Causal Discovery for Efficient Offline RL with Factored Action Spaces

Cecilia Ehrlichman CEHR@UMICH.EDU
Ann Arbor MI

Shengpu Tang SHENGPU.TANG@EMORY.EDU
Atlanta GA

Michael Dykstra MIDY@MED.UMICH.EDU
Maggie Makar MMAKAR@UMICH.EDU
Ann Arbor MI

Editors: Bijan Mazaheri and Niels Richard Hansen

Abstract

Offline policy optimization is often sample-inefficient, especially when the action space is large, a problem that commonly arises in healthcare applications and multi-agent tasks. Many domains, however, admit a combinatorial action space, where sub-actions affect future states and rewards independently of one another. Past work either makes *a priori* assumptions about sub-action independence leading to efficient but potentially biased policy optimization, or fails to leverage potential independence, sacrificing sample efficiency. In contrast, we propose a two-step framework that leverages causal discovery for efficient policy optimization without introducing bias. Our approach (i) discovers the causal structure underlying the environment's dynamics from observational data, and (ii) exploits this structure to restrict the admissible policy class to a simpler, unbiased class. We provide theoretical guarantees characterizing settings under which our approach leads to efficient unbiased policy learning. Empirically, we demonstrate that our approach leads to more efficient policy optimization in settings with limited observational data, across both single-agent healthcare tasks and multi-agent settings. Our code is available at https://github.com/cehr123/DiFaRL.

Keywords: Reinforcement Learning, Causal Discovery, Sample Efficiency

1. Introduction

Offline reinforcement learning (RL) has achieved significant success in environments with small action spaces or large datasets (Levine et al., 2020), (Antos et al., 2007), (Fujimoto et al., 2019), (Kostrikov et al., 2021). However, many real-world applications, such as healthcare (Hargrave et al., 2024) and education (Riedmann et al., 2025), exhibit high-dimensional action spaces and have limited observational data. In such settings, learning an optimal policy is challenging due to the combinatorial nature of the number of possible actions, which induces a prohibitively large policy class and results in statistical inefficiency.

To mitigate this, prior work imposes strong structural assumptions to constrain the policy space. For example, Tang et al. (2022) assume that different actions affects the transition function and the reward function independently of other action. While this independence assumption can simplify learning and improve sample efficiency, it is often unrealistic in practice leading to biased learning. In many domains, actions exhibit complex interactions that jointly influence outcomes. For instance, when managing blood pressure, the effect of phenylephrine may depend on the concurrent administration of steroid (Richards et al., 2023), an interaction that would be missed by treating the actions independently.

In this work, we take a different approach: we propose to learn if and where such independence holds by recovering the underlying causal structure from observational data. Our key insight is that many environments admit a sparse and structured causal graph over actions and state variables. By first learning this structure from observational data, we can steer policy learning to be consistent with the learned causal interaction patterns. This results in improved sample efficiency and better generalization, without relying on overly restrictive assumptions. Our proposed framework connects ideas from causal discovery and offline RL to address a key statistical bottleneck in policy learning with limited data.

We present our main analysis in the single-agent offline RL setting. However, our framework is directly applicable to offline multi-agent RL (MARL) scenarios, where actions corresponds to individual agents, and the interaction terms capture collaboration between agents. We demonstrate this extension in our experiments through an illustrative MARL example.

Our contributions are: **(1)** We propose a two-step approach that first identifies key causal structures from observational data and then leverages these structures to guide offline policy optimization; **(2)** We provide theoretical guarantees for the identifiability of the relevant causal structure and prove that our approach improves the finite-sample efficiency of the policy learning task; **(3)** We validate our framework empirically on three domains: two single-agent healthcare tasks and one multi-agent environment.

2. Related Work

Several prior works address the problem of RL environments with large combinatorial action spaces. Tang et al. (2022) proposes a linearly decomposed Q-function in the setting where the action space factors. Their solution requires assuming that the transition function and policy decompose multiplicatively and the reward decomposes additively over individual sub-actions, effectively treating sub-actions as independent. Their proposed approach leads to bias in settings where the action space does not satisfy these strict assumptions. Landers et al. (2025a) study offline RL with discrete combinatorial actions and suggest BraVE, a regularized tree-structured traversal approach. Their approach reduces the computational complexity of the task by evaluating only a subset of the actions selected by a greedy tree traversal procedure, but unlike us, they do not study factorization of the general (discrete/continuous) actions. Additionally, their approach does not leverage sub-action independence, which could harm sample efficiency. Landers et al. (2025b) propose an attention-based architecture that captures dependencies among sub-actions in an online, policy-based RL setting, and does not leverage any independencies or causal relationships between variables. Our attention-only baseline can be viewed as an extension of this work to the offline, value-based setting.

Similar to our approach, other work uses causal discovery to address the challenges inherent to RL problems with large combinatorial spaces. Zhu et al. (2022) leverage conditional independence tests to uncover *all* causal relationships between sub-states and sub-actions and employ the learned structure to create a model of the transition and reward function based on the learned causal relationships. This approach is similar to ours as we also utilize conditional independence testing. However, their method has three key limitations that our approach overcomes. First, it requires learning the entire causal structure underlying the environment, whereas we only recover the subset of causal relationships relevant to the action factorization, making our method substantially more lightweight. Second, their approach is model-based, inheriting issues such as error compounding Jiang (2024) and model bias, as we show in the experiment section. Third, unlike ours, their ap-

proach relies on meaningful factorization in the state space. Another model-based approach, Lei et al. (2024) propose learning a sparse world model via regularized attention weights to encourage sparsity. Like FOCUS, their approach learns the full transition function and requires disentangled state representations, whereas our method does not. Huang et al. (2022) learn minimal sufficient representations for the states by building a generative model of the environment with causal structural relationships. This differs from our approach as we do not attempt to compress or abstract the state or action space but rather we recover the causal interaction structure among action components in order to restrict the admissible policy class. Similar to us, Cao et al. (2025) attempt to solve the challenges imposed by large action spaces using causal reasoning: they show that it is possible to improve sample efficiency by leveraging learned causal relationships for counterfactual data augmentation and improved exploration. However, unlike us, they study an online setting.

3. Preliminaries

Problem setup. We consider finite-horizon Markov Decision Processes (MDPs), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{Z}, \mathcal{A}, p, r, \mu, \gamma, H)$, where $\mathcal{S}$ is the D_s-dimensional observed state space, $\mathcal{Z}$ is a possibly empty latent variable space that influences the state distribution, $\mathcal{A}$ is the D_a-dimensional action space, $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \mathbf{a}_t)$ denotes the transition dynamics, $r(\boldsymbol{s}_t, \mathbf{a}_t)$ is the reward function, $\mu(\boldsymbol{s}_0)$ is the initial state distribution, $\gamma \in [0, 1]$ is the discount factor and H is the time horizon. We define a policy $\pi(a_t|s_t)$ as a probabilistic mapping from a state to a distribution over possible next action.

Throughout, we use uppercase letters to denote random variables, lowercase letters for their instantiations, and bold lower case letters for vectors. We let $\mathbf{S}$ denote the observed state, $\mathbf{Z}$ the latent (unobserved) state, and $\mathbf{A}$ the action. We assume that $\mathbf{S}$ is a deterministic, invertible function of $\mathbf{Z}$. We denote by P the joint distribution over all MDP variables induced by the transition dynamics and the initial state distribution.

We assume access to an offline dataset $\mathcal{D} = \{(\boldsymbol{s}_t^{(i)}, \mathbf{a}_t^{(i)}, \boldsymbol{s}_{t+1}^{(i)})\}_{i=1,\dots,N;\, t=0,\dots,H-1}$, collected by executing an unknown behavior policy π_b. For simplicity, we assume that the reward is the terminal state s_{H-1}, but as we show in the experiment section, our results are applicable to more general reward functions.

We define the state-value function as $V^\pi(\boldsymbol{s}) = \mathbb{E}_\pi \left[\sum_{t=0}^{H-1} \gamma^{t-1} r_t \mid \boldsymbol{s}_0 = \boldsymbol{s}\right]$. We define the action-value function, $Q^\pi(\boldsymbol{s}, \mathbf{a})$, by further restricting the action taken from the starting state. Our goal is to learn an optimal policy π^* using only $\mathcal{D}$, such that $\pi^* = \arg\max_\pi \mathbb{E}_{s\sim\mu}[V^\pi(\boldsymbol{s})]$ using observational data only without online exploration.

Causal Structure of the MDP. We consider MDPs with structured action spaces where $\mathbf{a}$ can be partitioned into *groups*, $G_a = \{g_a^{(1)}, \dots, g_a^{(K)}\}$. Each group $g_a^{(k)} \subseteq \{1, \dots, D_a\}$ indexes a subset of actions such that $g_a^{(j)} \cap g_a^{(i)} = \emptyset$ and $\bigcup_k g_a^{(k)} = \{1, \dots, D_a\}$. We assume that each group affects the state transition and reward independently, meaning the transition dynamics and reward function *factor* over groups. We let $g_a(d)$ denote the group of d-th dimension of $\mathbf{a}$. We use the notation $\mathbf{a}_t^{g_a^{(k)}}$ to denote the vector of values indexed by $g_a^{(k)}$, at timestep t.

If the latent space $\mathcal{Z}$ is non-empty, we assume that there is one dimension of $\mathcal{Z}$ per action group, meaning $\mathcal{Z}$ is K-dimensional and $\mathbf{z} = [z^{(1)} \dots z^{(K)}]$. We assume that each action group $g_a^{(k)} \in G_a$ only effects one latent sub-state $z^{(k)}$.

Definition 1 states these independence properties of the MDP formally.

Definition 1 *[MDP factorization] Given $\boldsymbol{s}_t, \mathbf{a}_t, \boldsymbol{s}_{t+1}$ the following holds:*

$$p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \mathbf{a}_t) = \prod_{k=1}^{K} p_k(\mathbf{z}_{t+1}^{(k)} \mid \mathbf{z}_t^{(k)}, \mathbf{a}_t^{g_a^{(k)}}), \qquad r(\boldsymbol{s}_t, \mathbf{a}_t) \quad = \sum_{k=1}^{K} r_k(\mathbf{z}_t^{(k)}, \mathbf{a}_t^{(k)})$$

where $p_k : \mathcal{Z}^{(k)} \times \mathcal{A}^{g_a^{(k)}} \to \Delta(\mathcal{Z}^{(k)})$ and $r_k : \mathcal{Z}^{(k)} \times \mathcal{A}^{g_a^{(k)}} \to \mathbb{R}$.

The first equation states that the MDPs transition function factors over the action groups and the second states that the MDP's reward function factors over the action groups.

Furthermore, we assume the data-generating process is represented by a causal Directed Acyclic Graph (DAG) over the variables $\{\boldsymbol{S}_t, \boldsymbol{A}_t, \boldsymbol{S}_{t+1}\}$. Specifically, we assume that the distribution P satisfies the following:

Assumption 1 *[MDP DAG] The MDP is sampled from a distribution P that satisfies:*

(i) ***Acyclicity:*** *There exists no cycles between any subset of the variables.* [1]

(ii) ***Markov property:*** *P obeys the Markov property i.e.,*
$P(\boldsymbol{S}_{t+1}, \boldsymbol{A}_{t+1} | \boldsymbol{S}_{0:t}, \boldsymbol{A}_{0:t}) = P(\boldsymbol{S}_{t+1}, \boldsymbol{A}_{t+1} | \boldsymbol{S}_t, \boldsymbol{A}_t)$.

(iii) ***No hidden confounders:*** *All common causes of $A_t^{(i)}$, $A_t^{(j)}$ are observed.*

To facilitate our exposition, we present our analysis on a canonical DAG that satisfies assumption 1. Figure 1 (left) depicts this DAG, where action variables influence the next state via interactions mediated through the latent space $\mathbf{Z}$. We refer to this DAG as the entangled setting because the observed state variables are an entangled mixture of underlying latent components $\mathbf{Z}$. The figure illustrates a setting with $D = 3$ and grouping $G_a = \{\{1, 2\}, \{3\}\}$—meaning that action dimensions 1 and 2 interact while 3 acts independently. For example, when treating a patient, a clinician may observe the patient's vitals $(\mathbf{S})$ but not the underlying latent health state $(\mathbf{Z})$ that determine how treatments $(\mathbf{A})$ affect their body.

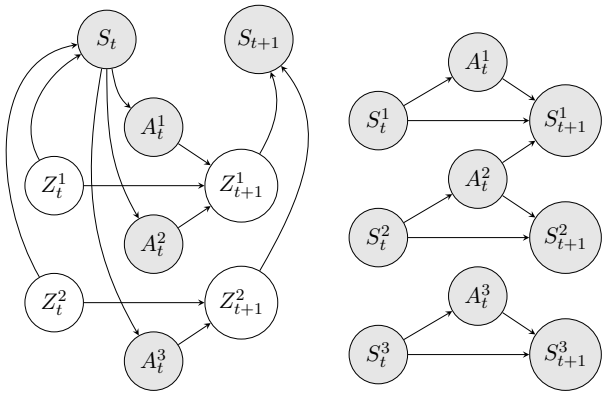


Figure 1: DAGs show causal relationships between MDP variables. Grey nodes are observed, white nodes are unobserved. Arrows denote a possible causal relationship between the variables. **Left: Entangled DAG** with latent variables. **Right: Disentangled DAG** with no latent variables. Since A_t^1 and A_t^2 affect Z_{t+1}^1 (in entangled DAG) and S_{t+1}^1 (in disentangled DAG) A_t^1 and A_t^2 are in the same action group. The DAGs show MDPs where the factored groups are $\{\{1, 2\}, \{3\}\}$.

In addition, we consider a special case that satisfies assumption 1 and further assumes that the observed states also factor in a way that corresponds to the action groupings G_a. This corresponds

1. The acyclicity assumption refers to the causal DAG, which represents the transition unfolded over time. Note that this is different from a state diagram, which is often used to represent the transition dynamics of an MDP, which could have cycles under our assumptions.

to a variant of the main setting where there is no latent $\mathbf{Z}$. We state this additional assumption formally:

Assumption 2 *[Disentangled State Factorization]*

(i) There exists a state grouping $G_s = \{g_s^{(1)}, \ldots, g_s^{(K)}\}$ where each group $g_s^{(k)} \subseteq \{1, \ldots, D_s\}$ indexes a subset of the state dimensions. We use the notation $g_s(d)$ to denote the state group that the d^{th} dimension of $\boldsymbol{S}$ belongs to and the notation $\mathbf{s}_t^{g_s^{(k)}}$ to denote the vector of values indexed by $g_s^{(k)}$, at timestep t. There exists a one-to-one mapping $G_a \to G_s$ such that each action group $g_a^{(k)} \in G_a$ affects only one state group, $g_s^{(k)} \in G_s$ and each state group only effects itself.

*(ii) **No hidden confounders**: All common causes of $A_t^{(i)}$, $A_t^{(j)}$, $S_t^{(l)}$, $S_{t+1}^{(l)}$, $S_{t+1}^{(m)}$ are observed.*

Similar to before, we present our analysis on a canonical DAG shown in figure 1 (right). This DAG retains the same action grouping G_a as figure 1 (left) but assumes that the observed state itself decomposes in accordance with the action groupings. This DAG, and generally assumption 2, is more relevant for MARL settings where the observable states of different agents are distinct from each other.

We stress that our analysis applies to any DAG satisfying our assumptions and is not limited to the examples in Figure 1. For example, although the figure depicts relatively simple behavior policies π_b, our results extend to more complex DAGs with richer dependencies among $\mathbf{Z}$, $\mathbf{S}$, and $\mathbf{A}$. Additional examples consistent with Definition 1 and Assumption 1 are in Appendix F.

4. Theory

Our goal is to leverage independencies implied by the causal structure to improve policy learning. To that end, we investigate two questions: **(1)** Can the causal structure be exploited to constrain the policy class leading to efficient policy learning? **(2)** Are this structure and the associated action groupings identifiable from observational data?

We answer both questions in the affirmative. The causal structure induces action groupings that constrain the policy space to factored policies, yielding more efficient learning. We also provide conditions under which the action groupings G_a are identifiable from observational data and develop minimal causal discovery methods to recover them. These results form the theoretical foundation of our causal structure learning and policy optimization framework.

4.1. Optimal policies of factored MDPs also factor

We show that if the data satisfies Assumption 1 and factors according to Definition 1, then there exists a factored policy that achieves the optimal value.

Formally, we define the factored policy class as follows:

$$\Pi(G_a) = \left\{ \pi : \pi(\boldsymbol{a}_t|\boldsymbol{s}_t) = \prod_{k=1}^{K} \pi_k(\mathbf{a}_t^{g_a^{(k)}}|\boldsymbol{s}_t), \pi \in \Pi \right\},$$

where Π is the set of all permissible (factored and unfactored) policies. $\Pi(G_a)$ is a restricted policy class in which each policy factorizes: at every time step the policies in this class are a product of the

sub-policies corresponding to each action group. Note that $\Pi(G_a) \subseteq \Pi$ and that $\Pi(G_a)$ assumes knowledge of the true G_a, an assumption that we will relax later in this section.

Our next result shows that whenever Definition 1 holds, any policy $\pi \in \Pi$ has an equivalent factored policy $\pi' \in \Pi(G_a)$ that achieves the same value.

Theorem 1 *Given an MDP which factors according to definition 1, then any policy $\pi \in \Pi$ has some corresponding policy $\pi' \in \Pi(G_a)$ with the same value function:*

$$Q^{\pi}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi'}(\boldsymbol{s}, \mathbf{a}) = \sum_{k=1}^{K} q^{\pi'_k}(\boldsymbol{s}, \mathbf{a}^{g_a^{(k)}}).$$

The proof for theorem 1 and all other statements in the main text is included in appendix. Intuitively, when each action group $\mathbf{a}^{g_a^{(k)}}$ influences the transition dynamics and rewards independently of the other groups, any unfactored policy admits an equivalent factored policy that selects $\mathbf{a}^{g_a^{(k)}}$ independently from other groups without altering the expected return.

Theorem 1 implies that when the MDP structure induces an action grouping G_a, the optimal value can always be achieved by a factored policy. This finding is valuable: it reduces the search space for an optimal policy from the full space Π to the smaller, structured class $\Pi(G_a)$, leading to better sample efficiency. However, to make use of this insight, we need to know the action groupings G_a. Next, we address this limitation by showing that G_a is identifiable from observational data.

4.2. Identifiability of the groupings G_a

Our key finding is that the action groupings G_a are identifiable via testable independence signatures that evaluate whether multiple actions are causal parents of the same sub-state. We first study the more general entangled setting depicted in figure 1 (left) and then examine the disentangled setting depicted in figure 1 (right) as a special case.

Identifiability under entangled observed states. First, we study the setting where the observed states are entangled and do not admit a factorization that corresponds with that of the actions.

Proposition 1 *Suppose P satisfies definition 1 and assumption 1, then for any pair of sub-actions $A^{(i)}, A^{(j)}$ with $i \neq j$, we have that $A_t^{(i)} \perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1} \iff g_a(i) \neq g_a(j)$*

Proposition 1 says that there is an independence signature that we can test for to identify if a pair of sub-actions belong in the same action group. Importantly, this independence signature relies on observed data only. The test checks if $A_t^{(i)}$ and $A_t^{(j)}$ effect the state transition independently as follows. Conditioning on the current state $\mathbf{S}_t$ blocks all confounding (backdoor) paths between the two actions. Conditioning on the future state $\mathbf{S}_{t+1}$ is equivalent to conditioning on the latent $\mathbf{Z}_{t+1}$ since the mapping from $\mathbf{Z}$ to $\mathbf{S}$ is deterministic and invertible. Conditioning on $\mathbf{Z}_{t+1}$, opens collider paths only when both actions jointly cause a sub-dimension of $\mathbf{Z}_{t+1}$. Thus, conditioning on $\mathbf{S}_t$ induces independence by blocking confounding, while $\mathbf{S}_{t+1}$ induces dependence if and only if the actions interact causally to shape the next state distribution.

Identifiability with disentangled observed states. Next, we study the disentangled setting in which the observed state space factorizes. Under Assumption 2, there exists a state partition G_s whose groups align with the action groups in G_a. As our next proposition shows, identifying the appropriate state groupings can help recover the corresponding action groups.

Proposition 2 *Let $\boldsymbol{S}_t^{(-l)}$ denote all the dimensions of $\boldsymbol{S}_t$ excluding l, and suppose P satisfies definition 1, and assumptions 1 and 2.*

(a) For some $S^{(l)}$ there exists $m \in g_s^{(k)}$ such that at least one of $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$, and $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ holds $\iff g_s(l) = k$.

(b) For a pair $A^{(i)}, A^{(j)}$ there exists $l, m \in g_s^{(k)}$ and $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ hold $\iff g_a(i) = g_a(j) = k$.

Proposition 2 provides an additional independence signature to identify the groupings when the observed state is disentangled. In part (a), the tests are symmetric: the first test, $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ checks if there exists an edge from $S_t^{(l)}$ to $S_{t+1}^{(m)}$ and the second test $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ checks if there is an edge from $S_t^{(m)}$ to $S_{t+1}^{(l)}$. If either edge is present, the states transition jointly and therefore belong to the same state group, according to assumption 2. The tests in part (b) check if there is an edge between each action and any state in the grouping.

While Propositions 1 and 2 both yield valid tests in the disentangled setting, the test implied by Proposition 2 is preferable when the state dimension exceeds the action dimension (i.e., $D_s > D_a$). This is because it avoids conditioning on a $2D_s$-dimensional variable, resulting in a simpler test.

Algorithm 1: Generic algorithm for estimating G_a

Input: $\mathcal{D}$
$K \leftarrow 0$;
$\widehat{G}_a \leftarrow \emptyset$;
for *each* (i, j) **do**
  if $A_t^{(i)} \not\perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ **then**
    if $g_a(i) = k$ ***or*** $g_a(j) = k$ **then**
      $g_a^{(k)} \leftarrow g_a^{(k)} \cup \{i, j\}$;
    end
    else
      $g_s^{(K)} \leftarrow \{i, j\}$;
      $K \leftarrow K + 1$;
    end
  end
end
Output: $\widehat{G}_a$

4.3. Finite sample considerations

In practice, the independence tests required for both the entangled and disentangled settings are conducted using finite samples and are therefore subject to statistical error. In the following corollary, we study the impact of using an estimated grouping $\widehat{G}_a$ with a bounded false negative rate $\leq \delta$ on the value of the optimal policy. We focus on the false negative rate because false positives do not result in a biased policy.

Corollary 1 *Let $\widehat{G}_a$ be a learned grouping such that $Pr[G_a \not\subseteq \widehat{G}_a] \leq \delta$, let $\pi^*_{G_a} := \arg\max_{\pi:\pi\in\Pi} \mathbb{E}_{s\sim\mu}[V^\pi(s)]$ and $\pi^*_{\widehat{G}_a} := \arg\max_{\pi:\pi\in\Pi(\widehat{G}_a)} \mathbb{E}_{s\sim\mu}[V^\pi(s)]$. We have that $\Pr[V_{\pi^*_{G_a}} < V_{\pi^*_{\widehat{G}_a}}] \leq \delta$.*

Corollary 1 implies that if the probability of missing any true dependencies in the estimated groupings $\widehat{G}_a$ is at most δ then the probability that the resulting policy fails to achieve the optimal value is also bounded by δ.

5. Approach

Leveraging our findings from section 4, we propose the Discovered Factorization RL approach (DiFaRL) which first learns the action groupings $\widehat{G}_a$ then identifies the optimal policy based on the learned factorization.

5.1. Learning the groupings $\widehat{G}_a$

We test the independence signatures in Propositions 1 and 2 using kernel conditional independence tests (KCIT) (Zhang et al., 2012). In settings where additional parametric assumptions about the relationship between different variables is available, more efficient parametric tests can be used. To account for the fact we are conducting multiple hypothesis testing, we set the p-value to be low $= 0.005$, following Zhang et al. (2012). Alternatively, a Bonferroni correction can be applied. We state the full procedures for the entangled settings in algorithm 1. We include the procedure for the disentangled setting in appendix A.

5.2. Policy optimization with learned groupings

Having obtained the estimated groupings, $\widehat{G}_a$, we use that grouping to inform the policy optimization step. We propose two fitted Q-iteration (FQI) (Riedmiller, 2005) approaches, though most offline RL algorithms can be adapted similarly.

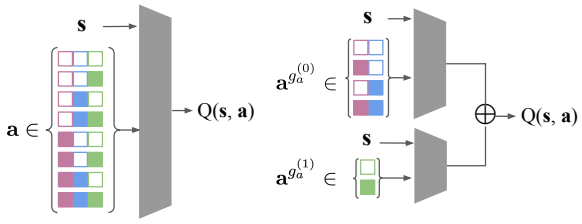


Figure 2: Two Q-network architectures for a binary action space with 3 sub-actions. **Left:** Dense network that evaluates all sub-action combinations. **Right:** Linearly decomposed network over learned action groupings.

5.2.1. LINEARLY DECOMPOSED FQI

A direct way to exploit the learned grouping $\widehat{G}_a$ is to incorporate it into the architecture of the Q-function. This induces a linear decomposition of Q consistent with $\widehat{G}_a$, as illustrated in Figure 2. We initialize $\widehat{Q}_0(\mathbf{s}_t, \mathbf{a}_t) = 0$, and for $t = 1, \ldots, T$ perform the iterative update

$$\widehat{Q}_t(\mathbf{s}, \mathbf{a}) = r_t + \gamma \sum_{g_a^{(k)} \in \widehat{G}_a} \max_{\mathbf{a}_t^{g_a^{(k)}}} Q_{t-1}\left(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}^{g_a^{(k)}}\right). \tag{1}$$

For each update the input to the Neural Network is the state-action pair and the output is the estimated Q-value. This decomposition can be interpreted as an extension of Tang et al. (2022), where we replace their fully factored action space with a learned factorization that better captures the underlying interaction structure. Equation 1 encodes the interactions based on the causal discovery results. However, a limitation of this linear decomposition is that it does not exploit the internal sub-action structure within the group. Meaning it treats all actions within a subgroup as equally influential in the interaction term, failing to capture heterogeneous contributions of individual actions within each subgroup. This shortcoming motivates our attention-based approach.

5.2.2. FQI WITH ATTENTION

Attention-based architectures (Vaswani et al., 2017) provide a flexible and expressive mechanism for modeling complex, state-dependent interactions among sub-actions Landers et al. (2025b). In settings with nonlinear and context-specific dependencies, attention layers can effectively capture these relationships.

To that end, we represent each sub-action with a learnable embedding, and state information is incorporated through Feature-wise Linear Modulation (FiLM) (Perez et al., 2018), yielding state-conditioned sub-action embeddings. Compared to simple concatenation of state and action features, FiLM offers a more parameter-efficient and structured means of modulating action representations based on the current state.

We then apply a masked self-attention layer that learns sub-action interactions. The attention mask is informed by the estimated grouping $\widehat{G}_a$: it enforces that only sub-actions within the same group may attend to one another. Attention weights between different groups are fixed to zero and are not updated during training. Specifically, let $A^{(h)}$ be the attention matrix for the h-th attention head. Then, we set $A^{(h)}_{ij} = 0$ if $g_a(i) \neq g_a(j)$.

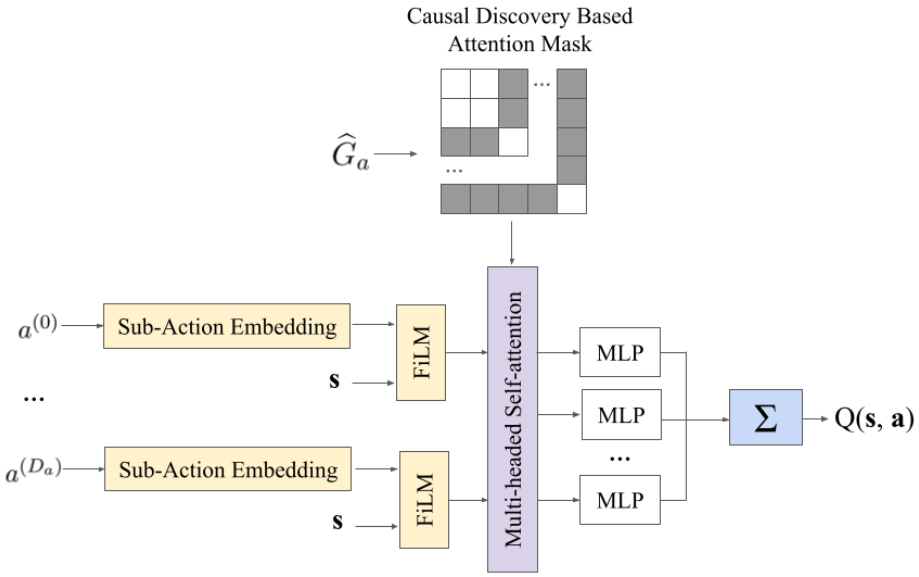


Figure 3: Overview of DiFaRL: the sub-action embeddings are conditioned on the state using FiLM. These are passed through a multi-headed self attention layer which is masked using the discovered groups $\widehat{G}_a$, injecting information from relevant sub-actions into each action embeddings. These embeddings are passed through an MLP and summed for the final Q-value estimate.

Finally, each attended sub-action representation is passed through its corresponding MLP, and the outputs are summed to produce the Q-value. The full architecture is shown in Figure 3.

6. Experiments

We evaluate our approach on a multi-agent setting, a simulated benchmark, and a healthcare task to assess the validity of our approach.

Baselines We compare our proposed approach against several baselines: **(1) Dense**: Dense network that not incorporate any factorization of the action space, **(2) Factored**: Assumes a full factorization of the sub-actions (Tang et al., 2022), **(3) Unfactored Attention**: FQI with an attention architecture between sub-actions, an offline value-based version of SAINT (Landers et al., 2025b), **(4) FOCUS**: A causal-discovery model based approach, only applies in the disentangled setting (Zhu et al., 2022), **(5) BraVE**: Dense network that selects and evaluates actions using a tree-structured action traversal (Landers et al., 2025a), **(6) Attention Oracle**: An unattainable model utilizes our attention architecture but has access to the ground truth mask G_a.

For our approach and FQI-based baselines, we set the maximum number of iterations for the fitted Q-iteration to 30 iterations using a neural network function approximator.

6.1. MARL: Simple Spread

We evaluate our method in the Multi-Particle Environments (MPE) Petting Zoo Simple Spread multi-agent environment. In this setting, agents must learn to cover landmarks while avoiding collisions. To test our approach, we construct a simulation where some agents interact while others behave independently by instantiating three parallel environments: one containing two interacting agents, and two additional environments each containing a single agent. Agents in the same environment interact by coordinating their behavior such that each agent navigates to a distinct landmark while simultaneously avoiding collisions with other agents; cross-instance interactions are not allowed.

This is an example of a disentangled setting, where the sub-actions correspond to the action of each agent and the sub-states correspond to the agent's states. We collect data using a uniformly random policy.

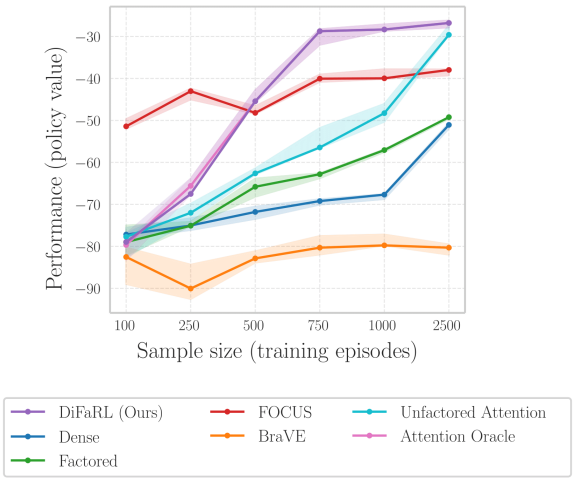


Figure 4: Performance in the Simple Spread simulator with a uniformly random behavioral policy over varying sample sizes. Plots display the median performance of the best FQI iteration over 5 runs. Performance is measured by evaluating in an online fashion in the simulator over 25 trials.

The results are shown in figure 4. Our approach performs comparably to the unattainable oracle model, and outperforms all models (excluding FOCUS) at all sample sizes. We achieve a median false negative rate (FNR) of 0 across all sample sizes. In very small samples, FOCUS achieves the best performance by leveraging state-space factorization for next state generation. However, since model-based approaches introduce bias through their structural assumptions, FOCUS stagnates with more data where other approaches continue to improve. In this example, FOCUS learns that each sub action effects only one sub-state, missing the fact that these actions interact through the state group. DiFaRL outperforms FOCUS with more data, because as a model-free approach it does not rely on a biased world model.

The factored approach is unable to attain high policy values since it introduces bias by ignoring interaction between the two agents in the same environment. The dense network and the unfactored

attention can in principle achieve high policy values but are unable to do so in small samples due to the higher complexity of their corresponding function classes.

6.2. Simulated Dataset

We evaluate our approach in a simulated setting with an entangled MDP. The objective is to stabilize patient's observed blood pressure (BP), which is influenced by the following unobserved factors: blood volume (BV), heart contraction strength (HC), and vascular resistance (VR). The clinician's action vector is $A = (A_{\text{IV}}, A_{\text{BB}}, A_{\text{PH}}, A_{\text{S}})$, where each binary variable indicates whether the corresponding treatment (intravenous fluid, beta-blockers, phenylephrine, or steroid) is administered.

We set up the simulation such that A_{PH} and A_{S} interact to jointly affect the state transition and reward while A_{BB} and A_{IV} influence the transition and reward independently. The clinician receives a non-negative reward if all unobserved variables are within their normal range. Full details of the transition and reward function are in appendix C.

We generate datasets with different sample sizes following different behavior policies, and run each episode for 20 time steps. We consider behavior policies that take the optimal action for each sub-action with probability ϵ and select randomly among non-optimal sub-actions with probability $1 - \epsilon$. We repeat each setting of sample size and behavior policy 10 times with a different random seed. We learn the policies offline with pre-collected data, but use the simulator to evaluate them in an "online" fashion, avoiding the need for off policy evaluation (OPE).

Figure 5 shows the results. The x-axis shows the sample size and the y-axis shows the policy value calculated analytically from the MDP's transition and reward function. Our approach performs better than all other feasible approaches across all sample sizes, especially when the sample size is small. Across all sample sizes, we have a median false negative rate of 0. The advantage of our model is most pronounced in small samples, indicating greater statistical efficiency than competing methods. By ignoring the interaction between A_{PH} and A_S, the factored policy introduces bias that persists even with large sample sizes. BraVE performs poorly especially in small samples because its tree traversal procedure does not leverage the underlying causal structure. FOCUS is excluded from the analysis since it requires that the state space be disentangled.

When the behavioral policy is non-uniform ($0 < \epsilon < 1$), certain states and actions are unlikely to be observed, causing all models to perform worse. We include results for $\epsilon = 1$ and $\epsilon = 0.2$ in appendix E.

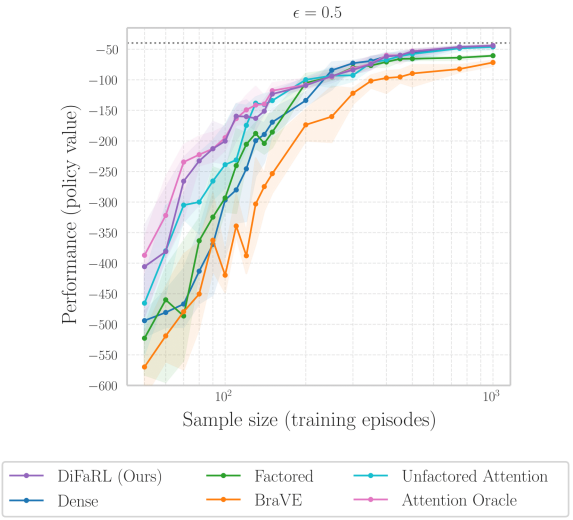


Figure 5: Performance in the blood pressure simulator with an ϵ greedy behavior policy ($\epsilon = 0.5$). Each plots displays the median performance of the best FQI iteration over 10 runs. The sample size is the number of training episodes and performance is measured analytically from the ground truth MDP.

Additionally, we include a stress-test relaxing the invertibility of the latent $\mathbf{Z}$ to the observed $\mathbf{S}$. We incorporate a NOTEARS based attention method as an additional baseline. We include both these results in appendix E.

6.3. Healthcare Data: Sepsis Treatment in MIMIC-III

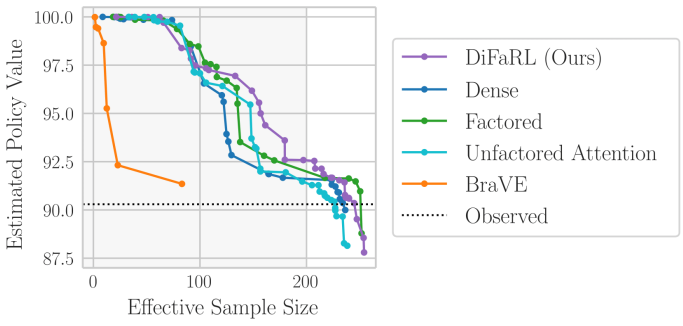
We apply our model to the task of ICU sepsis treatment using real healthcare data. To that end, we follow the same data extraction and experiment setup as Tang et al. (2022). Specifically, we extract a sepsis cohort from MIMIC-III and use a 70/15/15 split for training, validation and testing.

The data consists of 10 time-invariant demographic features and 33-dimensional time series data collected at 4-hour intervals. These measurements are from up to 24 hours before the onset of sepsis and up to 48 hours after. We use a recurrent neural network (RNN) with long short-term memory (LSTM) cells to create an approximate state that summarizes a patient's medical history into a D_s-dimensional $\mathbf{S}$ representing the entangled state vector.

Following Tang et al. (2022), we model the reward as 100 if the patient survives 48 hours and 0 if they do not; all intermediate rewards are 0. The actions are the volume of IV fluid administered and the amount of vasopressors. This results in a 5×5 action space. We adopt batch-constrained Q-learning (BCQ) (Fujimoto et al., 2019) and implement BCQ variants of Dense, Factored, Unfactored Attention, BraVE and DiFaRL. We do not implement FOCUS because it does not apply to the entangled setting.

We select the best model using OPE with the validation data set. We estimated the policy value using weighted importance sampling (WIS). We define the effective sample size (ESS) to be the equivalent number of samples that would yield the same amount of information if we could execute our policy in the MDP. Higher ESS implies that we can more accurately estimate the WIS from the test data. Following Tang et al. (2022), we estimate the behavioral policy using K nearest neighbors (KNN) in the embedding space and select our final policy to be the one with the highest value in the validation set and an ESS > 200 to ensure accurate value estimation. Because BraVE is not a BCQ variant, it does not remain close enough to the behavioral policy to achieve an ESS > 200, so we exclude it from the test data evaluation.

Figure 6 (left) shows the Pareto frontiers of the validation performance for the candidate policies. The figure shows that DiFaRL outperforms the baselines on the validation set, achieving a higher estimated policy values for the same ESS.



Policy	Test WIS
DiFaRL (Ours)	93.49 ± 1.83
Dense	93.47 ± 1.67
Factored	92.88 ± 1.85
Unfactored Attention	92.42 ± 1.93
Clinician	90.30 ± 0.64

Figure 6: **Left:** Pareto frontiers of the validation performance on the candidate policies. Shaded region is the region where the ESS is less than our prespecified cutoff (ESS < 200). **Right:** Performance on the test dataset. Standard deviation is measured from 100 bootstraps.

Figure 6 (right) shows test-set results, confirming that our approach achieves the best performance. Examining the learned groupings $\widehat{G}_a$, we observe that our method recovers a fully factored model, consistent with the findings from Tang et al. (2022). The improved performance over the Factored FQI, however, stems from architectural differences: the attention mechanism learns a weighted aggregation, enabling more influential sub-actions to exert greater impact on the Q-value.

7. Conclusion

We designed a two-step algorithm for efficient offline reinforcement that leverages sub-action group independence, without assuming complete factorization. By combining conditional independence testing for group discovery with a linear decomposition of the Q-function, our method enables scalable offline value-based RL while preserving expressivity. We further proposed an attention-based variant that constrains interactions within groups. Theoretically, we established that when an underlying grouping exists, any policy has a corresponding factored policy with identical action values. Empirically, we demonstrated that our approach achieves improvements over baselines, particularly in data-limited regimes. Together, these results highlight the promise of exploiting structured action spaces for advancing offline RL.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 2337529 and 2153083. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

We would like to thank the anonymous reviewers as well as Dylan Zapzalka, Marko Veljanovski and Katherine Anne Matton for their feedback.

References

András Antos, Csaba Szepesvári, and Rémi Munos. Fitted q-iteration in continuous action-space mdps. *Advances in neural information processing systems*, 20, 2007.

Hongye Cao, Fan Feng, Tianpei Yang, Jing Huo, and Yang Gao. Causal information prioritization for efficient reinforcement learning. *ICLR*, 2025.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

Mason Hargrave, Alex Spaeth, and Logan Grosenick. Epicare: a reinforcement learning benchmark for dynamic treatment regimes. *Advances in neural information processing systems*, 37:130536–130568, 2024.

Biwei Huang, Chaochao Lu, Liu Leqi, José Miguel Hernández-Lobato, Clark Glymour, Bernhard Schölkopf, and Kun Zhang. Action-sufficient state representation learning for control with structural constraints. In *International Conference on Machine Learning*, pages 9260–9279. PMLR, 2022.

Nan Jiang. A note on loss functions and error compounding in model-based reinforcement learning. *arXiv preprint arXiv:2404.09946*, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Matthew Landers, Taylor W Killian, Hugo Barnes, Thomas Hartvigsen, and Afsaneh Doryab. Brave: Offline reinforcement learning for discrete combinatorial action spaces. *Neurips*, 2025a.

Matthew Landers, Taylor W Killian, Thomas Hartvigsen, and Afsaneh Doryab. Saint: Attention-based modeling of sub-action dependencies in multi-action policies. *arXiv preprint arXiv:2505.12109*, 2025b.

Anson Lei, Bernhard Schölkopf, and Ingmar Posner. Spartan: A sparse transformer learning local causation. *arXiv preprint arXiv:2411.06890*, 2024.

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

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *AI&M*, 1(2):3, 2006.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

E. Richards, M. J. Lopez, and C. V. Maani. Phenylephrine. In *StatPearls [Internet]*. StatPearls Publishing, Treasure Island (FL), 2023. URL https://www.ncbi.nlm.nih.gov/books/NBK534801/.

Anna Riedmann, Philipp Schaper, and Birgit Lugrin. Reinforcement learning in education: A systematic literature review. *International Journal of Artificial Intelligence in Education*, pages 1–55, 2025.

Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European conference on machine learning*, pages 317–328. Springer, 2005.

Shengpu Tang, Maggie Makar, Michael Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. *Advances in Neural Information Processing Systems*, 35:34272–34286, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.

Zheng-Mao Zhu, Xiong-Hui Chen, Hong-Long Tian, Kun Zhang, and Yang Yu. Offline reinforcement learning with causal structured world models. *arXiv preprint arXiv:2206.01474*, 2022.

Appendix A. Algorithm for Estimating the Action Groupings with Disentangled States

Algorithm 2: Estimating G_a and G_s with disentangled states

Input: $\mathcal{D}$
$K_s \leftarrow 0$;
$K_a \leftarrow 0$;
$\widehat{G}_s \leftarrow \emptyset$;
for *each* (l, m) **do**
  if $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ **then**
    if $g_s(l) = k$ ***or*** $g_s(m) = k$ ***where*** $g_s^{(k)} \in \widehat{G}_s$ **then**
      $g_s^{(k)} \leftarrow g_s^{(k)} \cup \{l, m\}$;
    end
    else
      $g_s^{(K_s)} \leftarrow \{l, m\}$;
      $K_s \leftarrow K_s + 1$;
    end
  end
end
for *each* (i, l) **do**
  if $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$ **then**
    if $g_s(l) = k \in \widehat{G}_a$ **then**
      $g_a^{(k)} \leftarrow g_a^{(k)} \cup \{i\}$;
    end
    else
      $g_s^{(K_a)} \leftarrow \{i\}$;
      $K_a \leftarrow K_a + 1$;
    end
  end
end
Output: $\widehat{G}_a, \widehat{G}_s$

Appendix B. Detailed Theoretical Analyses

B.1. Identifiably Proofs

Proposition A1 *(Restatement of Proposition 1) Suppose P satisfies definition A1 and assumption 1, then for any pair of sub-actions $A^{(i)}, A^{(j)}$ with $i \neq j$, we have that $A_t^{(i)} \perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1} \iff g_a(i) \neq g_a(j)$*

Proof sketch. The proof proceeds by establishing that if $A_t^{(i)}, A_t^{(j)}$ do not interact, any potential pathways between the two will be blocked by conditioning on $\mathbf{S}_t$ If and only if $A_t^{(i)}, A_t^{(j)}$ do interact, the pathway between them is opened by conditioning on $\mathbf{S}_{t+1}$, which is equivalent to conditioning on $\mathbf{Z}_{t+1}$ by the assumption that $\mathbf{S}_{t+1}$ is an invertible deterministic function of $\mathbf{Z}_{t+1}$. This means that

the conditional independence will be satisfied if $A_t^{(i)}, A_t^{(j)}$ do not interact but it will not be satisfied if they do interact.

Proof [Proof of Proposition 1]

First, we will look at the causal structure induced by the MDP's transition function. For an MDP which admits a factored structure, there exists K latent sub-states corresponding to the K groups in G_a. By definition 1, the definition of a factored transition function, $Z_{t+1}^{(k)}$, depends only on $Z_t^{(k)}$, and all actions $A_t^{(i)} \in \mathbf{A}_t^{g_a^{(k)}}$, the group of actions indexed by $g_a^{(k)}$.

That means that the DAG will contain an edge $A_t^{(i)} \to Z_{t+1}^{(k)}$ from each $A^{(i)} \in \mathbf{A}^{g_a^{(k)}}$. So, if there are two actions $A^{(i)}, A^{(j)} \in \mathbf{A}^{g_a^{(k)}}$ there will be the edges $A_t^{(i)} \to Z_{t+1}^{(k)} \leftarrow A_t^{(j)}$. $Z_{t+1}^{(k)}$ acts as a collider on this path.

We are now ready to proceed with the main proof.

Step 1. First, we will show that if $A^{(i)}, A^{(j)}$ with $i \neq j$, and $A_t^{(i)} \perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ then $g_a(i) \neq g_a(j)$ We will proceed by contrapositive, showing if $i, j \in g_k$, i.e., if $g_a(i) = g_a(j)$ then $A_t^{(i)} \not\perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$.

By assumption, $\mathbf{S}$ is an invertible function of $\mathbf{Z}$. Therefore, conditioning on $\mathbf{S}$ is equivalent to conditioning on $\mathbf{Z}$ because no information is lost.

First, note that $\boldsymbol{S}_t$ precedes $A_t^{(i)}, A_t^{(j)}$. This means that if $\boldsymbol{S}_t$ affects $A_t^{(i)}, A_t^{(j)}$, it can only be a parent to either. That implies that any correlation induced due to $\boldsymbol{S}_t$ between $A_t^{(i)}, A_t^{(j)}$ must be because $\boldsymbol{S}_t$ is a common cause. By conditioning on $\boldsymbol{S}_t$, and by the assumption of no hidden confounders (assumption 1), all backdoor paths between $A_t^{(i)}, A_t^{(j)}$ are blocked. Since $\boldsymbol{S}_t$ can only be a parent to either $A_t^{(i)}, A_t^{(j)}$, it cannot be a collider on a pathway between $A_t^{(i)}$ and $A_t^{(j)}$.

Then, since $Z_{t+1}^{(k)}$ is a collider on a pathway between $A^{(i)}$ and $A^{(j)}$ and conditioning on $\mathbf{S}_{t+1}$ is equivalent to conditioning on $\mathbf{Z}_{t+1}$, conditioning on $\mathbf{S}_{t+1}$ opens a pathway between $A^{(i)}$ and $A^{(j)}$ if and only if $Z_{t+1}^{(k)}$ is affected by both $A^{(i)}$ and $A^{(j)}$, which occurs when $A_t^{(i)} \not\perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ i.e., when $g_a(i) = g_a(j)$.

Step 2. Next, we will show that if $g_a(i) \neq g_a(j)$ then $A_t^{(i)} \perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ holds. We will proceed by contrapositive, showing if $A_t^{(i)} \not\perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ then $g_a(i) = g_a(j)$.

Assume that $A_t^{(i)} \not\perp A_t^{(j)} \mid \mathbf{S}_t, \mathbf{S}_{t+1}$ holds. That means that an open pathway exists between $A_t^{(i)}$ and $A_t^{(j)}$. Following the same logic as step 1, conditioning on $\boldsymbol{S}_t$, and by the assumption of no hidden confounders (assumption 1), all backdoor paths between $A_t^{(i)}, A_t^{(j)}$ are blocked.

By the definition of the policy π, actions are only a function the previous state, so a direct edge from $A_t^{(i)}$ is impossible $A_t^{(j)}$. Therefore, there must be a pathway induced by conditioning on $\mathbf{S}_{t+1}$. Since $\mathbf{S}_{t+1}$ is an invertible function of $\mathbf{Z}_{t+1}$, conditioning on $\mathbf{S}_{t+1}$ is equivalent to conditioning on $\mathbf{Z}_{t+1}$. So, since conditioning on $\mathbf{Z}_{t+1}$ opens a pathway between $A_t^{(i)}$ and $A_t^{(j)}$, it must be a collider on this path, meaning $A_t^{(i)}$ and $A_t^{(j)}$ must both be causal parents of at least one dimension of $\mathbf{Z}_{t+1}$. So, by definition 1 which states that each $Z_{t+1}^{(k)}$, depends only on $Z_t^{(k)}$, and all actions $A_t^{(i)} \in \mathbf{A}_t^{g_a^{(k)}}$, these actions must be in the same $g_a^{(k)}$, ie $g_a(i) = g_a(j)$.

■

Proposition A2 *(Restatement of proposition 2) Let $\boldsymbol{S}_t^{(-l)}$ denote all the dimensions of $\boldsymbol{S}_t$ excluding l, and suppose P satisfies definition 1, and assumption 2.*

(a) For some $S^{(l)}$ there exists $m \in g_s^{(k)}$ such that at least one of $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$, and $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ holds $\iff g_s(l) = k$.

(b) For a pair $A^{(i)}, A^{(j)}$ there exists $l, m \in g_s^{(k)}$ and $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ hold $\iff g_a(i) = g_a(j) = k$.

Proof sketch. If $A_t^{(i)}$ and $A_t^{(j)}$ interact, then they will both be a causal parent of at least one sub-state in the corresponding state group.

If $S_t^{(l)}$ and $S_{t+1}^{(m)}$ are not conditionally independent given the other previous states and actions, then $S_t^{(l)}$ must be a causal parent to $S_{t+1}^{(m)}$ meaning these states belong to the same state group. If $A_t^{(i)}$ is not independent to $S_{t+1}^{(l)}$ conditioned on the previous state, then $A_t^{(i)}$ is a causal parent of $S_{t+1}^{(l)}$. If $A_t^{(i)}$ and $A_t^{(j)}$ are causal parents to states in the same state group, they belong to the same action group.

Proof [Proof of Proposition 2]
Proof of part (a).
Step 1. If for some $S^{(l)}$ there exists $m \in g_s^{(k)}$ such that at least one of $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ and $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ hold then $g_s(l) = k$.

If $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ holds then $S_t^{(l)}$ is a causal parent of $S_{t+1}^{(m)}$. By the Markov property in assumptions 1, states and actions can only depend on previous states and actions. In assumption 2, we assume no hidden confounders between $A_t^{(i)}$, $A_t^{(j)}$, $S_t^{(l)}$, $S_{t+1}^{(l)}$, $S_t^{(m)}$. So, by conditioning on $\boldsymbol{S}_t^{(-l)}$ and $\mathbf{A}_t$, we have blocked all pathways between $S_t^{(l)}$ and $S_{t+1}^{(m)}$ other than a direct edge from $S_t^{(l)}$ to $S_{t+1}^{(m)}$. Furthermore $\boldsymbol{S}_t^{(\neg m)}$ and $\boldsymbol{A}_t$ cannot be colliders on this pathway. The transition function is defined as $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \mathbf{a}_t)$ meaning $\boldsymbol{S}_t^{(\neg m)}$ cannot have an edge to $S_t^{(l)}$. By the definition of a policy $\pi(a_t|s_t)$ there cannot be an edge from $\boldsymbol{A}_t$ to $S_t^{(l)}$. So, $\boldsymbol{S}_t^{(\neg m)}$ and $\boldsymbol{A}_t$ will not be colliders on this path.

It follows that if $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ then $S_t^{(l)}$ is a causal parent of $S_{t+1}^{(m)}$. Following the same reasoning, if $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ holds then $S_t^{(m)}$ is a causal parent of $S_{t+1}^{(m)}$. Since $S^{(l)}$ either affects or is affected by a state in $g_s^{(k)}$ then by assumption 2 $g_s(l) = k$.

Step 2. If $g_s(l) = k$, then there exists $m \in g_s^{(k)}$ such that at least one of $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ and $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ hold.

Assume $g_s(l) = k$. Then by assumption 2 there must exist some $m \in g_s^{(k)}$ such that either $S_t^{(m)}$ is a causal parent of $S_{t+1}^{(l)}$ or $S_t^{(l)}$ is a causal parent of $S_{t+1}^{(m)}$. Therefore, by the same logic as step 1, if $S_t^{(l)}$ is a causal parent of $S_{t+1}^{(m)}$ then $S_t^{(l)} \not\perp S_{t+1}^{(m)} \mid \boldsymbol{S}_t^{(-l)}, \mathbf{A}_t$ will hold. If $S_t^{(m)}$ is a causal parent of $S_{t+1}^{(l)}$ then $S_t^{(m)} \not\perp S_{t+1}^{(l)} \mid \boldsymbol{S}_t^{(-m)}, \mathbf{A}_t$ will hold.

Proof of part (b).
Step 1. For a pair $A^{(i)}, A^{(j)}$ if there exists $l, m \in g_s^{(k)}$ and $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ hold then $g_a(i) = g_a(j) = k$.

Assume that for a pair $A^{(i)}, A^{(j)}$ there exists $l, m \in g_s^{(k)}$ such that $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$. First, let's show that if $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$ then $A_t^{(i)}$ is a causal parent of $S_{t+1}^{(l)}$.

By the Markov property in assumptions 1, states and actions can only depend on previous states and actions. In assumption 2, we assume no hidden confounders between $A_t^{(i)}$, $A_t^{(j)}$, $S_{t+1}^{(l)}$, $S_{t+1}^{(m)}$. So, by conditioning on $\boldsymbol{S}_t$, we have blocked all pathways between $S_{t+1}^{(l)}$ and $A_t^{(i)}$ other than a direct edge from $A_t^{(i)}$ to $S_{t+1}^{(l)}$. It follows that if $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$ then $A_{t-1}^{(i)}$ is a causal parent of $S_t^{(l)}$. By the same logic, $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ implies that $A_t^{(j)}$ is a causal parent of $S_{t+1}^{(m)}$. If $A_t^{(i)}$ and $A_t^{(j)}$ are both causal parents to at least one state in a state group k, then $A^{(i)}$ and $A^{(j)}$ are in the same action group $g_a(i) = g_a(j) = k$ by assumption 2.

Step 2. If $g_a(i) = g_a(j) = k$ then for $A^{(i)}, A^{(j)}$ there exists $l, m \in g_s^{(k)}$ such that $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ hold.

Let $g_a(i) = g_a(j) = k$. Then, by assumption 2 $A^{(i)}, A^{(j)}$ affect actions one state group $g_s^{(k)}$, so $A^{(i)}$ and $A^{(j)}$ will both be causal parents to at least on state in $g_s^{(k)}$. Meaning, there exists $l, m \in g_s^{(k)}$ such that $A^{(i)}$ is a causal parent to $S_{t+1}^{(l)}$ and $A^{(j)}$ is a causal parent to $S_{t+1}^{(m)}$. By the same reasoning as step 1, it follows that $S_{t+1}^{(l)} \not\perp A_t^{(i)} \mid \boldsymbol{S}_t$, and $S_{t+1}^{(m)} \not\perp A_t^{(j)} \mid \boldsymbol{S}_t$ hold. ■

B.2. Factored Value Functions

We first review some important background on state abstractions, which allow us to prove the following statements in a more general setting.

Background on State Abstractions. A state abstraction (also known as state aggregation) Li et al. (2006), is a mapping $\phi : \mathcal{S} \to \Omega$ that converts each element of the primitive state space $\mathcal{S}$ to an element of the abstract state space Ω. Intuitively, if two states s_1 and s_2 are mapped to the same element under ϕ, i.e., $\phi(s_1) = \phi(s_2)$, then they are treated as the same (abstract) state under the abstraction. Therefore, we can view an abstraction as a partitioning of the primitive state space into non-overlapping subsets. Since a state abstraction is a many-to-one mapping, we define its inverse as $\phi^{-1}(\omega) = \{\tilde{\boldsymbol{s}} : \phi(\tilde{\boldsymbol{s}}) = \omega\}$, a set containing all primitive states that are mapped to the abstract state ω.

We have the following property of summations involving state abstractions, where for any function $f : \mathcal{S} \to \mathbb{R}$,

$$\sum_{\boldsymbol{s} \in \mathcal{S}} f(\boldsymbol{s}) = \sum_{\omega \in \Omega} \sum_{\tilde{\boldsymbol{s}} \in \phi^{-1}(\omega)} f(\tilde{\boldsymbol{s}})$$

To understand this property, let us consider the sum of $f(\boldsymbol{s})$ for all states in $\mathcal{S}$ which can be obtained in two different ways: i) directly iterating through the elements of $\mathcal{S}$, ii) first iterating through the partitions of $\mathcal{S}$ (induced by the abstraction), and then iterating through the elements in each partition, giving rise to the double summation. This property allows us to change the index of summation from primitive states to abstract states. For multiple abstractions $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_K]$

where $\phi_k \neq \phi_{k'}$ if $k \neq k'$, denoting $\boldsymbol{\omega} = \boldsymbol{\phi}(s) = [\omega_1, \ldots, \omega_K]$, we can similarly define the inverse abstraction $\boldsymbol{\phi}^{-1}(\boldsymbol{\omega}) = \{\tilde{\boldsymbol{s}} : \boldsymbol{\phi}(\tilde{\boldsymbol{s}}) = \boldsymbol{\omega}\}$, and the summation property similarly applies.

We restate definition definition 1 using state abstractions to prove the following statements more generally.

Definition A1 *(Restatement of definition 1 using state abstractions) For an MDP which admits a factored structure, there exists K state abstractions corresponding to the K groups in G. $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_K]$ where $\phi_k : \mathcal{S} \to \Omega_k$, $\omega_k = \phi_k(\boldsymbol{s})$, such that for all $\boldsymbol{s}_t, \mathbf{a}_t, \boldsymbol{s}_{t+1}$ the following holds:*

$$\sum_{\tilde{\boldsymbol{s}} \in \boldsymbol{\phi}^{-1}(\boldsymbol{\phi}(\boldsymbol{s}_{t+1}))} p(\tilde{\boldsymbol{s}} | \boldsymbol{s}_t, \mathbf{a}_t) = \prod_{k=1}^{K} p_k(\omega_{t+1}^{(k)} | \omega_t^{(k)}, \mathbf{a}_t^{g_a^{(k)}}) \tag{2}$$

$$r(\boldsymbol{s}_t, \mathbf{a}_t) = \sum_{k=1}^{K} r_k(\omega_t^{(k)}, \mathbf{a}_t^{g_a^{(k)}}) \tag{3}$$

For some $p_k : \Omega_g \times \mathcal{A}^{g_a^{(k)}} \to \Delta\Omega_k$ and $r_k : \Omega_k \times \mathcal{A}^{g_a^{(k)}} \to \mathbb{R}$.

Theorem A1 *(Restatement of Theorem 1) Given an MDP which factors according to definition A1, then any policy $\pi \in \Pi$ has some corresponding policy $\pi' \in \Pi(G)$ with an equivalent value function:*

$$Q^{\pi}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi'}(\boldsymbol{s}, \mathbf{a}) = \sum_{k=1}^{K} q^{\pi'_k}(\boldsymbol{s}, \mathbf{a}^{g_a^{(k)}}).$$

Proof [Proof of Theorem 1] Without loss of generality, we consider the setting with $D = 2$, the extension to $D > 2$ is straightforward. The proof is based on mathematical induction on a sequence of h-step Q-functions of π defined as $Q^{\pi,(h)}(\boldsymbol{s}, \mathbf{a}) = \mathbb{E}[\sum_{t=1}^{h} \gamma^{t-1} r_t | \boldsymbol{s}_1 = \boldsymbol{s}, \mathbf{a}_1 = \mathbf{a}, \mathbf{a}_t \sim \pi]$.

Base case. For $h = 1$, the one-step Q-function is the reward, which by assumption is $r(\boldsymbol{s}, \mathbf{a}) = r_1(\omega_1, \mathbf{a}^{g_a^{(1)}}) + r_2(\omega_2, \mathbf{a}^{g_a^{(2)}})$. By virtue of being the first time step ($h = 1$), the policy does not affect the Q value except through $\mathbf{a}^{g_a^{(1)}}, \mathbf{a}^{g_a^{(2)}}$. So for any policies π'_1 and π'_2, $Q^{\pi,(1)}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi',(1)}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi'_1,(1)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) + Q^{\pi'_2,(1)}(\omega_2, \mathbf{a}^{g_a^{(2)}})$.

Inductive Step. Suppose $Q^{\pi,(h)}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi',(h)}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi'_1,(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) + Q^{\pi'_2,(h)}(\omega_2, \mathbf{a}^{g_a^{(2)}})$ where $\pi'_1(\mathbf{a}'^{g_a^{(1)}}, \omega'_1) = \sum_{\omega'_2} p(\omega'_2 | \omega_2, \mathbf{a}^{g_a^{(2)}}) \sum_{\mathbf{a}'^{g_a^{(2)}}} \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega'_1, \omega'_2)$ and $\pi'_2(\mathbf{a}'^{g_a^{(1)}}, \omega'_1) = \sum_{\omega'_1} p(\omega'_1 | \omega_1, \mathbf{a}^{g_a^{(1)}}) \sum_{\mathbf{a}'^{g_a^{(1)}}} \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega'_1, \omega'_2)$. We can express $Q^{\pi,(h+1)}(\boldsymbol{s}, \mathbf{a})$ in terms of $Q^{\pi,(h)}(\boldsymbol{s}, \mathbf{a})$ using the Bellman equation:

$$Q^{\pi,(h+1)}(\boldsymbol{s}, \mathbf{a}) = r(\boldsymbol{s}, \mathbf{a}) + \gamma \sum_{\boldsymbol{s}'} p(\boldsymbol{s}' | \boldsymbol{s}, \mathbf{a}) \sum_{a'} \pi(\mathbf{a}' | \boldsymbol{s}') Q^{\pi,(h)}(\boldsymbol{s}, \mathbf{a})$$

$$
\begin{aligned}
Q^{\pi,(h+1)}(\boldsymbol{s},\mathbf{a}) &= r(\boldsymbol{s},\mathbf{a}) + \gamma \sum_{\boldsymbol{s}'} p(\boldsymbol{s}'|\boldsymbol{s},\mathbf{a}) \sum_{\mathbf{a}'} \pi(\mathbf{a}'|\boldsymbol{s}') Q^{\pi,(h)}(\boldsymbol{s},\mathbf{a}) \\
&= r(\boldsymbol{s},\mathbf{a}) + \gamma \sum_{\boldsymbol{\omega}'} \sum_{\tilde{\boldsymbol{s}} \in \boldsymbol{\phi}^{-1}(\boldsymbol{\omega}')} p(\tilde{\boldsymbol{s}}|\boldsymbol{s},\mathbf{a}) \sum_{\mathbf{a}'} \pi(\mathbf{a}'|\tilde{\boldsymbol{s}}) Q^{\pi,(h)}(\tilde{\boldsymbol{s}},\mathbf{a}) \\
&= r(\boldsymbol{s},\mathbf{a}) + \gamma \sum_{\boldsymbol{\omega}'} \sum_{\tilde{\boldsymbol{s}} \in \boldsymbol{\phi}^{-1}(\boldsymbol{\omega}')} p(\tilde{\boldsymbol{s}}|\boldsymbol{s},\mathbf{a}) \sum_{\mathbf{a}'} \pi(\mathbf{a}'|\tilde{\boldsymbol{s}}) \\
&\quad \times \left(Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) + Q^{\pi_2',(h)}(\omega_2, \mathbf{a}^{g_a^{(2)}}) \right) \\
&= r(\boldsymbol{s},\mathbf{a}) + \gamma \sum_{\omega_1'} \sum_{\omega_2'} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}})\, p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \\
&\quad \times \sum_{\mathbf{a}'^{g_a^{(1)}}} \sum_{\mathbf{a}'^{g_a^{(2)}}} \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega_1', \omega_2') \left(Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) + Q^{\pi_2',(h)}(\omega_2, \mathbf{a}^{g_a^{(2)}}) \right) \\
&= r_1(\omega_1, \mathbf{a}^{g_a^{(1)}}) + r_2(\omega_2, \mathbf{a}^{g_a^{(2)}}) \\
&\quad + \gamma \sum_{\omega_1'} \sum_{\omega_2'} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}})\, p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \sum_{\mathbf{a}'^{g_a^{(1)}}} \sum_{\mathbf{a}'^{g_a^{(2)}}} \\
&\quad \times \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega_1', \omega_2')\, Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) \\
&\quad + \gamma \sum_{\omega_1'} \sum_{\omega_2'} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}})\, p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \\
&\quad \times \sum_{\mathbf{a}'^{g_a^{(1)}}} \sum_{\mathbf{a}'^{g_a^{(2)}}} \pi \mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega_1', \omega_2')\, Q^{\pi_2',(h)}(\omega_2, \mathbf{a}^{g_a^{(2)}}).
\end{aligned}
$$

Gathering all the terms that can be written as a function of $\omega_1, \mathbf{a}^{g_a^{(1)}}$, we get:

$$
\begin{aligned}
& r_1(\omega_1, \mathbf{a}^{g_a^{(1)}}) + \gamma \sum_{\omega_1'} \sum_{\omega_2'} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}}) p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \\
&\quad \times \sum_{\mathbf{a}'^{g_a^{(1)}}} \sum_{\mathbf{a}'^{g_a^{(2)}}} \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega_1', \omega_2') Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) \\
=& r_1(\omega_1, \mathbf{a}^{g_a^{(1)}}) + \gamma \sum_{\omega_1'} \sum_{\mathbf{a}'^{g_a^{(1)}}} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}}) \sum_{\omega_2'} p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \\
&\quad \times \sum_{\mathbf{a}'^{g_a^{(2)}}} \pi(\mathbf{a}'^{g_a^{(1)}}, \mathbf{a}'^{g_a^{(2)}} | \omega_1', \omega_2') Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) \\
=& r_1(\omega_1, \mathbf{a}^{g_a^{(1)}}) + \gamma \sum_{\omega_1'} \sum_{\mathbf{a}'^{g_a^{(1)}}} p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}}) \pi_1(\mathbf{a}'^{g_a^{(1)}}, \omega_1') Q^{\pi_1',(h)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) \\
=& Q^{\pi_1',(h+1)}(\omega_1, \mathbf{a}^{g_a^{(1)}})
\end{aligned}
$$

A similar argument can be made with respect to terms that can be written as a function of ω_2 and $\mathbf{a}^{g_a^{(2)}}$. Specifically,

$$\begin{aligned} r_2(\omega_2, \mathbf{a}^{g_a^{(2)}}) \; + \; \gamma \sum_{\omega_1'} \sum_{\omega_2'} & p(\omega_1'|\omega_1, \mathbf{a}^{g_a^{(1)}}) p(\omega_2'|\omega_2, \mathbf{a}^{g_a^{(2)}}) \\ & \times \sum_{\mathbf{a'}^{g_a^{(1)}}} \sum_{\mathbf{a'}^{g_a^{(2)}}} \pi(\mathbf{a'}^{g_a^{(1)}}, \mathbf{a'}^{g_a^{(2)}} | \omega_1', \omega_2') Q^{\pi_2', (h)}(\omega_2, \mathbf{a}^{g_a^{(2)}}) \\ & = Q^{\pi_2', (h+1)}(\omega_2, \mathbf{a}^{g_a^{(2)}}) \end{aligned}$$

It follows that $Q^{\pi,(h+1)}(\boldsymbol{s}, \mathbf{a}) = Q^{\pi_1',(h+1)}(\omega_1, \mathbf{a}^{g_a^{(1)}}) + Q^{\pi_2',(h+1)}(\omega_2, \mathbf{a}^{g_a^{(2)}})$.

By mathematical induction, this decomposition holds for any h-step Q-function. Letting $h \rightarrow \infty$ shows that this holds for the full Q-function. ■

B.3. Error Bounds

Corollary A1 *Given a grouping $\widehat{G}_a$ such that Pr$[G_a \not\subseteq \widehat{G}_a] \leq \delta$, let $\pi^*_{G_a} := \arg\max_{\pi:\pi\in\Pi} \mathbb{E}_{s\sim\mu}[V^\pi(s)]$ and $\pi^*_{\widehat{G}_a} := \arg\max_{\pi:\pi\in\Pi(\widehat{G}_a)} \mathbb{E}_{s\sim\mu}[V^\pi(s)]$. We have that*

$$\Pr[V_{\pi^*_{G_a}} < V_{\pi^*_{\widehat{G}_a}}] \leq \delta.$$

Proof [Proof of Corollary 1] We have that $\Pr[G_a \not\subseteq \widehat{G}_a] \leq \delta$, so $\Pr[G_a \subseteq \widehat{G}_a] > 1-\delta$. If $G_a \subseteq \widehat{G}_a$, then $\Pi(G_a) \subseteq \Pi(\widehat{G}_a)$ meaning that $V_{\pi^*_{G_a}} = V_{\pi^*_{\widehat{G}_a}}$. $\Pr[V_{\pi^*_{G_a}} = V_{\pi^*_{\widehat{G}_a}}] > 1-\delta$ so it follows that $\Pr[V_{\pi^*_{G_a}} < V_{\pi^*_{\widehat{G}_a}}] \leq \delta$.

■

Appendix C. Blood Pressure Simulation Details

The observed blood pressure (BP) is a function of the underlying latent health states: blood volume (BV), heart contraction rate (HC) and vascular resistance (VR).

$$BP = VR * 5^2 + HC * 5 + BV$$

Transition Function:
The latent state variables transition acording to the following tables:

Reward Function: The rewards are defined per latent state to reflect the respective significance of each on patient health.

A_{IV}	BV_{t+1}	**Probability**
1	$\text{clip}(BV_t + 1, 0, 4)$	0.56
1	$\text{clip}(BV_t + 2, 0, 4)$	0.14
1	BV_t	0.3
0	$\text{clip}(BV_t - 1, 0, 4)$	0.10
0	BV_t	0.90

A_{BB}	HC_{t+1}	**Probability**
1	$\text{clip}(HC_t - 1, 0, 4)$	0.80
1	$\text{clip}(HC_t, 0, 4)$	0.20
0	$\text{clip}(HC_t + 1, 0, 4)$	0.10
0	HC_t	0.90

A_{PH}	A_S	VR_{t+1}	**Probability**
1	1	$\text{clip}(VR_t + 1, 0, 4)$	0.90
1	1	$\text{clip}(VR_t, 0, 4)$	0.10
1	0	$\text{clip}(VR_t, 0, 4)$	1.00
0	1	$\text{clip}(VR_t - 1, 0, 4)$	0.10
0	1	$\text{clip}(VR_t, 0, 4)$	0.90
0	0	$\text{clip}(VR_t - 1, 0, 4)$	0.10
0	0	$\text{clip}(VR_t, 0, 4)$	0.90

$$r_{BV}(x) = \begin{cases} -4, & x = 0, \\ -2, & x = 1, \\ 0, & x = 2, \\ 0, & x = 3, \\ -1, & x = 4, \end{cases} \quad r_{HC}(x) = \begin{cases} -3, & x = 0, \\ -1, & x = 1, \\ 0, & x = 2, \\ 0, & x = 3, \\ -1, & x = 4, \end{cases} \quad r_{VR}(x) = \begin{cases} -4, & x = 0, \\ -2, & x = 1, \\ 0, & x = 2, \\ -2, & x = 3, \\ -4, & x = 4. \end{cases}$$

The total reward is a sum of the per-substate reward:

$$r(\boldsymbol{z}) = \sum_{i \in [BV, HC, VR]} r_i(z_i)$$

Appendix D. Simulation Implementation Details

D.1. Simple Spread

Attention Models
We use an embedding dimension of 64 and 8 attention heads. We have 2 hidden layers and 512 hidden units. We use a learning rate of 1e-3 and a weight decay of 0.
Dense Model
We have 2 hidden layers and 512 hidden units. We use a learning rate of 1e-3 and a weight decay of 1e-5.

Factored Model
We have 2 hidden layers and 512 hidden units. We use a learning rate of 3e-4 and a weight decay of 1e-5.

FOCUS
For the world model we have 3 hidden layers, with dimension 512. We use a learning rate of 3e-4 and a weight decay of 1e-4. For the Q-network we use an action-in architecture with 1024 hidden units and 2 layers. We use a learning weight of 1e-4 and weight decay of 1e-4.

BraVE
We follow the hyperparameters in the original implementation of BraVE. We have 6 hidden layers with dimension 256. We use a learning rate of 2.5e-4 and a weight decay of 0.

D.2. Blood Pressure Simulator

Attention Models
We use an embedding dimension of 32, 4 attention heads, a hidden size of 128 and a hidden dimension of 2. We use a learning rate of 3e-4.

Dense Model
We use a dimension of 128 with 3 hidden layers. We use a learning rate of 3e-4.

Factored Model
We use a dimension of 128 with 3 hidden layers. We use a learning rate of 3e-4.

BraVE
We use a hidden size of 256 with a hidden dimension of 2. We use a learning rate of 1e-3. We ran a version of BraVE with the same hyperparameters as the other models (A dimension of 128 with 3 hidden layers and a learning rate of 3e-4) which caused it to perform worse. We include those results here.

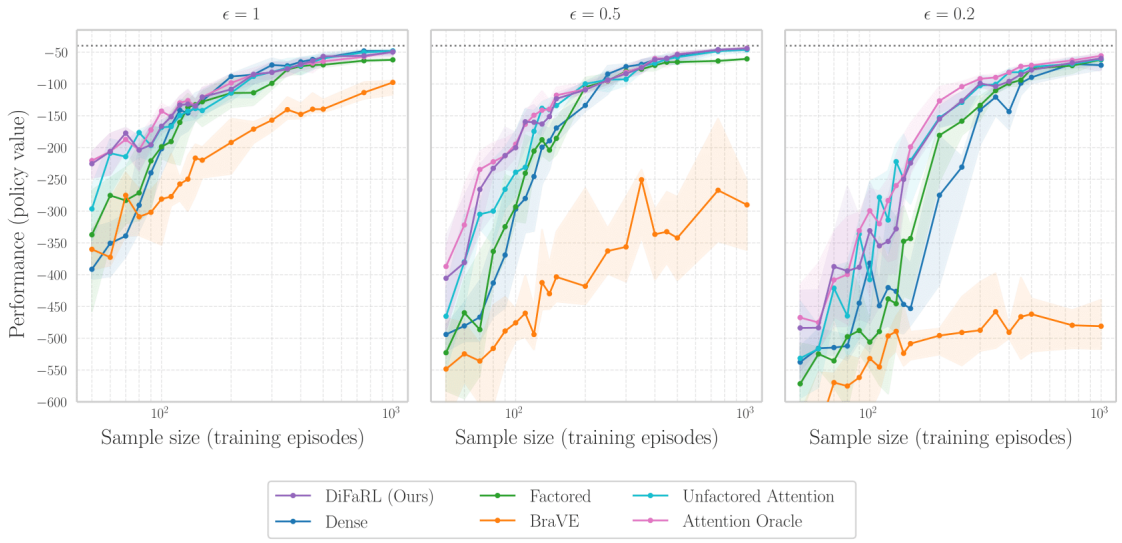


Figure 7: BraVE results with the same hyperparameter config as the other baselines.

Appendix E. Additional Blood Pressure Simulation Results

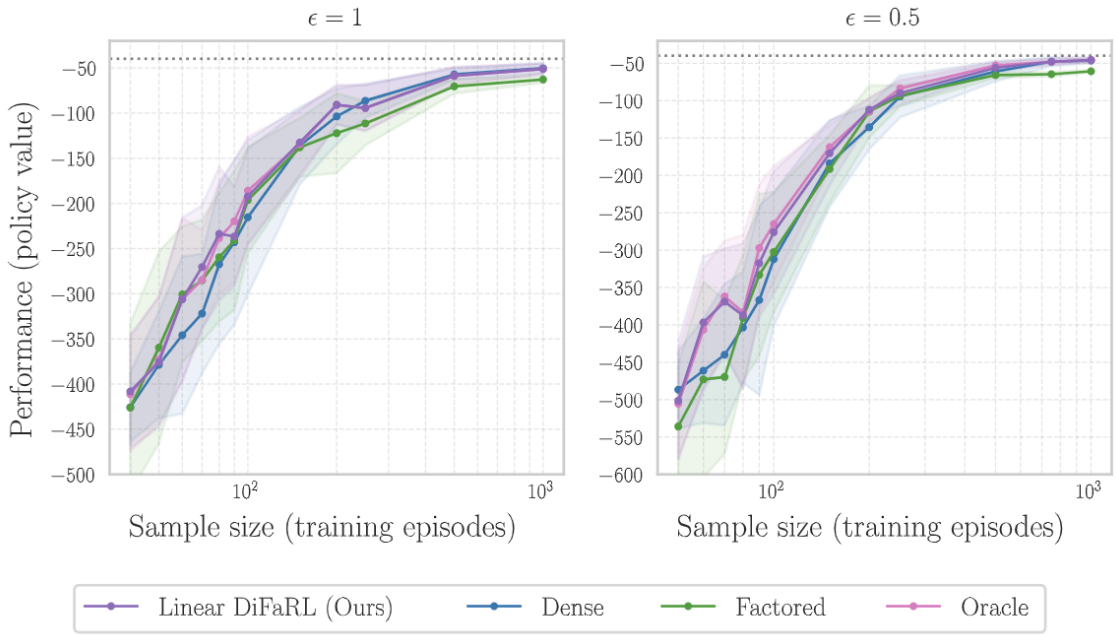


Figure 8: Linearly decomposed FQI results.

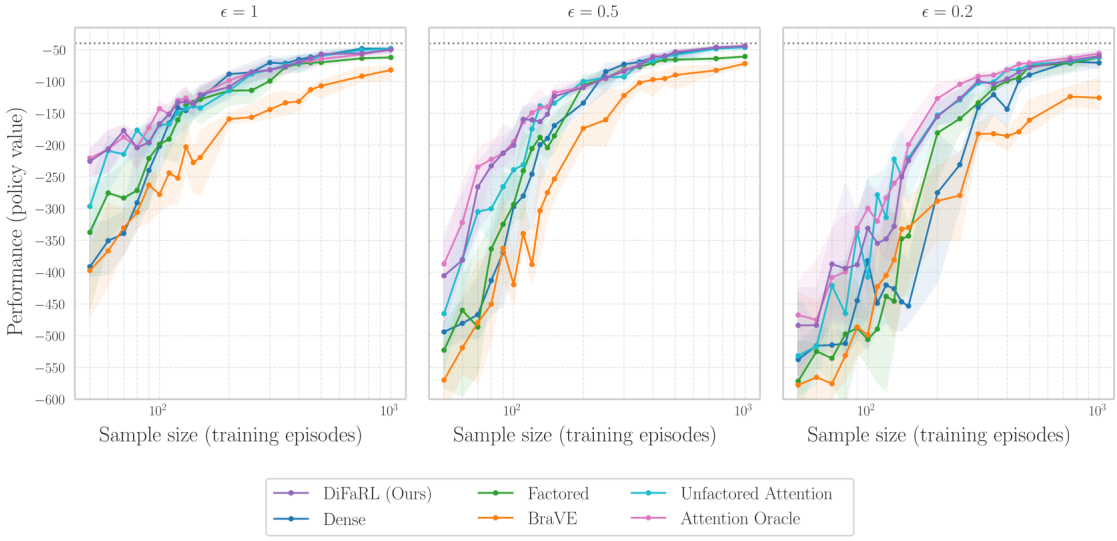


Figure 9: Blood pressure simulation results for three values of epsilon. As the value of ϵ decreases, the behavioral policy becomes closer to deterministic. It therefore becomes more difficult to avoid false positives, making our model more similar to the baseline attention model.

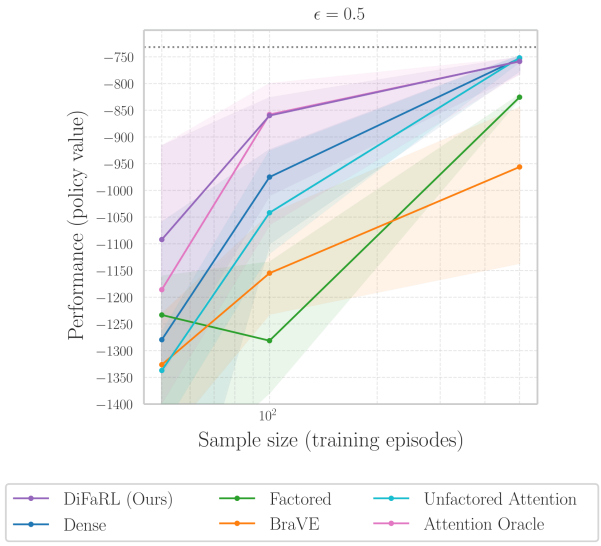


Figure 10: Modified blood pressure simulation with a non-invertible mapping from the latent $\mathbf{Z}$ to the observed $\mathbf{S}$.

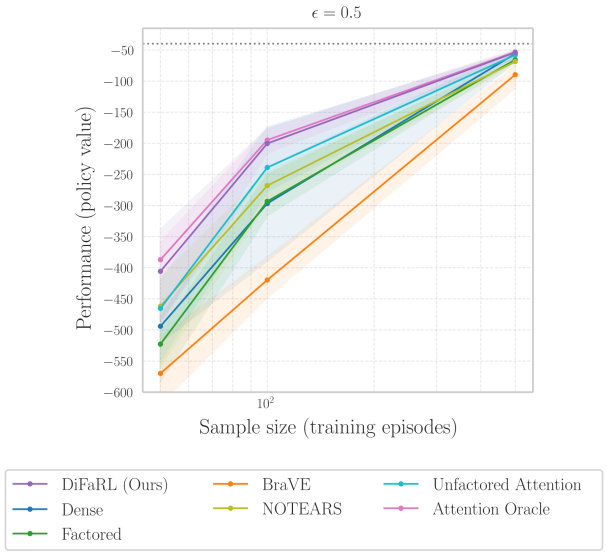


Figure 11: Blood pressure simulation with a NOTEARS based attention implementation.

Appendix F. DAG Variations