Generalizing Goal-Conditioned Reinforcement Learning with Variational Causal Reasoning

Wenhao Ding[1], **Haohong Lin**[1], **Bo Li**[2], **Ding Zhao**[1]
[1]Carnegie Mellon University
[2]University of Illinois Urbana-Champaign
{wenhaod, haohongl}@andrew.cmu.edu, lbo@illinois.edu, dingzhao@cmu.edu

Abstract

As a pivotal component to attaining generalizable solutions in human intelligence, reasoning provides great potential for reinforcement learning (RL) agents' generalization towards varied goals by summarizing part-to-whole arguments and discovering cause-and-effect relations. However, how to discover and represent causalities remains a huge gap that hinders the development of causal RL. In this paper, we augment Goal-Conditioned RL (GCRL) with *Causal Graph (CG)*, a structure built upon the relation between objects and events. We novelly formulate the GCRL problem into variational likelihood maximization with CG as latent variables. To optimize the derived objective, we propose a framework with theoretical performance guarantees that alternates between two steps: using interventional data to estimate the posterior of CG; using CG to learn generalizable models and interpretable policies. Due to the lack of public benchmarks that verify generalization capability under reasoning, we design nine tasks and then empirically show the effectiveness of the proposed method against five baselines on these tasks. Further theoretical analysis shows that our performance improvement is attributed to the virtuous cycle of causal discovery, transition modeling, and policy training, which aligns with the experimental evidence in extensive ablation studies.

1 Introduction

The generalizability, which enables an algorithm to handle unseen tasks, is fruitful yet challenging in multifarious decision-making domains. Recent literature [1, 2, 3] reveals the critical role of reasoning in improving the generalization of reinforcement learning (RL). However, most off-the-shelf RL algorithms [4] have not regarded reasoning as an indispensable accessory, thus usually suffer from data inefficiency and performance degradation due to the mismatch between training and testing settings. To attain generalization at the testing stage, some efforts were put into incorporating domain knowledge to learn structured information, including sub-task decomposition [5], or program generation [6, 7, 8] and execution [9, 10], which guide the model to solve complicated tasks in an explainable way. However, such symbolism-dominant methods heavily depend on the re-usability of sub-tasks and pre-defined grammars, which may not always be accessible in decision-making tasks.

Inspired by the close link between reasoning and the cause-and-effect relationship, causality is recently incorporated to compactly represent the aforementioned structured knowledge in RL training [11]. Based on the form of causal knowledge, we divide the related works into two categories, i.e., *implicit* and *explicit* causation. With *implicit* causal representation, researchers ignore the detailed causal structure. For instance, [12] extracts invariant features as one node that influences the reward function, while the other node consists of task-irrelevant features [13, 14, 15, 16]. This neat structure has good scalability but requires access to multiple environments that share the same invariant feature [12, 16, 17]. One can also turn to the *explicit* side by estimating detailed causal structures [18, 19, 20, 21],

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

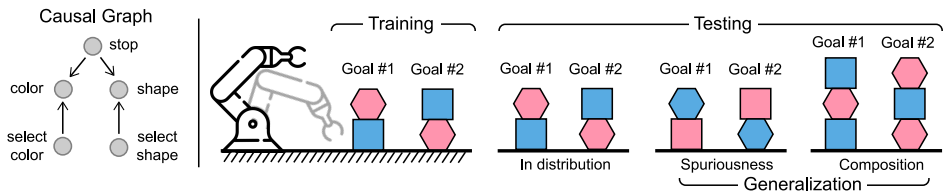

Figure 1: The agent picks and places objects to achieve given goals. **Left:** The causal graph of this task. **Right:** Training setting: the hexagon is always pink, and the box is always blue. Three testing settings: (1) In distribution: the same as the training setting. (2) Spuriousness: swap the color and shape to break the spurious correlation. (3) Composition: increase the height of the goal.

which uses directed graphical models to capture the causality in the environment. A pre-request for this estimation is the object-level or event-level abstraction of the observation, which is available in most tasks and also becoming a frequently studied problem [22, 23, 24]. However, existing *explicit* causal reasoning RL models either rely on heuristic design without theoretical guarantees [18] or require the true causal graph [25].

In this paper, we propose ***G**ene**RA**lizing by **D**iscov**ER**ing (GRADER)*, a causal reasoning method that augments the RL algorithm with data efficiency, interpretability, and generalizability. We mainly focus on Goal-Conditioned RL (GCRL), where different goal distributions during training and testing reflect the generalization. We formulate the GCRL into a probabilistic inference problem [26] with a learnable causal graph as the latent variable. This novel formulation naturally explains the learning objective with three components, i.e., transition model learning, planning, and causal graph discovery, leading to an optimization framework that alternates between the causal structure discovery policy learning to gain generalizability. Under some mild conditions, we prove the unique identifiability of the causal graph and the theoretical performance guarantee of the proposed framework.

To demonstrate the effectiveness of the proposed method, we conduct comprehensive experiments in environments that require strong reasoning capability. Specifically, we design two types of generalization settings, i.e., spuriousness and composition, and provide an example to illustrate these settings in Figure 1. The evaluation results confirm our method's advantages in several aspects. First, the proposed data-efficient discovery method provides an explainable causal graph yet requires much less data than score-based methods, increasing data efficiency and interpretability during task solving. Second, simultaneously discovering the causal graph during policy learning dramatically increases the success rate of solving tasks. In summary, the contribution of this paper is threefold:

- We use the causal graph as a latent variable to reformulate the GCRL problem and then derive an iterative training framework from solving this problem.
- We prove that our method uniquely identifies true causal graphs, and the performance of the iterative optimization is guaranteed with a lower bound given converged transition dynamics.
- We design nine tasks in three environments that require strong reasoning capability and show the effectiveness of the proposed method against five baselines on these tasks.

2 Problem Formulation and Preliminary

We start by discussing the setting of RL we use in this paper and the assumptions required in causal reasoning. Then we briefly introduce the necessary concepts related to causality and causal discovery.

2.1 Factorized Goal-conditioned RL

We assume the environment follows the Goal-conditioned Markov Decision Process (MDP) setting with full observation. This setting is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, G)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the probabilistic transition model, $G \subset \mathcal{S}$ is the goal space which is a set of assignment of values to states, and $r(s, g) = \mathbb{1}(s = g) \in \mathcal{R}$ is the sparse deterministic reward function that returns 1 only if the state s match the goal g. In this paper, we focus on the goal-conditioned generalization problem, where the goal for training and testing stages will be sampled from different distributions $p_{\text{train}}(g)$ and $p_{\text{test}}(g)$. We refer to a goal $g \in G$ as a task and use these two terms interchangeably. To accomplish the causal discovery methods, we make a further assumption similar to [20, 27] for the state and action space:

Assumption 1 (Space Factorization)**.** *The state space $\mathcal{S} = \{\mathcal{S}_1 \times \cdots \times \mathcal{S}_M\}$ and action space $\mathcal{A} = \{\mathcal{A}_1 \times \cdots \times \mathcal{A}_N\}$ can be factorized to disjoint components $\{\mathcal{S}_i\}_{i=1}^M$ and $\{\mathcal{A}_i\}_{i=1}^N$.*

The components representing one event or object's property usually have explicit semantic meanings for better interpretability. This assumption can be satisfied by state and action abstraction, which has been widely investigated in [22, 23, 24]. Such factorization also helps deal with the high-dimensional states since it might be impossible to treat each dimension as one random variable [18].

2.2 Causal Reasoning with Graphical Models

Reasoning with causality relies on specific causal structures, which is commonly represented as directed acyclic graphs (DAGs) [28] over variables. Consider random variables $\boldsymbol{X} = (X_1, \ldots, X_d)$ with index set $\boldsymbol{V} := \{1, \ldots, d\}$. A graph $\mathcal{G} = (\boldsymbol{V}, \mathcal{E})$ consists of nodes $\boldsymbol{V}$ and edges $\mathcal{E} \subseteq \boldsymbol{V}^2$ with (v, v) for any $v \in \boldsymbol{V}$. A node i is called a parent of j if $e_{ij} \in \mathcal{E}$ and $e_{ji} \notin \mathcal{E}$. The set of parents of j is denoted by $\mathbf{PA}_j^{\mathcal{G}}$. We formally discuss the graph representation of causality with the definitions:

Definition 1 (Structural Causal Models [28])**.** *A structural causal model (SCM) $\mathfrak{C} := (\boldsymbol{S}, \boldsymbol{U})$ consists of a collection $\boldsymbol{S}$ of d functions,*

$$X_j := f_j(\boldsymbol{PA}_j^{\mathcal{G}}, U_j), \quad j \in [d], \tag{1}$$

where $\boldsymbol{PA}_j \subset \{X_1, \ldots, X_d\} \backslash \{X_j\}$ are called parents of X_j; and a joint distribution $\boldsymbol{U} = \{U_1, \ldots, U_d\}$ over the noise variables, which are required to be jointly independent.

Definition 2 (Causal Graph [28], CG)**.** *The causal graph $\mathcal{G}$ of an SCM is obtained by creating one node for each X_j and drawing directed edges from each parent in $\boldsymbol{PA}_j^{\mathcal{G}}$ to X_j.*

We note that CG describes the structure of the causality, and SCM further considers the specific causation from the parents of X_j to X_j as well as exogenous noises U_j. To uncover the causal structure from data distribution, we assume that the CG satisfies the *Markov Property* and *Faithfulness* ([28]), which makes the independences consistent between the joint distribution $P(X_1, \ldots, X_n)$ and the graph $\mathcal{G}$. We also follow the *Causal Sufficiency* assumption [29] that supposes we have measured all the common causes of the measured variables.

Existing work [30, 20] believes that two objects have causation only if they are close enough while there is no edge between them if the distance is large. Instead of using such a local view of the causality, we assume the causal graph is consistent across all time steps, which also handles the local causal influence. The specific influence indicated by edges is estimated by the function $f_j(\mathbf{PA}_j, U_j)$.

3 Generalizing by Discovering (GRADER)

With proper definitions and assumptions, we now look at the proposed method. We first derive the framework of GRADER by formulating the GCRL problem as a latent variable model, which provides a variational lower bound to optimize. Then, we divide this objective function into three parts and iteratively update them with a specific performance guarantee.

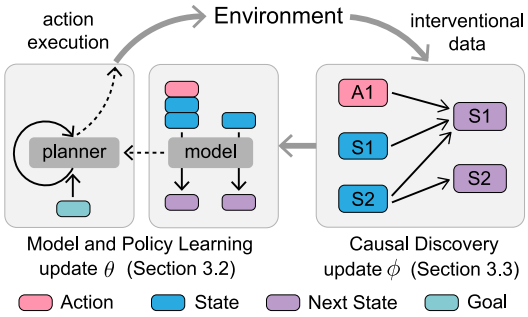

Figure 2: The paradigm of GRADER.

3.1 GCRL as Latent Variable Models

The general objective of RL is to maximize the expected reward function w.r.t. a learnable policy model π. Particularly, in the goal-conditioned setting, such objective is represented as $\max_\pi \mathbb{E}_{\tau \sim \pi, g \sim p(g)}[\sum_{t=0}^T r(s^t, g)]$, where $p(g)$ is the distribution of goals and $\tau := \{s^0, a^0, \ldots, s^T\}$ the action-state trajectory with maximal time step T. The trajectory ends only if the goal is achieved $g = s^T$ or the maximal step is reached.

Inspired by viewing the RL problem as probabilistic inference [31, 26], we replace the question from *"How to find actions to achieve the goal?"* to *"what are the actions if we achieve the goal?"*, leading to a likelihood maximization problem for $p(\tau | s^*)$ with $s^* := \mathbb{1}(g = s^T)$. Different from previous

work [32] that recasts actions as latent variables and infers actions that result in "observed" high reward, we decompose $p(\tau|s^*)$ with $\mathcal{G}$ as the latent variable to get the evidence lower bound (ELBO)

$$\log p(\tau|s^*) = \log \int p(\tau|\mathcal{G}, s^*)p(\mathcal{G}|s^*)d\mathcal{G} \geq \mathbb{E}_{q(\mathcal{G}|\tau)}[\log p(\tau|\mathcal{G}, s^*)] - \mathbb{D}_{\mathrm{KL}}[q(\mathcal{G}|\tau)||p(\mathcal{G})], \quad (2)$$

where the prior $p(\mathcal{G})$ and variational posterior $q(\mathcal{G}|\tau)$ represent distributions over a graph structure, i.e, the probability of the existences of edges in the graph. $\mathbb{D}_{\mathrm{KL}}$ represents the Kullback–Leibler (KL) divergence between two graphs, which will be explained in Section 3.3. Recall that the space of goal G is a subset of the state, here we extend the meaning of g and assume all trajectories achieve the goal in the final state [33], i.e., $g = s^T$. Such extension makes it possible to further decompose the first term of (2) as (refer to Appendix A.1.2)

$$\log p(\tau|\mathcal{G}, s^*) = \log p(s^0) + \sum_{t=0}^{T-1} \log p(s^{t+1}|s^t, a^t, \mathcal{G}) + \sum_{t=0}^{T-1} \log \pi(a^t|s^t, s^*, \mathcal{G}) + \log p(g). \quad (3)$$

Here, we use the fact that $\mathcal{G}$ is effective in both the transition model $p(s^{t+1}|a^t, s^t, \mathcal{G})$ and the policy model $\log p(a^t|s^t, s^*, \mathcal{G})$, g only influences the policy model, and the initial state s_0 depends neither on $\mathcal{G}$ nor g. We assume that both initial state $\log p(s_0)$ and goal $\log p(g)$ follow the uniform distribution; thus, the first and last terms of (3) are constants. The policy term π, selecting action a^t according to both current state s^t and goal g, is implemented with the planning method and is further discussed in Section 3.2. Finally, we maximize the likelihood $p(\tau|s^*)$ with the following reformulated ELBO as the objective

$$\mathcal{J}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathcal{G}|\tau)} \sum_{t=0}^{T-1} \left[\log p_\theta(s^{t+1}|s^t, a^t, \mathcal{G}) + \log \pi_\theta(a^t|s^t, s^*, \mathcal{G})\right] - \mathbb{D}_{\mathrm{KL}}[q_\phi(\mathcal{G}|\tau)||p(\mathcal{G})] \quad (4)$$

where θ is the shared parameter of transition model $p_\theta(s^{t+1}|a^t, s^t, \mathcal{G})$ and policy $\pi_\theta(a^t|s^t, s^*, \mathcal{G})$, and ϕ is the parameter of causal graph $q_\phi(\mathcal{G}|\tau)$. To efficiently solve this optimization problem, we iteratively updates parameter ϕ (causal discovery, Section 3.3) and parameter θ (model and policy learning, Section 3.2), as shown in Figure 2. Intuitively, these processes can be respectively viewed as the discovery of CG and update of SCM, which share tight connections discussed in Section 2.2.

3.2 Model and Policy Learning

Let us start with a simple case where we already obtain a $\mathcal{G}$ and use it to guide the learning of parameter θ via $\max_\theta \mathcal{J}(\theta, \phi)$. Since the KL divergence of $\mathcal{J}(\theta, \phi)$ does not involve θ, we only need to deal with the first expectation term, i.e., the likelihood of transition model and policy. For the transition $p_\theta(s^{t+1}|a^t, s^t, \mathcal{G})$, we connect it with causal structure by further defining a particular type of CG and denote it as $\mathcal{G}$ in the rest of this paper:

Definition 3 (Transition Causal Graph)**.** *We define a bipartite graph $\mathcal{G}$, whose vertices are divided into two disjoint sets $\mathcal{U} = \{\mathcal{A}^t, \mathcal{S}^t\}$ and $\mathcal{V} = \{\mathcal{S}^{t+1}\}$. $\mathcal{A}^t$ represents action nodes at step t, $\mathcal{S}$ state nodes at step t, and $\mathcal{S}^{t+1}$ the state nodes at step $t+1$. All edges start from set $\mathcal{U}$ and end in set $\mathcal{V}$.*

Model learning. This definition builds the causal graph between two consecutive time steps, which indicates that the values of states in step $t+1$ depend on values in step t. It also implies that the interventions [28] on nodes in $\mathcal{U}$ are directly obtained since they have no parents. We denote the marginal distribution of $\mathcal{S}$ as $p_{\mathcal{I}_\pi^s}$, which is collected by RL policy π. Combined with the Definition 1 of SCM, we find that $p_\theta(s^{t+1}|a^t, s^t, \mathcal{G})$ essentially approximates a collection of functions f_j following the structure $\mathcal{G}$, which take as input the values of parents of the state node s_j and outputs the value s_j. Thus, we propose to model the transition corresponding to $\mathcal{G}$ with a collection of neural networks $f_\theta(\mathcal{G}) := \{f_{\theta_j}\}_{j=1}^M$ to obtain

$$s_j^{t+1} = f_{\theta_j}([\mathbf{PA}_j^{\mathcal{G}}]^t, U_j), \quad (5)$$

where $[\mathbf{PA}_j^{\mathcal{G}}]^t$ represents the values of all parents of node s_j^t at time step t and U_j follows Gaussian noise $U_j \sim \mathcal{N}(0, \mathbf{I})$. In practice, we use Gated Recurrent Unit [34] as f_j because it supports varying numbers of input nodes. We take s_j^t as the initial hidden embedding and the rest parents $[\mathbf{PA}_j^{\mathcal{G}} \backslash s_j]^t$ as the input sequence to f_j. The entire model is optimized by stochastic gradient descent with the log-likelihood $\log p_\theta(s^{t+1}|a^t, s^t, \mathcal{G})$ as objective.

Policy learning with planning. Then we turn to the policy term $\pi_\theta(a^t|s^t, s^*, \mathcal{G})$ in $\mathcal{J}(\theta, \phi)$. We optimize it with planning methods that leverage the estimated transition model. Specifically, the policy aims to optimize an action-state value function $Q(s^t, a^t) = \mathbb{E}\left[\sum_{t'=0}^{H} \gamma^{t'} r(s^{t'+t}, a^{t'+t})|s^t, a^t\right]$, which can be obtained by unrolling the transition model with a horizon of H steps and discount factor γ. In practice, we use model predictive control (MPC) [35] with random shooting [36], which selects the first action in the fixed-horizon trajectory that has the highest action-state value $Q(s^t, a^t)$, i.e. $\hat{\pi}(s^t) = \arg\max_{a^t \in \mathcal{A}} Q_\theta^{\mathcal{G}}(s^t, a^t)$. The formulation we derived so far is highly correlated to the model-based RL framework [37]. However, the main difference is that we obtain it with variational inference by regarding the causal graph as a latent variable.

3.3 Data-Efficient Causal Discovery

In this step, we relax the assumption of knowing $\mathcal{G}$ and aim to estimate the posterior distribution $q(\mathcal{G}|\tau)$ to increase the ELBO (4) w.r.t. parameter ϕ. In most score-based methods [38], likelihood is used to evaluate the correctness of the causal graph, i.e., a better causal graph leads to a higher likelihood. Since the first term of (4) represents the likelihood of the transition model, we convert the problem of $\max_\phi \mathcal{J}(\theta, \phi)$ to the causal discovery that finds the true causal graph based on collected data samples. As for the second term of (4), the following proposition shows that the KL divergence between $q_\phi(\mathcal{G}|\tau)$ and $p(\mathcal{G})$ can be approximated by a sparsity regularization (proof in Appendix A.4.2).

Proposition 1 (KL Divergence as Sparsity Regularization)**.** *With entry-wise independent Bernoulli prior $p(\mathcal{G})$ and point mass variational distribution $q(\mathcal{G}|\tau)$ of DAGs, $\mathbb{D}_{KL}[q_\phi \| p]$ is equivalent to an ℓ_1 sparsity regularization for the discovered causal graph.*

Algorithm 1: GRADER Training

Input: Trajectory buffer $\mathcal{B}_\tau$, Causal graph $\mathcal{G}$, Transition model f_θ
while θ *not converged* **do**
  `// Policy from planning`
  Sample a goal $g \sim p_{\text{train}}(g)$
  while $t < T$ **do**
    $a^t \leftarrow$ Planner(f_θ, s^t, g)
    $s^{t+1}, r^t \leftarrow$ Env(a^t, g)
    $\mathcal{B}_\tau \leftarrow \mathcal{B}_\tau \cup \{a^t, s^t, s^{t+1}\}$
  `// Estimate causal graph`
  for $i \leq M + N$ **do**
    for $j \leq M$ **do**
      Infer edge $e_{ij} \leftarrow q_\phi(\cdot|\mathcal{B}, \eta)$
  `// Learn transition model`
  Update $f_\theta(\mathcal{G})$ via (5) with $\mathcal{B}$

We restrict the posterior $q_\phi(\mathcal{G}|\tau)$ to point mass distribution and use a threshold η to control the sparsity. We perform the discovery process from the classification perspective by proposing binary classifiers $q_\phi(e_{ij}|\tau, \eta)$ to determine the existence of an edge e_{ij}. This classifier $q_\phi(e_{ij}|\tau, \eta)$ is implemented by statistic *Independent Test* [39], and η is the threshold for the p-value of the hypothesis. A larger η corresponds to harder sparsity constraints, thus leading to a sparse $\mathcal{G}$ since two nodes are more likely to be considered independent. According to the definition 3, we only need to conduct classification to edges connecting nodes between $\mathcal{U}$ and $\mathcal{V}$. If two nodes are dependent, we add one edge directed from the node in $\mathcal{U}$ to the node in $\mathcal{V}$. This definition also ensures that we always have $q(\mathcal{G}|\tau) \in \mathcal{Q}_{DAG}$, where $\mathcal{Q}_{DAG}$ is the class of DAG. With this procedure, we identify a unique CG $\mathcal{G}^*$ under optimality:

Proposition 2 (Identifiability)**.** *Given an oracle independent test, with an optimal interventional data distribution $p^*_{\mathcal{I}^s_\pi}$, causal discovery obtains ϕ^* that correctly tells the independence between any two nodes, then the causal graph is uniquely identifiable, with $e^*_{ij} = q_{\phi^*}(e_{ij}|\tau), \forall i \in [M+N], j \in [M]$.*

In practice, we use χ^2-test for discrete variables and the Fast Conditional Independent Test [39] for continuous variables. The sample size needed for obtaining the oracle test has been widely investigated [40]. However, testing with finite data is not a trivial problem, as stated in [41], especially when the data is sampled from a Goal-conditioned MDP. Usually, the random policy is not enough to satisfy the oracle assumption because some nodes cannot be fully explored when the task is complicated and has a long horizon. To make this assumption empirically possible, it is necessary to simultaneously optimize $\pi_\theta(a^t|s^t, s^*, \mathcal{G})$ to access more samples close to finishing the task, which is further analyzed in Section 3.4. We also empirically support this argument in Section 4.2 and provide a detailed theoretical proof in Appendix A.3 and A.4.

3.4 Analysis of Performance Guarantee

The entire pipeline of GRADER is summarized in Algorithm 1. To analyze the performance of the optimization of (4), we first list important lemmas that connect previous steps and then show that the iteration of these steps in *GRADER* leads to a virtuous cycle.

By the following lemmas, we show the following performance guarantees step by step. Lemma 1 shows model learning is monotonically better at convergence given a better causal graph from causal discovery. Then the learned transition model helps to improve the lower bound of value function during planning according to Lemma 2. Lemma 3 reveals the connection between policy learning and interventional data distribution, which in turn improves the quality of our causal discovery, as is shown in Lemma 4 and Proposition 2.

Lemma 1 (Monotonicity of Transition Likelihood)**.** *Assume $\mathcal{G}^* = (V, E^*)$ be the true CG, for two CG $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$, if SHD$(\mathcal{G}_1, \mathcal{G}^*) <$ SHD$(\mathcal{G}_2, \mathcal{G}^*)$, $\exists\, e,\ s.t.\ E_1 \cup \{e\} = E_2$, when transition model θ converges, the following inequality holds for the transition model in (4):*

$$\log p_\theta(s^{t+1}|a^t, s^t, \mathcal{G}^*) \geq \log p_\theta(s^{t+1}|a^t, s^t, \mathcal{G}_1) \geq \log p_\theta(s^{t+1}|a^t, s^t, \mathcal{G}_2) \tag{6}$$

where SHD is the Structural Hamming Distance defined in Appendix A.2.

Lemma 2 (Bounded Value Function in Policy Learning)**.** *Given a planning horizon $H \to \infty$, if we already have an approximate transition model $\mathbb{D}_{TV}(\hat{p}(s'|s,a), p(s'|s,a)) \leq \epsilon_m$, the approximate policy $\hat{\pi}$ achieves a near-optimal value function (refer to Appendix A.3 for detailed analysis):*

$$\|V^{\pi^*}(s) - V^{\hat{\pi}}(s)\|_\infty \leq \frac{\gamma}{(1-\gamma)^2}\epsilon_m \tag{7}$$

Lemma 3 (Policy Learning Improves Interventional Data Distribution)**.** *With a step reward $r(s,a) = \mathbb{1}(s = g)$, we show that the value function determines an upper bound for TV divergence between the interventional distribution with its optimal (proof details in Appendix A.3):*

$$\mathbb{D}_{TV}(p_{\mathcal{I}^s_\pi}, p_g) \leq 1 - (1-\gamma)V^\pi(s). \tag{8}$$

where $p_{\mathcal{I}^s_\pi}$ is the marginal state distribution in interventional data, and p_g is the goal distribution. A better policy with larger $V^\pi(s)$ enforces the distribution of interventional data towards the goal.

Lemma 4 (Interventional Data Benefits Causal Discovery)**.** *For $\epsilon_g = \min_{p_g > 0} p_g$, $\mathbb{D}_{TV}(p_{\mathcal{I}^s_\pi}, p_g) < \epsilon_g$, the error of our causal discovery is upper bounded with $\mathbb{E}_{\hat{\mathcal{G}}}[SHD(\hat{\mathcal{G}}, \mathcal{G}^*)] \leq |\mathcal{S}| - 1$.*

After the close-loop analysis of our model, we are now able to analyze the overall performance of the proposed framework. Under the construction of $p_\theta(s^{t+1}|a^t, s^t, \mathcal{G})$ with NN-parameterized functions, the following theorem shows that the learning process will guarantee to perform a close estimation of true ELBO under the iterative optimization among model learning, planning, and causal discovery.

Theorem 1. *With T-step approximate transition dynamics $\mathbb{D}_{TV}\Big(\hat{p}(s'|s,a), p(s'|s,a)\Big) \leq \epsilon_m$, if the goal distribution satisfies $\epsilon_g > \frac{\gamma}{1-\gamma}\epsilon_m$, and the distribution prior CG is entry-wise independent Bernoulli$(\epsilon_{\mathcal{G}})$, GRADER guarantees to achieve an approximate ELBO $\hat{\mathcal{J}}$ with the true ELBO $\mathcal{J}^*$:*

$$\|\mathcal{J}^*(\theta, \phi) - \hat{\mathcal{J}}(\hat{\theta}, \hat{\phi})\|_\infty \leq \left[1 + \frac{\gamma}{(1-\gamma)^2}\right]\epsilon_m T + \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right)(|\mathcal{S}| - 1), \tag{9}$$

An intuitive understanding of the performance guarantee is that a better transition model indicates a better approximation of objective $\mathcal{J}$. Yet it's harder for *GRADER* to approximate a longer trajectory with large state space. The proof of this theorem and corresponding empirical results are in Appendix A.5.

4 Experiments

In this section, we first discuss the setting of our designed environments as well as the baselines used in the experiments. Then, we provide the numerical results and detailed discussions to answer the following important research questions: **Q1.** Compared to baselines, how does GRADER gain performance improvement under both in-distribution and generalization settings? **Q2.** Compared to an offline random policy, how does a well-trained policy improve the results of causal discovery? **Q3.** Compared to score-based causal discovery, does the proposed data-efficient causal discovery pipeline guarantee to identifying true causal graph as stated in Section 3.3? **Q4.** Compared to false causal graphs, how does the correctness of discovered causal graph influence the task-solving performance of RL agents?

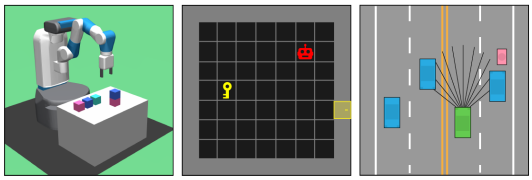

Figure 3: Three experiment environments.

Table 1: Success rate (%) for nine settings in three environments. **Bold** font means the best.

Method	Stack-I	Stack-S	Stack-C	Unlock-I	Unlock-S	Unlock-C	Crash-I	Crash-S	Crash-C
SAC	34.7±16.1	22.1±14.0	31.7±5.1	0.1±0.5	0.0±0.2	0.4±1.7	22.5±17.6	18.6±8.7	6.7±3.8
ICIN	71.8±6.9	71.0±7.4	58.6±8.3	31.7±9.6	32.7±8.6	31.5±8.5	27.9±6.1	15.8±17.2	7.8±8.8
PETS	**97.2±6.9**	77.7±13.5	73.7±10.3	59.5±7.2	20.6±5.9	28.3±10.0	52.3±11.5	44.6±12.5	37.1±5.1
TICSA	85.9±8.4	88.8±10.1	76.2±8.3	58.5±12.3	33.6±14.3	29.8±8.3	68.9±5.9	56.8±8.6	15.0±8.2
ICIL	93.7±5.9	81.2±14.4	62.8±13.0	**67.1±11.6**	15.9±4.7	53.6±15.3	55.3±20.9	21.7±17.7	14.3±7.3
GNN	45.7±9.1	39.0±10.4	41.7±8.6	3.4±2.3	3.4±2.4	4.5±3.0	4.2±4.0	5.1±5.1	3.8±2.8
Score	92.7±7.4	90.5±7.5	73.9±8.5	44.9±28.1	23.1±7.6	36.2±30.1	42.3±17.5	53.4±18.7	8.4±6.1
Full	92.9±6.3	86.0±9.5	75.7±10.3	63.8±9.2	18.3±7.4	53.7±14.3	69.8±14.0	52.6±12.8	42.0±17.2
Offline	96.8±5.8	95.4±6.1	81.4±7.8	13.8±8.1	13.9±7.5	11.7±6.9	13.1±16.2	30.2±16.5	14.9±12.4
GRADER	95.6±5.4	**97.6±6.0**	**93.7±8.4**	64.2±9.1	**61.4±4.4**	**82.1±9.2**	**91.5±4.4**	**84.3±10.0**	**84.7±7.3**

4.1 Environments and Baselines

Since most commonly used RL benchmarks do not explicitly require causal reasoning for generalization, we design three environments shown in Figure 3. These environments use the true state as observation to disentangle the reasoning task from visual understanding. For each environment, we design three settings – in-distribution (*I*), spuriousness (*S*), and composition (*C*) – corresponding to different goal distributions for generalization. We use $p_{\text{train}}(g)$ and $p_{\text{test}}(g)$ to represent the goal distribution during training and testing, respectively. I uses the same $p_{\text{train}}(g)$ and $p_{\text{test}}(g)$, S introduces spurious correlations in $p_{\text{train}}(g)$ but remove them in $p_{\text{test}}(g)$, and C contains more similar sub-goals in $p_{\text{test}}(g)$ than in $p_{\text{train}}(g)$. The details of these settings in three environments are briefly summarized in the following (details in Appendix C.2.1):

- **Stack**: We design this manipulation task inspired by the CausalWorld [42], where the agent must stack objects to match specific shapes and colors. In *Stack-S*, we let the same shape have the same color in $p_{\text{train}}(g)$ but randomly sample the color and shape in $p_{\text{test}}(g)$. In *Stack-C*, the maximum number of object is two in $p_{\text{train}}(g)$ but five in $p_{\text{test}}(g)$.
- **Unlock**: We design this task for the agent to collect a key to open doors built upon the Minigrid [43]. In *Stack-S*, the door and the key are always in the same row in $p_{\text{train}}(g)$ but uniformly sample in $p_{\text{test}}(g)$. In *Unlock-C*, there are one door in $p_{\text{train}}(g)$ but two doors in $p_{\text{test}}(g)$.
- **Crash**: The occurrence of accidents usually relies on causality, e.g., an autonomous vehicle (AV) collides with a jaywalker because its view is blocked by another car [44]. We design such a crash scenario based on highway-env [45], where the goals are to create crashes between a pedestrian and different AVs. In *Stack-S*, the initial distance between ego car and pedestrian is a constant in $p_{\text{train}}(g)$ but irrelevant in $p_{\text{test}}(g)$. In *Stack-C*, there is one pedestrian in $p_{\text{train}}(g)$ but two in $p_{\text{test}}(g)$.

We use the following methods as our baselines to fairly demonstrate the advantages of GRADER. **SAC:** [46] Soft Actor-Critic is a well-known model-free RL method; we augment it with HER [47] in *Stack* because the goal in this task does not have pre-conditions. **ICIN:** [25] It uses DAgger [48] to learn goal-conditioned policy with the causal graph estimated from the expert policy. We assume it can access the true causal graph for supervised learning. **PETS:** [49] We consider the ensemble transition model with random shoot planning as one baseline, which achieves generalization with the uncertainty-aware design. **TICSA:** [18] This is a causal-augmented MBRL method that simultaneously optimizes a soft adjacent matrix (representing the causality) and a transition model. **ICIL:** [16] This method proposes an invariant feature learning structure that captures the implicit causality of multiple tasks. We only use it for transition model learning since the original method is designed for imitation learning. **GNN:** [50] Since graph neural networks are good at learning structural information, we implement a GNN-based baseline using Relational Graph Convolutional Network.

4.2 Results Discussion

Overall Performance (Q1) We compare the testing reward of all methods under nine tasks and summarize the results in Table 1 to demonstrate the overall performance. Generally, our method outperforms other baselines in all tasks except *Stack-I*, which is simple for all methods. We note that the gap between our method and baselines under *S* and *C* settings is more significant than the *I* setting,

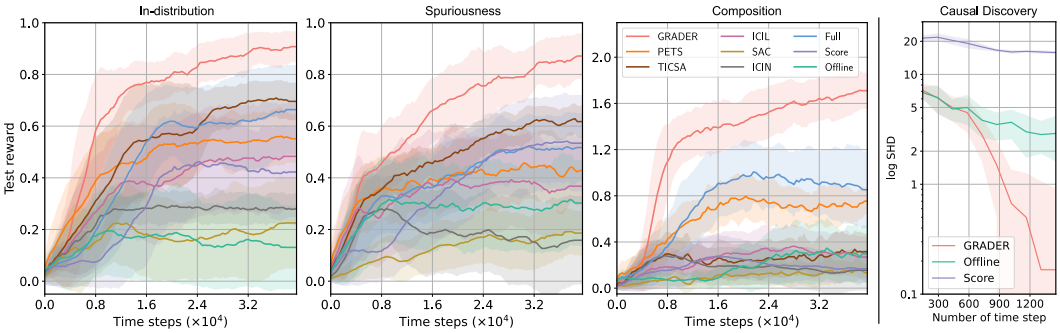

Figure 4: **Left:** Test reward of the *Crash* environment calculated by 30 trails. **Right:** The accuracy of causal graph discovery with samples from GRADER, Score, and Offline (random policy).

showing that our method still works well in the non-trivial generalization task. As a model-free method, SAC fails in all three tasks of *Unlock* and *Crash* environments since they have sparse rewards. Without learning the causal structure of the environment, PETS even cannot fully solve *Unlock-S*, *Unlock-C*, and all *Crash* tasks. Both TICSA and ICIL learn explicit or implicit causality of the task, so they are relatively better than SAC and PETS. However, they are still worse than GRADER in two generalization settings because of the unstable and inefficient causal reasoning mechanism. We also find that even if the ICIN is given the true causal graph, the policy learning part cannot efficiently leverage the causality, leading to worse performance in generalization settings.

To further analyze the tendency of learning, we plot the curves of all methods under *Crash* in Figure 4. Our method quickly learns to solve the tasks at the beginning of the training, demonstrating data efficiency. GRADER also outperforms other methods with large gaps in the later training phase. The results of the other two environments can be found in Appendix B.1.

Importance of Policy Learning (Q2) As we mentioned in Section 3.3, we empirically compare GRADER and **Offline** [51], which uses data from offline random policy, and plot results in the right part of Figure 4. We use SHD [52] to compute the distance between the estimated causal graph and the true causal graph, which is obtained by checking the best causal graph with human analysis. The true causal graph for each environment can be found in Appendix B.2. When we only use samples collected offline by random policy, we cannot obtain variables' values that require long-horizon reasoning, e.g., the door can be opened if the agent is close to the door and has the key, leading to unreliable discovery. As a consequence, the causal graph obtained by Offline indeed harms the performance, as shown in Figure 4. Instead, GRADER gradually explores more regions and quickly obtains the true causal graph when we iteratively discover the causal graph and update the policy.

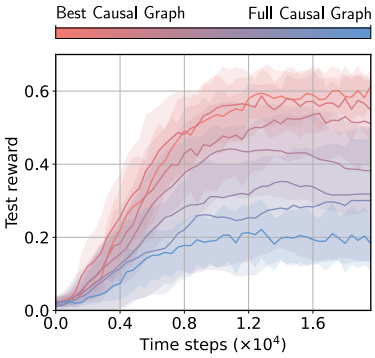

Figure 5: Influence of different causal graphs on *Unlock-S*.

Advantage of Data-efficient Causal Discovery (Q3) To show the advantage of proposed constraint-based methods, we design a model named **Score** that optimizes a soft adjacent matrix similar to the score-based method in the discovery step. Score-based discovery methods, also used in TICSA, are recently combined with NN for differentiable training. According to the discovery accuracy shown in the right part of Figure 4, we find that score-based discovery is inefficient. Based on the performance of the Score model summarized in Table 1, we also conclude that it is not as good as our constraint-based method and has a large variance due to the unstable learning of the causal graph.

Influence of Causal Graph (Q4) To illustrate the importance of the causal graph, we implement another variant of GRADER named **Full**, which uses a fixed full graph that connects all nodes between the sets $\mathcal{U}$ and $\mathcal{V}$. According to the performance shown in Table 1 and Figure 4, we find that the full graph achieves worse results than GRADER because of the redundant and spurious correlation. Intuitively, unrelated information causes additional noises to the learning procedure, and the spurious correlation creates a shortcut that makes the model extract wrong features, leading to worse results in the spuriousness generalization as shown in Table 1.

Table 2: Discovery results on *Chemistry* environment (GRADER / Score). **Bold** font means the best.

Metric	Collider	Chain	Jungle	Full
SHD (↓)	**3.70**±**1.79**/15.4±7.03	**2.80**±**1.83**/14.0±1.18	**7.00**±**2.19**/13.8±0.40	**2.40**±**1.20**/11.0±5.31
Accuracy (↑)	**0.99**±**0.00**/0.87±0.06	**0.99**±**0.00**/0.88±0.01	**0.98**±**0.00**/0.89±0.00	**0.99**±**0.00**/0.91±0.09
Precision (↑)	**0.90**±**0.05**/0.73±0.10	**0.94**±**0.04**/0.79±0.03	0.86±0.04/**0.88**±**0.01**	1.00±0.00/1.00±0.00
Recall (↑)	**0.99**±**0.02**/0.83±0.07	**0.96**±**0.03**/0.73±0.00	**0.96**±**0.02**/0.73±0.00	**0.96**±**0.02**/0.83±0.17
F-score (↑)	**0.94**±**0.03**/0.77±0.06	**0.95**±**0.03**/0.76±0.02	**0.91**±**0.03**/0.80±0.00	**0.98**±**0.01**/0.90±0.10

We then investigate how the correctness of the causal graph influences the performance. We use fixed graphs interpolating from the best causal graph to the full graph to train a GRADER model on *Unlock-S* and summarize the results in Figure 5 The more correct the graph is, the higher reward the agent obtains, which supports our statements in Section 3.4 that the causal graph is important for the reasoning tasks – a better causal graph helps the model have better task-solving performance.

4.3 Further Analysis of Causal Discovery

Finally, we conduct further analysis of the discovery performance on the *Chemistry* environment [53], which is a standard benchmark for evaluating causal discovery methods. In this environment, the underlying causal graph controls the color-changing mechanism of nodes, and the goal of the task is to match the given target colors by changing the color of one node at one time (i.e., doing intervention). Finding the true causal graph makes this task much easier. We consider four types of causal graphs (*Collider*, *Chain*, *Jungle*, *Full*) with 10 nodes in this experiment.

The discovery performance is shown in Table 2 with five metrics indicating the classification error. We can see that GRADER outperforms the Score method in all 4 types of graphs. In Figure 6, we show the discovered graphs from GRADER (averaged over 10 seeds) and the true causal graphs of *Collider* and *Jungle* settings. We also show that GRADER achieves advantages over other baselines in solving the color matching downstream task, which can be found with detailed experiment results in Appendix B.4.

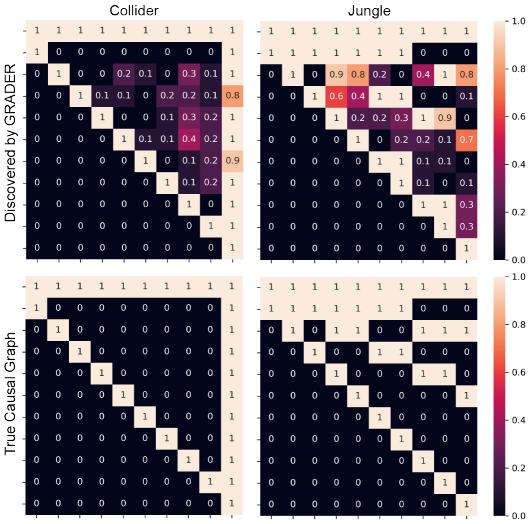

Figure 6: Top: Discovered causal graph from GRADER. Bottom: true causal graph.

5 Related Works

RL Generalization From the agent's view, algorithms focus on actively obtaining the structure of the task or environment. Some decompose the given task into sub-tasks [54, 55, 56, 5] and when they encounter an unseen task, they rearrange those sub-tasks to solve new tasks. Instead of dividing tasks, Symbolic RL learns a program-based policy consisting of domain language [6, 8] or grammar [7, 57]. The generated program is then executed [9] to interact with the environment, which has the potential to solve unseen tasks by searching the symbolic space. From the other view, we generate environments to augment agents' experience for better generalization. One straightforward way is data augmentation of image-based observation [58, 59, 60, 61, 62, 63]. When extended to other factors of the environment, *Domain Randomization* [64] and its variants [65, 66] are proposed. Considering the interaction between agent and environment, *Curriculum Learning* [67] also gradually generates difficult tasks to train generalizable agents.

Goal-Conditioned RL (GCRL) The generalization problem is naturally related to GCRL [68], which aims to train an agent for multiple tasks. From the optimization perspective, Universal Value Function [69], reward shaping [70], and latent dynamic model [71] are widely used tools to solve GCRL problem. Sub-goal generation [72] is another intuitive idea to tackle the long horizon with

sparse reward, where the core thing is to make sure the generated sub-goals are solvable. Finally, *HER* [33], belonging to the relabelling category, is a ground-breaking yet straightforward method that treats visited states as "fake" goals when the goal and state share the same space. Later on, improved versions of HER [47, 73, 74] were widely studied. One limitation is that we cannot directly use a visited state as a goal if the goal has pre-conditions. Similar to our setting, [75] and [76] convert the GCRL problem to variational inference by regarding control as inference [26]. [75] propose an EM framework under the HER setting and [76] treats the last state as the goal and estimates a shaped reward during training.

RL with Causal Reasoning Causality is now frequently discussed in the machine learning field to complement the interpretability of neural networks [28]. RL algorithms also incorporate causality to improve the reasoning capability [77]. For instance, [25] and [19] explicitly estimate causal structures with the interventional data obtained from the environment. These structures can be used to constraint output space [78] or adjust the buffer priority [20]. Building dynamic models in model-based RL [18, 79, 51] based on causal graph is also widely studied recently. Implicitly, we can abstract the causal structure and formulate it using the *Block MDP* [12] setting or training multiple encoders to extract different kinds of representations [15]. Following the idea of invariant risk minimization [80], they assume task-relevant features are invariant and shared across all environments, which can be used as the only cause of the reward.

Causal Discovery Causal discovery [81] is a long-stand topic in economics and sociology, where the traditional methods can be generally categorized into constraint-based and score-based. Constraint-based methods [29] start from a complete graph and iteratively remove edges with conditional independent test [82, 83] as constraints. Score-based methods [38, 84] use metrics such as *Bayesian Information Criterion* [85] as scores and prefer edges that maximize the score given the dataset. Recently, researchers extend score-based methods with RL [86] or differentiable discovery [87, 88, 89]. The former selects edges with a learned policy, and the latter learns a soft adjacent matrix with observational or interventional data. Active intervention methods are also explored [90] to increase the efficiency of data collection and decrease the cost of conducting intervention [91].

6 Conclusion

This paper proposes a latent variable model that injects a causal graph reasoning process into transition model learning and planning to solve GCRL problems under the generalization setting. We theoretically prove that our iterative optimization process can obtain the true causal graph. To evaluate the performance of the proposed method, we designed nine tasks in three environments. The comprehensive experiment results show that our method has better data efficiency and performance than baselines. Our method also provides interpretability by the explicitly discovered causal graph.

The main limitation of this work is that the explicit estimation of causal structure does not scale well to the number of nodes. Developing efficient gradient-based discovery methods could be a promising direction. In addition, the factorized state and action space assumption may restrict the usage of this work to semantic representations, which need to be processed with abstraction methods. We further discuss the potential negative social impact and additional limitations in Appendix C.3.

References

[1] Kai Xu, Akash Srivastava, Dan Gutfreund, Felix Sosa, Tomer Ullman, Josh Tenenbaum, and Charles Sutton. A bayesian-symbolic approach to reasoning and learning in intuitive physics. *Advances in Neural Information Processing Systems*, 34, 2021.

[2] Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases (2020). *arXiv preprint arXiv:2006.11287*, 2006.

[3] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun. Zhu. Closed loop neural-symbolic learning via integrating neural perception, grammar parsing, and symbolic reasoning. In *International Conference on Machine Learning (ICML)*, 2020.

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

[5] Yuchen Lu, Yikang Shen, Siyuan Zhou, Aaron Courville, Joshua B Tenenbaum, and Chuang Gan. Learning task decomposition with ordered memory policy network. *arXiv preprint arXiv:2103.10972*, 2021.

[6] Yichen Yang, Jeevana Priya Inala, Osbert Bastani, Yewen Pu, Armando Solar-Lezama, and Martin Rinard. Program synthesis guided reinforcement learning for partially observed environments. *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Mikel Landajuela, Brenden K Petersen, Sookyung Kim, Claudio P Santiago, Ruben Glatt, Nathan Mundhenk, Jacob F Pettit, and Daniel Faissol. Discovering symbolic policies with deep reinforcement learning. In *International Conference on Machine Learning*, pages 5979–5989. PMLR, 2021.

[8] Jiankai Sun1 Hao Sun1 Tian Han and Bolei Zhou. Neuro-symbolic program search for autonomous driving decision module design. In *Conference on Robot Learning (CoRL)*, 2020.

[9] Zelin Zhao, Karan Samel, Binghong Chen, et al. Proto: Program-guided transformer for program-guided tasks. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Shao-Hua Sun, Te-Lin Wu, and Joseph J Lim. Program guided agent. In *International Conference on Learning Representations*, 2019.

[11] Samuel J Gershman. Reinforcement learning and causal models. *The Oxford handbook of causal reasoning*, 1:295, 2017.

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

[13] Manan Tomar, Amy Zhang, Roberto Calandra, Matthew E Taylor, and Joelle Pineau. Model-invariant state abstractions for model-based reinforcement learning. *arXiv preprint arXiv:2102.09850*, 2021.

[14] Sumedh A Sontakke, Arash Mehrjou, Laurent Itti, and Bernhard Schölkopf. Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning. In *International Conference on Machine Learning*, pages 9848–9858. PMLR, 2021.

[15] Shagun Sodhani, Sergey Levine, and Amy Zhang. Improving generalization with approximate factored value functions. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[16] Ioana Bica, Daniel Jarrett, and Mihaela van der Schaar. Invariant causal imitation learning for generalizable policies. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Beining Han, Chongyi Zheng, Harris Chan, Keiran Paster, Michael Zhang, and Jimmy Ba. Learning domain invariant representations in goal-conditioned block mdps. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Zizhao Wang, Xuesu Xiao, Yuke Zhu, and Peter Stone. Task-independent causal state abstraction. *Workshop on Robot Learning: Self-Supervised and Lifelong Learning, NeurIPS*, 2021.

[19] Sergei Volodin, Nevan Wichers, and Jeremy Nixon. Resolving spurious correlations in causal models of environments via interventions. *arXiv preprint arXiv:2002.05217*, 2020.

[20] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Maxime Gasse, Damien Grasset, Guillaume Gaudron, and Pierre-Yves Oudeyer. Causal reinforcement learning using observational and interventional data. *arXiv preprint arXiv:2106.14421*, 2021.

[22] David Abel. A theory of abstraction in reinforcement learning. *arXiv preprint arXiv:2203.00397*, 2022.

[23] Murray Shanahan and Melanie Mitchell. Abstraction for deep reinforcement learning. *arXiv preprint arXiv:2202.05839*, 2022.

[24] David Abel, Dilip Arumugam, Lucas Lehnert, and Michael Littman. State abstractions for lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 10–19. PMLR, 2018.

[25] Suraj Nair, Yuke Zhu, Silvio Savarese, and Li Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

[26] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[27] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1-2):49–107, 2000.

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

[29] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[30] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

[31] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[32] Joseph Marino and Yisong Yue. An inference perspective on model-based reinforcement learning. In *ICML Workshop on Generative Modeling and Model-Based Reasoning for Robotics and AI*, 2019.

[33] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[34] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[35] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[36] Arthur George Richards. *Robust constrained model predictive control*. PhD thesis, Massachusetts Institute of Technology, 2005.

[37] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

[38] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[39] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Fast conditional independence test for vector variables with large sample sizes. *arXiv preprint arXiv:1804.02747*, 2018.

[40] Clément L Canonne, Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Testing conditional independence of discrete distributions. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–57. IEEE, 2018.

[41] Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514–1538, 2020.

[42] Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Yoshua Bengio, Bernhard Schölkopf, Manuel Wüthrich, and Stefan Bauer. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *arXiv preprint arXiv:2010.04296*, 2020.

[43] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

[44] Zenna Tavares, James Koppel, Xin Zhang, Ria Das, and Armando Solar-Lezama. A language for counterfactual generative models. In *International Conference on Machine Learning*, pages 10173–10182. PMLR, 2021.

[45] Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

[46] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[47] Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. Exploration via hindsight goal generation. *Advances in Neural Information Processing Systems*, 32, 2019.

[48] Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. In *In AISTATS*. Citeseer, 2011.

[49] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[50] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

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

[52] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

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

[54] Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pages 3418–3428. PMLR, 2019.

[55] Danfei Xu, Roberto Martín-Martín, De-An Huang, Yuke Zhu, Silvio Savarese, and Li F Fei-Fei. Regression planning networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[56] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2019.

[57] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

[58] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[59] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[60] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[61] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

[62] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. *arXiv preprint arXiv:1910.05396*, 2019.

[63] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[65] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *Conference on Robot Learning*, pages 1162–1176. PMLR, 2020.

[66] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. IEEE, 2019.

[67] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *arXiv preprint arXiv:2003.04960*, 2020.

[68] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

[69] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[70] Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

[71] Suraj Nair, Silvio Savarese, and Chelsea Finn. Goal-aware prediction: Learning to model what matters. In *International Conference on Machine Learning*, pages 7207–7219. PMLR, 2020.

[72] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR, 2018.

[73] Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pages 7750–7761. PMLR, 2020.

[74] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32, 2019.

[75] Yunhao Tang and Alp Kucukelbir. Hindsight expectation maximization for goal-conditioned reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2863–2871. PMLR, 2021.

[76] Tim GJ Rudner, Vitchyr Pong, Rowan McAllister, Yarin Gal, and Sergey Levine. Outcome-driven reinforcement learning via variational inference. *Advances in Neural Information Processing Systems*, 34, 2021.

[77] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2493–2500, 2020.

[78] Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. Causalaf: Causal autoregressive flow for goal-directed safety-critical scenes generation. *arXiv preprint arXiv:2110.13939*, 2021.

[79] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. *arXiv preprint arXiv:2206.13452*, 2022.

[80] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[81] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[82] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

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

[84] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[85] Andrew A Neath and Joseph E Cavanaugh. The bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

[86] Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.

[87] Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *arXiv preprint arXiv:2007.01754*, 2020.

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

[89] Yunzhu Li, Antonio Torralba, Anima Anandkumar, Dieter Fox, and Animesh Garg. Causal discovery in physical systems from videos. *Advances in Neural Information Processing Systems*, 33:9180–9192, 2020.

[90] Nino Scherrer, Olexa Bilaniuk, Yashas Annadani, Anirudh Goyal, Patrick Schwab, Bernhard Schölkopf, Michael C Mozer, Yoshua Bengio, Stefan Bauer, and Nan Rosemary Ke. Learning neural causal models with active interventions. *arXiv preprint arXiv:2109.02429*, 2021.

[91] Erik Lindgren, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Experimental design for cost-aware learning of causal graphs. *Advances in Neural Information Processing Systems*, 31, 2018.

[92] Dar Gilboa, B. Chang, Minmin Chen, Greg Yang, Samuel S. Schoenholz, Ed H. Chi, and Jeffrey Pennington. Dynamical isometry and a mean field theory of lstms and grus. *ArXiv*, abs/1901.08987, 2019.

[93] Karren D. Yang, Abigail Katoff, and Caroline Uhler. Characterizing and learning equivalence classes of causal dags under interventions. In *ICML*, 2018.

[94] Raghavendra Addanki, Shiva Kasiviswanathan, Andrew Mcgregor, and Cameron Musco. Efficient intervention design for causal discovery with latents. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 63–73. PMLR, 13–18 Jul 2020.

[95] Wenhao Ding, Chejian Xu, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical scenario generation from methodological perspective. *arXiv preprint arXiv:2202.02215*, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In the conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In the Appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In Supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] In the Supplementary.
   (b) Did you mention the license of the assets? [Yes] In the Supplementary.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the Supplementary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] In the Supplementary
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] In the Supplementary.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

Table of Contents

A Theoretical Proofs **18**
- A.1 Formulation of Latent Variable Models . . . 18
- A.2 Transition Model Learning . . . 19
- A.3 Derivation of Planner Module . . . 20
- A.4 Causal Discovery . . . 23
- A.5 Overall Performance Guarantee of Iterative Optimization . . . 26

B Additional Experiment **27**
- B.1 Overall Performance . . . 27
- B.2 Causal Graph Analysis . . . 28
- B.3 Distance between Goal and State Distribution . . . 28
- B.4 Task Performance of Chemistry Experiment . . . 28

C Additional Information **29**
- C.1 Details about Conditional Independence Test . . . 29
- C.2 Experiment Details . . . 30
- C.3 Broader Social Impact and Additional Limitation . . . 31

A Theoretical Proofs

In Appendix A, we first show the derivation of the latent variable models A.1, then provide some analytical results in the iterative optimization of model learning A.2, planning A.3, and causal discovery A.4. Finally, we give the proof of the theorem of overall performance guarantee A.5 given some common assumptions.

To compactly write down our formulas, we slightly abuse the notations by representing s^t, a^t as the joint states and actions at timestep t, while using s_i, a_i to denote the i-th dimension of factorized states or actions. Without the loss of generality, we implement our reward in a deterministic way, which only involves $r(s, g)$ in its notation, we will slightly generalize to some state-action reward function for our analysis as well.

A.1 Formulation of Latent Variable Models

A.1.1 Derivation of Equation (2)

The ELBO of the likelihood of the trajectory is obtained by

$$\begin{aligned}
\log p(\tau|s^*) &= \log \int p(\tau|\mathcal{G}, s^*) p(\mathcal{G}|s^*) d\mathcal{G} \\
&= \log \int q(\mathcal{G}|\tau) \frac{p(\tau|\mathcal{G}, s^*) p(\mathcal{G}|s^*)}{q(\mathcal{G}|\tau)} d\mathcal{G} \\
&\geq \int q(\mathcal{G}|\tau) \log \frac{p(\tau|\mathcal{G}, s^*) p(\mathcal{G}|s^*)}{q(\mathcal{G}|\tau)} d\mathcal{G} \\
&= \int q(\mathcal{G}|\tau) \left(\log p(\tau|\mathcal{G}, s^*) + \log \frac{p(\mathcal{G}|s^*)}{q(\mathcal{G}|\tau)} \right) d\mathcal{G} \\
&= \int q(\mathcal{G}|\tau) \log p(\tau|\mathcal{G}, s^*) d\mathcal{G} + \int q(\mathcal{G}|\tau) \log \frac{p(\mathcal{G}|s^*)}{q(\mathcal{G}|\tau)} d\mathcal{G} \\
&= \mathbb{E}_{q(\mathcal{G}|\tau)} [\log p(\tau|\mathcal{G}, s^*)] - \mathbb{D}_{\mathrm{KL}} [q(\mathcal{G}|\tau) || p(\mathcal{G})]
\end{aligned} \tag{10}$$

where the third line is obtained by Jensen's inequality and the last line is because the prior of the causal graph $\mathcal{G}$ is independent of the achieved goal s^*.

A.1.2 Derivation of Equation (3)

According to the decomposition of state-action trajectory

$$p(\tau) = p(s^0) \sum_{t=0}^{T-1} p(s^{t+1}|s^t, a^t) p(a^t|s^t) \tag{11}$$

we can get the following

$$\begin{aligned}
\log p(\tau|\mathcal{G}, s^*) &= \log(s^0, a^0, s^1, a^1, \cdots, a^{T-1}, s^T|\mathcal{G}, s^*) \\
&= \log p(s^0|\mathcal{G}, s^*) + \sum_{t=0}^{T-1} \log p(s^{t+1}|s^t, a^t, \mathcal{G}, s^*) + \sum_{t=0}^{T-1} \log p(a^t|s^t, \mathcal{G}, s^*) \\
&= \log p(s^0) + \sum_{t=0}^{T-1} \log p(s^{t+1}|s^t, a^t, \mathcal{G}) + \sum_{t=0}^{T-1} \log p(a^t|s^t, \mathcal{G}, s^*)
\end{aligned} \tag{12}$$

A.2 Transition Model Learning

The optimization in Model learning step can be described below:

$$\arg\max_{\theta} [\sum_{t} \log p_\theta(s_{t+1}|s_t, a_t, \mathcal{G})] \tag{13}$$

where $\tau = [s_1, a_1, \cdots, s_T]$ is the trajectory in data buffer, and $\mathcal{G}$ is the given causal graph.

Here below, we show some necessary definitions and propositions to prove the Lemma 1.

Definition 4 (Structural Hamming Distance (SHD))**.** *For any two DAGs $\mathcal{G}, \mathcal{H}$ with identical vertices set V, we define the following function SHD: $\mathcal{G} \times \mathcal{H} \to \mathbb{R}$,*

$$\textit{SHD}(\mathcal{G}, \mathcal{H}) = \#\{(i,j) \in V^2 \mid \mathcal{G} \textit{ and } \mathcal{H} \textit{ have different edges } e_{ij}\} \tag{14}$$

Definition 5 (Respect the graph)**.** *For any given transition model with specific causal graph $\mathcal{G}$, the transition model respects the graph if the distribution $p(s_{t+1}|a_t, s_t, \mathcal{G})$ can be factorized as:*

$$p(s'|s, a, \mathcal{G}) = \prod_{i \in [M]} p(s'_i | \boldsymbol{PA}(s'_i), \mathcal{G}) \tag{15}$$

where M is the total number of factorized state, $\boldsymbol{PA}(\cdot)$ represents the parents in the causal graph.

Proposition 3 (GRU model respects the graph)**.** *As the parameterized transition model $p_\theta(s'|s, a, \mathcal{G})$ reaches the steady state, it respects the graph.*

Proof of Proposition 3. The GRU modules with parameter $\theta = [W, U]$ can be rewritten as a message passing process, where $AGG.(\cdot)$ is the iterative aggregation function.

$$\begin{aligned}
\text{Node Encoder}: h_j^{(0)} &= f_{\text{encoder}}(x_j) \\
\text{Aggregation}: h_j^{(\ell)} &= AGG_{i \in \mathcal{N}(j)}(f_\theta(x_j^{(\ell-1)}, h_i^{(\ell-1)})) \\
\text{Node Decoder}: \boldsymbol{x}_i^{(\ell)} &= f_{\text{decoder}}(h_j^{(\ell-1)})
\end{aligned} \tag{16}$$

As an iterated process of message passing, where the input causal graph controls the information flow between different entities, this GRU model can be rewritten as a fixed point iteration [92]:

$$\boldsymbol{x}_i^{(\ell)} = F_\theta(\mathbf{PA}(\boldsymbol{x}_i)^{(\ell-1)}, \boldsymbol{x}_i^{(\ell-1)}) \tag{17}$$

With proper initialization and some sufficient conditions provided by [92], F has unique equilibrium point, where

$$\boldsymbol{x}_i^{\infty} = F_\theta(\mathbf{PA}(\boldsymbol{x}_i)^{\infty}, \boldsymbol{x}_i^{\infty}) \tag{18}$$

In our bipartite graph, when GRU reaches the equilibrium point, we can get an structural causal model:

$$s'_i = F_\theta(\mathbf{PA}(s'_i), s_i), \quad \text{where } s'_i \in \mathcal{S}', \mathbf{PA}(s'_i) \in \mathcal{S} \cup \mathcal{A} \tag{19}$$

Based on the SCM derivation 19, we can then factorize the transition model as:

$$p_\theta(s'|s, a, \mathcal{G}) = \prod_{i \in [M]} p_\theta(s'_i|\mathbf{PA}(s'_i), \mathcal{G}) \tag{20}$$

We denote the ground truth causal graph as $G^* = (V, E^*)$, and $\mathbf{PA}^*(s'_i)$ as the true parents of s'_i in G^*. ■

Definition 6 (Causal optimality at equlibrium point)**.** *For any $G' \neq G^*$ with at least one pair of flawed parental relationship $\boldsymbol{PA}'(s'_i) \neq \boldsymbol{PA}^*(s'_i)$, the following inequality holds:*

$$p_\theta(s'_i|\boldsymbol{PA}'(s'_i), \mathcal{G}) \leq p_\theta(s'_i|\boldsymbol{PA}^*(s'_i), \mathcal{G}) \tag{21}$$

Lemma 5 (Local monotonicity)**.** *Given one state variable s_i and its any parental relationship $\boldsymbol{PA}^1(s_i), \boldsymbol{PA}^2(s_i)$, if $\#(\boldsymbol{PA}^1(s_i) \cup \boldsymbol{PA}^*(s_i)) \geq \#(\boldsymbol{PA}^2(s_i) \cup \boldsymbol{PA}^*(s_i))$, then at steady state, the SCM derived in 19 will miss part of the message provided from the true parents, therefore $p_\theta(s'_i|\boldsymbol{PA}^1(s'_i), \mathcal{G}_1) \geq p_\theta(s'_i|\boldsymbol{PA}^2(s'_i), \mathcal{G}_2)$*

Proof of Lemma 1. Based on the factorization defined in 20, we denote the parental relationship in $\mathcal{G}_1$ as $\mathbf{PA}^1(\cdot)$,

$$p_\theta(s'|a, s, \mathcal{G}^*) = \prod_{i \in [M]} p_\theta(s'_i|\mathbf{PA}^*(s'_i), \mathcal{G}) \geq \prod_{i \in [M]} p_\theta(s'_i|\mathbf{PA}^1(s'_i), \mathcal{G}) = p_\theta(s'|a, s, \mathcal{G}_1) \tag{22}$$

For $\mathcal{G}_1$ and $\mathcal{G}_2$, suppose the only different edges e has a target node s'_j, with SHD$(\mathcal{G}_1, \mathcal{G}^*) <$ SHD$(\mathcal{G}_2, \mathcal{G}^*)$, based on Lemma 5:

$$\begin{aligned} p_\theta(s'|a, s, \mathcal{G}_1) &= p_\theta(s'_j|\mathbf{PA}^1(s'_j), \mathcal{G}_1) \prod_{i \in [M] \setminus j} p_\theta(s'_i|\mathbf{PA}^1(s'_i), \mathcal{G}_1) \\ &\geq p_\theta(s'_j|\mathbf{PA}^2(s'_j), \mathcal{G}_2) \prod_{i \in [M] \setminus j} p_\theta(s'_i|\mathbf{PA}^1(s'_i), \mathcal{G}_2) \\ &= p_\theta(s'_j|\mathbf{PA}^2(s'_j), \mathcal{G}_2) \prod_{i \in [M] \setminus j} p_\theta(s'_i|\mathbf{PA}^2(s'_i), \mathcal{G}_2) \\ &= p_\theta(s'|a, s, \mathcal{G}_2). \end{aligned} \tag{23}$$

Based on the inequality derived in 22 and 23,

$$\log p_\theta(s'|s, a, \mathcal{G}^*) \geq \log p_\theta(s'|s, a, \mathcal{G}_1) \geq \log p_\theta(s'|s, a, \mathcal{G}_2). \tag{24}$$

the monotonicity of likelihood in Lemma 1 is proved. ■

A.3 Derivation of Planner Module

The optimization in the planning part is:

$$\max_\pi \sum_{t=0}^{T-1} \log \pi_\theta(a^t|s^t, \mathcal{G}, s^*) = \max_{[a_0, \cdots, a^{T-1}]} \sum_{t=0}^{T-1} \log \hat{Q}(s^t, a^t) \tag{25}$$

Ideally, given the access to real dynamics $p(s'|s, a)$ and goal distribution $p(g)$. We first define the expected goal-conditioned state-action reward $r(s, a, g) = \mathbb{E}_{s' \sim p(\cdot|s,a)} r(s', g)$, and the expected state-action reward $r(s, a) = \mathbb{E}_{g \sim p_g(\cdot)} r(s, a, g)$. In practice, due to the inaccuracy of transition model, we can only query the following reward estimation at certain state-action pair: $r(s, a, g) = \mathbb{E}_{s' \sim p_\theta(\cdot|s,a,\mathcal{G})} r(s', g), r(s, a) = \mathbb{E}_{g \sim p_g(\cdot)} r(s, a, g)$.

Then we consider the distribution of the goal $g \sim p_g(\cdot)$, which is supported on the state space $\mathcal{S}$. Based on Algorithm 1, our interventional data is collected by the MPC algorithm that maximizes the expected discounted cumulative reward from a learned dynamics. Thus, we could denote the interventional distribution of state (depending on the current policy π) in the data buffer as $p_{\mathcal{I}^s_\pi}$, $s \sim p_{\mathcal{I}^s_\pi}(\cdot)$ which is also supported on the state space $\mathcal{S}$.

Proof of Lemma 3. Assume our planning algorithm has an infinite planning horizon, with the optimal transition dynamics and optimal policy, the action-value function Q^* can be expressed as:

$$Q^*(s^t, a^t) \stackrel{\text{def}}{=} \mathbb{E}_{s\sim p_{\mathcal{I}^s_{\pi^*}}(\cdot), a\sim\pi^*(s)} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s^{t'}, a^{t'}) \mid s^t, a^t \right] \tag{26}$$

The estimation of action value function $\hat{Q}(s^t, a^t) = Q^{\hat{\pi}}_{\theta,\mathcal{G}}(s^t, a^t)$ can be written as:

$$\begin{aligned} \hat{Q}(s^t, a^t) &\stackrel{\text{def}}{=} \mathbb{E}_{s\sim p_{\mathcal{I}^s_{\hat{\pi}}}(\cdot), a\sim\hat{\pi}(s)} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s^{t'}, a^{t'}) \mid s^t, a^t \right] \\ &= \mathbb{E}_{a\sim\hat{\pi}(s)} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{E}_{g\sim p_g(\cdot), s\sim p_{\mathcal{I}^s_{\hat{\pi}}}(\cdot)} (1 - \mathbb{1}(s' = g)) \mid s^t, a^t \right] \end{aligned} \tag{27}$$

The policy by MPC in algorithm 1 can be deducted by: $\hat{\pi}(s^t) = \arg\max_{a^t\in\mathcal{A}} \hat{Q}(s^t, a^t)$, let $s^0 = s$, and we could derive value function under the MPC policy as follows:

$$\begin{aligned} V(s) &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p_g} \mathbb{E}_{p_{\mathcal{I}^s_\pi}} \mathbb{1}(s = g) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p_g} \mathbb{E}_{p_{\mathcal{I}^s_\pi}} [1 - \mathbb{1}(s \neq g)] \\ &= \sum_{t=0}^{\infty} \gamma^t - \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p_g} \mathbb{E}_{p_{\mathcal{I}^s_\pi}} \mathbb{1}(s \neq g) \\ &\leq \frac{1 - \mathbb{D}_{TV}(p_{\mathcal{I}^s_{\hat{\pi}}}, p_g)}{1-\gamma} \end{aligned} \tag{28}$$

where $\mathbb{D}_{\text{TV}}(p_{\mathcal{I}^s_\pi}, p_g)$ is the total variation distance between the marginal state distribution $p_{\mathcal{I}^s_\pi}$ in the data buffer, as well as the goal distribution p_g, both of which share the same support. ■

In addition, we could define a more general form of goal-conditioned reward as based on the distance: $r(s, g) = 1 - d(s', g)$. Where $\mathbb{D}$ is a (normalized) distance measure between two vectors in the state space, s.t. $\forall s, g \in \mathcal{S}$, $0 \leq d(s, g) \leq 1$. For instance, if we pick $d(s, g) = \mathbb{1}(s \neq g)$, the derived reward under this distance measure will go back to the reward function defined in section 2.1. By defining a (normalized) ℓ_p distance between s' and g, $d(s, g) = \frac{\|s-g\|_p}{\max_{s_1,s_2\in\mathcal{S}} \|s_1-s_2\|_p}$, we can also shape a continuous form of goal-conditioned step reward $r(s, g)$ between 0 and 1. Notice that all the Euclidean-based distance are all valid metrics with symmetry, non-negativity, identity of indiscernibles, and triangle inequality. With such definition, the estimated value function will fits in with:

$$V(s) \leq \frac{1 - \mathcal{W}(p_{\mathcal{I}^s_\pi}, p_g)}{1-\gamma} \tag{29}$$

where $\mathcal{W}$ is some Wasserstein distance between marginal state distribution and goal distribution. Therefore, optimizing the Q value is equivalent to minimizing an upper bound for some types of the statistical distance between goal distribution and target distribution.

For the term related to policy in (4), we can define the goal-conditioned policy distribution as:

$$\pi(a^t|s^t, g) \propto \exp(Q(s^t, a^t)) \tag{30}$$

As a result, $\arg\max_\pi \sum_{t=0}^{T-1} \log \pi_\theta(a^t|s^t, s^*, \mathcal{G}) = \arg\max_\pi \sum_{t=0}^{T-1} Q(s^t, a^t)$ However, the real $Q(s^t, a^t)$ is intractable, so we alternatively optimize the $\hat{Q}(s^t, a^t)$ at each time step. Next, we'll start to derive a bound between $\hat{Q}(s, \hat{\pi}(s))$ and $Q(s, \pi^*(s))$

Proof of Lemma 2. For simplicity, we denote the learned transition function $\hat{p}(s'|s, a) = p_\theta(s'|s, a, \mathcal{G})$, which is ϵ_m-approximate dynamics, $\mathbb{D}_{\text{TV}}(\hat{p}, p) = \|\hat{p}(s'|s, a) - p(s'|s, a)\|_\infty \leq \epsilon_m$,

Firstly, we show by value iteration that the estimated value function $\hat{V}(s)$ will converge to $V(s)$: Assume exists $K > 0$, s.t. $\forall k > K$, $\|\hat{p}(s'|, s, a) - p(s, |s, a)\|_\infty \leq \epsilon_m$

$$\hat{V}^{(k+1)}(s) = r(s, \pi(s)) + \gamma \sum_{s'} p(s'|s, \pi(s)) \hat{V}^{(k)}(s). \tag{31}$$

Given the result of Bellman Contraction,

$$\|\hat{V}^{(k+1)}(s) - V^{\pi^*}(s)\|_\infty \leq \gamma \|\hat{V}^{(k)}(s) - V^{\pi^*}(s)\|_\infty, \quad \lim_{k\to\infty} \|\hat{V}^{(k+1)}(s) - V^{\pi^*}(s)\|_\infty = 0. \tag{32}$$

Based on the definition of greedy policy in planning: $\hat{\pi}(s) = \arg\max_{a\in\mathcal{A}} \hat{Q}(s,a)$, we can derive the inequality:

$$\begin{aligned}
& r(s,\hat{\pi}(s)) + \gamma \sum_{s'} \hat{p}(s'|s,\hat{\pi}(s))\hat{V}(s') \geq r(s,\pi^*(s)) + \gamma \sum_{s'} \hat{p}(s'|s,\pi^*(s))\hat{V}(s') \\
\implies \quad & r(s,\pi^*(s)) - r(s,\hat{\pi}(s)) \leq \gamma \left[\sum_{s'} \hat{p}(s'|s,\hat{\pi}(s))\hat{V}(s') - \sum_{s'} \hat{p}(s'|s,\pi^*(s))\hat{V}(s') \right] \\
\overset{\|\hat{V}(s)-V^{\pi^*}(s)\|_\infty \to 0}{\implies} \quad & r(s,\pi^*(s)) - r(s,\hat{\pi}(s)) \leq \gamma \left[\sum_{s'} \hat{p}(s'|s,\hat{\pi}(s))V^{\pi^*}(s') - \sum_{s'} \hat{p}(s'|s,\pi^*(s))V^{\pi^*}(s') \right].
\end{aligned} \tag{33}$$

Then let s be the state with largest value error.

$$\begin{aligned}
V^{\pi^*}(s) - V^{\hat{\pi}}(s) =& r(s,\pi^*(s)) - r(s,\hat{\pi}(s)) \\
& + \gamma \Big[\sum_{s'} p(s'|s,\pi^*(s))V^{\pi^*}(s') - \sum_{s'} p(s'|s,\hat{\pi}(s))V^{\hat{\pi}}(s') \Big] \\
\leq& \gamma \sum_{s'} \Big[\hat{p}(s'|s,\hat{\pi}(s))V^{\pi^*}(s') - \hat{p}(s'|s,\pi^*(s))V^{\pi^*}(s') \Big] \\
& + \gamma \sum_{s'} \Big[p(s'|s,\pi^*(s))V^{\pi^*}(s') - p(s'|s,\hat{\pi}(s))V^{\hat{\pi}}(s') \Big] \\
=& \gamma \sum_{s'} \Big[p(s'|s,\pi^*(s)) - \hat{p}(s'|s,\pi^*(s)) \Big] V^{\pi^*}(s') \\
& - \gamma \sum_{s'} \Big[p(s'|s,\hat{\pi}(s)) - \hat{p}(s'|s,\hat{\pi}(s)) \Big] V^{\pi^*}(s') \\
& + \gamma \sum_{s'} p(s'|s,\hat{\pi}(s)) \Big[V^{\pi^*}(s) - V^{\hat{\pi}}(s) \Big]
\end{aligned} \tag{34}$$

Since $r(s,g) \in [0,1]$, the value function $V(s) \in [0, \frac{1}{1-\gamma}]$, also by $\|\hat{p}(s'|s,\hat{\pi}(s)) - p(s'|s,\hat{\pi}(s))\| \leq \epsilon$, we have

$$\begin{aligned}
V^{\pi^*}(s) - V^{\hat{\pi}}(s) &\leq \gamma\epsilon_m(V_{\text{max}} - V_{\text{min}}) + \gamma \sum_{s'} p(s'|s,\hat{\pi}(s)) \Big[V^{\pi^*}(s) - V^{\hat{\pi}}(s) \Big] \\
&= \frac{\gamma\epsilon_m}{1-\gamma} + \gamma \sum_{s'} p(s'|s,\hat{\pi}(s)) \left[V^{\pi^*}(s') - V^{\hat{\pi}}(s') \right].
\end{aligned} \tag{35}$$

We already analyze the state s with largest value error, and it's sufficient to show:

$$\begin{aligned}
\|V^{\pi^*}(s) - V^{\hat{\pi}}(s)\|_\infty &\leq \frac{\gamma\epsilon_m}{1-\gamma} + \gamma \sum_{s'} p(s'|s,\hat{\pi}) \|V^{\hat{\pi}}(s) - V^{\pi^*}(s)\|_\infty \\
&= \frac{\gamma\epsilon_m}{1-\gamma} + \gamma \|V^{\pi^*}(s) - V^{\hat{\pi}}(s)\|_\infty
\end{aligned} \tag{36}$$

By combining $\|V^{\pi^*}(s) - V^{\hat{\pi}}(s)\|_\infty$ on both sides, we have

$$\|V^{\pi^*}(s) - V^{\hat{\pi}}(s)\|_\infty \leq \frac{\gamma}{(1-\gamma)^2}\epsilon_m \tag{37}$$

■

A.4 Causal Discovery

A.4.1 Assumptions of Causality

Assumption 2 (Markov property)**.** *Given a DAG $\mathcal{G}$ and a joint distribution $P_{\boldsymbol{X}}$, this distribution is said to satisfy*

- *(i) the global Markov property with respect to the DAG $\mathcal{G}$ if*

$$\boldsymbol{A} \perp\!\!\!\perp_{\mathcal{G}} \boldsymbol{B} | \boldsymbol{C} \Rightarrow \boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B} | \boldsymbol{C} \tag{38}$$

  for all disjoint vertex sets $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$. The symbol $independent_{\mathcal{G}}$ denotes d-separation.

- *(ii) the local Markov property with respect to the DAG $\mathcal{G}$ if each variable is independent of its non-descendants (without its parents) given its parents, and*

- *(iii) the Markov factorization property with respect to the DAG $\mathcal{G}$ if*

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_d) = \prod_{j=1}^{d} p(x_j | \boldsymbol{PA}^{\mathcal{G}}(x_j)) \tag{39}$$

  where we assume that $P_{\boldsymbol{X}}$ has a density p.

Assumption 3 (Faithfulness)**.** *Consider a distribution $P_{\boldsymbol{X}}$ and a DAG $\mathcal{G}$, $P_{\boldsymbol{X}}$ is faithful to the DAG $\mathcal{G}$ if we know*

$$\boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B} | \boldsymbol{C} \Rightarrow \boldsymbol{A} \perp\!\!\!\perp_{\mathcal{G}} \boldsymbol{B} | \boldsymbol{C} \tag{40}$$

for all disjoint vertex sets $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$.

A.4.2 KL Divergence as Sparsity Regularization

Similar to the assumption in Factorized MDP, the existence of causal relationship between two arbitrary entities among x is can also be treated as independent. Therefore, we construct the prior distribution $p(\mathcal{G})$ as independent Bernoulli Distribution of in the transition causal graph.

$$p(\mathcal{G}) = \prod_{i \in [M+N], j \in [M]} p(\mathcal{G}_{ij}) = \prod_{i \in [M+N], j \in [M]} p_{ij} \tag{41}$$

where $\mathcal{G}_{ij}$ represents the edge from i-th node in source node set $\mathcal{U} = \{\mathcal{A} \cup \mathcal{S}\}$ to the j-th node in target node set $\mathcal{V} = \{\mathcal{S}'\}$ in the bipartite transition causal graph.

On the other hand, for the variational posterior $q(\mathcal{G}|\tau)$, for the discovered transition causal graph, it needs to satisfy two constraints: (i) $q(\mathcal{G}|\tau)$ needs to be a DAG, denoted $\mathcal{Q}_{DAG}$, and more specifically, a bipartite graph. We denote such subset of DAG as $\mathcal{Q}_{Bi}$, (ii) $q(\mathcal{G}|\tau)$ needs to be as sparse as possible.

Common score-based causal discovery works use two regularization terms, DAGNess and ℓ_1 regularization to constrain the discovered causal graph in the constraint set, while in our work, we explicitly constrain the posterior variational distribution $q(\mathcal{G}|\tau) \in \mathcal{Q}_{Bi} \subset \mathcal{Q}_{DAG}$. We then show in the following section that by defining a certain independent Bernoulli prior $p(\mathcal{G})$, the KL divergence between variational posterior $q(\mathcal{G}|\tau)$ and $p(\mathcal{G})$ can be equivalent to a sparsity regularization.

According to our constraint-based causal reasoning modules, $\mathcal{Q}_{Bi}$ consists of $M(M+N)$ independent binary classifiers (that forms a DAG) parameterized by our kernal-based independent testing modules ϕ, i.e.

$$q_\phi(\mathcal{G}|\tau) = \prod_{i \in [M+N], j \in [M]} q_\phi(\mathcal{G}_{ij}|\tau) \triangleq \prod_{i \in [M+N], j \in [M]} q_{ij} \tag{42}$$

Proof of Proposition 1. Let the prior $p_{ij} = \epsilon_{\mathcal{G}} \in (0, \frac{1}{2}], \forall i \in [M+N], j \in [M]$, based on the definition above, the KL divergence term in (4) can be expanded as follows:

$$\begin{aligned}
\mathbb{D}_{\text{KL}}(q_\phi(\mathcal{G}|\tau)\|p(\mathcal{G})) &= \sum_{q \in \mathcal{Q}_{Bi}} \prod_{i,j} q_{ij} \log \frac{\prod_{i,j} q_{ij}}{\prod_{i,j} p_{ij}} = \sum_{i,j} \left[q_{ij} \log \frac{q_{ij}}{p_{ij}} + (1-q_{ij}) \log \frac{1-q_{ij}}{1-p_{ij}} \right] \\
&= \sum_{i,j} \left[q_{ij} \log \frac{q_{ij}}{\epsilon_{\mathcal{G}}} + (1-q_{ij}) \log \frac{1-q_{ij}}{1-\epsilon_{\mathcal{G}}} \right] \\
&= \sum_{i,j} \left[q_{ij} \log q_{ij} + (1-q_{ij}) \log(1-q_{ij}) - q_{ij} \log \epsilon_{\mathcal{G}} - (1-q_{ij}) \log(1-\epsilon_{\mathcal{G}}) \right]
\end{aligned} \tag{43}$$

Since $q_{ij} \in \{0,1\}$, $\lim_{q_{ij} \to 0} q_{ij} \log q_{ij} = \lim_{q_{ij} \to 1} (1-q_{ij}) \log(1-q_{ij}) = 0$,

$$\begin{aligned}
\mathbb{D}_{\text{KL}}(q_\phi(\mathcal{G}|\tau)\|p(\mathcal{G})) &= \sum_{i,j} \left[-q_{ij} \log(\epsilon_{\mathcal{G}}) - (1-q_{ij}) \log(1-\epsilon_{\mathcal{G}}) \right] \\
&= \sum_{i,j} \left[-\mathbb{I}(q_{ij}=1) \log \epsilon_{\mathcal{G}} - \mathbb{I}(q_{ij}=0) \log(1-\epsilon_{\mathcal{G}}) \right] \\
&= \sum_{i,j} \left[-(1-\mathbb{I}(q_{ij}=1)) \log(1-\epsilon_{\mathcal{G}}) - \mathbb{I}(q_{ij}=1) \log \epsilon_{\mathcal{G}} \right] \\
&= \log \frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}} \sum_{i,j} \mathbb{I}(q_{ij}=1)) - \sum_{i,j} \log(1-\epsilon_{\mathcal{G}}) \\
&= \log \left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}} \right) |q_\phi(\mathcal{G}|\tau)|_1 + \text{const} \\
&\triangleq \eta |q_\phi(\mathcal{G}|\tau)|_1 + \text{const}
\end{aligned} \tag{44}$$

Therefore, the KL divergence term is equivalent to an ℓ_1 sparsity regularizer in score-based causal discovery [87]. The strength of this regularizer $\eta = \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \in [0, \infty)$. The larger $\epsilon_{\mathcal{G}}$ in prior Bernoulli distribution indicates the smaller strength of this sparsity regularizer (e.g. when $\epsilon_{\mathcal{G}} = \frac{1}{2}, \eta = 0$). In the implementation of data-efficient causal discovery, we adjust the parameter of the classifier to set the strength of this sparsity constraint. ■

A.4.3 Unique Identifiability of Causal Graph

We construct our causal model based on the Factorized MDP in Assumption 1. According to the definition, the causal graph is a directed bipartite graph, with s^t, a^t on the source side, and s^{t+1} on the target side. For the theoretical analysis part in Section A.4.3 and A.2, we denote s^{t+1} as s', s_t as s, a_t as a and $\boldsymbol{x} = \{\mathcal{A} \cup \mathcal{S} \cup \mathcal{S}'\}$, $\boldsymbol{x} \in \mathbb{R}^{2M+N}$ for simplicity.

Definition 7 (Interventional Family $\mathcal{I}$)**.** *For any DAG $\mathcal{G}$, we define the interventional family $\mathcal{I} = (I_1, I_2, \cdots, I_K)$. Here $I_1 := \emptyset$ corresponds to the pure observational setting. The joint distribution for the interventional family can be rewritten as:*

$$p^{(k)}(x_1, \cdots, x_{[2M+N]}) = \prod_{j \notin I_k} p_j^{(1)}(x_j | \boldsymbol{PA}^{\mathcal{G}}(x_j)) \prod_{j \in I_k} p_j^{(k)}(x_j | \boldsymbol{PA}^{\mathcal{G}}(x_j)) \tag{45}$$

Definition 8. *For a specific DAG $\mathcal{G}$, we define $\mathcal{M}(\mathcal{G})$ to be the set of strictly positive densities $p : \mathbb{R}^{2|\mathcal{S}|+|\mathcal{A}|} \to \mathbb{R}$ which satisfies:*

$$p(x_1, \cdots, x_{[2M+N]}) = \prod_{j \in [2M+N]} p_j(x_j | \boldsymbol{PA}^{\mathcal{G}}(x_j)) \tag{46}$$

where $\int_{\mathcal{X}_j} f_j(x_j | \boldsymbol{PA}^{\mathcal{G}}(x_j)) dx_j = 1$ for all $\boldsymbol{PA}^{\mathcal{G}}(x_j) \in \mathcal{X}_j$ and all $j \in [2M+N]$.

Definition 9. *For a specific DAG $\mathcal{G}$ and an interventional family $\mathcal{I}$, we define*

$$\mathcal{M}_{\mathcal{I}}(\mathcal{G}) := \{[p^{(k)}]_{k \in [K]} \mid \forall k \in [K], p^{(k)} \in \mathcal{M}(\mathcal{G}), \forall j \notin I_k, p_j^{(k)}(x | \boldsymbol{PA}^{\mathcal{G}}(x)) = p_j^{(1)}(x | \boldsymbol{PA}^{\mathcal{G}}(x))\} \tag{47}$$

Such set of functions is conherent with condition of strictly positive densities in (46) as well as factorization of interventional distribution in (45).

Definition 10 ($\mathcal{I}$-Markov Equivalence Class, $\mathcal{I}$-MEC)**.** *Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathcal{I}$-Markov equivalent iff $\mathcal{M}_{\mathcal{I}}(\mathcal{G}_1) = \mathcal{M}_{\mathcal{I}}(\mathcal{G}_2)$. We denote by $\mathcal{I} - MEC(\mathcal{G}_1)$ the set of all DAGs which are $\mathcal{I}$-Markov equivalent to $\mathcal{G}_1$, this is the $\mathcal{I}$-Markov equivalence class of $\mathcal{G}_1$.*

Lemma 6 (Sufficient and Necessary Conditions for $\mathcal{I}$-MEC, Yang el. al. [93])**.** *Suppose the interventional family $\mathcal{I}$ is such that $\mathcal{I}_1 := \emptyset$. Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathcal{I}$-Markov equivalent iff their I-DAGs $\mathcal{G}_{\mathcal{I}_1}$ and $\mathcal{G}_{\mathcal{I}_2}$ share the same skeleton and v-structures.*

Proof of Proposition 2. In the bipartite graph $(\mathcal{U}, \mathcal{V}, E)$, for the discovered graph $\hat{\mathcal{G}}$ that is in the $\mathcal{I}$-Markov equivalence class of the ground truth causal graph, $\hat{\mathcal{G}}$ is unique.

Based on the Lemma 6, all possible $\hat{\mathcal{G}}$ that are $\mathcal{I}$-Markov equivalent will share an identical skeleton with G^*, so we consider only graphs obtained by reversing edges in $\hat{\mathcal{G}}$.

Due to the bipartite nature of the transition causal graph defined in Definition 3, for all the v-structured colliders $c \in \mathcal{C}$, we know that $c \in \mathcal{S}'$, therefore, reversing any edge of $\hat{\mathcal{G}}$ will harm the immorality of $\hat{\mathcal{G}}$, and the new graph will no longer be an $\mathcal{I}$-MEC to $\mathcal{G}^*$ Therefore, $\hat{\mathcal{G}}$ is the only graph in the $\mathcal{I}$-MEC of $\mathcal{G}^*$, i.e. $\hat{\mathcal{G}} = \mathcal{G}^*$. ■

A.4.4 Causal Discovery Benefits from Policy Learning

In this section, we would like to show how the learned GCRL model could aid the performance of causal discovery. Before we put the formal proof, we first list several assumptions which is quite common in causal discovery literatures [94].

Assumption 4 (Oracle Conditional Independence Test)**.** *The conditional independent test could tell the independence between any two variables in the causal graph.*

Proof of Lemma 4. Given the oracle conditional independence test in appendix 4, if the state distribution covers all the support of goal distribution, with abundant actions from action space, we can cover all the connections between the current state and the next states.

When $\mathbb{D}_{\text{TV}}(p_{\mathcal{I}_\pi^s}, p_g) < \epsilon_g$, it is sufficient to derive that $p_{\mathcal{I}_\pi^s}(s) > 0, \forall s \in \text{Supp}_g$, where Supp_g is the support set of the goal distribution p_g.

We then discuss the three possible circumstances under such condition:

- Case 1: Both source state and target state distribution in the buffer are fully supported on Supp_g. In this case, given our Assumption 4 and abundant samples, our causal discovery $q_\phi(\mathcal{G}|\tau)$ will correctly classify all the edges in the transition causal graph.
- Case 2: Only the target state distribution is supported on Supp_g, while the source side leaves away from the goal node. In this case, the independent tests may not be able to distinguish the (in)dependence relationship between goal nodes and other state nodes, $\text{SHD}(\hat{\mathcal{G}}, \mathcal{G}^*) \leq |\mathcal{S}| - 1$.
- Case 3: Only source state is supported on Supp_g, while the target side leaves away from the goal node. This case corresponds to the case where some initial states hit the goal, while the learned transition model and policy fail to guide the future states to the goal. The causal discovery model ϕ may falsely classify the edges from all the source states (except the source goal state) towards the target goal states. Thus $\text{SHD}(\hat{\mathcal{G}}, \mathcal{G}^*) \leq |\mathcal{S}| - 1$.

In conclusion, for all the three cases that satisfy $\mathbb{D}_{\text{TV}}(p_{\mathcal{I}_\pi^s}, p_g) \leq \epsilon_g$, we have

$$\max_{\hat{\mathcal{G}} \sim q_\phi(\cdot|\tau)} \Big[\text{SHD}(\hat{\mathcal{G}}, \mathcal{G}^*)\Big] \leq |\mathcal{S}| - 1 \tag{48}$$

thus

$$\mathbb{E}_{\hat{\mathcal{G}} \sim q_\phi(\cdot|\tau)} \Big[\text{SHD}(\hat{\mathcal{G}}, \mathcal{G}^*)\Big] \leq \max_{\hat{\mathcal{G}} \sim q_\phi(\cdot|\tau)} \Big[\text{SHD}(\hat{\mathcal{G}}, \mathcal{G}^*)\Big] \leq |\mathcal{S}| - 1 \tag{49}$$

■

A.5 Overall Performance Guarantee of Iterative Optimization

Based on all the derivation from previous sections, we finally give out the proof of overall performance of our proposed iterative optimization in *GRADER*.

Proof of Theorem 1. Let $d_{max} = \max_{s_1,s_2\in\mathcal{S}} \|s_1 - s_2\|^2, d_\theta = \|\hat{s}'(\theta) - s'\|^2$, then the log likelihood term becomes

$$p_\theta(s'|s,a) \propto \exp(d_{max} - d_\theta) \tag{50}$$

In the model learning part, since we take the log space, we have $\log p_\theta(s'|s,a) = (d_{max} - d_\theta) - C$. We neglect the constant term C when deriving the bound. Without the loss of generality, we set $p_\theta(s'|s,a) = \exp(d_{max} - d_\theta), \log p_\theta(s'|s,a) = d_{max} - d_\theta$ in (4). As $d_{max} - d_\theta \geq 0$, we have the Lipchitz $L \leq 1$ of log function,

$$\begin{aligned}\Big\| \log \hat{p}(s'|s,a) - \log p(s'|s,a) \Big\|_\infty &\leq L \Big\| \hat{p}(s'|s,a) - p(s'|s,a) \Big\|_\infty \\ &\leq \Big\| \hat{p}(s'|s,a) - p(s'|s,a) \Big\|_\infty \leq \epsilon_m\end{aligned} \tag{51}$$

Based on Lemma 2 that is derived in Appendix A.3, we have the policy learning term

$$\begin{aligned}\Big\| \log \hat{\pi}(a|s,g) - \log \pi^*(a|s,g)] \Big\|_\infty &= \Big\| \hat{Q}(s,\hat{\pi}(s)) - Q(s,\pi^*(s)) \Big\|_\infty \\ &= \Big\| Q(s,\hat{\pi}(s)) - Q(s,\pi^*(s)) \Big\|_\infty \\ &= \Big\| V^{\hat{\pi}}(s) - V^{\pi^*}(s) \Big\|_\infty \\ &\leq \frac{\gamma}{(1-\gamma)^2}\epsilon_m\end{aligned} \tag{52}$$

For the KL divergence term, if the goal distribution satisfies $\epsilon_g > \frac{\gamma}{1-\gamma}\epsilon_m$, the following conditions hold:

$$V^{\hat{\pi}}(s) > V^{\pi^*}(s) - \frac{\gamma}{(1-\gamma)^2}\epsilon_m \tag{53}$$

According to Lemma 3 and the condition that $V(s) \in [0, \frac{1}{1-\gamma}]$,

$$\begin{aligned}\mathbb{D}_{\text{TV}}(p_{\mathcal{I}_\pi^s}, p_g) &\leq 1 - (1-\gamma)V^{\hat{\pi}}(s) \\ &< 1 - (1-\gamma)V^{\pi^*}(s) + \frac{\gamma}{1-\gamma}\epsilon_m \\ &= (1-\gamma^{t^*-1}) + \epsilon_g \overset{t^*=1}{=} \epsilon_g\end{aligned} \tag{54}$$

where t^* is the shortest time steps to reach the goal. Here we assume $t^* = 1$ for optimal policy in theoretical design part, while in practice, the bound may get loosen when larger t^* or smaller γ.

Since $\mathbb{D}_{\text{TV}}(p_{\mathcal{I}_\pi^s}, p_g) \leq \epsilon_g$, according to Lemma 4 proved in Appendix A.4, we have

$$\begin{aligned}\Big\| \mathbb{D}_{\text{KL}}(q_\phi||p) - \mathbb{D}_{\text{KL}}(q_\phi^*||p) \Big\|_\infty &= \Big\| \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \|q_\phi(\mathcal{G}|\tau)\|_1 - \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \|q_\phi^*(\mathcal{G}|\tau)\|_1 \Big\|_\infty \\ &= \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \Big\| \|q_\phi(\mathcal{G}|\tau)\|_1 - \|q_\phi^*(\mathcal{G}|\tau)\|_1 \Big\|_\infty \\ &\leq \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \Big\| q_\phi(\mathcal{G}|\tau) - q_\phi^*(\mathcal{G}|\tau) \Big\|_\infty \\ &= \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \max_{\hat{\mathcal{G}}} \Big[\text{SHD}(\mathcal{G}, \mathcal{G}^*)\Big] \\ &\leq \log\left(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\right) \Big(|\mathcal{S}| - 1\Big)\end{aligned} \tag{55}$$

Finally, we can derive the overall performance guarantee as follows:

$$
\begin{aligned}
\|\mathcal{J}^*(\theta,\phi) - \hat{\mathcal{J}}(\hat{\theta},\hat{\phi})\|_\infty &= \Big\| \sum_{t=0}^{T-1} \Big\{ \Big[\log \hat{p}(s^{t+1}|s^t,a^t) - \log p(s^{t+1}|s^t,a^t) \Big] \\
&\quad + \Big[\log \hat{\pi}(a^t|s^t,s^*) - \log \pi^*(a^t|s^t,s^*) \Big] \Big\} + \Big[\mathbb{D}_{\mathrm{KL}}(\hat{q}_\phi||p) - \mathbb{D}_{\mathrm{KL}}(q^*_\phi||p) \Big] \Big\|_\infty \\
&\leq \sum_{t=0}^{T-1} \Big\{ \Big\| \log \hat{p}(s^{t+1}|s^t,a^t) - \log p(s^{t+1}|s^t,a^t) \Big\|_\infty \\
&\quad + \Big\| \log \hat{\pi}(a^t|s^t,s^*) - \log \pi^*(a^t|s^t,s^*) \Big\|_\infty \Big\} + \Big\| \mathbb{D}_{\mathrm{KL}}(\hat{q}_\phi||p) - \mathbb{D}_{\mathrm{KL}}(q^*_\phi||p) \Big\|_\infty \\
&\leq \sum_{t=0}^{T-1} \Big(\epsilon_m + \frac{\gamma}{(1-\gamma)^2}\epsilon_m \Big) + \log\Big(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\Big)\Big(|\mathcal{S}|-1\Big) \\
&= \Big[1 + \frac{\gamma}{(1-\gamma)^2}\Big]\epsilon_m T + \log\Big(\frac{1-\epsilon_{\mathcal{G}}}{\epsilon_{\mathcal{G}}}\Big)\Big(|\mathcal{S}|-1\Big)
\end{aligned}
\tag{56}
$$

■

B Additional Experiment

B.1 Overall Performance

The overall performance results corresponding to the Table 1 for *Stack* and *Unlock* environments are shown in Figure 7 and Figure 8.

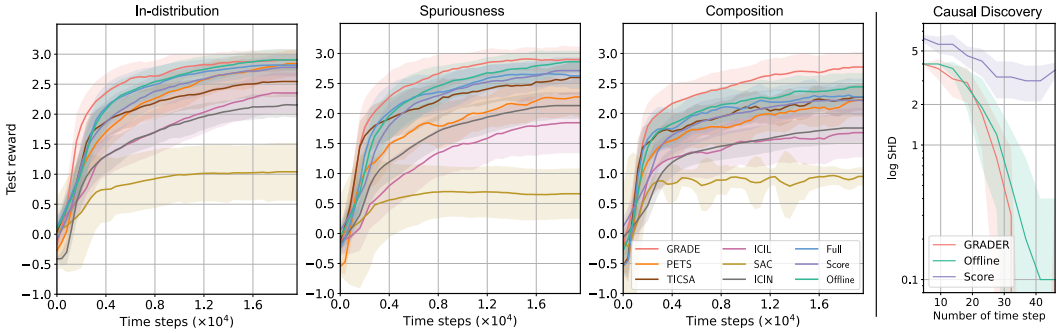

Figure 7: The testing reward and causal discovery results of *Stack* environment.

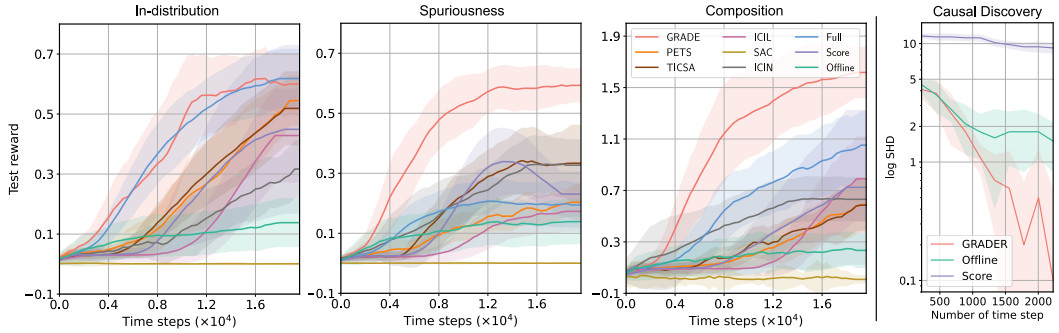

Figure 8: The testing reward and causal discovery results of *Unlock* environment.

In all *Stack* experiments, we find that the advantage of GRADER over other methods is small. The reason is that this task is simple and the true causal graph only contains 7 nodes as shown in

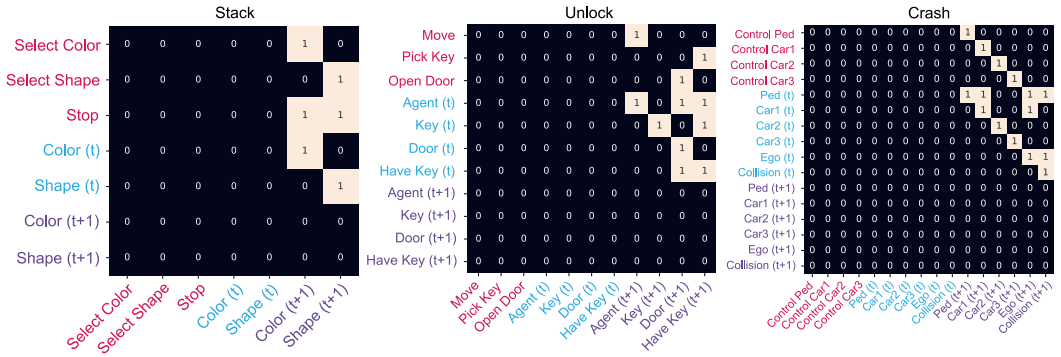

Figure 9: Discovered causal graphs of three environments. Color meaning: Action, State, Next state.

Figure B.2. Due to the simple causal graph, even the Offline random policy can obtain the true causal graph, thus there is almost no difference between the discovery efficiency between GRADER and Offline as shown in the right part of Figure 7.

In the *Unlock-I* experiment, there is no gap between GRADER and Full, which means the causal graph may not have many contributions to solving this task. However, there are large gaps in *Unlock-S* and *Unlock-C* settings since indicating that the causal graph helps the model obtain better generalizable performance. As for the Offline method, since the causal graph is wrongly discovered, the performance is bad in all three settings.

B.2 Causal Graph Analysis

Since the environments we designed have clear and explicit causality, we can get the true causal graph with human analysis. We plot the true causal graphs corresponding to the three environments in Figure 9, where the semantic meanings of all nodes are explained in Appendix C.2.1. We observe that the causal graphs are sparse with very few edges, indicating that non-causal methods that use the full graph may import redundant or even wrong information.

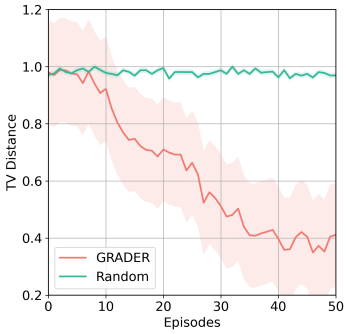

Figure 10: TV distance between goal and state distributions.

B.3 Distance between Goal and State Distribution

In Figure 10, we empirically show the upper bound proved in (28), which describes the TV distance between the goal distribution and the state distribution collected from the GRADER policy. We use 10 trails and plot the mean and standard derivation of the distance. We observe that the distance becomes smaller as the policy gets better in GRADER. This supports our statement that the planning module helps to collect better data samples, which will be used in the causal discovery module. We also plot the distance with a random policy, which is always large since the goal is not easy to be achieved by random actions.

B.4 Task Performance of Chemistry Experiment

In the main context, we only show the discovery results of the Chemistry experiment. We provide the results of task performance in this section. The downstream task is to change the color of nodes to match given colors within maximum steps ($T = 10$). A reward $r = 1$ is received if all colors are matched. Results are reported with 200 episodes. We use planning horizon $H = 5$. We provide the RL downstream task results in Table 3 (ID setting) and Table 4 (OOD setting). The testing reward is shown in Figure 11. The graphs in ID setting have 10 nodes while in OOD setting have 5 nodes. In ID setting, we randomly sample the target colors in the goal. In OOD setting, we set the target colors of all nodes to the same color during the training to create spurious correlation, then randomly set the target colors during testing.

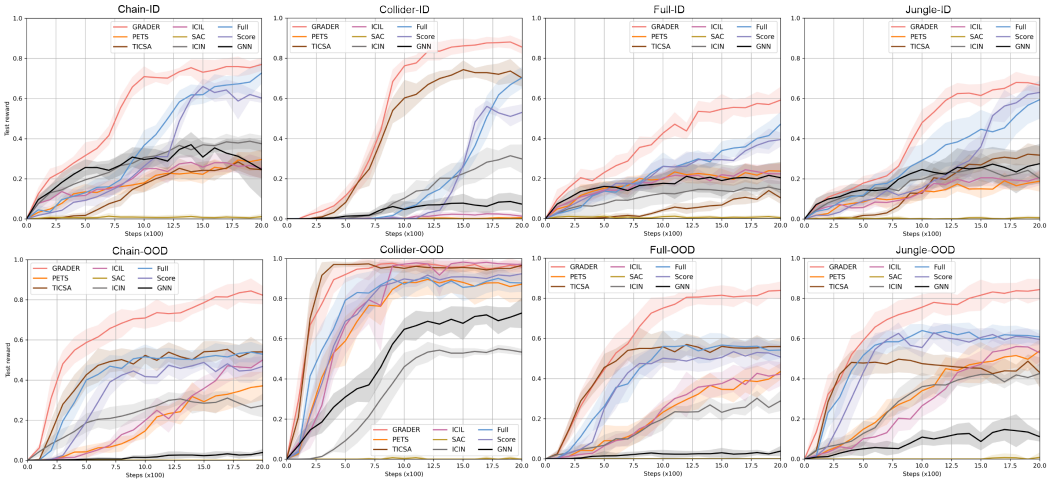

Figure 11: Reward of Chemistry environment under ID and OOD setting.

Table 3: Success rate (%) for Chemistry environment (ID). **Bold** font means the best.

Env	SAC	ICIN	PETS	TICSA	ICIL	GNN	GRADER	Score	Full
Collider	0.0±0.0	29.8±7.2	0.6±0.8	70.1±4.2	1.3±1.3	7.3±4.3	**85.5±3.4**	53.0±4.1	70.3±5.3
Chain	1.1±1.3	37.5±4.0	29.6±4.9	24.5±4.3	25.3±5.1	24.6±14.5	**77.0±3.2**	60.2±2.4	72.7±5.3
Jungle	0.6±0.8	20.2±1.5	18.8±5.2	31.8±4.5	20.6±3.9	27.5±9.8	**69.6±4.3**	63.0±2.3	59.4±9.5
Full	0.5±0.8	4.5±4.0	23.7±4.3	10.4±3.0	22.3±5.1	20.4±7.8	**59.1±6.6**	39.4±3.5	47.1±6.1

C Additional Information

C.1 Details about Conditional Independence Test

In Algorithm 1, we describe the discovery of causal graph with edge inference $e_{ij} \leftarrow q_\phi(\cdot|\mathcal{B}, \eta)$ implemented by conditional independent test. We ignore the details about the test process in the main context thus provide more details in this section.

For discrete variables, we use Pearson's chi-square test[1], which is a statistical test applied to sets of categorical data to evaluate how likely it is that any observed difference between the sets arose by chance. In our experiment, we use the implementation provided by package Scipy [2].

We first define the null hypothesis, which is true when two random variables are statistically independent. These two variables have samples stored in a contingency table O, which has c columns and r rows. Then, the "theoretical frequency" for a cell is:

$$E_{ij} = N_{p_{i\cdot} p_{\cdot j}}, \quad p_{i\cdot} = \sum_{j=1}^{c} \frac{O_{i,j}}{N}, \quad p_{i\cdot} = \sum_{i=1}^{r} \frac{O_{i,j}}{N} \tag{57}$$

where N is the total sample size in the table, $O_{i,j}$ is the sample size of cell (i, j). Then, we can calculate the value of the test-statistic:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{58}$$

Now, we can obtain a p-value (falls in $[0, 1]$) that indicates the significance of this statistic follows the χ^2 distribution from chi-square probability[3]. We compare this p-value with a threshold η and reject

[1] https://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test
[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html
[3] https://people.richland.edu/james/lecture/m170/tbl-chi.html

Table 4: Success rate (%) for Chemistry environment (OOD). **Bold** font means the best.

Env	SAC	ICIN	PETS	TICSA	ICIL	GNN	GRADER	Score	Full
Collider	0.0±0.0	53.3±1.6	87.2±8.5	96.6±1.4	97.0±2.0	72.8±7.5	**95.8±2.6**	92.4±3.5	87.8±4.4
Chain	0.0±0.0	27.3±5.9	37.1±7.0	54.0±3.8	50.0±5.8	3.9±1.6	**82.3±4.5**	46.8±5.0	52.9±4.3
Jungle	0.8±2.4	42.6±4.9	53.9±5.5	43.1±7.5	52.9±7.0	11.1±2.4	**84.4±5.1**	59.5±2.7	60.8±3.5
Full	0.0±0.0	28.9±5.0	43.5±4.1	55.9±4.5	42.2±5.9	3.8±2.5	**83.9±4.4**	50.7±6.0	54.2±4.1

the null hypothesis if p-value is smaller than η. Therefore, the larger we set η, the more likely we find the two variables are dependent. This test process is summarized in Algorithm 2.

If the two variables are continuous, we cannot use the above statistical test anymore. We turn to a more advanced test method proposed in [39]. The general idea is that if $P(X|Y,Z) = P(X,Y)$, Z is not useful as a feature to predict X. To achieve this, the authors propose to use decision tree regression to predict Y using both X, Z, and also using Z only.

Algorithm 2: Independence Test for Discrete Variables.

Input: A contingency table O with samples for two variables X and Y.
Define Null hypothesis: X and Y are independent.
Calculate $p_{i\cdot} = \sum_{j=1}^{c} \frac{O_{i,j}}{N}$ and $p_{i\cdot} = \sum_{i=1}^{r} \frac{O_{i,j}}{N}$
Calculate expected frequencies $E_{ij} = N_{p_{i\cdot} p_{\cdot j}}$
Calculate the chi-square statistic $\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$
Obtain p-value p from chi-square probability
if $p < \eta$ **then**
| Reject the Null hypothesis, i.e., X and Y are dependent.

C.2 Experiment Details

C.2.1 Environment Design

More details about the design of the environments are summarized below:

Stack: Manipulation is important for house-holding and factory assembly. Sometimes, the color of the object is not relevant to the task but may leak information by sharing spuriousness with the task. Also, the goal could compose several previously seen goals such as repeating similar actions. Totally, we have 5 different shapes and 5 different colors and the goals are some combinations of the colors and shapes. At each step, the agent can either stack an object with a chosen color and shape or stop stacking. The state is the colors and shapes of all current objects. The agent receives a positive reward when the goal is achieved and a punishment if it stacks a new object.

Unlock: Collecting specific objects to fulfill required conditions is useful for mobile robots. The causality in this environment exists between the key and door. The action contains six operations, including four-direction movements (Move), pick key (Pick Key), and open door (Open Door). The state is the position of the agent, the position of the key, and the status of the door. In the first generalization setting, we intentionally create a spurious correlation between the position of the key and the door. If the agent figures out that key can open the door no matter what its position is, it will ignore the spurious correlation. In the second generalization setting, we increase the number of the door from one to two. This setting contains two same sub-tasks and can be used to test the compositional generalization.

Crash: The causality in this environment mainly exists between the pedestrian (Ped), the ego vehicle (Ego), and another vehicle (Car 1) [95]. The collision between Ped and Ego only happens when the view of Ego is blocked by Car 1. To make this happen, we design a rule-based AV, which will brake if it detects any obstacles within a certain distance. Therefore, if the pedestrian directly hits the AV, the AV will stop and the crash will not happen. To make this task difficult, we also place two other vehicles (Car 2 and Car 3) on the scene but they will not interrupt the crash scenario. The agent can control the acceleration and steering of Ped, Car 1, Car 2, and Car 3. The state is the position and velocity of all objects plus the status of whether a collision happens. To create a spurious correlation, we fix the initial distance between Ego and Ped to a constant since this creates a shortcut for the

feature extractor. However, remembering this distance is not enough since we change the initial distance in the testing stage.

Table 5: Environment configurations used in experiments

Parameters	Stack	Unlock	Crash	Chemistry
Max step size	5	15	30	10
State dimension	50	110	22	100
Action dimension	12	8	8	100
Action type	Discrete	Discrete	Continuous	Discrete

C.2.2 Model Structures and Hyper-parameters

Since different nodes have different dimensions, we design a set of encoders E_j and a set of decoders D_j to convert the dimension of features. Thus, the entire model structure is

$$s_j^{t+1} = D_j(f_{\theta_j}(E_j([\mathbf{PA}_j^{\mathcal{G}}]^t, N_j))), \quad \forall j \in [M] \tag{59}$$

We list all important hyper-parameters in the implementation for three environments in Table 6.

C.3 Broader Social Impact and Additional Limitation

C.3.1 Broader Social Impact

We identify several important social impacts of our proposed method, including both positive and potential negative impacts:

1) Incorporating causality into reinforcement learning methods increases both the interpretability and generalizability of artificial intelligence, which helps users easily check the working progress of agents and the source of failures.
2) Insufficient data and training may cause flawed causal graphs, which may lead to a wrong understanding of the causation of the task. This wrong understanding of the task may cause risky and irrational actions of agents.
3) The discovered causal graph could be accessed and modified by users to manipulate the behaviors of agents on purpose. If the task contains private information, the discovered causal graph may cause privacy issues when the graph is interpreted by other users.

To mitigate the potential negative societal impacts mentioned above, we encourage research to follow these instructions:

1) People should always check the convergence of the causal discovery step and verify the discovered causal graph with domain knowledge.
2) The discovered causal graph should be frequently checked and verified with the training data to ensure its correctness. The causal graphs also need to be encrypted and only accessible to algorithms and trustworthy users.

C.3.2 Additional Limitation

Causal discovery methods. The gradient-based discovery method are widely investigated recently for large datasets since they have good scalability. However, these methods also require lots of training data to converge. In Online RL, we don't have enough data at the beginning of training. Thus, constraint-based methods are more suitable for causal RL tasks.

Although our constraint-based causal discovery does not scale as well as the score-based methods, our proposed independence tests achieve a time complexity of $\Omega(|S|(|S| + |A|))$, which is tolerable for most RL problems with lower dimensional state space. Empirical studies also show that our independent tests enjoy better data efficiency.

Assumptions in our theoretical analysis. Faithfulness and Markov properties are commonly used in causal discovery literature such as [87, 38]. It is claimed in [38] that the oracle independent test can

Table 6: Hyper-parameters of models used in experiments

Models	Parameters	Environment			
		Stack	Unlock	Crash	Chemistry
GRADER	Learning rate	0.001	0.001	0.0001	0.001
	Size of buffer $\mathcal{B}$	4000	10000	10000	4000
	Epoch per iteration	20	5	10	20
	Batch size	256	256	256	256
	Planning horizon H	5	10	20	5
	Planning population	500	100	1000	700
	Reward discount γ	0.99	0.99	0.99	0.99
	ϵ-greedy ratio	0.4	0.4	0.5	0.5
	Causal Discovery η	0.01	0.01	0.01	0.01
	GRU hiddens	32	64	128	32
PETS*	MLP hiddens	32	64	128	32
	MLP layers	2	2	2	2
	Ensemble number	5	5	5	5
TICSA*	Size of buffer $\mathcal{B}$	20000	400000	40000	20000
	Pretrain buffer	200	2000	5000	200
	Initialized mask coef.	1.0	1.0	1.0	1.0
	MLP hiddens	32	64	128	32
	Sparsity regularizer	0.5	1.0	0.2	0.5
ICIL*	Size of buffer $\mathcal{B}$	20000	400000	40000	20000
	Learning rate of MINE	0.0001	0.0001	0.0001	0.0001
	MLP hiddens	32	64	128	32
	MINE hiddens	32	64	128	32
	Env. Numbers	5	3	3	5
SAC	Learning rate	0.001	0.001	0.0001	0.001
	Size of buffer $\mathcal{B}$	4000	10000	10000	4000
	Update step τ	0.005	0.005	0.0001	0.005
	Update iteration	3	3	3	3
	Entropy α	0.2	0.2	0.2	0.2
	Batch size	256	256	256	256
	Reward discount γ	0.99	0.99	0.99	0.99
	MLP hiddens	64	128	256	64
ICIN	Learning rate	0.001	0.001	0.0001	0.001
	Size of buffer $\mathcal{B}$	4000	10000	10000	4000
	Batch size	256	256	256	256
	MLP hiddens	64	128	256	64
	MLP layers	3	3	3	3

* Use the same planning parameters as GRADER.

be ensured by the satisfaction of Markov property and faithfulness. Recent work [94] also assumes the oracle of conditional independent test in its Assumption 2.1. Practically, the oracle test can be implemented with certain sub-linear sample complexity, as is investigated in [40].

In reinforcement learning tasks, agents interact with the environment by doing interventions, which is achieved by assigning values to action nodes. Then, the intervention results are reflected by the states. Under fully observable Markov settings, the value of these states contains all information about the intervention. Thus, our RL setting usually satisfies the assumptions we use in the theoretical proof.