Efficient Hierarchical Reinforcement Learning with Targeted Causal Interventions

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Hierarchical reinforcement learning (HRL) is a promising approach for enhancing the efficiency of long-horizon reinforcement-learning tasks with sparse rewards. It breaks down the learning task into a hierarchy of subgoals. One of the main challenges in HRL is discovering the hierarchical structure among subgoals and effectively utilizing this knowledge to achieve the final subgoal. In this study, we address this challenge by modeling the subgoal structure as a causal graph. We characterize, in particular, the extent to which we can recover the underlying causal graph by defining a notion of discoverable parents. Furthermore, we propose a causal discovery algorithm that identifies discoverable parents by processing the time series of subgoal activations. We introduce three heuristics that use the knowledge derived from the recovered causal graph. These heuristics prioritize the subgoals and select those that are influential in achieving the final subgoal rather than selecting a mere random selection. Thus, the policy is guided toward achieving the final subgoal, efficiently in terms of training cost. Unlike prior work that does not provide a theoretical analysis of their methodologies, our work provides a formal analysis of the problem as it enables theoretical comparisons. In tree structures and Erdős-Réyni random graphs, the proposed heuristics provide improvements, respectively, $O(n^2/\log^2(n))$ and $O(n^{\frac{2}{3}-\frac{2}{3}c}/\log(n))$ as compared with random selection in terms of the training cost, where n is the number of subgoals. Experimental results across various HRL tasks illustrate that our proposed methods outperform existing work in terms of training costs.

1 Related Work

The concept of learning multiple levels of policies has been a topic of interest for many years [18, 8, 15]. [18] introduced an option framework, which offers an abstraction over primitive actions. This framework allows an agent to choose between executing a primitive action or an option, which is a higher-level decision-making process that continues for multiple time steps. Further studies have been made on this framework to define the reward function or generalize the value function [12, 17]. Similarly, [5] proposed a framework that takes decisions over two levels of hierarchy: a higher-level policy that picks a goal, and a lower-level policy that selects primitive actions regarding that goal. [13] proposed an HRL approach to learn multiple levels of policies. Building on this, [6], proposed HAC, a framework that accelerates learning by enabling hierarchical agents to jointly learn a hierarchy of policies. Based on HAC, a modified goal-conditioned HRL method is proposed in [7] to discover subgoals from slowly changing features.

Causal discovery in RL has been the focus of some research [11, 9]. For instance, [14] proposed a measure of situation-dependent causal influence to improve the efficiency of reinforcement learning. However, most of these approaches have prior assumptions or information about the causal graph.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Additionally, [20] learns the causal world model without prior knowledge for better explainability. [21] shows theoretically that a causal world-model outperforms a plain world-model in offline RL. [19] learns a theoretically proved causal dynamics model that removes unnecessary dependencies between state variables and the action. Establishing a hierarchy of goals is crucial in learning environments, and some studies impose restrictions on the structure's form to uncover this hierarchical structure [3, 2]. [4] proposed an approach that automatically discovers the causality graph in the environment to guide the exploration of hierarchical structure.

2 Preliminaries

In this section, we provide the fundamental concepts and notations that form the basis of our study on hierarchical reinforcement learning (HRL). One way to model HRL is through subgoal-based MDP. The subgoal-based MDP is formalized as a tuple $(\mathcal{S}, \mathcal{G}, \mathcal{A}, T, R, \gamma)$, where

- $\mathcal{S}$ is the set of all possible states,
- $\mathcal{G}$ is the subgoal space, which contains intermediary objectives that guide the policy learning,
- $\mathcal{A}$ is the set of actions available to the agent,
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function that denotes the probability of transitioning from state s to state s' under action a,
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$ is the reward function, that indicates if the agent achieves a subgoal g, after transitioning from state s to state s' after taking action a, specifically:

$$R(s, a, g) = \begin{cases} 1, & \text{if the subgoal } g \text{ is achieved in transitioning from state } s \text{ to state } s', \\ 0, & \text{otherwise,} \end{cases}$$

- γ is the discount factor that quantifies the diminishing value of future rewards.

We call the number of observed state-action pairs as the system probes. For a given time horizon H, the agent observes a sequence of state-action pairs $\tau = (s_0, a_0, \cdots, s_{H-1}, a_{H-1})$, called a trajectory. Given a state $s \in \mathcal{S}$ and a subgoal $g \in \mathcal{G}$, the objective in a subgoal-based MDP is to learn a subgoal-based policy $\pi(a|s, g) : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$ that maximizes value function defined as

$$V^{\pi}(s, g) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, g) \mid s_0 = s\right]. \tag{1}$$

In the subgoal-based MDP, we also need to model how the agent perceives the environment and interacts with it. In order to model the perception process, we assume that the agent has access to certain environment variables (EVs) that are disentangled factors of the environment observations. We denote the set of EVs by $\mathcal{E} = \{E_1, \cdots, E_m\}$, where m is the total number of EVs (see the example 7.2). We denote the vector of these EVs at time step t by $\mathbf{E}^t = (E_1^t, \cdots, E_m^t)$ as the state of the system. We focus on a subset $\mathcal{E}_s \subseteq \mathcal{E}$, where each variable $E_i \in \mathcal{E}_s$ takes a value of either zero or one. Without loss of generality, we assume that the first n EVs are binary, hence $\mathcal{E}_s = \{E_1, \cdots, E_n\}$, where $n \leq m$. We denote the vector of $\mathcal{E}_s$ at time step t by $\mathbf{E}_s^t = (E_1^t, \cdots, E_n^t)$. Given this setup, we define a set of subgoals $\mathcal{X} = \{X_1, \cdots, X_n\}$, where each subgoal being associated with a binary EV from $\mathcal{E}_s$. At time step t, the subgoal X_i associated with E_i, is said to be achieved ($X_i^t = 1$) if and only if $E_i^t = 1$. In our setting, the subgoal space $\mathcal{G}$ is $\mathcal{X}$, with X_n representing the only desired subgoal. We will refer to X_n as the "final subgoal" in the remainder of this paper. The action variable at time step t is denoted by $A^t \in \mathcal{A}$. We assume that the process $\{A^t, \mathbf{E}^t\}_{t \in \mathbb{Z}^+}$ can be described by a structural causal model (SCM) (see the definition of SCM 7.1) in the following form:

$$\begin{aligned} E_i^{t+1} &= f_i(\text{pa}(E_i^{t+1}), A^t, \epsilon_i^{t+1}), \qquad \forall t \geq 1 \text{ and } 1 \leq i \leq m, \\ A^{t+1} &= f_0(\mathbf{E}^{t+1}, \epsilon_0^{t+1}), \qquad \forall t \geq 0, \end{aligned}$$

where $\text{pa}(E_i^{t+1})$ represents the parent environment variables of E_i^{t+1}, and f_i is the causal mechanism showing how $\text{pa}(E_i^{t+1})$ influences E_i at time $t + 1$. Furthermore, ϵ_i^{t+1} and ϵ_0^{t+1} represent the corresponding exogenous noise, of E_i and A at time $t + 1$, respectively.

The summary graph $\mathscr{G}$ is used to graphically represent the causal relationships in the SCM where there is a node for each variable in $\mathcal{E} \cup \{A\}$. Furthermore, an edge from E_j to E_i is drawn in $\mathscr{G}$ if $E_j^t \in \text{pa}(E_i^{t+1})$ for any t. Additionally, there exist directed edges from the action variable A to each environment variable E_i and from E_i back to A.

Definition 2.1 (Subgoal Structure)**.** The subgoal structure, denoted as $\mathscr{S}$, is a directed graph where the nodes represent subgoals in the set $\mathcal{X}$. In $\mathscr{S}$, a directed edge from subgoal X_j to subgoal X_i exists if and only if there is at least one path in the summary graph $\mathscr{G}$ from E_j to E_i such that all intermediate nodes along this path belong to the set $(\mathcal{E} \setminus \mathcal{E}_s)$. These edges in $\mathscr{S}$ represent a one-step causal relationship from one subgoal to another by ignoring intermediate environmental variables not designated as subgoals (see the example 7.2).

In this paper, the terms 'subgoal' and 'node' are used interchangeably. In the subgoal structure $\mathscr{S}$, the parent and children sets of a subgoal X_i are denoted by PA_{X_i} and CH_{X_i}, respectively. We denote the estimate of the subgoal structure $\mathscr{S}$ by $C_{\mathscr{S}}$. Additionally, the parent set of a subgoal X_i in the graph $C_{\mathscr{S}}$ is represented as $PA_{X_i}^{C_{\mathscr{S}}}$.

In our considered setup, the subgoals are further categorized into two types, based on their corresponding causal mechanisms:

- **AND subgoal:** X_i is an "AND" subgoal if it can be achieved (or become 1) at time step t if and only if all its parents in the set PA_{X_i} have been achieved prior to time step t, indicating a strict conjunctive requirement for the achievement of X_i.
- **OR subgoal:** X_i is an "OR" subgoal if it can be achieved (or become 1) at time step t if at least one of its parents in the set PA_{X_i} has been achieved before time step t. This indicates a disjunctive requirement for the activation of X_i, where the achievement of any single parent is sufficient to achieve X_i.

In the rest of the paper, we represent an AND subgoal with a square and an OR subgoal with a circle in the figures.

Definition 2.2 (Hierarchical Structure)**.** The hierarchical structure is denoted by $\mathcal{H} = (\mathscr{H}, \mathcal{L})$, where $\mathscr{H}$ is a directed acyclic graph (DAG) which is a subgraph of the subgoal structure $\mathscr{S}$, and $\mathcal{L} : \mathcal{X} \to \mathbb{N}$ is a level assignment function, where $\mathcal{L}(X_i)$ denotes the hierarchy level of subgoal X_i. In $\mathcal{H}$, each node is assigned to a hierarchy level such that: i) For an AND subgoal, all of its parents in the summary graph $\mathscr{S}$ are positioned at higher levels (highest level has level number 0). ii) For an OR subgoal, at least one of its parents in $\mathscr{S}$ must be at higher levels (see the example 7.2). In other words, the hierarchical structure $\mathcal{H}$ should satisfy the following properties:

- For an AND subgoal X_i, $\mathcal{L}(X_i) \geq \max_{X_j \in PA_{X_i}} \mathcal{L}(X_j) + 1$.
- For an OR subgoal X_i, $\mathcal{L}(X_i) \geq \min_{X_j \in PA_{X_i}} \mathcal{L}(X_j) + 1$.

Definition 2.3 (Intervention, Interventional Data)**.** An intervention on a subgoal X_i at time t^*, denoted by $\text{do}(X_i^{t^*} = a)$, is considered as replacing the structural assignment of the associated environment variable E_i with $E_i^t = a$ for all $t \geq t^*$. The interventional data of the subgoal X_i is denoted by $D_{\text{do}(X_i^{t^*}=a)}$ and consists of the state-action pairs that the agent collects (the actions are taken randomly) until the time step $t^* + \Delta$, where Δ is a positive integer and $t^* + \Delta < H$.

For a set or list A and a number a, we define the notation $A = a$ showing that every element of A is equal to a.

Definition 2.4. Let A and B be subsets of subgoals, where $X_n \notin (A \cup B)$. The expected causal effect (ECE) of A on X_n at some time step $t^* + \Delta$ (where Δ is a positive integer) condition that B is not achieved at time t^* is defined as:

$$ECE_{t^*}^{\Delta}(A, B, X_n) = \mathbb{E}[X_n^{t^*+\Delta} \mid \text{do}(X_A^{t^*} = 1), X_B^{t^*} = 0] - \mathbb{E}[X_n^{t^*+\Delta} \mid \text{do}(X_A^{t^*} = 0), X_B^{t^*} = 0],$$

where:

- The first term, $\mathbb{E}[X_n^{t^*+\Delta} \mid \text{do}(X_A^{t^*} = 1), X_B^{t^*} = 0]$, shows the expected value of X_n at time $t^* + \Delta$, where subgoals in subset A are forced to be 1 and those in subset B are not achieved at time t^*.

- The second term, $\mathbb{E}[X_n^{t^*+\Delta} \mid \mathrm{do}(X_A^{t^*} = 0), X_B^{t^*} = 0]$, represents the expected value of X_n at time $t^* + \Delta$, where subgoals in subset A are forced to be 0 and those in subset B are not achieved at time t^*.

This expression measures the causal effect of subgoals in A in achieving the final subgoal X_n, whereas the sub-goals in B are not achieved at time step t^*.

3 Hierarchical Reinforcement Learning via Causality (HRC)

In this section, we introduce Hierarchical Reinforcement Learning via Causality (HRC) framework which incorporates hierarchical structure (see Definition 2.2) to guide the policy training.

3.1 Algorithm

In the following, we first define "controllable sub-goal" which is used in the description of the proposed framework.

Definition 3.1. For a given $\epsilon > 0$, a subgoal X_i is called $(1-\epsilon)$**-controllable**, if there exists a time step $t^* < H$ such that X_i can be achieved at t^*, with probability at least $1-\epsilon$, given that actions are taken based on the subgoal-based policy $\pi(a \mid s, X_i)$. Mathematically speaking, subgoal X_i is $(1-\epsilon)$-controllable if:

$$\exists t^* \in \mathbb{N} \text{ such that } \mathbb{P}(X_i^{t^*} = 1 \mid \pi(a \mid s, X_i)) \geq 1 - \epsilon,$$

where π denotes the policy guiding the actions to achieve subgoals.

Assumption 3.2. In our setting, we assume that if a subgoal X_i is achieved at a time step t^* (i.e., $X_i^{t^*} = 1$), it will remain achieved in all future time steps. Formally, if $X_i^{t^*} = 1$, then $X_i^t = 1$ for all $t \geq t^*$.

Remark 3.3. Suppose a subgoal X_i is $(1-\epsilon)$-controllable. In this case, based on Definition 3.1 and Assumption 3.2, there exists a time step $t^* < H$ such that with probability at least $1-\epsilon$, we have: $X_i^t = 1$ for all $t \geq t^*$. Thus, according to Definition 2.3, it is equivalent to have the intervention $\mathrm{do}(X_i^{t^*} = 1)$ with probability at least $1-\epsilon$. This intervention can be performed by taking actions based on the subgoal-based policy $\pi(a \mid s, X_i)$.

Remark 3.4. For the rest of the paper, to simplify the presentation, when we say a subgoal is controllable, we mean it is $(1-\epsilon)$-controllable for a predetermined ϵ.

Algorithm 1 Hierarchical Reinforcement Learning via Causality (HRC)

```
Initialize SCM's parameters $\phi$; subgoal-based policy $\pi$; the intervention set $SI_0 = \{\}$; controllable set $SC_0 = \{\}$; hierarchical structure $\mathcal{H} = (\mathscr{H}, \mathcal{L})$; iteration counter $t = 1$.
repeat
    $X_{\text{sel, t}} \leftarrow$ Choose a controllable subgoal from $SC_{t-1}$
    $SI_t \leftarrow SI_{t-1} \cup X_{\text{sel, t}}$, $SC_t \leftarrow SC_{t-1} \setminus \{X_{\text{sel, t}}\}$
    $D_I \leftarrow$ InterventionSampling($\pi, SI_t$)
    $C_{\mathscr{S}} \leftarrow$ CausalDiscovery($\phi, D_I, SI_t$)
    $SCC_t \leftarrow \{X_i \mid X_i \notin (SI_t \cup SC_t)$ and $PA_{X_i}^{C_{\mathscr{S}}} \subset SI_t\}$
    Update-$\mathcal{H}$( $C_{\mathscr{S}}, SI_t, SCC_t$)
    SubgoalTraining($\pi, \mathcal{H}, SI_t, SCC_t$)
    $SC_t \leftarrow SC_t \cup SCC_t$
    $t \leftarrow t + 1$
until $X_n \in SI_t$ or $SC_t$ is empty
```

The pseudocode of HRC framework is given in Algorithm 1. In particular, HRC has the following key steps:

1. **Initialization:** Initializing the SCM's parameters, intervention set (SI_0), controllable set (SC_0), and an empty hierarchical structure $\mathcal{H} = (\mathscr{H}, \mathcal{L})$, where the graph $\mathscr{H}$ contains only subgoals as nodes, with no edges between them, and setting all entries in $\mathcal{L}$ to `NAN`.

2. **Intervention Sampling:** The InterventionSampling subroutine collects interventional data D_I for the subgoals listed in SI_t. It collects T trajectories. In each trajectory, it randomly intervenes on the subgoals in SI_t until the horizon H is reached or all the subgoals in SI are achieved (see Appendix 12 for more details).
3. **Causal Discovery:** After collecting interventional data (D_I), the algorithm estimates the subgoal structure ($C_{\mathscr{S}}$) using a causal discovery method (see 5 for the proposed causal discovery algorithm). From $C_{\mathscr{S}}$, a set of subgoals, which are the children of SI_t (not necessarily all the children) but do not exist in SI_t or SC_t, is determined. We denote this set by SCC_t. The subgoals in SCC_t are called reachable subgoals.
4. **Update-$\mathcal{H}$:** We update the graph $\mathscr{H}$ such that for each subgoal $X_i \in SCC_t$, we add an edge from each parent $X_j \in PA_{X_i}^{C_{\mathscr{S}}}$ to X_i. Additionally, the level of X_i is updated as:

   $$\mathcal{L}(X_i) = \begin{cases} 0, & \text{if } |PA_{X_i}^{C_{\mathscr{S}}}| = 0, \\ \max_{X_j \in PA_{X_i}^{C_{\mathscr{S}}}} \mathcal{L}(X_j) + 1, & \text{otherwise.} \end{cases}$$

   This satisfies the conditions in Definition 2.2, as all necessary parents of X_i must be in SI_t for X_i to be reachable. Specifically, for an AND subgoal, all its parents must already be in SI_t (thus having lower level numbers), and for an OR subgoal, at least one parent must be in SI_t (thus having a lower level number).
5. **Subgoal Training:** The SubgoalTraining subroutine trains the subgoal-based policy for achieving the subgoals in SCC_t [1] (see Appendix 13 for more details).
6. **Update Controllable Set:** Adding the trained subgoals to the controllable set SC_t.

We refer to steps 2 and 3 as **Stage 1**, and steps 4, 5, and 6 as **Stage 2**.

HRC can use a random strategy for selecting a controllable subgoal from SC_t in line 3 of Algorithm 1. We call this “random strategy" and will refer to it as the baseline and denote it as HRC$_\text{b}$. If we apply the causal discovery method described in [1] in line 6 of HRC$_\text{b}$, it becomes similar to CDHRL [4]. In the sections 3.2 and 3.3, we assume the CasualDiscovery subroutine has no error in finding new edges.

3.2 Illustrative Example

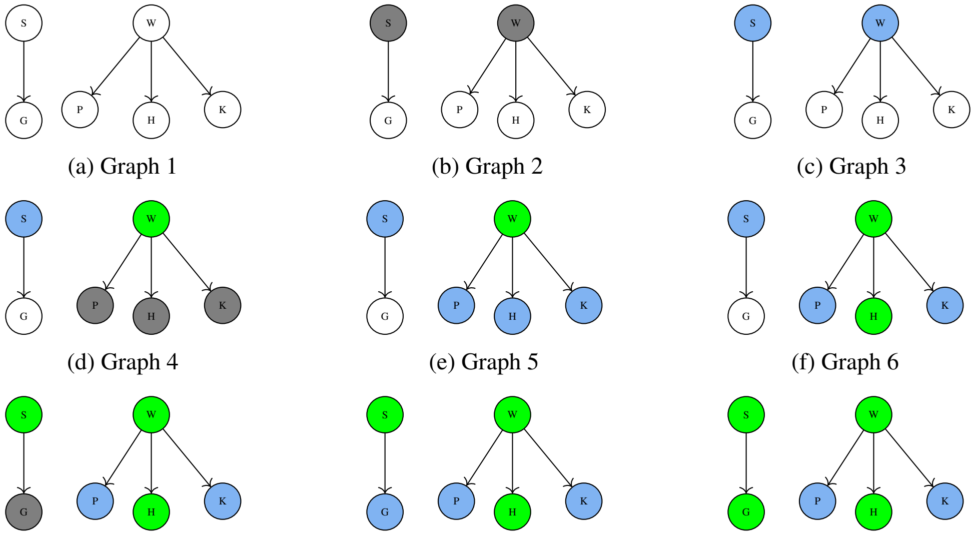
In Figure 1, we depict the stages of the HRC$_\text{b}$ and demonstrate how the algorithm achieves the final subgoal. In this example, we assume that every subgoal is OR type. The final subgoal is node G. We show reachable subgoals with gray color. When a reachable subgoal becomes controllable (or added to set SC), we show it with blue color. If a controllable subgoal is added to the set SI, it becomes green. Graph 1 represents the initial state where the algorithm knows nothing about the subgoal structure. Graph 2 shows the end of stage 1 of the iteration $t = 1$ where InterventionSampling is done and CausalDiscovery subroutine identifies S and W as reachable subgoals. Hence, subgoals S and W become gray at the end of stage 1 (Graph 2) and become blue at the end of stage 2 (Graph 3). Graph 4 shows stage 1 of the iteration $t = 2$ where W is selected randomly as $X_{\text{sel, }2}$. This process continues until the final subgoal G is added to SI (becomes green), and the algorithm terminates at $t = 5$ (Graph 9).

In this example, we could have terminated the algorithm at $t = 3$ if subgoal S had been selected as $X_{\text{sel, t}}$, rather than W, in Graph 4. This shows that a more strategic selection of controllable variables from SC in line 3 of Algorithm 1 can significantly improve the performance.

In the following, we propose new approaches that improves HRC mainly in lines 3 and 6 of Algorithm 1. We introduce: 1- A more strategic selection mechanism for $X_{\text{sel, t}}$, which is detailed in the subsequent section. 2- A new causal discovery algorithm (for line 6), to efficiently identify the causal relationships between subgoals, with consistency guarantee. Similar work, CDHRL [4], does not prioritize the selection of controllable subgoals for addition to the intervention set. It employs a causal discovery algorithm from [1] without considering the identifiability or scalability of the subgoal structure.

[1] If a subgoal becomes reachable, it should become controllable after Subgoal Training step. However, in practice, this may not always be the case. Therefore, we consider a threshold and remove the ones that are not trained well from SCC_t (see Appendix 13.2 for more details).

To evaluate the effectiveness of any strategy used in line 3 of Algorithm 1, it is essential to mathematically formulate the cost of Algorithm 1.



(a) Graph 1 (b) Graph 2 (c) Graph 3

(d) Graph 4 (e) Graph 5 (f) Graph 6

(g) Graph 7 (h) Graph 8 (i) Graph 9

Figure 1: An execution of the HRC$_\text{b}$ algorithm through various stages. The sets SI, SC, and SCC are illustrated with green, blue, and gray colors, respectively.

3.3 Formulating the Cost

In the previous part, we introduced HRC framework and explained HRC$_\text{b}$ with an example. We showed that a strategic selection of the subgoal $X_{\text{sel, t}}$, can speed up the process of achieving the final subgoal X_n. However, to measure the effectiveness of a strategy to select subgoals in line 3 of Algorithm 1, we need to formalize the algorithm's cost, which is the expected system probes during Intervention Sampling and Subgoal Training. We start by formulating the cost at each iteration t:

$$C(t) = \left[\underbrace{\sum_{j=1}^{T}\left(\sum_{g'\in SI_t} w^{\tau_j}_{\text{int},g'} + w^{\tau_j}_{\text{exp}}\right)}_{\text{InterventionSampling}} + \underbrace{\sum_{g''\in SCC_t}\sum_{j=1}^{T'}\left(\sum_{g'\in SI_t} w^{\tau_j}_{\text{int},g'} + w^{\tau_j}_{\text{train},g''}\right)}_{\text{SubgoalTraining}} |SI_t \right], \qquad (2)$$

where SI_t is the intervention set, T and T' are the number of trajectories collected in Intervention Samping and Subgoal Training steps, respectively. τ is a trajectory and $w^{\tau_j}_{\text{int},g'}$, $w^{\tau_j}_{\text{exp}}$, and $w^{\tau_j}_{\text{train},g''}$ represent system probes. During Intervention Sampling (Algorithm 2), we collect T trajectories. In each trajectory τ_j, we randomly select a subgoal $X_i \in SI_t$ and perform an intervention on it. This can be accomplished by taking actions based on the subgoal-based policy $\pi(a \mid s, X_i)$ until the subgoal X_i is achieved at some time t^* (see Remark 3.3). After each intervention, interventional data is gathered as specified in Definition 2.3. A trajectory is terminated when all subgoals $X_i \in SI_t$ have been achieved or when the horizon is reached. Note that, during an intervention on a randomly chosen subgoal X_i, it may be necessary to first achieve other subgoals $g' \in SI_t$. Therefore, for each τ_j, we consider the system probes obtained toward achieving each subgoal $g' \in SI_t$ regardless of whether it was selected as the intervention subgoal (X_i) or not. We denote this cost by $w^{\tau_j}_{\text{int},g'}$. Additionally, we need to consider interventional data collected during the trajectory τ_j. This data is gathered in order to explore the environment and discover reachable subgoals. The number of system probes for exploration in trajectory τ_j is denoted by $w^{\tau_j}_{\text{exp}}$.

In Subgoal Training, for each reachable subgoal in SCC_t, which is denoted by g'', we gather T' trajectories to train the policy for achieving it. For each trajectory τ_j, similar to the reasoning above, we consider the system probes obtained toward achieving each subgoal. Note that $w^{\tau}_{\text{train},g''}$ represents the portion of system probes towards achieving the subgoal g'', during the intervention on g'' by taking actions based on the subgoal-based policy $\pi(a \mid s, g'')$.

In order to compute the total cost of Algorithm 1, we model its execution as an MDP, where the state space is the power set $\mathcal{P}(\mathcal{X})$. A state, denoted by $\mathcal{I}$, represents the intervention set SI. The action space is $\mathcal{X}$, and an action is determined by X_{sel}. If we are in state $\mathcal{I}$ and choose X_{sel}, then we transit to state $\mathcal{I} \cup \{X_{\text{sel}}\}$ with a probability determined by a taken strategy.

Note: In the first iteration of the algorithm, no $X_{\text{sel, t}}$ is chosen because the controllable set SC initially contains no elements. Therefore, the cost formulation explicitly applies to iterations where $t > 0$. Let $C_{X_n}(\mathcal{I})$ be the expected cost of adding the final subgoal X_n to the intervention set; when the current set is $\mathcal{I}$:

$$C_{X_n}(\mathcal{I}) = \sum_{X_{\text{sel, t}} \in V} p_{\mathcal{I},\mathcal{I}\cup\{X_{\text{sel}}\}} \Big[C_{trans}(\mathcal{I}, \mathcal{I} \cup \{X_{\text{sel}}\}) + C_{X_n}(\mathcal{I} \cup \{X_{\text{sel}}\}) \Big], \tag{3}$$

where it has two terms: 1-the transiting cost and 2-the cost-to-go. Specifically, the transiting cost $C_{trans}(\mathcal{I}, \mathcal{I} \cup \{X_{\text{sel}}\})$ is the cost of transiting from the current interventions set $\mathcal{I}$ to a new set by adding X_{sel} to it ($\mathcal{I} \cup \{X_{\text{sel}}\}$), which is indeed the formulated cost in equation (2). The cost-to-go $C_{X_n}(\mathcal{I} \cup \{X_{\text{sel}}\})$ shows the future costs from the new state onwards. Both components are weighted by the transition probability $p_{\mathcal{I},\mathcal{I}\cup\{X_{\text{sel}}\}}$, which shows the probability of transitioning from state $\mathcal{I}$ to $\mathcal{I} \cup \{X_{\text{sel}}\}$ which is determined by the strategy. Based on this setting, the total cost is $C_{X_n}(\{\})$, which is as follows (proof in Appendix 9.1):

$$R_{\mathcal{I}_1} + \sum_j q^{(1)}_{1j} R_{\mathcal{I}_j} + \sum_j q^{(2)}_{1j} R_{\mathcal{I}_j} + \ldots + \sum_j q^{(n-1)}_{1j} R_{\mathcal{I}_j}, \tag{4}$$

where, $q^{(k)}_{ij}$ denote the ij-th entry of the k-th power of the transition probability matrix P. In other words, $(P^k)_{ij} = q^{(k)}_{ij}$. $R_{\mathcal{I}_j}$ is defined in the equation (60). The above equation shows that different strategies in selecting X_{sel}, will be reflected through transition probabilities, denoted as $q^{(k)}_{1j}$'s in the final cost.

4 Subgoal Discovery with Targeted Strategy

In this section, we suggest three heuristics that utilize the estimated causal model to prioritize controllable subgoals for selecting $X_{\text{sel, t}}$ in line 3 of Algorithm 1.

4.1 Causal Impact Heuristic

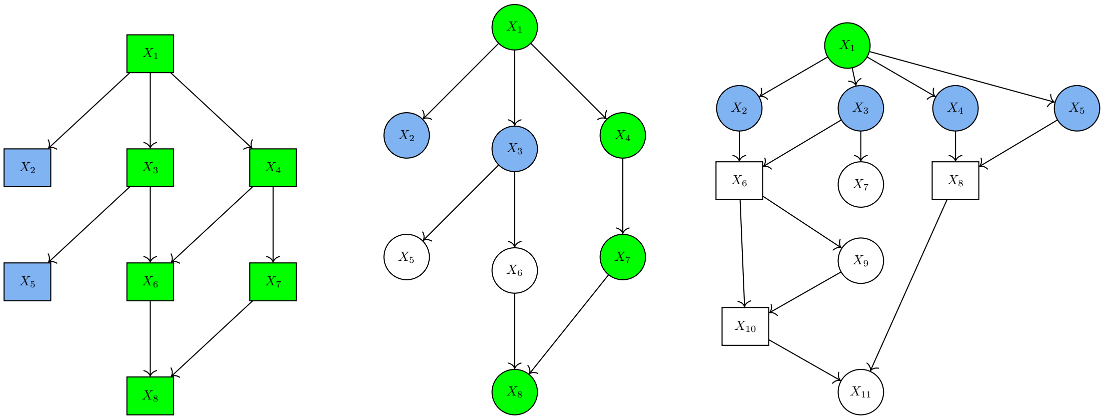
In this heuristic, for every $X_i \in SC$, we calculate the expected causal effect of X_i on the final subgoal X_n and select the subgoal with the maximum causal effect. Specifically,

$$X_{\text{sel}} = \underset{X_i \in SC}{\arg\max} \left(ECE^{\Delta}_{t^*}(\{X_i\}, \{\}, X_n) \right) \text{ (see 14 for more details about } \Delta \text{ and } t^*.)$$

For a scenario that every subgoal in the set $\mathcal{X}$ is of the AND type, all subgoals that have a path to X_n must have a non-zero causal effect and should be added to the intervention set SI. This guarantees that, for this scenario, this heuristic yields the minimum cost under a good estimate of the true causal model. For example, in Figure 2a, in order to achieve the final subgoal X_8, all the green subgoals must be achieved. Consequently, we expect only $ECE^{\Delta}_{t^*}(\{X_2\}, \{\}, X_8)$ and $ECE^{\Delta}_{t^*}(\{X_5\}, \{\}, X_8)$ to be zero. In other words, Algorithm 1 will have the minimum cost if it only adds green nodes to the intervention set.

4.2 Shortest Path Heuristic

In this heuristic, we utilize A* search approach, and incorporate the adjacency matrix of the causal graph $C_{\mathscr{S}}$ as a heuristic. The A* algorithm is designed to find the weighted shortest path in a graph



(a) AND type subgoals (b) OR type subgoals (c) AND/OR type subgoals

Figure 2: Subgoal structures with different subgoal types. In Figures (a) and (b), our strategies add only the green nodes to the intervention set, which is the minimum cost path. Figure (c) shows a stage of the algorithm where X_1 is in the intervention set and X_2, X_3, X_4, X_5 are controllable. Our Hybrid heuristic aims to select X_4 and X_5 as X_{sel} for the next steps.

(see Appendix 10 for details of A* algorithm in general), by minimizing a cost function defined as follows:

$$f(X_i) = g(X_i) + h(X_i),$$

where in our setting, $g(X_i)$ represents the accumulated cost from the start of the search to the current subgoal X_i, i.e, accumulated cost up to the iteration k in which the subgoal X_i is added to the set SC:

$$g(X_i) = \sum_{t=1}^{k} C(t).$$

The heuristic function $h(X_i)$ estimates the remaining cost for achieving the final subgoal X_n from the current subgoal X_i. To compute $h(X_i)$, we use the structure of the recovered causal graph $C_{\mathscr{S}}$ and calculate the weighted shortest path from X_i to X_n within the adjacency matrix of the causal graph. The weight $w(X_u, X_v)$ for the edge from the node X_u to the node X_v is defined as the out-degree $\text{deg}_{\text{out}}(X_u)$ of the node X_u. Then we define $h(X_i)$ as follows:

$$h(X_i) = \text{WeightedShortestPath}_{C_{\mathscr{S}}}(X_i, X_n).$$

For a scenario that every subgoal in the set $\mathcal{X}$, is of the OR type, and based on Definition 10.1 in Appendix for A* search algorithms, this heuristic is admissible, thus guaranteeing that the total cost to the final subgoal ($g(X_n)$) is minimum. For example in Figure 2b, we achieve the final subgoal X_8 with the minimum cost if we only add green nodes to the intervention set during the algorithm.

4.3 Hybrid Heuristic

In this heuristic, we integrate the above two heuristics into a two-phase heuristic:

(i) For each subset $S \subseteq SC$, we calculate the $ECE^{\Delta}_{t^*}(S, SC \setminus S, X_n)$. This measures the impact of S on the final subgoal X_n. We keep subsets, where the ECE is not zero, and put them into a collection $\mathcal{S}$.

(ii) Next, we apply the A^* search method. For every $S \in \mathcal{S}$, function $f(S)$ is defined as follows:

$$f(S) = g(S) + h(S),$$

where:

- $g(S)$ represents the cumulative sum of transition costs up to the iteration k that all subgoals in S are added to SC. It can be defined as follows:

$$g(S) = \sum_{t=1}^{k} C(t),$$

where iteration k is the last iteration that all the subgoals $X_i \in S$ are added to SC.

- $h(S)$ is the estimated cost from S to the final subgoal X_n. The heuristic function $h(S)$ is computed as the number of distinct nodes that appear on the paths from each $X_i \in S$ to the final subgoal X_n, based on using the estimated adjacency matrix, i.e.,

$$h(S) = \sum_{X_j \in \mathcal{X}} \mathbf{1}_{X_j \in \text{path}(S \to X_n)},$$

where path$(S \to X_n)$ represents all the paths from any subgoal $X_i \in S$ to X_n including X_i and X_n.

We select the subset S that minimizes $f(S)$. Then, we add all the subgoals in S to the set SI in the subsequent iterations, one by one in an arbitrary order.

In Figure 2c, consider that we are in line 3 of Algorithm 1 at iteration t where $SI_{t-1} = \{X_1\}$ and $SC_{t-1} = \{X_2, X_3, X_4, X_5\}$ are the intervention set (green) and controllable set (blue) at the end of the iteration $t-1$, respectively. (i) We compute the expected causal effect ($ECE^{\triangle}_{t^*}$) for every subset of SC. Subsets with non-zero $ECE^{\triangle}_{t^*}$ will be added to the collection $\mathcal{S}$. For instance, if we select subset $\{X_2, X_3\}$ from SC, $ECE^{\triangle}_{t^*}(\{X_2, X_3\}, \{X_4, X_5\}, X_8)$ will be non-zero. Furthermore, if we select $\{X_3, X_4\}$, $ECE^{\triangle}_{t^*}(\{X_3, X_4\}, \{X_2, X_5\}, X_8)$ will be zero. With further evaluations, we identify three candidate sets with non-zero $ECE^{\triangle}_{t^*}$s: $S_1 = \{X_2, X_3\}$, $S_2 = \{X_4, X_5\}$, and $S_3 = \{X_2, X_3, X_4, X_5\}$, resulting in the collection $\mathcal{S} = \{S_1, S_2, S_3\}$. (ii) In this simple example, $g(S_1) = g(S_2) = g(S_3)$. Now, we compute the h function for each of these sets: $h(S_1) = |\{X_2, X_3, X_6, X_9, X_{10}\}| = 5$, $h(S_2) = |\{X_4, X_5, X_8\}| = 3$ and $h(S_3) = |\{X_2, X_3, X_6, X_9, X_{10}, X_4, X_5, X_8\}| = 8$. Therefore, $f(S_2) < f(S_1)$ and $f(S_2) < f(S_3)$, and we choose $S_2 = \{X_4, X_5\}$ for the next selections of X_{sel}.

4.4 Cost Analysis

In this section, we analyze the cost of Algorithm 1 for two different strategies (random strategy and targeted strategy) by using the equation (4). We consider two subgoal structures: a Tree graph $G(n, b)$, where b is the branching factor; and a semi-Erdős–Rényi graph $G(n, p)$, with $p = \frac{c \log(n)}{n-1}$ and $0 < c < 1$ (see exact definition in Appendix 9.3).

Theorem 4.1. *Let $G(n, b)$ represent a Tree graph with branching factor b, and let $G(n, p)$ be a semi-Erdős–Rényi graph where $p = \frac{c \log(n)}{n-1}$ and $0 < c < 1$. Under these graph structures, the worst-case complexity of the targeted strategy (Hybrid heuristic which is denoted by HRC$_h$) is significantly better than the random strategy (HRC$_b$) in terms of the number of subgoals (n). In our analysis, we assume that CasualDiscovery subroutine in 1 has no error in finding reachable subgoals. Moreover, we assume that our heuristics have no error under a good estimate of the causal model. In Table 1, we provide a comparative analysis between the HRC$_b$ and HRC$_h$. Proofs are provided in Appendices 9.2 and 9.3.*

	Tree $G(n, b)$	**semi-Erdős–Rényi** $G(n, p)$
Targeted selection (HRC$_\mathrm{h}$)	$O(\log^2(n)b)$	$O(n^{4/3+2/3c}\log(n))$
Random selection (HRC$_\mathrm{b}$)	$\Omega(n^2 b)$	$\Omega(n^2)$

Table 1: The table shows the comparative cost, measured in terms of the number of subgoals involved n. HRC$_\mathrm{h}$ (our targeted strategy) significantly reduces the cost compared to the random strategy denoted by the HRC$_\mathrm{b}$. (Proofs can be found in Appendix 9).

Table 1 shows that HRC$_\mathrm{h}$ significantly reduces the cost compared to HRC$_\mathrm{b}$, particularly, when the subgoal structure is not very dense, or when there is a possibility to achieve the final goal earlier.

5 Causal Discovery

In this section, we focus on the Causal Discovery part of Algorithm 1 and propose a new causal discovery algorithm that enhances the efficiency of learning the subgoal structure. For this purpose, it is essential to identify the scale at which the subgoal structure can be learned. First we need the definitions bellow:

Definition 5.1 (One-sided valid assignment)**.** Let $\mathbf{X} \in \{0,1\}^n$. We say $\mathbf{X}$ is a *one-sided valid assignment* if it satisfies the following conditions for every subgoal indexed by i:

$$\begin{cases} \text{OR subgoal:} & (X_i = 1) \implies \exists X_j \in PA_{X_i}, X_j = 1, \\ \text{AND subgoal:} & (X_i = 1) \implies \forall X_j \in PA_{X_i}, X_j = 1. \end{cases}$$

Definition 5.2 (Discoverable Parent)**.** Consider a subgoal X_i. A parent $X_j \in PA_{X_i}$ of this subgoal is called *discoverable* if the following condition is satisfied for OR and AND subgoals, respectively:

- **For an OR subgoal:** There exist vectors $\mathbf{X}, \mathbf{X}' \in \{0,1\}^n$ (one-sided valid assignments) such that

  $$X_j = 1 \text{ and } X'_j = 0,$$
  $$\forall X_k \in PA_{X_i} \setminus \{X_j\}, X_k = X'_k = 0,$$

  thus indicating that the output of X_i changes value from 0 to 1 due to the presence of X_j, with all other parents set to 0.

- **For an AND subgoal:** There exist vectors $\mathbf{X}, \mathbf{X}' \in \{0,1\}^n$ (one-sided valid assignments) such that

  $$X_j = 1 \text{ and } X'_j = 0,$$
  $$\forall X_k \in PA_{X_i} \setminus \{X_j\}, X_k = X'_k = 1,$$

  thus indicating that the output of X_i can only achieve a value of 1 when X_j is 1, provided all other parents are already set to 1.

Remark 5.3. Generally speaking, under Assumption 3.2, the interventional data collected in Algorithm 1 is not faithful to the underlying subgoal structure (see the section 7.3). Consequently, causal discovery algorithms are limited to learning the subgoal structure only to the extent of the discoverable parents.

To model the subgoal structure of subgoals within our setting, we introduce an abstracted structural causal model (a-SCM) where the variables represent subgoals, denoted as $\mathcal{X} = \{X_1, \cdots, X_n\}$. This abstraction enables us to focus on the relationships between subgoals while skipping non-binary intermediate environment variables ($\mathcal{E} \setminus \mathcal{E}_s$). Note that the time step t in our a-SCM is not on the same scale as the SCM defined in 2. In our a-SCM, the value of X_i^{t+1} is determined as a function of the variables in the system at time t and an error term ϵ_i^{t+1}:

$$X_i^{t+1} = g_i(\mathbf{X}^t) \oplus \epsilon_i^{t+1}, \qquad 1 \leq i \leq n, \tag{5}$$

where g_i is either AND or an OR operation defined as

$$g_i(\mathbf{X}^t) = \begin{cases} \bigwedge_{X_j \in PA_{X_i}} X_j^t & \text{if AND operation,} \\ \bigvee_{X_j \in PA_{X_i}} X_j^t & \text{if OR operation,} \end{cases}$$

and $\oplus$ denotes the XOR operation. Moreover, the error term ϵ_i^{t+1} has Bernoulli distribution with parameter $\rho < 1/2$.

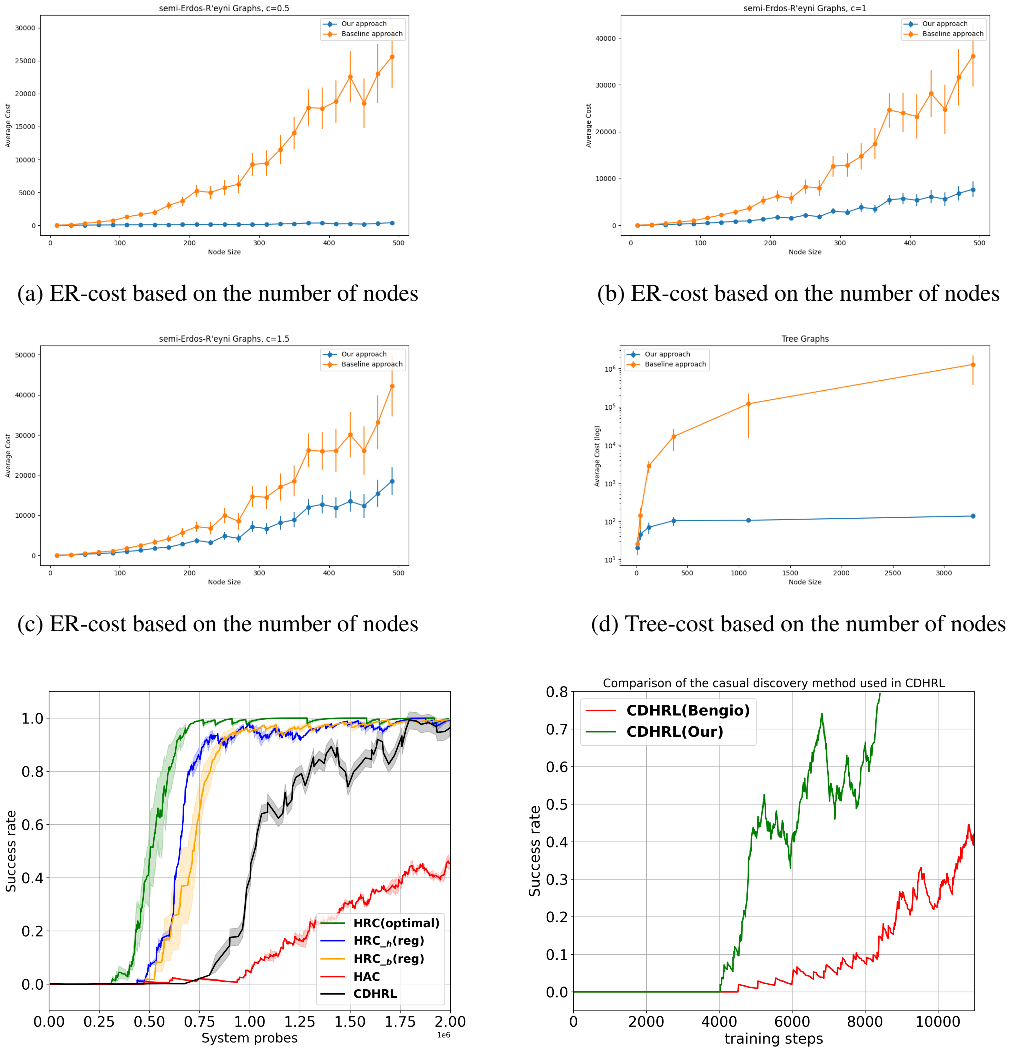
Theorem 5.4. *Given the a-SCM defined above, for a variable X_i^{t+1} and $\boldsymbol{\beta} \in \mathbb{R}^n$, consider function S as*

$$S(\mathbf{X}^t, \boldsymbol{\beta}) = \sum_j \beta_j X_j^t + \beta_0. \tag{6}$$

Let $\hat{X}_i^{t+1} = \mathbb{1}\{S(\mathbf{X}^t, \boldsymbol{\beta}) > 0\}$ be an estimate of X_i^{t+1}. For any vector $\boldsymbol{\beta}$, consider the following loss function:

$$\mathcal{L}(\boldsymbol{\beta}) = \mathbb{E}[(\hat{X}_i^{t+1} - X_i^{t+1})^2] + \lambda \|\boldsymbol{\beta}\|_0. \tag{7}$$

There exists a $\lambda > 0$ such that for any optimal solution $\boldsymbol{\beta}^*$ minimizing the loss function in (7), the positive coefficients in $\boldsymbol{\beta}^*$ correspond to the parents of X_i^{t+1} in $\mathbf{X}^t$.



(a) ER-cost based on the number of nodes

(b) ER-cost based on the number of nodes

(c) ER-cost based on the number of nodes

(d) Tree-cost based on the number of nodes

(e) Minecraft

(f) Minecraft

6 Experimental Results

In this section, we present experimental results that demonstrate the superior performance of our heuristics over $\mathrm{HRC_b}$ referred to as the "Baseline approach". In the case of synthetic data, Figures 3a, 3b, and 3c show the cost results under semi-Erdős–Rényi graph structures $G(n, p)$ with $p = \frac{c \log(n)}{n-1}$, for varying values of c. Our heuristic approach consistently outperforms the $\mathrm{HRC_b}$. Additionally, Figure 3d illustrates that the cost of our methodology outperforms under a Tree graph $G(n, b)$, where b represents the branching factor. Note that we used a form of error $\frac{1}{1+t}$ for the heuristic applied.

In real-world applications, we choose 2d-Minecraft [16] the same as [4].

References

[1] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

[2] 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.

[3] Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *Journal of Machine Learning Research*, 23(152):1–41, 2022.

[4] Xing Hu, Rui Zhang, Ke Tang, Jiaming Guo, Qi Yi, Ruizhi Chen, Zidong Du, Ling Li, Qi Guo, Yunji Chen, et al. Causality-driven hierarchical structure discovery for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20064–20076, 2022.

[5] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[6] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[7] Siyuan Li, Lulu Zheng, Jianhao Wang, and Chongjie Zhang. Learning subgoal representations with slow dynamics. In *International Conference on Learning Representations*, 2021.

[8] Amy McGovern and Andrew G Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. 2001.

[9] Arquímides Méndez-Molina, Ivan Feliciano-Avelino, Eduardo F Morales, and Luis Enrique Sucar. Causal based q-learning. *Res. Comput. Sci.*, 149(3):95–104, 2020.

[10] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

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

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

[13] Jürgen Schmidhuber. Learning to generate sub-goals for action sequences. In *Artificial neural networks*, pages 967–972, 1991.

[14] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:22905–22918, 2021.

[15] Özgür Şimşek, Alicia P Wolfe, and Andrew G Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pages 816–823, 2005.

[16] Sungryull Sohn, Junhyuk Oh, and Honglak Lee. Hierarchical reinforcement learning for zero-shot generalization with subtask dependencies. *Advances in neural information processing systems*, 31, 2018.

[17] Jonathan Sorg and Satinder Singh. Linear options. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 31–38, 2010.

[18] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[19] 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.

[20] Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Explainable reinforcement learning via a causal world model. *arXiv preprint arXiv:2305.02749*, 2023.

[21] 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.