Bad Habits: Policy Confounding and Out-of-Trajectory Generalization in RL

Miguel Suau
Phaidra
Delft University of Technology
miguel.suau@phaidra.ai

Matthijs T. J. Spaan
Delft University of Technology
m.t.j.spaan@tudelft.nl

Frans A. Oliehoek
Delft University of Technology
f.a.oliehoek@tudelft.nl

Abstract

Reinforcement learning agents tend to develop habits that are effective only under specific policies. Following an initial exploration phase where agents try out different actions, they eventually converge onto a particular policy. As this occurs, the distribution over state-action trajectories becomes narrower, leading agents to repeatedly experience the same transitions. This repetitive exposure fosters spurious correlations between certain observations and rewards. Agents may then pick up on these correlations and develop simplistic habits tailored to the specific set of trajectories dictated by their policy. The problem is that these habits may yield incorrect outcomes when agents are forced to deviate from their typical trajectories, prompted by changes in the environment. This paper presents a mathematical characterization of this phenomenon, termed policy confounding, and illustrates, through a series of examples, the circumstances under which it occurs.

1 Introduction

This morning, I went to the kitchen for a coffee. When I arrived,
I forgot why I was there, so I got myself a coffee.—

How often do you do something without paying attention to your actions? Have you ever caught yourself lost in thought while washing the dishes, making coffee, or cycling? Acting out of habit is a crucial human skill as it enables us to focus on more important matters while executing routine tasks. You can commute to work while thinking about how to persuade your boss to give you a salary raise or prepare dinner while daydreaming about your next holiday in the Alps. However, habits can also lead to unintended consequences when we fail to recognize that the context has changed. You might hop in your car and drive toward work even though it is a Sunday and you want to go to the grocery store, or you might flip the switch when leaving a room even though the lights are already off.

Surprisingly, reinforcement learning (RL) agents also struggle with this same issue. This is due to a phenomenon we term *policy confounding*, which reflects how policies, as a result of influencing past and future observation variables, can inadvertently induce spurious correlations. These correlations can lead to the development of seemingly sensible but incorrect habits, such as flipping the switch upon leaving a room, without confirming whether the lights are on. The problem here is that these habits can produce incorrect results when agents are forced to deviate from their usual trajectories due to changes in the environment; a problem we refer to as *out-of-trajectory* (OOT) generalization.

Contributions This paper introduces and characterizes the phenomenon of *policy confounding*. To do so, we provide a mathematical framework that helps us describe the different types of state representations, and reveal how, as a result of policy confounding, the agent may learn representations based on spurious correlations that do not guarantee OOT generalization. Moreover, we include a series of clarifying examples that illustrate how this occurs. Unfortunately, we do not have a complete

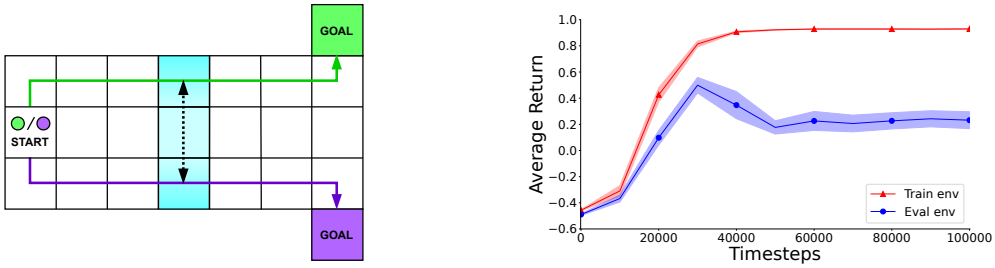

Figure 1: Left: An illustration of the Frozen T-Maze environment. Right: Learning curves when evaluated in the Frozen T-Maze environment with (blue curve) and without (red curve) ice.

answer for how to prevent policy confounding. However, we suggest a few off-the-shelf solutions that may help mitigate its effects. We hope this paper will create awareness among the RL community about the risks of policy confounding and inspire further research on this topic.

2 Example: Frozen T-Maze

We now provide an example to illustrate the phenomenon and motivate the need for careful analysis. Figure 1 shows a variant of the popular T-Maze environment (Bakker, 2001). In this environment, the agent receives a binary signal, green or purple, at the starting location. The task for the agent is to navigate to the right and reach the correct goal at the end of the maze. The agent obtains a reward of $+1$ for moving to the green (purple) goal when having received the green (purple) signal and a penalty of -1 otherwise. Additionally, there is a -0.1 penalty per timestep to incentivize the agent to take the most direct path to the goal. At every timestep, the agent can observe its location within the maze, but the signal is only observed at the starting location. Importantly, the agent has the capability to remember past observations.

At first glance, it might seem crucial for the agent to remember the initial signal at every cell along its trajectory. After all, how else would it determine the correct goal? However, once the agent figures out the shortest path to each of the two goals (depicted by the green and purple arrows), the agent may safely forget the initial signal. The agent knows that whenever it is on the green (purple) path, it must have received the green (purple) signal. Hence, it can simply navigate toward the correct goal based solely on its current location. Sticking to this habit is optimal so long as the agent commits to these two paths. It is also essential that the environment dynamics remain the same as any changes in the agent's trajectories could erase the spurious correlation (Pearl et al., 2016) induced by the policy between the agent's location and the correct goal.[1]

To demonstrate that this occurs in practice, we trained agents with PPO (Schulman et al., 2017) in the original environment (train env) and evaluated them in a modified version (eval env), where there is an icy surface (depicted in blue) in the middle of the maze. The ice causes the agent to slip from the upper cell to the bottom cell and vice versa.[2] The plot on the right of Figure 1 shows the return averaged over 10 trials. The performance drop in the evaluation environment (blue curve) suggests that the agents' policies fail to generalize to alternative trajectories within the same environment. The ice confuses the agents, who, after being pushed away from their preferred trajectories, can no longer choose the correct goal. More details about this experiment are provided in Section 7.

3 Related Work

The presence of spurious correlations in the training data is a well-studied problem in machine learning. These correlations often provide convenient shortcuts that a model can exploit to make

[1]Note that the two paths highlighted in Figure 1 are not the only optimal paths. However, for the agent to be able to ignore the initial signal, the paths must not overlap.

[2]The ice compels the agent to take alternate trajectories by causing it to move down twice from the top cell or up twice from the bottom cell. Importantly, these trajectories are feasible within the original environment.

predictions (Beery et al., 2018). However, a model relying on these shortcuts may experience significant performance degradation when faced with different data distributions (Quionero-Candela et al., 2009). Langosco et al. (2022) show that RL agents may use certain environment features as proxies for choosing their actions. These features are intentionally introduced in the training environments to artificially correlate with the agent's objectives. In contrast, our work demonstrates that agents, due to policy confounding, may actively contribute to the formation of spurious correlations.

Previous studies have reported empirical evidence of specific forms of policy confounding, revealing that in deterministic environments, agents can utilize information that correlates with their progress in an episode to determine optimal actions. This strategy is effective because, under fixed policies, features like timers (Song et al., 2020), agent postures (Lan et al., 2023), or previous action sequences (Machado et al., 2018) can be directly mapped to the agent's state. While these studies offer various hypotheses to explain their experimental observations, we contribute an overarching theory that explains the underlying causes and mechanisms behind these results, along with a series of examples illustrating other types of policy confounding.

Out-of-trajectory (OOT) generalization is a particular instance of the more general problem of out-of-distribution (OOD) generalization in RL (Kirk et al., 2023). The objective of OOT generalization is not to generalize to environments with different rewards (Taylor & Parr, 2009), observations (Mandlekar et al., 2017; Zhang et al., 2020), and transitions (Higgins et al., 2017) but simply to alternative trajectories within the same environment. In our experiments, agents are evaluated in altered environments with different dynamics. These alterations are only intended to force the agent to take alternative trajectories within the same environment. Importantly, these alternative trajectories are both possible and probable in the original environment. Example 5 illustrates the distinction between OOT and OOD. Please refer to Appendix C for more details on related work.

4 Preliminaries

4.1 Notation

We denote random variables with capital letters (e.g., S), their corresponding values with lowercase letters (e.g., s), and their domains with calligraphic letters (e.g., $\mathcal{S}$). To denote the domain of a set of random variables $F = \{F^1, \dots, F^{|F|}\}$, we use $\times\mathcal{F}$ as a shorthand for the Cartesian product $\mathcal{F}^1 \times \cdots \times \mathcal{F}^{|F|}$. This notation represents all possible combinations of values for the variables in F.

4.2 Problem formulation

Definition 1 (MDP)**.** A Markov decision process (MDP) is a tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where $\mathcal{S}$ represents the set of states, $\mathcal{A}$ denotes the set of actions available to the agent, $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transitions function, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function.

In particular, we focus on problems where states are represented by a set of observation variables, or factors (Boutilier et al., 1999). This representation is common when modeling policies and value functions using function approximators (Sutton & Barto, 2018). These observation variables typically describe features of the agent's state in the environment.

Definition 2 (FMDP)**.** A Factored Markov decision process (FMDP) is an MDP where the set of states is defined by a set of observation variables, or factors, $F = \{F^1, \dots, F^{|F|}\}$. Each variable F^i can take any of the values in its domain $\mathcal{F}^i$. Consequently, each state s corresponds to a unique combination of values for the variables in F, $s = \langle f^1, \dots, f^{|F|} \rangle \in \times\mathcal{F} = \mathcal{S}$.

While, for simplicity, we employ the MDP formulation, the insights presented here are not exclusive to fully observable environments. In cases where the current observation variables F do not satisfy the Markov property, F is considered to be the history of action and observation variables, which is guaranteed to satisfy the Markov property (Kaelbling et al., 1998).

5 State representations

The agent's objective is to learn a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected discounted sum of rewards (Sutton & Barto, 2018). However, learning a policy that conditions on every observation variable might be impractical, particularly in scenarios with a large number of variables. Fortunately, in many problems, not all variables are strictly essential, and compact state representations can be found that are sufficient for solving the task at hand (McCallum, 1995). This is where function approximators, such as neural networks, come into the picture (François-Lavet et al., 2018; Ni et al., 2024). If we use them to model policies and value functions, they will learn to ignore certain observation variables in F if they are deemed unnecessary for estimating values and optimal actions.

As we shall see, the phenomenon of policy confounding plays a fundamental role in this quest for simpler state representations, tricking the function approximator into forming state representations that are based on mere spurious correlations. Before delving into these intricacies, let us establish some key definitions regarding state representations. To enhance clarity, we will use the environment introduced in Section 2 as a running example throughout this and the next section.

Example 1. Refer to the first paragraph of Section 2 for a description of the environment. We denote the agent's location by L. The variable G indicates whether the goal is to reach the green or purple cell. G is sampled at the beginning and its value remains constant throughout the episode. The value of G is hidden and only passed to the agent at the starting location through the variable X, representing the signal. At any other location, X takes a dummy value, rendering the environment partially observable. Consequently, the set F_t is the history of actions, locations, and signal variables, $F_t = \{L_0, X_0, A_0, \ldots, A_{t-1}, L_t, X_t\}$. Each unique combination of values defines a state s_t. Here, the subscript t is used to denote that the number of variables in F depends on t.

Definition 3 (State representation)**.** A state representation is a function $\Phi : \mathcal{S} \to \bar{\mathcal{S}}$, where $\mathcal{S} = \times\mathcal{F}$, $\bar{\mathcal{S}} = \times\bar{\mathcal{F}}$, and $\bar{F} \subseteq F$.

Intuitively, a state representation $\Phi(s_t)$ is a state-specific projection of a state $s \in \mathcal{S} = \times\mathcal{F}$ onto a lower-dimensional space $\bar{S} = \times\bar{\mathcal{F}}$ defined by a subset of its variables, $\bar{F} \subseteq F$. We use $\{s\}^{\Phi} = \{s' \in \mathcal{S} : \Phi(s') = \Phi(s)\}$ to denote the equivalence class of s under Φ. In Example 1, a potential state representation could be $\Phi(s_t) = \langle l_0, x_0 \rangle$ for all $s_t \in \mathcal{S}$. This representation retains only L_0 and X_0, ignoring all other variables in F. Hence, all states that share the same values for L_0 and X_0 belong to the same equivalence class.

5.1 Markov state representations

Not all state representations are sufficient to learn the optimal policy; some, like the one discussed in the above paragraph, may exclude variables that carry valuable information for the task at hand.

Definition 4 (Markov state representation)**.** A state representation $\Phi(s_t)$ is said to be Markov if, for all $s_t, s_{t+1} \in \mathcal{S}$, $a_t \in \mathcal{A}$,

$$R(s_t, a_t) = R(\Phi(s_t), a_t) \quad \text{and} \quad \sum_{s'_{t+1} \in \{s_{t+1}\}^{\Phi}} T(s'_{t+1} \mid s_t, a_t) = \Pr(\Phi(s_{t+1}) \mid \Phi(s_t), a_t),$$

where $R(\Phi(s_t), a_t)$ denotes the reward $R(s'_t, a_t)$ at any $s'_t \in \{s_t\}^{\Phi}$.

The above definition is analogous to the notion of bisimulation (Dean & Givan, 1997; Givan et al., 2003) or model-irrelevance state abstraction (Li et al., 2006). Representations satisfying these conditions are guaranteed to be behaviorally equivalent to the original representation. That is, for any given policy and initial state, the expected return (i.e., cumulative reward; Sutton & Barto, 2018) is the same when conditioning on the full set of observation variables F or on the Markov state representation Φ.

Definition 5 (Minimal state representation)**.** A state representation $\Phi^* : \mathcal{S} \to \bar{\mathcal{S}}^*$ with $\bar{\mathcal{S}}^* = \times\bar{\mathcal{F}}^*$ is said to be *minimal*, if all other state representations $\Phi : \mathcal{S} \to \bar{\mathcal{S}}$ with $\bar{\mathcal{S}} = \times\bar{\mathcal{F}}$ and $\bar{F} \subset \bar{F}^*$, for some $s \in \mathcal{S}$, are not Markov.

In simpler terms, Φ^* is *minimal* when none of the remaining variables can be removed while the representation remains Markov. Hence, we say that a minimal state representation Φ^* is a sufficient statistic of the full set of observation variables F. In Example 1, representations like $\Phi(s_t) = \langle x_0, a_t, l_t \rangle$ or $\Phi(s_t) = \langle x_0, x_t, l_t \rangle$ are valid Markov state representations. However, only representations that exclusively retain the initial signal X_0 and the agent's current location L_t (i.e., $\Phi^*(s_t) = \langle x_0, l_t \rangle$) are considered minimal, as only these two variables are necessary to capture rewards and transitions.

Definition 6 (Superfluous variable)**.** Let $\{\bar{F}^*\}_{\cup \Phi^*}$ be the union of variables in all possible minimal state representations. A variable $F^i \in F$ is said to be superfluous, if $F^i \notin \{\bar{F}^*\}_{\cup \Phi^*}$.

In Example 1, any variable other than the signal and the current location, $F \setminus \{X_0, L_t\}$, is superfluous.

5.2 π-Markov state representations

Considering that the agent's policy will rarely visit all states, the notion of Markov state representation might be overly strict. We now introduce a relaxed version that guarantees the representation is Markov when following specific policy π.

Definition 7 (π-Markov state representation)**.** A state representation $\Phi^\pi(h_t)$ is said to be π-Markov if, for all $s_t, s_{t+1} \in \mathcal{S}^\pi$, $a_t \in \text{supp}(\pi(\cdot \mid s_t))$,

$$R(s_t, a_t) = R^\pi(\Phi^\pi(s_t), a_t) \quad \text{and} \quad \sum_{s'_{t+1} \in \{s_{t+1}\}^\Phi_\pi} T(s'_{t+1} \mid s_t, a_t) = \Pr^\pi(\Phi^\pi(s_{t+1}) \mid \Phi^\pi(s_t), a_t),$$

where $S^\pi \subseteq S$ denotes the set of states visited under π, $R^\pi(\Phi^\pi(s_t), a_t)$ is the reward $R(s'_t, a_t)$ at any $s'_t \in \{s_t\}^\Phi_\pi$, with $\{s\}^\Phi_\pi = \{s' \in S^\pi : \Phi^\pi(s') = \Phi^\pi(s)\}$, and $\Pr^\pi$ is probability under π.

Definition 8 (π-minimal state representation)**.** A state representation $\Phi^{\pi *} : \mathcal{S}^\pi \to \bar{\mathcal{S}}^{\pi *}$ with $\bar{\mathcal{S}}^{\pi *} = \times \bar{\mathcal{F}}^{\pi *}$ is said to be *π-minimal*, if all other state representations $\Phi : \mathcal{S}^\pi \to \bar{\mathcal{S}}^\pi$ with $\bar{\mathcal{S}}^\pi = \times \bar{\mathcal{F}}$ and $\bar{F} \subset \bar{F}^{\pi *}$, for some $s \in \mathcal{S}^\pi$, are not π-Markov.

The next result demonstrates that a π-Markov state representation Φ^π requires at most the same variables, and in some cases fewer, than a minimal state representation Φ^*, while still satisfying the Markov conditions for those states visited under π, $s \in S^\pi$.

Proposition 1. *Let $\mathbf{\Phi}^*$ be the set of all possible minimal state representations, where every $\Phi^* \in \mathbf{\Phi}^*$ is defined as $\Phi^* : \mathcal{S} \to \bar{\mathcal{S}}^*$ with $\bar{\mathcal{S}}^* = \times \bar{\mathcal{F}}^*$. For all π and all $\Phi^* \in \mathbf{\Phi}^*$, there exists a π-Markov state representation $\Phi^\pi : \mathcal{S}^\pi \to \bar{\mathcal{S}}^\pi$ with $\bar{\mathcal{S}}^\pi = \times \bar{\mathcal{F}}^\pi$ such that for all $s \in \mathcal{S}^\pi$, $\bar{F}^\pi \subseteq \bar{F}^*$. Moreover, there exist cases where $\bar{F}^\pi$ is a proper subset, $\bar{F}^\pi \subset \bar{F}^*$.*

It is clear that Φ^π will never require more variables than the corresponding minimal state representation Φ^* because, as per Definition 4, Φ^* captures all the essential information. The situation where $\bar{F}^\pi \subset \bar{F}^*$ arises with particular policies that exclusively visit a subset of states. In such cases, the agent may require fewer variables within that subset to accurately capture rewards and transitions. Take, for instance, a policy that makes the agent stay put. The π-minimal representation under such a policy is the empty set, $\Phi(s_t) = \emptyset$ for all $s_t \in \mathcal{S}^\pi$, as the agent consistently receives the same reward and does not move from the initial state.

6 Policy Confounding

Judging from the previous example, it might be tempting to assume that having $\bar{F}^\pi \subset \bar{F}^*$ is merely an incidental outcome of following a policy π that visits a subset of states, where some variables coincidentally happen to be unnecessary. Moreover, considering that $\bar{F}^*$ is constructed to capture the essential variables necessary for the task, one may further conclude that a policy π inducing representations such that $\bar{F}^\pi \subset \bar{F}^*$ can never be optimal. However, as demonstrated by the following example, it turns out that the states visited by a particular policy, especially if it is the optimal policy, tend to contain a lot of redundant information. This is particularly true in environments where future states are heavily influenced by past actions.

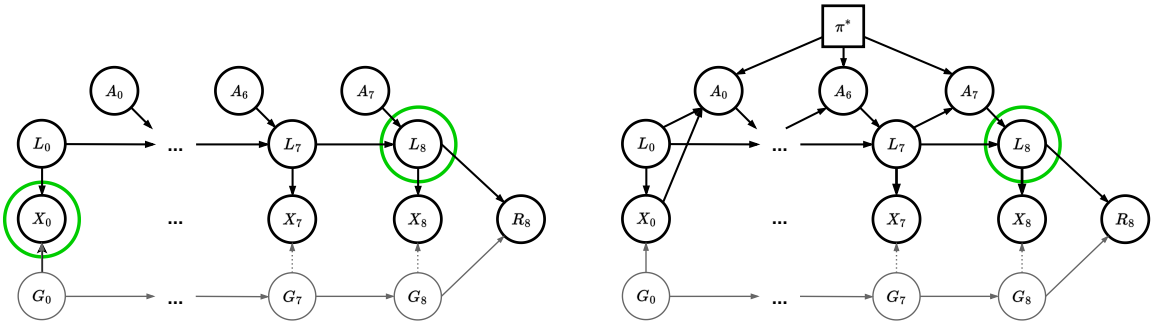

Figure 2: Two DBNs representing the dynamics of the Frozen T-Maze environment, when actions are sampled at random (left), and when they are determined by the optimal policy (right). The green circles highlight the π-mininal state representation in each of the two cases.

Let us revisit Example 1. Figure 2 shows two dynamic Bayesian networks (DBNs) describing the environment's dynamics: one with random action sampling (left) and the other with actions determined by the optimal policy (right). Both networks are unrolled from $t = 0$ to $t = 8$, with intermediate nodes omitted for simplicity. Suppose we aim to predict the reward R_8 given $s_8 = \langle l_0, x_0, a_0, \ldots, a_7, l_8, x_8 \rangle$. In the case of random action sampling (left DBN), to predict R_8, one needs X_0 and L_8. This is because $\langle A_0, \ldots, A_7 \rangle$ appear as exogenous and can take any possible value. Hence, the reward could be either -0.1 (per timestep penalty), -1 (wrong goal), or $+1$ (correct goal) depending on the actual values of X_0 and L_8. As established in Section 5.1, we know that $\Phi^*(s_t) = \langle x_0, l_t \rangle$ is a minimal state representation.

Conversely, when actions are determined by the optimal policy π^* (right DBN), knowing L_8 alone suffices to determine R_8. The reason is that, under π^*, the agent always takes the action 'move up' at the starting location when receiving the green signal or 'move down' when receiving the purple signal and then follows the shortest path toward each of the goals. As shown by the diagram, this makes the action A_0, and thus all future agent locations, dependent on the initial signal X_0. Hence, the agent's location L_t becomes a proxy for X_0, allowing the agent to ignore X_0 and still predict transitions and rewards. Consequently, from $t = 1$ onward, $\Phi^{\pi^*}(s_t) = l_t$ is a π-minimal state representation (Definition 8) as it constitutes a sufficient statistic of the state s_t under π^*. For the same reason, from $t = 1$, actions may also condition only on L_t.

The phenomenon highlighted by the previous example results from a spurious correlation induced by the optimal policy between the initial signal X_0 and the agent's future locations $\langle L_1, \ldots, L_8 \rangle$. Generally speaking, this occurs because policies act as confounders, opening backdoor paths between future observation variables F_{t+1} and the variables in the current state F_t (Pearl et al., 2016). This is illustrated by the DBN depicted in Figure 3, where the policy influences both the variables in F_t and the variables in F_{t+1}, potentially altering their correlations.

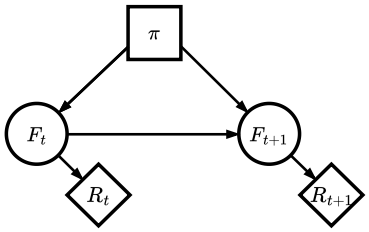

Figure 3: A DBN illustrating the phenomenon of policy confounding. The policy opens a backdoor path that can affect conditional relations between the variables in F_t and F_{t+1}.

Definition 9 (Policy Confounding)**.** A state representation $\Phi : \mathcal{S} \to \bar{\mathcal{S}}$ is said to be confounded by a policy π if, for some $s_t, s_{t+1} \in \mathcal{S}$, $a_t \in \mathcal{A}$,

$$R^\pi(\Phi(s_t), a_t) \neq R^\pi(\text{do}(\Phi(s_t)), a_t) \quad \text{or} \quad \Pr^\pi(\Phi(s_{t+1}) \mid \Phi(s_t), a_t) \neq \Pr^\pi(\Phi(s_{t+1}) \mid \text{do}(\Phi(s_t)), a_t).$$

The operator $\text{do}(\cdot)$ is known as the do-operator, and it is used to represent physical interventions in a system (Pearl et al., 2016). These interventions are meant to distinguish cause-effect relations from mere statistical associations. In our case, $\text{do}(\Phi(s_t))$ means setting the variables forming the state representation $\Phi(s_t)$ to a particular value and considering all possible states in the equivalence class, $s'_t \in \{s_t\}^\Phi$, (i.e., all states that share the same value for the observation variables that are 'selected'

by Φ; independently of whether these are visited by the policy being followed). For instance, in the above example, $R^{\pi^*}(L_8 = \text{'green goal'}) = +1$ when following π^* since we know that $X_0 = \text{'green'}$, while $R^{\pi^*}(\text{do}(L_8 = \text{'green goal'})) = \pm 1$ since X_0 can be either 'green' or 'purple'.

It is easy to show that the underlying reason why a π-Markov state representation may require fewer variables than the minimal state representation is indeed policy confounding.

Theorem 1. *Let $\Phi^* : \mathcal{S} \to \bar{\mathcal{S}}^*$ with $\bar{\mathcal{S}}^* = \times \bar{F}^*$ be a minimal state representation. If, for some π, there is a π-Markov state representation $\Phi^\pi : \mathcal{S}^\pi \to \bar{\mathcal{S}}^\pi$ with $\bar{\mathcal{S}}^\pi = \times \bar{F}^\pi$, such that $\bar{F}^\pi \subset \bar{F}^*$ for some $s \in \mathcal{S}$, then Φ^π is confounded by policy π.*

Finally, it is worth noting that even though, in Example 1, the variables included in the π-minimal state representation are a subset of the variables in the minimal state representation, $\bar{F}^{\pi*} \subset \bar{F}^*$, this is not always the case, as $\bar{F}^{\pi*}$ may contain superfluous variables (Definition 6). An example illustrating this situation is provided in Appendix B (Example 4).

Proposition 2. *Let $\{\bar{F}^*\}_{\cup \Phi^*}$ be the union of variables in all possible minimal state representations. There exist cases where, for some π, there is a π-minimal state representation $\Phi^{\pi*} : \mathcal{S}^\pi \to \bar{\mathcal{S}}^{\pi*}$ with $\bar{\mathcal{S}}^{\pi*} = \times \bar{F}^{\pi*}$ such that $\bar{F}^{\pi*} \setminus \{\bar{F}^*\}_{\cup \Phi^*} \neq \emptyset$.*

6.1 Why should we care about policy confounding?

Leveraging spurious correlations to develop simple habits can be advantageous when resources such as memory, computing power, or data are limited. Agents can exclude variables from the state representation if they are redundant under their policy. However, the challenge is that some of these variables may be crucial to ensure that the agent behaves correctly when the context changes. In the Frozen T-Maze example from Section 2, we observed how the agent could no longer find the correct goal when the ice pushed it away from the optimal trajectory. This is a specific case of a well-researched issue known as out-of-distribution (OOD) generalization (Quionero-Candela et al., 2009; Arjovsky, 2021). We refer to it as *out-of-trajectory* (OOT) generalization to highlight that the problem here is that the agent is unable to generalize to alternative trajectories within the same environment. This is in contrast to previous works (Kirk et al., 2023) that address generalization to environments that differ in some way from the training environment. Example 5 illustrates the distinction between OOT and OOD.

Ideally, the agent should aim to learn representations that enable it to predict future rewards and transitions even when experiencing slight variations in its trajectory. Based on Definition 4, we know that, in general, only a Markov state representation satisfies these requirements. However, computing such representations is typically intractable (Ferns et al., 2006), and thus most standard RL methods usually learn representations by maximizing an objective function that depends on the distribution of trajectories $P^b(\tau)$ visited under a behavior policy b (e.g., expected return, $\mathbb{E}_{\tau \sim P^b(\tau)}[G(\tau)]$; Sutton & Barto, 2018). The problem is that b may favor certain trajectories over others, which may lead to the exploitation of spurious correlations in the learned representation.

6.2 When should we worry about OOT generalization in practice?

Function approximation Function approximation has enabled traditional RL methods to scale to high-dimensional problems, where storing values in lookup tables is infeasible (François-Lavet et al., 2018). Using parametric functions (e.g., neural networks) to model policies and value functions, agents can learn abstractions by grouping together states if these yield the same transitions and rewards. As mentioned before, abstractions occur naturally when states are represented by a set of variables since the functions simply need to ignore some of these variables. However, this also implies that value functions and policies are exposed to spurious correlations. If a particular variable becomes irrelevant due to policy confounding, the function may learn to ignore it and remove it from its representation (Example 1). This is in contrast to tabular representations, where, every state takes a separate entry, and even though there exist algorithms that perform state abstractions in

tabular settings (Andre & Russell, 2002; Givan et al., 2003), these abstractions are normally formed offline before learning the policy, hence avoiding the risk of policy confounding.

Narrow trajectory distributions In practice, agents are less prone to policy confounding when the trajectory distribution $P^b(\tau)$ is broad (i.e., when b encompasses a wide set of trajectories) than when it is narrow. This is because the spurious correlations present in certain trajectories are less likely to have an effect on the learned representations. On-policy methods (e.g., SARSA, Actor-Critic; Sutton & Barto, 2018) are particularly troublesome for this reason since the same policy being updated must also be used to collect the samples. Yet, even when the trajectory distribution is narrow, there is no reason why the agent should pick up on spurious correlations while its policy is still being updated. Only when the agent commits to a particular policy should we start worrying about policy confounding. At this point, lots of the same trajectories are being used for training, and the agent may *'forget'* (French, 1999) that, even though certain variables may no longer be needed to represent the states, they were important under previous policies. This generally occurs at the end of training when the agent has converged to a particular policy. However, if policy confounding occurs earlier during training, it may prevent the agent from further improving its policy (Nikishin et al., 2022; please refer to Appendix C for more details).

6.3 What can we do to improve OOT generalization?

As mentioned in the introduction, we do not have a complete answer to the problem of policy confounding. Yet, here we offer a few off-the-shelf solutions that, while perhaps limited in scope, can help mitigate the problem in some situations. These solutions revolve around the idea of broadening the distribution of trajectories to dilute the spurious correlations introduced by certain policies.

Off-policy methods We already explained in Section 6.2 that on-policy methods are particularly prone to policy confounding since they are restricted to using samples coming from the same policy. A rather obvious solution is to instead use off-policy methods, which allow using data generated from previous policies. Because the samples belong to a mixture of policies it is less likely that the model will pick up the spurious correlations present on specific trajectories. However, as we shall see in the experiments, this alternative works only when replay buffers are large enough. This is because standard replay buffers are implemented as queues, and hence the first experiences coming in are the first being removed. This implies that a replay buffer that is too small will contain samples coming from few and very similar policies. Since there is a limit on how large replay buffers are allowed to be, future research could explore other, more sophisticated, ways of deciding what samples to store and which ones to remove (Schaul et al., 2016).

Exploration and domain randomization When allowed, exploration may mitigate the effects of policy confounding and prevent agents from overfitting their preferred trajectories. Exploration strategies have already been used for the purpose of generalization; to guarantee robustness to perturbations in the environment dynamics (Eysenbach & Levine, 2022), or to boost generalization to unseen environments (Jiang et al., 2022). The goal for us is to remove, to the extent possible, the spurious correlations introduced by the current policy. Unfortunately, though, exploration is not always without cost. Safety-critical applications require the agent to stay within certain boundaries (Altman, 1999; García & Fernández, 2015). When training on a simulator, an alternative to exploration is domain randomization (Tobin et al., 2017; Peng et al., 2018; Machado et al., 2018). The empirical results reported in the next section suggest that agents become less susceptible to policy confounding when adding enough stochasticity. Yet, there is a limit on how much noise can be added to the environment or the policy without altering the optimal policy (Sutton & Barto, 2018, Example 6.6: Cliff Walking).

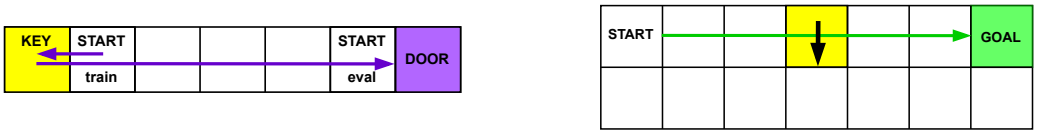

Figure 4: Illustrations of the Key2Door (left) and Diversion (right) environments.

7 Experiments

The experiments aim to (1) validate the occurrence of policy confounding as described by the theory, (2) identify the conditions under which agents are most susceptible to the effects of policy confounding and fail to generalize, and (3) assess the effectiveness of strategies proposed in the previous section in mitigating these effects. To achieve this, a series of simple grid-world environments has been devised as pedagogical examples to highlight the issue and clarify the theory. We would like to emphasize that our primary contribution lies in characterizing the phenomenon of policy confounding. The extent to which this phenomenon manifests in more realistic settings is beyond the scope of this paper. However, we believe that the failure of standard RL methods in these simplistic environments raises important concerns. Moreover, we refer the reader to Appendix C for a review of prior works reporting evidence of particular forms of policy confounding in high-dimensional environments.

7.1 Experimental setup

Agents are trained with an off-policy method, DQN (Mnih et al., 2015) and an on-policy method, PPO (Schulman et al., 2017). We represent policies and value functions as feedforward neural networks and use a stack of past observations as input in the environments that require memory. We report the mean return as a function of the number of training steps. Training is interleaved with periodic evaluations on the original environments and variants thereof used for validation. The results are averaged over 10 random seeds. Refer to Appendix F for more details about the setup.

7.2 Environments

We run our experiments on three grid-world environments: the **Frozen T-Maze** from Section 2, and the below described **Key2Door**, and **Diversion** environments.

Example 2. Key2Door. Here, the agent needs to collect a key placed at the beginning of the corridor in Figure 4 (left) and then open the door at the end. The current observation variables do not show whether the key has already been collected. The states are thus given by the history of past locations $s_t = \langle l_0, \dots, l_t \rangle$. This is because to solve the task in the minimum number of steps, the agent must remember that it already got the key when going to the door. Yet, since during training, the agent always starts the episode at the first cell from the left, when moving toward the door, the agent can forget about the key (i.e., ignore past locations) once it has reached the third cell. As in the Frozen T-Maze example, the agent can build the habit of using its own location to tell whether it has or has not got the key yet. This, can only occur when the agent consistently follows the optimal policy, depicted by the purple arrow. Otherwise, if the agent moves randomly through the corridor, it is impossible to tell whether the key has or has not been collected. In contrast, in the evaluation environment, the agent always starts at the second to last cell, this confuses the agent, which is used to already having the key by the time it reaches said cell. A DBN is provided in Appendix D.

Example 3. Diversion. Here, the agent must move from the start state to the goal state in Figure 4 (right). The observations are length-8 binary vectors. The first 7 elements indicate the column where the agent is located. The last element indicates the row. This environment aims to show that policy confounding can occur not only when the environment is partially observable, as was the case in the previous examples, but also in fully observable scenarios. After the agent learns the optimal trajectory depicted by the green arrow, it can disregard the last element in the observation vector. This is because, if the agent does not deviate, the bottom row is never visited. Rather than forgetting past information, the agent ignores the last element in the current observation vector for being

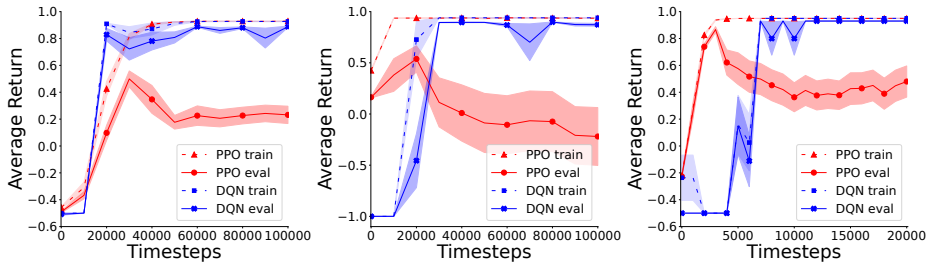

Figure 5: DQN vs. PPO in the train and evaluation variants of Frozen T-Maze (left), Key2Door (middle), and Diversion (right).

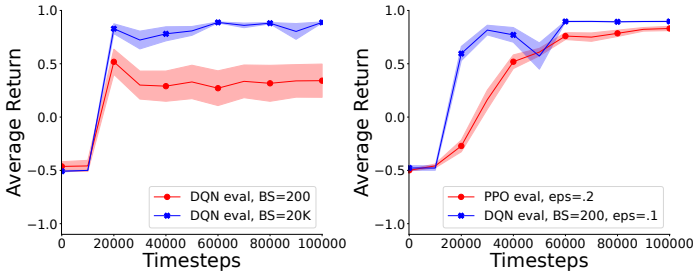

Figure 6: Frozen T-Maze. Left: DQN small vs. large buffer sizes. Right: PPO and DQN when adding stochasticity.

irrelevant when following the optimal trajectory. We train the agent in the original environment and evaluate it in a version with a yellow diversion sign in the middle of the maze that forces the agent to move to the bottom row. A DBN is provided in Appendix D.

7.3 Results

On-policy vs. off-policy The results in Figure 5 consistently reveal a common trend across all three environments. PPO struggles with generalization beyond the agent's preferred trajectories. After an initial phase where the average returns on the training and evaluation environments increase ('PPO train' and 'PPO eval'), the return on the evaluation environments ('PPO eval') starts decreasing when the agent commits to a particular trajectory, as a result of policy confounding. In contrast, since the training samples come from a mixture of policies, DQN performs optimally in both variants of the environment ('DQN train' and 'DQN eval') long after converging to the optimal policy. A visualization of the state representations learned with PPO, showing that the policy does ignore necessary variables, is provided in Appendix E.1.

Large vs. small replay buffers We mentioned in Section 6.3 that the effectiveness of off-policy methods against policy confounding depends on the size of the replay buffer. The results in Figure 6 (left) confirm this claim. The plot shows the performance of DQN in the Frozen T-Maze environment when the size of the replay buffer contains 100K experiences and when it only contains the last 10K experiences. We see that in the second case, the agents performance in the evaluation environment decreases (red curve left plot). This is because, after the initial exploration phase, the distribution of trajectories becomes too narrow, and the spurious correlations induced by the latest policies dominate the replay buffer. Similar results for the other two environments are provided in Appendix E.2.

Exploration and domain randomization The last experiment shows that if sufficient exploration is allowed, DQN may still generalize to different trajectories, even when using small replay buffers (blue curve right plot on Figure 6). In the original configuration, the exploration rate ϵ for DQN starts at $\epsilon = 1$ and decays to $\epsilon = 0.0$ after 20K steps. For this experiment, we set the final rate $\epsilon = 0.1$. In contrast, since exploration in PPO is controlled by the entropy bonus, which makes it hard to ensure fixed exploration rates, we add noise to the environment instead. The red curve

in Figure 6 (right) shows that when the agent's actions are overridden by a random action with 20% probability, the performance of PPO in the evaluation environment does not degrade after the agent has converged to the optimal policy. This suggests that the added noise prevents spurious correlations from dominating training batches. However, it may also happen that random noise is insufficient to remove the spurious correlations, as occurs in the Key2Door environment (Figure 13; Appendix E.2). Similar results for Diversion are provided in Appendix E.2.

8 Conclusion

This paper described the phenomenon of policy confounding. We demonstrated both theoretically and empirically how, as a result of following certain trajectories, agents may pick up on spurious correlations and develop habits that are not robust to trajectory deviations. We also identified the circumstances under which policy confounding is most likely to occur in practice and suggested a few ad hoc solutions that may mitigate its effects. We view this paper as a stepping stone to exploring more sophisticated solutions. An interesting avenue for future research is the integration of tools from the field of causal inference (Hernán & Robins, 2010; Peters et al., 2017) to assist the agent in forming state representations grounded in causal relationships rather than mere statistical associations (Lu et al., 2018; Zhang et al., 2020; Sontakke et al., 2021; Saengkyongam et al., 2023).

Acknowledgements

This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 758824 —INFLUENCE).

References

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC press, 1999.

David Andre and Stuart J. Russell. State abstraction for programmable reinforcement learning agents. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 119–125, 2002.

Martin Arjovsky. Out of distribution generalization in machine learning. *arXiv preprint arXiv:2103.02667*, 2021.

Bram Bakker. Reinforcement learning with long short-term memory. *Advances in neural information processing systems*, 14, 2001.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Thomas Dean and Robert Givan. Model minimization in Markov decision processes. In *Proc. of the National Conference on Artificial Intelligence*, pp. 106–111, 1997.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. In *International Conference on Learning Representations*, 2022.

Norm Ferns, Pablo Samuel Castro, Doina Precup, and Prakash Panangaden. Methods for computing state similarity in markov decision processes. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, pp. 174–181, 2006.

Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, 2018.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 14(1–2):163–223, 2003.

Miguel A Hernán and James M Robins. *Causal inference*. CRC Boca Raton, FL, 2010.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 1480–1490. PMLR, 2017.

Yiding Jiang, J Zico Kolter, and Roberta Raileanu. Uncertainty-driven exploration for generalization in reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

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

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Li-Cheng Lan, Huan Zhang, and Cho-Jui Hsieh. Can agents run relay race with strangers? generalization of RL to out-of-distribution trajectories. In *The Eleventh International Conference on Learning Representations*, 2023.

Lauro Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 12004–12019. PMLR, 2022.

Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. *Reinforcement Learning: State-of-the-Art*, pp. 143–173, 2012.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*, 2006.

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

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3932–3939. IEEE, 2017.

Andrew K. McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, University of Rochester, 1995.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Matthias Muller-Brockhausen, Mike Preuss, and Aske Plaat. Procedural content generation: Better benchmarks for transfer reinforcement learning. In *2021 IEEE Conference on games (CoG)*, pp. 01–08. IEEE, 2021.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in RL? decoupling memory from credit assignment. *Advances in Neural Information Processing Systems*, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 16828–16847. PMLR, 2022.

Daniel Jarne Ornia, Licio Romao, Lewis Hammond, Manuel Mazo Jr, and Alessandro Abate. Observational robustness and invariances in reinforcement learning via lexicographic objectives. *arXiv preprint arXiv:2209.15320*, 2022.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. *Advances in Neural Information Processing Systems*, 2021.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810. IEEE, 2018.

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

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Sorawit Saengkyongam, Nikolaj Thams, Jonas Peters, and Niklas Pfister. Invariant policy learning: A causal perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. In *International Conference on Learning Representations*, 2020.

Sumedh A Sontakke, Arash Mehrjou, Laurent Itti, and Bernhard Schölkopf. Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning. In *International conference on machine learning*, pp. 9848–9858. PMLR, 2021.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

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

Gavin Taylor and Ronald Parr. Kernelized value function approximation for reinforcement learning. In *Proc. of the Twenty-Sixth International Conference on Machine learning*, pp. 128, 2009.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Amy Zhang, Nicolas Ballas, and Joelle Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 2018a.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pp. 11214–11224. PMLR, 2020.

Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018b.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

A Proofs

Lemma 1. *Let $\mathbf{\Phi}^{\pi_1 *}$ be the set of all possible π-minimal state representations under π_1, where every $\Phi^{\pi_1 *} \in \mathbf{\Phi}^{\pi_1 *}$ is defined as $\Phi^{\pi_1 *} : \mathcal{S}^{\pi_1} \to \bar{\mathcal{S}}^{\pi_1 *}$ and $\bar{\mathcal{S}}^{\pi_1 *} = \times \bar{\mathcal{F}}^{\pi_1 *}$, and let π_2 be a second policy such that for all $s_t \in \mathcal{S}^{\pi_1} \cap \mathcal{S}^{\pi_2}$,*

$$\operatorname{supp}\left(\pi_2(\cdot \mid s_t)\right) \subseteq \operatorname{supp}\left(\pi_1(\cdot \mid s_t)\right).$$

For all $\Phi^{\pi_1 *} \in \mathbf{\Phi}^{\pi_1 *}$, there exists a π-Markov state representation under policy π_2, $\Phi^{\pi_2} : \mathcal{S}^{\pi_2} \to \bar{\mathcal{S}}^{\pi_2}$ with $\bar{\mathcal{S}}^{\pi_2} = \times \bar{\mathcal{F}}^{\pi_2}$, such that $\bar{F}^{\pi_2} \subseteq \bar{F}^{\pi_1 *}$ for all $s_t \in \mathcal{S}^{\pi_1} \cap \mathcal{S}^{\pi_2}$. Moreover, there exist cases where $\bar{F}_t^{\pi_2} \neq \bar{F}_t^{\pi_1 *}$.

Proof. First, it is easy to show that

$$\forall s_t \in \mathcal{S}, \operatorname{supp}\left(\pi_2(\cdot \mid s_t)\right) \subseteq \operatorname{supp}\left(\pi_1(\cdot \mid s_t)\right) \iff \mathcal{S}^{\pi_2} \subseteq \mathcal{S}^{\pi_1},$$

and

$$\forall s_t \in \mathcal{S}, \operatorname{supp}\left(\pi_2(\cdot \mid s_t)\right) = \operatorname{supp}\left(\pi_1(\cdot \mid s_t)\right) \iff \mathcal{S}^{\pi_2} = \mathcal{S}^{\pi_1}.$$

In particular, $\mathcal{S}^{\pi_2} \subset \mathcal{S}^{\pi_1}$ if there is at least one state $s'_t \in \mathcal{S}^{\pi_1} \cap \mathcal{S}^{\pi_2}$ such that

$$\operatorname{supp}\left(\pi_2(\cdot \mid s'_t)\right) \subset \operatorname{supp}\left(\pi_1(\cdot \mid s'_t)\right)$$

while

$$\operatorname{supp}\left(\pi_2(\cdot \mid s_t)\right) = \operatorname{supp}\left(\pi_1(\cdot \mid s_t)\right)$$

for all other $s_t \in \mathcal{S}^{\pi_1} \cap \mathcal{S}^{\pi_2}$.

In such cases, we know that there is at least one action a' for which $\pi_2(a'_t \mid s'_t) = 0$ but $\pi_1(a'_t \mid s'_t) \neq 0$. Hence, if there was a state (or group of states) that could only be reached by taking action a'_t at s'_t, π_2 would never visit it and thus $\mathcal{S}^{\pi_2} \subset \mathcal{S}^{\pi_1}$.

Further, if $\mathcal{S}^{\pi_2} \subset \mathcal{S}^{\pi_1}$, we know that, for every $\Phi^{\pi_1 *} \in \mathbf{\Phi}^{\pi_1 *}$, there must be a $\Phi^{\pi_2 *}$ that requires, at most, the same number of variables, $\bar{F}_t^{\pi_2} \subseteq \bar{F}_t^{\pi_1 *}$ and, in some cases, fewer, $\bar{F}_t^{\pi_1 *} \neq \bar{F}_t^{\pi_2 *}$ (e.g., Frozen T-Maze example).

□

Proposition 1. *Let $\mathbf{\Phi}^*$ be the set of all possible minimal state representations, where every $\Phi^* \in \mathbf{\Phi}^*$ is defined as $\Phi^* : \mathcal{S} \to \bar{\mathcal{S}}^*$ with $\bar{\mathcal{S}}^* = \times \bar{\mathcal{F}}^*$. For all π and all $\Phi^* \in \mathbf{\Phi}^*$, there exists a π-Markov state representation $\Phi^\pi : \mathcal{S}^\pi \to \bar{\mathcal{S}}^\pi$ with $\bar{\mathcal{S}}^\pi = \times \bar{\mathcal{F}}^\pi$ such that for all $s \in \mathcal{S}^\pi$, $\bar{F}^\pi \subseteq \bar{F}^*$. Moreover, there exist cases where $\bar{F}^\pi$ is a proper subset, $\bar{F}^\pi \subset \bar{F}^*$.*

Proof. The proof follows from Lemma 1. We know that, in general, $\mathcal{S}^\pi \subseteq \mathcal{S}$, and if $\pi(a'_t|s'_t) = 0$ for at least one pair $a'_t \in \mathcal{A}, s'_t \in \mathcal{S}$ for which there is a state (or group of states) that can only be reached by taking action a'_t at s'_t, then $\mathcal{S}^\pi \subset \mathcal{S}$. Hence, for every Φ^* there is a Φ^π such that $\bar{F}^\pi \subseteq \bar{F}^*$, and in some cases, we may have $\bar{F}^\pi \subset \bar{F}^*$ (e.g., Frozen T-Maze example).

□

Theorem 1. *Let $\Phi^* : \mathcal{S} \to \bar{\mathcal{S}}^*$ with $\bar{\mathcal{S}}^* = \times \bar{\mathcal{F}}^*$ be a minimal state representation. If, for some π, there is a π-Markov state representation $\Phi^\pi : \mathcal{S}^\pi \to \bar{\mathcal{S}}^\pi$ with $\bar{\mathcal{S}}^\pi = \times \bar{\mathcal{F}}^\pi$, such that $\bar{F}^\pi \subset \bar{F}^*$ for some $s \in \mathcal{S}$, then Φ^π is confounded by policy π.*

Proof. Proof by contradiction. Let us assume that $\bar{F}^\pi \subset \bar{F}^*$, and yet there is no policy confounding. I.e., for all $s_t, s_{t+1} \in \mathcal{S}$, $a_t \in \mathcal{A}$,

$$R^\pi(\Phi^\pi(s_t), a_t) = R^\pi(\text{do}(\Phi^\pi(s_t)), a_t)$$

and

$$\Pr^{\pi}(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s_t), a_t) = \Pr^{\pi}(\Phi^{\pi}(s_{t+1}) \mid \text{do}(\Phi^{\pi}(s_t)), a_t)$$

First, note that the do-operator implies that the equality must hold for *all* s'_t in the equivalence of s_t class under Φ^{π}, $s'_t \in \{s_t\}^{\Phi^{\pi}} = \{s'_t \in \mathcal{S} : \Phi(s'_t) = \Phi(s_t)\}$, i.e., not just those s'_t that are visited under π,

$$R^{\pi}(\Phi^{\pi}(s_t), a_t) = R^{\pi}(\text{do}(\Phi^{\pi}(s_t)), a_t) = R(s'_t, a_t) \quad \text{for all} \quad s'_t \in \{s_t\}^{\Phi}$$

which is precisely the first condition in Definition 4,

$$R(s_t, a_t) = R(\Phi^{\pi}(s_t), a_t) \tag{1}$$

for all $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$.

Analogously, we have that,

$$\begin{aligned}\Pr^{\pi}(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s_t), a_t) &= \Pr^{\pi}(\Phi^{\pi}(s_{t+1}) \mid \text{do}(\Phi^{\pi}(s_t)), a_t) \\ &= \Pr(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s_t), a_t)\end{aligned}$$

where the second equality reflects that the above must hold independently of π. Hence, we have that for all $s_t, s_{t+1} \in \mathcal{S}$ and $s'_t \in \{s_t\}^{\Phi}$,

$$\Pr(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s_t), a_t) = \Pr(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s'_t), a_t),$$

which means that, for all $s_t, s_{t+1} \in \mathcal{S}$ and $s_t \in \mathcal{A}$,

$$\begin{aligned}\Pr(\Phi^{\pi}(s_{t+1}) \mid \Phi^{\pi}(s_t), a_t) &= \Pr(\Phi^{\pi}(s_{t+1}) \mid s_t, a_t) \\ &= \sum_{s'_{t+1} \in \{s_{t+1}\}^{\Phi^{\pi}}} T(s'_{t+1} \mid s_t, a_t),\end{aligned} \tag{2}$$

which is the second condition in Definition 4.

Equations (1) and (2) reveal that if the assumption is true (i.e., Φ^{π} is not confounded by the policy), then Φ^{π} is not just π-Markov but actually strictly Markov (Definition 4). However, we know that $\Phi^*(s_t)$ is the minimal state representation, which contradicts the above statement, since, according to Definition 5, there is no proper subset of $\bar{F}^*$, for all $s_t \in \mathcal{S}$, such that the representation remains Markov. Hence, $\bar{F}^{\pi} \subset \bar{F}^*$ implies policy confounding. □

Proposition 2. *Let $\{\bar{F}^*\}_{\cup\Phi^*}$ be the union of variables in all possible minimal state representations. There exist cases where, for some π, there is a π-minimal state representation $\Phi^{\pi*} : \mathcal{S}^{\pi} \to \bar{\mathcal{S}}^{\pi*}$ with $\bar{\mathcal{S}}^{\pi*} = \times\bar{F}^{\pi*}$ such that $\bar{F}^{\pi*} \setminus \{\bar{F}^*\}_{\cup\Phi^*} \neq \emptyset$.*

Proof (sketch). Consider a deterministic MDP with a deterministic policy. Imagine there exists a variable X that is perfectly correlated with the episode's timestep t, but that is generally irrelevant to the task. The variable X would constitute in itself a valid π-Markov state representation since it can be used to determine transitions and rewards so long as a deterministic policy is followed. At the same time, X would not enter the minimal Markov state representation because it is useless under stochastic policies. Example 4 below illustrates this situation. □

B Example: Watch the Time

Example 4. (Watch the Time) This example is inspired by the empirical results of Song et al. (2020). Figure 7 shows a grid world environment. The agent must go from the start cell to the goal cell. The agent must avoid the pink cells; stepping on those yields a -0.1 penalty. There is a $+1$ reward for reaching the goal. The agent can observe its own location within the maze L and the current timestep t. The two diagrams in Figure 8 are DBNs describing the environment dynamics.

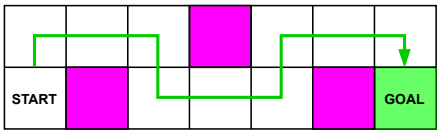

Figure 7: An illustration of the watch-the-time environment.

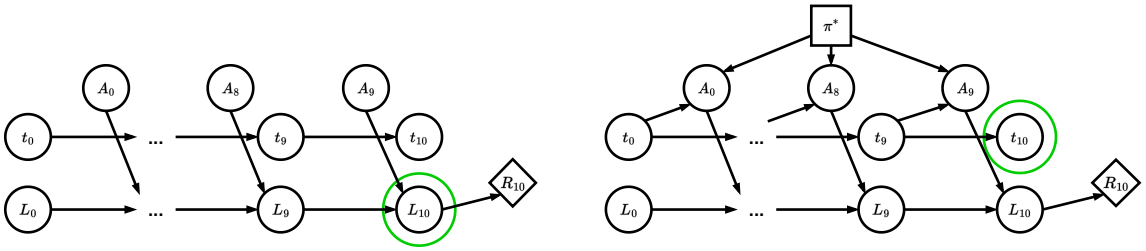

Figure 8: Two DBNs representing the dynamics of the watch-the-time environment, when actions are sampled at random (left), and when they are determined by the optimal policy (right).

When actions are considered exogenous random variables (left diagram), the only way to estimate the reward at $t = 10$ is by looking at the agent's location L_{10}. In contrast, when actions are determined by the policy (right diagram), t becomes a proxy for the agent's location. This is because the start location and the sequence of actions are fixed. This implies that t is a perfectly valid π-Markov state representation under π^*. Moreover, as shown by the DBN on the right, the optimal policy may simply rely on t to determine the optimal action.

C Further Related Work

Early evidence of policy confounding Although to the best of our knowledge, we are the first to bring forward and describe mathematically the idea of policy confounding, a few prior works have reported evidence of particular forms of policy confounding. In their review of the Arcade Learning Environment (ALE; Bellemare et al., 2013), Machado et al. (2018) explain that because the games are fully deterministic (i.e., initial states are fixed and transitions are deterministic), open-loop policies that memorize good action sequences can achieve high scores in ALE. Clearly, this can only occur if the policies themselves are also deterministic. In such cases, policies, acting as confounders, induce a spurious correlation between the past action sequences and the environment states. Similarly, Song et al. (2020) show, by means of saliency maps, how agents may learn to use irrelevant features of the environment that happen to be correlated with the agent's progress, such as background clouds or the game timer, as clues for outputting optimal actions. In this case, the policy is again a confounder for all these, a priori irrelevant, features. Zhang et al. (2018b) provide empirical results showing how large neural networks may overfit their training environments and, even when trained on a collection of procedurally generated environments, memorize the optimal action for each observation. Zhang et al. (2018a) show how, when trained on a small subset of trajectories, agents fail to generalize to a set of test trajectories generated by the same simulator. Ostrovski et al. (2021) empirically show that agents passively trained on observational data generated by other agents tend to perform poorly due to extrapolation errors caused by some of the state-action pairs being underrepresented in the data. Lan et al. (2023) report evidence of well-trained agents failing to perform well on Mujoco environments when starting from trajectories (states) that are out of the distribution induced by the agent's policy. We conceive this as a simple form of policy confounding. Since the Mujoco environments are also deterministic, agents following a fixed policy can memorize the best actions to take for each state instantiation, potentially relying on superfluous features. Hence, they can overfit to unnatural postures that would not occur under different policies. Finally, Nikishin et al. (2022) describe a phenomenon named 'primacy bias', which prevents agents trained on poor trajectories

from further improving their policies. The authors show that this issue is particularly relevant when training relies heavily on early data coming from a fixed random policy. We hypothesize that one of the causes for this is also policy confounding. The random policy may induce spurious correlations that lead to the formation of rigid state (state) representations that are hard to recover from.

Generalization Generalization is a hot topic in machine learning. The promise of a model performing well in contexts other than those encountered during training is undoubtedly appealing. In the realm of reinforcement learning, the majority of research focuses on generalization to environments that, despite sharing a similar structure, differ somewhat from the training environment (Kirk et al., 2023). These differences range from small variations in the transition dynamics (e.g., sim-to-real transfer; Higgins et al., 2017; Tobin et al., 2017; Peng et al., 2018; Zhao et al., 2020), changes in the observations (i.e., modifying irrelevant information, such as noise: Mandlekar et al., 2017; Ornia et al., 2022, or background variables: Zhang et al., 2020; Stone et al., 2021), to alterations in the reward function, resulting in different goals or tasks (Taylor & Stone, 2009; Lazaric, 2012; Muller-Brockhausen et al., 2021). Instead, we address the problem of OOT generalization, where the objective is to generalize to different trajectories within the same environment.

Example 5. To illustrate the difference between OOD generalization and OOT generalization, let us consider a robot trained via RL to go from our office to the coffee machine, get coffee, and come back, as well as from our office to the printer, make copies, and come back. There are two possible routes to the coffee machine: either through the printer room or through a corridor directly leading to the coffee machine. The path through the printer room is longer, so the robot typically avoids it when coffee is ordered. However, one day, when we order coffee and the corridor is blocked, the robot attempts to go through the printer room and returns with a copy of a new paper titled 'Bad Habits' instead of the coffee. This serves as an example of out-of-trajectory generalization. Since the robot is accustomed to obtaining copies in the copy room, it disregards the coffee order. An example of the more general problem of OOD generalization could involve instructing the robot to navigate the office when the floor is wet or to fetch something different, like a glass of water. The crucial distinction is that, in these last two examples, the states the robot visits or the rewards it receives differ. The robot has not been trained on a wet floor, and it has never retrieved a glass of water before. However, in the first example, we would expect the robot to recognize that being in the copy room does not necessarily imply getting copies. To be fair, the blocked corridor represents a change in the environment; nevertheless, this change is intended to prompt the agent to choose an alternative path. It is worth noting that this alternative path was also feasible in the original environment.

State abstraction State abstraction is concerned with removing from the representation all that state information that is irrelevant to the task. In contrast, we are worried about learning representations containing too little information, which can lead to state aliasing. Nonetheless, as argued by McCallum (1995), state abstraction and state aliasing are two sides of the same coin. That is why we borrowed the mathematical frameworks of state abstraction to describe the phenomenon of policy confounding. Li et al. (2006) provide a taxonomy of the types of state abstraction and how they relate to one another. Givan et al. (2003) introduce the concept of bisimulation, which is equivalent to our definition of Markov state representation (Definition 4). Ferns et al. (2006) propose a method for measuring the similarity between two states. Castro (2020) notes that this metric is prohibitively expensive and suggests using a relaxed version that computes state similarity relative to a given policy. This is similar to our notion of π-Markov state representation (Definition 7). While the end goal of this metric is to group together states that are similar under a given policy, here we argue that this may lead to poor OOT generalization.

D Dynamic Bayesian Networks

Figures 9 and 10 show the DBNs for the Key2Door and Diversion environments, respectively.

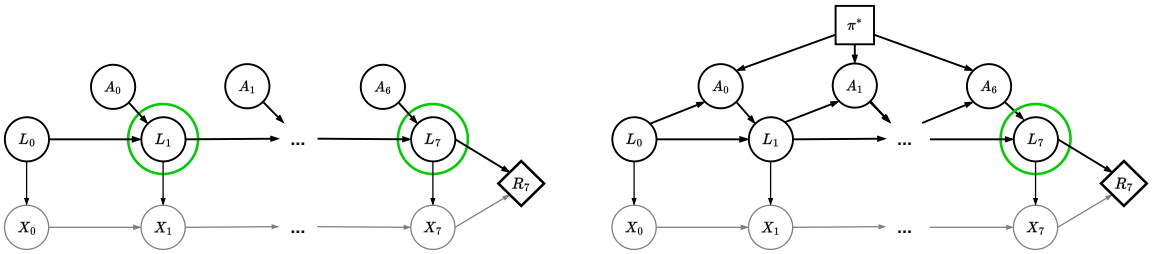

Figure 9: Two DBNs representing the dynamics of the Key2Door environment, when actions are sampled at random (left), and when they are determined by the optimal policy (right). The nodes labeled as L represent the agent's location, while the nodes labeled as X represent whether or not the key has been collected. The agent can only see L. Hence, when actions that are sampled are random (left), the agent must remember its past locations to determine the reward R_7. Note that only L_1 and L_7 are highlighted in the left DBN. However, other variables in $\langle L_2, \ldots, L_6 \rangle$ might be needed, depending on when the key is collected. In contrast, when following the optimal policy, only L_7 is needed. In this second case, knowing the location is sufficient to determine whether the key has been collected.

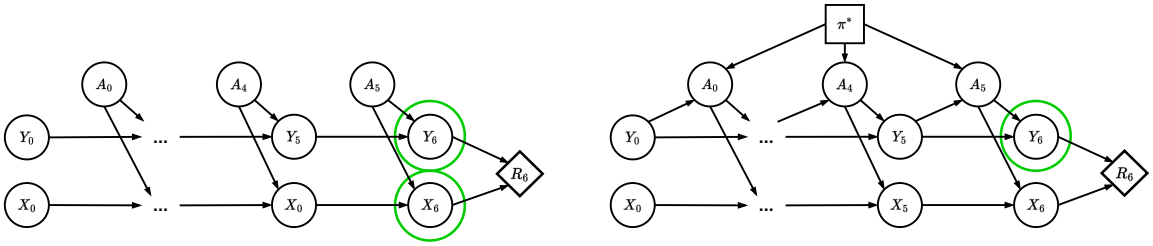

Figure 10: Two DBNs representing the dynamics of the Diversion environment, when actions are sampled at random (left), and when they are determined by the optimal policy (right). The nodes labeled as X indicate the row where the agent is located; the nodes labeled as Y indicate the column. We see that when actions are sampled at random, both X_6 and Y_6 are necessary to determine R_6. However, when actions are determined by the optimal policy, Y_6 is sufficient, as the agent always stays at the top row.

E Experimental Results

E.1 Learned state representations

The results reported in Section 7 show that the OOT generalization problem exists. However, some may still wonder if the underlying reason is truly policy confounding. To confirm this, we compare the outputs of the policy at every state in the Frozen T-Maze when being fed the same states (observation stack) but two different signals. That is, we permute the variable containing the signal (X in the diagram of Figure 2) and leave the rest of the variables in the observation stack unchanged. We then feed the two versions to the policy network and measure the KL divergence between the two output probabilities. This metric is a proxy for how much the agent attends to the signal in every state. The heatmaps in Figure 11 show the KL divergences at various points during training (0, 10K, 30K, and 100K timesteps) when the true signal is 'green' and we replace it with 'purple'. We omit the two goal states since no actions are taken there. We see that initially (top left heatmap), the signal has very little influence on the policy (note the scale of the colormap is 10^{-6}), after 10K steps, the agent learns that the signal is very important when at the top right state (top right heatmap). After this, we start seeing how the influence of the signal at the top right state becomes less strong (bottom left heatmap) until it eventually disappears (bottom right heatmap). In contrast, the influence of the signal at the initial state becomes more and more important, indicating that after taking the

first action, the agent ignores the signal and only attends to its own location. The results for the alternative case, 'purple' signal being replaced by 'green' signal, are shown in Figure 12.

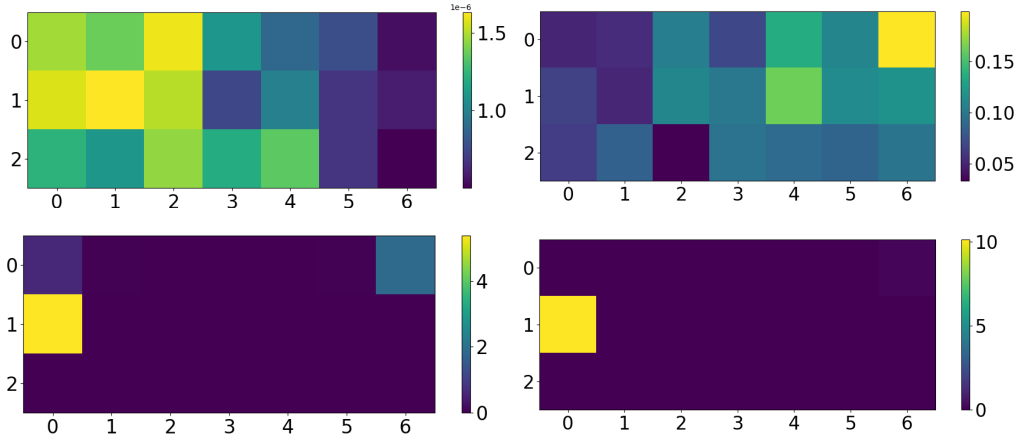

Figure 11: A visualization of the learned state representations. The heatmaps show the KL divergence between the action probabilities when feeding the policy network a stack of the past 10 observations and when feeding the same stack but with the value of the signal being switched from green to purple, after 0 (top left), 10K (top right), 30K (bottom left), and 100K (bottom right) timesteps of training.

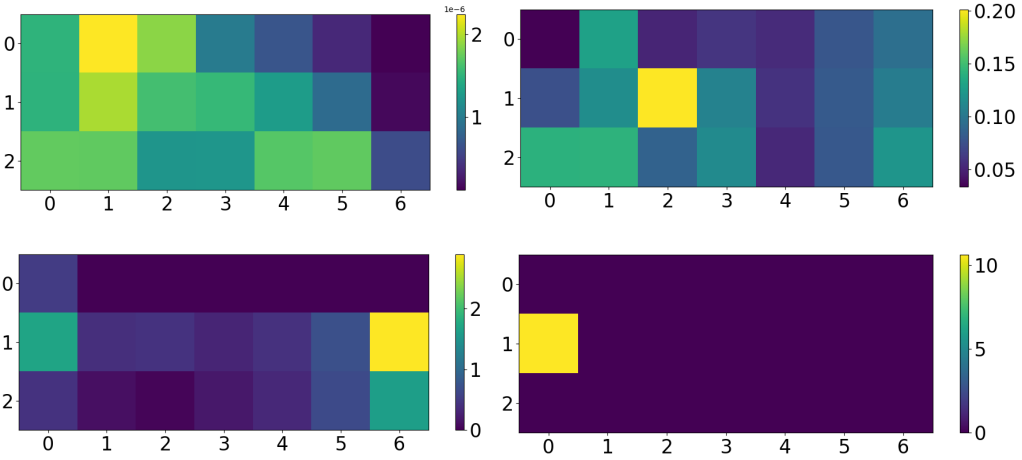

Figure 12: A visualization of the learned state representations. The heatmaps show the KL divergence between the action probabilities when feeding the policy network a stack of the past 10 observations and when feeding the same stack but with the value of the signal being switched from purple to green, after 0 (top left), 10K (top right), 30K (bottom left), and 100K (bottom right) timesteps of training.

E.2 Buffer size and exploration/domain randomization

Figures 13 and 14 report the results of the experiments described in Section 7 (paragraphs 2 and 3) for Key2Door and Diversion. We see how the buffer size also affects the performance of DQN in the two environments (left plots). We also see that exploration/domain randomization does improve OOT generalization in Diversion but not in Key2Door.

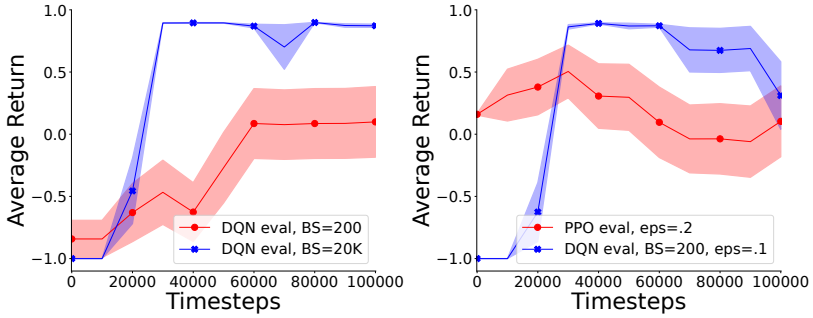

Figure 13: Key2Door. Left: DQN small vs. large buffer sizes. Right: PPO and DQN when adding stochasticity.

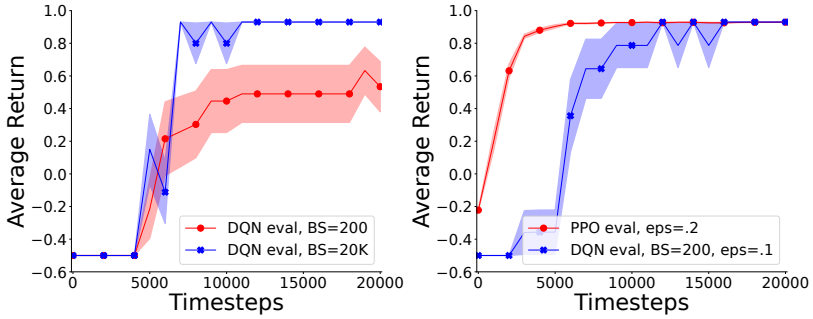

Figure 14: Diversion. Left: DQN small vs. large buffer sizes. Right: PPO and DQN when adding stochasticity.

F Further Experimental Details

We ran our experiments on an Intel i7-8650U CPU with 8 cores. Agents were trained with Stable Baselines3 (Raffin et al., 2021). Most hyperparameters were set to their default values except for the ones reported in Tables 1 (PPO) and 2 (DQN), which worked better than the default values for these particular environments.

Table 1: PPO hyperparameters.

Rollout steps	128
Batch size	32
Learning rate	2.5e-4
Number epoch	3
Entropy coefficient	1.0e-2
Clip range	0.1
Value coefficient	1
Number Neurons 1st layer	128
Number Neurons 2nd layer	128

Table 2: DQN hyperparameters.

Buffer size	1.0e5
Learning starts	1.0e3
Learning rate	2.5e-4
Batch size	256
Initial exploration bonus	1.0
Final exploration bonus	0.0
Exploration fraction	0.2
Training frequency	5
Number Neurons 1st layer	128
Number Neurons 2nd layer	128