Learning Causal Dynamics Models in Object-Oriented Environments

Zhongwei Yu [1 2] Jingqing Ruan [1 2] Dengpeng Xing [1 2]

Abstract

Causal dynamics models (CDMs) have demonstrated significant potential in addressing various challenges in reinforcement learning. To learn CDMs, recent studies have performed causal discovery to capture the causal dependencies among environmental variables. However, the learning of CDMs is still confined to small-scale environments due to computational complexity and sample efficiency constraints. This paper aims to extend CDMs to large-scale object-oriented environments, which consist of a multitude of objects classified into different categories. We introduce the Object-Oriented CDM (OOCDM) that shares causalities and parameters among objects belonging to the same class. Furthermore, we propose a learning method for OOCDM that enables it to adapt to a varying number of objects. Experiments on large-scale tasks indicate that OOCDM outperforms existing CDMs in terms of causal discovery, prediction accuracy, generalization, and computational efficiency.

1. Introduction

Reinforcement learning (RL) (Sutton & Barto, 2018) and causal inference (Pearl, 2000) have separately made much progress over the past decades. Recently, the combination of both fields has led to a series of successes (Zeng et al., 2023), where the use of *causal dynamics models* (CDMs) proves a promising direction. CDMs capture the causal structures of environmental dynamics and have been applied to address a wide range of challenges in RL, including learning efficiency, explainability, generalization, state representation, subtask decomposition, and transfer learning (see Section 2.1). For example, a major function of CDMs is to reduce spurious correlations (Ding et al., 2022; Wang et al., 2022), which are particularly prevalent in the non-i.i.d. data produced in sequential decision-making.

Early research of CDMs exploits given causal structures of environments (Boutilier et al., 2000; Guestrin et al., 2003b; Madumal et al., 2020b), which may not be available in many applications. Therefore, some recent studies have proposed to develop CDMs using causal discovery techniques to learn such causal structures, i.e. causal graphs (CGs), from the data of history interactions (Volodin, 2021; Wang et al., 2021; 2022; Zhu et al., 2022). These approaches have been successful in relatively small environments consisting of a few variables. Unfortunately, some RL tasks involve many objects (e.g., multiple agents and environment entities in multi-agent domains (Malysheva et al., 2019)), which together contribute to a large set of environment variables. The applicability of CDMs in such large-scale environments remains questionable — the excessive number of potential causal dependencies (i.e., edges in CGs) makes causal discovery extremely expensive, and more samples and effort are required to correctly discriminate causal dependencies.

Interestingly, humans can efficiently reason causality from enormous real-world information. One possible explanation for this is that we intuitively perceive tasks through an object-oriented (OO) perspective (Hadar & Leron, 2008) — we decompose the world into objects and categorize them into classes, allowing us to summarize and share causal rules for each class. For example, "exercise causes good health of each person" is a shared rule of the class "Human", and "each person" represents any instance of that class. This OO intuition has been widely adopted in modern programming languages, referred to as object-oriented programming (OOP), to organize and manipulate data in a more methodical and readable fashion (Stroustrup, 1988).

This work aims to extend CDMs to large-scale OO environments. Inspired by OOP, we investigate how an OO description of the environment can be exploited to facilitate causal discovery and dynamics learning. We propose the *Object-Oriented Causal Dynamics Model* (OOCDM), a novel type of CDM that allows the sharing of causalities and model parameters among objects. To learn the OOCDM, we present a modified version of Causal Dynamics Learning (CDL) (Wang et al., 2022) that can accommodate varying numbers of objects. We theoretically prove

[1]Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China [2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, P. R. China. Correspondence to: Dengpeng Xing <dengpeng.xing@ia.ac.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

that the proposed approach discovers the ground-truth CG under a few natural assumptions. Additionally, we apply OOCDM to several OO domains and demonstrate that it outperforms state-of-the-art CDMs in terms of causal graph accuracy, prediction accuracy, generalization ability, and computational efficiency, especially for large-scale tasks. To the best of our knowledge, OOCDM is the first dynamics model to combine causality with the object-oriented settings in RL. The source code of this work is made available at `https://github.com/EaseOnway/oocdm`.

2. Related works

2.1. Causality and Reinforcement Learning

Causality (see basics in Appendix B) formulates dependencies among random variables and is used across various disciplines (Pearl, 2000; Pearl et al., 2016; Pearl & Mackenzie, 2019). One direction to combine causality with RL is to formulate a known causal structure among *macro* elements (e.g., the observation, action, reward, and hidden states) of the Markov Decision Process (MDP). Algorithms with improved robustness and efficiency are then derived by applying the causal inference techniques (Buesing et al., 2018; Lu et al., 2018; Zhang et al., 2020; Liao et al., 2021; Guo et al., 2022).

This paper follows another direction on the *micro* causality that exists among specific components of the environment. Modular models prove capable of capturing such causality using independent sub-modules, leading to better generalization and learning performance (Ke et al., 2021; Mittal et al., 2020; 2022). A popular setting for the micro causality is *Factored MDP* (FMDP) (Boutilier et al., 2000), where the transition dynamics is modeled by a CDM. Knowledge to this CDM benefits RL in many ways, including 1) efficiently solving optimal policies (Guestrin et al., 2003b; Osband & Van Roy, 2014; Xu & Tewari, 2020), 2) sub-task decomposition (Jonsson & Barto, 2006; Peng et al., 2022), 3) improving explainability (Madumal et al., 2020a;b; Volodin, 2021; Yu et al., 2023), 4) improving generalization of policies (Nair et al., 2019) and dynamic models (Ding et al., 2022; Wang et al., 2022; Zhu et al., 2022), 5) learning task-irrelevant state representations (Wang et al., 2021; 2022), 6) policy transfer to unseen domains (Huang et al., 2022). Additionally, Feng et al. (2022) presents an extension of FMDP that has non-stationary CDMs.

2.2. Object-Oriented Reinforcement Learning

It is common in RL to describe environments using multiple objects. Researchers have largely explored object-centric representation (OCR), especially in visual domains, to facilitate policy learning (Zambaldi et al., 2018; Zadaianchuk et al., 2020; Zhou et al., 2022; Yoon et al., 2023) or dynamic modeling (Zhu et al., 2018; 2019; Kipf et al., 2020; Locatello et al., 2020). However, OCR typically uses homogeneous representations of objects and struggles to capture the diverse nature of objects. Goyal et al. (2020; 2022) overcome this problem by extracting a set of dynamics templates (called *schemata* or *rules*) that are matched with objects to predict next states. Prior to our work, Guestrin et al. (2003a) and Diuk et al. (2008) investigated OOP-style MDP representations using predefined classes of objects.

2.3. Relational Causal Discovery

It is common to use the structural priors to improve the efficiency of causal discovery. Similar to this work, relational causal discovery (Maier et al., 2010) attempts to modularize causal dependencies using additional knowledge about objects in relation domains. Marazopoulou et al. (2015) further provides a temporal extension of the technique for sequential data. However, relational causal discovery requires stronger priors than our work – not only the description of classes and objects but also the formulation of inter-object relations. Our work focuses on the FMDP settings where relations are implicit and unknown, which may contribute to more general use. In addition, relational causal discovery does not include a good dynamics model that fully exploits the object-oriented priors.

3. Preliminaries

Notations A random variable is denoted by a capital letter (e.g., X_1 and X_2). Brackets may combine variables or subgroups into a *group* (an ordered set) denoted by a bold letter, e.g. $\mathbf{X} = (\mathrm{X}_1, \mathrm{X}_2)$ and $\mathbf{Z} = (\mathbf{X}, \mathrm{Y}_1, \mathrm{Y}_2)$. We use p to denote a distribution. In addition, Appendix A provides a thorough list of notations in this paper.

3.1. Causal Dynamics Models

We consider the FMDP setting where the state and action consist of multiple random variables, denoted as $\mathbf{S} = (\mathrm{S}_1, \cdots, \mathrm{S}_{n_s})$ and $\mathbf{A} = (\mathrm{A}_1, \cdots, \mathrm{A}_{n_a})$, respectively. S'_i (or $\mathbf{S}'$) denotes the state variable(s) in the next step. The transition probability $p(\mathbf{S}'|\mathbf{S}, \mathbf{A})$ is modeled by a CDM (see Definition 3.1), which is also referred to as a *Dynamics Bayesian Network* (DBN) (Dean & Kanazawa, 1989) adapted to the context of RL. For clarity, we illustrate a simple deterministic CDM in Appendix C.4.

Definition 3.1. A *causal dynamics model* is a tuple $\langle \mathcal{G}, p \rangle$. $\mathcal{G}$ is the *causal graph*, i.e. a directed acyclic graph (DAG) on $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$, defining the parent set $Pa(\mathrm{S}'_j)$ for each S'_j in $\mathbf{S}'$; p is a transition distribution on $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$ such that

$$p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}_j | Pa(\mathrm{S}'_j)). \tag{1}$$

In this work, $\mathcal{G}$ is unknown and must be learned from the data. Some studies learn CGs using sparsity constraints, which encourage models to predict the transition using fewer inputs (Volodin, 2021; Wang et al., 2021). However, there exists no theoretical guarantee that sparsity can lead to sound causality. In several recent studies (Wang et al., 2022; Ding et al., 2022; Zhu et al., 2022; Yu et al., 2023), conditional independence tests (CITs) are used to discover CGs (Eberhardt, 2017). Theorem 3.2 presents a prevalent approach for causal discovery, with proof in Appendix C.3. Additionally, we also prove the robustness of this approach against mild observational noise in Theorem C.13.

Theorem 3.2 (Causal discovery for CDMs)**.** *Assuming that state variables transit independently, i.e.* $p(\mathbf{S}'|\mathbf{S},\mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j|\mathbf{S},\mathbf{A})$*, then the ground-truth causal graph* $\mathcal{G}$ *is bipartite. That is, all edges start in* $(\mathbf{S},\mathbf{A})$ *and end in* $\mathbf{S}'$*; if* p *is a faithful probability function consistent with the dynamics, then* $\mathcal{G}$ *is uniquely identified by*

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow \neg(\mathrm{X}_i \perp\!\!\!\perp_p \mathrm{S}'_j \,|\, (\mathbf{S},\mathbf{A}) \setminus \{\mathrm{X}_i\}), \tag{2}$$

for each $\mathrm{X}_i \in (\mathbf{S},\mathbf{A})$ *and* $\mathrm{S}'_j \in \mathbf{S}'$*. Here, "*$\perp\!\!\!\perp_p$*" denotes the conditional independence under* p*, and "*$\setminus$*" means set-subtraction.*

The independence "$\perp\!\!\!\perp_p$" here can be determined by CITs, which utilize samples drawn from p to evaluate whether the conditional independence holds. There are many tools for CITs, such as Fast CIT (Chalupka et al., 2018), Kernel-based CIT (Zhang et al., 2012), and Conditional Mutual Information (CMI) used in this work. Read Appendix B.4 for more information about CITs and CMI.

Testing Eq. 2 leads to sound CGs, yet is hardly scalable. Let $n := n_a + n_s$ denote the total number of environment variables. Then, the time complexity of mainstream approaches reaches up to $O(n^3)$, since $O(n^2)$ edges must be tested, each costing $O(n)$. Additionally, a larger n impairs sampling efficiency, as CITs require more samples to recover the joint distribution of condition variables.

3.2. Object-Oriented Markov Decision Process

Following Guestrin et al. (2003a), we formulate the task as an *Object-Oriented MDP* (OOMDP) containing a set $\mathcal{O} = \{O_1, \cdots, O_N\}$ of *objects*. Each object O_i corresponds to a subset of variables (called its *attributes*), written as $\mathbf{O}_i = (O_i.\mathbf{S}, O_i.\mathbf{A})$, where $O_i.\mathbf{S} \subseteq \mathbf{S}$ and $O_i.\mathbf{A} \subseteq \mathbf{A}$ respectively are its state attributes and action attributes. The objects are divided into a set of classes $\mathcal{C} = \{C_1, \cdots, C_K\}$. We call O_i an *instance* of C_k if O_i belongs to some class C_k, denoted as $O_i \in C_k$. C_k specifies a set $\mathcal{F}[C_k]$ of *fields*, which determine the attributes of O_i as well as other instances of C_k. Each field in $\mathcal{F}[C_k]$, typically written as $C_k.U$ (where U can be replaced by any identifier), signifies an attribute $O_i.\mathrm{U} \in \mathbf{O}_i$ for each $O_i \in C_k$. Note that italic letters are used for identifiers (e.g., $C_k.U$), while Roman letters are used for attributes (e.g., $O_i.\mathrm{U}$) to highlight that attributes are random variables. A more rigorous definition of OOMDP is given in Appendix D.1.

The dynamics of the OOMDP satisfy that *the state variables of objects from the same class transit according to the same (unknown) class-level transition function*:

$$p(O_i.\mathbf{S}'|\mathbf{S},\mathbf{A}) = p_{C_k}(O_i.\mathbf{S}'|\mathbf{O}_i; \mathbf{O}_1, \cdots, \mathbf{O}_{i-1}, \mathbf{O}_{i+1}, \cdots, \mathbf{O}_N) \tag{3}$$

for all $O_i \in C_k$, which we refer to as the *result symmetry*. Diuk et al. (2008) further formulates the dynamics using logical rules, which is not necessarily required here.

This OOMDP representation is inherently available in many simulation platforms or can be intuitively specified from human experience. Therefore, we consider the OOMDP representation as prior knowledge and leave its learning to future work. To illustrate our setting, we present Example 3.3 as the OOMDP for a StarCraft environment.

Example 3.3. In a StarCraft scenario shown in Figure 1, the set of objects is $\mathcal{O} = \{M_1, M_2, Z_1, Z_2, Z_3\}$ and the set of classes is $\mathcal{C} = \{C_M, C_Z\}$. C_M is the class for marines M_1 and M_2. Similarly, C_Z is the class for zerglings Z_1, Z_2, and Z_3. The fields for both $C = C_M, C_Z$ are given by $\mathcal{F}[C] = \{C.H, C.P, C.A\}$ — the **H**ealth, **P**osition, and **A**ction (e.g., move or attack). Therefore, for example, $M_1.\mathrm{H}$ is the health of marine M_1, and $\mathbf{M}_1 = (M_1.\mathrm{H}, M_1.\mathrm{P}, M_1.\mathrm{A})$.

4. Method

The core of an OOCDM is the *Object-Oriented Causal Graph* (OOCG), which allows for class-level causality sharing based on the dynamic similarity between objects of the same class (see Section 4.1). Equation 3 has illustrated this similarity with respect to the result terms of the transition probabilities. Furthermore, we introduce an assumption 4.1 concerning the condition terms, called *causation symmetry*. It provides a natural notion that objects of the same class produce symmetrical effects on other objects. Figure 1 illustrates this assumption using the StarCraft scenario described above — swapping all attributes between two zerglings Z_2 and Z_3 makes no difference to the transition of other objects such as the marine M_2. We also assume that all state variables (attributes) transit independently in accordance with FMDPs (Guestrin et al., 2003b).

Assumption 4.1 (Causation Symmetry)**.** Suppose $O_i \in C_k$. Assuming that $O_a, O_b \in C_l$ $(a, b \neq i)$ are two other objects from the same class, then O_a and O_b are **interchangeable** to the transition of O_i:

$$p(O_i.\mathbf{S}'|\mathbf{O}_a = \boldsymbol{a}, \mathbf{O}_b = \boldsymbol{b}, \cdots) = p(O_i.\mathbf{S}'|\mathbf{O}_a = \boldsymbol{b}, \mathbf{O}_b = \boldsymbol{a}, \cdots). \tag{4}$$

The workflow for using an OOCDM is illustrated in Figure 2.

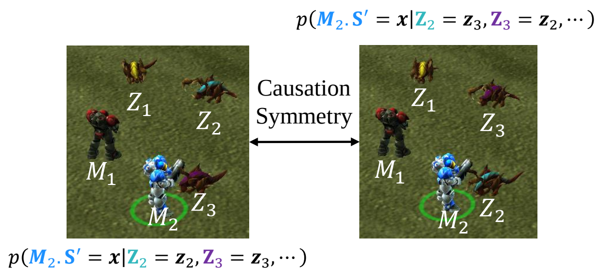

$p(M_2.S' = x | Z_2 = z_3, Z_3 = z_2, \cdots)$
Causation Symmetry
$p(M_2.S' = x | Z_2 = z_2, Z_3 = z_3, \cdots)$
Z_1
Z_2
Z_3
M_1
M_2

Figure 1. An OOMDP for Starcraft with the causation symmetry.

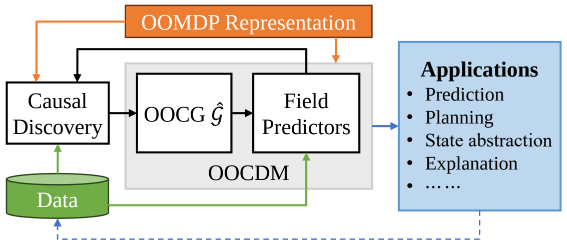
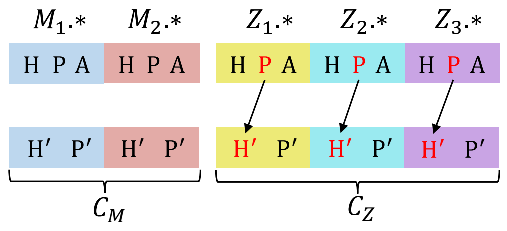
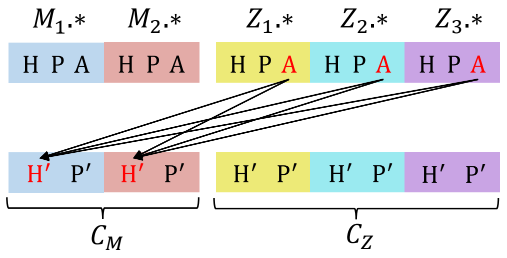


Figure 2. The workflow overview.



(a) $C_Z.P \to H'$.



(b) $C_Z.A \to C_M.H'$

Figure 3. The class-level causalities in Example 3.3.

First, we use domain knowledge about the task to construct its OOMDP representation (Section 3.2). Subsequently, we initialize the OOCDM inclusive of field predictors (Section 4.2) and an OOCG estimation $\hat{G}$. This estimation is updated by performing causal discovery on the transition data and the predictors (Section 4.3), and these predictors are optimized using the current OOCG estimation and the stored data (Section 4.4). The learned OOCDM can then be applied to problems that require a CDM or causal graph (some basic applications are tested in Section 5).

4.1. Object-Oriented Causal Graph

According to Theorem 3.2, the ground-truth CG of an OOMDP follows a *bipartite causal graph* (BCG) structure, where no lateral edge is present in $\mathbf{S}'$. In order to simplify the process of causal discovery, we impose a restriction on the structure of $\mathcal{G}$ and introduce a special form of CGs that allows class-level causal sharing.

Definition 4.2. Let $\mathcal{F}_s[C_k] \subseteq \mathcal{F}[C_k]$ be the set of **state** fields of class C_k. An *Object-Oriented Causal Graph* is a BCG where **all** causal edges are given by a series of class-level causalities:

1. A *class-level local causality* for class C_k from field $C_k.U \in \mathcal{F}[C_k]$ to state field $C_k.V \in \mathcal{F}_s[C_k]$, denoted as $C_k.U \to V'$, means that $O.\mathrm{U} \in Pa(O.\mathrm{V}')$ for every instance $O \in C_k$.
2. A *class-level global causality* from field $C_l.U \in \mathcal{F}[C_l]$ to state field $C_k.V \in \mathcal{F}_s[C_k]$, denoted as $C_l.U \to C_k.V'$, means that $O_j.\mathrm{U} \in Pa(O_i.\mathrm{V}')$ for every $O_i \in C_k$ and every $O_j \in C_l$ $(j \neq i)$.

Definition 4.2 enables causality sharing by two types of class-level causalities, which are invariant with the number of instances of each class. Similar to relational causal discovery (Marazopoulou et al., 2015), this causality sharing greatly simplifies causal discovery and improves the readability of CGs. The *local* causality describes shared structures within individual objects of the same class, as illustrated in Figure 3(a). The *global* causality accounts for shared structures of object pairs, as illustrated in Figure 3(b). Note that the global causality $C_k.U \to C_k.V'$ (i.e., when $k = l$) is different from the local causality $C_k.U \to V'$ by definition. For clarity, the global and local causalities here are different from those considered by Pitis et al. (2020), where "local" means that $(\mathbf{S}, \mathbf{A})$ is confined in a small region in the entire space.

An OOCG representing the dynamics (i.e., Eq. 1 holds) always exists, as a fully-connected BCG is apparently an OOCG. As shown in Theorem 4.3, our major result is that OOCGs reveal the ground-truth causality given the symmetry assumption, with proof in Appendix D.2.

Theorem 4.3. *The ground-truth CG of any OOMDP where Assumption 4.1 holds is exactly an OOCG.*

4.2. Object-Oriented Causal Dynamics Model

Definition 4.4. An *object-oriented causal dynamics model* is a CDM $\langle \mathcal{G}, \hat{p} \rangle$ (see Definition 3.1) such that 1) $\mathcal{G}$ is an OOCG, and 2) $\hat{p}$ satisfies Eqs. 3 and 4.

Based on OOCGs, we are able to define CDMs in an object-oriented manner (see Definition 4.4). In conventional CDMs, there exists an independent predictor for each

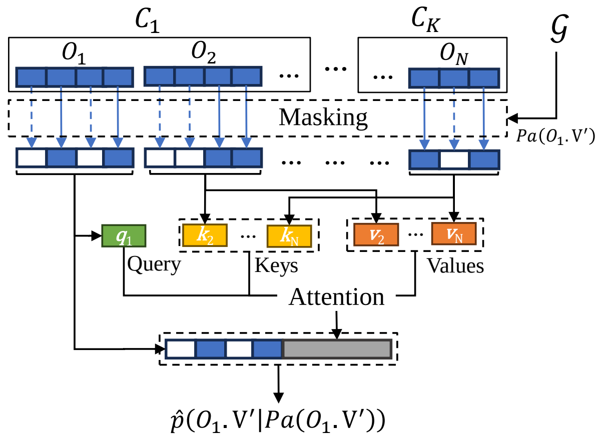


Figure 4. The illustration of $f_{C_1.V}(O_1.\mathrm{V}'|\mathbf{S},\mathbf{A};\mathcal{G})$.

next-state attribute (variable) in $\mathbf{S}'$. However, Equation 3 offers an opportunity to reduce the number of predictors by class-level sharing. That is, a shared *field predictor* $f_{C.V}$ is used for each state field $C.V \in \mathcal{F}[C]$ to predict the corresponding attribute $O.\mathrm{V}'$ for every instance $O \in C$.

We now briefly describe how an OOCDM is implemented in our work. Inspired by Wang et al. (2022), we let an OOCG $\mathcal{G}$ be an argument of the predictor $f_{C.V}$, making it adaptable to various graph structures. Therefore, in our implementation, it follows that

$$\hat{p}(O.\mathrm{V}'|Pa_{\mathcal{G}}(O.\mathrm{V}')) = f_{C.V}(O.\mathrm{V}'|\mathbf{S},\mathbf{A};\mathcal{G}) \tag{5}$$

for every $O \in C$, where $Pa_{\mathcal{G}}(O.\mathrm{V}')$ is the parent set of $O.\mathrm{V}'$ in $\mathcal{G}$. We ensure that $f_{C.V}$ adheres to $\mathcal{G}$ by masking off the non-parental variables. In addition, we adopt key-value attention (Vaswani et al., 2017) to ensure causation symmetry (Eq. 4) and enable adaptation to varying numbers of objects. A simple illustration of our implementation of $f_{C.V}$ is given as Figure 4, and detail is in Appendix E.

4.3. Object-Oriented Causal Discovery

Theorem 4.3 indicates that causal discovery in an OOMDP with Assumption 4.1 becomes looking for an OOCG. If the numbers of instances are fixed, checking each class-level causality in the OOCG only requires one CIT (see Appendix D.3), where most CIT tools are applicable. Further, to perform CITs in environments with changeable instance numbers, we introduce an adaptation of CDL using the class-level conditional mutual information.

Assume that we have a dataset $\mathcal{D} = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})\}_{t=1}^{T}$, where $\boldsymbol{s}_t$, $\boldsymbol{a}_t$ and $\boldsymbol{s}_{t+1}$ are the observed values of $\mathbf{S}$, $\mathbf{A}$ and $\mathbf{S}'$ at step t, respectively. We use $O.v_{t+1}$ to denote the observed $O.\mathrm{V}$ in $\boldsymbol{s}_{t+1}$ for each state field $C_k.V$ and instance $O \in C_k$. Some OOCGs are helpful to the estimation of CMI: **I)** $\mathcal{G}_1$ is the **full** bipartite CG containing all causalities, which is also an OOCG by definition; **II)** $\mathcal{G}_{C.U\not\to V'}$ contains all causalities except for $C.U \to V'$; and **III)** $\mathcal{G}_{C_k.U\not\to C.V'}$ contains all causalities except for $C_k.U \to C.V'$. Letting C_k^t denotes the set of instances of class C_k at step t, with the predictors introduced in Section 4.2, we respectively write the CMIs for class-level local and global causalities as

$$\mathcal{I}_{\mathcal{D}}^{C_k.U\to V'} := \frac{1}{\sum_{t=1}^{T}|C_k^t|}\sum_{t=1}^{T}\sum_{O\in C_k^t} \log\frac{f_{C_k.V}(O.v_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t;\mathcal{G}_1)}{f_{C_k.V}(O.v_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t;\mathcal{G}_{C_k.U\not\to V'})}, \tag{6}$$

$$\mathcal{I}_{\mathcal{D}}^{C_l.U\to C_k.V'} := \frac{1}{\sum_{t=1}^{T}|C_k^t|}\cdot\sum_{t=1}^{T}\sum_{O_j\in C_k^t} \log\frac{f_{C_k.V}(O.v_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t;\mathcal{G}_1)}{f_{C_k.V}(O.v_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t;\mathcal{G}_{C_l.U\not\to C_k.V'})}. \tag{7}$$

Then, each class-level causality (denoted as ς) is confirmed if $\mathcal{I}_{\mathcal{D}}^{\varsigma} > \varepsilon$, where ε is a positive threshold parameter. In other words, $\mathcal{I}_{\mathcal{D}}^{\varsigma}$ compare the predictions made with and without the concerned parents within ς, and we confirm the causality if the difference is significant. In this way, no extra models are needed for causal discovery. Finally, the whole OOCG is obtained by checking CMIs for all possible causalities (see Appendix E.3 for the pseudo-code).

Our approach greatly reduces the computational complexities of causal discovery, from a magnitude (asymptotic boundary) of n^3 to a magnitude of Nmn, where m denotes the overall number of fields and n denotes the overall number of variables in $(\mathbf{S},\mathbf{A})$. See proofs and more conclusions about computational complexities in Appendix F.

4.4. Model Learning

Dynamics models are usually optimized through Maximum Likelihood Estimation. To better adapt to the varying numbers of instances, we define the *average instance log-likelihood* (AILL) function on a transition dataset $\mathcal{D}$ of T steps for any CDM $\langle\mathcal{G},\hat{p}\rangle$ as

$$\mathcal{L}_{\mathcal{G}}(\mathcal{D}) := \sum_{k=1}^{K}\frac{1}{\sum_{t=1}^{T}|C_k^t|}\sum_{t=1}^{T}\sum_{C_k.V\in\mathcal{F}_s[C_k]}\sum_{O\in C_k^t} \log\hat{p}(O.\mathrm{V}'|Pa_{\mathcal{G}}(O.\mathrm{V}'))_t, \tag{8}$$

where $\hat{p}(\cdot)_t$ is the estimated probability when variables take the values observed at step t in $\mathcal{D}$.

The learning target of an OOCDM mimics that of CDL. First, we optimize the AILL function under a random OOCG denoted as $\mathcal{G}_\lambda$ (re-sampled when each time used) where the probability of each class-level causality item is λ. This will make our model capable of handling incomplete information and adaptable to different OOCGs including those like $\mathcal{G}_{C.U\not\to V'}$ or $\mathcal{G}_{C_k.U\not\to C.V'}$. Furthermore, we also hope to strengthen our model in two particular OOCGs: 1) the estimation of ground-truth $\hat{\mathcal{G}}$ obtained by causal discovery, where CMIs are estimated by the current model, and 2) the full OOCG $\mathcal{G}_1$ to better estimate CMIs in Eqs. 6 and 7. Therefore, two additional items, $\mathcal{L}_{\mathcal{G}_1}(\mathcal{D})$ and $\mathcal{L}_{\hat{\mathcal{G}}}(\mathcal{D})$, respectively weighted by α and β, are considered in the overall target function:

$$J(\mathcal{D}) = \mathcal{L}_{\mathcal{G}_\lambda}(\mathcal{D}) + \alpha\mathcal{L}_{\mathcal{G}_1}(\mathcal{D}) + \beta\mathcal{L}_{\hat{\mathcal{G}}}(\mathcal{D}), \tag{9}$$

which is optimized by gradient ascent. Pseudo-code of the learning algorithm is in Appendix E.4. During the test phase, all predictions of our OOCDM are made using the discovered OOCG $\hat{\mathcal{G}}$.

4.5. Releasing Dynamic Symmetries

According to the above definition, OOCDMs comply with the dynamic symmetries (Eqs. 3 and 4), and thus the environmental dynamics are required to have the same properties. This paper mainly discusses symmetric dynamics, as dynamic symmetries are natural in large-scale environments with varying numbers of objects, and they can always be ensured by using a proper representation of OOMDP that provides sufficient detail. For instance, if we formulate the StarCraft case in Example 3.3 using only one class C_{Unit} for both marines and zerglings, it may violate the causation symmetries, whereas using the two-class representation or introducing more attributes (e.g., the attacking range and faction) can ensure the symmetries.

However, the refinement of representations is sometimes not possible, due to the limited access or insufficient domain knowledge of users. Therefore, we look forward to releasing the requirement of dynamic symmetries on the already-given OOMDPs. Theoretically, it is always plausible to ensure dynamic symmetries via *auxiliary attributes* (see Appendix D.4). Therefore, we can augment the OOCDM using built-in auxiliary attributes, which carry the information personalizing each object. In fact, the simplest way to do so is to include the indices of the objects as attributes. However, mapping indices into the personal dynamics seems indirect for the neural predictors. Instead, the augmented OOCDM utilizes the hidden encoding of auxiliary attributes:

$$\hat{p}(O.\mathrm{V}'|Pa_{\mathcal{G}}(O.\mathrm{V}')) = f_{C.V}(O.\mathrm{V}'|\mathbf{O}_1, \boldsymbol{h}_1, \cdots, \mathbf{O}_N, \boldsymbol{h}_N; \mathcal{G}), \tag{10}$$

where $\boldsymbol{h}_i$ is the learned hidden encoding of auxiliary attributes for O_i. The hidden encodings for each class are generated by a bi-directional Gate Recurrent Unit, which can handle varying numbers of instances. The implementation detail is provided in Appendix E.2.

The augmentation allows OOCDM to learn the personal dynamics of each object. Since the ground-truth CG in an asymmetric OOMDP may not be an OOCG, object-oriented causal discovery leads to the minimal OOCG that represents the dynamics. To obtain a precise CG, we can apply OOCDM to non-OO causal discovery based on Theorem 3.2 (e.g., using the CDL approach), which takes advantage of class-level parameter sharing and inter-object attention. However, we suggest using the minimal OOCG to capture the approximate causality, which strikes a good balance between computational efficiency and structural precision, especially in large environments where non-OO CDMs fail. Additional experiments of the augmented OOCDM using OOCGs are included in Appendix H.7.

5. Experiments

OOCDM was compared with several state-of-the-art CDMs. **CDL** uses pooling-based predictors and also adopts CMIs for causal discovery. **CDL-A** is the attention-based variant of CDL, used to make a fair comparison with our model. **GRADER** (Ding et al., 2022) employs Fast CIT for causal discovery and Gated Recurrent Units as predictors. **TICSA** (Wang et al., 2021) utilizes score-based causal discovery. Meanwhile, OOCDM was compared to non-causal baselines, including a widely used multi-layer perceptron (**MLP**) in model-based RL (MBRL) and an object-aware Graph Neural Network (**GNN**) that uses the architecture of (Kipf et al., 2020) to learn inter-object relationships. Additionally, we assessed the performance of the dense version of our OOCDM, namely **OOFULL**, which employs the full OOCG $\mathcal{G}_1$ and is trained by optimizing $\mathcal{L}_{\mathcal{G}_1}$.

As mentioned in Section 2.1, CDMs are used for various purposes, and this work does not aim to specify the use of OOCDMs. Therefore, we evaluate the performance of causal discovery and the predicting accuracy, as most applications can benefit from such criteria. As a common application in MBRL, we also evaluate the performance of planning using dynamics models. Our experiments aim to 1) demonstrate that the OO framework greatly improves the effectiveness of CDMs in large-scale environments, and 2) investigate in what occasions causality brings significant advantages. Moreover, additional results on noisy data are included in Appendix I, which evaluate the robustness against observational noise. Results are presented by the means and standard variances of 5 random seeds. Experimental details are presented in Appendix H.

5.1. Environments

We conducted experiments in 4 environments. The **Block** environment consists of several instances of class $Block$ and one instance of class $Total$. The attributes of each $Block$ object transit via a linear transform; and the attributes of the $Total$ object transit based on the maximums of attributes of the $Block$ objects. The **Mouse** environment is an 8×8 grid world containing an instance of class $Mouse$, and several instances of class $Food$, $Monster$, and $Trap$. The mouse

Table 1. The accuracy (in percentage) of discovered causal graphs. n indicates the number of environmental variables.

Env	n	GRADER	CDL	CDL-A	TICSA	**OOCDM**
Block$_2$	12	94.8$\pm$1.3	99.4$\pm$0.3	99.2$\pm$1.3	97.0$\pm$0.4	**99.7**$\pm$0.6
Block$_5$	24	94.0$\pm$1.5	97.5$\pm$1.5	99.3$\pm$0.6	96.3$\pm$0.6	**100.0**$\pm$0.0
Block$_{10}$	44	92.3$\pm$0.9	97.6$\pm$0.3	99.5$\pm$0.3	97.7$\pm$0.5	**100.0**$\pm$0.0
Mouse	28	90.5$\pm$0.8	90.4$\pm$3.2	94.7$\pm$0.2	94.1$\pm$0.2	**100.0**$\pm$0.0

Table 2. The time (seconds) used in causal discovery. m denotes the number of all fields. The sample sizes are 10k, 50k, 100k, and 200k, respectively in Block, Mouse, CMD, and DZB. The only exception is GRADER in DZB, which only uses 20k samples to make sure that causal discovery finishes within a tolerable time. TICSA is excluded from comparison as it does not involve an explicit causal discovery phase.

Env	n	m	GRADER	CDL	CDL-A	**OOCDM**
Block$_2$	12	8	114.0$\pm$1.5	**1.4**$\pm$0.1	2.1$\pm$0.4	2.1$\pm$0.2
Block$_5$	24	8	927.0$\pm$69.3	5.2$\pm$0.6	8.5$\pm$2.8	**2.1**$\pm$0.2
Block$_{10}$	44	8	$7.0e3$$\pm$217.7	15.6$\pm$0.7	22.8$\pm$5.3	**2.2**$\pm$0.2
Mouse	29	10	$1.7e4$$\pm$138.4	57.8$\pm$2.4	45.3$\pm$2.6	**17.7**$\pm$4.0
CMS	44	4	$5.5e4$$\pm$397.1	209.8$\pm$18.9	252.5$\pm$27.3	**7.4**$\pm$0.5
DZB	66	10	$2.7e4$$\pm$387.1	715.6$\pm$10.9	$1.1e3$$\pm$274.7	**66.1**$\pm$0.6

can be killed by hunger or monsters, and its goal is to survive as long as possible. The Collect-Mineral-Shards (**CMS**) and Defeat-Zerglings-Baineling (**DZB**) environments are StarCraftII mini-games (Vinyals et al., 2017). In CMS, the player controls two marines to collect 20 mineral shards scattered on the map, and in DZB the player controls a group of marines to kill hostile zerglings and banelings. Read Appendix G for detailed descriptions of these environments.

The Block and Mouse environments are ideal OOMDPs as they guarantee Eqs. 3 and 4. In addition, we intentionally insert spurious correlations in them to verify the effectiveness of causal discovery. In CMS and DZB environments, we formulate the objects and classes based on the units and their types in StarCraftII. Such formulation accounts for more piratical cases of imperfect OOMDPs, as the StarCraftII engine may not guarantee Eqs. 3 and 4, yet dynamic symmetries should roughly hold by intuition. For dynamics that are clearly asymmetric, please read Appendix H.7.

5.2. Performance of Causal Discovery

We measured the performance of causal discovery using offline data in Block and Mouse environments. Since non-OO baselines only accept a fixed number of variables, the number of instances of each class is fixed in these environments. Especially, we use "Block$_k$" to denote the Block environment where the number of $Block$ instances is fixed to k. We exclude CMS and DZB here as their ground-truth CGs are unknown (see learned OOCGs in Appendix H.6). We measure the accuracy of discovered CGs by the Structural Hamming Distance within the edges from $(\mathbf{S}, \mathbf{A})$ to $\mathbf{S}'$. As shown in Table 1, OOCDM outperforms other CDMs in all environments and recovers ground-truth CGs in 3 out of 4 environments. These results demonstrate the improved sample efficiency of OOCDM in large-scale environments.

Table 2 shows the computation time used by causal discovery. We note that such results may be influenced by implementation detail and hardware conditions, yet the OOCDM excels baselines with a significant gap beyond these extraneous influences. In addition, Appendix H.5 shows that OOCDM achieves better performance with a relatively smaller size (i.e. fewer model parameters).

5.3. Predicting Accuracy

We use the AILL functions (Eq. 8) to measure the predicting accuracy of dynamics models. The models are learned using offline **train**ing data. Then, the AILL functions of these models are evaluated on the **i.d.** (in-distribution) test data sampled from the same distribution as the training data. Especially, in Block and Mouse environments, we can modify the distribution of the starting state of each episode (see Appendix H.3) and obtain the **o.o.d.** (out-of-distribution) test data, which contains samples that are unlikely to appear during training. The i.d. and o.o.d. test data measure two levels of generalization, respectively considering situations that are alike and unalike to those in training. We do not collect the o.o.d. data for CMS and DZB, as the PySC2 platform provides limited access to modify the initialization process in the StarCraft engine (Vinyals et al., 2017).

The results are shown in Table 3. In small-scale environments like Block$_2$, causal models show better generalization ability than dense models on both i.d. and o.o.d. test data. However, in larger-scale environments, the performance of non-OO models declines sharply, and OO models (OOFULL and OOCDM) obtain the highest performance on the i.d. data. In addition, our OOCDM exhibits the best generalization ability on the o.o.d. data; in contrast, the performance of OOFULL is extremely low on such data. These results demonstrate that OO models are more effective in large-scale environments, and that causality greatly improves the generalization of OO models.

5.4. Combining Models with Planning

In this experiment, we trained dynamics models using offline data (collected through random actions). Given a reward function, we used these models to guide decision-making using Model Predictive Control (Camacho & Bordons, 1999) combined with Cross-Entropy Method (Botev et al., 2013) (see Appendix E.5), which is widely used in MBRL. The Block environment is not included here as it does not involve rewards. In the Mouse environment, the o.o.d. initialization mentioned in Section 5.3 is also consid-

Table 3. The average instance log-likelihoods of the dynamics models on various datasets. We do not show the standard variances for obviously over-fitting results (less than −100.0, highlighted in brown), as their variances are all extremely large.

Env	data	GRADER	CDL	CDL-A	TICSA	GNN	MLP	OOFULL	**OOCDM**
Block_2	train	21.1±0.3	20.9±1.5	19.3±1.9	17.4±2.2	18.8±0.6	16.5±1.2	21.5±0.9	**22.4**±0.7
	i.d.	17.1±2.5	20.2±1.8	10.4±16.8	16.4±1.9	17.9±0.7	10.1±4.4	−568.2	**22.2**±0.7
	o.o.d.	−65.4	11.5±6.7	−6.0e5	−60.1±2.8	−5.0±23.4	−7.2e4	−4.6e4	**21.3**±1.9
Block_5	train	19.1±3.4	16.5±2.1	18.9±0.7	12.0±0.7	14.9±14.4	12.6±0.5	**20.4**±1.7	19.6±1.7
	i.d.	6.7±4.3	−45.3±113.2	−1.4e7	10.8±0.7	14.4±0.4	−2.2±6.3	**19.8**±1.7	19.5±1.7
	o.o.d.	−95.6±41.7	−5.3e6	−1.1e9	−5.5e3	−13.4±3.4	−1.5e7	−4.0e7	**13.5**±4.3
Block_{10}	train	19.3±0.6	12.9±0.8	16.0±0.6	11.1±1.3	13.3±0.15	8.9±0.6	20.3±0.6	**21.2**±0.3
	i.d.	−26.7±8.4	6.9±6.4	−9.2±42.5	−10.4±39.8	12.9±0.2	−75.3±20.0	20.2±0.6	**21.1**±0.3
	o.o.d.	−119.1	−4.2e6	−1.9e8	−139.4	−17.3±17.3	−780.9	−5.4e3	**15.6**±5.4
Mouse	train	24.2±0.6	13.9±1.8	22.3±1.4	13.6±3.5	25.6±1.8	5.7±0.4	30.0±1.4	**32.2**±1.1
	i.d.	−3.2e3	−2.0e5	−3.6e4	−1.5e4	−2.7e4	−1.6e7	−65.0±153.3	**26.8**±6.7
	o.o.d.	−7.1e4	−1.1e10	−2.0e10	−2.5e7	−6.3e10	−8.0e10	−1.5e9	**11.2**±17.2
CMS	train	−1.2±0.1	3.6±0.8	4.1±1.5	2.8±1.6	6.4±6.2	−2.0±1.5	8.5±1.1	**9.0**±0.5
	i.d.	−1.3±0.1	−1.0e6	4.1±1.5	−16.3±7.4	6.3±0.1	−6.4e9	8.5±1.1	**8.9**±0.5
DZB	train	11.0±1.0	4.2±2.5	12.1±0.1	13.2±1.2	18.0±10.0	−0.9±0.8	**29.0**±0.6	27.2±2.5
	i.d.	−14.9±21.8	−3.3±6.6	5.3±5.3	−2.4e5	13.0±12.8	−1.6e12	22.6±5.6	**24.4**±5.9

Table 4. The average return of episodes when models are used for planning. In the Mouse environment, "o.o.d." indicates the initial states are sampled from a new distribution.

Env	GRADER	CDL	CDL-A	TICSA	GNN	MLP	OOFULL	**OOCDM**
Mouse	−1.2±1.9	3.9±3.0	−5.0±1.3	−0.8±0.7	6.6±3.2	0.6±2.0	77.9±18.1	**80.1**±16.9
o.o.d.	−0.4±1.7	1.8±2.5	−0.9±1.1	−1.2±0.6	0.6±0.2	−1.3±0.7	62.2±8.7	**75.1**±17.5
CMS	−9.5±1.1	−9.8±1.1	−8.8±0.4	−9.3±0.9	−9.8±0.7	−8.8±0.5	−4.1±3.3	**3.4**±6.3
DZB	202.9±12.3	217.3±12.4	171.7±18.2	188.9±8.5	233.8±19.8	205.4±6.7	**269.8**±21.5	266.2±11.4

Table 5. Results on various tasks in the Mouse environment, measuring the average instance log-likelihood and the episodic return. "seen" and "unseen" respectively indicate the performances measured in seen and unseen tasks.

Model	log-likelihood			episodic return	
	train	seen	unseen	seen	unseen
OOCDM	26.9±3.5	**25.4**±2.8	**24.8**±2.8	**94.8**±29.7	**88.8**±34.8
OOFULL	**30.7**±1.9	22.5±3.2	7.9±29.8	77.0±24.6	70.8±22.4

ered. The average returns of episodes are shown in Table 4, showing that OOFULL and OOCDM are significantly better than non-OO approaches.

Between the OO models, OOCDM obtains higher returns than OOFULL in 3 of 4 environments, which demonstrates that OOCDM better generalizes to the unseen state-action pairs produced by planning. Taking CMS for example, the agent collects only a few mineral shards in the training data. When the agent plans, it encounters unseen states where most mineral shards have been collected. However, we note that OOFULL performs slightly better than OOCDM in DZB. One reason for this is that DZB possesses a joint action space of 9 marines, which is too large to conduct effective planning. Therefore, planning does not lead to states that are significantly different from those in training, prohibiting the advantage of generalization from converting to the advantage of returns. Additionally, the true CG of DZB is possibly less sparse than those in other environments, making OOFULL contain less spurious edges. Therefore, CDMs would be more helpful, if the true CG is sparse, and there exists a large divergence between the data distributions in training and testing.

5.5. Handling Varying Numbers of Instances

In the Mouse environment, we tested whether OOCDM and OOFULL are adaptable to various tasks with different numbers of $Food$, $Moster$, and $Trap$ instances. We randomly divide tasks into the *seen* and *unseen* tasks (see Appendix H.4). Dynamics models are first trained in *seen* tasks and then transferred to the *unseen* without further training. We measured the log-likelihoods on the training data, the i.d. test data on seen tasks, and the test data on unseen tasks. The average episodic returns of planning were also evaluated, separately on seen and unseen tasks. As shown in Table 5, our results demonstrate that 1) OO models can be learned using data from different tasks, 2) OO models perform a zero-shot transfer to unseen tasks with a mild reduction of performance, and 3) the overall performance is improved when combing the model with causality.

5.6. Discussion on Results

OOCDM greatly outperforms baselines in the generalization performance, where both prior knowledge and causality play a role. We note two possible sources of generalization error for dynamics models: 1) the spurious correlations in the data and 2) the insufficient data representing the joint space of numerous variables. Conventional CDMs can only reduce the first source of errors and still suffer from the second source. For example, CDL-A generalizes worse to o.o.d. datasets even though it implements better causal discovery than CDL, as the attention in CDL-A has a greater regression capacity than the pooling mechanism in CDL, leading to a higher risk of overfitting on the limited data. OOCDM can reduce error from the second source by using shared predictors for each class. Therefore, the improved performance of OOCDM partly derives from good exploitation of OOMDP priors, which suggests a further investigation to learn OOMDP representations in future studies.

However, the prior knowledge alone is insufficient for a good generalization. Lacking a causal structure, OOFULL and GNN fail on the o.o.d. datasets even though they are aware of the object-oriented representation. Therefore, the causal structure is crucial in the generalization performance of OOCDM.

6. Conclusion

This paper proposes OOCDMs that capture the causal relationships within OOMDPs. Our main innovations are the OOCGs that share class-level causalities and the use of attention-based field predictors. Furthermore, we present a CMI-based method that discovers OOCGs in environments with changing numbers of objects. Theoretical and empirical data indicate that OOCDM greatly enhances the computational efficiency and accuracy of causal discovery in large-scale environments, surpassing state-of-the-art CDMs. Moreover, OOCDM well generalizes to unseen states and tasks, yielding commendable planning outcomes. In conclusion, this study provides OOCDM as a promising solution to learn and apply CDMs in large-scale object-oriented environments.

Future work would include extracting OOMDP representations from raw pixel-based features, considering potential unobserved confounders, and modeling the relational interaction of objects.

Acknowledgements

This work was supported by the National Nature Science Foundation of China under Grant 62073324. The paper has gone through multiple revisions, in which suggestions from reviewers are extremely useful in improving the quality of the work, especially in improving the readability and presentation.

Impact Statement

This paper explores the causality of world models in reinforcement learning, which increases the transparency and robustness of the models. Therefore, as a positive impact, we expect this work to contribute to more reliable model-based agents in the future. To the best of our knowledge, this work does not raise any ethical issues.

References

Botev, Z. I., Kroese, D. P., Rubinstein, R. Y., and L'Ecuyer, P. The Cross-Entropy Method for Optimization. In *Handbook of Statistics*, volume 31, pp. 35–59. Elsevier, 2013. ISBN 978-0-444-53859-8. doi: 10.1016/B978-0-444-53859-8.00003-5. URL https://linkinghub.elsevier.com/retrieve/pii/B9780444538598000035.

Boutilier, C., Dearden, R., and Goldszmidt, M. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1):49–107, August 2000. ISSN 0004-3702. doi: 10.1016/S0004-3702(00)00033-3. URL https://www.sciencedirect.com/science/article/pii/S0004370200000333.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym, June 2016. URL http://arxiv.org/abs/1606.01540. arXiv:1606.01540 [cs].

Buesing, L., Weber, T., Zwols, Y., Racaniere, S., Guez, A., Lespiau, J.-B., and Heess, N. Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search, November 2018. URL http://arxiv.org/abs/1811.06272. arXiv:1811.06272 [cs, stat].

Camacho, E. F. and Bordons, C. *Model predictive control*. Advanced textbooks in control and signal processing. Springer, Berlin ; New York, 1999. ISBN 978-3-540-76241-6.

Chalupka, K., Perona, P., and Eberhardt, F. Fast Conditional Independence Test for Vector Variables with Large Sample Sizes, April 2018. URL http://arxiv.org/abs/1804.02747. arXiv:1804.02747 [cs, stat].

Dean, T. and Kanazawa, K. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, February 1989. ISSN 0824-7935, 1467-8640. doi: 10.1111/j.1467-8640.1989.tb00324.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1467-8640.1989.tb00324.x.

Ding, W., Lin, H., Li, B., and Zhao, D. Generalizing Goal-Conditioned Reinforcement Learning with Variational Causal Reasoning, July 2022. URL http://arxiv.org/abs/2207.09081. arXiv:2207.09081 [cs, stat].

Diuk, C., Cohen, A., and Littman, M. L. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 240–247, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390187. URL http://portal.acm.org/citation.cfm?doid=1390156.1390187.

Eberhardt, F. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3(2):81–91, March 2017. ISSN 2364-415X, 2364-4168. doi: 10.1007/s41060-016-0038-6. URL http://link.springer.com/10.1007/s41060-016-0038-6.

Feng, F., Huang, B., Zhang, K., and Magliacane, S. Factored Adaptation for Non-Stationary Reinforcement Learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 31957–31971. Curran Associates, Inc., 2022.

Goyal, A., Lamb, A., Gampa, P., Beaudoin, P., Levine, S., Blundell, C., Bengio, Y., and Mozer, M. Object Files and Schemata: Factorizing Declarative and Procedural Knowledge in Dynamical Systems, November 2020. URL http://arxiv.org/abs/2006.16225. arXiv:2006.16225 [cs, stat].

Goyal, A., Didolkar, A., Ke, N. R., Blundell, C., Beaudoin, P., Heess, N., Mozer, M., and Bengio, Y. Neural Production Systems: Learning Rule-Governed Visual Dynamics, March 2022. URL http://arxiv.org/abs/2103.01937. arXiv:2103.01937 [cs, stat].

Guestrin, C., Koller, D., Gearhart, C., and Kanodia, N. Generalizing plans to new environments in relational MDPs. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, August 2003a. URL https://ai.stanford.edu/~koller/Papers/Guestrin+al:IJCAI03.pdf.

Guestrin, C., Koller, D., Parr, R., and Venkataraman, S. Efficient Solution Algorithms for Factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, October 2003b. ISSN 1076-9757. doi: 10.1613/jair.1000. URL http://arxiv.org/abs/1106.1822. arXiv:1106.1822 [cs].

Guo, J., Gong, M., and Tao, D. A Relational Intervention Approach for Unsupervised Dynamics Generalization in Model-Based Reinforcement Learning, June 2022.

Hadar, I. and Leron, U. How intuitive is object-oriented design? *Communications of the ACM*, 51(5):41–46, May 2008. ISSN 0001-0782, 1557-7317. doi: 10.1145/1342327.1342336. URL https://dl.acm.org/doi/10.1145/1342327.1342336.

Huang, B., Feng, F., Lu, C., Magliacane, S., and Zhang, K. AdaRL: What, Where, and How to Adapt in Transfer Reinforcement Learning, March 2022. URL http://arxiv.org/abs/2107.02729. arXiv:2107.02729 [cs, stat].

Jang, E., Gu, S., and Poole, B. Categorical Reparameterization with Gumbel-Softmax, August 2017. URL http://arxiv.org/abs/1611.01144. arXiv:1611.01144 [cs, stat].

Jonsson, A. and Barto, A. Causal Graph Based Decomposition of Factored MDPs. *The Journal of Machine Learning Research*, 7:2259–2301, December 2006.

Ke, N. R., Didolkar, A., Mittal, S., Goyal, A., Lajoie, G., Bauer, S., Rezende, D., Bengio, Y., Mozer, M., and Pal, C. Systematic Evaluation of Causal Discovery in Visual Model Based Reinforcement Learning, July 2021. URL http://arxiv.org/abs/2107.00848. arXiv:2107.00848 [cs, stat].

Kipf, T., van der Pol, E., and Welling, M. Contrastive Learning of Structured World Models, January 2020. URL http://arxiv.org/abs/1911.12247. arXiv:1911.12247 [cs, stat].

Liao, L., Fu, Z., Yang, Z., Wang, Y., Kolar, M., and Wang, Z. Instrumental Variable Value Iteration for Causal Offline Reinforcement Learning, July 2021. URL http://arxiv.org/abs/2102.09907. arXiv:2102.09907 [cs, stat].

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-Centric Learning with Slot Attention, October 2020. URL http://arxiv.org/abs/2006.15055. arXiv:2006.15055 [cs, stat].

Lu, C., Schölkopf, B., and Hernández-Lobato, J. M. Deconfounding Reinforcement Learning in Observational Settings, December 2018. URL http://arxiv.org/abs/1812.10576. arXiv:1812.10576 [cs, stat].

Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. Distal Explanations for Model-free Explainable Reinforcement Learning, September 2020a. URL http://arxiv.org/abs/2001.10284. arXiv:2001.10284 [cs].

Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. Explainable Reinforcement Learning through a Causal Lens. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2493–2500, April 2020b. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i03.5631. URL https://aaai.org/ojs/index.php/AAAI/article/view/5631.

Maier, M., Taylor, B., Oktay, H., and Jensen, D. Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 531–538. AAAI Press, 2010.

Malysheva, A., Kudenko, D., and Shpilman, A. MAGNet: Multi-agent Graph Network for Deep Multi-agent Reinforcement Learning. In *2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*, pp. 171–176, Moscow, Russia, October 2019. IEEE. ISBN 978-1-72811-944-1. doi: 10.1109/REDUNDANCY48165.2019.9003345. URL https://ieeexplore.ieee.org/document/9003345/.

Marazopoulou, K., Maier, M., and Jensen, D. Learning the structure of causal models with relational and temporal dependence. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pp. 572–581, Arlington, Virginia, USA, 2015. AUAI Press. ISBN 9780996643108.

Mittal, S., Lamb, A., Goyal, A., Voleti, V., Shanahan, M., Lajoie, G., Mozer, M., and Bengio, Y. Learning to Combine Top-Down and Bottom-Up Signals in Recurrent Neural Networks with Attention over Modules, November 2020. URL http://arxiv.org/abs/2006.16981. arXiv:2006.16981 [cs, stat].

Mittal, S., Bengio, Y., and Lajoie, G. Is a Modular Architecture Enough?, June 2022. URL http://arxiv.org/abs/2206.02713. arXiv:2206.02713 [cs] version: 1.

Nair, S., Zhu, Y., Savarese, S., and Fei-Fei, L. Causal Induction from Visual Observations for Goal Directed Tasks, October 2019. URL http://arxiv.org/abs/1910.01751. arXiv:1910.01751 [cs, stat].

Osband, I. and Van Roy, B. Near-optimal Reinforcement Learning in Factored MDPs, October 2014. URL http://arxiv.org/abs/1403.3741. arXiv:1403.3741 [cs, stat].

Pearl, J. *Causality: models, reasoning, and inference*. Cambridge University Press, Cambridge, U.K. ; New York, 2000. ISBN 978-0-521-89560-6 978-0-521-77362-1.

Pearl, J. and Mackenzie, D. *The book of why: the new science of cause and effect*. Penguin science. Penguin Books, London, 2019. ISBN 978-0-14-198241-0.

Pearl, J., Glymour, M., and Jewell, N. P. *Causal inference in statistics: a primer*. Wiley, Chichester, West Sussex, 2016. ISBN 978-1-119-18684-7.

Peng, S., Hu, X., Zhang, R., Tang, K., Guo, J., Yi, Q., Chen, R., Zhang, X., Du, Z., Li, L., Guo, Q., and Chen, Y. Causality-driven Hierarchical Structure Discovery for Reinforcement Learning, October 2022. URL http://arxiv.org/abs/2210.06964. arXiv:2210.06964 [cs].

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachuestts, 2017. ISBN 978-0-262-03731-0.

Pitis, S., Creager, E., and Garg, A. Counterfactual Data Augmentation using Locally Factored Dynamics, December 2020. URL http://arxiv.org/abs/2007.02863. arXiv:2007.02863 [cs, stat].

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Stroustrup, B. What is object-oriented programming? *IEEE Software*, 5(3):10–20, 1988. doi: 10.1109/52.2020.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, 2 edition, 2018. ISBN 978-0-262-36401-0. URL https://mitpress.ublish.com/book/reinforcement-learning-an-introduction-2.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J., Quan, J., Gaffney, S., Petersen, S., Simonyan, K., Schaul, T., van Hasselt, H., Silver, D., Lillicrap, T., Calderone, K., Keet, P., Brunasso, A., Lawrence, D., Ekermo, A., Repp, J., and Tsing, R. StarCraft II: A New Challenge for Reinforcement Learning, August 2017. URL http://arxiv.org/abs/1708.04782. arXiv:1708.04782 [cs].

Volodin, S. *CauseOccam : Learning Interpretable Abstract Representations in Reinforcement Learning Environments via Model Sparsity*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2021.

Wang, Z., Xiao, X., Zhu, Y., and Stone, P. Task-Independent Causal State Abstraction. In *NeurIPS 2021 Workshop on Robot Learning: Self-Supervised and Lifelong Learning*, pp. 10, Virtual, 2021.

Wang, Z., Xiao, X., Xu, Z., Zhu, Y., and Stone, P. Causal Dynamics Learning for Task-Independent State Abstraction, June 2022. URL http://arxiv.org/abs/2206.13452. arXiv:2206.13452 [cs].

Xu, Z. and Tewari, A. Reinforcement Learning in Factored MDPs: Oracle-Efficient Algorithms and Tighter Regret Bounds for the Non-Episodic Setting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18226–18236. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d3b1fb02964aa64e257f9f26a31f72cf-Abstract.html.

Yoon, J., Wu, Y.-F., Bae, H., and Ahn, S. An Investigation into Pre-Training Object-Centric Representations for Reinforcement Learning, June 2023. URL http://arxiv.org/abs/2302.04419. arXiv:2302.04419 [cs].

Yu, Z., Ruan, J., and Xing, D. Explainable Reinforcement Learning via a Causal World Model, May 2023. URL http://arxiv.org/abs/2305.02749. arXiv:2305.02749 [cs].

Zadaianchuk, A., Seitzer, M., and Martius, G. Self-supervised Visual Reinforcement Learning with Object-centric Representations, November 2020. URL http://arxiv.org/abs/2011.14381. arXiv:2011.14381 [cs].

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., Shanahan, M., Langston, V., Pascanu, R., Botvinick, M., Vinyals, O., and Battaglia, P. Relational Deep Reinforcement Learning, June 2018. URL http://arxiv.org/abs/1806.01830. arXiv:1806.01830 [cs, stat].

Zeng, Y., Cai, R., Sun, F., Huang, L., and Hao, Z. A Survey on Causal Reinforcement Learning, February 2023. URL http://arxiv.org/abs/2302.05209. arXiv:2302.05209 [cs].

Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., and Precup, D. Invariant Causal Prediction for Block MDPs. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 11214–11224. PMLR, November 2020. URL https://proceedings.mlr.press/v119/zhang20t.html. ISSN: 2640-3498.

Zhang, K., Peters, J., Janzing, D., and Schoelkopf, B. Kernel-based Conditional Independence Test and Application in Causal Discovery, February 2012. URL http://arxiv.org/abs/1202.3775. arXiv:1202.3775 [cs, stat].

Zhou, A., Kumar, V., Finn, C., and Rajeswaran, A. Policy Architectures for Compositional Generalization in Control, March 2022. URL http://arxiv.org/abs/2203.05960. arXiv:2203.05960 [cs].

Zhu, G., Huang, Z., and Zhang, C. Object-Oriented Dynamics Predictor, October 2018. URL http://arxiv.org/abs/1806.07371. arXiv:1806.07371 [cs].

Zhu, G., Wang, J., Ren, Z., Lin, Z., and Zhang, C. Object-Oriented Dynamics Learning through Multi-Level Abstraction, December 2019. URL http://arxiv.org/abs/1904.07482. arXiv:1904.07482 [cs, stat].

Zhu, Z.-M., Chen, X.-H., Tian, H.-L., Zhang, K., and Yu, Y. Offline Reinforcement Learning with Causal Structured World Models, June 2022. URL http://arxiv.org/abs/2206.01474. arXiv:2206.01474 [cs, stat].

Table 6. Symbols used in the paper and appendices

Symbol(s)	Explanation
S_i	The i-th state variable in a FMDP.
S'_i	The i-th next-state variable in a FMDP.
$\mathbf{S}$	The group of state variables in a FMDP.
$\mathbf{S}'$	The group of next-state variables in a FMDP.
A_i	The i-th action variable in a FMDP.
$\mathbf{A}$	The group of action variables in a FMDP.
$\mathbf{\Delta}$	The group of all variables in a transition, i.e. $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$.
n_s	The number of state variables in a FMDP.
n_a	The number of action variables in a FMDP.
p	The probability distribution of random variables.
$\hat{p}$	The estimated distribution for p in a dynamics model.
$\mathcal{G}$	The DAG of a causal model (e.g., a Bayesian network, CDM, or OOCDM).
$\mathrm{X} \to \mathrm{Y}$	Variable X is a parent of variable Y in some given DAG.
$Pa_{\mathcal{G}}(\mathrm{X})$	The parent set of variable X in DAG $\mathcal{G}$.
$Pa(\mathrm{X})$	The parent set of variable X in the ground-truth causal graph.
C (or C_i)	A (or the i-th) class in an OOMDP.
$\mathcal{C}$	The set of classes in an OOMDP.
$\mathcal{F}[C]$	The set of fields of class C, i.e. $\mathcal{F}_s[C] \cup \mathcal{F}_a[C]$.
$\mathcal{F}_s[C]$	The set of state fields of class C.
$\mathcal{F}_a[C]$	The set of action fields of class C.
$\mathcal{F}$	The set of all fields in an OOMDP, i.e. $\bigcup_{C\in\mathcal{C}} \mathcal{F}[C]$.
$\mathcal{F}_s$	The set of all state fields in an OOMDP, i.e. $\bigcup_{C\in\mathcal{C}} \mathcal{F}_s[C]$.
$C.U$	Some filed of C in $\mathcal{F}[C]$.
$C.V$	Some state field of C in $\mathcal{F}_s[C]$.
$Dom_{C.U}$	The domain of some field $C.U$.
O, O_i	An object in an OOMDP.
N	The number of objects in an OOMDP.
K	The number of classes in an OOMDP.
$O \in C$	Object O is an instance of class C.
$O.\mathrm{U}$	An attribute of O (derived from the field $C.U \in \mathcal{F}[C]$ where $O \in C$).
$O.\mathrm{V}$	A state attribute of O (derived from the field $C.\mathrm{S} \in \mathcal{F}_s[C]$ where $O \in C$).
$O.\mathbf{S}$	The group of all state attributes of O.
$O.\mathbf{A}$	The group of all action attributes of O.
$\mathbf{O}$	All attributes of O, i.e. $(O.\mathbf{S}, O.\mathbf{A})$.
$O.\mathrm{V}'$	The variable of state attribute $O.\mathrm{V}$ in the next-step.
$O.\mathbf{S}'$	The group of state variables $O.\mathbf{S}$ in the next-step.
$O_a \sim O_b$	O_a and O_b are instances of the same class.
$C.U \to V'$	A local causality expression from $C.U$ to $C.V$.
$C_l.U \to C_k.V'$	A global causality expression from $C_l.U$ to $C_k.V$.
$\mathcal{D}$	A dataset of transition samples.
C_k^t	The set of instances of class C_k at step t.
$\hat{p}(\cdot)_t$	The estimation of p when variables take the observed values at step t.
$\mathcal{I}_{\mathcal{D}}^{\varsigma}$	The CMI for class-level causality ς on data $\mathcal{D}$.
$f_{C.V}$	The predictor for the state field $C.V$ in the OOCDM.
$\mathcal{L}_{\mathcal{G}}(\mathcal{D})$	The AILL function of data $\mathcal{D}$ under the CG $\mathcal{G}$.
$J(\mathcal{D})$	The overall target function for model learning.

A. Acronyms and Symbols

The meanings of symbols used in the paper or will be used in the appendices are described in Table 6 unless otherwise specified. The acronyms that appear in our paper are explained in Table 7.

Table 7. The meanings of acronyms that appear in the paper.

Acronym	Explanation
AILL	Average instance log-likelihood.
BCG	Bipartite causal graph.
BN	Bayesian Network.
CDM	Causal dynamics model.
CDL	A baseline proposed by Wang et al. (2022).
CG	Causal graph.
CIT	Conditional independence test.
CLCE	Class-level causality expression.
CMS	Collect-Mineral-Shards, a StarCraftII minigame.
CMI	Conditional mutual information.
DAG	Directed acyclic graph.
DBN	Dynamics Bayesian Network.
DZB	Defeat-Zerglings-Banelings, a StarCraftII minigame.
GNN	A baseline based on a graph neural network (Kipf et al., 2020).
i.d.	In-distribution.
MBRL	Model-based reinforcement learning.
MDP	Markov decision process.
MLP	Multi-layer perceptron.
OO	Object-oriented.
OOCDM	Object-oriented causal dynamics model
OOCG	Object-oriented causal graph.
o.o.d.	Out-of-distribution.
OOFULL	Object-oriented full model (a variant of OOCDM that uses full OOCGs).
OOMDP	Object-oriented Markov decision process.
RL	Reinforcement learning.
TICSA	A baseline proposed by Wang et al. (2021).

B. Basics of Causality

B.1. Causal Models

In this section, we present some of the basic concepts and theorems of causality, which form the foundation of our theory. We first introduce Markov Compatibility (Pearl, 2000), which defines whether a graph can correctly reflect the relationships among variables given a probability function.

Definition B.1 (Markov Compatibility)**.** Assume $\mathcal{G}$ is an directional acyclic graph (DAG) on a group of random variables $\mathbf{X} = (\mathrm{X}_1, ..., \mathrm{X}_n)$. Given any probability function p of these variables, if the *rule of production decomposition* holds:

$$p(\mathrm{X}_1, ..., \mathrm{X}_n) = \prod_{j=1}^{n} p(\mathrm{X}_j | Pa_{\mathcal{G}}(\mathrm{X}_j)), \tag{11}$$

then we say that p is *compatible* with $\mathcal{G}$, or that $\mathcal{G}$ *represents* p.

Causality (the DAG) is a universal concept. The following theorem shows, that no matter what the probability function is, the dependencies between variables can always be represented by some DAG. This leads to a general form of a causal model called the Bayesian Network (BN).

Theorem B.2 (Existence of causal graphs)**.** *For any probability function p of variables $\mathbf{X} = (\mathrm{X}_1, \cdots, \mathrm{X}_n)$, there always exists a DAG $\mathcal{G}$ that p is compatible with.*

Proof. Using the chain rule of probability functions, we have

$$p(\mathrm{X}_1, ..., \mathrm{X}_n) = p(\mathrm{X}_1) p(\mathrm{X}_2|\mathrm{X}_1) p(\mathrm{X}_3|\mathrm{X}_1, \mathrm{X}_2) \cdots p(\mathrm{X}_n|\mathrm{X}_1, ..., \mathrm{X}_{n-1}). \tag{12}$$

Letting $Pa(\mathrm{X}_j) \subseteq \{\mathrm{X}_1, ..., \mathrm{X}_{j-1}\}$ denote the minimal subset such that $p(\mathrm{X}_j|\mathrm{X}_1, ..., \mathrm{X}_{j-1}) = p(\mathrm{X}_j|Pa(\mathrm{X}_j))$ (the Markovian parents (Pearl, 2000)) for $j = 1, ..., n$, we obtain Eq. 11. □

Definition B.3 (Bayesian Network)**.** A *Bayesian Netowrk* is a tuple $\langle\mathcal{G}, p\rangle$, where $\mathcal{G}$ is a DAG on a set of random variables $\mathbf{X} = (\mathrm{X}_1, ..., \mathrm{X}_n)$, and p is a probability function of $\mathbf{X}$ such that p is compatible with $\mathcal{G}$.

Especially, according to Laplacian's conception, most stochastic phenomenons in nature are due to deterministic functions combined with unobserved disturbances. This conception leads to a special type of BN called the Structural Causal Model (SCM), which is the most popular model in causal inference.

Definition B.4 (Structural Causal Model)**.** A *Structural Causal Model* is a tuple $\langle\mathcal{G}, p, \mathbf{U}, \mathcal{F}\rangle$, where $\langle\mathcal{G}, p\rangle$ forms a Bayesian Network on variables $\mathbf{X} = (\mathrm{X}_1, ..., \mathrm{X}_n)$. $\mathbf{U} = (\mathrm{U}_1, ..., \mathrm{U}_n)$ is a set of disturbance variables that are independent of each other. $\mathcal{F} = \{f_1, f_2, \cdots, f_n\}$ is a set of structural equations, such that

$$\mathrm{X}_i = f_i(Pa(\mathrm{X}_i); \mathrm{U}_i). \tag{13}$$

B.2. D-Separation

The concept of *d-seperation* plays an important role in causal inference. Given a DAG $\mathcal{G}$, the criterion of d-separation provides an effective way to determine on what condition two groups of variables are independent.

Definition B.5 (d-separation)**.** Assume $\mathcal{G}$ is a DAG on a set of variables $\mathbf{V}$. Assume $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are three disjoint groups of variables in $\mathbf{V}$. We say an un-directional path between $\mathbf{X}$ and $\mathbf{Y}$ is *blocked* by $\mathbf{Z}$ if one of the following requirements is met: 1) The path contains a chain $\mathrm{A} \to \mathrm{B} \to \mathrm{C}$ or a fork $\mathrm{A} \leftarrow \mathrm{B} \to \mathrm{C}$ such that $\mathrm{B} \in \mathbf{Z}$; or 2) the path contains a collider $\mathrm{A} \to \mathrm{B} \leftarrow \mathrm{C}$ such that $\mathbf{Z}$ contains no descendent of B. We say $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated* by $\mathbf{Z}$, if $\mathbf{Z}$ blocks all un-directional paths between $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, denoted as

$$\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z}. \tag{14}$$

Theorem B.6 (d-separation criterion)**.** *Assume $\mathcal{G}$ is a DAG on a set of variables $\mathbf{V}$. Assume $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are three disjoint groups of variables in $\mathbf{V}$. We have:*
1) if p is any probability function compatible with $\mathcal{G}$, then

$$(\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z}) \Rightarrow (\mathbf{X} \perp\!\!\!\perp_{p} \mathbf{Y} \mid \mathbf{Z}), \tag{15}$$

where $\perp\!\!\!\perp_p$ means conditional independence under p, namely $p(\mathbf{Y}|\mathbf{Z}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{Z})$;
2) if $(\mathbf{X} \perp\!\!\!\perp_{p} \mathbf{Y} \mid \mathbf{Z})$ holds for all p that is compatible with $\mathcal{G}$, then $(\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z})$ also holds.

Using the d-separation criterion, the following rule is proven by Pearl (2000).

Theorem B.7 (Causal Markov Condition)**.** *Assume $\mathcal{G}$ is a DAG on a set of variables $\mathbf{V}$. Let p denote a probability function for these variables. Then p is compatible with $\mathcal{G}$ if and only if $(\mathrm{X} \perp\!\!\!\perp_p \mathrm{Y}|Pa_{\mathcal{G}}(X))$ holds for any $\mathrm{X}, \mathrm{Y} \in \mathbf{V}$ such that Y is not a descendant of X.*

B.3. Causal Discovery

Consider that $\mathbf{V}$ is a set of variables, and that p is a probability function of these variables. The goal of causal discovery is to recover a DAG $\mathcal{G}$ that is compatible with p from a set of observation data (sampled from p) of these variables. However, a probability function p may be compatible with more than one DAG. For example, consider two SCMs on variables $\{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ where X is the only exogenous variable:

$$\mathcal{M}_1 : \mathrm{X} = \mathrm{U}_{\mathrm{X}},\ \mathrm{Y} = \mathrm{X}^2 + \mathrm{U}_{\mathrm{Y}},\ \mathrm{Z} = \mathrm{X} + \mathrm{X}^2 + \mathrm{U}_{\mathrm{Z}}; \tag{16}$$

$$\mathcal{M}_2 : \mathrm{X} = \mathrm{U}_{\mathrm{X}},\ \mathrm{Y} = \mathrm{X}^2 + \mathrm{U}_{\mathrm{Y}},\ \mathrm{Z} = \mathrm{X} + \mathrm{Y} + \mathrm{U}_{\mathrm{Z}}. \tag{17}$$

If the distributions of disturbances are the same in both SCMs and $\mathrm{U}_Y \equiv 0$, then the two SCMs lead to identical probability functions. Therefore, this probability function is compatible with two different DAGs: In $\mathcal{M}_1$, we have $Pa(\mathrm{Z}) = \{\mathrm{X}\}$; in $\mathcal{M}_2$, we have $Pa(\mathrm{Z}) = \{\mathrm{X}, \mathrm{Y}\}$.

Since there exists more than one DAG that p may be compatible with, Definition B.9 suggests that we may look for the minimal DAG that can represent the fewest probability functions, i.e. the DAG that focuses most on p. It is worth mentioning that in the original definitions of Pearl (2000), the observability of variables is considered, which is ignored here since all variables are observable in our work.

Definition B.8 (Structural preference and equivalence)**.** Assume $\mathcal{G}_1$ and $\mathcal{G}_2$ are DAGs on the same set of variables. If any probability function p compatible with $\mathcal{G}_1$ is also compatible with $\mathcal{G}_2$, we say $\mathcal{G}$ is preferred to $\mathcal{G}_2$, denoted as $\mathcal{G}_1 \preceq \mathcal{G}_2$. If we have $\mathcal{G}_1 \preceq \mathcal{G}_2$ and $\mathcal{G}_2 \preceq \mathcal{G}_1$, we say $\mathcal{G}_1$ and $\mathcal{G}_2$ are equivalent, denoted as $\mathcal{G}_1 \equiv \mathcal{G}_2$.

Definition B.9 (Minimal structure)**.** Assume $\boldsymbol{G}$ is a family of DAGs defined on the same set of variables. We say $\mathcal{G} \in \boldsymbol{G}$ is the minimal DAG among $\boldsymbol{G}$ if every $\mathcal{G}' \in \boldsymbol{G}$ satisfies that $\mathcal{G} \preceq \mathcal{G}'$.

Faithfulness (also known as stability) is an important concept for causal discovery. We say p is faithful to a DAG $\mathcal{G}$ if all the conditional independence relationships in p are "stored" in the structure of $\mathcal{G}$. In other words, the independent relationships in p stem purely from the causal structure $\mathcal{G}$ rather than coincidence. In addition, faithfulness offers a stronger condition than minimality, as it implies a unique minimal structure. Therefore, the faithfulness condition becomes a vital assumption for causal discovery, which makes the structure of the DAG identifiable. If p is faithful to $\mathcal{G}$, then $\mathcal{G}$ precludes all spurious correlations. By assuming that the probability p of data follows a stable distribution, we can use the Causal Faithfulness Property (Theorem B.11) to identify the CG $\mathcal{G}$ that p is compatible with and faithful to.

Definition B.10 (Faithfulness and stable distribution)**.** Assume $\mathcal{G}$ is a DAG and probability function p is compatible with $\mathcal{G}$. If we have

$$(\mathbf{X} \perp\!\!\!\perp_p \mathbf{Y}|\mathbf{Z}) \Rightarrow (\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y}|\mathbf{Z}) \tag{18}$$

for any disjoint variable groups $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, we say p is *faithful* to the DAG $\mathcal{G}$. Consider p a probability function of a set of variables. If there exists a DAG $\mathcal{G}$ on these variables such that p is compatible with and faithful to $\mathcal{G}$, we say p follows a *stable distribution*.

Theorem B.11 (Causal faithfulness property)**.** *Assume $\mathcal{G}$ is a DAG. If a probability function p is compatible with and faithful to $\mathcal{G}$, we have*

$$(\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z}) \Leftrightarrow (\mathbf{X} \perp\!\!\!\perp_p \mathbf{Y} \mid \mathbf{Z}) \tag{19}$$

for any disjoint variable groups $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ in $\mathcal{G}$.

The above Theorem B.11 can be easily derived from Theorem B.6. The following theorem (Peters et al., 2017) shows that faithfulness is a stronger requirement than minimality.

Theorem B.12 (Faithfulness implicates an minimal structure)**.** *Assume p is a probability function of a set of variables and $\boldsymbol{G}$ is the set of DAGs that p is compatible with. If p is faithful to $\mathcal{G} \in \boldsymbol{G}$, then $\mathcal{G}$ is a minimal DAG in $\boldsymbol{G}$.*

B.4. Conditional Independence Tests

According to Theorem B.11, the discovery of the graph structure is converted into the determination of the conditional independence relations under a faithful probability p. However, the exact formulation of p is usually unknown, and we have to make the judgment using samples drawn from p.

The technique for testing whether variables are conditionally independent is called the Conditional Independence Test (CIT). In other words, CIT uses a dataset $\{(x_i, y_i, z_i)\}_{i=1}^N$ drawn from p to estimate whether the hypothesis $(\mathrm{X} \perp\!\!\!\perp_p \mathrm{Y} \mid \mathrm{Z})$ holds. A simple way to implement a CIT is to learn two linear regression models, $\hat{y} = f(x, z)$ and $\hat{y} = g(z)$, using the given data. We then define the square errors of both models:

$$\epsilon_f(x_i, y_i, z_i) = \frac{1}{N} \sum (y_i - f(x_i, z_i))^2, \tag{20}$$

$$\epsilon_g(x_i, y_i, z_i) = \frac{1}{N} \sum (y_i - g(z_i))^2. \tag{21}$$

If conditional independence holds, then $\mathbf{X}$ does not carry any information about $\mathbf{Y}$, and thus the argument x will not change the regression error. Therefore, a Student t-test can be used to check whether ϵ_f / ϵ_g is expected to be 1, which confirms the independence. Additionally, there are many more advanced tools to perform this test, such as Fast CIT (Chalupka et al., 2018) and Kernel-based CIT (Zhang et al., 2012).

Another way to perform the CIT is to estimate the conditional mutual information (CMI). The CMI between X and Y conditional on Z is defined as

$$\mathcal{I}(\mathrm{X}, \mathrm{Y} \mid \mathrm{Z}) := \mathbb{E}_{\mathrm{X,Y,Z}}\left[\log \frac{p(\mathrm{X}, \mathrm{Y}|\mathrm{Z})}{p(\mathrm{X}|\mathrm{Z})p(\mathrm{Y}|\mathrm{Z})}\right] = \mathbb{E}_{\mathrm{X,Y,Z}}\left[\log \frac{p(\mathrm{Y}|\mathrm{X}, \mathrm{Z})}{p(\mathrm{Y}|\mathrm{Z})}\right]. \tag{22}$$

Theorem B.13. $\mathcal{I}(\mathrm{X}, \mathrm{Y} \mid \mathrm{Z}) \geq 0$, *where equality holds if and only if* $(\mathrm{X} \perp\!\!\!\perp_p \mathrm{Y} \mid \mathrm{Z})$.

Using the theory above, we can determine whether conditional independence holds by checking whether the CMI is 0. As suggested by Wang et al. (2022), we can estimate the CMI using the neural approximates of $p(\mathrm{Y}|\mathrm{X},\mathrm{Z})$ and $p(\mathrm{Y}|\mathrm{Z})$.

C. The Theory of Causal Dynamics Models

We assume the state in a Factored Markov Decision Process (FMDP) is composed of n_s state variables written as $\mathbf{S} = (\mathrm{S}_1, \cdots, \mathrm{S}_{n_s})$. Similarly, we have $\mathbf{A} = (\mathrm{A}_1, \cdots, \mathrm{A}_{n_a})$. In this section, we show 1) that there always exists a causal dynamics model (a.k.a., dynamics Bayesian network) to formulate the dynamics of an FMDP, 2) that this causal dynamics model has bipartite causal graph if state variables transit independently, and 3) how the ground-truth causal graph of an FMDP can be uniquely identified. The following discussion is based on the general form of FMDPs. However, the definitions, analysis, and conclusion are also applicable in OOMDPs, where the attributes are merely variables organized in an OO framework.

C.1. Causal Structure of Factored Markov Decision Process

The following theorem describes the general causal structure for in transition $\mathbf{\Delta}$ of an FMDP. We first define the concept of consistency, which means that the probability function of $\mathbf{\Delta}$ follows the transition function of the FMDP whereas the state distribution $p(\mathbf{S})$ and policy $p(\mathbf{A}|\mathbf{S})$ can be arbitrary.

Definition C.1 (Consistent Probability Function)**.** Assume $\mathbf{\Delta} = (\mathbf{S}, \mathbf{A}, \mathbf{S}')$ is the set of transition variables of an FMDP. Suppose that p is a probability function of variables $\mathbf{\Delta}$. We say it is *consistent* with the dynamics, if

$$p(\mathbf{S}, \mathbf{A}, \mathbf{S}') = p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) p(\mathbf{A}|\mathbf{S}) p(\mathbf{S}), \tag{23}$$

and $p(\mathbf{S}'|\mathbf{S}, \mathbf{A})$ is exactly the transition function of the concerned FMDP.

Theorem C.2. *Assume* p *is any probability function of a variables* $\mathbf{\Delta} = (\mathbf{S}, \mathbf{A}, \mathbf{S}')$. *If it is consistent with an FMDP, then there exists a DAG* $\mathcal{G}$ *on* $\mathbf{\Delta}$ *such that:*

1. p *is compatible with* $\mathcal{G}$*;*
2. $Pa(\mathrm{S}_i) \subseteq \mathbf{S}$ *for every* $\mathrm{S}_i \in \mathbf{S}$*;*
3. $Pa(\mathrm{A}_i) \subseteq (\mathbf{S}, \mathbf{A})$ *for every* $\mathrm{A}_i \in \mathbf{A}$
4. $Pa(\mathrm{S}'_j) \subseteq (\mathbf{S}, \mathbf{A}, \mathbf{S}')$ *for every* $\mathrm{S}'_j \in \mathbf{S}'$*;*
5. $\mathcal{G}$ *contains no backward edge like* $\mathrm{S}'_j \to \mathrm{S}_i$ *or* $\mathrm{A}_j \to \mathrm{S}_i$.

Proof. We have

$$p(\mathbf{\Delta}) = p(\mathbf{S}) p(\mathbf{A}|\mathbf{S}) p(\mathbf{S}'|\mathbf{S}, \mathbf{A}). \tag{24}$$

It is easy to see that $Pa(\mathrm{A}_i) \subseteq \mathbf{S}$ for every $\mathrm{A}_i \in \mathbf{A}$ and $Pa(\mathrm{S}'_j) \in (\mathbf{S}, \mathbf{A})$ for every $\mathrm{S}'_j \in \mathbf{S}'$ if we decompose the probabilities using the chain rule. For $p(\mathbf{S}'|\mathbf{S}, \mathbf{A})$, we can write

$$p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}_j|\mathbf{S}, \mathbf{A}, \mathrm{S}'_1, ..., \mathrm{S}'_{j-1}), \tag{25}$$

and define $Pa(\mathrm{S}'_j) \subseteq (\mathbf{S}, \mathbf{A}, \mathrm{S}_1, ..., \mathrm{S}_{j-1})$ as the minimal subset such that $p(\mathrm{S}'_j|\mathbf{S}, \mathbf{A}, \mathrm{S}'_1, ..., \mathrm{S}'_{j-1}) = p(\mathrm{S}'_j|Pa(\mathrm{S}'_j))$. For $p(\mathbf{S})$ and $p(\mathbf{A}|\mathbf{S})$, we can perform similar decomposition. Therefore, the above conclusions are easy to draw. □

The definition of CDMs has been provided in the paper's Definition 3.1. From the above proof, we can see that the parenthood of next-state variables $\mathbf{S}'$ is not affected by the choice of policy $p(\mathbf{A}|\mathbf{S})$ and the prior distribution of state $p(\mathbf{S})$. Therefore, Causal Dynamics Models (CDMs) only care about the causality of $\mathbf{S}'$. Like Definition B.1, we define whether the causal graph can represent the dynamics of an FMDP using the product decomposition.

Definition C.3 (Represented dynamics)**.** Assume $\mathcal{M} = \langle \mathcal{G}, p \rangle$ is a CDM for an FMDP. If $\mathcal{G}$ satisfies that

$$p^*(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = \prod_{j=1}^{n_s} p^*(\mathrm{S}'_j|Pa(\mathrm{S}'_j)) \tag{26}$$

for every probability function p^* of $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$ that is consistent with the FMDP, then we say $\mathcal{G}$ *represents* the FMDP's dynamics. Further, if we also have that $p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = p^*(\mathbf{S}'|\mathbf{S}, \mathbf{A})$ for every consistent probability function p^*, we say the CDM $\mathcal{M}$ *matches* the dynamics of the FMDP.

It is important to note that a CDM is **not** a causal model (Bayesian network). It does not specify the causality of $\mathbf{S}$ and $\mathbf{A}$ but focuses on the causality of the next state $\mathbf{S}'$. In other words, a CDM hopes to capture the universal rule of transitions, no matter what policy the agent uses, how each episode begins, and how each episode terminates. It is concretized to a real causal model when the policy $p(\mathbf{A}|\mathbf{S})$ and the state distribution $p(\mathbf{S})$ are given.

Theorem C.4 (Concretization)**.** *Assume $\mathcal{G}$ is a causal graph that represents the dynamics of an FMDP. Then for every probability function p consistent with the FMDP's dynamics, there exists a DAG $\mathcal{G}_p$ on $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$ such that 1) $\mathcal{G}_p$ satisfies all propositions in Theorem C.2, 2) $\mathcal{G}$ is a sub-graph of $\mathcal{G}_p$, and 3) $\mathcal{G}_p$ and $\mathcal{G}$ share the same parent set $Pa(\mathrm{S}'_j)$ for each next-sate variable $\mathrm{S}'_j \in \mathbf{S}'$. We call this DAG G_P as a concretization of $\mathcal{G}$ under p.*

Proof. Because $\mathcal{G}$ represents the dynamics of an FMDP, we have

$$p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j | Pa(\mathrm{S}'_j)).$$

Now, we use $Pa_{\mathcal{G}_p}(\mathrm{X})$ to denote the parent set of variable X in $\mathcal{G}_p$. Since p is consistent with the FMDP, we have

$$p(\mathbf{S}, \mathbf{A}, \mathbf{S}') = p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) p(\mathbf{A}|\mathbf{S}) p(\mathbf{S}).$$

Using the chain rule to decompose $p(\mathbf{A}|\mathbf{S})$ and $p(\mathbf{S})$, we have

$$p(\mathbf{S}, \mathbf{A}, \mathbf{S}') = p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) \prod_{j=1}^{n_a} p(\mathrm{A}_j | \mathbf{S}, \mathrm{A}_1, ..., \mathrm{A}_{j-1}) \prod_{i=1}^{n_s} p(\mathrm{S}_i | \mathrm{S}_1, ..., \mathrm{S}_{i-1}).$$

Then there exists $\mathcal{G}_p$ where $Pa_{\mathcal{G}_p}(\mathrm{A}_j) \subseteq (\mathbf{S}, \mathrm{A}_1, ..., \mathrm{A}_{j-1})$, $Pa_{\mathcal{G}_p}(\mathrm{S}_i) \subseteq \{\mathrm{S}_1, ..., \mathrm{S}_{i-1}\}$, and $Pa_{\mathcal{G}_p}(\mathrm{S}'_k) = Pa(\mathrm{S}'_k)$, such that

$$\begin{aligned} p(\mathbf{S}, \mathbf{A}, \mathbf{S}') &= p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) \prod_{j=1}^{n_a} p(\mathrm{A}_j | Pa_{\mathcal{G}_p}(\mathrm{A}_j)) \prod_{i=1}^{n_s} p(\mathrm{S}_i | Pa_{\mathcal{G}_p}(\mathrm{S}_i)) \\ &= \prod_{k=1}^{n_s} p(\mathrm{S}'_k | Pa_{\mathcal{G}_p}(\mathrm{S}'_k)) \prod_{j=1}^{n_a} p(\mathrm{A}_j | Pa_{\mathcal{G}_p}(\mathrm{A}_j)) \prod_{i=1}^{n_s} p(\mathrm{S}_i | Pa_{\mathcal{G}_p}(\mathrm{S}_i)) \\ &= \prod_{k=1}^{n_s} p(\mathrm{S}'_k | Pa(\mathrm{S}'_k)) \prod_{j=1}^{n_a} p(\mathrm{A}_j | Pa_{\mathcal{G}_p}(\mathrm{A}_j)) \prod_{i=1}^{n_s} p(\mathrm{S}_i | Pa_{\mathcal{G}_p}(\mathrm{S}_i)). \end{aligned}$$

Then the theorem is proven with the above equations. □

There may exist more than one causal graph that represents the dynamics of the dynamics. However, not all these graphs are "good" as they may contain redundant edges. In order to remove spurious correlations and improve the generalization of dynamics models, we want the CG to capture as many independent relationships as possible. Most importantly, these independent relationships should be universal. That is, they hold for every other probability function that is consistent with the dynamics so that they will not be destroyed if the agent changes its policy or we change the distribution of states. Therefore, the desired property of the causal graph is given in the following definition.

Definition C.5 (Dynamical faithfulness)**.** Assume $\mathcal{G}$ is a causal graph of a CDM and p is a probability function of $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$. We say p is *dynamically faithful* to $\mathcal{G}$, if there exists a DAG $\mathcal{G}_*$ such that 1) p is compatible with and faithful to $\mathcal{G}_*$, and 2) $\mathcal{G}_*$ is a concretization of $\mathcal{G}$ under p.

Definition C.6 (Ground-truth causal graph)**.** Assume $\mathcal{G}$ is a causal graph of a CDM for an FMDP. We say $\mathcal{G}$ a *ground-truth* causal graph of the FMDP's dynamics if it is the unique causal graph inferred from all dynamically faithful probability functions. That is, for every consistent probability function p of $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$ and any causal graph $\mathcal{G}'$, we have

$$p \text{ is dynamically faithful to } \mathcal{G}' \implies \mathcal{G}' = \mathcal{G}.$$

Theorem C.7. *A necessary and sufficient condition for a CG $\mathcal{G}$ to be the ground-truth causal graph of the FMDP dynamics is that, for every consistent probability function p and any DAG $\mathcal{G}_*$ on $(\mathbf{S}, \mathbf{A}, \mathbf{S}')$, we have*

$$p \textit{ is compatible with and faithful to } \mathcal{G}_* \implies \mathcal{G}_* \textit{ is a concretization of } \mathcal{G} \textit{ under } p.$$

Proof. (Necessity) Let $\mathcal{G}'$ denotes a sub-graph of $\mathcal{G}_*$ such that $\mathcal{G}_*$ is a concretization of $\mathcal{G}'$ under p. We have that

$$\begin{aligned}
&p \text{ is compatible with and faithful to } \mathcal{G}_* \\
\Longrightarrow &p \text{ is dynamically faithful to } \mathcal{G}' \\
\Longrightarrow &\mathcal{G}' = \mathcal{G} \\
\Longrightarrow &\mathcal{G}_* \text{ is a concretization of } \mathcal{G} \text{ under } p.
\end{aligned}$$

(Sufficiency) Let $\mathcal{G}'_p$ be some concretization of $\mathcal{G}'$ under p. We have that

$$\begin{aligned}
&p \text{ is dynamically faithful to } \mathcal{G}' \\
\Longrightarrow &\exists \mathcal{G}'_p \text{ which } p \text{ is compatible with and faithful to} \\
\Longrightarrow &\exists \mathcal{G}'_p \text{ is a concretization of } \mathcal{G} \\
\Longrightarrow &\mathcal{G} = \mathcal{G}'.
\end{aligned}$$

□

C.2. Bipartite Causal Graphs

Not all MDPs are suitable to be modeled by causality. For example, if the state variables are raw pixels of an image, then transitions of variables are densely correlated, leading to a dense causal graph. In this case, CDMs are deprecated, unless certain abstraction and representation techniques are performed to simplify the causal structure (not included in our work). We will make decent assumptions about the FMDP, which greatly simplifies the structure of the causal graph.

Assumption C.8 (Independent transition)**.** The transition function of the FMDP follows that

$$p(\mathbf{S}'|\mathbf{S},\mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j|\mathbf{S},\mathbf{A}). \tag{27}$$

Several studies have assumed that the causal graph of a CDM is bipartite (Volodin, 2021; Wang et al., 2021; 2022; Ding et al., 2022). We formally define a bipartite causal graph (BCG) below. If the transition is independent (Assumption C.8), we argue that: 1) we can use BCGs as they always exist, and 2) we should use BCGs as they are necessary for faithfulness.

Definition C.9 (Bipartite causal graph)**.** Consider that $\mathcal{G}$ is the CG of a CDM. If we have $Pa(\mathrm{S}'_j) \subseteq (\mathbf{S},\mathbf{A})$ for every $\mathrm{S}'_j \in \mathbf{S}$, we say $\mathcal{G}$ is a *bipartite causal graph* (BCG). In other words, no lateral edge like $\mathrm{S}'_i \to \mathrm{S}'_j$ exists among $\mathbf{S}'$.

Theorem C.10 (Existence of BCGs)**.** *If an FMDP follows Assumption C.8 then it is matched by some CDM whose causal graph is a BCG. In addition, in this BCG we have*

$$p(\mathrm{S}'_j|Pa(\mathrm{S}'_j)) = p(\mathrm{S}'_j|\mathbf{S},\mathbf{A}), \quad \text{for } \mathrm{S}'_j \in \mathbf{S}'. \tag{28}$$

Proof. Assuming p gives the transition function of the SCM. We can define the CDM as $\langle \mathcal{G}, p\rangle$. In $\mathcal{G}$, we let $Pa(\mathrm{S}'_j)$ be any subset of $(\mathbf{S},\mathbf{A})$ such that $p(\mathrm{S}'_j|Pa(\mathrm{S}'_j)) = p(\mathrm{S}'_j|\mathbf{S},\mathbf{A})$. Such a subset always exists since it may directly be $(\mathbf{S},\mathbf{A})$. Using Assumption C.8, we have

$$p(\mathbf{S}'|\mathbf{S},\mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j|\mathbf{S},\mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j|Pa(\mathrm{S}'_j)).$$

Then, we let $p(\mathrm{S}'_j|Pa(\mathrm{S}'_j))$ be equal to $p(\mathrm{S}'_j|Pa(\mathrm{S}'_j))$. As a result, the dynamics are matched by the CDM and $\mathcal{G}$ is a BCG. □

Theorem C.11 (Faithfulness for BCGs)**.** *Assume that the dynamics of an FMDP are represented by $\mathcal{G}$ and p is a probability function consistent with the FMDP. If Assumption C.8 holds, then a necessary condition of that p is dynamically faithful to $\mathcal{G}$ (see Definition C.5) is that $\mathcal{G}$ is a BCG.*

Proof. Assume $\mathcal{G}_p$ is the concretization of $\mathcal{G}$ under p such that p is faithful to $\mathcal{G}$. If $\mathcal{G}$ is not bipartite, there exist j,k such that $\mathrm{S}'_j \to \mathrm{S}'_k$ in $\mathcal{G}_p$. In this case, we have $(\mathrm{S}'_j \not\perp\!\!\!\perp_{\mathcal{G}} \mathrm{S}'_k|\mathbf{S})$. According to Assumption C.8, we have $(\mathrm{S}'_j \perp\!\!\!\perp_p \mathrm{S}'_k|\mathbf{S})$. Therefore, we have that

$$(\mathrm{S}'_j \perp\!\!\!\perp_p \mathrm{S}'_k|\mathbf{S}) \not\Rightarrow (\mathrm{S}'_j \perp\!\!\!\perp_{\mathcal{G}_p} \mathrm{S}'_k|\mathbf{S}).$$

That is, p is not faithful to $\mathcal{G}_p$. Using reduction to absurdity, we prove that $\mathcal{G}$ is a BCG. □

Humans decompose the world into components based on independence. Therefore, it is rational to assume that state variables transit independently (Assumption C.8), which brings many benefits: 1) The ground-truth causal graph is a BCG so that the complexity of causal discovery is reduced; 2) The ground-truth causal graph can be uniquely identified by conditional independent tests, and 3) The computation of CDM can be implemented in parallel using GPUs.

Instead of BCGs, we note that there exists research that considers learning arbitrary CGs for CDMs (Zhu et al., 2022), where the requirement of independent transition can be released. However, this kind of CDM can not be computed in parallel, and the procedure of causal discovery is much more complicated. Learning CGs is already very expensive even though we consider only BCGs. Therefore, we suggest that Assumption C.8 is vital to make causal discovery applicable in large-scale environments.

C.3. Causal Discovery for Causal Dynamics Models

The approach to identifying the CG representing the dynamics of the FMDP is already introduced in the paper's Theorem 3.2. However, the expression of the theorem is rather vague. Given the above definitions, we now rewrite the theorem in a more rigorous way.

Theorem C.12 (Causal Discovery for FMDPs)**.** *Consider that probability function p is consistent (see Definition C.1) with the dynamics of an FMDP, where Assumption C.8 holds. Then, there exists a causal graph $\mathcal{G}$ that represents the dynamics of the FMDP (see Definition C.3). Assuming that p is dynamically faithful to $\mathcal{G}$ (see Definition C.5), we have*

1. *$\mathcal{G}$ is a bipartite causal graph (see Definition C.9),*
2. *$\mathcal{G}$ is the ground-truth causal graph (see Definition C.6) of the dynamics, and*
3. *$\mathcal{G}$ is uniquely identified by the rule that*

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow (\mathrm{X}_i \not\perp\!\!\!\perp_P \mathrm{S}'_j \,|\, (\mathbf{S},\mathbf{A}) \setminus \{\mathrm{X}_i\}) \tag{29}$$

for every $\mathrm{X}_i \in (\mathbf{S},\mathbf{A})$ and every $\mathrm{S}'_j \in \mathbf{S}'$.

Proof. Since p is consistent with the FMDP, then the transition function is $p(\mathbf{S}'|\mathbf{S},\mathbf{A})$. Using the chain rule, we have

$$p(\mathbf{S}'|\mathbf{S},\mathbf{A}) = \prod_{j=1}^{n_s} p(\mathrm{S}'_j|\mathbf{S},\mathbf{A},\mathrm{S}'_1,...,\mathrm{S}'_{j-1}).$$

By defining $Pa(\mathrm{S}'_j) \subseteq (\mathbf{S},\mathbf{A},\mathrm{S}'_1,...,\mathrm{S}'_{j-1})$ as any subset such that

$$p(\mathrm{S}'_j|\mathbf{S},\mathbf{A},\mathrm{S}'_1,...,\mathrm{S}'_{j-1}) = p(\mathrm{S}'_j|Pa(\mathrm{S}'_j)).$$

we have that $\mathcal{G}$ represents the transition dynamics of the FMDP.

We use $\mathcal{G}_p$ to denote the concretization of $\mathcal{G}$ under p. According to Theorem C.4, p is compatible with $\mathcal{G}_p$. Having assumed that p is dynamically faithful to $\mathcal{G}$, we can further assume that p is also faithful to $\mathcal{G}_p$. According to Theorem B.11, we have

$$(\mathbf{X} \perp\!\!\!\perp_p \mathbf{Y}|\mathbf{Z}) \Leftrightarrow (\mathbf{X} \perp\!\!\!\perp_{\mathcal{G}_p} \mathbf{Y}|\mathbf{Z})$$

for any disjoint variable groups $\mathbf{X},\mathbf{Y},\mathbf{Z}$ in $(\mathbf{S},\mathbf{A},\mathbf{S}')$. In addition, we have that $\mathcal{G}$ is a BCG according to Theorem C.11.

Assume that X_i is a variable in $(\mathbf{S},\mathbf{A})$. According to the definition of d-separation, if $\mathrm{X}_i \in Pa(\mathrm{S}'_j)$, X_i and S'_j can not be d-separated by any group $\mathbf{Z}$ of variables such that $\mathrm{X}_i, \mathrm{S}'_j \notin \mathbf{Z}$. Letting $\mathbf{Z} = (\mathbf{S},\mathbf{A}) \setminus \{\mathrm{X}_i\}$, we have

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Rightarrow (\mathrm{X}_i \not\perp\!\!\!\perp_{\mathcal{G}_p} \mathrm{S}'_j|\mathbf{Z}).$$

Noticing that $\mathcal{G}$ is a BCG and $\mathcal{G}$ is a sub-graph of $\mathcal{G}_p$ (according to C.4), every path from X_i to S'_j in $\mathcal{G}_p$ is blocked by $\mathbf{Z}$ unless $\mathrm{X}_i \in Pa(\mathrm{S}'_j)$. Therefore, we have

$$\mathrm{X}_i \notin Pa(\mathrm{S}'_j) \Rightarrow (\mathrm{X}_i \perp\!\!\!\perp_{\mathcal{G}_p} \mathrm{S}'_j|\mathbf{Z}).$$

In other words, we have

$$(\mathrm{X}_i \not\perp\!\!\!\perp_{\mathcal{G}_p} \mathrm{S}'_j|\mathbf{Z}) \Rightarrow \mathrm{X}_i \in Pa(\mathrm{S}'_j).$$

Combing the above conclusions, we prove that

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow (\mathrm{X}_i \not\perp\!\!\!\perp_{\mathcal{G}_p} \mathrm{S}'_j | (\mathbf{S}, \mathbf{A}) \setminus \{\mathrm{X}_i\}) \Leftrightarrow (\mathrm{X}_i \not\perp\!\!\!\perp_P \mathrm{S}'_j | (\mathbf{S}, \mathbf{A}) \setminus \{\mathrm{X}_i\}).$$

Therefore, we have that the causal graph $\mathcal{G}$ representing the dynamics of the FMDP is uniquely identified using the above rule.

Now we consider replacing p with any other probability p^* such that p^* is faithful to some concretization $\mathcal{G}^*_{p^*}$. We use $Pa^*(\mathrm{S}'_j)$ to denote the parent set of $\mathrm{S}'_j \in \mathbf{S}'$ in $\mathcal{G}^*_{p^*}$.

Consider that X_i is a variable in $(\mathbf{S}, \mathbf{A})$ and define $\mathbf{Z} := (\mathbf{S}, \mathbf{A}) \setminus \{\mathrm{X}_i\}$. If $\mathrm{X}_i \in Pa(\mathrm{S}'_j)$ and $\mathrm{X}_i \notin Pa^*(\mathrm{S}'_j)$, using the above rule we have

$$(\mathrm{X}_i \not\perp\!\!\!\perp_p \mathrm{S}'_j | \mathbf{Z}) \wedge (\mathrm{X}_i \perp\!\!\!\perp_{p^*} \mathrm{S}'_j | \mathbf{Z}).$$

In other words, we have

$$\begin{aligned} p(\mathrm{S}'_j|\mathbf{Z}) &\neq p(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i), \\ p^*(\mathrm{S}'_j|\mathbf{Z}) &= p^*(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i). \end{aligned}$$

Noting that $(\mathbf{Z}, \mathrm{X}_i) = (\mathbf{S}, \mathbf{A})$, $p^*(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i)$ is identical to $p(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i)$ for every $\mathrm{S}'_j \in \mathbf{S}'$ as they are both given by the transition function of the FMDP. This leads to that

$$p(\mathrm{S}'_j|\mathbf{Z}) \neq p^*(\mathrm{S}'_j|\mathbf{Z}).$$

However, we can also write that

$$\begin{aligned} p(\mathrm{S}'_j|\mathbf{Z}) &= \int_x p(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i = x) p(\mathrm{X}_i = x|\mathbf{Z}) \\ &= \int_x p^*(\mathrm{S}'_j|\mathbf{Z}, \mathrm{X}_i = x) p(\mathrm{X}_i = x|\mathbf{Z}) \\ &= \int_x p^*(\mathrm{S}'_j|\mathbf{Z}) p(\mathrm{X}_i = x|\mathbf{Z}) \\ &= p^*(\mathrm{S}'_j|\mathbf{Z}) \int_x p(\mathrm{X}_i = x|\mathbf{Z}) \\ &= p^*(\mathrm{S}'_j|\mathbf{Z}). \end{aligned}$$

From the above equations, we obtain the paradox that

$$p(\mathrm{S}'_j|\mathbf{Z}) = p^*(\mathrm{S}'_j|\mathbf{Z}).$$

Using reduction to absurdity, $\mathrm{X}_i \in Pa(\mathrm{S}'_j)$ implies that $\mathrm{X}_i \in Pa^*(\mathrm{S}'_j)$. Similarly, we can prove the opposite direction of this implication. As a result, we have

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow \mathrm{X}_i \in Pa^*(\mathrm{S}'_j),$$

which shows that $\mathcal{G}^* = \mathcal{G}$. Therefore, we have proven that $\mathcal{G}$ is the ground-truth causal graph.

□

The following Theorem C.13 presents the robustness of the Theorem C.12 when variables are not accurately observed. We show that Eq. 2 works well if variables are inflicted with any independent restorable perturbations (e.g., additive independent Gaussian noises). However, the robustness may not hold if the data is compressed or incomplete, and such circumstances should be further investigated in the future.

Theorem C.13. *For simplicity, we use $\mathbf{X} = (\mathrm{X}_1, \dots, \mathrm{X}_{n_s+n_a}) = (\mathbf{S}, \mathbf{A})$ to denote all environment variables. Assume that there exists independent noise on the observed variables:*

$$\hat{\mathrm{X}}_i = \rho_i(\mathrm{X}_i, \epsilon_i), \qquad X_i \in \mathbf{X}; \tag{30}$$

$$\hat{\mathrm{S}}'_j = \rho'_j(\mathrm{S}'_j, \epsilon'_j), \qquad \mathrm{S}'_j \in \mathbf{S}'. \tag{31}$$

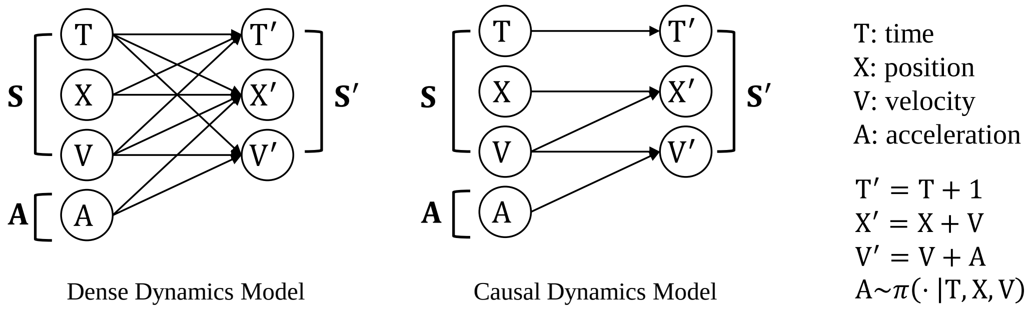


Figure 5. The dense dynamics model and causal dynamics model for a simple kinetic system.

Here, ρ_i and ρ'_j are perturbation functions applied to the environment variables. Additionally, ϵ_i and ϵ'_j are independent noise variables. If ***each perturbed variable*** $\hat{\mathrm{X}}_i$ ***can be restored as*** X_i ***given the value of*** ϵ_i*, then*

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow (\hat{\mathrm{X}}_i \not\perp\!\!\!\perp_P \hat{\mathrm{S}}'_j \,|\, (\hat{\mathbf{S}}, \hat{\mathbf{A}}) \setminus \{\hat{\mathrm{X}}_i\}). \tag{32}$$

Proof. Since each X_i can be restored from $\hat{\mathrm{X}}_i$ using ε_i, two forms of decomposition hold:

$$p(\mathbf{X}, \boldsymbol{\varepsilon}, \hat{\mathbf{X}}, \mathbf{S}') = p(\mathbf{X}) \left(\prod_i p(\varepsilon_i) p(\hat{rx}_i | \mathrm{X}_i, \varepsilon_i) \right) p(\mathbf{S}'|\mathbf{X}) \tag{33}$$

$$= p(\hat{\mathbf{X}}) \left(\prod_i p(\varepsilon_i | \hat{\mathrm{X}}_i) p(rx_i | \hat{\mathrm{X}}_i, \varepsilon_i) \right) p(\mathbf{S}'|\mathbf{X}) \tag{34}$$

$$\tag{35}$$

Therefore, if $\mathrm{X}_i \to \mathrm{S}'_j$, then $\varepsilon_i \to \hat{\mathrm{X}}_i \leftarrow \mathrm{X}_i \to \mathrm{S}'_j \to \hat{\mathrm{S}}_j$ and $(\varepsilon_i, \mathrm{X}_i) \to \mathrm{X}_i \to \mathrm{S}'_j \to \hat{\mathrm{S}}_j$ are two sub-graphs that comply with the Markov Compatibility. However, only Eq. 33 satisfies the faithfulness assumption, as the unconditional independence between ε_i and ε'_j can not be derived from Eq. 34.

Under the faithfulness assumption, the decomposition of Eq. 33 implies that

$$\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Rightarrow \left(\hat{\mathrm{X}}_i \not\perp\!\!\!\perp_{\mathcal{G}} \hat{\mathrm{S}}'_j \middle| \hat{\mathbf{X}}_{-i} \right) \Leftrightarrow \left(\hat{\mathrm{X}}_i \not\perp\!\!\!\perp_p \hat{\mathrm{S}}'_j \middle| \hat{\mathbf{X}}_{-i} \right)$$

Meanwhile, in the decomposition of Eq. 34, $\hat{\mathbf{X}}_{-i}$ blocks all potential back-doors from $\hat{\mathrm{X}}_i$ to $\hat{\mathrm{S}}'_j$. Using Theorem B.6 we have that

$$\mathrm{X}_i \notin Pa(\mathrm{S}'_j) \Rightarrow \left(\hat{\mathrm{X}}_i \perp\!\!\!\perp_p \hat{\mathrm{S}}'_j \middle| \hat{\mathbf{X}}_{-i} \right)$$

Then, $\mathrm{X}_i \in Pa(\mathrm{S}'_j) \Leftrightarrow \left(\hat{\mathrm{X}}_i \not\perp\!\!\!\perp_p \hat{\mathrm{S}}'_j \middle| \hat{\mathbf{X}}_{-i} \right)$ has been proved. □

C.4. An Example of Causal Dynamics Model

Consider a simple kinetic system where variables include T (time), X (position), V (velocity), and A (acceleration), where A is the action determined by the agent. Their dynamics are given in Figure 5. The dense dynamics model predicts the next-state variables using the entire input $(\mathrm{T}, \mathrm{X}, \mathrm{V}, \mathrm{A})$. However, the CDM predicts the next-state variables using only causal parents. In Figure 5, we present the CDM with a ground-truth causal graph, and a dense dynamics model has a fully-connected structure.

Suppose that X, V, and T all start with 0. Assume that the agent approximately uses a deterministic policy:

$$\pi(\mathrm{X}, \mathrm{V}, \mathrm{T}) \approx \begin{cases} 0, & \mathrm{V} = 1, \\ 1, & \mathrm{V} < 1, \\ -1, & \mathrm{V} > 1. \end{cases}$$

Then V is likely to be around 1 except for the initial step. Then, a spurious correlation that $\mathrm{X}' \approx \mathrm{T}$ would emerge in the data. In a dense dynamics model, this spurious correlation will possibly be learned, leading to serious generalization errors when the agent changes its policy. However, the CDM does not have such a problem, as T is not a parent of X'.

D. The Theory of Object-Oriented MDPs

In the paper's Section 2.2 we have introduced the concept of OOMDP, where variables are composed of the attributes of objects which are described by several classes. Now, we provide more information about OOMDPs, including rigorous definitions and the proof of the paper's Theorem 4.3.

D.1. Rigorous Definitions

Definition D.1 (Class)**.** A *class*, usually denoted as C, is a tuple $\langle \mathcal{F}_s[C], \mathcal{F}_a[C], \boldsymbol{Dom}_C \rangle$. Here, $\mathcal{F}_s[C]$ and $\mathcal{F}_a[C]$ are disjoint sets of *fields*, where $\mathcal{F}_s[C]$ is for *state fields* and $\mathcal{F}_a[C]$ is for *action fields*; the set of all fields C is defined as $\mathcal{F}[C] = \mathcal{F}_a[C] \cup \mathcal{F}_s[C]$; Each *field* in $\mathcal{F}[C]$ is a tuple like $\langle C, U \rangle$ (written as $C.U$ for short), where C is exactly the class symbol C, and U is the identifier of the field. $\boldsymbol{Dom}_C = \{Dom_{C.U}\}_{C.U \in \mathcal{F}[C]}$ gives the set of domains for each field.

Definition D.2 (Instance and attributes)**.** Consider that $\mathbf{O} \subseteq (\mathbf{S}, \mathbf{A})$ is a sub-group of variables at the current step of an FMDP, and that $C = \langle \mathcal{F}_s[C], \mathcal{F}_a[C], \boldsymbol{Dom}_C \rangle$ is a class. If there exist:

1. a bijection $\beta^s : \mathcal{F}_s[C] \to \mathbf{O} \cap \mathbf{S}$ such that the domain of $\beta^s(C.U)$ is exactly $Dom_{C.U}$ for every state field $C.U \in \mathcal{F}_s[C]$,
2. and a bijection $\beta^a : \mathcal{F}_a[C] \to \mathbf{O} \cap \mathbf{A}$ such that the domain of $\beta^a(C.U)$ is exactly $Dom_{C.U}$ for every action field $C.U \in= \mathcal{F}_a[C]$,

then we say that the FMDP contains an *instance* O (we use the corresponding, non-bold letter) of C, denoted as $O \in C$. Variables in $\mathbf{O}$ are called the *attributes* of O, denoted by attribute symbols: $O.\mathrm{U} := \beta^s(C.U)$ for every $C.U \in \mathcal{F}_s[C]$, or $O.\mathrm{U} := \beta^a(C.U)$ for every $C.U \in \mathcal{F}_a[C]$.

Definition D.3 (Object-oriented decomposition for FMDP)**.** Consider that $\mathcal{C} = \{C_1, \cdots, C_K\}$ is a set of classes. If $(\mathbf{S}, \mathbf{A})$ can be devided into N sub-groups $(\mathbf{O}_1, \cdots, \mathbf{O}_N)$ and each $\mathbf{O}_i$ forms the attributes of an instance O_i of some class in $\mathcal{C}$, we say the FMDP is *decomposed* by $\mathcal{C}$ and call each instance O_i as an *object*.

With the paper's assumptions about the result symmetry and the causation symmetry, we finally give the definition of an OOMDP below.

Definition D.4 (Object-oriented FMDP)**.** We say that an FMDP is an *object-oriented FMDP* (OOMDP) on a set of classes $\mathcal{C} = \{C_1, \cdots, C_K\}$, if

1. the state variables transit independently (see Assumption C.8),
2. the FMDP is decomposed by $\mathcal{C}$, and
3. Eqs. 3 and 4 hold under this decomposition.

D.2. Causal Graph for OOMDP

First, we prove that there always exists an OOCG to represent the dynamics of any OOMDP.

Theorem D.5. *In an OOMDP, there always exists a causal graph $\mathcal{G}$ such that $\mathcal{G}$ represents the dynamics of the OOMDP (see Definition C.3) and $\mathcal{G}$ is an OOCG.*

Proof. The proof is direct. Because variables transit independently, we have

$$p(\mathbf{S}'|\mathbf{S}, \mathbf{A}) = \prod_{C \in \mathcal{C}} \prod_{O \in C} \prod_{C.V \in \mathcal{F}_s[C]} p(O.\mathrm{V}'|\mathbf{S}, \mathbf{A}).$$

Therefore, the full OOCG, where $Pa(O.\mathrm{V}') = (\mathbf{S}, \mathbf{A})$ for each next-state attribute $O.\mathrm{V}'$, will always represent the dynamics of the OOMDP. □

Now we prove the paper's Theorem 4.3 that the ground-truth causal graph (see Difinition* C.6) of an OOMDP is always an OOCG.

Proof of the paper's Theorem 4.3. Assume that $\mathcal{G}$ is the ground-truth causal graph of the OOMDP. Based on Assumption C.8 and Theorem C.12, we have that $\mathcal{G}$ exists, is a bipartite causal graph (BCG), and can be uniquely identified. Consider any consistent probability function p and a DAG $\mathcal{G}_p$ that p is compatible with and faithful to. We know that this DAG is a concretization of $\mathcal{G}$ since $\mathcal{G}$ is the ground-truth causal graph.

Assume O_a and O_b are both instances of class C, and $C.U \in \mathcal{F}[C], C.V \in \mathcal{F}_s[C]$ are fields of C. According to Theorem B.11, we have that $(O_a.\mathrm{U} \perp\!\!\!\perp_p O_a.\mathrm{V}'|(\mathbf{S},\mathbf{A}) \setminus \{O_a.\mathrm{U}\})$ if $O_a.\mathrm{U} \notin Pa(O_a.\mathrm{V}')$. In other words, if $O_a.\mathrm{U} \notin Pa(O_a.\mathrm{V}')$ we have

$$p(O_a.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a,\cdots,\mathbf{O}_b,\cdots,\mathbf{O}_N) = p(O_a.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a^{-U},\cdots,\mathbf{O}_b,\cdots,\mathbf{O}_N),$$

where $\mathbf{O}_a^{-U}$ denotes $\mathbf{O}_a \setminus \{O_a.\mathrm{U}\}$. We define another consistent probability function q such that

$$q(\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b=\boldsymbol{y},\cdots,\mathbf{O}_N) \coloneqq p(\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{y},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots,\mathbf{O}_N),$$
$$q(\mathbf{S}'|\mathbf{S},\mathbf{A}) \coloneqq p(\mathbf{S}'|\mathbf{S},\mathbf{A}),$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of values assigned to the objects' attributes (If $p=q$ we enforce $\boldsymbol{x}=\boldsymbol{y}$). We use $\boldsymbol{y}_{-U}$ to denote the vector where the value for the field $C.U \in \mathcal{F}[C]$ is missing, so that $\boldsymbol{y}=(\boldsymbol{y}_{-U}, y_U)$ where y_U is the value for $C.U$. Then, we have

$$\begin{aligned}
&q(O_b.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b^{-U}=\boldsymbol{y}_{-U},\cdots,\mathbf{O}_N)\\
=&\int_{y_U} q(O_b.\mathrm{V}'|\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b=\boldsymbol{y},\cdots)q(O_b.\mathrm{U}=y_U|\cdots,\mathbf{O}_a,\cdots,\mathbf{O}_b^{-U}=\boldsymbol{y}_{-U},\cdots)\\
=&\int_{y_U} p(O_b.\mathrm{V}'|\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b=\boldsymbol{y},\cdots)p(O_a.\mathrm{U}=y_U|\cdots,\mathbf{O}_a^{-U}=\boldsymbol{y}_{-U},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots).
\end{aligned}$$

Using the result symmetry (the paper's equation 3), we have (continuing from the above equations)

$$\begin{aligned}
=&\int_{y_U} p(O_a.\mathrm{V}'|\cdots,\mathbf{O}_a=\boldsymbol{y},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots)p(O_a.\mathrm{U}=y_U|\cdots,\mathbf{O}_a^{-U}=\boldsymbol{y}_{-U},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots)\\
=&p(O_a.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{y},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots,\mathbf{O}_N)\\
=&q(O_a.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{y},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots,\mathbf{O}_N).
\end{aligned}$$

Using the result symmetry again, we have

$$q(O_a.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{y},\cdots,\mathbf{O}_b=\boldsymbol{x},\cdots,\mathbf{O}_N) = q(O_b.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b=\boldsymbol{y},\cdots,\mathbf{O}_N).$$

Combining the above formulae, we have

$$q(O_b.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b^{-U}=\boldsymbol{y}_{-U},\cdots,\mathbf{O}_N) = q(O_b.\mathrm{V}'|\mathbf{O}_1,\cdots,\mathbf{O}_a=\boldsymbol{x},\cdots,\mathbf{O}_b=\boldsymbol{y},\cdots,\mathbf{O}_N),$$

which says $(O_b.\mathrm{U} \perp\!\!\!\perp_q O_b.\mathrm{V}' \mid (\mathbf{S},\mathbf{A}) \setminus \{O_b.\mathrm{U}\})$.

Since p is faithful to $\mathcal{G}_p$, it is easy to prove that there exists a concretization $\mathcal{G}_q$ that q is faithful to. According to Theorem B.11, we have $(O_b.\mathrm{U} \perp\!\!\!\perp_{\mathcal{G}_q} O_b.\mathrm{V}'|(\mathbf{S},\mathbf{A})\setminus\{O_b.\mathrm{U}\})$. This leads to the corollary that $O_b.\mathrm{U} \notin Pa(O_b.\mathrm{V}')$. Therefore, we have proven that $O_a.\mathrm{U} \notin Pa(O_a.\mathrm{V}') \Rightarrow O_b.\mathrm{U} \notin Pa(O_b.\mathrm{V}')$. Similarly, we can prove that $O_a.\mathrm{U} \notin Pa(O_a.\mathrm{V}') \Leftarrow O_b.\mathrm{U} \notin Pa(O_b.\mathrm{V}')$. As a result, it is obvious that

$$O_a.\mathrm{U} \in Pa(O_a.\mathrm{V}') \Leftrightarrow O_b.\mathrm{U} \in Pa(O_b.\mathrm{V}') \tag{36}$$

So far, we have proven the shared local causality in the CG. Now, we follow a similar methodology to prove the shared global causality (we will skip some of the similar details). Assume O_a and O_b are both instances of C_k; Assume O_i and O_j are both instances of C, where $\{i,j\} \cap \{p,q\} = \varnothing$.

According to Theorem B.11, we have that $(O_a.\mathrm{U} \perp\!\!\!\perp_p O_i.\mathrm{V}'|(\mathbf{S},\mathbf{A}) \setminus \{O_a.\mathrm{U}\})$ if $O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}')$ In other words, if $O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}')$ we have

$$p(O_i.\mathrm{V}'|\mathbf{O}_a,\mathbf{O}_b,\mathbf{O}_i,\mathbf{O}_j,\cdots) = p(O_i.\mathrm{V}'|\mathbf{O}_a^{-U},\mathbf{O}_b,\mathbf{O}_i,\mathbf{O}_j,\cdots).$$

We re-define probability function q such that

$$q(\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \cdots) := p(\mathbf{O}_a = \boldsymbol{y}, \mathbf{O}_b = \boldsymbol{x}, \cdots),$$
$$q(\mathbf{S}'|\mathbf{S}, \mathbf{A}) := p(\mathbf{S}'|\mathbf{S}, \mathbf{A}).$$

where $\boldsymbol{x}$, $\boldsymbol{y}$ are vectors of values assigned to the objects' attributes. We have

$$\begin{aligned}
&q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b^{-U} = \boldsymbol{y}_{-U}, \cdots) \\
=& \int_{y_U} q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \cdots) q(O_b.\mathrm{U} = y_U|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b^{-U} = \boldsymbol{y}_{-U}, \cdots) \\
=& \int_{y_U} p(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \cdots) p(O_b.\mathrm{U} = y_U|\mathbf{O}_a = \boldsymbol{y}_{-U}, \mathbf{O}_b = \boldsymbol{x}, \cdots).
\end{aligned}$$

Using the causation symmetry (the paper's equation 4), we have (continuing the above equations)

$$\begin{aligned}
=& \int_{y_U} p(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{y}, \mathbf{O}_b = \boldsymbol{x}, \cdots) p(O_b.\mathrm{U} = y_U|\mathbf{O}_a = \boldsymbol{y}_{-U}, \mathbf{O}_b = \boldsymbol{x}, \cdots) \\
=& p(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{y}, \mathbf{O}_b = \boldsymbol{x}, \cdots) \\
=& q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{y}, \mathbf{O}_b = \boldsymbol{x}, \cdots).
\end{aligned}$$

Using the causation symmetry again, we obtain

$$q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b^{-U} = \boldsymbol{y}_{-U}, \cdots) = q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \cdots),$$

which says $(O_b.\mathrm{U} \perp\!\!\!\perp_q O_i.\mathrm{V}'|(\mathbf{S}, \mathbf{A}) \setminus \{O_b.\mathrm{U}\})$. This leads to that $O_b.\mathrm{U} \notin Pa(O_i.\mathrm{V}')$. Therefore, we can prove that $O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}') \Rightarrow O_b.\mathrm{U} \notin Pa(O_i.\mathrm{V}')$. Similarly, we can easily prove the other direction, leading to that

$$O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}') \Leftrightarrow O_b.\mathrm{U} \notin Pa(O_i.\mathrm{V}').$$

Using the result symmetry (the paper's equation 3), it is easy to get that

$$\begin{aligned}
&q(O_j.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b^{-U} = \boldsymbol{y}_{-U}, \mathbf{O}_i = \boldsymbol{z}, \mathbf{O}_j = \boldsymbol{w}, \cdots) \\
=& q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b^{-U} = \boldsymbol{y}_{-U}, \mathbf{O}_i = \boldsymbol{w}, \mathbf{O}_j = \boldsymbol{z}, \cdots) \\
=& q(O_i.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \mathbf{O}_i = \boldsymbol{w}, \mathbf{O}_j = \boldsymbol{z}, \cdots) \\
=& q(O_j.\mathrm{V}'|\mathbf{O}_a = \boldsymbol{x}, \mathbf{O}_b = \boldsymbol{y}, \mathbf{O}_i = \boldsymbol{z}, \mathbf{O}_j = \boldsymbol{w}, \cdots).
\end{aligned}$$

which says $(O_b.\mathrm{U} \perp\!\!\!\perp_q O_j.\mathrm{V}'|(\mathbf{S}, \mathbf{A}) \setminus \{O_b.\mathrm{U}\})$. This leads to that $O_b.\mathrm{U} \notin Pa(O_j.\mathrm{V}')$. Combining with the conclusion that we have just drawn, we have $O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}') \Rightarrow O_b.\mathrm{U} \notin Pa(O_j.\mathrm{V}') \Rightarrow O_a.\mathrm{U} \notin Pa(O_j.\mathrm{V}')$, and the other direction is proven similarly.

Finally, we obtain that

$$O_a.\mathrm{U} \notin Pa(O_i.\mathrm{V}') \Leftrightarrow O_b.\mathrm{U} \notin Pa(O_i.\mathrm{V}') \Leftrightarrow O_a.\mathrm{U} \notin Pa(O_j.\mathrm{V}') \Leftrightarrow O_b.\mathrm{U} \notin Pa(O_j.\mathrm{V}'). \tag{37}$$

Eqs. 36 and 37 together indicate that the causal graph is an OOCG, according to Definition 4.2. □

D.3. Object-Oriented Causal Discovery

In the main paper, we suggest using CMI for CITs, as it allows for varying numbers of instances and integrates causal discovery with model learning. The following Theorem D.6 describes how class-level causalities can be identified using CITs, providing the theoretic basis of our causal discovery. In Eqs. 38 and 39, the independence relationships in the right can be jointly tested through only one CIT, by merging the data of all concerned objects. We also note that CIT tools other than CMI are also applicable if the environment has a fixed number of instances for each class.

Theorem D.6 (Causal discovery for OOMDPs)**.** *If the ground-truth CG $\mathcal{G}$ of an OOMDP is an OOCG, it is uniquely identified under any probability function p that is dynamically faithful to $\mathcal{G}$ according to the following rules:*

$$C.U \to V' \iff \forall O \in C\big(O.\mathrm{U} \not\perp\!\!\!\perp_p O.\mathrm{V}' \,\big|\, (\mathbf{S}, \mathbf{A}) \setminus \{O.\mathrm{U}\}\big), \tag{38}$$

$$C_k.U \to C.V' \iff \forall O_j \in C\big(\mathbf{U}_{C_k.U|j} \not\perp\!\!\!\perp_p O_j.\mathrm{V}' \,\big|\, \mathbf{U}_{-C_k.U|j}\big), \tag{39}$$

where $\mathbf{U}_{C_k.U|j} \coloneqq \{O_r.\mathrm{U} \,|\, O_r \in C_k, r \neq j\}$ *and* $\mathbf{U}_{-C_k.U|j} \coloneqq (\mathbf{S}, \mathbf{A}) \setminus \mathbf{U}_{C_k.U|j}$.

Proof. Using Theorem C.12 and the paper's Definition 4.2, it is obvious that

$$\begin{aligned} C.U \to V' &\Leftrightarrow \forall O \in C(O.\mathrm{U} \in Pa(O.\mathrm{V}')) \\ &\Leftrightarrow \forall O \in C\big(O.\mathrm{U} \not\perp\!\!\!\perp_p O.\mathrm{V}' \,\big|\, (\mathbf{S}, \mathbf{A}) \setminus \{O.\mathrm{U}\}\big). \end{aligned}$$

From the paper's Definition 4.2 we have

$$C_k.U \to C.V' \Leftrightarrow \forall O_r \in C_k \forall O_j \in C(r = j \vee O_r.\mathrm{U} \in Pa(O_j.\mathrm{V}')).$$

From the paper's Theorem 4.3, we know that $\mathcal{G}$ is an OOCG, which guarantees d-separations in $\mathcal{G}$:

$$\forall O_r \in C_k \forall O_j \in C(r = j \vee O_r.\mathrm{U} \in Pa(O_j.\mathrm{V}')) \iff \forall O_j \in C\big(\mathbf{U}_{C_k.U|j} \not\perp\!\!\!\perp_{\mathcal{G}} O_j.\mathrm{V}' \,\big|\, \mathbf{U}_{-C_k.U|j}\big).$$

Using Theorem B.11 then we have

$$\begin{aligned} &\forall O_j \in C\big(\mathbf{U}_{C_k.U|j} \not\perp\!\!\!\perp_{\mathcal{G}} O_j.\mathrm{V}' \,\big|\, \mathbf{U}_{-C_k.U|j}\big) \\ &\quad\iff \forall O_j \in C\big(\mathbf{U}_{C_k.U|j} \not\perp\!\!\!\perp_p O_j.\mathrm{V}' \,\big|\, \mathbf{U}_{-C_k.U|j}\big). \end{aligned}$$

That is, we have

$$C_k.U \to C.V' \Leftrightarrow \forall O \in C\big(\mathbf{U}_{C_k.U|j} \not\perp\!\!\!\perp_p O.\mathrm{V}' \,\big|\, \mathbf{U}_{-C_k.U|j}\big).$$

□

D.4. Ensuring the Result and Causation Symmetries

Result symmetry (Eq. 3) and causation symmetry (Eq. 4) may be too strong to hold true in some cases. In an *asymmetric environment* (where one of the symmetries does not hold), the ground-truth causal graph may not be an OOCG, and some objects might possess their unique causal connections and dynamics, which may greatly compromise the performance of OOCDMs that strictly comply with these symmetries.

However, the dynamics cannot be modeled symmetrically typically because the attributes of objects provide insufficient information to do so. The following theorem indicates that both result symmetry and causation symmetry can be guaranteed by adding additional state attributes for the objects.

Theorem D.7. *Assume $\mathcal{M}$ is an OOMDP where the classes are $\{C_1, \cdots, C_K\}$, where the result symmetry (Eq. 3) and causation symmetry (Eq. 3) may be violated. There always exist an extended OOMDP $\widetilde{\mathcal{M}}$ such that the following statements hold ($\widetilde{*}$ denotes the corresponding item in $\widetilde{\mathcal{M}}$):*

1. *$\mathcal{F}_s[C_k] \subseteq \mathcal{F}_s[\widetilde{C}_k]$ and $\mathcal{F}_a[C_k] = \mathcal{F}_a[\widetilde{C}_k]$ for $k = 1, \cdots, K$.*
2. *There exist Φ that maps $(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}})$ into $(\mathbf{S}, \mathbf{A})$ and Ψ that maps $\widetilde{\mathbf{S}}'$ into $\mathbf{S}'$.*
3. *$\widetilde{\mathcal{M}}$ mirrors the dynamics of $\mathcal{M}$. In other words, $\tilde{p}(\widetilde{\mathbf{S}}'|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}) = p(\Psi(\widetilde{\mathbf{S}}')|\Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}))$.*
4. *Result symmetry and causation symmetry both hold in $\widetilde{\mathcal{M}}$.*

Proof. Let N_k denote the number of instances of C_k. Then, we define $\mathcal{F}_s[\widetilde{C}_k] = \mathcal{F}_s[C_k] \cup \{C_k.Id\}$, where $Dom_{C_k.Id} = \{1, 2, \cdots, N_k\}$. The distribution of the start state in $\widetilde{\mathcal{M}}$ ensures that each instance $\widetilde{O}$ has a unique $\widetilde{O}$.Id among all instances of $\widetilde{C}_k$.

We define $\phi(k, c)$ as the function that finds the index $i \in \{1, 2, \cdots, N\}$ such that $\widetilde{O}_{\phi(k,c)}$ is an instance of $\widetilde{C}_k$ and $\widetilde{O}_{\phi(k,c)}$.Id $= c$. In other words, $\phi(k, c)$ outputs the overall index of the c-th instance of $\widetilde{C}_k$ in $\widetilde{\mathcal{M}}$. Meanwhile, we use $RemoveId(\cdot)$ to remove all variables like $\widetilde{O}$.Id in the given variables.

In $\mathcal{M}$, we assume O_i is the c_i-th instance of class C_{k_i}. Then we may define Φ and Ψ as:

$$(\mathbf{S}, \mathbf{A}) = \Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}) := \left(RemoveId(\widetilde{\mathbf{O}}_{\phi(k_1,c_1)}), \cdots, RemoveId(\widetilde{\mathbf{O}}_{\phi(k_N,c_N)})\right)$$

$$\mathbf{S}' = \Psi(\widetilde{\mathbf{S}}') := \left(RemoveId(\widetilde{O}_{\phi(k_1,c_1)}.\mathbf{S}'), \cdots, RemoveId(\widetilde{O}_{\phi(k_N,c_N)}.\mathbf{S}')\right)$$

Then we directly define the dynamics of $\widetilde{\mathcal{M}}$ by

$$\widetilde{O}_i.\mathrm{Id}' \equiv \widetilde{O}_i.\mathrm{Id}, \qquad i = 1, \cdots, N.$$

$$\tilde{p}\left(RemoveId(\widetilde{\mathbf{S}}')|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}\right) := p(\Psi(\widetilde{\mathbf{S}}')|\Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}})),$$

Then we will have $\tilde{p}(\widetilde{\mathbf{S}}'|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}) := p(\Psi(\widetilde{\mathbf{S}}')|\Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}))$.

Therefore, assuming $\widetilde{O}_i \in \widetilde{C}_k$, we have

$$\tilde{p}(\widetilde{O}_i.\mathbf{S}'|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}) = \tilde{p}_{\widetilde{C}_k}(\widetilde{O}_i.\mathbf{S}'|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}}) := p\left(O_{\phi(k,\tilde{O}.\mathrm{Id})}.\mathbf{S}'|\Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}})\right).$$

Therefore, $\widetilde{\mathcal{M}}$ satisfies the result symmetry.

Moreover, it is easy to prove that $\widetilde{\mathcal{M}}$ satisfies the causation symmetry. Assuming $\widetilde{O}_x, \widetilde{O}_y \in \widetilde{C}_l$, swapping their attributes (which include $\widetilde{O}_x$.Id and $\widetilde{O}_x$.Id) does not affect $\phi(l, c)$ for any $c \in \{1, \cdots, N_l\}$. Therefore, it does not affect the result of $\Phi(\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}})$ and $\Psi(\widetilde{\mathbf{S}})$. Eventually, swapping the attributes of $\widetilde{O}_x$ and $\widetilde{O}_y$ has no influence on $\tilde{p}(\widetilde{O}_i.\mathbf{S}'|\widetilde{\mathbf{S}}, \widetilde{\mathbf{A}})$. □

The proof provides an easy way to ensure the result and causation symmetries – the OOMDP can simply include an *identity attribute* O.Id which gives the unique index of the object among all instances of its class. This is always plausible since no additional feature must be observed. Meanwhile, these identity attributes are fixed throughout an episode, and thus we do not need to learn their predictors in the implementation of OOCDM.

E. Details of Implementation

E.1. Structure of Object-Oriented Causal Dynamics Models

In an OOMDP, each attribute (variable) may contain one or several scalars. To handle the heterogeneous nature of different attributes, the OOCDM uses an *attribute encoder* $\mathrm{AttrEnc}_{C.U} : Dom_{C.U} \to \mathbb{R}^{d_e}$ for each field $C.U \in \mathcal{F} = \bigcup_{C \in \mathcal{C}} \mathcal{F}[C]$. It maps the attribute O.U of every instance $O \in C$ to a d_e dimensional *attribute-encoding vector*. All attribute encoders are implemented by a multi-layer perceptron where we use ReLU as the activation function.

Consider that $f_{C.V}$ is the predictor for the state field $C.V \in \mathcal{F}_s$ in an OOCDM. To compute $f_{C.V}(O_j.\mathrm{V}'|\mathbf{O}_j; \mathbf{U}_{-O_j}; \mathcal{G})$ for any $O_j \in C$, we first use the above encoders to encode all observed variables. Assume that the value of the attribute O_i.U of an object $O_i \in \mathcal{O}$ is observed to be $O_i.u$ (the corresponding lower-case letter is used) and the class of O_i is C_k, Then, this attribute is encoded into the attribute-encoding vector denoted as:

$$O_i.\boldsymbol{u} := \mathrm{AttrEnc}_{C_k.U}(O_i.u) \in \mathbb{R}^{d_e}, \quad U \in \mathcal{F}[C_k].$$

We now mask off the encoding vector if the attribute is not a parent variable for $O_j.\mathrm{V}'$ based on the OOCG $\mathcal{G}$. That is, we define the *masked attribute-encoding vector* of attribute O_i.U for $O_j.\mathrm{V}'$ as:

$$[O_i.\boldsymbol{u}]_{O_j.\mathrm{V}'} := \begin{cases} \mathbf{0}, & \text{if } j \neq i \text{ and } C_k.U \nrightarrow C.V' \\ \mathbf{0}, & \text{if } j = i \text{ and } C.U \nrightarrow V' \\ O_i.\boldsymbol{u}, & \text{otherwise.} \end{cases}$$

We concatenate all masked attribute-encoding vectors of O_i, and then we obtain a $(|\mathcal{F}[C_k]|d_e)$-dimensional vector called the *object-encoding vector* of O_i, denoted as $\boldsymbol{x}_i$:

$$\boldsymbol{x}_i = Concat\left([O_i.\boldsymbol{u}]_{O_j.\mathrm{V}'} \text{ for } C_k.U \in \mathcal{F}[C_k]\right).$$

Then, we apply a *query encoder*, denoted as $\mathrm{QEnc}_{C.V}$ that maps $\boldsymbol{x}_j$ to the *query vector* $\boldsymbol{q}$:

$$\boldsymbol{q} = \mathrm{QEnc}_{C.V}(\boldsymbol{x}_j) \in \mathbb{R}^{d_k}.$$

For every other object O_i such that $j \neq i$ (we denote the class of O_i as C_k), we apply a *key encoder* $\mathrm{KEnc}_{C_k \to C.V}$ and a *value encoder* $\mathrm{VEnc}_{C_k \to C.V}$ that respectively map $\boldsymbol{x}_i$ to a key-vector $\boldsymbol{k}_i$ and a value-vector $\boldsymbol{v}_i$:

$$\boldsymbol{k}_i = \mathrm{KEnc}_{C_k \to C.V}(\boldsymbol{x}_i) \in \mathbb{R}^{d_k},$$
$$\boldsymbol{v}_i = \mathrm{VEnc}_{C_k \to C.V}(\boldsymbol{x}_i) \in \mathbb{R}^{d_v}.$$

Then, we perform the key-value attention (Vaswani et al., 2017):

$$\alpha_i = \frac{\exp\left(\boldsymbol{q}^T \boldsymbol{k}_i / \sqrt{d_k}\right)}{\sum_{r \neq i} \exp\left(\boldsymbol{q}^T \boldsymbol{k}_r / \sqrt{d_k}\right)},$$
$$\boldsymbol{h} := (\boldsymbol{q}, \sum_{j \neq i} \alpha_i \boldsymbol{v}_i) \in \mathbb{R}^{d_k + d_v},$$

where $\boldsymbol{h}$ is called the *distribution embedding* of $O_j.\mathrm{V}'$.

Finally, we use a distribution decoder $\mathrm{Dec}_{\mathrm{C.V}}$ to map $\boldsymbol{h}$ into the distribution of $\hat{p}(O_j.\mathrm{V}'|Pa(O_j.\mathrm{V}'))$. If $Dom_{C.V}$ is continuous, it outputs the mean and standard variance of a normal distribution:

$$(\mu, \sigma) = Dec_{C.V}(\boldsymbol{h}); \quad p(O_j.\mathrm{V}'|Pa(O_j.\mathrm{V}')) \sim \mathcal{N}(\mu, \sigma).$$

If $Dom_{C.V}$ is discrete (we assume that $Dom_{C.V}$ has m elements), then $\mathrm{Dec}_{\mathrm{C.V}}$ outputs the probability of each choice:

$$(p_1, \cdots, p_m) = Dec_{C.V}(\boldsymbol{h}); \quad p(O_j.\mathrm{V}'|Pa(O_j.\mathrm{V}')) \sim \mathrm{Categorical}(p_1, \cdots, p_m).$$

The illustration of the structure of such a predictor is presented in Figure 4 of the main paper, where $i = 1$. So far, we have described the structure of one single predictor $f_{C.V}$, and other predictors follow the same design as $f_{C.V}$. In addition, it is possible to compute $\hat{p}(O_j.\mathrm{V}'|Pa(O_j.\mathrm{V}'))$ for all $O_j \in C$ in parallel. Therefore, the predictor $f_{C.V}$ actually outputs $\hat{p}(O_j.\mathrm{V}'|Pa(O_j.\mathrm{V}'))$ for all $O_j \in C$ once-for-all in our implementation (read our code for more detail).

E.2. Augmented OOCDM for Asymmetric Environments

Appendix D.4 introduces a way to ensure the result and causation symmetries by adding additional attributes about the identity of objects. To extend OOCDM to asymmetric environments, we implement built-in auxiliary attributes by augmenting the predictors in Appendix E.1. The augmented predictors are able to handle the asymmetric dynamics without explicitly modifying the representation of the OOMDP. The new architecture is illustrated in Figure 6. Each class contains a bi-directional Gated Recurrent Unit (GRU) to generate the hidden encodings of auxiliary attributes for each instance. The GRU of C_k recurrently produces the hidden encodings of all instances, starting from a trainable initial encoding $\boldsymbol{h}^k_{init}$. Let $k = 1$ for example. Assuming N_1 is the number of instances of C_1, we have

$$(\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{N_1}) = \mathrm{GRU}_1(\boldsymbol{h}^1_{init})$$

The hidden encodings are integrated into the object-encoding vectors:

$$\boldsymbol{x}_i = Concat\left([O_i.\boldsymbol{u}]_{O_j.\mathrm{V}'} \text{ for } C_k.U \in \mathcal{F}[C_k]; \boldsymbol{h}_i\right).$$

The other parts are the same as the architecture in Appendix E.1. For simplicity, all predictors in the OOCDM share the same set of GRUs to produce hidden attributes.

E.3. The Algorithm of Object-Oriented Causal Discovery

We define the following notations:

1. $\boldsymbol{s}_t$, $\boldsymbol{a}_t$, and $\boldsymbol{s}_{t+1}$ are the observed values of $\mathbf{S}$, $\mathbf{A}$, and $\mathbf{S}'$ at step t.
2. $O_j.v_{t+1}$ to denote the observed value of attribute $O_j.\mathrm{V}$ at step $t+1$.
3. C^t denotes the set of instances of class C at step t.

Then, the pseudo-code of object-oriented causal discovery is provided in Algorithm 1.

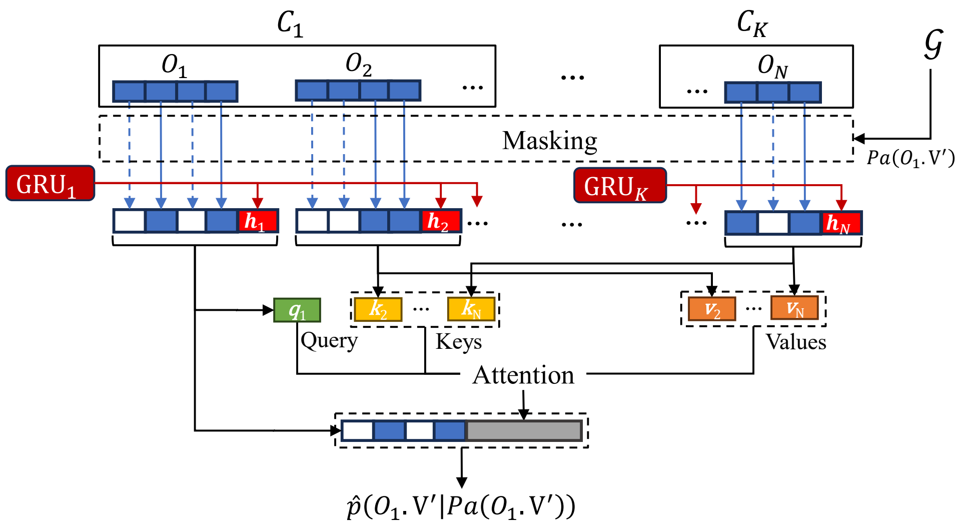


Figure 6. The illustration of $f_{C_1.V}(O_1.\mathrm{V}'|\mathbf{O}_1, \boldsymbol{h}_1, \cdots, \mathbf{O}_N, \boldsymbol{h}_N; \mathcal{G})$ in the augmented OOCDM.

E.4. The Algorithm of Model Learning

The model is learned by optimizing the target function defined in the paper's equation 9 $J(\mathcal{D})$, with a given data-set $\mathcal{D}$. However, it is impractical and expensive to compute $J(\mathcal{D})$ if $\mathcal{D}$ contains too many samples. Therefore, we use stochastic gradient ascent, in which we repeatedly sample a batch $\mathcal{B} \subset \mathcal{D}$ and maximize $J(\mathcal{B})$. The pseudo-code of learning our OOCDM is provided in Algorithm 2. In this algorithm, we consider both online and offline settings, although in our experiments we only adopt offline learning to best exploit the advantage of generalization.

E.5. Planning with Dynamics Models

We combine dynamics models with Model Predictive Control (MPC) (Camacho & Bordons, 1999), where the Cross-Entropy Method (CEM) (Botev et al., 2013) is used as the planning algorithm to determine the agents' actions. Given a planning horizon H, the following process is repeated several times: 1) First, we sample k action sequences with lengths of H from a distribution $p_{\mathbf{\Theta}}(\mathbf{A}_1, \cdots, \mathbf{A}_H)$ parameterized by $\mathbf{\Theta}$; 2) then, we use the dynamics models to perform counterfactual reasoning with these action sequences, which generates k H-step trajectories; 3) among these trajectories, we choose the top-k^* (we have $k^* < K$) trajectories with the highest returns to update the parameter $\mathbf{\Theta}$. In the final iteration, we return the first action in the trajectory that produces the highest return.

Since our work only focuses on the dynamics, we assume that true reward function $R(\mathbf{S}, \mathbf{A}, \mathbf{S}')$ of the environment is given so that an extra reward model is not required. This makes sure that no reward bias is introduced in our comparison between different kinds of dynamics models. We present the pseudo-code of planning in Algorithm 3.

F. Complexity Analysis

In this section, we only consider one OOMDP so that the numbers of the instances of classes are fixed. The following symbols are used in this section:

1. N_i denotes the number of instances of the i-th class C_i.
2. K denotes the number of classes.
3. $N := \sum_{i=1}^{K} N_i$ denotes the number of objects.
4. $m_i := |\mathcal{F}[C_i]|$ denotes the number of fields of the i-th class C_i.
5. $m := \sum_{i=1}^{K} m_i$ denotes the overall number of fields in the OOMDP.
6. $n := \sum_{i=1}^{K} N_i m_i$ denotes the number of variables (attributes) at each step in the OOMDP.
7. k denotes the number of samples used in predicting, causal discovery, or planning.
8. k^* denotes the number of elite samples used in the Cross-Entropy Method (CEM) for planning.
9. H denotes the planning horizon in Model Predictive Control (MPE) for planning.

Algorithm 1 Object-oriented causal discovery

```
Require: The dataset $\mathcal{D} = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{a}_{t+1})\}_{t=1}^{T}$, predictors $\{f_{C.V}\}_{C \in \mathcal{C}, C.V \in \mathcal{F}_s[C]}$, and $\varepsilon \geq 0$.
 1: Initialize $\mathcal{G} \longleftarrow$ empty OOCG.
 2: for $C.V$ in $\bigcup_{C \in \mathcal{C}} \mathcal{F}_s[C]$ do
 3:   $\mathcal{L} \leftarrow \sum_{t=1}^{T} \sum_{O_j \in C^t} \log f_{C.V}(O_j.v_{t+1} | \boldsymbol{s}_t, \boldsymbol{a}_t; \mathcal{G}_1)$.
 4:   for $C.U$ in $\mathcal{F}[C]$ do
 5:     $\tilde{\mathcal{L}} \leftarrow \sum_{t=1}^{T} \sum_{O_j \in C^t} \log f_{C.V}(O_j.v_{t+1} | \boldsymbol{s}_t, \boldsymbol{a}_t; \mathcal{G}_{C.U \not\to V'})$.
 6:     $\mathcal{I}_{\mathcal{D}}^{C.U \to V'} \leftarrow \frac{1}{\sum_{t=1}^{T} |C^t|}(\mathcal{L} - \tilde{\mathcal{L}})$.
 7:     Add $C.U \to V'$ into $\mathcal{G}$ if $\mathcal{I}_{\mathcal{D}}^{C.U \to V'} > \varepsilon$.
 8:   end for
 9:   for $C_k.U$ in $\bigcup_{C_k \in \mathcal{C}} \mathcal{F}[C_k]$ do
10:     $\tilde{\mathcal{L}} \leftarrow \sum_{t=1}^{T} \sum_{O_j \in C^t} \log f_{C.V}(O_j.v_{t+1} | \boldsymbol{s}_t, \boldsymbol{a}_t; \mathcal{G}_{C_k.U \not\to C.V'})$.
11:     $\mathcal{I}_{\mathcal{D}}^{C_k.U \to C.V'} \leftarrow \frac{1}{\sum_{t=1}^{T} |C^t|}(\mathcal{L} - \tilde{\mathcal{L}})$.
12:     Add $C_k.U \to C.V'$ into $\mathcal{G}$ if $\mathcal{I}_{\mathcal{D}}^{C_k.U \to C.V'} > \varepsilon$.
13:   end for
14: end for
15: return $\mathcal{G}$
```

Algorithm 2 Learning Object-oriented Causal Dynamics Model

```
Require: The dataset $\mathcal{D}$, number $n_{iter}$ of iterations, and number $n_{batch}$ of batches in each iteration.
 1: Initialize predictors $f_{C.V}$ for every $C.V \in \bigcup_{C \in \mathcal{C}} \mathcal{F}_s[C]$.
 2: for $i_{iter} = 1, \cdots, n_{iter}$ do
 3:   Obtain $\hat{\mathcal{G}}$ using causal discovery (Algorithm 1).
 4:   for $i_{batch} = 1, \cdots, n_{batch}$ do
 5:     Sample batch $\mathcal{B} \subset \mathcal{D}$.
 6:     Perform gradient ascent on $\mathcal{J}(\mathcal{B})$ defined in the paper's equation 9.
 7:   end for
 8:   Optionally, collect new data into $\mathcal{D}$ using the latest policy. {for online learning only}
 9: end for
10: return predictors $\{f_{C.V}\}_{C \in \mathcal{C}, C.V \in \mathcal{F}_s[C]}$ and $\hat{\mathcal{G}}$.
```

10. l denotes the number of iterations in CEM.
11. Most importantly, O becomes the symbol for an asymptotic boundary rather than an object (in this section only).

It is obvious that $n \geq N$ and $n \geq m$ hold in all OOMDPs. Especially, in large-scale environments, we have $n >> m$. We assume that the number of fields of each class is bounded by $m_{\max}$, and then n and N has the same magnitude since $N \leq n \leq m_{\max} N$.

The theorems about the complexities of our OOCDM (implemented as described in Appendix E) are presented and proven in the following.

Theorem F.1. *The time complexity of predicting the next states using our OOCDM is $O(nNk)$.*

Proof. Since attribute encoders are shared by encoders, then computing all attribute-encoding vectors costs $O(nk)$. Then, for every state field $C_i.V \in \mathcal{F}_s[C_i]$ of every class C_i, the predictor spends:

1. $O(nk)$ in applying masks to and concatenating attribute-encoding vectors into object-encoding vectors;
2. $O(Nk)$ in deriving key, value, and query vectors from object-encoding vectors;
3. $O(N_i(N - N_i)k + N_i k) = O(N_i N k)$ in the attention operation;
4. $O(N_i k)$ in decoding the distribution embedding.

Therefore, each state field in f_{C_i} leads to a cost of $O(nk) + O(Nk) + O(N_i N k) + O(N_i k) = O(N_i N k)$. By summing up

Algorithm 3 Planning with Cross Entropy Method

```
Require: The dynamics model $\hat{p}$, the reward function $R$, the current state $\boldsymbol{s}$, the planning horizon $H$, the number $n_{plan}$ of
    iterations, the number $k$ of samples, the number $k^*$ of elite samples, and the discount factor $\gamma$.
 1: Initialize the parameter $\boldsymbol{\Theta}$.
 2: for $i = 1, \cdots, n_{plan}$ do
 3:   for $j = 1, \cdots, k$ do
 4:     Sample the $j$-th $H$-step action sequences $(\boldsymbol{a}_1^{(j)}, \cdots, \boldsymbol{a}_H^{(j)})$ with $p_{\boldsymbol{\Theta}}(\mathbf{A}_1, \cdots, \mathbf{A}_H)$.
 5:     $\boldsymbol{s}_1^{(j)} \leftarrow \boldsymbol{s}$.
 6:     for $t = 1, \cdots, H$ do
 7:       Sample $\boldsymbol{s}_{t+1}^{(j)}$ using $\hat{p}(\mathbf{S}'|\mathbf{S} = \boldsymbol{s}_t^{(j)}, \mathbf{A} = \boldsymbol{a}_t^{(j)})$.
 8:       Compute the reward $r_t^{(j)} \leftarrow R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$.
 9:     end for
10:     Compute the return $r_j \leftarrow \sum_{t=1}^{H} \gamma^{t-1} r_t^{(j)}$.
11:   end for
12:   if $i < n_{plan}$ then
13:     $\boldsymbol{E} \leftarrow$ the set of top-$k^*$ action sequences with the highest return $r_j$ $(j \in \{1, \cdots, k\})$.
14:     $\boldsymbol{\Theta} \leftarrow$ Maximum-Likelihood-Estimation$(\boldsymbol{E})$.
15:   else
16:     $j^* \leftarrow \arg\max_j r_j$.
17:     return $\boldsymbol{a}_1^{(j^*)}$.
18:   end if
19: end for
```

the costs of all state fields, the cost of predicting the next states is

$$
\begin{aligned}
&O(nk) + \sum_{i=1}^{K} m_i \cdot O(N_i N k) \\
=&O(nk) + O\left(N \sum_{i=1}^{K} m_i N_i k\right) \\
=&O(nk) + O(nNk) \\
=&O(nNk).
\end{aligned}
$$

□

Theorem F.2. *The time complexity of causal discovery using our OOCDM is $O(nmNk)$.*

Proof. In the process of proving Theorem F.1, we know that each predictor costs $O(N_i N k)$ to predict the next-state attribute.

First, we consider the local causalities. For each class C_i, we have m_i^2 local causalities. For each local causality expression $localcausC_iUV$, the predictor $f_{C_i.V}$ is used twice for each sample to estimate $\mathcal{I}_{\mathcal{D}}^{localcausC_iUV}$. Therefore, the complexity of discovering all local causalities shared by C_i is $O(m_i^2 N_i N k)$

Then, we consider the global causalities. For each class C_i, we have $m_i m$ global causalities. For each global causality expression like $C_j.U \to C_i.V'$, the predictor $f_{C_i.V}$ is used twice for each sample to estimate $\mathcal{I}_{\mathcal{D}}^{C_j.U \to C_i.V'}$. Therefore, the complexity of discovering all global causalities shared by C_i is $O(m_i m N_i N k)$

Combing the above results, all causalities (local and global) shared by C_i cost

$$O(m_i^2 N_i N k) + O(m_i m N_i N k) = O(m_i m N_i N k).$$

Finally, the time complexity for causal discovery is

$$\sum_{i=1}^{K} O(m_i m N_i N k) = O\left(mNk \sum_{i=1}^{K} O(m_i N_i)\right) = O(nmNk).$$

Table 8. Comparison of computational complexity between our OOCDM and the state-of-the-art CDMs. Here "t.c." means "time complexity" and "s.c." means "space complexity".

	OOCDM	CDL	GRADER
t.c. of predicting	$O(nNk)$	$O(n^2k)$	$O(n^2k)$
t.c. of causal discovery	$O(nmNk)$	$O(n^3k)$	$O(n^3k\log k)$
t.c. of planning	$O(lk(HnN+\log k^*))$	$O(lk(Hn^2+\log k^*))$	$O(lk(Hn^2+\log k^*))$
s.c. of model weights	$O(m^2)$	$O(n^2)$	$O(n^2)$

□

Theorem F.3. *The time complexity of planning for an action using our OOCDM is* $O(lk(HnN+\log k^*))$.

Proof. In each iteration, we have k action sequences with lengths of H. Therefore, sampling the action sequences costs $O(kH)$. Then, using models simulating trajectories and computing returns cost $H\cdot O(nNk)=O(HnNk)$. Identifying the top-k^* trajectories costs $O(k\log k^*)$. Re-estimating parameters costs $O(k*H)$. Therefore, the cost of each iteration is

$$O(kH)+O(HnNk)+O(k\log k^*)+O(Hk^*)=O(k(HnN+\log k^*)).$$

Finally, considering l iterations, the time complexity of planning for an action of our OOCDM is $O(lk(HnN+\log k^*))$ □

Theorem F.4. *The space complexity of model weights of our OOCDM is* $O(m^2)$.

Proof. The space complexity of attribute encoders is $O(m)$. In each predictor, there exists K key encoders, K value encoders, one query encoder, and one distribution decoder. Here, the space complexity of the key-encoder, value-encoder, or query-encoder for each class C_i is $O(m_i)$; and the space complexity of the distribution decoder is $O(1)$.

Finally, the space complexity of the entire OOCDM is

$$\begin{aligned}&\sum_{i=1}^{K} m_i\left(2\sum_{j=1}^{K}O(m_j)+2\cdot O(m_i)+O(1)\right)+O(m)\\=&\sum_{i=1}^{K}m_iO(m)+O(m)\\=&O(m^2)\end{aligned}$$

□

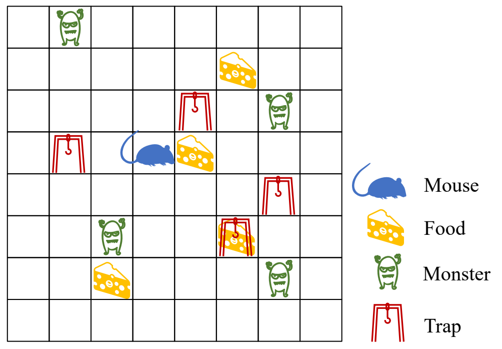
In Table 8, we further compare our OOCDM with the state-of-the-art CDMs in terms of the above-mentioned aspects of computational complexity. These baselines include 1) **CDL**, which learns the SCM underlying the environmental dynamics by estimating CMIs like us (Wang et al., 2022), and 2) **GRADER**, which uses Fast CIT to discover causalities and uses GRUs to fit structural equations (Ding et al., 2022). We can see that our OOCDM utilizes object-oriented information to share sub-models (predictors) and causality among objects of each class, leading to a great reduction of computational complexity, especially the scale of model weights and the time complexity of causal discovery. It is worth noting that all predictors are implemented in parallel in practice, making our OOCDM even more computationally efficient if GPUs are used.

G. Environments

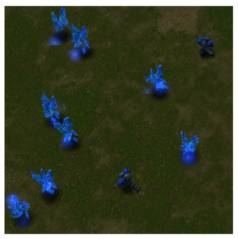
G.1. Block

The Block environment is a simple environment designed to validate the effectiveness of causal discovery for different numbers of environmental variables. It contains two classes: $\mathcal{C}=\{Block,Total\}$. The fields of these classes are given by

- $\mathcal{F}_s[Block]=\{Block.S_1,Block.S_2,Block.S_3\}$



(a) Mouse



(b) CMS

(c) DZB

Figure 7. Illustrations of the (a) Mouse, (b) Collect-Mineral-Shards, and (c) Defeat-Zerglings-Banelings environments.

- $\mathcal{F}_a[Block] = \{Block.A\}$,
- $\mathcal{F}_s[Total] = \{Total.S_1, Total.S_2, Total.S_3, Total.T\}$,
- $\mathcal{F}_a[Total] = \varnothing$.

The transition of each $O \in Block$ follows a linear transform:

$$\begin{pmatrix} O.\mathrm{S}_1' \\ O.\mathrm{S}_2' \\ O.\mathrm{S}_3' \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & -0.3 \\ 0.5 & 1.0 & 0 & 0 \\ 0 & 0.25 & 0.75 & 1.0 \end{bmatrix} \begin{pmatrix} O.\mathrm{S}_1 \\ O.\mathrm{S}_2 \\ O.\mathrm{S}_3 \\ \tanh O.\mathrm{A} \end{pmatrix} + \mathcal{N}(\mathbf{0}, 0.01^2 \boldsymbol{I}). \tag{40}$$

The transition of the instance of $Total$ follows that

$$O.\mathrm{S}_j' = \frac{1}{2} O.\mathrm{S}_j + \frac{1}{2} \max_{O_i \in Block} O_i.\mathrm{S}_j, \qquad j = 1, 2, 3, \tag{41}$$

$$O.\mathrm{T}' = O.\mathrm{T} + 1 + \mathcal{N}(0, 0.01^2). \tag{42}$$

The Block environment contains no rewards. That is, $R(\mathbf{S}, \mathbf{A}, \mathbf{S}') \equiv 0$.

At the beginning of each episode, We initialize the attributes of each $Block$ object by

$$(O.\mathrm{S}_1,\ O.\mathrm{S}_2,\ O.\mathrm{S}_3)^T \sim \mathcal{N}\left((1, 0, 0)^T, \operatorname{diag}(0.25, 1, 1)\right), \tag{43}$$

and initialize the $Total$ instance by

$$(O.\mathrm{S}_1,\ O.\mathrm{S}_2,\ O.\mathrm{S}_3,\ O.\mathrm{T})^T \sim \mathcal{N}\left(\mathbf{0}, \operatorname{diag}\left(0.01^2, 0.01^2, 0.01^2, 0\right)\right). \tag{44}$$

We use a random policy (which produces Gaussian actions) to obtain the training data. Therefore, $O.\mathrm{S}_1$ for every $O \in Block$ is likely to stay close to 1. Further, this leads to spurious correlations such as $Total.T \to Block.S_2'$.

The ground-truth causal graph of the Block environment is an OOCG, which we visualize in Figure 8.

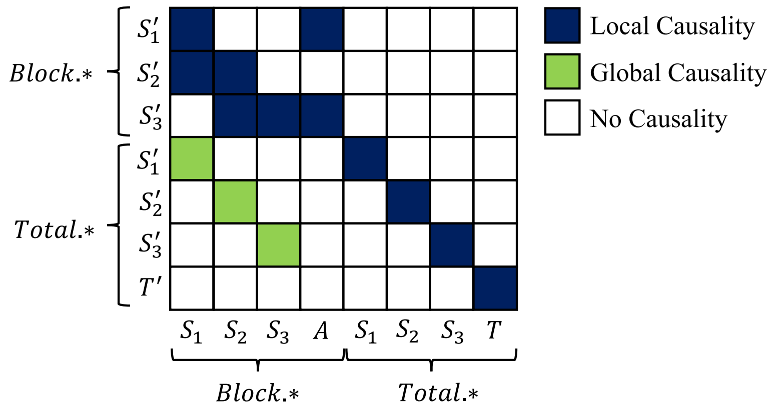


Figure 8. The visualization of the ground-truth OOCG (the adjacency matrix of class-level causalities) of the Block environment.

G.2. Mouse

The Mouse Environment aims to validate the performance of dynamics models in a more complicated OOMDP. It contains four classes: $\mathcal{C} = \{Mouse, Food, Monster, Trap\}$, whose fields are

- $\mathcal{F}_s[Mouse] = \{Mouse.Health, Mouse.Hunger, Mouse.Position\}$,
- $\mathcal{F}_a[Mouse] = \{Mouse.Move\}$,
- $\mathcal{F}_s[Food] = \{Food.Amount, Food.Position\}$,
- $\mathcal{F}_s[Monster] = \{Monster.Noise, Monster.Position\}$,
- $\mathcal{F}_s[Trap] = \{Trap.Duration, Trap.Position\}$,
- $\mathcal{F}_a[Food] = \mathcal{F}_a[Monster] = \mathcal{F}_a[Trap] = \varnothing$

All objects are located in an 8×8 grid world. That is, the domain of the field $Position$ of every class is in $\{0, 1, \cdots, 7\}^2$. Typically, the environment contains only one instance of $Mouse$ and arbitrary numbers of instances of other classes. We illustrate the Mouse environment in Figure 7(a).

The instance O_{mouse} of $Mouse$ has an attribute O_{mouse}.Health ≤ 10 and O_{mouse}.Hunger $\in [0, 100]$. The hunger point O_{mouse}.Hunger is reduced by 1 for each step unless the mouse reaches any instance of food. For each $O_{food} \in Food$ that is reached by the mouse (i.e., O_{food}.Position $=$ O_{mouse}.Position), the mouse consumes all amount of the food (O_{food}.Amount$'$ $\leftarrow 0$) and restores the equal amount of $O.Hunger$. If the mouse is starving (O_{mouse}.Hunger < 25), it loses one point of O.Health for each step. However, if the mouse is full (O_{mouse}.Hunger > 75), it restores one point of O.Health for each step. If the health O_{mouse}.Health drops below 0, the episode terminates because the mouse is dead.

The mouse has an action attribute O_{mouse}.Move, which can be chosen from 5 choices: North, South, East, West, and Staying. Except for Staying, the mouse's position O_{mouse}.Position changes to the nearby grid based on the chosen direction (unless it reaches the boundary of the world). However, if the mouse is trapped by a trap $O_{trap} \in Trap$ (i.e., O_{trap}.Position $=$ O_{mouse}.Position and O_{trap}.Duration > 0), then the mouse's position will not be changed no matter what O_{mouse}.Move is chosen, and O_{trap}.Duration is reduced by 1.

The positions of $Food$ instances are fixed after being randomly initialized. The amount O_{food}.Amount of an instance O_{food} slowly accrues with time. That is O_{food}.Amount$'$ $\leftarrow$ O_{food}.Amount $+ \mathcal{N}(1, 0.01)$ unless it is consumed by the mouse. We note that O_{food}.Amount increases slower than that O_{mouse}.Hunger decreases. Therefore, the mouse must constantly navigate from one food to another to prevent from starving.

An instance $O_{monster}$ of $Monster$ randomly wanders in the world. That is, its position randomly changes into a nearby grid at each step. If the mouse is reached by a monster (i.e., $O_{monster}$.Position $=$ O_{mouse}.Position), the mouse directly

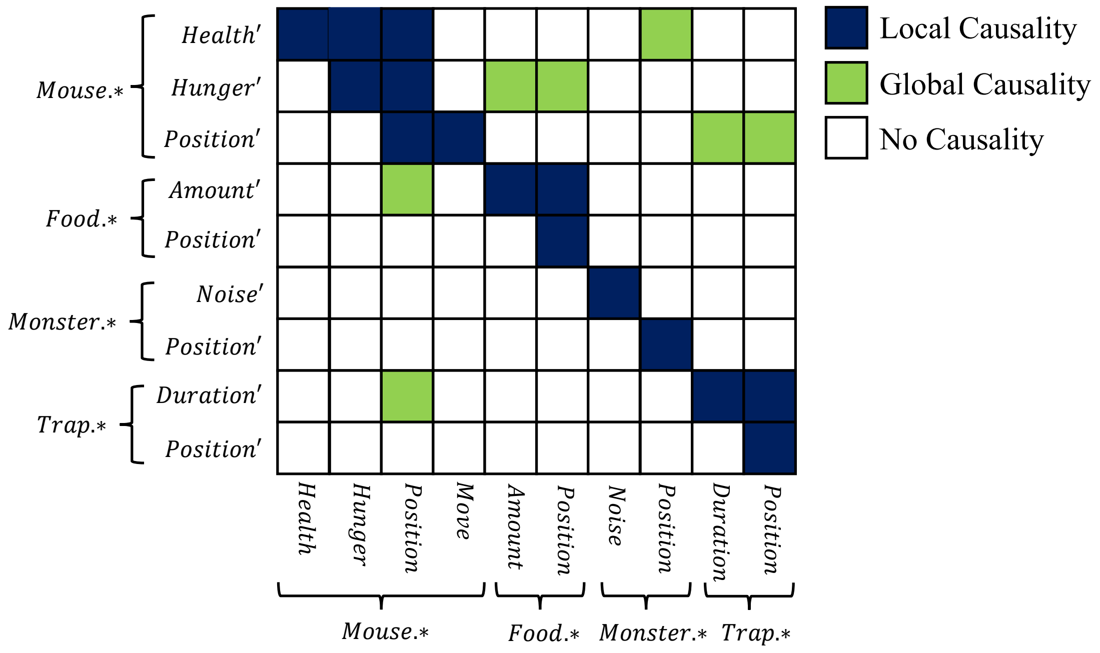


Figure 9. The visualization of the ground-truth OOCG (the adjacency matrix of class-level causalities) of the Mouse environment.

loses 5 points of O_{mouse}.Health. Each monster also contains an attribute $O_{monster}$.Noise of noise, which is used to create spurious correlations and confuse non-causal dynamics models. The transition of $O_{monster}$.Noise is given by $O_{monster}.\text{Noise}' = O_{monster}.\text{Noise} + \mathcal{N}(0, 0.01)$.

The goal of the agent is to make the mouse live as long as possible and stay away from starving. Therefore, the reward function is given by:

$$\begin{aligned} R(\mathbf{S}, \mathbf{A}, \mathbf{S}') =& 0.01 \cdot O_{mouse}.\text{Hunger} + (O_{mouse}.\text{Health}' - O_{mouse}.\text{Health} \\ & + 0.05 \cdot (O_{mouse}.\text{Hunger}' - O_{mouse}.\text{Hunger}). \end{aligned} \tag{45}$$

The ground-truth causal graph of the Mouse environment is an OOCG, which we visualize in Figure 9.

G.3. StarCraft Mini-Games

Our experiments consider two StarCraft mini-games as environments. We formulate these environments as OOMDPs merely based on our intuition. That is, the objects correspond to units in the StarCraftII game and classes correspond to the type of the unit. The values of the attributes are observed through the PySC2 (Vinyals et al., 2017) interface.

The true dynamics of these environments are implemented in the StarCraftII engine. Being non-developers of the game, we do not know the precise dynamics of these environments. For example, we observe that if a marine chooses NOOP as its action, it automatically attacks a hostile unit (if any) in its attacking range. However, we have no clue based on what rule it chooses the unit that it attacks. Therefore, we do not know whether the Definition* D.4 of OOMDP is strictly satisfied (possibly not), not to mention the ground-truth causal graph of these environments.

We believe that 1) humans factorize the world into components (variables) based on independent relationships, and 2) We discriminate and categorize objects based on structural and dynamical similarity. Therefore, we believe that the Definition* D.4 is roughly satisfied, even though the object-oriented description is provided by non-experts. Through these StarCraft mini-games, we hope to show that our OOCDM is applicable to a wide range of RL problems.

G.3.1. Collect-Mineral-Shards

The Collect-Mineral-Shards (CMS) environment is a StarCraftII mini-game. The game contains 2 marines and 20 mineral shards. The player (agent) needs to control the movement of the marines to collect all mineral shards as fast as possible. We illustrate the CMS environment in Figure 7(b).

We decompose the environment by 2 classes $\mathcal{C} = \{Marine, Mineral\}$ such that

- $\mathcal{F}_s[Marine] = \{Marine.Position\}$, $\mathcal{F}_a[Marine] = \{Marine.Move\}$,
- $\mathcal{F}_s[Mineral] = \{Mineral.Position, Mineral.Collected\}$, $\mathcal{F}_a[Mineral] = \varnothing$,

where we set

- $Dom_{Marine.Position} = Dom_{Mineral.Position} = [-99, 99]^2$,
- $Dom_{Marine.Move} = \{\text{North}, \text{East}, \text{South}, \text{West}, \text{NOOP}\}$,
- and $Dom_{Mineral.Collected} = \{\text{True}, \text{False}\}$.

We define the reward function as the number of collected mineral shards in each step:

$$R(\mathbf{S}, \mathbf{A}, \mathbf{S}') = \sum_{O \in Mineral} \begin{cases} 1, & O.\text{Collected}' \wedge \neg O.\text{Collected}, \\ 0, & \text{otherwise}. \end{cases} \tag{46}$$

G.3.2. Defeat-Zerglings-Banelings

The Defeat-Zerglings-Banelings (CMS) environment is also a StarCraftII mini-game. The game contains 9 marines (controlled by the player), 6 zerglings (hostile), and 4 banelings (hostile). The player needs to control the marines to deal as much damage as possible to the hostile zerglings and banelings. We illustrate the DZB environment in Figure 7(c).

We decompose the environment by 3 classes $\mathcal{C} = \{Marine, Zergling, Baneling\}$ such that

- $\mathcal{F}_s[C] = \{C.Position, C.Health, C.Alive\}$ for every $C \in \mathcal{C}$,
- $\mathcal{F}_a[Marine] = \{Marine.Move\}$.

where we set

- $Dom_{C.Position} = [-99, 99]^2$ for every $C \in \mathcal{C}$.
- $Dom_{C.Health} = [-1, 999]$ for every $C \in \mathcal{C}$.
- $Dom_{C.Alive} = \{\text{True}, \text{False}\}$ for every $C \in \mathcal{C}$.
- $Dom_{Marine.Move} = \{\text{North}, \text{East}, \text{South}, \text{West}, \text{NOOP}\}$.

We define the reward function as the total damage dealt to the hostile zerglings and banelings in each step:

$$R(\mathbf{S}, \mathbf{A}, \mathbf{S}') = \sum_{O \in Zergling} (O.\text{Health} - O.\text{Health}') + \sum_{O \in Baneling} (O.\text{Health} - O.\text{Health}'). \tag{47}$$

H. Additional Information of Experiments

H.1. Experiment Settings

In all experiments, the dynamics models are trained using offline data that is collected by a random policy that produces uniform actions. However, it should be noted that data generated during the application of the OOCDM can also contribute to further training in practice (Ding et al., 2022).

All models are trained and evaluated using one GPU (NVIDIA TITAN XP). The only exception is the causal discovery for GRADER, which is implemented by an open-source toolkit called Fast CIT and computed on 4 CPUs in parallel in our experiments. All experiments were repeated 5 times using different random seeds; the means and standard variances of the performances are reported.

H.2. Implementation of Baselines

CDL We perform causal discovery based on Theorem 3.2 and the conditional independence tests are implemented using Conditional Mutual Information (CMI), which is estimated by the model. First, each variable in $(\mathbf{S}, \mathbf{A})$ is encoded into an encoding vector. Then, in the predictor of each state variable (attribute), the encoding vectors of parental variables are aggregated by an element-wise maximum operation. Finally, the aggregated encoding is mapped to the distribution of the next-state variable by an MLP.

CDL-A Most of the parts are identical to the original CDL. However, the encoding vectors of parental variables are aggregated by attention operation instead of max-pooling. Each input variable's encoding vector is transformed into a value and a key vector, and we learn a query vector for each output variable. Key-value attention is performed to obtain the aggregated encoding of the output variable (the attention weights of non-parental variables masked to 0).

GRADER We perform causal discovery based on Theorem 3.2 and the conditional independent tests are implemented using Fast CIT (Chalupka et al., 2018). The model contains an individual predictor for each state variable (attribute), which aggregates all parents by a 2-directional Gated Recurrent Unit and then produces the distribution of the next-state variable.

TICSA This algorithm learns a probability matrix M that stores the probability of each causal edge in the BCG. To infer the next state, the model first samples the causal graph from M. Then, it masks off non-parental input features and predicts next-state variables using an MLP architecture. To learn M, the loss function includes a sparsity penalty $\|M\|_1$, and the causal graphs are sampled using Gumbel softmax (Jang et al., 2017) during the training phase.

MLP In the MLP model, all input variables are concatenated into a vector. Then, we pass this vector into a 3-layer multi-layer perceptron and obtain an embedding vector $\boldsymbol{x}$. Finally, each variable (attribute) is decoded into the posterior distribution of $\hat{p}(O.V'|\boldsymbol{S}, \boldsymbol{A})$ by applying an individual transform on $\boldsymbol{x}$.

GNN The model architecture follows the design of Structural World Model (Kipf et al., 2020). We encode objects into *object state encodings* and *object action encodings* using individual encoders. Then, we transform these object encodings via a GNN based on a complete graph, where objects correspond to the nodes: 1) We compute the edge embeddings with the state encodings of each pair of objects; 2) we compute the node embedding for each object O_i, using its state encoding, its object action encoding, and all edge embeddings of in-degrees; 3) we decode the node embeddings of O_i to the distributions of its next-state attributes.

OOFULL The model follows identical structure as described in Appendix E. However, the training loss only contains $\mathcal{L}_{\mathcal{G}_1}$ (Eq. 8) and always uses the full OOCG $\mathcal{G}_1$ in evaluation.

To make our comparison fair, we do not want these baselines to perform badly in large-scale environments simply due to insufficient model capacity. Therefore, the number of hidden units in the non-object-oriented models (MLP, GRADER, and CDL) are adjusted according to the scale of the environment, making sure that the capacity of these models is pertinent to the complexity of the environments. However, our object-orient models (OOFULL and OOCDM) have a fixed number of parameters as long as the classes are fixed, no matter how many instances are in the environment.

H.3. Out-of-Distribution Data

We construct o.o.d. data by changing the distribution of initial states of episodes, which is easy to implement in the Block and Mouse environments. However, the CMS and DZB environments are StarCraftII mini-games provided by the PySC2 platform (Vinyals et al., 2017). The platform offers limited access to the StarcraftII engine, and thus modifying the initialization process of episodes is very difficult. Therefore, we did not construct o.o.d. data for CMS and DZB.

Table 9. The total sizes (bytes) of model parameters. n denotes the number of variables in the environment, and m denotes the number of all fields. OOCDM$^+$ means the augmented OOCDM described in Appendix E.2, which is designed for asymmetric environments, i.e. AsymBlock and Walker. The model sizes of OOFULL are the same as those of OOCDM.

	n	m	GRADER	CDL	CDL-A	TICSA	GNN	MLP	**OOCDM**	**OOCDM$^+$**
Block-2	12	8	2.3M	319.7K	348.6K	184.8K	66.4K	99.4K	401.6K	-
Block-5	24	8	6.0M	1.0M	1.1M	582.3K	100.0K	224.4K	401.6K	-
Block-10	44	8	15.7M	3.1M	3.2M	2.0M	147.9K	538.0K	401.6K	-
Mouse	28	10	15.1 M	2.7M	3.0M	720.4K	432.3K	701.4K	546.6K	-
CMS	44	4	23.5M	4.3M	4.6M	1.4M	250.7K	1.1M	140.3K	-
DZB	66	10	38.8M	7.7M	8.2M	2.6M	616.3K	1.7M	549.8K	-
AsymBlock	40	8	11.0M	1.7M	1.8M	1.4M	144.3K	272.2K	401.6K	624.9K
Walker	16	8	8.1M	1.6M	1.8M	411.2K	950.7K	595.7K	410.0K	634.3K

Block To obtain the o.o.d. data, we initialize the attributes of each $Block$ object from a new distribution at the beginning of each episode:

$$(O.\mathrm{S}_1, O.\mathrm{S}_2, O.\mathrm{S}_3)^T \sim \mathcal{N}\left((0.5, 0, 0)^T, \mathrm{diag}\,(0.25, 4, 4)\right). \tag{48}$$

Mouse In the i.d. data, the attributes from the field $Monster.Noise$ are initialized from a normal distribution $\mathcal{N}(0,1)$ at the beginning of each episode. To construct the o.o.d. data, we increase the standard variance to 3. To confuse non-causal models, the initialization of $O_{food}.Amount$ is correlated to $O_{food}.Position$. During training, $O_{food}.Amount$ will be assigned with a larger value if $O_{food}.Position$ is in the east of the world; in the o.o.d. data, however, $O_{food}.Amount$ will be assigned with a larger value if $O_{food}.Position$ is in the north.

H.4. Unseen Tasks

In the Mouse environment, we sample the numbers of food, monsters, and traps respectively from $[3, 6]$, $[1, 5]$, and $[1, 5]$. Thereby, we obtain a task pool containing $4 \times 5 \times 5 = 100$ tasks. We randomly split these tasks into 47 seen tasks and 53 unseen tasks. The dynamics models are trained using offline data collected in seen tasks and then transferred into unseen tasks without further training.

H.5. Computational Costs

We provide additional results about the model size (see Table 9), and the computation time of causal discovery is shown in Table 2. These results show that our OOCDM greatly reduces the model complexity in large-scale environments. We stress that these results are for reference only, as they are affected by many factors, including the implementation details, software (we use Python and PyTorch here), and computation devices. We are also aware that some of the comparisons made here are not perfectly fair, as these CDMs perform causal discovery using different devices. However, Our approach shows the advantages of several orders of magnitude compared to GRADER, reducing the computation time from multiple hours to several seconds. Such a huge gap in these results cannot be caused solely by the differences in devices. Combining the results in the paper's Table 1, we conclude that our OOCDM uses relatively fewer parameters and the least computation time to discover the most accurate causal graphs.

H.6. Learned OOCGs of StarcraftII Mini-Games

Since the ground-truth OOCG of the CMS and DZB environments are not known, we here present the OOCGs learned by our causal discovery algorithm in Figure 10. The learned OOCGs for CMS are identical for all seeds. However, The learned OOCGs for DZB are slightly different between seeds, and thus the OOCG of the seed that produces the highest likelihood is presented.

H.7. Experiments in Asymmetric Environments

In this section, we investigate whether the requirement of dynamic symmetries (i.e., the result symmetry and causation symmetry) can be released by using the augmented architecture described in Appendix E.2. This issue is of great importance

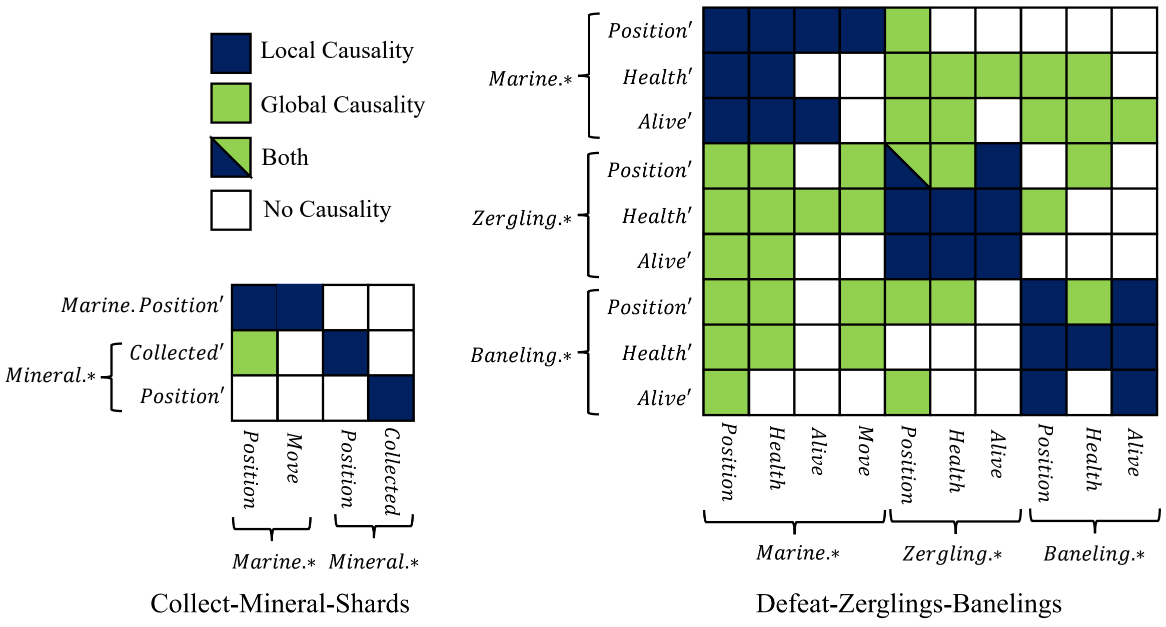


Figure 10. The visualization of the discovered OOCGs (the adjacency matrix of class-level causalities) of the CMS and DZB environments.

in extending the applicability of OOCDMs. Therefore, we perform additional experiments to test the performance of handling asymmetric environments. Here, we first introduce the environments used in the experiments and then produce the results.

H.7.1. Environments and Experiment Settings

We performed the experiments in two additional environments with asymmetric dynamics. The first environment, AsymBlock, is adapted from the Block environment. Second, we consider Walker, a robot-control task in a physics simulator, which poses a more realistic challenge for causal models.

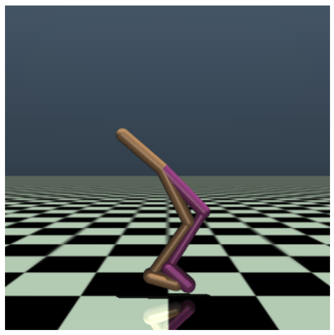
AsymBlock. We designed the **AsymBlock** Environment with similar dynamics to the Block environment, yet dynamic symmetries are violated in AsymBlock. In AsymBlock$_k$, there will be k instances of $Block$ and also k instances of $Total$. For the c-th instance of $Total$, we have

$$O_c.\mathrm{S}_j' = \frac{1}{2}O_c.\mathrm{S}_j + \frac{1}{2}\max_{O_i \in Block, i \le c} O_i.\mathrm{S}_j, \qquad j = 1,2,3;\ c = 1, \cdots, k.$$

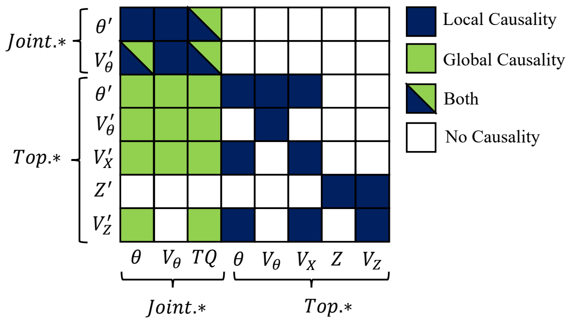
The transition of other attributes is the same as the symmetric Block environment. Causation Symmetry does not hold in the environment, as each $Total$ object summarizes a unique set of $block$ objects. Here, we set the number of blocks to be $k = 5$.

Walker. The environment is adapted from the Walker2D environment, which is based on Mujoco, a popular physics simulator in the OpenAI Gym platform (Brockman et al., 2016). Figure 11(a) gives the illustration of the environment. The agent controls the torques inflicted on the six joints (left thine, left kneel, left foot, right thine, right kneel, and right foot), in order to make the robot stand up and move forward. The environment is originally factored, and a detailed introduction can be found in the official document [1]. We represent the environment as an OOMDP containing two classes $Joint$ (which has 6 instances) and Top (which has one instance only). The fields of a joint include its angle $Joint.\theta$, angular velocity $Joint.V_\theta$, and the torque $Joint.TQ$ that the agent inflicted. The fields of the top instance include its angle $Top.\theta$, angular velocity $Top.V_\theta$, horizontal velocity $Top.V_x$, vertical position $Top.Z$, and vertical velocity $Top.V_z$. The agent is rewarded for a positive horizontal velocity of the top and every step that the robot stays alive. Moreover, a punishment is given as to the cost of the action. Since the specific topology of the skeleton is not described by the attributes of joints, causation symmetry does not hold in the environment.

[1]https://gymnasium.farama.org/environments/mujoco/walker2d/



(a)



(b)

Figure 11. The (a) visualization and (b) discovered OOCG of the Walker environment.

Table 10. The average instance log-likelihoods on asymmetric environments. The "+" superscript for OOFULL and OOCDM means using the augmented architecture in Appendix E.2 to handle the asymmetric dynamics. We do not show the standard variances for obviously over-fitting results (log-likelihoods less than −100.0, highlighted in brown).

Env	data	GRADER	CDL	CDL-A	TICSA	GNN	MLP	OOFULL	OOCDM	OOFULL$^+$	OOCDM$^+$
AsymBlock	train	19.7±0.6	16.4±1.4	18.6±1.8	15.7±1.0	15.7±0.3	8.0±2.7	18.7±0.3	16.4±2.5	19.6±0.9	**20.0**±0.3
	i.d.	−1.0±9.9	−7.9±12.2	−1.5e6	−3.8e4	13.9±2.3	−32.5±57.2	15.7±0.5	15.2±2.4	5.7±28.0	**19.7**±0.5
	o.o.d.	−895.5	−130.1	−1.8e8	−1.7e7	−2.7e5	−3.8e6	−14.2±23.9	−34.0±18.0	−43.1±94.7	**13.8**±6.7
Walker	train	8.0±0.2	7.5±0.8	8.6±0.2	7.0±1.7	3.5±0.2	10.0±0.3	6.9±0.3	6.0±0.4	**10.8**±1.0	10.3±0.6
	i.d.	7.7±0 .1	7.4±0.7	8.6±0.2	2.1±1.7	3.2±0.2	9.5±0.3	6.6±0.4	5.9±0.3	**10.7**±1.0	10.2±0.5
	o.o.d.	5.8±1.3	6.1±0.8	6.6±0.5	−6.6±10.8	3.6±0.2	6.1±0.9	6.4±0.6	5.5±0.6	8.0±1.6	**9.0**±0.3

Table 11. The causal discovery times (seconds) on the asymmetric environments. The "+" superscript for OOCDM means using the augmented architecture to handle asymmetric dynamics.

Env	n	m	GRADER	CDL	CDL-A	OOCDM	OOCDM$^+$
AsymBlock	40	8	4789.6±177.4	19.1±5.7	19.1±2.8	**2.1**±0.1	2.4±0.2
Walker	16	8	7957.3±444.9	58.8±7.4	62.9±11.6	**21.5**±0.1	24.5±3.0

The action space of Walker is composed of the torques on six joints. Such a multi-variable action space poses a great challenge for a random policy to sufficiently explore the space in this task. Therefore, we train the models using the data produced by a policy-based collector, which is trained using the PPO algorithm (Schulman et al., 2017). The training and in-distribution data is collected from a weak policy, which has gone through a few PPO updates. Due to the large action space, finding a good action sequence is difficult for our CEM-based planning algorithm. Therefore, the planning performance is not presented here. We test whether the model can generalize to the out-of-distribution data produced by a stronger collector (trained with more PPO updates) instead of the planning policy.

H.7.2. Results

The results of prediction accuracy are given in Table 10. In particular, we use the "+" superscript for OOFULL and OOCDM to signify that the architectures are modified according to Appendix E.2 to handle the asymmetric dynamics. According to these results, the augmented OOCDM and OOFULL perform well in capturing the unique dynamics of each object, leading to significant improvement against the symmetric version. However, OOCDM$^+$ shows better generalization ability than OOFULL$^+$, which demonstrates that the learned OOCG can help reduce spurious correlations. Since the ground-truth causal graph is not an OOCG in asymmetric environments, we do not compare the accuracy of causal graphs. However, the results show that the learned OOCG is helpful with the generalization ability. In AsymBlock where the causal dependencies are inherently sparse, OOCDM$^+$ raises the best performance. In Walker where dependencies are inherently denser, OOFULL$^+$ shows the best performance in the training and in-distribution data, whereas OOCDM$^+$ best generalizes to o.o.d. data. Here, Figure 11(b) shows the discovered OOCG using OOCDM$^+$.

Table 12. The accuracy (in percentage) of discovered causal graphs in Block$_5$ using noisy data.

data	GRADER	CDL	CDL-A	TICSA	OOCDM
original	$94.0_{\pm 1.5}$	$97.5_{\pm 1.5}$	$99.3_{\pm 0.6}$	$96.3_{\pm 0.6}$	$100.0_{\pm 0.0}$
noisy	$94.4_{\pm 0.7}$	$96.8_{\pm 0.8}$	$99.5_{\pm 0.5}$	$95.7_{\pm 3.0}$	$100.0_{\pm 0.0}$

Table 13. The average instance log-likelihoods of the dynamics models (trained with noisy data) on various datasets in Block$_5$. We do not show the standard variances for obviously over-fitting results (less than -100.0, highlighted in brown).

data	GRADER	CDL	CDL-A	TICSA	GNN	MLP	OOFULL	**OOCDM**
train (noisy)	$15.5_{\pm 0.6}$	$12.5_{\pm 1.1}$	$16.0_{\pm 0.2}$	$12.2_{\pm 1.3}$	$13.5_{\pm 0.4}$	$8.0_{\pm 0.3}$	$\mathbf{16.5}_{\pm 0.8}$	$15.7_{\pm 0.9}$
i.d.	$2.2_{\pm 5.1}$	$12.3_{\pm 1.2}$	$-5.6e6_{\pm}$	$-2.2e5_{\pm}$	$13.9_{\pm 0.4}$	$-13.9_{\pm 9.2}$	$-25.3_{\pm 79.5}$	$17.5_{\pm 0.5}$
o.o.d.	-516.8	$-3.2e7$	$-4.2e8$	$-5.2e7$	$-12.7_{\pm 18.9}$	$-1.7e4$	$-2.7e7$	$-3.2_{\pm 12.3}$

Table 11 presents the computational time of causal discovery in these environments. The sample sizes for causal discovery are 10,000 and 100,000 for AsymBlock and Walker, respectively. The only exception is GRADER in Walker, which uses only 10,000 samples, otherwise the causal discovery would take excessive time to complete. From these results, we notice that the augmented OOCDM does not result in a significant increase in the computation time of causal discovery. Additionally, Table 9 includes the model size of the augmented model in the Asymblock and Walker environments.

These additional experiments demonstrate that the augmented OOCDM may handle environments where result and causation symmetries do not hold. The object-oriented causal discovery remains computationally efficient and is helpful in improving the generalization performance. Therefore, we conclude that the requirement of dynamic symmetries can be released by using the augmented OOCDM. On the one hand, OOCDM has better generalization ability resulting from an approximate causal graph, compared to dense dynamics models. On the other hand, learning an OOCDM is more computationally efficient than learning existing CDMs, which may not be tractable in large-scale environments.

I. Robustness on Noisy Data

Theorem C.13 implies that causal discovery using CITs is robust against limited noise. In this section, we present the performance of dynamics models in the Block$_5$ environment, using the data added with independent Gaussian noises. The scale (i.e. the standard variance) of observational noises is 0.01, which is equal to the scale of the transitional noises.

The causal graph accuracy (percentage) is given in Table 12. The results show that causal discovery is robust to the noisy data. Apart from CDL, all CDMs have almost identical performance. We also present the average-instance log-likelihood of models learned from noisy data in Table 13. The test data contains no noise, so it is possible for models to perform better on the test data. Prediction Accuracy decreases due to the noise for all models. While the decrease is reasonable on the training and test data, the risk of overfitting significantly rises for CDMs on o.o.d. data. Even with oracle causal parents, noise can lead to generalization errors within the learned structural equations. However, the performance on o.o.d. data of OOCDM remains competitive among all CDMs.

I.1. Hyper-Parameters

Main hyper-parameters are listed in Table 14, and more details are contained in our code.

J. Weaknesses and Future Works

A weakness of this work is the requirement of domain knowledge to formulate environments as OOMDPs. Although using objects and classes to describe the world is natural, intuitive formulation may violate result symmetry and causation symmetry. As mentioned in Section 2.2, many studies have investigated the learning of object-centric representation. However, extracting OOP-style representation (i.e. involving multiple classes) remains an open problem, especially when Eq. 4 needs to be satisfied. Therefore, future work will investigate how to extract properly-categorized objects from raw observations. Meanwhile, more effective methods to release result and causation symmetries should be further explored, where modeling relational interactions from raw factorization may be a potential direction.

Table 14. The main hyper-parameters used in our experiments.

		Values for environments					
Symbol	Meaning	Block	Mouse	CMS	DZB	AsymBlock	Walker
d_e	The dimension of attribution-encoding vectors	16	16	16	16	16	16
d_k	The dimension of key vectors	32	32	32	32	32	32
d_v	The dimension of value vectors	32	32	32	32	32	32
d_h	The dimension of the hidden encoding $\boldsymbol{h}_i$ for each object	-	-	-	-	64	64
ε	The threshold of CMIs in causal discovery	0.3	0.1	0.2	0.03	0.3	0.1
n_{plan}	The number of planning iteration in CEM	-	5	5	5	5	5
H	The planning horizon in MPC	-	20	20	20	20	20
α	The weight of $\mathcal{L}_{\mathcal{G}_1}$ in the target function	1	1	1	1	1	1
β	The weight of $\mathcal{L}_{\hat{\mathcal{G}}}$ in the target function	1	1	1	1	1	1
λ	The probability of dependencies when sampling $\mathcal{G}_\lambda$	0.9	0.9	0.9	0.9	0.9	0.9
γ	The discount of rewards in MPC	-	0.95	0.95	0.95	-	-
	The number of samples in CEM	-	500	500	500	-	-
	The number of elite samples in CEM	-	100	100	100	-	-
n_{iter}	The number of iteration in training	50	200	80	200	75	60
n_{batch}	The number of batches in each iteration	1000	1000	1000	1000	1000	500

Another weakness of this work is that FMDP imposes strong constraints that may not hold in more complicated tasks involving confounders, partial observability, or non-Markovian dynamics. Addressing these challenges in an object-oriented framework is important to extend the applicability of our approach. Therefore, we propose to explore these directions using more rigorous tools of causality in the future.