathcal{U}^\sigma(P_t^c)} \inf_{P_{t+1} \in \mathcal{U}^\sigma(P_{t+1}^c)} \mathbb{E} \Bigg[ r_t(s, a_t) + \\
&\quad \mathbb{E}_{c_t \sim P_t} \left[ r_{t+1}(s_{t+1}, a_{t+1}) + \mathbb{E}_{c_{t+1} \sim P_{t+1}} \left[ \widetilde{V}_{t+2}^{\star,\sigma}(s_{t+1}) \right] \right] \mid (a_t, a_{t+1}) = (\widetilde{\pi}_t(s), \widetilde{\pi}_{t+1}(s_{t+1})) \Bigg] \\
&\le \cdots \le \inf_{P_{+t} \in \mathcal{U}^\sigma(P_{+t}^c))\}} \mathbb{E}_{\pi, P} \left[ \sum_{k=t}^{T} r_k(s_k, a_k) \right] = \widetilde{V}_t^{\widetilde{\pi},\sigma}(s). \qquad (12)
\end{aligned}
$$

where (i) holds by the Markov properties of the rewards.

Observing from equation 12 that

$$
\forall s \in \mathcal{S}: \quad \widetilde{V}_t^{\star,\sigma}(s) \le \widetilde{V}_t^{\widetilde{\pi},\sigma}(s) \le \sup_{\pi \in \Pi} \widetilde{V}_t^{\pi,\sigma}(s) = \widetilde{V}_t^{\star,\sigma}(s), \qquad (13)
$$

which directly verifies the first assertion in equation 8 $\widetilde{V}_t^{\widetilde{\pi},\sigma}(s) = \widetilde{V}_t^{\star,\sigma}(s)$ for all $s \in \mathcal{S}$. The second assertion in equation 8 can be achieved analogously. Until now, we verify that there exists at least a policy $\widetilde{\pi}$ that obeys equation 8, which we refer it as an optimal policy since its value is equal to or larger than any other non-stationary and stochastic policies over all states $s \in \mathcal{S}$.

### C.2 Proof of Theorem 2

**Constructing a hard instance of the standard MDP.** In this section, we consider the following standard MDP instance $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P^0, T, r\}$, where $\mathcal{S} = \{[0,0], [0,1], [1,0], [1,1]\}$ is the state space consisting of four elements in dimension $n = 2$, and $\mathcal{A} = \{0, 1\}$ is the action space with only two options. The transition kernel $P^0 = \{P_t^0\}_{1 \le t \le T}$ at different time steps $1 \le t \le T$ is defined as

$$
P_1^0(s' \mid s, a) = \begin{cases} \mathbb{1}(s' = [0,0]) \mathbb{1}(a = 0) + \mathbb{1}(s' = [0,1]) \mathbb{1}(a = 1) & \text{if}(s, a) = ([0,0], a) \\ \mathbb{1}(s' = s) & \text{otherwise} \end{cases}, \qquad (14)
$$

and

$$
P_t^0(s' \mid s, a) = \mathbb{1}(s' = s), \quad \forall (t, s, a) \in \{2, 3, \cdots, T\} \times \mathcal{S} \times \mathcal{A}. \qquad (15)
$$

Note that this transition kernel $P^0$ ensures the next state transitioned from the state $[0,0]$ is either $[0,0]$ or $[0,1]$. The reward function is specified as follows: for all time steps $1 \le t \le T$,

$$
r_t(s, a) = \begin{cases} 1 & \text{if } s = [0,0] \text{ or } s = [1,1] \\ 0 & \text{otherwise} \end{cases}. \qquad (16)
$$

**The equivalence to one SC-MDP.** Then, we shall show that the constructed standard MDP $\mathcal{M}$ can be equivalently represented by one SC-MDP $\mathcal{M}_{\mathsf{sc}} = \{\mathcal{S}, \mathcal{A}, T, r, \mathcal{C}, \{\mathcal{P}_t^i\}, P^c\}$ with $\mathcal{C} := [0,1]$, which yields the sequential observations $\{s_t, a_t, r_t\}_{1 \le t \le T}$ induced by any policy and any initial state distribution in two processes are identical. To specify, $\mathcal{S}, \mathcal{A}, T, r$ are kept the same as $\mathcal{M}$. Here, $\{\mathcal{P}_t^i\}$ shall be specified in a while, which determines the transition to each dimension of the next state conditioned on the current state, action, and confounder for all time steps, i.e., $s_{t+1}^i \sim \mathbb{E}_{c_t \sim P_t^c} \left[ \mathcal{P}_t^i(\cdot \mid s_t, a_t, c_t) \right]$ for any $i$-th dimension of the state ($i \in \{1, 2\}$ and all timestep $1 \le t \le T$.

For convenience, we denote $\mathcal{P}_t := [\mathcal{P}_t^1, \mathcal{P}_t^2] \in \Delta(\mathcal{S})$ as the transition kernel towards the next state, namely, $s_{t+1} \sim \mathbb{E}_{c_t \sim P_t^c} [\mathcal{P}_t(\cdot \mid s_t, a_t, c_t)]$.

To ensure the marginalized transition probability from any state-action pair $(s_t, a_t)$ to the next state $s_{t+1}$ in $\mathcal{M}_{\mathsf{sc}}$ aligns with the one in the MDP $\mathcal{M}$, we set

$$P_t^c(c) = \mathbb{1}(c = 0), \quad \forall 1 \leq t \leq T. \tag{17}$$

In addition, before introducing the transition kernel $\{\mathcal{P}_t^i\}$ of the SC-MDP $\mathcal{M}_{\mathsf{sc}}$, we introduce an auxiliary transition kernel $P^{\mathrm{sc}} = \{P_t^{\mathrm{sc}}\}$ as follows:

$$P_1^{\mathrm{sc}}(s' \mid s, a) = \begin{cases} \mathbb{1}(s' = [1,0])\mathbb{1}(a = 0) + \mathbb{1}(s' = [1,1])\mathbb{1}(a = 1) & \text{if} (s, a) = ([0,0], 0) \\ \mathbb{1}(s' = s) & \text{otherwise} \end{cases}, \tag{18}$$

and

$$P_t^{\mathrm{sc}}(s' \mid s, a) = \mathbb{1}(s' = s), \quad \forall (t, s, a) \in \{2, 3, \cdots, T\} \times \mathcal{S} \times \mathcal{A}. \tag{19}$$

It can be observed that $P^{\mathrm{sc}}$ is similar to $P^0$ except for the transition in the state $[0, 0]$.

Armed with this transition kernel $P^{\mathrm{sc}}$, the $\{\mathcal{P}_t^i\}$ of the SC-MDP $\mathcal{M}_{\mathsf{sc}}$ is set to obey

$$\mathcal{P}_1(s' \mid s, a, c) = \begin{cases} (1 - c)P_1^0(s' \mid s, a) + cP_1^{\mathrm{sc}}(s' \mid s, a) & \text{if} (s, a) = ([0,0], a) \\ \mathbb{1}(s' = s) & \text{otherwise} \end{cases}, \tag{20}$$

and

$$\mathcal{P}_t(s' \mid s, a, c) = \mathbb{1}(s' = s), \quad \forall (t, s, a, c) \in \{2, 3, \cdots, T\} \times \mathcal{S} \times \mathcal{A} \times \mathcal{C}. \tag{21}$$

With the above preparation, we are ready to verify that the marginalized transition from the current state and action to the next state in the SC-MDP $\mathcal{M}_{\mathsf{sc}}$ is identical to the one in MDP $\mathcal{M}$: for all $(t, s_t, a_t, s_{t+1}) \in \{1, 2, \cdots, T\} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$:

$$\mathbb{P}(s_{t+1} \mid s_t, a_t) = \mathbb{E}_{c_t \sim P_t^c} [\mathcal{P}_t(s_{t+1} \mid s_t, a_t, c_t)] = \mathcal{P}_t(s_{t+1} \mid s_t, a_t, 0) = P^0(s_{t+1} \mid s_t, a_t) \tag{22}$$

where the second equality holds by the definition of $P^c$ in equation 17, and the last equality holds by the definitions of $P^0$ (see equation 14 and equation 15) and $\mathcal{P}$ (see equation 20 and equation 21).

In summary, we verified that the standard MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P^0, T, r\}$ is equal to the above specified SC-MDP $\mathcal{M}_{\mathsf{sc}}$.

**Defining the corresponding RMDP and RSC-MDP.** Equipped with the equivalent MDP $\mathcal{M}$ and SC-MDP $\mathcal{M}_{\mathsf{sc}}$, people could consider the robust variants of them respectively — a RMDP $\mathcal{M}_{\mathsf{rob}} = \{\mathcal{S}, \mathcal{A}, \mathcal{U}^{\sigma_1}(P^0), T, r\}$ with the uncertainty level $\sigma_1$, and the proposed RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}} = \{\mathcal{S}, \mathcal{A}, T, r, \mathcal{C}, \{\mathcal{P}_t^i\}, \mathcal{U}^{\sigma_2}(P^c)\}$ with the uncertainty level $\sigma_2$.

In this section, without loss of generality, we consider total deviation as the 'distance' function $\rho$ for the uncertainty sets of both RMDP $\mathcal{M}_{\mathsf{rob}}$ and RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$, i.e., for any probability vectors $P', P \in \Delta(\mathcal{C})$ (or $P', P \in \Delta(\mathcal{S})$), $\rho(P', P) := \frac{1}{2}\|P' - P\|_1$. Consequently, for any uncertainty set $\sigma \in [0, 1]$, the uncertainty set $\mathcal{U}^{\sigma_1}(P^0)$ of the RMDP (see equation 1) and $\mathcal{U}^{\sigma_2}(P^c)$ of the RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$ (see equation 4) are defined as follows:

$$\begin{aligned} &\mathcal{U}^{\sigma}(P^0) := \otimes\, \mathcal{U}^{\sigma}(P^0_{t,s,a}), \qquad \mathcal{U}^{\sigma}(P^0_{t,s,a}) := \left\{ P_{t,s,a} \in \Delta(\mathcal{S}) : \frac{1}{2}\left\|P_{t,s,a} - P^0_{t,s,a}\right\|_1 \leq \sigma \right\}, \\ &\mathcal{U}^{\sigma}(P^c) := \otimes\, \mathcal{U}^{\sigma}(P_t^c), \qquad \mathcal{U}^{\sigma}(P_t^c) := \left\{ P \in \Delta(\mathcal{C}) : \frac{1}{2}\left\|P - P_t^c\right\|_1 \leq \sigma \right\}. \end{aligned} \tag{23}$$

To continue, the proof is established by specifying the robust optimal policy $\pi^{\star,\sigma_1}_{\mathsf{RMDP}}$ associated with $\mathcal{M}_{\mathsf{rob}}$ and $\pi^{\star,\sigma_2}_{\mathsf{RSC}}$ associated with $\mathcal{M}_{\mathsf{sc\text{-}rob}}$ and then compare their performance on RSC-MDP with some initial state distribution.

**The performance comparisons between $\pi^{\star,\sigma_1}_{\mathsf{RMDP}}$ of RMDP $\mathcal{M}_{\mathsf{rob}}$ and $\pi^{\star,\sigma_2}_{\mathsf{RSC}}$ of RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$.**

To begin, we introduce the following lemma which specifies the robust optimal policy $\pi^{\star,\sigma_1}_{\mathsf{RMDP}}$ associated with the RMDP $\mathcal{M}_{\mathsf{rob}}$.

**Lemma 1.** *For any $\sigma_1 \in (0,1]$, the robust optimal policy and its corresponding robust SC-value functions satisfy*

$$\pi^{\star,\sigma_1}_{\mathsf{RMDP}}(0\,|\,s) = 1, \qquad\qquad \textit{for } s \in \mathcal{S}. \tag{24a}$$

In addition, we characterize the robust SC-value functions of the RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$ associated with any policy, combined with the robust optimal policy $\pi^{\star,\sigma_2}_{\mathsf{RSC}}$ of $\mathcal{M}_{\mathsf{sc\text{-}rob}}$ — the optimal robust SC-value functions, shown in the following lemma.

**Lemma 2.** *Consider any $\sigma_2 \in (\frac{3}{4},1]$ and the RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}} = \big\{\mathcal{S},\mathcal{A},T,r,\mathcal{C},\{\mathcal{P}^i_t\},\mathcal{U}^{\sigma_2}(P^c)\big\}$. For any policy $\pi$, the corresponding robust SC-value functions satisfy*

$$\widetilde{V}^{\pi,\sigma_2}_1([0,0]) = 1 + (T-1)\inf_{P\in\mathcal{U}^{\sigma}(P^c_1)} \mathbb{E}_{c_1\sim P}\Bigg[\pi_1(0\,|\,[0,0])(1-c_1) + \pi_1(1\,|\,[0,0])c_1\Bigg]. \tag{25a}$$

*In addition, the optimal robust SC-value function and the robust optimal policy $\pi^{\star,\sigma_2}_{\mathsf{RSC}}$ of the RMDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$ obeys:*

$$\widetilde{V}^{\pi^{\star,\sigma_2}_{\mathsf{RSC}},\sigma_2}_1([0,0]) = \widetilde{V}^{\star,\sigma_2}_1([0,0]) = 1 + \frac{T-1}{2}. \tag{26}$$

Applying Lemma 2 with policy $\pi = \pi^{\star,\sigma_1}_{\mathsf{RMDP}}$ in Lemma 1, one has

$$\widetilde{V}^{\pi^{\star,\sigma_1}_{\mathsf{RMDP}},\sigma_2}_1([0,0]) = 1 + (T-1)\inf_{P\in\mathcal{U}^{\sigma}_2(P^c_1)} \mathbb{E}_{c_1\sim P}\Bigg[1-c_1\Bigg] \le 1 + \frac{T-1}{4}, \tag{27}$$

where the last inequality holds by the probability distribution $P$ obeying $P_1(0) = \frac{1}{4}$ and $P_1(1) = \frac{3}{4}$ is inside the uncertainty set $\mathcal{U}^{\sigma}_2(P^c_1)$.

Finally, putting equation 27 and equation 26 together, we complete the proof by showing that with the initial state distribution $\phi$ define as $\rho(s_1 = [0,0]) = 1$, we arrive at

$$\widetilde{V}^{\pi^{\star,\sigma_2}_{\mathsf{RSC}},\sigma_2}_1(\phi) - \widetilde{V}^{\pi^{\star,\sigma_1}_{\mathsf{RMDP}},\sigma_2}_1(\phi) = \widetilde{V}^{\star,\sigma_2}_1(\phi) - \widetilde{V}^{\pi^{\star,\sigma_1}_{\mathsf{RMDP}},\sigma_2}_1(\phi) \ge \frac{T-1}{4} \approx \frac{T}{4}. \tag{28}$$

### C.2.1 Proof of Lemma 1

**Specifying the minimum of the robust value functions in different states.** For any uncertainty set $\sigma_1 \in (0,1]$, we first characterize the robust value function of any policy $\pi$ over different states. To start, we denote the minimum of the robust value function over states at each time step $t$ as below:

$$V^{\pi,\sigma_1}_{\min,t} \coloneqq \min_{s\in\mathcal{S}} V^{\pi,\sigma_1}_t(s) \ge 0, \tag{29}$$

where the last inequality holds by that the reward function defined in equation 16 is always non-negative. Obviously, there exists at least one state $s^{\pi}_{\min,t}$ that satisfies $V^{\pi,\sigma_1}_t(s^{\pi}_{\min,t}) = V^{\pi,\sigma_1}_{\min,t}$.

With this in mind, we shall verify that for any policy $\pi$,

$$\forall 1 \le t \le T: \quad V^{\pi,\sigma_1}_t([0,1]) = V^{\pi,\sigma_1}_t([1,0]) = 0. \tag{30}$$

To achieve this, we will use a recursive argument. First, the base case can be verified since when $t+1 = T+1$, the value functions are all zeros at $T+1$ step, i.e., $V^{\pi,\sigma_1}_{t+1}(s) = V^{\pi,\sigma_1}_{T+1}(s) = 0$ for all $s \in \mathcal{S}$. Then, the goal is to verify the following fact

$$V^{\pi,\sigma_1}_t([0,1]) = V^{\pi,\sigma_1}_t([1,0]) = 0 \tag{31}$$

with the assumption that $V^{\pi,\sigma_1}_{t+1}(s) = 0$ for any state $s = \{[0,1],[1,0]\}$. It is easily observed that for any policy $\pi$, the robust value function when state $s = \{[0,1],[1,0]\}$ at any time step $t$ obeys

$$\begin{aligned} 0 \le V^{\pi,\sigma_1}_t(s) &= \mathbb{E}_{a\sim\pi_t(\cdot\,|\,s)}\Bigg[r_t(s,a) + \inf_{P\in\mathcal{U}^{\sigma_1}(P^0_{t,s,a})} PV^{\pi,\sigma_1}_{t+1}\Bigg] \\ &\overset{\text{(i)}}{=} 0 + (1-\sigma_1)V^{\pi,\sigma_1}_{t+1}(s) + \sigma_1 V^{\pi,\sigma_1}_{\min,t+1} \overset{\text{(ii)}}{=} 0 + \sigma_1 V^{\pi,\sigma_1}_{\min,t+1} \end{aligned}$$

$$\leq 0 + \sigma_1 V_{t+1}^{\pi,\sigma_1}(s) = 0 \tag{32}$$

where (i) holds by $r_t(s,a) = 0$ for all $s = \{[0,1],[1,0]\}$, the fact $P_t^0(s \,|\, s,a) = 1$ (see equation 14 and equation 15), and the definition of the uncertainty set $\mathcal{U}^{\sigma_1}(P^0)$ in equation 23, (ii) follows from the recursive assumption $V_{t+1}^{\pi,\sigma_1}(s) = 0$ for any state $s = \{[0,1],[1,0]\}$, and the last equality holds by $V_{\min,t+1}^{\pi,\sigma_1} \leq V_{t+1}^{\pi,\sigma_1}(s)$ (see equation 29). Until now, we complete the proof for equation 31 and then verify equation 30.

Note that equation 30 direcly leads to

$$\forall 1 \leq t \leq T: \quad V_{\min,t}^{\pi,\sigma_1} = 0. \tag{33}$$

**Considering the robust value function at state** $[0,0]$**.** Armed with above facts, we are now ready to derive the robust value function for the state $[0,0]$.

When $2 \leq t \leq T$, one has

$$\begin{aligned}
V_t^{\pi,\sigma_1}([0,0]) &= \mathbb{E}_{a\sim\pi_t(\cdot \,|\, [0,0])}\left[r_t([0,0],a) + \inf_{P\in\mathcal{U}^{\sigma_1}(P_{t,[0,0],a})} PV_{t+1}^{\pi,\sigma_1}\right] \\
&\overset{\text{(i)}}{=} 1 + \left[(1-\sigma_1)V_{t+1}^{\pi,\sigma_1}([0,0]) + \sigma_1 V_{\min,t+1}^{\pi,\sigma_1}\right] \\
&= 1 + (1-\sigma_1)V_{t+1}^{\pi,\sigma_1}([0,0])
\end{aligned} \tag{34}$$

where (i) holds by $r_t([0,0],a) = 1$ for all $a \in \{0,1\}$ and the definition of $P^0$ (see equation 14 and equation 15), and the last equality arises from equation 33 .

Applying equation 34 recursively for $t, t+1, \cdots, T$ yields that

$$V_t^{\pi,\sigma_1}([0,0]) = \sum_{k=t}^{T}(1-\sigma_1)^{k-t} \geq 1. \tag{35}$$

At the first step, the robust value function obeys:

$$\begin{aligned}
V_1^{\pi,\sigma_1}([0,0]) &= \mathbb{E}_{a\sim\pi_1(\cdot \,|\, [0,0])}\left[r_t([0,0],a) + \inf_{P\in\mathcal{U}^{\sigma_1}(P_{1,[0,0],a})} PV_2^{\pi,\sigma_1}\right] \\
&\overset{\text{(i)}}{=} 1 + \pi_1(0 \,|\, [0,0]) \inf_{P\in\mathcal{U}^{\sigma_1}(P_{1,[0,0],0})} PV_2^{\pi,\sigma_1} + \pi_1(1 \,|\, [0,0]) \inf_{P\in\mathcal{U}^{\sigma_1}(P_{1,[0,0],1})} PV_2^{\pi,\sigma_1} \\
&\overset{\text{(ii)}}{=} 1 + \pi_1(0 \,|\, [0,0]) \left[(1-\sigma_1)V_2^{\pi,\sigma_1}([0,0]) + \sigma_1 V_{\min,2}^{\pi,\sigma_1}\right] \\
&\quad + \pi_1(1 \,|\, [0,0]) \left[(1-\sigma_1)V_2^{\pi,\sigma_1}([0,1]) + \sigma_1 V_{\min,2}^{\pi,\sigma_1}\right] \\
&= 1 + \pi_1(0 \,|\, [0,0])(1-\sigma_1)V_2^{\pi,\sigma_1}([0,0])
\end{aligned} \tag{36}$$

where (i) holds by $r_t([0,0],a) = 1$ for all $a \in \{0,1\}$, (ii) follows from the definition of $P^0$ (see equation 14 and equation 15), and the last equality arises from equation 30 and equation 33.

**The optimal policy** $\pi_{\mathsf{RMDP}}^{\star,\sigma_1}$**.** Observing that the positive value of $V_2^{\pi,\sigma_1}([0,0])$ verified in equation 35, as $V_1^{\pi,\sigma_1}([0,0])$ is increasing monotonically as $\pi_1(0 \,|\, [0,0])$ is larger, we directly have that $\pi_{\mathsf{RMDP}}^{\star,\sigma_1}(0 \,|\, [0,0]) = 1$.

Considering that the action does not influence the state transition for all other states $s \neq [0,0]$, without loss of generality, we choose the robust optimal policy to obey

$$\forall s \in \mathcal{S}: \quad \pi_{\mathsf{RMDP}}^{\star,\sigma_1}(0 \,|\, s) = 1. \tag{37}$$

#### C.2.2 Proof of Lemma 2

To begin with, for any uncertainty level $\sigma_2 \in (\frac{1}{2}, 1]$ and any policy $\pi = \{\pi_t\}$, we consider the robust SC-value function $\widetilde{V}_1^{\pi,\sigma_2}$ of the RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}}$.

**Deriving** $\widetilde{V}_t^{\pi,\sigma_2}$ **for** $2 \leq t \leq T$**.** Towards this, for any $2 \leq t \leq T$ and $s \in \mathcal{S}$, one has

$$\widetilde{V}_t^{\pi,\sigma_2}(s) \overset{\text{(i)}}{=} \inf_{P\in\mathcal{U}^{\sigma}(P^c)} \widetilde{V}_t^{\pi,P}(s) = \inf_{P\in\mathcal{U}^{\sigma}(P_t^c)} \mathbb{E}_{a\sim\pi_t(s)}\left[\widetilde{Q}_t^{\pi,P}(s,a)\right]$$

$$
\begin{aligned}
&\overset{\text{(ii)}}{=} \inf_{P \in \mathcal{U}^\sigma(P_t^c)} \mathbb{E}_{a\sim \pi_t(s)} \left[ r_t(s,a) + \mathbb{E}_{c_t \sim P} \left[ \mathcal{P}_{t,s,a,c_t} \widetilde{V}_{t+1}^{\pi,\sigma} \right] \right] \\
&\overset{\text{(iii)}}{=} r_t(s,a) + \inf_{P \in \mathcal{U}^\sigma(P_t^c)} \mathbb{E}_{c_t \sim P} \left[ \mathcal{P}_{t,s,a,c_t} \widetilde{V}_{t+1}^{\pi,\sigma} \right] \\
&= r_t(s,a) + \widetilde{V}_{t+1}^{\pi,\sigma}(s),
\end{aligned} \tag{38}
$$

where (i) holds by the definition in equation 5, (ii) follows from the *state-confounded* Bellman consistency equation in equation 47, (iii) follows from that the reward function $r$ and $\mathcal{P}_t$ are all independent from the action (see equation 16, equation 17 and equation 21), and the last inequality holds by $\mathcal{P}_t(s' \mid s,a,c) = \mathbb{1}(s' = s)$ is independent from $c_t$ (see equation 21).

Applying the above fact recursively for $t, t+1, \cdots, T$ leads to that for any $s \in \mathcal{S}$,

$$
\begin{aligned}
\widetilde{V}_t^{\pi,\sigma_2}(s) &= r_t(s,a_t) + \widetilde{V}_{t+1}^{\pi,\sigma}(s) = r_t(s,a) + r_{t+1}(s,a_{t+1}) + \widetilde{V}_{t+2}^{\pi,\sigma}(s) \\
&= \cdots = r_t(s,a_t) + \sum_{k=t+1}^{T} r_k(s_k,a_k),
\end{aligned} \tag{39}
$$

which directly yields

$$
\widetilde{V}_2^{\pi,\sigma_2}([0,0]) = \widetilde{V}_2^{\pi,\sigma_2}([1,1]) = T-1 \quad \text{and} \quad \widetilde{V}_2^{\pi,\sigma_2}([0,1]) = \widetilde{V}_2^{\pi,\sigma_2}([1,0]) = 0. \tag{40}
$$

**Characterizing $\widetilde{V}_1^{\pi,\sigma_2}([0,0])$ for any policy $\pi$.** In this section, we are especially interested in the value of $\widetilde{V}_1^{\pi,\sigma_2}$ on the state $[0,0]$. To proceed, one has

$$
\begin{aligned}
&\widetilde{V}_1^{\pi,\sigma_2}([0,0]) \overset{\text{(i)}}{=} \inf_{P \in \mathcal{U}^\sigma(P^c)} \widetilde{V}_1^{\pi,P}([0,0]) = \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{a \sim \pi_1([0,0])} \left[ \widetilde{Q}_1^{\pi,P}([0,0],a) \right] \\
&\overset{\text{(ii)}}{=} \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{a\sim \pi_t([0,0])} \left[ r_1([0,0],a) + \mathbb{E}_{c_t\sim P} \left[ \mathcal{P}_{1,[0,0],a,c_t} \widetilde{V}_2^{\pi,\sigma} \right] \right] \\
&\overset{\text{(iii)}}{=} 1 + \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{c_1 \sim P} \left[ \left( \pi_1(0 \,|\, [0,0]) \mathcal{P}_{1,[0,0],0,c_1} + \pi_t(1 \,|\, [0,0]) \mathcal{P}_{1,[0,0],1,c_1} \right) \widetilde{V}_2^{\pi,\sigma} \right] \\
&\overset{\text{(iv)}}{=} 1 + \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{c_1 \sim P} \Bigg[ \pi_1(0 \,|\, [0,0]) \left( (1-c_1) P^0_{1,[0,0],0} + c_1 P^{\text{sc}}_{1,[0,0],0} \right) \widetilde{V}_2^{\pi,\sigma} \\
&\quad + \pi_1(1 \,|\, [0,0]) \left( (1-c_1) P^0_{1,[0,0],1} + c_1 P^{\text{sc}}_{1,[0,0],1} \right) \widetilde{V}_2^{\pi,\sigma} \Bigg] \\
&\overset{\text{(v)}}{=} 1 + \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{c_1 \sim P} \Bigg[ \pi_1(0 \,|\, [0,0]) \left( (1-c_1) \widetilde{V}_2^{\pi,\sigma}([0,0]) + c_1 \widetilde{V}_2^{\pi,\sigma}([1,0]) \right) \\
&\quad + \pi_1(1 \,|\, [0,0]) \left( (1-c_1) \widetilde{V}_2^{\pi,\sigma}([0,1]) + c_1 \widetilde{V}_2^{\pi,\sigma}([1,1]) \right) \Bigg] \\
&= 1 + (T-1) \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{c_1 \sim P} \Big[ \pi_1(0 \,|\, [0,0])(1-c_1) + \pi_1(1 \,|\, [0,0]) c_1 \Big] \\
&= 1 + (T-1)\pi_1(0 \,|\, [0,0]) + (T-1) \inf_{P \in \mathcal{U}^\sigma(P_1^c)} \mathbb{E}_{c_1 \sim P} \Big[ c_1 \big( 1 - 2\pi_1(0 \,|\, [0,0]) \big) \Big],
\end{aligned} \tag{41}
$$

where (i) holds by the definition in equation 5, (ii) follows from the *state-confounded* Bellman consistency equation in equation 47, (iii) follows from $r_1([0,0],a) = 1$ for all $a \in \{0,1\}$ which is independent from $c_t$. (iv) arises from the definition of $\mathcal{P}$ in equation 20, (v) can be verified by plugging in the definitions from equation 14 and equation 18, and the penultimate equality holds by equation 40.

**Characterizing the optimal robust SC-value functions.**

To further consider equation 41, we recall the fact that $\mathcal{U}^\sigma(P_1^c) = \left\{ P \in \Delta(\mathcal{C}) : \frac{1}{2} \left\| P - P_1^c \right\|_1 \leq \sigma_2 \right\}$.

Observing from equation 41 that for any fixed $\pi_1(0\,|\,[0,0])$, $c_1\big(1-2\pi_1(0\,|\,[0,0])\big)$ is monotonously increasing with $c_1$ when $1-2\pi_1(0\,|\,[0,0]) >= 0$ and decreasing with $c_1$ otherwise, it is easily verified that the solution of

$$f(\pi_1(0\,|\,[0,0])) \coloneqq (T-1) \inf_{P\in\mathcal{U}^{\sigma}(P_1^c)} \mathbb{E}_{c_1\sim P}\left[c_1\big(1-2\pi_1(0\,|\,[0,0])\big)\right] \tag{42}$$

satisfies

$$f(\pi_1(0\,|\,[0,0])) = \begin{cases} 0 & \text{if } \pi_1(0\,|\,[0,0]) \geq \frac{1}{2} \\ (T-1)\sigma_2\big(1-2\pi_1(0\,|\,[0,0])\big) & \text{otherwise} \end{cases}. \tag{43}$$

And note that the value of $\widetilde{V}_1^{\pi,\sigma_2}([0,0])$ only depends on $\pi_1(\cdot\,|\,[0,0])$ which can be represent by $\pi_1(0\,|\,[0,0])$. Plugging in equation 43 into equation 41, we have that when $\pi_1(0\,|\,[0,0]) \geq \frac{1}{2}$,

$$\begin{aligned} \max_{\pi} \widetilde{V}_1^{\pi,\sigma_2}([0,0]) &= \max_{\pi_1(0\,|\,[0,0])\geq\frac{1}{2}} 1 + (T-1)\pi_1(0\,|\,[0,0]) + (T-1)\sigma_2\big(1-2\pi_1(0\,|\,[0,0])\big) \\ &= 1 + (T-1)\sigma_2 + (T-1) \max_{\pi_1(0\,|\,[0,0])\geq\frac{1}{2}} (1-2\sigma_2)\pi_1(0\,|\,[0,0]) \\ &= 1 + \frac{T-1}{2}, \end{aligned} \tag{44}$$

where the last equality holds by $\sigma_2 > \frac{1}{2}$ and letting $\pi_1(0\,|\,[0,0]) = \frac{1}{2}$. Similarly, when $\pi_1(0\,|\,[0,0]) < \frac{1}{2}$,

$$\max_{\pi} \widetilde{V}_1^{\pi,\sigma_2}([0,0]) = \max_{\pi_1(0\,|\,[0,0])<\frac{1}{2}} 1 + (T-1)\pi_1(0\,|\,[0,0]) < 1 + \frac{T-1}{2}. \tag{45}$$

Consequently, we complete the proof by concluding that

$$\widetilde{V}_1^{\pi^{\star,\sigma_2}_{\mathsf{RSC}},\sigma_2}([0,0]) = \widetilde{V}_1^{\star,\sigma_2}([0,0]) = \max_{\pi} \widetilde{V}_1^{\pi,\sigma_2}([0,0]) = 1 + \frac{T-1}{2}. \tag{46}$$

### C.3 Auxiliary results of SC-MDPs and RSC-MDPs

**Facts about SC-MDPs.** For any state-confounded MDPs (SC-MDPs) $\mathcal{M}_{\mathsf{SC}} = \big\{\mathcal{S},\mathcal{A},T,r,\mathcal{C},\{\mathcal{P}_t^i\},P^c\big\}$, denoting the optimal policy as $\pi^\star$ and the corresponding optimal SC-value function as $\widetilde{V}$, any policy $\pi$ satisfies the corresponding *state-confounded* Bellman consistency equation as below:

$$\widetilde{Q}_t^{\pi,P^c}(s,a) = r_t(s,a) + \mathbb{E}_{c_t\sim P_t^c}\left[\mathcal{P}_{t,s,a,c_t}\widetilde{V}_{t+1}^{\pi,\sigma}\right], \tag{47}$$

where $\mathcal{P}_{t,s,a,c_t} \in \mathbb{R}^{1\times S}$ such that $\mathcal{P}_{t,s,a,c_t}(s') \coloneqq \mathcal{P}_t(s'\,|\,s,a,c_t)$ for $s'\in\mathcal{S}$.

**Facts about RSC-MDPs.** It is easily verified that for any RSC-MDP $\mathcal{M}_{\mathsf{sc\text{-}rob}} = \big\{\mathcal{S},\mathcal{A},T,r,\mathcal{C},\{\mathcal{P}_t^i\},\mathcal{U}^{\sigma_2}(P^c)\big\}$, any policy $\pi$ and the optimal policy $\pi^\star$ satisfy the corresponding *robust state-confounded* Bellman consistency equation and Bellman optimality equation shown below, respectively:

$$\begin{aligned} \widetilde{Q}_t^{\pi,\sigma}(s,a) &= r_t(s,a) + \inf_{P\in\mathcal{U}^{\sigma}(P_t^c)} \mathbb{E}_{c_t\sim P}\left[\mathcal{P}_{t,s,a,c_t}\widetilde{V}_{t+1}^{\pi,\sigma}\right], \\ \widetilde{Q}_t^{\star,\sigma}(s,a) &= r_t(s,a) + \inf_{P\in\mathcal{U}^{\sigma}(P_t^c)} \mathbb{E}_{c_t\sim P}\left[\mathcal{P}_{t,s,a,c_t}\widetilde{V}_{t+1}^{\star,\sigma}\right], \end{aligned} \tag{48}$$

where $\mathcal{P}_{t,s,a,c_t} \in \mathbb{R}^{1\times S}$ such that $\mathcal{P}_{t,s,a,c_t}(s') \coloneqq \mathcal{P}_t(s'\,|\,s,a,c_t)$ for $s'\in\mathcal{S}$, and $\widetilde{V}_{t+1}^{\star,\sigma}(s) = \max_a \widetilde{Q}_{t+1}^{\star,\sigma}(s,a)$.

# D Experiment Details

### D.1 Architecture of the structural causal model

We plot the architecture of the structural causal model we used in our method in Figure 6. In normal neural networks, the input is treated as a whole to pass through linear layers or convolution layers.

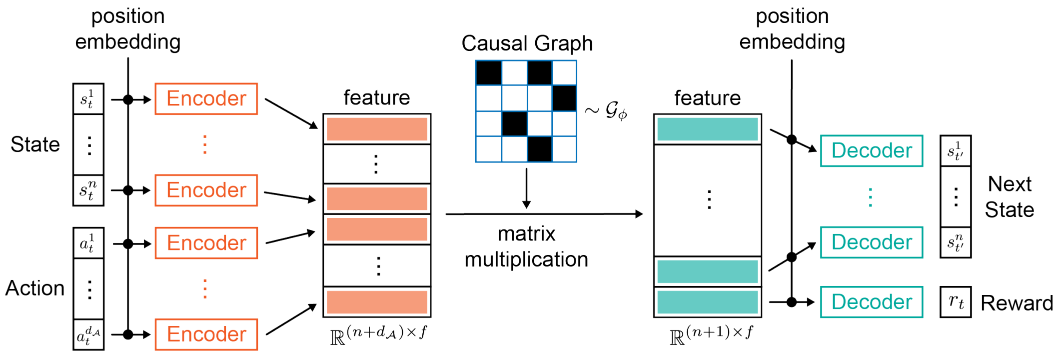


Figure 6: Model architecture of the structural causal model. Encoder, Decoder, position embedding, and Causal Graph are learnable during the training stage.

However, this structure blends all information in the input, making the causal graph useless to separate cause and effect. Thus, in our model, we design an encoder that is shared across all dimensions of the input. Since different dimensions could have exactly the same values, we add a learnable position embedding to the input of the encoder. In summary, the input dimension of the encoder is $1 + d_{pos}$, where $d_{pos}$ is the dimension of the position embedding.

After the encoder, we obtain a set of independent features for each dimension of the input. We now multiply the features with a learnable binary causal graph $\boldsymbol{G}$. The element $(i, j)$ of the graph is sampled from a Gumbel-Softmax distribution with parameter $\phi_{i,j}$ to ensure the loss function is differentiable w.r.t $\phi$.

The multiplication of the causal graph and the input feature creates a linear combination of the input feature with respect to the causal graph. The obtained features are then passed through a decoder to predict the next state and reward. Again, the decoder is shared across all dimensions to avoid information leaking between dimensions. Position embedding is included in the input to the decoder and the output dimension of the decoder is 1.

### D.2 Environments

We design four self-driving tasks in the Carla simulator [22] and four manipulation tasks in the Robosuite platform [23]. All of these realistic tasks contain strong spurious correlations that are explicit to humans. We provide detailed descriptions of all these environments in the following.

**Brightness.** The nominal environments are shown in the 1th column of Figure 7, where the brightness and the traffic density are correlated. When the ego vehicle drives in the daytime, there are many surrounding vehicles (first row). When the ego vehicle drives in the evening, there is no surrounding vehicle (second row). The shifted environment swaps the brightness and traffic density in the nominal environment, i.e., many surrounding vehicles in the evening and no surrounding vehicles in the daytime.

**Behavior.** The nominal environments are shown in the 2nd column of Figure 7, where the other vehicle has aggressive driving behavior. When the ego vehicle is in front of the other vehicle, the other vehicle always accelerates and overtakes the ego vehicle in the left lane. When the ego vehicle is behind the other vehicle, the other vehicle will always accelerate. In the shifted environment, the behavior of the other vehicle is conservative, i.e., the other vehicle always decelerates to block the ego vehicle.

**Crossing.** The nominal environments are shown in the 3rd column of Figure 7, where the pedestrian follows the traffic rule and only cross the road when the traffic light is green. In the shifted environment, the pedestrian disobeys the traffic rule and crosses the road when the traffic light is red.

**CarType.** The nominal environments are shown in the 4th column of Figure 7, where the type of vehicle and the speed of the vehicle are correlated. When the vehicle is a truck, the speed is low and when the vehicle is a motorcycle, the speed is high. In the shifted environment, the truck drives very fast and the motorcycle drives very slow.

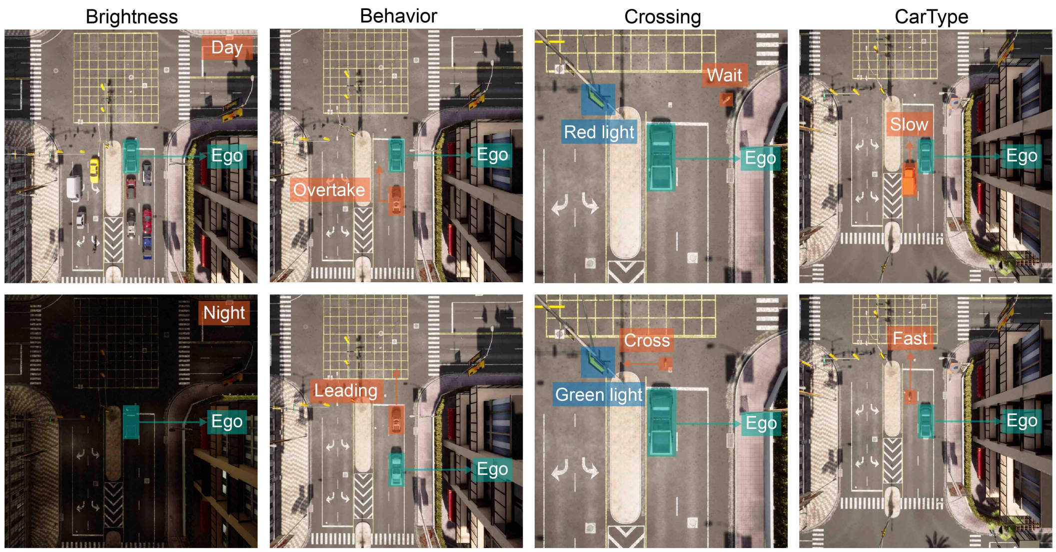


Figure 7: Illustration of tasks in the Carla simulator.

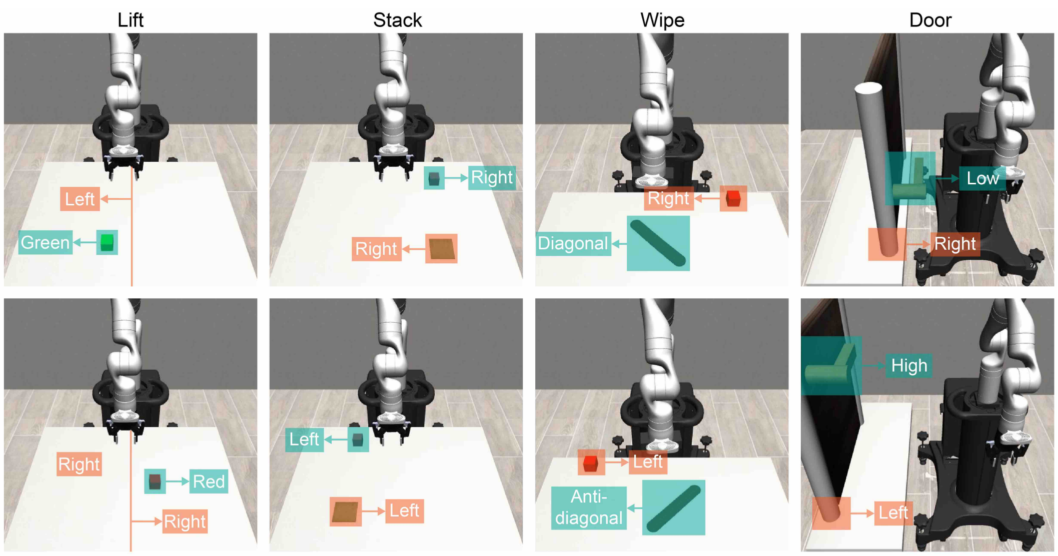


Figure 8: Illustration of tasks in the Robosuite simulator.

**Lift.** The nominal environments are shown in the 1$^{\text{th}}$ column of Figure 8, where the position of the cube and the color of the cube are correlated. When the cube is in the left part of the table, the color of the cube is green, when the cube is in the right part of the table, the color of the cube is red. The shifted environment swaps the color and position of the cube in the nominal environment, i.e., the cube is green when it is in the right part and the cube is red when it is in the left part.

**Stack.** The nominal environments are shown in the 2$^{\text{nd}}$ column of Figure 8, where the position of the red cube and green plate are correlated. When the cube is in the left part of the table, the plate is also in the left part; when the cube is in the right part of the table, the plate is also in the right part. In the shifted environment, the relative position of the cube and the plate changes, i.e., When the cube is in the left part of the table, the plate is in the right part; when the cube is in the right part of the table, the plate is in the left part.

**Wipe.** The nominal environments are shown in the 3$^{\text{rd}}$ column of Figure 8, where the shape of the dirty region is correlated to the position of the cube. When the dirty region is diagonal, the cube is on the right-hand side of the robot arm. When the dirty region is anti-diagonal, the cube is on the left-hand side of the robot arm. In the shifted environment, the correlation changes, i.e., when the dirty region is diagonal, the cube is on the left-hand side of the robot arm; when the dirty region is anti-diagonal, the cube is on the right-hand side of the robot arm.

**Door.** The nominal environments are shown in the $4^{\text{th}}$ column of Figure 8, where the height of the handle and the position of the door is correlated. When the door is closed to the robot arm, the handle is in a low position. When the door is far from the robot arm, the handle is in a high position. In the shifted environment, the correlation changes, i.e., when the door is closed to the robot arm, the handle is in a high position; when the door is far from the robot arm, the handle is in a low position.

### D.3 Computation resources

Our algorithm is implemented on top of the Tianshou [88] package. All of our experiments are conducted on a machine with an Intel i9-9900K CPU@3.60GHz (16 core) CPU, an NVIDIA GeForce GTX 1080Ti GPU, and 64GB memory.

### D.4 Hyperparameters

We summarize all hyper-parameters used in the Carla experiments (Table 5) and Robosuite experiments (Table 6). The source code of experiments will be released after double-blind reviewing.

Table 5: Hyper-parameters in Carla experiments

| Parameters | Notation | Environment | | | |
|---|---|---|---|---|---|
| | | Brightness | Behavior | Crossing | CarType |
| Horizon steps | $T$ | 100 | 100 | 100 | 100 |
| State dimension | $n$ | 24 | 12 | 12 | 12 |
| Action dimension | $d_{\mathcal{A}}$ | 2 | 2 | 2 | 2 |
| Max training steps | | $1\times10^5$ | $1\times10^5$ | $5\times10^5$ | $5\times10^5$ |
| Weight of $\|\|\boldsymbol{G}\|\|_p$ | $\lambda$ | 0.1 | - | - | - |
| norm of $\|\|\boldsymbol{G}\|\|_p$ | $p$ | 0.1 | - | - | - |
| Actor learning rate | | $3 \times 10^{-4}$ | - | - | - |
| Critic learning rate | | $1 \times 10^{-3}$ | - | - | - |
| Batch size | | 256 | - | - | - |
| Discount factor | $\gamma$ in SAC | 0.99 | - | - | - |
| Soft update weight | $\tau$ in SAC | 0.005 | - | - | - |
| Weight of entropy | $\alpha$ in SAC | 0.1 | - | - | - |
| Hidden layers | | [256, 256, 256] | - | - | - |
| Returns estimation step | | 4 | - | - | - |
| Buffer size | | $1 \times 10^5$ | - | - | - |
| Steps per update | | 10 | - | - | - |

### D.5 Discovered Causal Graph in SCM

To show the performance of our learned SCM, we plot the estimated causal graphs of all experiments in Figure 9, Figure 10, Figure 11, Figure 12, and Figure 13.

Table 6: Hyper-parameters in Robosuite experiments

| Parameters | Notation | Environment | | | |
|---|---|---|---|---|---|
| | | Lift | Stack | Door | Wipe |
| Horizon steps | $T$ | 300 | 300 | 300 | 500 |
| Control frequency (Hz) | | 20 | 20 | 20 | 20 |
| State dimension | $n$ | 50 | 110 | 22 | 30 |
| Action dimension | $d_{\mathcal{A}}$ | 4 | 4 | 8 | 7 |
| Controller type | | OSC position | OSC position | Joint velocity | Joint velocity |
| Max training steps | | $1\times10^6$ | $5\times10^6$ | $1\times10^6$ | $1\times10^6$ |
| Weight of $\|\boldsymbol{G}\|_p$ | $\lambda$ | 0.01 | - | - | - |
| norm of $\|\boldsymbol{G}\|_p$ | $p$ | 0.1 | - | - | - |
| Actor learning rate | | $3 \times 10^{-4}$ | - | - | - |
| Critic learning rate | | $1 \times 10^{-3}$ | - | - | - |
| Batch size | | 128 | - | - | - |
| Discount factor | $\gamma$ in SAC | 0.99 | - | - | - |
| Soft update weight | $\tau$ in SAC | 0.005 | - | - | - |
| alpha learning rate | $lr_\alpha$ in SAC | $3 \times 10^{-4}$ | - | - | - |
| Hidden layers | | [256, 256, 256] | - | - | - |
| Returns estimation step | | 4 | - | - | - |
| Buffer size | | $1 \times 10^6$ | - | - | - |
| Steps per update | | 10 | - | - | - |

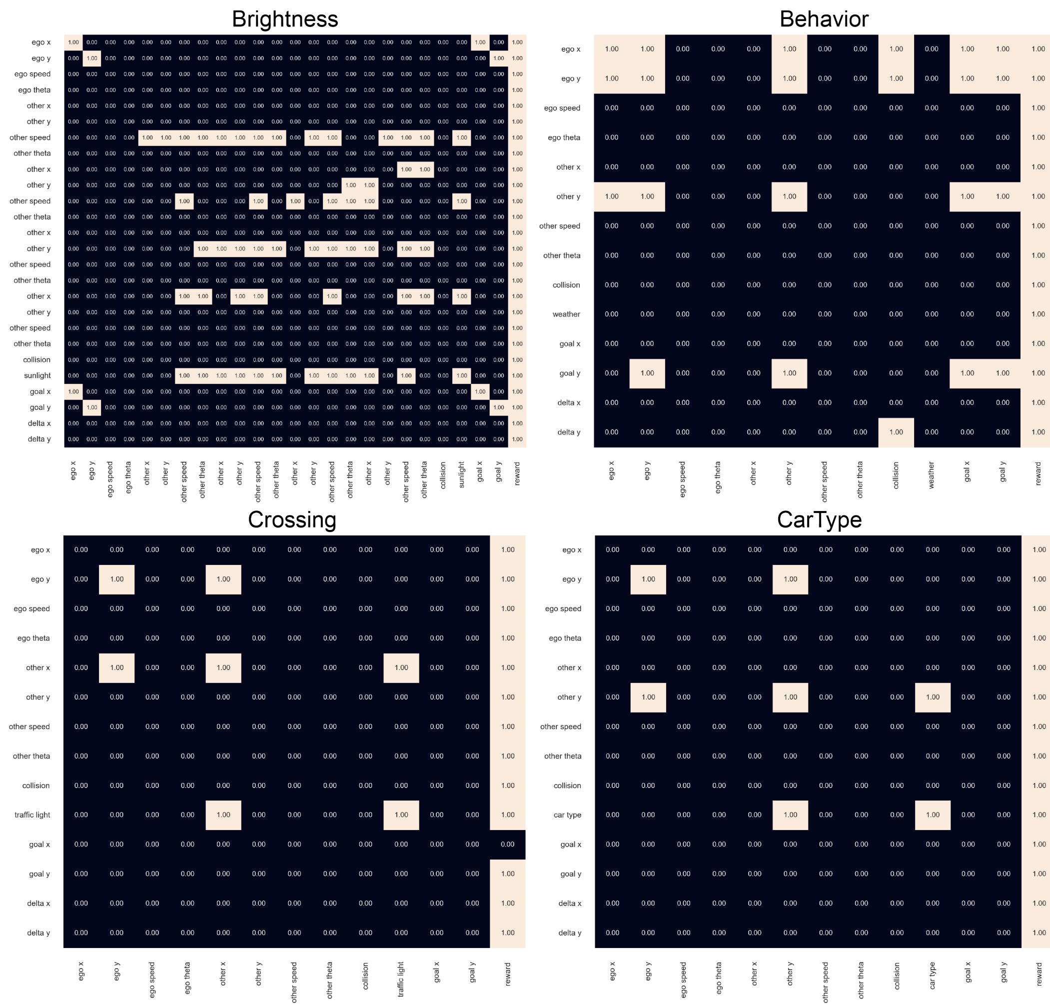


Figure 9: Estimated Causal Graphs of four tasks in Carla.

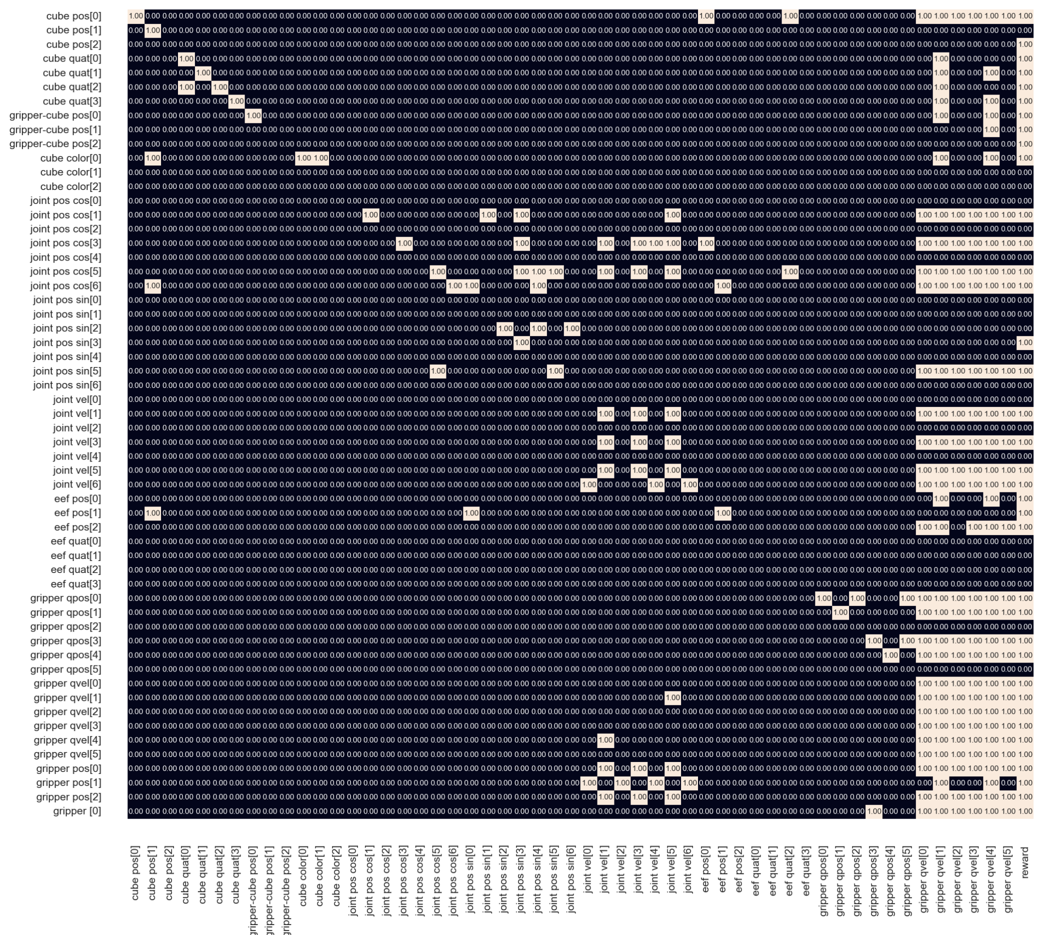


Figure 10: Estimated Causal Graphs of the Lift task in Robosuite.

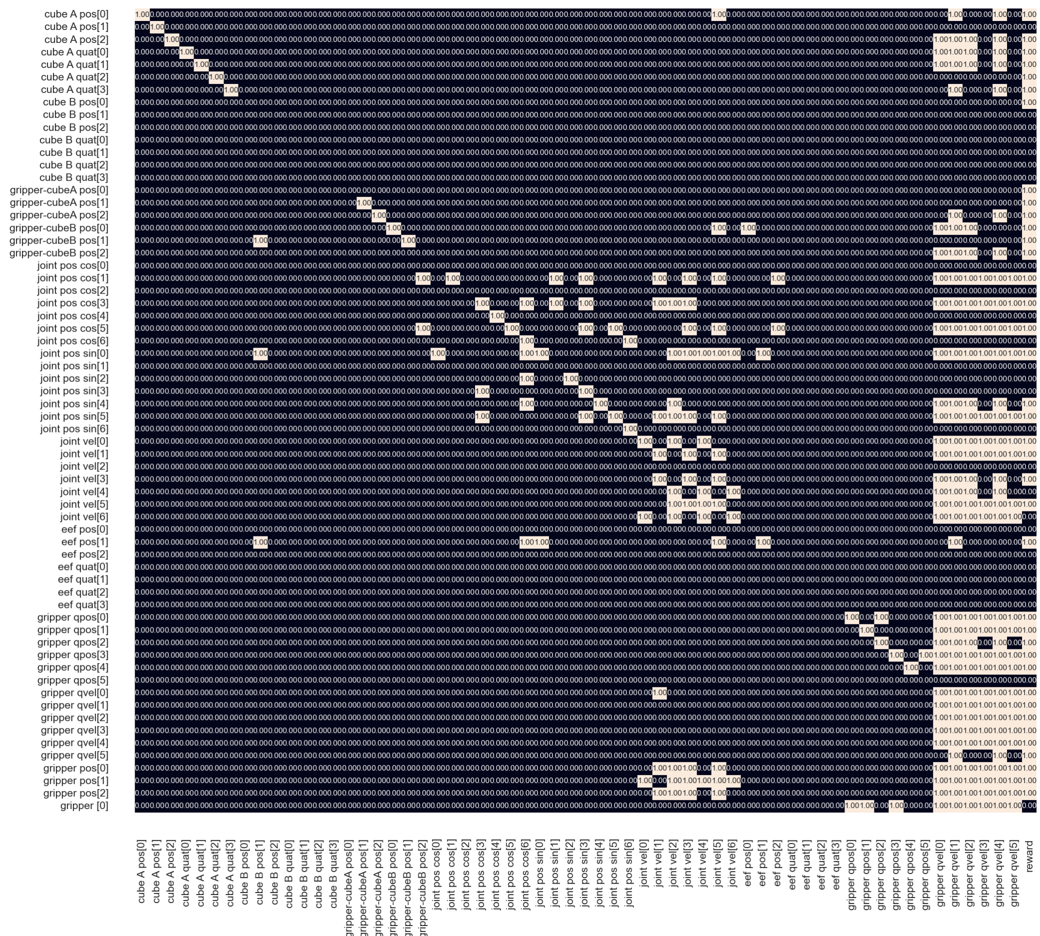

Figure 11: Estimated Causal Graphs of the Stack task in Robosuite.

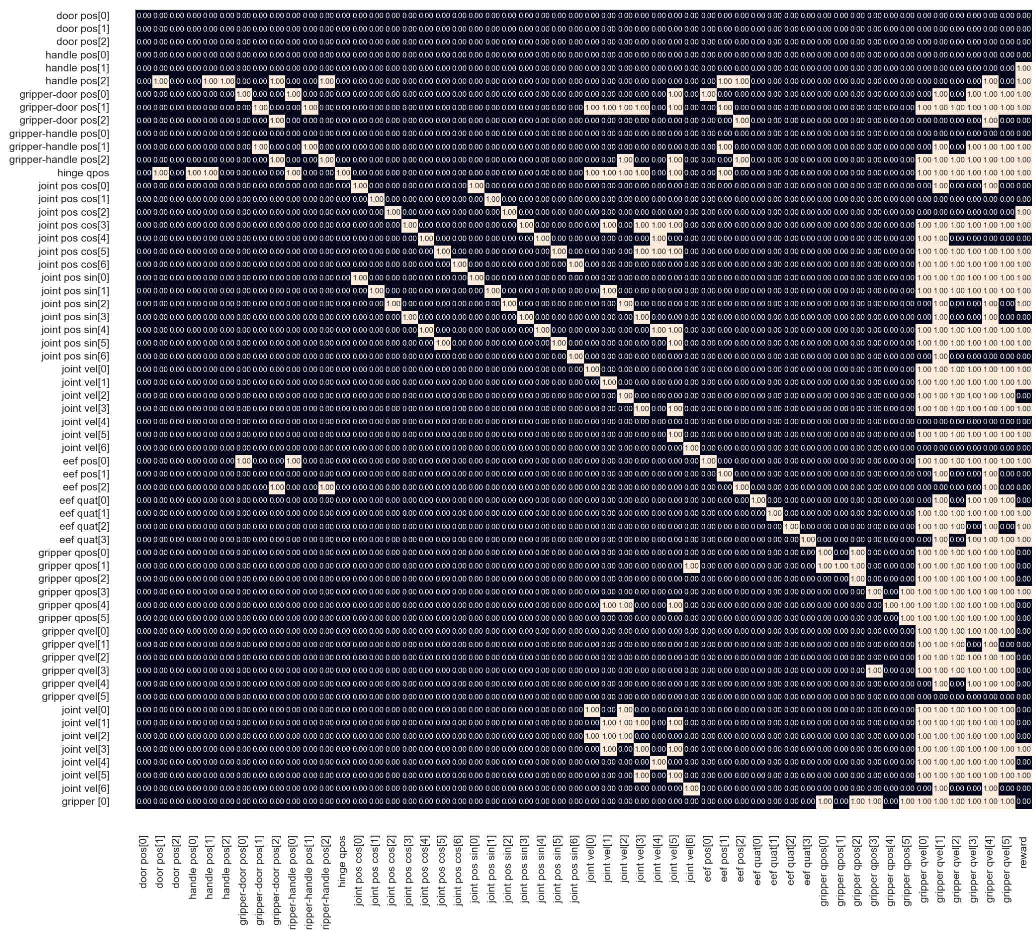

Figure 12: Estimated Causal Graphs of the Door task in Robosuite.

Figure 13: Estimated Causal Graphs of the Wipe task in Robosuite.