Model-free Causal Reinforcement Learning with Causal Diagrams

Junkyu Lee, **Tian Gao**, **Elliot Nelson**, **Miao Liu** and **Debarun Bhattacharjya**
IBM Research
{junkyu.lee,enelson,miao.liu1}@ibm.com, {tgao,debarunb}@us.ibm.com

Abstract

We present a new model-free causal reinforcement learning approach that utilizes the structure of causal diagrams, which could be learned during causal representation learning and causal discovery. Unlike the majority of approaches in causal reinforcement learning that focus on model-based approaches and off-policy evaluations, we explore another direction: online model-free methods. We achieve this by extending a causal sequential decision-making formulation with factored Markov decision process (FMDP) and MDP with unobserved confounders (MDPUC), and by incorporating the concept of action as intervention. The choice of extending MDPUC addresses the issue of bidirectional arcs in learned causal diagrams. The action as intervention idea allows for the incorporation of high-level action models into the action space in an RL environment as a vector of interventions to the causal variables. We also present a value decomposition method and utilize the value decomposition network architecture popular in multi-agent reinforcement learning, showing encouraging preliminary evaluation results.

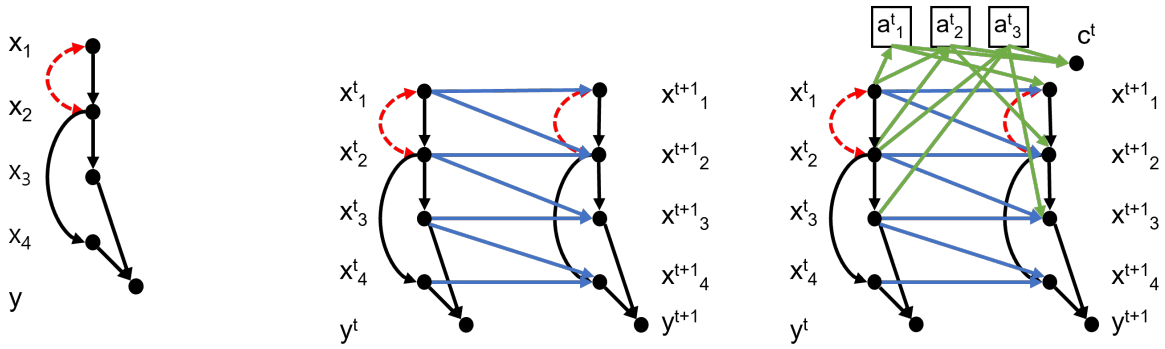
1 Introduction

Deep reinforcement learning (RL) has shown remarkable achievements in various sequential decision making problems, sometimes reaching super-human levels of performance [Mnih *et al.*, 2015; Silver *et al.*, 2017]. Deep RL utilizes powerful neural networks for solving challenging problems such as those involving a high-dimensional feature space or complex non-linear control. Although such methods perform well in simulated environments, it has also been pointed out that methods relying on deep neural networks have significant scope for improvement, particularly with regard to robustness and reusability for wider acceptance and deployment in real-world applications. For instance, causal graphical models [Pearl, 2009; Schölkopf, 2022] have recently emerged as a means of achieving out-of-distribution generalization in sequential decision problems, when leveraged using rapid advances in causal representation learning [Schölkopf *et al.*, 2021].

The burgeoning field of causal reinforcement learning combines ideas from causal inference/discovery and reinforcement learning to tackle several challenges, such as learning world models [Kipf *et al.*, 2020; Locatello *et al.*, 2020; Ke *et al.*, 2019; Brehmer *et al.*, 2022; Zhao *et al.*, 2022], deconfounding the influence of hidden variables [Bareinboim *et al.*, 2015; Zhang and Bareinboim, 2016], off-policy evaluation or imitation learning under missing variables [Buesing *et al.*, 2019; Namkoong *et al.*, 2020; Kumor *et al.*, 2021; Ruan *et al.*, 2023], and improving sample efficiency [Lattimore *et al.*, 2016; Wang *et al.*, 2021; Pitis *et al.*, 2020].

Causal graphical models can offer high-level inductive biases for deep reinforcement learning. In particular, one can view the low-dimensional feature space extracted by causal representation learning as causal state variables and assume discovered causal diagrams may capture the structure of causal mechanisms. When the learned causal diagrams are faithful or ground truth graphs are provided, causal diagrams have been utilized to identify irrelevant state variables by applying graph separation criteria, resulting in improved sample efficiency in model-based reinforcement learning [Wang *et al.*, 2022; Zhang *et al.*, 2019; Huang *et al.*, 2022]. Furthermore, when causal discovery successfully learns not only the structure but also the parameters, applying dynamic programming on a learned transition model directly solves the problem and shows promising results in simple environments [Kipf *et al.*, 2020; Ke *et al.*, 2021].

From a sequential decision-making perspective, it is natural to handle causal variables as state variables, as in a factored state Markov decision process (FMDP) [Boutilier *et al.*, 1999; Guestrin *et al.*, 2003] which uses a dynamic Bayesian network (DBN) [Dean and Kanazawa, 1989; Murphy, 2002] to describe underlying state transitions. Following this perspective, we focus on the usage of causal diagrams for online model-free reinforcement learning agents while assuming that causal diagrams can be learned during the pre-training stages in causal representation learning or causal discovery. Rather than treat an action as a passive label of an unknown state transition function embedded inside the environment, we assume that the agent makes a deliberate choice among available actions and knows the direct effect of performing an action in the factored causal state representation, following the common assumption that the state space is known to the agent. In other words, the agent knows in ad-



(a) Causal Graph (b) Dynamic Structural Causal Model (c) Factored MDP

Figure 1: Example of Graphs with 5 State Variables. Figure 1a shows a causal graph with five variables x_1, x_2, x_3, x_4, and y. The red dotted bi-directed arc indicates that there exists an unobserved confounding variable between x_1 and x_2. Figure 1b shows a dynamic structural causal model between time steps t and $t+1$. The blue arcs represent the influence from the previous time step t, and we assume that causal diagrams over the variables at two time steps have a restricted structure where unobserved confounders do not involve state transitions. However, we allow arcs within the same time steps. Figure 1c shows a popular factored MDP template that encodes equivalent information with three additional action variables a_1^t, a_2^t, and a_3^t. We also added c^t that encodes deterministic constraints over the action variables. The green arcs are introduced to encode the scope of policy functions and auxiliary constraints over the action variables.

vance how any of its actions would intervene in the causal states and replace default causal mechanisms in the environment, although directly modifying state transitions themselves is out of reach. This allows for not only reducing the size of the action space relative to the number of all possible combinations of changes but also makes an explicit connection to action models in high-level representations, effectively capturing the causal structure in formal or natural languages.

We make the following contributions in this work: (1) We extend a causal sequential decision-making paradigm – FMDP and MDP with unobserved confounders (MDPUC) [Zhang and Bareinboim, 2022] – to accommodate the concept of action as intervention. The choice of extending MDPUC addresses the issue around bi-directed arcs in learned causal diagrams due to imperfectness in graph learning techniques that cannot detect all the causal directions, or causal variable discovery that fails to identify all the necessary causal variables. (2) We propose how formal high-level action models can be mapped to the action space in an RL environment as a vector of interventions to the causal state variables, and demonstrate that such a causal action space, mapped from high-level action models, factors out implicit inference tasks embedded in the environment, such as determining whether the agent action is executable in the environment by checking the precondition of the action. (3) We show how to decompose the value function of factored state MDPUC, one per intervention variable, under the action as intervention idea, and utilize a value decomposition network architecture [Sunehag *et al.*, 2018] for reducing the size of the action space, with encouraging preliminary evaluation results.

2 Background

2.1 Causal Graphical Models

In this section, we introduce notation and review basic definitions of the causal graphical model framework [Pearl, 2009]. We use an uppercase letter to denote a random variable (X), a lowercase letter to denote a value ($X = x$) from its domain $\text{dom}(X)$, and a set of variables in boldface ($\mathbf{X}$) and a set of value assignments is also in lowercase boldface $\mathbf{x}$. The cardinality of a set $\mathbf{X}$ is denoted by $|\mathbf{X}|$, and the scope of a function f is denoted by $sc(f)$.

Structural Causal Model

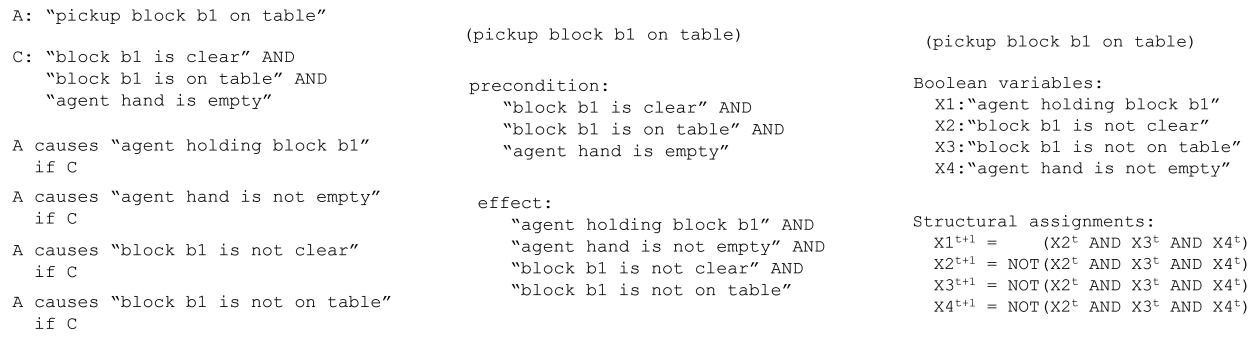
A *structural causal model* (SCM) $\mathcal{M}$ is a tuple $\langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P \rangle$, where $\mathbf{V}$ is a set of endogenous variables, $\mathbf{U}$ is a set of exogenous variables, $\mathcal{F}$ is a set of functions or structural assignments $\{f_V \mid V \in \mathbf{V}\}$ such that $V \leftarrow f_V(\mathbf{PA}_V, \mathbf{U}_V)$ where $\mathbf{PA}_V \subseteq \mathbf{V}$ and $\mathbf{U}_V \subseteq \mathbf{U}$, and $P(\mathbf{U})$ is a joint probability function for the mutually independent exogenous variables. The exogenous variables $\mathbf{U}$ are also called hidden variables or background variables that are not accessible to the agent. Then, an SCM induces a joint distribution $P(\mathbf{V})$ over endogenous variables $\mathbf{V}$ as

$$P(\mathbf{V}) = \sum_{U \in \mathbf{U}} \Big[\prod_{V \in \mathbf{V}} P(V \mid \mathbf{PA}_V, \mathbf{U}_V) \Big] \cdot P(\mathbf{U}). \quad (1)$$

A *causal diagram* $\mathcal{G}$ associated with $\mathcal{M}$ is a directed acyclic graph (DAG) defined over the nodes associated with variables $\mathbf{V}$ and $\mathbf{U}$. For a node X in $\mathcal{G}$, $pa(X)$ and $an(X)$ denotes a set of parent nodes and a set of ancestor nodes, respectively. For a set of nodes $\mathbf{X}$, we take the union of sets associated with individual variables, for example, $pa(\mathbf{X}) = \cup_{X \in \mathbf{X}} pa(X)$. For each $f_V \in \mathcal{F}$, we introduce directed edges from the scope of f_V to V, namely, $pa(V) = \mathbf{PA}_V \cup \mathbf{U}_V$. Given $\mathcal{G}$, a path from X to Y is a sequence of edges without revisiting the same node. We say two sets of nodes $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated* by $\mathbf{Z}$ in $\mathcal{G}$ if every path from nodes in $\mathbf{X}$ to nodes $\mathbf{Y}$ is blocked by nodes in $\mathbf{Z}$, denoted by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_{\mathcal{G}}$.

Interventions in Structural Causal Model

A causal graphical model is a collection of modular and independent mechanisms f_V, and we may intervene on a variable



(a) Action Language $\mathcal{A}$ (b) STRIPS Action Operator (c) Structural Assignments

Figure 2: Examples of High-Level Action Description. Three examples show the description of high-level action *pickup block b1 on table*. Figure 2a shows an informal action expression in action language $\mathcal{A}$, Figure 2b shows a STRIPS operator encoding the same information, and Figure 2c shows a collection of structural assignments equivalent to the previous action descriptions.

$X \in \mathbf{V}$ such that the mechanism f_X follows a different one denoted by a *regime indicator* σ_X that specifies the change of the conditional probability $P\big(X \mid pa(X); \sigma_X\big)$ [Didelez *et al.*, 2006]. An *atomic intervention* fixes the value of X to x by using the do-operator, and a *conditional intervention* fixes the value of X according to the previously observed values of $\mathbf{Z}$,

$$P\big(X \,|\, pa(X); \sigma_X = do(X = g(\mathbf{Z}))\big) = \mathbb{I}\big(X = g(\mathbf{Z})\big), \quad (2)$$

where $\mathbb{I}$ is the indicator function returning 1 if the argument evaluates true and 0 otherwise. We define a *strategy* $\boldsymbol{\sigma}_{\mathbf{X}}$ as a set of regime indicators $\{\sigma_X \mid X \in \mathbf{X}\}$. We will often abuse the notation of the do-operator such that $do(\mathbf{X} = \boldsymbol{\sigma}_{\mathbf{X}})$ denotes the intervention following a strategy $\boldsymbol{\sigma}_{\mathbf{X}}$. We also use a subscript notation for interventions, $P_{\sigma_X}\big(X \mid pa(X)\big)$ for $P\big(X \mid pa(X); do(X = \sigma_X)\big)$, which can be written by

$$P_{\boldsymbol{\sigma}_{\mathbf{X}}}(\mathbf{V}) = \sum_{\mathbf{U}} \Big[\prod_{V \in \mathbf{V} \setminus \mathbf{X}} P\big(V \mid pa(V)\big) \Big] \cdot \Big[\prod_{X \in \mathbf{X}} P_{\sigma_X}\big(X \mid pa(X)\big) \Big] \cdot P(\mathbf{U}). \quad (3)$$

Identifiability of Causal Effects

For any arbitrary DAG $\mathcal{G}$ associated with an SCM $\mathcal{M}$, we can reduce the graph topology of $\mathcal{G}$ via *latent projection* such that exogenous variables $U \in \mathbf{U}$ introduce a bi-directed edge between endogenous variables $V_i, V_j \in \mathbf{V}$ if the direction toward V_i and V_j are opposite and there is no converging arrows between an exogenous node along the path [Pearl and Verma, 1995]. Figures 1a and 1b show such reduced diagrams. In the presence of bi-directed edges in $\mathcal{G}$, $\mathbf{V}$ can be partitioned into sets, called *c-components*, each c-component is a set of nodes that are connected only through bi-directed edges [Tian and Pearl, 2002]. We also define *c-factor* as a collection of functions in the same c-component. Then, $P(\mathbf{V})$ can be factorized over the *c-factors* denoted by $Q\big[\mathbf{S}\big]$ as

$$P(\mathbf{V}) = \prod_{V_i \in \mathbf{S}_0} P\big(V_i | pa(V_i)\big) \prod_{i \in \{1..k\}} Q\big[\mathbf{S}_i\big](\mathbf{V}),$$
$$Q\big[\mathbf{S}_i\big](\mathbf{V}) = \prod_{V \in \mathbf{S}_i} P(V | pa(V)) \sum_{U \in An^{\mathbf{U}}(\mathbf{S}_i)} P(U | pa(U)), \quad (4)$$

where a sequence of numbers from 1 to K is denoted by $\{1..K\}$, $\mathbf{S}_i$ is a c-component, $Q\big[\mathbf{S}_i\big]$ is a c-factor, $An^{\mathbf{U}}(\mathbf{S}_j)$ is a set of exogenous ancestor nodes $\big(\mathbf{U} \cap An(\mathbf{S}_j)\big)$. Finally, a *causal effect* on outcome variables $\mathbf{Y} \subset \mathbf{V}$ subject to an intervention $\mathbf{X} \subset \mathbf{V}$ under condition variables $\mathbf{Z}$ can be written as $P_{\mathbf{X}}\big(\mathbf{Y} \mid \mathbf{Z}\big)$. Informally, we say a causal effect is *identifiable* if we can express the causal effect only in terms of endogenous variables $\mathbf{V}$. The importance of c-components in causal graphical models is due to the fact that the identifiable causal effects can be expressed by c-factors using complete identification algorithms [Tian and Pearl, 2002; Shpitser and Pearl, 2006; Huang and Valtorta, 2006; Tian, 2008].

2.2 Action Models

High-Level Action Models

In a real-world environment, especially in a multi-task RL setting, an agent may have a class of tasks that share the same state space but have varying action spaces, or it could acquire or lose skills along the way while solving tasks. Such multi-task action spaces show more complex action structures than flat action labels, and if we ignore inherent structures in the action space, the number of action labels may grow combinatorially due to the compositional nature; this is commonly observed in popular robotics domains or text-based interactive environments, where an action can be composed of more elementary actions or elements [Kootbally *et al.*, 2015; Correa and Bareinboim, 2020]. Note that we could still define state transition probability per action labels as done in usual MDP formulations, but incorporating causal models handles individual mechanisms in a more explicit manner.

High-level description of such tasks or skills either in natural language or in formal languages often captures the causal

structure, namely the changes in the states as a consequence of applying an action. Figure 2 shows a typical action description of *pickup block on table* in the blocks world domain, where Figure 2a shows a description [1] following the action language $\mathcal{A}$ [Gelfond and Lifschitz, 1998] and Figure 2b shows an equivalent STRIPS action operator [Fikes and Nilsson, 1971].

Structural Assignments from Action Model

Although those action descriptions are not expressed in terms of structural assignments in causal graphical models, it is not difficult to see that action in formal action description languages can be translated into structural assignments. In action language $\mathcal{A}$, action A is a collection of expressions in the form "A causes X if Z", where X is an effect literal and Z is a conjunction literal for the precondition [2]. Given an action A in action language $\mathcal{A}$, we collect effect literals $\mathbf{X}$ and a precondition expression Z. Then, structural assignments $\boldsymbol{\sigma}_{(A,\mathbf{X})}$ can be written as a collection of conditional interventions, $\sigma_{(A,X_i)} = do\big(X_i = \pi_{X_i}(\mathbf{Z})\big)$, where $X_i \in \mathbf{X}$ is a Boolean variable associated with effect literals ($\text{eff}(\tilde{A})$), $\mathbf{Z}$ is a set of Boolean variables associated with a precondition expression ($\text{pre}(\tilde{A})$) for an effect literal X_i, and π_{X_i} is a Boolean function. Although we only show the simplest translations, other expressions in richer action languages [Gelfond and Lifschitz, 1998], or action operators in planning domain definition languages [Fikes and Nilsson, 1971; Pednault, 1989; McDermott *et al.*, 1998] can also be translated into structural assignments in a similar manner [3].

Note that variable binding between a structural causal model and a high-level formal action model was performed in a syntactic manner. When such high-level action description is given in informal natural language texts, we may extract and bind causal variables and statements using natural language techniques, which could be a potential direction for future work around causal action representation learning.

3 Causal Factored Markov Decision Processes

In this section, we extend a factored state Markov decision process (FMDP) following prior work on a structural causal model framework in the presence of unobserved confounders [Zhang and Bareinboim, 2016; Zhang and Bareinboim, 2022].

3.1 Dynamic Structural Causal Models

Similar to dynamic Bayesian networks (DBN) [Dean and Kanazawa, 1989; Murphy, 2002] and (dynamic) influence diagrams [Howard and Matheson, 2005] that replicate a graphical model over unrolled time steps, we define a dynamic structural causal model (DSCM) as a pair of SCMs $\langle \mathcal{M}^0, \mathcal{M}^{\rightarrow} \rangle$, where $\mathcal{M}^0$ is an SCM inducing the probability distribution at time $t = 0$ and $\mathcal{M}^{\rightarrow}$ induces the transition probability from time step t to $t+1$. $\mathcal{M}^0$ and $\mathcal{M}^{\rightarrow}$ share the same set of variables $\mathbf{V}$ and $\mathbf{U}$, and they are unrolled over time steps, $\mathbf{V}^t$ and $\mathbf{U}^t$.

It is known that an arbitrary choice of the transition probability $P(\mathbf{V}^{t+1}, \mathbf{U}^{t+1} \mid \mathbf{V}^t, \mathbf{U}^t)$ may result in a *c-component* that spreads over all time steps [Zhang and Bareinboim, 2016; Srinivasan *et al.*, 2021; Bruns-Smith, 2021]. Following the standard assumptions in MDPs with unobserved confounders (MDPUC) [Zhang and Bareinboim, 2016; Zhang and Bareinboim, 2022], we restrict the causal structure in DSCM such that exogenous variables can only influence variables within the same time step, and transition probabilities do not depend on exogenous variables. Namely,

$$P(\mathbf{V}^{t+1}, \mathbf{U}^{t+1} \mid \mathbf{V}^t, \mathbf{U}^t) = P(\mathbf{V}^{t+1} \mid \mathbf{V}^t, \mathbf{U}^{t+1}) P(\mathbf{U}^{t+1}),$$
$$P(\mathbf{V}^0, \mathbf{U}^0) = P(\mathbf{V}^0 \mid \mathbf{U}^0) \cdot P(\mathbf{U}^0). \tag{5}$$

3.2 Causal FMDP with Unobserved Confounders

Next, we define a factored state MDPUC with unobserved confounders (FMDPUC) as a tuple $\langle \mathcal{S}, \mathcal{A}, P^0, P^{\rightarrow}, \mathcal{R}, \gamma \rangle$ relative to a DSCM $\langle \mathcal{M}^0, \mathcal{M}^{\rightarrow} \rangle$. The state space $\mathcal{S}$ is the product space of all variables, $(\times_{V_i \in \mathbf{V}} V_i) \times (\times_{U_i \in \mathbf{U}} U_i)$, and the action space $\mathcal{A}$ is the product space of all intervention variables $\mathbf{X} \subseteq \mathbf{V}$, $(\times_{X_i \in \mathbf{X}} X_i)$.

Before defining the state transition probability function $P^{\rightarrow}$, let us consider a strategy $\boldsymbol{\sigma}_{\mathbf{X}^{t+1}}$ in $\mathcal{M}^{\rightarrow}$ and its associated causal diagram $\mathcal{G}^{\rightarrow}$. For a set of intervention variables $\mathbf{X}^{t+1}$ at time $t+1$, a partial order $\mathcal{O}^{\rightarrow}$ over the variables in $\mathcal{M}^{\rightarrow}$ consistent with the topological order in $\mathcal{G}^{\rightarrow}$ can be written as

$$\mathcal{O}^{\rightarrow} := \mathbf{Z}_0^{t+1} \prec X_1^{t+1} \prec \cdots \prec \mathbf{Z}_{K-1}^{t+1} \prec X_K^{t+1} \prec \mathbf{Z}_K^{t+1}, \tag{6}$$

where $\mathbf{V}^t \subseteq \mathbf{Z}_0^{t+1}$ and $\mathbf{Z}_i^{t+1} \subseteq \big(\mathbf{V}^t \cup \mathbf{V}^{t+1}\big) \backslash \mathbf{X}^{t+1}$. Given an intervention strategy $\boldsymbol{\sigma}_{\mathbf{X}^{t+1}}$ comprised of a sequence of conditional interventions $\big(\sigma_{X_1^{t+1}}, \sigma_{X_2^{t+1}}, \ldots, \sigma_{X_K^{t+1}}\big)$, which also follows the topological order $\mathcal{O}^{\rightarrow}$, we define $\sigma_{X_i^{t+1}}$ as a function over the variables that precede X_i^{t+1} in $\mathcal{O}^{\rightarrow}$,

$$\sigma_{X_i^{t+1}} = \pi_{X_i^{t+1}}\big(\mathbf{Z}_0^{t+1} \cup \big(\bigcup_{j=1}^{i-1} \mathbf{Z}_j^{t+1} \cup \mathbf{X}_j^{t+1}\big)\big). \tag{7}$$

We define a policy of FMDPUC as a collection of functions associated with $\boldsymbol{\sigma}_{\mathbf{X}^{t+1}}$, $\boldsymbol{\pi}_{\mathbf{X}^{t+1}} = \{\pi_{X_i^{t+1}} \mid X_i^{t+1} \in \mathbf{X}^{t+1}\}$, and denote the set of all possible policies by Π. We define *policy scope* [Lee and Bareinboim, 2020] of $\boldsymbol{\pi}_{\mathbf{X}^{t+1}}$ or $\boldsymbol{\sigma}_{\mathbf{X}^{t+1}}$ by $\mathcal{SC}_{\boldsymbol{\pi}_{\mathbf{X}^{t+1}}} = \{(X_i^{t+1}, sc(\pi_{X_i^{t+1}})) \mid X_i^{t+1} \in \mathbf{X}^{t+1}\}$, and denote a set of intervention variables and a set of condition variables by $\mathcal{X}_{\boldsymbol{\pi}_{\mathbf{X}^{t+1}}} = \mathbf{X}^{t+1}$, and $\mathcal{C}_{\boldsymbol{\pi}_{\mathbf{X}^{t+1}}} = \bigcup_{X_i^{t+1} \in \mathbf{X}^{t+1}} sc(\pi_{X_i^{t+1}})$. Note that not every realization of the intervention strategy is feasible in the actual environment, as there could be constraints imposed on the combinations of intervention variables, or a certain condition in some states may prevent changing the value of some intervention variables. In practice, such invalid interventions are often handled implicitly by an environment as invalid actions, resulting

[1] We show informal textual descriptions and skip defining all logical foundations behind formal languages, hoping that this example is intuitive enough to understand the basic idea.

[2] In propositional logic, a literal is either a positive or negative atom, where an atom is a single proposition. Atoms can be viewed as Boolean random variables.

[3] A translation from a STRIPS operator to structural assignments is provided in [Pearl, 2009].

in no change in the environment. Since we define FMDPUCs through DSCM and actions are interventions on the environment states, which is not merely an action label that indirectly relates to some underlying intervention, it is more natural for the agent to restrict the action space such that it only defines feasible interventions[4].

In this paper, we focus on stationary policies, and therefore we drop superscripts indicating time steps in $\boldsymbol{\sigma}_{\mathbf{X}^{t+1}}$ and $\boldsymbol{\pi}_{\mathbf{X}^{t+1}}$ if it is clear from the context. The state transition probability $P^{\rightarrow}$ is induced by the interventional distribution subject to $\boldsymbol{\pi}_{\mathbf{X}}$ in $\mathcal{M}^{\rightarrow}$,

$$P^{\rightarrow}_{\mathbf{X}^{t+1}}(\mathbf{V}^{t+1}, \mathbf{U}^{t+1} \mid \mathbf{V}^t, \mathbf{U}^t) = \prod_{V^{t+1} \notin \mathbf{X}^{t+1}} P\big(V^{t+1} \mid pa(V^{t+1})\big) \prod_{X_i^{t+1} \in \mathbf{X}^{t+1}} \pi_{X_i}\big(X_i^{t+1} \mid sc(\pi_{X_i^{t+1}})\big) \prod_{U^{t+1} \in \mathbf{U}^{t+1}} P\big(U^{t+1}\big), \tag{8}$$

where $pa(V^t) \subseteq \mathbf{V}^t \cup \mathbf{V}^{t+1} \cup \mathbf{U}^{t+1}$, and $sc(\pi_{X_i}) \subseteq \mathbf{V}^t \cup \mathbf{V}^{t+1}$. P^0 is an initial state distribution induced by $\mathcal{M}^0$,

$$P^0(\mathbf{V}^0, \mathbf{U}^0) = \prod_{V^0 \in \mathbf{V}^0} P\big(V^0 \mid pa(V^0)\big) \cdot \prod_{U^0 \in \mathbf{U}^0} P\big(U^0\big), \tag{9}$$

where $pa(V^0) \subseteq \mathbf{V}^0 \cup \mathbf{U}^0$. $\mathcal{R}$ is a set of reward functions defined over the endogenous variables $\mathbf{V}$,

$$\mathcal{R} = \big\{R_i^{t+1}(\mathbf{V}_i^t, \mathbf{X}_i^{t+1}) \,|\, \mathbf{V}_i^t \subseteq \mathbf{V}^t, \mathbf{X}_i^{t+1} \subseteq \mathbf{X}^{t+1}, i \in \{1..|\mathcal{R}|\}\big\}, \tag{10}$$

and γ is a discount factor between 0 and 1.

3.3 Dynamic Programming with Action Models

Next, we generalize the actions in FMDPUC with the intervention strategies derived from high-level action description models. Given a high-level action model, each high-level action A in a collection $\mathcal{A}$ can be translated into an intervention strategy $\boldsymbol{\sigma}_{(A,\mathbf{X}_A)}$, where $\mathbf{X}_A$ denotes the effect variables (eff(A)) of the high-level action A. Then, decisions made by an agent at each time involve selecting a high-level action A in each state. A decision rule Δ is a mapping from endogenous variables $\mathbf{V}$ to the high-level action space $\mathcal{A}$, where each action A translates into a strategy $\boldsymbol{\sigma}_{(A,\mathbf{X}_A)}$ that intervenes on $\mathbf{X}_A$, and we denote a space of all decision rules by $\boldsymbol{\Delta}$. Recall that in the previous FMDPUC formulation, we derived a policy $\boldsymbol{\pi}_{\mathbf{X}}$ from a single strategy $\boldsymbol{\sigma}_{\mathbf{X}}$. With an action model, each action A induces $\boldsymbol{\sigma}_{(A,\mathbf{X}_A)}$ associated with a set of deterministic functions $\boldsymbol{\pi}_{(A,\mathbf{X}_A)} = \{\pi_{(A,X)} \mid X \in \mathbf{X}_A\}$ that intervene on effect variables $\mathbf{X}_A$. In this setting, an agent maximizes the discounted sum of rewards,

$$\max_{\Delta \in \boldsymbol{\Delta}} \sum_{\mathbf{V}^{\{0..\infty\}}} \sum_{\mathbf{U}^{\{0..\infty\}}} \Big[\prod_{t=0}^{\infty} P_{\boldsymbol{\sigma}_\Delta}\big(\mathbf{V}^{t+1} | \mathbf{V}^t, \mathbf{U}^{t+1}\big) P\big(\mathbf{U}^{t+1}\big)\Big] \cdot \Big[P\big(\mathbf{V}^0, \mathbf{U}^0\big)\Big] \cdot \Big[\sum_{t=0}^{\infty} \gamma^t R^t(\mathbf{V}^t, \mathbf{X}^{t+1})\Big], \tag{11}$$

[4]Action spaces in factored MDP [Boutilier *et al.*, 1999; Guestrin *et al.*, 2003] as potentially represented by influence diagrams [Howard and Matheson, 2005] are also defined by introducing factored action variables; in this work, these are intervention variables in a causal graphical model.

where we abused notations: $R^t(\mathbf{V}^t, \mathbf{X}^{t+1})$ is the total sum of local rewards over local scopes $\mathbf{V}_i^t \subseteq \mathbf{V}^t$ and $\mathbf{X}_i^{t+1} \subseteq \mathbf{X}^{t+1}$, namely, $\sum_{i=1}^{|\mathcal{R}|} R_i^t(\mathbf{V}_i^t, \mathbf{X}_i^{t+1})$, and $\boldsymbol{\sigma}_\Delta$ is an intervention strategy determined by Δ, namely, $\boldsymbol{\sigma}_{\Delta(\mathbf{V}^t)}$. $\mathcal{X}_{\boldsymbol{\pi}_\Delta}$ denotes the set of intervention variables subject to Δ applied in the current state, $\mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{V}^t)}} \subseteq \mathbf{X}^{t+1}$. Note that the state transition probability remains Markovian regardless of the presence of unobserved variables $\mathbf{U}^{t+1}$, and the value function of a stationary decision rule Δ can be written by

$$\mathcal{V}_\Delta(\mathbf{v}) = \mathbb{E}_{\boldsymbol{\sigma}_\Delta}\Big[\sum_{h=0}^{\infty} \gamma^h R\big(\mathbf{V}^{h+t}, \mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{V}^{h+t})}}\big) | \mathbf{V}^t = \mathbf{v}\Big] = \sum_{\mathbf{V}^{t+1}, \mathbf{U}^{t+1}} P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{v})}}}\big(\mathbf{V}^{t+1}, \mathbf{U}^{t+1} | \mathbf{v}\big)[R(\mathbf{v}, \mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{v})}}) + \gamma \mathcal{V}_\Delta(\mathbf{V}^{t+1}))], \tag{12}$$

and the Q-function can also be written by

$$\mathcal{Q}_\Delta(\mathbf{v}, A) = \sum_{\mathbf{V}^{t+1}, \mathbf{U}^{t+1}} P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}\big(\mathbf{V}^{t+1}, \mathbf{U}^{t+1} | \mathbf{v}\big) \cdot [R(\mathbf{v}, \mathcal{X}_{\boldsymbol{\pi}_A}) + \gamma \mathcal{V}_\Delta(\mathbf{V}^{t+1}))]. \tag{13}$$

Then, we can write the Dynamic Programming operator $\mathcal{T}_{\boldsymbol{\Delta}}$ for a stationary decision rule $\boldsymbol{\Delta}$ by

$$\mathcal{T}_{\boldsymbol{\Delta}} \mathcal{V}(\mathbf{v}) = \sum_{\mathbf{V}', \mathbf{U}'} P_{\boldsymbol{\sigma}_\Delta}\big(\mathbf{V}', \mathbf{U}' | \mathbf{v}\big)\big[R\big(\mathbf{v}, \mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{v})}}\big) + \mathcal{V}(\mathbf{V}')\big]. \tag{14}$$

The above equation shows that evaluating decision rule Δ depends on the identification of the causal effects on the reward outcomes. To see this, we introduce an outcome variable Y that takes all the values of the function $\mathbf{R}(\mathbf{v}, \mathcal{X}_{\boldsymbol{\pi}_{\Delta(\mathbf{v})}}) + \mathcal{V}(\mathbf{V}')$ as its domain, and rewrite conditional expectations by

$$\mathbb{E}_{\boldsymbol{\sigma}_\Delta}[Y \,|\, \mathbf{V} = \mathbf{v}] = \sum_{Y} Y \sum_{\mathbf{V}'} \sum_{\mathbf{U}'} \big[P_{\boldsymbol{\sigma}_\Delta}\big(Y, \mathbf{V}', \mathbf{U}' \,|\, \mathbf{V} = \mathbf{v}\big)\big]. \tag{15}$$

$\mathbb{E}_{\boldsymbol{\sigma}_\Delta}[Y | \mathbf{V} = \mathbf{v}]$ can be estimated without confounding bias if and only if the causal effect $P_{\boldsymbol{\sigma}_\Delta}(Y | \mathbf{V} = \mathbf{v})$ is identifiable.

In Figure 1b, an agent could apply conditional interventions on variables X_1^{t+1}, X_2^{t+1}, and X_3^{t+1} by conditioning on the parent variables drawn with blue edges. We can check that the conditional interventions on a single variable X_1 or two variables (X_1, X_2) are not identifiable. When all the causal effects of actions are identifiable, we could solve FMDPUC as a dynamic influence diagram if model parameters are known. In typical model-based reinforcement learning approaches, we learn identifiable model parameters to optimize the policy. In the following part, we introduce inductive bias for a model-free reinforcement learning agent to leverage the causal structure of the environment and the action model.

4 Decomposing Value Functions

In this section, we show a relaxation scheme that decomposes the value function defined over the joint of all variables into a collection of functions defined per each variable. Then, we utilize the existing value decomposition network [Sunehag *et al.*, 2018] for implementing multiple online deep

Q-learning (DQN) subagents for solving decomposed FMDPUC. We also show a preliminary result of applying a decomposition scheme.

4.1 Decomposing Value with Lower Bounds

We observe that $\mathcal{Q}_\Delta(\mathbf{v}, A)$ in Eq. (13) can be rewritten as

$$\mathcal{Q}_\Delta(\mathbf{v}, A) = \sum_{\mathbf{V}', \mathbf{U}'} P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \big[R(\mathbf{v}, A) + \gamma \mathcal{V}(\mathbf{V}') \big], \quad (16)$$

and we can rewrite $P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}(\mathbf{V}', \mathbf{U}' \mid \mathbf{v})$ by

$$P(\mathbf{U}') \prod_{V'_i \in \mathbf{V}'} P\big(V'_i | pa(V'_i)\big) q(V'_i, pa(V'_i)), \quad (17)$$

with each $q(V'_i, pa(V'_i))$ can be expressed by

$$q(V'_i, pa(V'_i)) = \left[\frac{\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big)}{P\big(V'_i \mid pa(V'_i)\big)} \right]^{\mathbb{I}\left(V'_i \in \text{eff}(A)\right)}, \quad (18)$$

where $\mathbb{I}\big(V'_i \in \text{eff}(A)\big)$ denotes an indicator function that returns 1 if V'_i is in the effect variables of action A ($\text{eff}(A)$) and 0 otherwise, and $\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big)$ denotes an indicator function for fixing the value of V'_i to the outcome of action A depending on its precondition values in $\mathbf{v}$. Note that $\mathcal{Q}_\Delta(\mathbf{v}, A)$ is an action-value function defined over the intervention distribution $P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}(\mathbf{V}', \mathbf{U}' | \mathbf{v})$ subject to action A.

Next, we will decompose $\mathcal{Q}_\Delta(\mathbf{v}, A)$ as a weighted sum of action value functions $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ subject to a single variable intervention V'_j for all $V'_j \in \mathbf{V}'$. Observing that the indicator functions in $q\big(V'_i, pa(V'_i)\big)$ can be extended by

$$\begin{aligned} &\big[\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big)\big]^{\mathbb{I}\left(V'_i \in \text{eff}(A)\right)} \\ =&\big[\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big)\big]^{\mathbb{I}\left(V'_i \in \text{eff}(A)\right) + \mathbb{I}\left(V'_i \in \text{eff}(A) \cap \{V'_j\}\right)}, \end{aligned} \quad (19)$$

we can find a lower bound of $P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}(\mathbf{V}', \mathbf{U}' \,|\, \mathbf{v})$ in terms of $P^{\rightarrow}_{V'_j}(\mathbf{V}', \mathbf{U}' | \mathbf{v})$ as

$$\begin{aligned} P^{\rightarrow}_{\mathcal{X}_{\boldsymbol{\pi}_A}}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \geqslant &\Big[P^{\rightarrow}_{V'_j}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \mathbb{I}(V'_j \in \text{eff}(A)) \\ &+ P^{\rightarrow}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \mathbb{I}(V'_j \notin \text{eff}(A)) \Big] \\ &\cdot \prod_{V'_i \in \mathbf{V}'} \Big[\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big) \Big]^{\mathbb{I}\left(V'_i \in \text{eff}(A)\right)}. \end{aligned} \quad (20)$$

Introducing a positive weight w_j per V'_j that sums to 1 over all variables in $\mathbf{V}'$, the global Q-function $\mathcal{Q}_\Delta(\mathbf{v}, A)$ can be bounded below by the following weighted sum of the local Q-functions $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ over all $V'_j \in \mathbf{V}'$,

$$\begin{aligned} \mathcal{Q}_\Delta(\mathbf{v}, A) \geqslant &\sum_{j=1}^{|\mathbf{V}'|} w_j \mathcal{Q}_\Delta(\mathbf{v}, A; V'_j) \\ &\cdot \prod_{V'_i \in \mathbf{V}'} \Big[\mathbb{I}\big(V'_i = \pi_{(A,V'_i)}(\mathbf{v})\big) \Big]^{\mathbb{I}\left(V'_i \in \text{eff}(A)\right)}, \end{aligned} \quad (21)$$

where $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ is

$$\begin{aligned} \sum_{\mathbf{V}', \mathbf{U}'} \Big[&P^{\rightarrow}_{V'_j}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \cdot \mathbb{I}(V'_j \in \text{eff}(A)) + P^{\rightarrow}(\mathbf{V}', \mathbf{U}' | \mathbf{v}) \cdot \\ &\mathbb{I}(V'_j \notin \text{eff}(A)) \Big] \cdot \big[R(\mathbf{v}, A) + \gamma \mathcal{V}(\mathbf{V}') \big]. \end{aligned} \quad (22)$$

Indicator functions ensure the consistency of the value of $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ with respect to the interventions subject to action A. In Eq. (21), if any value assignment to V'_i is not consistent with the effect of the action, the overall value becomes zero. In Eq. (22), indicator functions switch the probability between the intervention distribution and the observation distribution depending whether V'_j is an effect variable of A or not. Namely, $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ computes the value using the intervention distribution $P^{\rightarrow}_{V'_j}(\mathbf{V}' \mid \mathbf{v})$ if V'_j is in the effect variables of action A. Otherwise, $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$ uses the observation distribution $P^{\rightarrow}(\mathbf{V}' \mid \mathbf{v})$ and When the intervention distribution $P^{\rightarrow}_{V'_j}(\mathbf{V}' \mid \mathbf{v})$ is not identifiable, we may generate lower bounds in Eq. (20) using an alternative set of local Q-functions that are defined only over identifiable intervention distributions.

4.2 Value Decomposition Network

In multi-agent reinforcement learning, one popular approach is training each agent in a centralized manner and executing individual agents independently. Although our problem is a single agent FMDPUC, we can reformulate the problem as a decentralized MDP problem based on Eq. (21). Namely, we learn individual local Q-functions, $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_j)$, in a centralized manner by maximizing the lower bound by utilizing the value decomposition network (VDN) architecture [Sunehag *et al.*, 2018]. Figure 3 shows a value decomposition net-

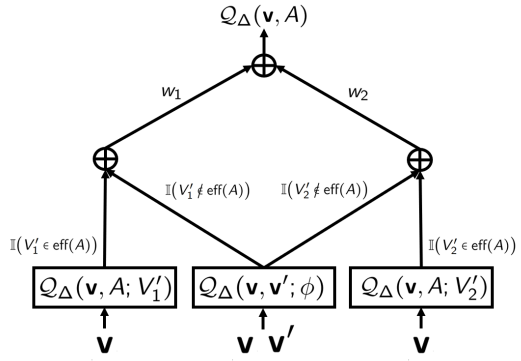


Figure 3: Examples of Value Decomposition Network. The network architecture follows the decomposition shown in Eq. (21) There are two variables V_1 and V_2, where $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_1)$ corresponds to the DQN subagent associated with the local value function subject to the intervention on V'_1 and $\mathcal{Q}_\Delta(\mathbf{v}, A; V'_1)$ corresponds to the subagent associated with the intervention on V'_2. The last subagent $\mathcal{Q}_\Delta(\mathbf{v}, \mathbf{v}'; \phi)$ is associated with the local value function defined over the observational distribution in the absence of intervention variables, and therefore, it learns the value given endogenous state variables $\mathbf{v}$ and $\mathbf{v}'$ in two consecutive time steps.

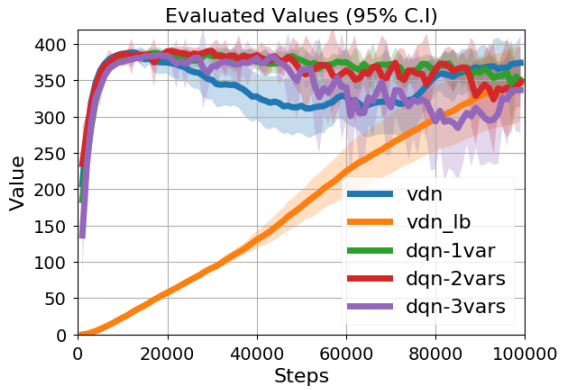


Figure 4: Comparison of Discounted Sum of Returns. The x-axis is the total number of training steps, and the y-axis is the discounted accumulated rewards. In the experiment, the discounting factor, the maximum episode length, and the highest accumulated discounted return was 0.99, 1000, and 400, respectively. The blue curve corresponds to VDN with the action space with at most 1 intervention variable. The orange curve `vdn-lb` shows the lower bounds shown in Eq. (21). The following three curves are associated with DQNs, varying the maximum number of intervention variables from 1 to 3.

work in a small example with two endogenous variables V_1 and V_2.

In model-free causal reinforcement learning, there are mainly two different aspects compared with multi-agent reinforcement learning. The first difference is that FMDPUC is not a partially observable environment, and therefore all subagents share the observed state variables $\mathbf{v}$. In Figure 3, we see that all subagents receive $\mathbf{v}$ as input to the local DQN subagent, and the subagent $\mathcal{Q}_\Delta(\mathbf{v}, \mathbf{v}'; \phi)$ receives additional state variables $\mathbf{v}'$. In the absence of a dynamics model for predicting $\mathbf{v}'$, we generate the next state $\mathbf{v}'$ using the action model by applying action A in $\mathbf{v}$. The second difference is the action space. The action space of FMDPUC is defined relative to the high-level action model $\mathcal{A}$, and we cannot execute arbitrary combinations of the individual decisions made by each subagent as done in the multi-agent setting. Such a restriction is, in part, already reflected in the decomposition lower bound by indicator functions in Eq. (21). To select the best action given local Q-functions during the training or testing phase, there are two possible approaches: either we iterate over all applicable actions in $\mathcal{A}$ and evaluate the value for each action A and select the best one, or we project the intervention vector collected from subagents to a set of valid actions that are consistent with the intervention strategy and achieve a higher value. When we implement a model-free reinforcement learning agent using DQN, this choice of utilizing VDN greatly reduces the action space since the original DQN agent must enumerate all possible combinations of interventions. In the decomposed architecture, the action space remains bounded per individual subagent.

4.3 Preliminary Results

We evaluated a modification of the chemistry environment [Ke *et al.*, 2021] in sprites world [Watters *et al.*, 2019]. We used a causal graph similar to the one shown in Fig 1b, with four objects, each having 2 colors, say black and white. The reward was given by matching the colors of two pairs, (x_1, x_2) and (x_3, x_4). Specifically, if both colors are black, the reward is 2, and if both are white, the reward is 1. Although this is a small toy environment, flattening the action space results in a total of 9, 33, and 65 action labels for the flat DQN agent when we increase the number of objects that the agent can select from 1 to 3 in a single step. However, our approach bounds the number of intervention variables to 1 and fixes the size of the action space to 2 per subagent regardless of such variations in the action space. Figure 4 compares the discounted sum of the rewards from two architectures: (1) DQN with 4 hidden layers, each in dimension 64 by 64, followed by RELU activation function, and (2) VDN with 4 DQNs, each only has 1 hidden layer. We averaged over 10 trials, and we see that VDN shows comparable performance or outperforms DQN. Next, we also observe that VDN trained on the action space that only allows changing at most one variable at each step can be transferred to other environments in the out-of-distribution action spaces. For the environment constrained to select precisely two and three variables at each step, the average over 20 episodic discounted accumulated returns is 357.68 and 350.03, respectively. Lastly, the zero-shot transfer performance in the action spaces identical to DQN-2vars and DQN-3vars is 355.24 and 349.0, respectively.

5 Conclusion

We presented a model-free, online, causal reinforcement learning approach, focusing on an extension of Q-learning-based methods. The main idea is to utilize causal diagrams with unobserved confounders, which are often ignored in related prior work, to extend the factored Markov decision process formulation. Unlike the traditional approach of introducing action variables and policy functions into the MDP template, we directly extend the action space as interventions originated from agent action models. Such a mapping from agent actions to interventions could also be learned during causal discovery since causal graph learning requires intervention data from the agent, or we could directly map the high-level action into interventions in the structural causal models, as demonstrated in this paper. One advantage of the presented formulation is that we can decompose the value function of FMDPUC with a set of local value functions using causal diagrams, which may mitigate the confounding bias during policy learning if the local intervention distributions are all identifiable. Additionally, we show problem decomposition by using the lower bounds of the Q-function such that we reformulate a single-agent learning problem into a multi-agent learning formulation. Although it is approximate, this reformulation reduces the action space exponentially, resulting in encouraging preliminary evaluation results compared to a monolithic DQN agent. We tested our approach utilizing the value decomposition network, the most basic multi-agent reinforcement learning architecture. Given the close connection with solving FMDPUC problems with causal diagrams, it could be fruitful for future work to explore more powerful neural architectures that are popular in the multi-agent reinforcement learning literature.

References

[Bareinboim *et al.*, 2015] Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems*, 2015.

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

[Brehmer *et al.*, 2022] Johann Brehmer, Pim De Haan, Phillip Lippe, and Taco S Cohen. Weakly supervised causal representation learning. In *Advances in Neural Information Processing Systems*, 2022.

[Bruns-Smith, 2021] David A Bruns-Smith. Model-free and model-based policy evaluation when causality is uncertain. In *International Conference on Machine Learning*, 2021.

[Buesing *et al.*, 2019] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representation*, 2019.

[Correa and Bareinboim, 2020] Juan Correa and Elias Bareinboim. A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In *Uncertainty in Artificial Intelligence*, 2020.

[Dean and Kanazawa, 1989] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2):142–150, 1989.

[Didelez *et al.*, 2006] Vanessa Didelez, Philip Dawid, and Sara Geneletti. Direct and indirect effects of sequential treatments. In *Uncertainty in Artificial Intelligence*, 2006.

[Fikes and Nilsson, 1971] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[Gelfond and Lifschitz, 1998] Michael Gelfond and Vladimir Lifschitz. *Action languages*. Linköping University Electronic Press, 1998.

[Guestrin *et al.*, 2003] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[Howard and Matheson, 2005] Ronald A. Howard and James E. Matheson. Influence diagrams. *Decision Analysis*, 2(3):127–143, sep 2005.

[Huang and Valtorta, 2006] Yimin Huang and Marco Valtorta. Pearl's calculus of intervention is complete. In *Uncertainty in Artificial Intelligence*, 2006.

[Huang *et al.*, 2022] 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*, 2022.

[Ke *et al.*, 2019] Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

[Ke *et al.*, 2021] Nan Rosemary Ke, Aniket Didolkar, Sarthak Mittal, Anirudh Goyal, Guillaume Lajoie, Stefan Bauer, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Christopher Pal. Systematic evaluation of causal discovery in visual model-based reinforcement learning. *arXiv preprint arXiv:2107.00848*, 2021.

[Kipf *et al.*, 2020] Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representation*, 2020.

[Kootbally *et al.*, 2015] Zeid Kootbally, Craig Schlenoff, Christopher Lawler, Thomas Kramer, and Satyandra K Gupta. Towards robust assembly with knowledge representation for the planning domain definition language (pddl). *Robotics and Computer-Integrated Manufacturing*, 33:42–55, 2015.

[Kumor *et al.*, 2021] Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. Sequential causal imitation learning with unobserved confounders. In *Advances in Neural Information Processing Systems*, 2021.

[Lattimore *et al.*, 2016] Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems*, 2016.

[Lee and Bareinboim, 2020] Sanghack Lee and Elias Bareinboim. Characterizing optimal mixed policies: Where to intervene and what to observe. In *Advances in Neural Information Processing Systems*, 2020.

[Locatello *et al.*, 2020] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

[McDermott *et al.*, 1998] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl: the planning domain definition language. 1998.

[Mnih *et al.*, 2015] 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):529–533, 2015.

[Murphy, 2002] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. University of California, Berkeley, 2002.

[Namkoong *et al.*, 2020] Hongseok Namkoong, Ramtin Keramati, Steve Yadlowsky, and Emma Brunskill. Off-policy policy evaluation for sequential decisions

under unobserved confounding. In *Advances in Neural Information Processing Systems*, 2020.

[Pearl and Verma, 1995] Judea Pearl and Thomas S Verma. A theory of inferred causation. In *Studies in Logic and the Foundations of Mathematics*, volume 134, pages 789–811. Elsevier, 1995.

[Pearl, 2009] Judea Pearl. *Causality*. Cambridge university press, 2009.

[Pednault, 1989] Edwin P. D. Pednault. Adl: Exploring the middle ground between strips and the situation calculus. In *International Conference on Principles of Knowledge Representation and Reasoning*, 1989.

[Pitis *et al.*, 2020] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. In *Advances in Neural Information Processing Systems*, 2020.

[Ruan *et al.*, 2023] Kangrui Ruan, Junzhe Zhang, Xuan Di, and Elias Bareinboim. Causal imitation learning via inverse reinforcement learning. In *International Conference on Learning Representations*, 2023.

[Schölkopf *et al.*, 2021] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[Schölkopf, 2022] Bernhard Schölkopf. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 765–804. 2022.

[Shpitser and Pearl, 2006] Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *AAAI Conference on Artificial Intelligence*, 2006.

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[Srinivasan *et al.*, 2021] Ranjani Srinivasan, Jaron JR Lee, Rohit Bhattacharya, and Ilya Shpitser. Path dependent structural equation models. In *Uncertainty in Artificial Intelligence*, 2021.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and MultiAgent Systems*, 2018.

[Tian and Pearl, 2002] Jin Tian and Judea Pearl. A general identification condition for causal effects. In *AAAI Conference on Artificial Intelligence*, 2002.

[Tian, 2008] Jin Tian. Identifying dynamic sequential plans. In *Uncertainty in Artificial Intelligence*, 2008.

[Wang *et al.*, 2021] Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. In *Advances in Neural Information Processing Systems*, 2021.

[Wang *et al.*, 2022] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. In *International Conference on Machine Learning*, 2022.

[Watters *et al.*, 2019] Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

[Zhang and Bareinboim, 2016] Junzhe Zhang and Elias Bareinboim. Markov decision processes with unobserved confounders: A causal approach. Technical report, R-23, Purdue AI Lab, 2016.

[Zhang and Bareinboim, 2022] Junzhe Zhang and Elias Bareinboim. Can humans be out of the loop? In *Conference on Causal Learning and Reasoning*, 2022.

[Zhang *et al.*, 2019] Amy Zhang, Zachary C Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, and Tommaso Furlanello. Learning causal state representations of partially observable environments. *arXiv preprint arXiv:1906.10437*, 2019.

[Zhao *et al.*, 2022] Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson LS Wong. Toward compositional generalization in object-oriented world modeling. In *International Conference on Machine Learning*. PMLR, 2022.