Tackling Non-Stationarity in Reinforcement Learning via Causal-Origin Representation

Wanpeng Zhang Peking University
Yilin Li Peking University
Boyu Yang Fudan University
Zongqing Lu Peking University

Abstract

In real-world scenarios, the application of reinforcement learning is significantly challenged by complex non-stationarity. Most existing methods attempt to model the changes of the environment explicitly, often requiring impractical prior knowledge. In this paper, we propose a new perspective, positing that non-stationarity can propagate and accumulate through complex causal relationships during state transitions, thereby compounding its sophistication and affecting policy learning. We believe that this challenge can be more effectively addressed by tracing the causal origin of non-stationarity. To this end, we introduce the **C**ausal-**O**rigin **REP**resentation (**COREP**) algorithm. COREP primarily employs a guided updating mechanism to learn a stable graph representation for states termed as causal-origin representation. By leveraging this representation, the learned policy exhibits impressive resilience to non-stationarity. We supplement our approach with a theoretical analysis grounded in the causal interpretation for non-stationary reinforcement learning, advocating for the validity of the causal-origin representation. Experimental results further demonstrate the superior performance of COREP over existing methods in tackling non-stationarity.

1 Introduction

The rapid progress in reinforcement learning (RL) [1, 2] over recent years has led to impressive performance improvements across a diverse range of applications [3, 4]. However, a common assumption in many RL algorithms is the stationarity of the environment, which can limit their applicability in real-world scenarios characterized by varying dynamics [5, 6]. While recent meta-RL [7] methods have shown some promise in addressing non-stationarity through adaptation [8], their performance often degrades when facing more complex changes in the dynamics [9, 10]. Several recent works, such as FN-VAE [10] and LILAC [11], have made strides towards improving RL algorithms in non-stationary environments by explicitly modeling the change factors of the environment. Nevertheless, they may not comprehensively capture the complexity of real-world non-stationarity.

In this paper, we propose a novel setting for efficiently tackling non-stationarity in RL from a new perspective inspired by the causality literature [12, 13]. We argue that minor changes in dynamics can cause significant shifts in observations due to their propagation through intricate causal relationships among state elements. We believe that this challenge can be more effectively addressed by tracing the causal origin of non-stationarity. However, directly constructing an accurate causal graph of observations in non-stationary environments is challenging due to the instability of dynamics [14]. To overcome this challenge, we propose the **COREP** (**C**ausal-**O**rigin **REP**resentation) algorithm. COREP primarily utilizes a guided updating mechanism to learn a stable graph representation for states termed as causal-origin representation.

Preprint. Under review.

We first propose a novel formulation of non-stationarity in RL as a complex mixture of stationary environments. We take the masks defined in transition functions as causal relationships and assume they are invariant within each environment. However, without access to any type of prior information about the environment, it is quite challenging to identify the causal relationship of each environment. Therefore, we consider an environment-shared representation defined by the union of maximal ancestral graphs (MAG) from each environment, and utilize it to design a stable RL approach. In line with the concept of using the union of MAGs as the causal interpretation for non-stationary RL, we introduce a dual graph structure, termed core-graph and general-graph. The core-graph is designed for learning a stable graph representation, guided by a TD error-based updating mechanism. As the core-graph primarily concentrates on learning the most vital parts of the graph representation, some edges might be discarded in the process. Thus, a continuously updating general-graph is designed to compensate for this potential information loss and to improve the algorithm's adaptability. Ultimately, we integrate the core-graph and general-graph to construct the causal-origin representation, providing a comprehensive understanding of the environment's dynamics and significantly mitigating the impact of non-stationarity.

Specifically, our method starts by transforming observed states into node matrices. We then generate weighted adjacency matrices from the nodes to serve as inputs for two Graph Attention Networks (GAT) [15], *i.e.*, core-GAT and general-GAT, representing the core-graph and general-graph, respectively. As a variant of Graph Neural Network (GNN) [16, 17], GAT can efficiently manage directed acyclic graphs (DAG) by prioritizing neighboring nodes via masked attention. This capability aligns perfectly with our goal of tracing the origins of non-stationarity through causal relationships. Then we determine whether to update the core-GAT by checking if the recent TD errors lie within the confidence interval of the replay buffer. The choice of using TD error as the metric for detection is based on the fact that significant changes in TD error imply notable alterations in the causal graph of observations, since TD error serves as a performance indicator for the learned policy in the current environment. This dual-GAT structure is then incorporated into a Variational AutoEncoder (VAE) [18] to further enhance the learning efficiency. Our main contributions can be summarized as follows:

- We provide a causal interpretation for non-stationary RL and propose a novel setting that focuses on the causal relationships within states and learn the causal-origin representation to tackle more complex non-stationarity problems in RL.
- Based on the proposed formulation and setting, we design a modular algorithm that can be readily integrated into existing RL algorithms.
- We provide a theoretical analysis that offers both inspiration and theoretical support for our algorithm. Experimental results on a variety of non-stationary environments further demonstrate the effectiveness of our algorithm.

2 Preliminaries

Problem Formulation. Reinforcement learning problems are typically modeled as Markov Decision Processes (MDPs), defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the transition probability, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. We may also use the form of transition function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ when the environment is deterministic. The goal of an agent in RL is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward, defined as the value function $V^\pi(\boldsymbol{s}) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | \boldsymbol{s}_0 = \boldsymbol{s}]$, where r_t is the reward at time step t.

In non-stationary environments, the dynamics of the environment change over time. Our goal is to learn a policy π that can adapt to the non-stationary environment and still achieve high performance. For simplicity of notations, we provide theoretical analysis with the form of deterministic transition function f, while the results still hold for the probabilistic setting.

Causal Structure Discovery. Causal structure discovery usually aims at inferring causation from data, modeled with a directed acyclic graph (DAG) $\mathcal{D} = (V, E)$, where the set of nodes V includes the variables of interest, and the set of directed edges E contains the direct causal effects between these variables [19]. The causal graph is a practical tool that relates the conditional independence relations in the generating distribution to separation statements in the DAG (d-separation) through the Markov property [20]. If unobserved confounders exist, maximal ancestral graphs (MAGs) $\mathcal{M} = (V, D, B)$

are often used to represent the observed variables by generalizing DAGs with bidirected edges which depicts the presence of latent confounders between pairs of variables, see *e.g.* [21] for details. The sets D, B stand for directed and bidirected edges, respectively.

Graph Neural Networks. Graph neural networks (GNNs) are a class of neural networks specially designed for graph-structured data. Given a graph $\mathcal{G} = (V, E)$, GNNs aim at learning a vector representation for each node $v \in V$ or the entire graph $\mathcal{G}$, leveraging the information of both graph structure and node features. Modern GNNs usually follow the message passing framework [22]. Layers in GNN can be formulated as $\boldsymbol{H}^{(l)} = \sigma\big(\sum_s \boldsymbol{L}_s \boldsymbol{H}^{(l-1)} \boldsymbol{W}_s^{(l)}\big)$, where $\boldsymbol{H}^{(l)}$ is the node representations output by the l-th layer ($\boldsymbol{H}^{(0)}$ is the initial input node features), $\boldsymbol{L}_s$ is the s-th convolution support which defines how the node features are propagated, $\boldsymbol{W}_s^{(l)}$ is learnable parameters for the s-th convolution support in the l-th layer, and $\sigma(\cdot)$ is the activation function. Graph Attention Network (GAT) [15] is a special type of GNN following the message passing framework. Instead of handcrafting, the self-attention mechanism is used to compute the support convolutions in each GAT layer, where the adjacency matrix plays the role as a mask matrix for computing the attention.

3 Methodology

3.1 Key Idea

In the **C**ausal-**O**rigin **REP**resentation (**COREP**) algorithm, the primary goal is to address the issue of non-stationarity in RL by learning the underlying graph structure of the environment termed as causal-origin representation, which is desired to be causal and stable. This is achieved by designing a dual GAT structure, *i.e.*, a core-GAT and a general-GAT. The core-GAT is designed to learn the most essential parts of the environment's graph and its learning is controlled by a guided updating mechanism. The general-GAT, on the other hand, is continuously updated to compensate for any information that the core-GAT might overlook, and together they form a comprehensive understanding of the environment's dynamics.

The causal-origin representation involves transforming states into node features and generating the weighted adjacency matrix. A self-attention mechanism is then applied to these nodes to compute the graph representation. To improve the learning efficiency, the causal-origin representation is incorporated into the VAE framework. To guide the updating of the core-GAT, a TD error-based detection mechanism is employed. Furthermore, regularization terms are introduced to enhance the identifiability and guarantee the structure of the causal-origin representation.

3.2 Causal Interpretation for Non-Stationary RL

In this part, we will provide a causal interpretation for non-stationarity in RL, which provides us with inspiration and theoretical support for the algorithm design. We assume that the underlying dynamics can be described in terms of functions f_i, g_j, k:

$$\begin{aligned} s_i' &= f_i\left(\boldsymbol{c}_i^{s\to s} \odot \boldsymbol{s}, \boldsymbol{c}_i^{a\to s} \odot \boldsymbol{a}, \varepsilon_i^s\right), \forall i = 1, \cdots, d_s, \\ h_j' &= g_j\left(\boldsymbol{c}_j^{h\to h} \odot \boldsymbol{h}, \boldsymbol{c}_j^{s\to h} \odot \boldsymbol{s}, \boldsymbol{c}_j^{a\to h} \odot \boldsymbol{a}, \varepsilon_j^h\right), \forall j = 1, \ldots, d_h, \\ r &= k\left(\boldsymbol{c}^{s\to r} \odot \boldsymbol{s}, \boldsymbol{c}^{h\to r} \odot \boldsymbol{h}, \boldsymbol{c}^{a\to r} \odot \boldsymbol{a}, \varepsilon^r\right), \end{aligned} \tag{3.1}$$

where $\boldsymbol{s}$ denotes the state with dimension d_s, $\boldsymbol{h}$ denotes the hidden state with dimension d_h, r represents the reward, s_i' is the i-th element of next state $\boldsymbol{s}'$, h_j' is the j-th element of next hidden state $\boldsymbol{h}'$, and $\odot$ denotes the Hadamard product. The random noises $\varepsilon_i^s, \varepsilon_j^h, \varepsilon^r$ are independent, and the distribution of each noise is identical over time. We consider the non-stationarity as the change of binary masks $\boldsymbol{c}^{\cdot\to\cdot}$. The masks represent the structural dependence in the following way. For example, in function f_i of Equation (3.1), the j-the element of $\boldsymbol{c}_i^{s\to s} \in \{0,1\}^n$ equals to 1 if and only if s_j directly affects s_i. Other masks are defined in the same way. We rewrite the masks in matrix forms, letting $\boldsymbol{C}^{\cdot\to s} := [\boldsymbol{c}_i^{\cdot\to\cdot}]_{i=1}^{d_s}$, $\boldsymbol{C}^{\cdot\to h} := [\boldsymbol{c}_j^{\cdot\to\cdot}]_{j=1}^{d_h}$. The masks $\boldsymbol{C}^{\cdot\to\cdot}$ are allowed to be time-varying. Some previous research assumed that the masks are invariant across time, and encoded the changes over environments into some change factors [10, 23]. However, we do not rely on any assumptions of such change factors. Instead, we propose a novel causal interpretation based on an environment-shared union graph representation to capture the transition information in the non-stationary environment.

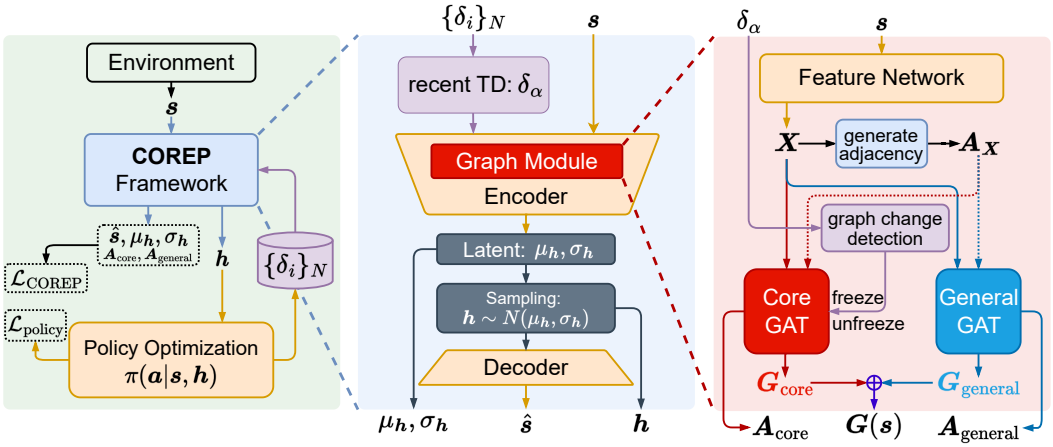


Figure 3.1: Overview of the COREP framework. (1) The left part illustrates that the COREP framework can be seamlessly incorporated into any RL algorithm. It takes the state of the environment as input and outputs the causal-origin representation for policy optimization. The TD errors are stored for subsequent graph change detection. (2) The middle part showcases the VAE structure employed by the COREP framework, which is utilized to enhance the learning efficiency. (3) The right part highlights the key components of COREP. The dual GAT structure is designed in line with the concept of causal-origin representation to retain the essential parts of the graph. The TD error detection can guide the Core-GAT to learn the environment-shared union graph based on our theory. The General-GAT is continuously updated to compensate for the loss of information. Ultimately, this framework utilizes the causal-origin representation for policy optimization to tackle the non-stationarity problem.

We regard the non-stationary environment as a dynamic mixture of various stationary environments. In other words, from this perspective, the non-stationarity of the environment can be interpreted as the variations over time in this mixture distribution. We assume there are an unknown number of K distinct stationary environments in total. If a sample is from the k-th environment, then there exists mask $\boldsymbol{C}^{\cdot\to\cdot}_{(k)}, k \in \{1, \ldots, K\}$, such that the current $\boldsymbol{C}^{\cdot\to\cdot}_{\text{current}} = \boldsymbol{C}^{\cdot\to\cdot}_{(k)}$, *i.e.*, the masks are invariant for each environment. The relationships among variables determined by masks can also be characterized by DAGs $\mathcal{D}_{(k)} = (V, E_{(k)})$ over the same set of nodes $V = \{\boldsymbol{s}, \boldsymbol{h}, \boldsymbol{a}, \boldsymbol{s}', \boldsymbol{h}', \boldsymbol{a}', r, e\}$, where $e \in \{1, \ldots, K\}$ is a node denoting the label of the environment with in-degree 0. The set of directed edges $E_{(k)}$ contains an edge from one node to another node if and only if the corresponding mask $\boldsymbol{c}^{\cdot\to\cdot}_{(k)} = 1$, and it also contains additional edges from e to $(\boldsymbol{s}, \boldsymbol{h})$ whose marginal distributions vary across environments. In this setting, non-stationarity can also be reflected by the changes in the underlying graph structure governing the interactions among different variables, and such changes are allowed to be either within-episode or across-episode.

It is challenging to learn the structure of the causal graph independently when the environment label is unavailable, as the variable e can be considered a latent confounder that leads to spurious correlations. In the presence of unobservable nodes, maximal ancestral graphs (MAGs) are a useful representation that generalizes DAGs [21]. For each DAG $\mathcal{D}_{(k)}$, we can construct a corresponding MAG $\mathcal{M}_{(k)}$[24], see Algorithm A.1 in Appendix A. The MAGs make use of bidirected edges ($\leftrightarrow$) to characterize the change of marginal distribution of $\boldsymbol{s}, \boldsymbol{h}$ over different environments. To model the structural relationships in the non-stationary RL with a unified approach, we further encode the relations among all actions, states, hidden states, and rewards with an environment-shared union graph $\mathcal{M}_\cup$. The definition of the environment-shared union graph is as follows.

Definition 3.1 (Environment-shared union graph)**.** *The environment-shared representation union graph $\mathcal{M}_\cup := (V, D, B)$ has the set of nodes V, and the set of directed edges*

$$D = \{u \to v : u, v \in V, \exists k \text{ such that } u \to v \text{ in } \mathcal{M}_{(k)}\},$$

and the set of bidirected edges

$$B = \{u \leftrightarrow v : u, v \in V, \exists k \text{ such that } u \leftrightarrow v \text{ in } \mathcal{M}_{(k)}\}.$$

The above defined union graph $\mathcal{M}_\cup$ contains no cycle because Equation (3.1) implies that there exists a common topological ordering for $\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(K)}$, see Appendix A for details. Without knowing the label of environment k, we cannot generally identify the structure of $\mathcal{M}_{(k)}$ for each k from the observed data. However, we show that $\mathcal{M}_\cup$ is still a MAG, hence any non-adjacent pair of nodes is d-separated given some subset of nodes.

Proposition 3.1. *Suppose the generative process follows Equation (3.1), then there exists a partial order π on V such that for all $i \in \{1, \ldots, K\}$, it holds that (a) u is an ancestor of $v \Rightarrow u <_\pi v$ in $\mathcal{M}_{(k)}$; and (b) $u \leftrightarrow v \Rightarrow u \not\lessgtr_\pi v$ in $\mathcal{M}_{(k)}$. As a consequence, the environment-shared representation union graph $\mathcal{M}_\cup$ is a MAG.*

We provide the full proofs and a detailed explanation of conditions in Appendix A. Proposition 3.1 provides theoretical support for recovering the structure of environment-shared union graph $\mathcal{M}_\cup$. In the following sections, we will describe how the COREP algorithm learns a policy that is stable under non-stationarity by utilizing the environment-shared representation union graph $\mathcal{M}_\cup$.

3.3 Dual Graph Attention Network Structure

In accordance with Proposition 3.1, we aim to learn the causal-origin representation encapsulating the environment-shared union graph $\mathcal{M}_\cup$. For this purpose, we propose the structure of dual Graph Attention Networks (GAT), *i.e.*, core-GAT and general-GAT. The core-GAT is designed for learning the stable graph representation. We consequently control the update of core-GAT by employing TD error as a detector of significant changes in the environment's underlying graph structure. It is important to note that COREP uses the TD-error detection merely as an indicator of substantial environmental changes to decide whether or not to update the core-GAT, so we do not need a complex detection mechanism like other methods [25, 26] that are required to explicitly recognize the specific changes in the environment.

Since the core-GAT primarily focuses on learning the most essential part of the graph representation, some edges may be overlooked or lost in the process. To compensate for this potential loss of information and to enhance the algorithm's adaptation capabilities, we introduce the continuously updating general-GAT. In the end, we integrate the core-GAT and general-GAT to construct the causal-origin representation, thereby providing a comprehensive understanding of the environment's dynamics and significantly mitigating the impact of non-stationarity on decision-making in RL.

Specifically, we first transform states into node features using an MLP network $f_{\text{MLP}} : \mathbb{R}^{d_s} \to \mathbb{R}^{N \cdot d_f}$, then reshape the output into node feature matrix $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^{N \times d_f}$, where N is the number of nodes, and d_f is the number of features in each node $\boldsymbol{x}_i$. We then compute the weighted adjacency matrix which represents the probabilities of edges by using Softmax on the similarity matrix of nodes:

$$\boldsymbol{A}_{\boldsymbol{X}} = \text{Softmax}\left(\boldsymbol{X}\boldsymbol{X}^{\text{T}} \odot (\boldsymbol{1}_N - \boldsymbol{I}_N)\right), \tag{3.2}$$

where $\boldsymbol{1}_N \in \mathbb{R}^{N \times N}$ represents the matrix with all elements equal to 1, $\boldsymbol{I}_N \in \mathbb{R}^{N \times N}$ represents the identity matrix, and $\odot$ denotes the Hadamard product. Multiplying $(\boldsymbol{1}_N - \boldsymbol{I}_N)$ is to remove the self-loop similarity when computing the weighted adjacency matrix. A learnable weight matrix $\boldsymbol{W} \in \mathbb{R}^{d_f \times d_g}$ is then applied to the nodes for transforming $\boldsymbol{X}$ into graph features $\boldsymbol{X}\boldsymbol{W} \in \mathbb{R}^{N \times d_g}$ where d_g denotes the dimension of the graph feature. We can then perform the self-attention mechanism on the nodes, *i.e.*, $\alpha_{ij} = \text{attention}(\boldsymbol{x}_i\boldsymbol{W}, \boldsymbol{x}_j\boldsymbol{W} | \boldsymbol{A}_{\boldsymbol{X}})$. The conditioned $\boldsymbol{A}_{\boldsymbol{X}}$ allows us to perform the masked attention, *i.e.*, we only compute α_{ij} for node $j \in \mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})$ where $\mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})$ is the neighbor set of node i computed by the weighted adjacency matrix $\boldsymbol{A}_{\boldsymbol{X}}$. We can consider deeper-depth neighbors of each node by combining multiple $\text{attention}(\cdot)$ into a multi-layer network.

For the n-th graph attention layer, the coefficients computed by the self-attention mechanism can be specifically expressed as:

$$\alpha_{ij} = \frac{\delta_{\mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})}(j) \cdot \exp\left(\sigma\left(\boldsymbol{l}_n \left[\boldsymbol{x}_i\boldsymbol{W} \oplus \boldsymbol{x}_j\boldsymbol{W}\right]^{\text{T}}\right)\right)}{\sum_{k \in \mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})} \exp\left(\sigma\left(\boldsymbol{l}_n \left[\boldsymbol{x}_i\boldsymbol{W} \oplus \boldsymbol{x}_k\boldsymbol{W}\right]^{\text{T}}\right)\right)}, \tag{3.3}$$

where $\oplus$ is the concatenation operation, σ is the activation function, $\boldsymbol{l}_n \in \mathbb{R}^{2d_g}$ is the learnable weight vector for the n-th graph attention layer, and $\delta_{\mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})}(j)$ is the indicator function, *i.e.*, $\delta_{\mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})}(j) = 1$ if $j \in \mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})$ otherwise 0.

We employ the multi-head attention [27] to stabilize the learning process. Specifically, we perform M separate self-attention mechanisms as Equation (3.3) in each layer. Subsequently, the resulting features are concatenated to form the graph node:

$$\boldsymbol{g}_i = \mathop{\Big\Vert}_{m=1}^{M} \sigma\Big(\sum_{j \in \mathbf{N}_i(\boldsymbol{A}_{\boldsymbol{X}})} \alpha_{ij}^{(m)} \boldsymbol{x}_j \boldsymbol{W}^{(m)} \Big). \tag{3.4}$$

The core-GAT and general-GAT output $\boldsymbol{G}_{\text{core}}$ and $\boldsymbol{G}_{\text{general}}$, respectively, which are concatenated to form the final causal-origin representation. Specifically, we denote the entire process of obtaining the causal-origin representation from $\boldsymbol{s}$ as a function $\boldsymbol{G} : \mathbb{R}^{d_s} \to \mathbb{R}^{N \times 2Md_g}$, such that $\boldsymbol{G}(\boldsymbol{s}) \doteq \boldsymbol{G}_{\text{core}} \oplus \boldsymbol{G}_{\text{general}} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N\}_{\text{core}}^{\text{T}} \oplus \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N\}_{\text{general}}^{\text{T}}$.

3.4 Guided Updating for Core-GAT

As the theoretical analysis in Section 3.2, our goal is to learn the causal-origin representation encapsulating the environment-shared union graph $\mathcal{M}_{\cup}$. To achieve this, we design a TD error-based detection mechanism to guide the updating of core-GAT. Specifically, we store the computed TD errors of policy optimization into the TD buffer $\mathcal{B}_\delta$. We compute the mean of recent TD errors as $\delta_\alpha = \left(\sum_{|\mathcal{B}_\delta|-\alpha|\mathcal{B}_\delta|<k<|\mathcal{B}_\delta|} \delta_k\right) / \alpha|\mathcal{B}_\delta|$, where α controls the proportion of recent TD errors for detection, and $|\mathcal{B}_\delta|$ indicates the number of elements in $\mathcal{B}_\delta$. We then check whether δ_α lies within the confidence interval $(\mu_\delta - \eta\sigma_\delta, \mu_\delta + \eta\sigma_\delta)$, where $\mu_\delta, \sigma_\delta$ are the mean and standard deviation of the TD buffer, and η represents the confidence level. If the recent TD error δ_α lies within this interval, we freeze the weights of the core-GAT and halt its updates; otherwise, we unfreeze its weights and proceed with updating the core-GAT.

As we previously discussed, since the core-GAT primarily focuses on learning the most essential part of the graph representation, some edges might be discarded in the process. To compensate for the loss of information, we add a continuously updating general-GAT and guide the learning of core-GAT by introducing a regularization that penalizes the difference between the output adjacency matrices of the core-GAT and general-GAT:

$$\mathcal{L}_{\text{guide}} = \|\boldsymbol{A}_{\text{core}} - \boldsymbol{A}_{\text{general}}\|_2. \tag{3.5}$$

3.5 Incorporated with Variational AutoEncoder

To improve the efficiency of learning the causal-origin representation, we incorporate the causal-origin representation into the Variational AutoEncoder (VAE) framework [18]. Specifically, we feed the output $\boldsymbol{G}(\boldsymbol{s})$ into the VAE inference process to derive the mean and variance $(\mu_{\boldsymbol{h}}, \sigma_{\boldsymbol{h}})$ of the latent representation $\boldsymbol{h}$. Subsequently, we can sample $\boldsymbol{h} \sim \mathcal{N}(\mu_{\boldsymbol{h}}, \sigma_{\boldsymbol{h}}), \boldsymbol{h} \in \mathbb{R}^{d_h}$. The loss function for VAE is defined as

$$\begin{aligned} \mathcal{L}_{\text{VAE}}(\boldsymbol{s}; \theta, \phi) &= \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{G}(\boldsymbol{s}))} \left[\log p_\theta(\boldsymbol{G}(\boldsymbol{s})|\boldsymbol{h})\right] - \text{KL}\left[q_\phi(\boldsymbol{h}|\boldsymbol{G}(\boldsymbol{s}))||p(\boldsymbol{h})\right] \\ &\approx \text{MSE}(\boldsymbol{s}, \hat{\boldsymbol{s}}) - \text{KL}\left[q_\phi\left(\boldsymbol{h}|\boldsymbol{G}(\boldsymbol{s})\right)||\mathcal{N}(0, \boldsymbol{I})\right], \end{aligned} \tag{3.6}$$

where p_θ, q_ϕ represent the parameterized decoder and encoder respectively, $\text{KL}(\cdot)$ denotes the Kullback-Leibler divergence, and $\text{MSE}(\boldsymbol{s}, \hat{\boldsymbol{s}})$ is an estimation of $\mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{G}(\boldsymbol{s}))} \left[\log p_\theta(\boldsymbol{G}(\boldsymbol{s})|\boldsymbol{h})\right]$ which measures the mean square error between the original state and the reconstructed state with the causal-origin representation. It is noteworthy that the VAE structure serves solely as a tool to enhance learning efficiency, therefore it is not a strictly necessary component of our COREP algorithm. The latent $\boldsymbol{h}$ is then provided to the policy $\pi(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{h})$ for policy optimization.

3.6 Training and Policy Optimization

To further enhance the identifiability, we introduce the regularization terms for the sparsity and the MAG structure:

$$\begin{aligned} \mathcal{L}_{\text{sparsity}} &= \|\boldsymbol{A}_{\text{core}}\|_1 + \|\boldsymbol{A}_{\text{general}}\|_1, \\ \mathcal{L}_{\text{MAG}} &= \|\boldsymbol{A}_{\text{core}} - \boldsymbol{A}_{\text{core}}^{\text{T}}\|_2 + \|\boldsymbol{A}_{\text{general}} - \boldsymbol{A}_{\text{general}}^{\text{T}}\|_2, \end{aligned} \tag{3.7}$$

where $\mathcal{L}_{\text{MAG}}$ is used for penalizing the asymmetry of the adjacency matrices to meet the design of the MAG which utilizes bidirected edges to characterize the change of marginal distribution of $\boldsymbol{s}, \boldsymbol{h}$ over

different environments. Subsequently, we compute the total loss function by combining the objective of policy optimization $\mathcal{L}_{\text{policy}}$ with the aforementioned loss functions:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{policy}} + \lambda_2 \mathcal{L}_{\text{VAE}} + \lambda_3 \mathcal{L}_{\text{guide}} + \lambda_4 \mathcal{L}_{\text{sparsity}} + \lambda_5 \mathcal{L}_{\text{MAG}}. \tag{3.8}$$

For policy optimization, we choose the classic PPO algorithm [28] as the backbone, which can be substituted with any other policy optimization algorithm. By computing and backpropagating the gradient of $\mathcal{L}_{\text{total}}$ to update the entire COREP framework, along with policy optimization in an end-to-end manner, we can prevent these distinct loss functions from resulting in abnormal causal structures. This ultimately ensures that the learned policy can effectively tackle non-stationarity. The overall framework is shown in Figure 3.1 and the detailed steps can be seen in Algorithm C.1.

4 Experiments

In this section, we primarily aim to address the following questions:

- Is COREP effective in addressing non-stationarity?
- What is the contribution of each component in COREP?
- Does COREP perform consistently under different degrees and settings of non-stationarity?

Baselines. To answer the above questions, we compare COREP with the following baselines: FN-VAE [10], VariBAD [29], and PPO [28]. FN-VAE is the SOTA method for tackling non-stationarity, VariBAD is one of the SOTA algorithms in meta-RL that also has certain capabilities in handling non-stationarity, and PPO is a classical algorithm known for its strong stability. Furthermore, to examine the performance degradation caused by non-stationarity, we include an Oracle that has full access to non-stationarity information.

Experimental settings. We conducted experiments on various modified environments from the DeepMind Control Suite [30]. The DeepMind Control Suite is a widely-used benchmark for RL algorithms, and we modify it to introduce non-stationarity, enabling a comprehensive evaluation of our method. Due to the page limitation, we only present the performance results for 8 of them (*Cartpole Swingup, Reacher Hard, Cup Catch, Cheetah Run, Swimmer Swimmer6, Finger Spin, Fish Upright, and Quadruped Walk*) in our main manuscript. Full results and details of environments are provided in Appendix B. In our experimental settings, similar to FN-VAE, we introduce periodic noise to the dynamics to represent non-stationarity. To support our claim that COREP can handle more complex non-stationarity, we design a more intricate setting, *i.e.*, we randomly sample the coefficients for both within-episode and across-episode non-stationarity at every time step. Specifically, our modification can be expressed as:

$$s' = f(s,a) + f(s,a) \cdot \alpha_d \left[c_1^t \cos(c_2^t \cdot t) + c_3^i \sin(c_4^i \cdot i) \right] \tag{4.1}$$

Here, α_d controls the overall degree of non-stationarity, and $c_k^t, c_k^i \sim \mathcal{N}(0.5, 0.5)$ represent the changing coefficients of within-episode and across-episode non-stationarity, respectively. This design generates various combinations of non-stationarity for each time step and episode, posing more significant challenges to our algorithm and the baselines.

Performance. As illustrated in Figure 4.1, the experimental results demonstrate that COREP outperforms all baselines in various environments, highlighting its consistent performance in the face of non-stationarity. Moreover, COREP exhibits significant performance improvements in more complex environments such as *Swimmer Swimmer6, Fish Upright, and Quadruped Walk*, accompanied by smaller variances, indicating its resilience against non-stationarity. Although VariBAD, as a meta RL method, can somewhat resist non-stationarity due to its ability of adaption, its large variances indicate insufficient stability. While the FN-VAE method, which directly models change factors, has shown performance similar to our COREP algorithm in certain simple environments, its performance in more complex environments proves that it cannot consistently handle complex non-stationarity. The narrower gap between COREP and the Oracle further suggests that COREP effectively reduces the performance degradation caused by non-stationarity.

Ablation study. We conduct ablation studies to analyze the contribution of each component in COREP. To ensure consistency in our conclusion, the experiments are conducted under various non-stationarity settings, which include 'within-episode & across-episode', 'within-episode', and

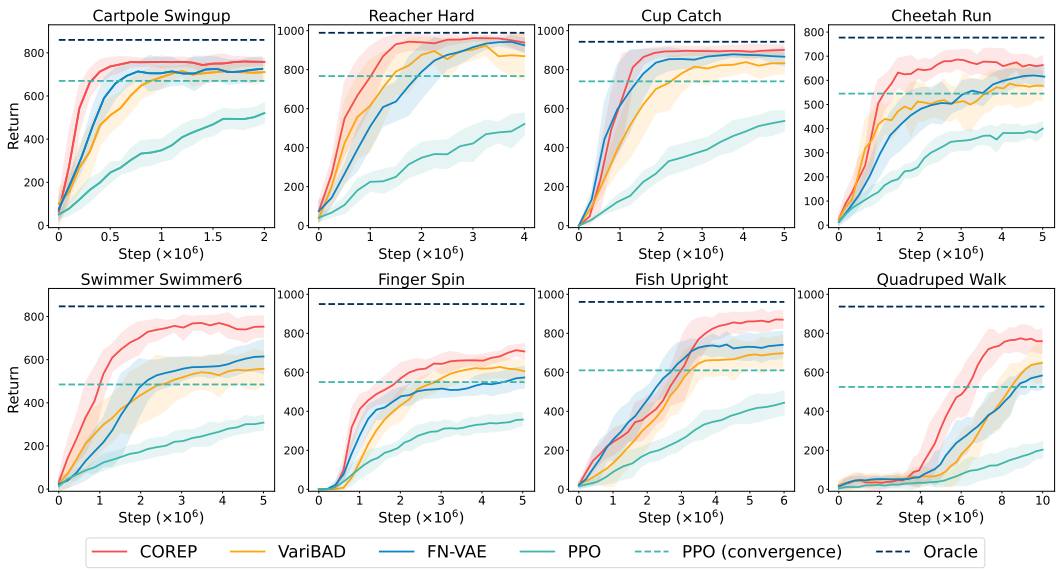
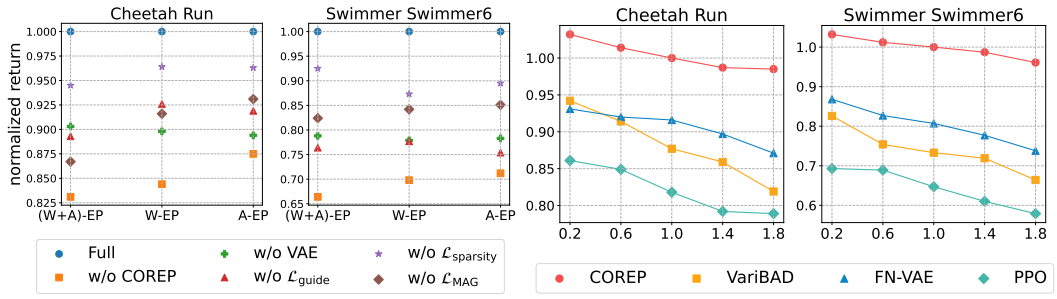


Figure 4.1: Learning curves of COREP and baselines in different environments. Solid curves indicate the mean of all trials with 5 different seeds. Shaded regions correspond to standard deviation among trials. The dashed reference lines are the asymptotic performance of PPO and Oracle.



(a) Ablation study.

(b) Non-stationarity degree.

Figure 4.2: Final mean returns of 3 different trials on Cheetah Run and Swimmer Swimmer6 environment with: (**a**) Different components and non-stationarity settings. The value of returns are normalized with the full version of COREP in each environment; (**b**) Different non-stationarity degrees. The values of returns are normalized with the COREP in each environment.

'across-episode' non-stationarities. These settings are respectively denoted as (W+A)-EP, W-EP, and A-EP. As depicted in Figure 4.2a, eliminating the overall COREP structure and the TD error detection mechanism (corresponding to $\mathcal{L}_{\text{guide}}$) from COREP both results in significant performance degradation, thereby substantiating the effectiveness of these two key designs in tackling non-stationarity. Moreover, the inclusion of VAE proves crucial as it improves the learning efficiency. As expected, the regularization terms $\mathcal{L}_{\text{MAG}}$ and $\mathcal{L}_{\text{sparse}}$ also play important roles in enhancing the framework's ability to address non-stationarity.

Different non-stationarity degrees. As demonstrated by the consistent performance across different types of non-stationarities in Figure 4.2a, we also analyze the impact of varying degrees of non-stationarity in the environment, as depicted in Figure 4.2b. The results suggest that the performance of the compared baselines is more affected by the degree of non-stationarity. Conversely, COREP exhibits consistent performance when encountering different degrees of non-stationarity, further affirming our claim that COREP can effective tackle more complex non-stationarities.

Due to the page limitation, only partial experimental results are presented in the main manuscript. For complete results, specific implementation details, and environmental settings, please refer to Appendix B and Appendix C.

5 Related Work

Nonstationary RL. Pioneering research in non-stationary reinforcement learning primarily focused on detecting changes that had already occurred [25, 26], rather than anticipating them. Various methods have been developed to anticipate changes in non-stationary deep reinforcement learning settings. For example, Chandak *et al*. [31] maximized future rewards without explicitly modeling non-stationary environments, while Alegre *et al*. [32] employed change-point detection to determine whether an agent should learn a new policy or reuse existing ones. However, change-point detection may not work well in complex non-stationary environments and often requires providing priors. In cases where the evolution of non-stationary environments can be represented as a Semi-Markov chain, Hidden Markov-MDPs or Hierarchical Semi-Markov Decision Processes can be employed to address non-stationarity [33, 34]. Some later work attempts to resist non-stationarity by leveraging the generalization of meta-learning [7]. For example, Al-Shedivat *et al*. [35] integrated continuous adaptation into the learning-to-learn framework to solve the non-stationarity. Poiani *et al*. [8] track non-stationarity by inferring the evolution of latent parameters, capturing the temporal change factors during the meta-testing phase. Nagabandi *et al*. [36] meta-trained dynamic priors, enabling efficient adaptation to local contexts. However, these methods require meta-training and pre-defining non-stationary tasks, but in real-world scenarios, we cannot obtain information about non-stationarity. An alternative line of research directly learns latent representations to capture non-stationary components, leveraging latent variable models to directly model change factors in environments or estimating latent vectors describing the non-stationary or variable aspects of dynamics. Xie *et al*. [11] regarded the change factor as a latent variable and explicitly modeled the latent MDP. Feng *et al*. [10] modeled multiple latent variables of non-stationarities to achieve better performance. However, in real-world scenarios, non-stationarity itself is often more complex. Simply modeling the latent dynamics may not solve such complex scenarios well. Interpreting non-stationarity from a causal perspective is another novel approach. Mirhoseini *et al*. [37] attempts to find an invariant causal structure to address non-stationarity, which shares some similar points with us. However, their method relies on offline data and has only been verified in simple Contextual Bandits environments. In contrast, our COREP algorithm can do online learning and tackle non-stationarity in complex environments.

Causal Structure Learning. Various approaches for learning causal structure from observed data have been proposed, see *e.g*. [38] for a review. These approaches mainly fall into two broad categories: constraint-based methods and score-based methods. The constraint-based methods check the existence of edges by performing conditional independence tests between each pair of variables, *e.g*. PC [39], IC [19], and FCI [40, 41]. In contrast, the score-based methods generally view causal structure learning as a combinatorial optimization problem, and measure the goodness of fit of graphs over the data with a score, then optimize such score to find an optimal graph or equivalent classes [42–46]. Recently, some gradient-based methods which transform the discrete search into a continuous optimization by relaxing the space over DAGs have been proposed. These methods allow for applying continuous optimizations such as gradient descent to causal structure learning. For example, Zheng *et al*. [47] reformulated the structure learning problem as a continuous optimization problem, and ensured acyclicity with a weighted adjacency matrix. Yu *et al*. [48] proposed a generative model parameterized by a GNN and apply a variant of the structural constraint to learn the DAG. Saeed *et al*. [49] considered the distribution arising from a mixture of causal DAGs, used MAGs to represent DAGs with unobserved nodes, and showed the identifiability of the union of component MAGs. In this work, we characterize the non-stationarity by the union of MAGs, which enlightens us to design the COREP algorithm modelled by GNN.

6 Conclusions, Limitations and Future Work

In this work, we present a causal interpretation for non-stationary RL and propose a novel algorithm that focuses on the causal relationships within states. This algorithm learns a causal-origin representation to tackle more complex non-stationarity problems. Grounded in our proposed formulation, we design a modular algorithm that can be seamlessly integrated with existing RL algorithms. The theoretical analysis provides both inspiration and theoretical support for our algorithm. Experimental results from various non-stationary environments further demonstrate the efficacy of our algorithm.

However, our method does have certain limitations. The current approach may encounter scalability issues in high-dimensional state spaces, as the graph-based representation could be computationally intensive. In future work, we aim to address this issue by incorporating causal-origin representation learning into other types of latent variable models, such as normalizing flows and probabilistic graphical models. This could further enhance scalability and performance of our method.

References

[1] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

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

[3] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[4] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2020.

[5] Sindhu Padakandla, Prabuchandran KJ, and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50:3590–3606, 2020.

[6] Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, 54(6):1–25, 2021.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[8] Riccardo Poiani, Andrea Tirinzoni, and Marcello Restelli. Meta-reinforcement learning by tracking task non-stationarity. *arXiv preprint arXiv:2105.08834*, 2021.

[9] Shagun Sodhani, Franziska Meier, Joelle Pineau, and Amy Zhang. Block contextual mdps for continual learning. In *Learning for Dynamics and Control Conference*, pages 608–623. PMLR, 2022.

[10] Fan Feng, Biwei Huang, Kun Zhang, and Sara Magliacane. Factored adaptation for non-stationary reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[11] Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst lifelong non-stationarity. *arXiv preprint arXiv:2006.10701*, 2020.

[12] Kun Zhang, Mingming Gong, Petar Stojanov, Biwei Huang, Qingsong Liu, and Clark Glymour. Domain adaptation as a problem of inference on graphical models. *Advances in neural information processing systems*, 33:4965–4976, 2020.

[13] Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *The Journal of Machine Learning Research*, 21(1):3482–3534, 2020.

[14] Eric V Strobl. Improved causal discovery from longitudinal data using a mixture of dags. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*, pages 100–133. PMLR, 2019.

[15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[16] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[17] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: Cambridge University Press*, 19(2), 2000.

[20] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[21] Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

[22] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[23] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. AdaRL: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations*, 2022.

[24] Kayvan Sadeghi. Stable mixed graphs. *Bernoulli*, 19(5B):2330–2358, 2013.

[25] Bruno C Da Silva, Eduardo W Basso, Ana LC Bazzan, and Paulo M Engel. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd international conference on Machine learning*, pages 217–224, 2006.

[26] Richard S Sutton, Anna Koop, and David Silver. On the role of tracking in stationary environments. In *Proceedings of the 24th international conference on Machine learning*, pages 871–878, 2007.

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

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

[29] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

[30] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[31] Yash Chandak, Georgios Theocharous, Shiv Shankar, Martha White, Sridhar Mahadevan, and Philip Thomas. Optimizing for the future in non-stationary mdps. In *International Conference on Machine Learning*, pages 1414–1425. PMLR, 2020.

[32] Lucas N Alegre, Ana LC Bazzan, and Bruno C da Silva. Minimum-delay adaptation in non-stationary reinforcement learning via online high-confidence change-point detection. *arXiv preprint arXiv:2105.09452*, 2021.

[33] Samuel Choi, Dit-Yan Yeung, and Nevin Zhang. An environment model for nonstationary reinforcement learning. *Advances in neural information processing systems*, 12, 1999.

[34] Emmanuel Hadoux, Aurélie Beynier, and Paul Weng. Solving hidden-semi-markov-mode markov decision problems. In *Scalable Uncertainty Management: 8th International Conference, SUM 2014, Oxford, UK, September 15-17, 2014. Proceedings 8*, pages 176–189. Springer, 2014.

[35] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017.

[36] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

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

[38] Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022.

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

[40] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 499–506, 1995.

[41] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

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

[43] Mikko Koivisto and Kismat Sood. Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

[44] Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 445–452, 2006.

[45] James Cussens, David Haws, and Milan Studenỳ. Polyhedral aspects of score equivalence in bayesian network structure learning. *Mathematical Programming*, 164:285–324, 2017.

[46] Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized score functions for causal discovery. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1551–1560, 2018.

[47] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

[48] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

[49] Basil Saeed, Snigdha Panigrahi, and Caroline Uhler. Causal structure discovery from distributions arising from mixtures of dags. In *International Conference on Machine Learning*, pages 8336–8345. PMLR, 2020.

A Causality Background and Proofs

We first review the definition of the Markov condition, the faithfulness assumption and some graphical concepts shown in the condition of Theorem 3.1. We use $\mathrm{pa}_{\mathcal{D}}(v)$, $\mathrm{ch}_{\mathcal{D}}(v)$, and $\mathrm{an}_{\mathcal{D}}(v)$ to denote the parents, children and ancestor of node v, respectively; for the detailed definitions, see *e.g.* Lauritzen [20].

Definition A.1 (Global Markov Condition [19])**.** *A distribution P over V satisfies the global Markov condition on graph $\mathcal{D}_{(k)}$ if for any partition (X, Y, Z) such that X is d-separated from Y given Z,* i.e. *$X \perp\!\!\!\perp Y \mid Z$ (X and Y are conditionally independent given Z).*

Definition A.2 (Faithfulness [19])**.** *There are no independencies between variables that are not entailed by the Markov Condition.*

Under the above assumptions, we can tell the conditional independences using the d-separation criterion from a given DAG $\mathcal{D}$ [19]. Similarly, the MAGs are ancestral graphs where any non-adjacent pair of nodes is d-separated [21]. The following algorithm shows how to construct a MAG from DAG [49]:

Algorithm A.1 Construction of the maximal ancestral graph

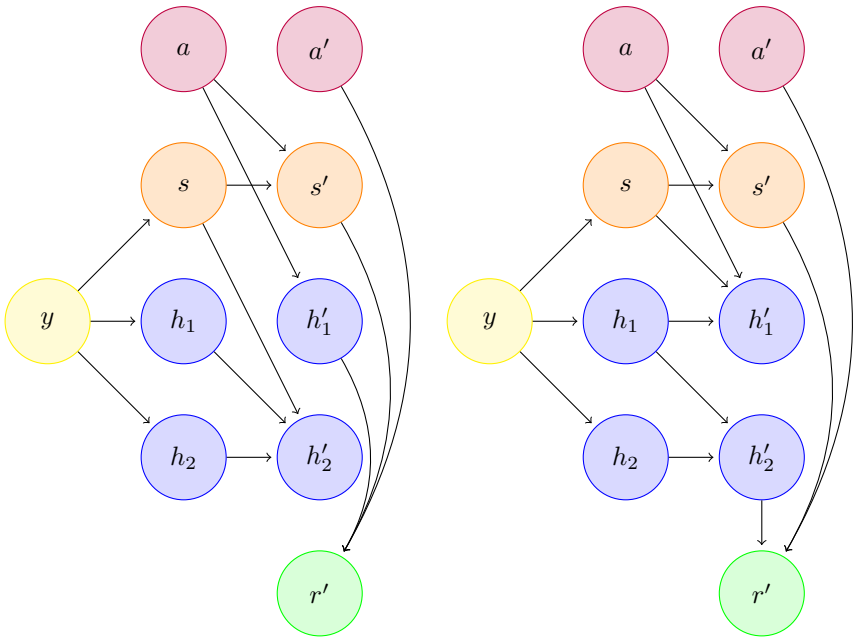
Input: DAG $\boldsymbol{G} = (V, E)$
Initialize $D = \emptyset, B = \emptyset$
for $u, v \in \mathrm{ch}_{\mathcal{D}}(y)$ **do**
    add $u \leftrightarrow v$ to B.
end for
for t, u, v such that $(t \to u) \in E$ and $(u \leftrightarrow v) \in B$ **do**
    if $u \in \mathrm{an}_{\mathcal{D}}(v)$ **then**
        add $t \to v$ to D
    end if
end for
for u, v such that $(u \leftrightarrow v) \in B$ **do**
    if $u \in \mathrm{an}_{\mathcal{D}}(v)$ **then**
        remove $u \leftrightarrow v$ from B and add $u \to v$ to D
    end if
end for

To illustrate the above algorithm, we provide two figures. Figure A.1 shows the underlying causal DAGs for the two environments, and Figure A.2 depicts the output of Algorithm A.1 as well as the corresponding environment-shared union graph.

Proof of Theorem 3.1. The outline of the proof are as follows. We first construct a strict partial order π on V. Then, we induce the MAGs $\mathcal{M}_{(1)}, \ldots, \mathcal{M}_{(k)}$ from the DAGs $\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(k)}$ by applying the rules defined in Algorithm A.1. We show the constructed partial order π is *compatible*, that for all $1 \leq k \leq K$, it holds that (a) $u \in \mathrm{an}(v) \Rightarrow u <_\pi v$ in $\mathcal{M}^{(k)}$; and (b) $u \leftrightarrow v \Rightarrow u \not\lesseqgtr_\pi v$ in $\mathcal{M}^{(k)}$. Finally we leverage the existing results in [49] to conclude that $\mathcal{M}_\cup$ is a MAG.

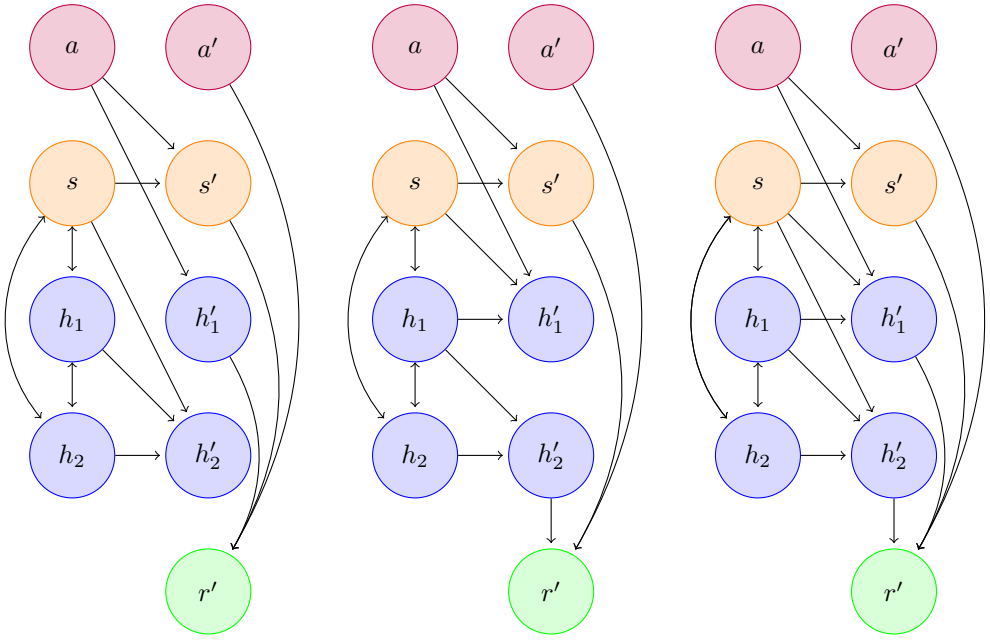
We define a relation π on V as following: for any variable $u \in \{\boldsymbol{s}, \boldsymbol{h}, \boldsymbol{a}\}$ and any variable $v \in \{\boldsymbol{s}', \boldsymbol{h}', \boldsymbol{a}'\}$, we have (i) $u <_\pi v$; (ii) $v <_\pi r'$. To show the above defined π is a strict partial order, we first notice that π is irreflexive, because $u \not<_\pi u, v \not<_\pi v$ and $r' \not<_\pi r'$. The transitivity and asymmetry also hold by definition of π. Therefore, π is a partial order on V.

The Algorithm A.1 constructs an MAG from DAG with three steps. The first step is to add bidirected edges among the nodes in $\mathrm{ch}(e)$. Different values of e leads different marginal distribution of $\boldsymbol{s}, \boldsymbol{h}$, hence $\mathrm{ch}(y) \subseteq \{\boldsymbol{s}, \boldsymbol{h}\}$. Therefore, the bidirected edges are added with both nodes belonging to $\{\boldsymbol{s}, \boldsymbol{h}\}$. For the second step, there is no such node t, with $(t \to u) \in E$ and $(u \leftrightarrow v) \in B$, because the nodes in $\{\boldsymbol{s}, \boldsymbol{h}\}$ have no ancestor other than itself. So the second step adds the directed edges when $u = v$. The third step in our case is redundant. Equation (3.1) shows that there is no instantaneous causal effects in the system, so there is no u, v such that $(u \leftrightarrow v) \in B$ while $u \in \mathrm{an}_{\mathcal{D}}(v)$. From all above, if the input of Algorithm A.1 is $\mathcal{D}_{(k)}$, then it outputs a MAG $\mathcal{M}_{(k)} = (V, D_{(k)}, B_{(k)})$ with the set of nodes V, the set directed edges D equals to the set of directed edges $E_{(k)}$ after removing the node y, and the set of bidirected edges $B_{(k)}$ consists edges among nodes in $\{\boldsymbol{s}, \boldsymbol{h}\}$.



(a) DAG $\mathcal{D}_{(1)}$. (b) DAG $\mathcal{D}_{(2)}$.

Figure A.1: DAG representations for two different environments.



(a) MAG $\mathcal{M}_{(1)}$. (b) MAG $\mathcal{M}_{(2)}$. (c) Union graph $\mathcal{M}_{\cup}$

Figure A.2: MAG representations for two different environments and the union graph of two environments.

Then, we check the condition (a) and (b) to show $\mathcal{M}_{(1)}, \ldots, \mathcal{M}_{(k)}$ are compatible with the above defined π. For (a), if u is the ancestor node of v, then the structure of $\mathcal{M}_{(k)}$ implies that either $u \in \{\boldsymbol{s}, \boldsymbol{h}\}$ and $v \in \{\boldsymbol{s}', \boldsymbol{h}'\}$ are nodes in $\{\boldsymbol{s}', \boldsymbol{h}'\}$, or u is a node from $\{\boldsymbol{s}', \boldsymbol{h}'\}$ and $v = r'$. For (b), if $u \leftrightarrow v$, then u, v are nodes in $\{\boldsymbol{s}, \boldsymbol{h}\}$, hence $u \not\preccurlyeq_\pi v$ in $\mathcal{M}^{(k)}$. These means that π is a common strict partial order on V for all MAGs. In this setup, we can leverage existing results from Lemma 4.3 [49] to show that the environment-shared union graph $\mathcal{M}_\cup$ is also a maximal ancestral graph.

□

B Full Experiment Details

C Implementation and Training Details

C.1 Pseudo code for COREP

Algorithm C.1 **C**ausal-**O**rigin **REP**resentation (**COREP**)

Init:
for $i = 0, 1, \ldots$ **do**
  Collect trajectory τ_i with $\pi(\boldsymbol{a}|\boldsymbol{h})$
  Update replay buffer $\mathcal{B}[i] \leftarrow \tau_i$
  for $j = 0, 1, \ldots, N$ **do**
    Sample a batch of episodes E_j from $\mathcal{B}$ and TD errors $\{\delta_k\}$ from $\mathcal{B}_\delta$.
    Encode states into node embedding g.
    Compute $\boldsymbol{A_g} = \text{Gumbel-Softmax}\left(\frac{\boldsymbol{g}\cdot\boldsymbol{g}^{\mathrm{T}}}{|\boldsymbol{g}|\cdot|\boldsymbol{g}^{\mathrm{T}}|} \odot (\mathbf{1}_N - \boldsymbol{I}_N)\right)$.
    Compute $\delta_\alpha = \frac{\sum_{|\mathcal{B}_\delta| - \alpha|\mathcal{B}_\delta| < k < |\mathcal{B}_\delta|} \delta_k}{\alpha|\mathcal{B}_\delta|}$, $\bar{\delta} = \frac{\sum_{\mathcal{B}_\delta} \delta_k}{|\mathcal{B}_\delta|}$
    if $\delta_\alpha \notin (\bar{\delta} - \eta, \bar{\delta} + \eta)$ **then** ▷ Confidence detection of TD error.
      unfreeze weights of core-GAT.
    else
      freeze weights of core-GAT.
    end if
    Get graph embedding $g_{\text{core}}, g_{\text{general}}$ from core-GAT and general-GAT.
    Compute $\mathcal{L}_{\text{MAG}}, \mathcal{L}_{\text{stable}}, \mathcal{L}_{\text{sparse}}$
    Infer μ_g, σ_g^2 of $g = g_{\text{core}} \oplus g_{\text{general}}$
    Sample $\boldsymbol{h} \sim \mathcal{N}\left(\mu_g, \sigma_g^2\right)$
    Decode $\hat{s}$ from $\boldsymbol{h}$, then compute $\mathcal{L}_{\text{VAE}}$ and TD error $\boldsymbol{\delta}$
    Push δ into TD buffer $\mathcal{B}_\delta$
    $\theta_Q \leftarrow \theta_Q - \alpha_Q \nabla_{\theta_Q} \mathcal{J}_Q$ ▷ Update critic
    $\theta_\pi \leftarrow \theta_\pi - \alpha_\pi \nabla_{\theta_\pi} \mathcal{J}_\pi$ ▷ Update actor
    $\phi \leftarrow \phi - \alpha_\phi \nabla_\phi \left(\mathcal{J}_Q + \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{MAG}} + \mathcal{L}_{\text{stable}} + \mathcal{L}_{\text{sparse}}\right)$ ▷ Update **COREP**
  end for
end for

D Compute Resource Details

E Licenses