Landmark Guided Active Exploration with Stable Low-level Policy Learning

Fei Cui, Jiaojiao Fang, Mengke Yang, Guizhong Liu

***Abstract*—Goal-conditioned hierarchical reinforcement learning (GCHRL) decomposes long-horizon tasks into sub-tasks through a hierarchical framework and it has demonstrated promising results across a variety of domains. However, the high-level policy's action space is often excessively large, presenting a significant challenge to effective exploration and resulting in potentially inefficient training. Moreover, the dynamic variability of the low-level policy introduces non-stationarity to the high-level state transition function, significantly impeding the learning of the high-level policy. In this paper, we design a measure of *prospect* for subgoals by planning in the goal space based on the goal-conditioned value function. Building upon the measure of prospect, we propose a landmark-guided exploration strategy by integrating the measures of prospect and novelty which aims to guide the agent to explore efficiently and improve sample efficiency. To address the non-stationarity arising from the dynamic changes of the low-level policy, we apply a state-specific regularization to the learning of low-level policy, which facilitates stable learning of the hierarchical policy. The experimental results demonstrate that our proposed exploration strategy significantly outperforms the baseline methods across multiple tasks.**

***Index Terms*—Hierarchical reinforcement learning (HRL), subgoal, exploration-exploitation, state-specific regularization.**

I. Introduction

DEEP Reinforcement Learning (DRL) is a powerful approach for solving sequential decision-making problems, such as video games [1], [2] and robot navigation [3]–[5]. DRL models these problems as partially observable Markov decision processes (POMDPs), and learns the optimal policies by maximizing the cumulative discounted reward. However, in many complex tasks, agents often struggle to collect sufficient high-reward trajectories. Hierarchical reinforcement learning decomposes complex, long-horizon decision-making tasks into sub-tasks of different time scales and is a promising method for solving such long-horizon tasks. Goal-conditioned hierarchical reinforcement learning [6]–[9] is a two-level hierarchical reinforcement learning paradigm, where the high-level policy decomposes the original task into a series of subgoals, and the low-level policy guides the agent to achieve these subgoals. The learning objective of the hierarchical policy is still to learn a decision function that maximizes the cumulative expected reward, so the learning of the high-level policy depends on the external rewards, while the learning of the low-level policy depends on the intrinsic rewards defined through the subgoals.

The authors are with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. Correspondence to: Guizhong Liu <liugz@xjtu.edu.cn>.

Effective subgoals are crucial for achieving good performance and efficiency in goal-conditioned hierarchical reinforcement learning. Selecting reasonable subgoals that capture the task's semantics provides meaningful guidance for low-level policy learning. Pre-defined subgoal space [10], [11] and task-specific subgoal representation space [12]–[15] learned online can be employed to better represent the action space of the high-level policy. Pre-defined subgoals can quickly supervise low-level policy learning via intrinsic rewards, while learned subgoal representations can be optimized for specific tasks. However, sampling actions in a large subgoal space can lead to inadequate exploration and inefficient training of the agent's high-level policy.

To guide the agent to explore efficiently, some approaches [10], [11] reduce the complexity of the high-level action space by using adjacency constraints, which promotes the agent to explore reasonable states. To further avoid introducing additional non-stationarity in HRL, HESS [15] proposes an active exploration strategy that considers a combined measure of *novelty* and *potential* for subgoals after stabilizing the learning of subgoal representation space. The novelty measure aims to enhance the agent's ability to explore new states, while the potential measure guides the agent to explore in the direction that is more likely to expand the explored area. HESS is effective in guiding the agent to unexplored areas, but it ignores the impact of the final goal on exploration. Therefore, relying solely on the measures of novelty and potential may not always guide the agent to the most promising areas.

Our insight is that the agent not only needs to expand the exploration area but also needs to pay attention to the areas that are more likely to achieve the final task goal. To this end, we design a *prospect* measure for subgoals through landmark-based planning in the subgoal representation space. This measure can reflect the likelihood of exploring in the direction of a subgoal leading the agent to states closer to achieving the final task goal. Considering the measure of prospect and novelty for subgoals, we propose a **L**andmark-guided active **E**xploration strategy with **S**table low-level **P**olicy learning (LESP). The strategy incorporates the measure of prospect to guide the agent to explore subgoals that are more likely to lead to the final task goal. Additionally, In goal-conditioned hierarchical reinforcement learning, the high-level state transition function is dependent not only on the physical environment but also on the online-learned low-level policy. Due to the dynamic changes of the low-level policy, the non-stationarity of the high-level state transition function greatly hinders the learning of the high-level policy. Some prior works [16], [17] have attempted to re-label the actions of the high-

level state transitions (i.e., subgoals) to make them adaptable to the dynamic changes in the low-level policy. In this paper, we apply state-specific regularization to the learning of low-level policy to alleviate the non-stationarity caused by the dynamic changes in the low-level policy and facilitate the learning of the high-level policy. The proposed LESP strategy effectively balances the exploration-exploitation trade-off, and enables efficient learning of goal-conditioned hierarchical policies.

We compare the proposed method LESP with the state-of-the-art baselines in the Mujoco experimental environment [18]. The experimental results demonstrate that LESP, which takes into account the guidance of the task goal for exploration, outperforms the baseline methods. Additionally, we conduct ablation experiments to verify the roles of different components of LESP.

The article is organized as follows: Section II covers the relevant works. Section III discusses the essential preliminary concepts. Section IV describes the proposed method LESP. In Section V, experiments and results are discussed. Finally, this article is concluded in Section VI.

II. Related Work

When dealing with long horizon decision-making tasks, how to guide the agent to explore promising trajectories is a crucial problem. Hierarchical reinforcement learning (HRL) [13], [19]–[23] improves sample efficiency by decomposing complex tasks. Selecting reasonable subgoals during interaction with the environment can also avoid blind exploration. Related works are described from these two aspects.

A. Hierarchical Reinforcement Learning

Hierarchical Reinforcement Learning divides the original task into sub-tasks at different timescales using a hierarchical structure. The high-level policy communicates with the low-level policy through subgoals, and the signal passed from the high-level policy to the low-level policy can vary across different tasks, ranging from using discrete value for option [24]–[26] to employing pre-defined subgoal space [10], [11], [16] or subgoal representation space learned online [12]–[15], [27], [28]. The use of discrete-valued options naturally reduces the complexity of the high-level action space, but the limited rules decrease the adaptability of the solution to different tasks. On the other hand, learning subgoal representation often results in a high-dimensional action space, which can hinder the agent's ability to explore effectively. To explore effectively, HRAC [10] restricts the high-level action space within the k-step adjacency area through adjacency constraints, while HIGL [11] samples landmarks through coverage-based sampling and novelty-based sampling in the replay buffer, restricting the actions of the high-level policy within the domain of the most urgent landmark, but introduces additional non-stationarity. In order to choose appropriate subgoals to guide exploration, HESS [15] proposes an exploration strategy by considering the measure of novelty and potential for subgoals but ignores the guidance of the task final goal for exploration. In contrast, our approach plans landmarks in the subgoal representation space according to the task goal and designs a measure of *prospect* that considers the influence of the task's final goal, proposing a more efficient hierarchical exploration strategy.

B. Subgoal Selection

When employing deep reinforcement learning to solve complex sequential decision-making tasks, selecting appropriate subgoals for the agent can be an effective strategy. Several studies [29]–[33] focus on learning goal-conditioned value functions and rationally design reward functions to make the value function reflect the reachability between two states. These approaches allow for planning in the goal space based on reachability, with states on the planned path selected as subgoals. With good perception of the environment map, some rule-based methods [3], [34], [35] utilize heuristic search method to find an optimal trajectory, and sample subgoals along the trajectory based on physical priors. After utilizing a value function to learn the reachability between states in the state space, L3P [32] clusters the candidate states based on their reachability, with each cluster center representing a potential landmark. Then, a sparse topological graph is constructed with the potential landmarks as nodes and reachability as edges. Finally, subgoals are generated by conducting graph search algorithm on the constructed graph. On the other hand, HIGL [11] uses coverage-based sampling and novelty to sample landmarks from the replay buffer. To facilitate effective exploration with reasonable subgoals, the latest work HESS [15] samples candidate landmarks in the neighborhood of the agent's current state and selects subgoals based on the measures of novelty and potential. This active exploration strategy avoids introducing additional non-stationarity and speeds up the learning process of the hierarchical policy. Like previous works [11], [32], we also utilize a goal-conditioned value function to plan landmarks in the goal space. We designed a measure of *prospect* based on the landmarks. Considering measures of *novelty* and *prospect* to select subgoals, we proposed an active hierarchical exploration strategy.

III. Preliminaries

Reinforcement learning formulates the sequential decision-making problem as a Markov decision process (MDP) [36], defined as a tuple $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma >$ where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ represents the state transition function that reflects the dynamics of the environment, r is the reward function typically designed by human experts for the task, and $\gamma \in [0, 1)$ is a discount factor. A policy $\pi(a|s)$ maps a given state s to a probability distribution over the action a. The goal of reinforcement learning is to learn a policy that maximizes the expected cumulative discounted reward $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$, where r_t is the immediate reward that the agent receives from the environment after taking action a_t at state s_t. Reinforcement learning is mainly divided into two categories: the value-based methods [37], [38] and the policy gradient methods [39]–[41]. Value-based methods compute the state-action value function and choose actions greedily based on the computed value function. Policy gradient methods optimize the policy directly by the policy gradient computed on the value function. To encourage

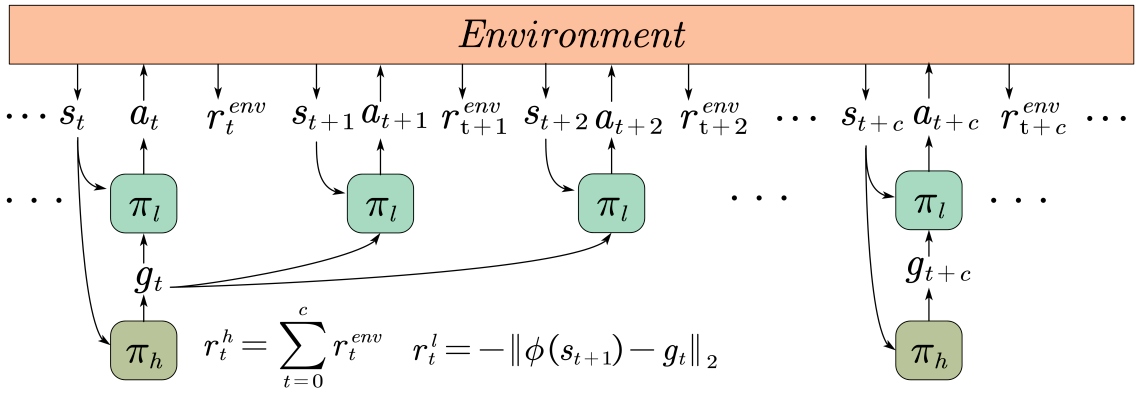

Fig. 1. The framework of goal-conditioned hierarchical reinforcement learning (GCHRL, [6]–[9]). ϕ is the subgoal representation function that maps the state to the goal space. The hierarchical framework consists of a high-level policy and a low-level policy. The reward of the high-level policy is a sum of c (the low-level policy length) external rewards, while the reward of the low-level policy is the negative distance between the state and subgoal in the latent space.

the agent to explore the state space, we adopt the Soft Actor-Critic (SAC) [39] algorithm for both the high-level and the low-level policies in our experiments. In SAC, the standard value loss function is:

$$L_Q(\theta) = \mathbb{E}_{s_t,a_t,s_{t+1}\sim\mathcal{B},a_{t+1}\sim\pi_\phi}[\frac{1}{2}(Q_\theta(s_t,a_t) - (r(s_t,a_t) + \gamma Q_\theta(s_{t+1},a_{t+1}) - \alpha\log(\pi_\phi(a_{t+1}|s_{t+1})))^2] \tag{1}$$

where γ is the discount factor, α denotes the temperature coefficient, $\mathcal{B}$ is the replay buffer, and π_ψ represents the policy. The policy loss function can be expressed as follows:

$$L_\pi(\psi) = \mathbb{E}_{s_t\sim\mathcal{B},a_t\sim\pi_\psi}[\log\pi_\psi(a_t|s_t) - \frac{1}{\alpha}Q_\theta(s_t,a_t)] \tag{2}$$

Goal-conditioned hierarchical reinforcement learning models long-horizon decision-making tasks as a goal-conditioned Markov decision process, $M =< \mathcal{S},\mathcal{G},\mathcal{A},\mathcal{P},r,\gamma >$, where $\mathcal{G}$ represents the goal space. As illustrated in Figure 1, goal-conditioned reinforcement learning is a hierarchical framework consisting of two policies. The high-level policy $\pi_h(g|s)$ operates at a lower temporal resolution, sampling a high-level action every c time steps (i.e., $t \equiv 0(\mathrm{mod}c)$. When $t \not\equiv 0(\mathrm{mod}c)$, a predefined function such as the identity function is used to specify a subgoal for the low-level policy. Given the current state s_t and subgoal g_t, the low-level policy $\pi_l(a_t|s_t,g_t)$ produce low-level actions that are executed by the agent to interact with the environment. During training, since the goal of the low-level policy is to enable the agent to achieve the subgoals specified by the high-level policy, the low-level policy is optimized using intrinsic rewards $-\|\phi(s_{t+1}) - g_t\|_2$ computed from the sub-goal, where ϕ is the sub-goal representation function that maps the state to the goal space. Similar to the standard reinforcement learning paradigm, the learning objective of the hierarchical structure is still to enable the agent to interact efficiently with the external environment, so the reward of the high-level policy is defined as the sum of c external rewards $\sum_{t=0}^{c} r_t^{env}$ after executing a high-level action.

In the goal-conditioned reinforcement learning, the high-level policy and the low-level policy can be trained simultaneously in an end-to-end manner. The high-level policy provides real-time subgoals to the low-level policy and guides its learning through intrinsic rewards. The dynamic low-level policy learned online influences the stationarity of the high-level state transitions. Therefore, a well-designed hierarchical exploration strategy can significantly enhance the learning efficiency of the hierarchical policy.

IV. Method

In this section, we propose LESP: **L**andmark guided active **E**xploration with **S**table Low-level **P**olicy learning. We describe LESP from three aspects: measures for subgoals, stable low-level policy learning and hierarchical exploration strategy.

A. Measures for Subgoals

In goal-conditioned hierarchical reinforcement learning, effectively guiding the agent's exploration is crucial for improving the algorithm performance. Previous count-based exploration methods [11], [15] define the novelty of subgoals based on the number of visits to states. However, solely relying on visit counts may not always guide the agent to explore promising states. The state-of-the-art exploration approach [15] introduces the measure of potential for subgoals, aiming to effectively guide the agent to explore the unexplored region. Our insight is that reasonable subgoals should not only guide the agent towards expanding the exploration area but also guide the agent to explore regions that are likely to lead to the ultimate goal. To address this, we design the measure of *prospect* for subgoals, which reflects the subgoal's positivity towards achieving the final goal.

The prospect measure requires specifying the exploration direction for the agent. Following prior work [11], [32], we choose to plan landmarks in the subgoal representation space. We aim to sample landmarks that cover a wide range of the goal space. To achieve this, we employ the Farthest Point Sampling (FPS) [43] algorithm to sample n_{cov} landmarks from the goal space. Starting with an initial candidate set, we iteratively add the farthest landmark to the sampled landmark set until a sufficient number of landmarks are sampled where

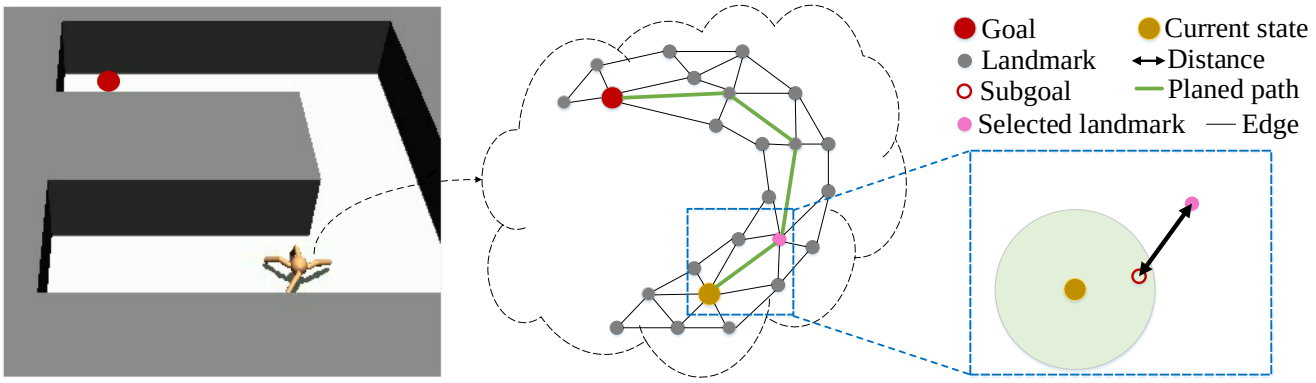

Fig. 2. landmark selection and prospect calculation process. The calculation of prospect involves four stages: 1) **Sampling**: An adequate number of sample points are randomly selected from the state space. Then, the FPS algorithm is employed to sample n_{cov} landmark points. 2) **Building a graph**: The sampled landmarks, current position, and goal are used as nodes to build a graph. The edges of the graph represent the reachability between two nodes. 3) **Path planing**: Using the shortest path planning algorithm, a feasible path from the current position to the goal is determined based on the constructed graph. 4) **Calculation**: Landmark is sampled along the trajectory and selected as l_{sel}. The prospect of the subgoals (within the neighborhood of the current position) is then calculated based on the selected landmark.

the distance between two states is measured by the Euclidean distance in the goal space. This sampling process ensures that the sampled landmarks are spread out and cover a diverse range of region in the goal space. To search for a promising path from the current state to the goal, we build a graph consisting of the current state, the goal, and the sampled landmarks as nodes; following previous works [11], [29], [30], [32] and [42], the edges between nodes are weighted based on the goal-conditioned value function $V(s_1, \phi(s_2))$. This value function reflects the reachability between two states s_1 and s_2. Once the graph is built, we can plan the landmarks on the graph. By applying a shortest path planning algorithm, we can find a feasible path from the current state to the goal, selecting the landmark l_{sel} on the path that is closest to the current state as the region the agent should explore. Algorithm 1 describes the process of landmark selection.

Algorithm 1 Landmark Selection

Input: current state s_t, goal g, threshold τ, $\phi(s)$ and goal-conditioned value function $V(s, g)$
Output: selected lanmark l_{sel}
Initial n transitions $T = (s, a, s')$ from buffer B_{pre}
$N \leftarrow$ **FPS**$(S = \{s \parallel (s, a, s') \in T\}) \cup \{g\}$
$W_{i,j} \leftarrow \infty$
for $\forall(n_i, n_j) \in N \times N$ **do**
  $w_{i,j} \leftarrow [-V(n_i, \phi(n_j))]$
  if $w_{i,j} \leq \tau$ **then**
    $W_{i,j} = w_{i,j}$
  end if
end for
$Trajectory\ K \leftarrow$ **Shortest Path Planing**(N, W)
$k_{sel} \leftarrow \text{argmin}_{k_i \in K} - V(s_t, \phi(k_i))$
$l_{sel} = \phi(k_{sel})$
return l_{sel}

In practice, before training the hierarchical policy, we allow the agent randomly walk in the environment and collect trajectories stored in buffer B_{pre} (not the replay buffer $\mathcal{B}$). We then use these trajectories to train the goal-conditioned value function $V(s, g)$. For the current state s_t, after selecting landmark l_{sel} with Algorithm 1, the exploration strategy considers selecting a subgoal g_t near the current state s_t, where the *prospect* of the subgoal g_t is defined as follows:

$$P(g_t) = - \|g_t - l_{sel}\|_2 \tag{3}$$

The landmark selection and prospect calculation process are depicted in Figure 2. The prospect measure takes into account the influence of the goal on the subgoal selection by considering the feasible path to the goal in the latent space. In contrast to previous work, the prospect measure not only encourages the agent to explore unexplored region but also guides the agent to explore region that have a positive impact on achieving the task goal. To maintain the agent's ability to explore new states, similar to previous work [11], [15], we also consider the novelty measure for subgoals. The novelty of a subgoal is measured by discrete counting in the replay buffer $\mathcal{B}$, defined as:

$$N(s_i) = \mathbb{E}_{s_{i+jc} \sim \mathcal{B}}[- \sum_{j=0}^{\lfloor (T-i)/c \rfloor} \gamma^j n(\phi(s_{i+jc}))] \tag{4}$$

where c denotes the horizon for the low-level policy, γ is the discount factor and $n(\phi(s))$ indicates the immediate count of s in the replay buffer $\mathcal{B}$. Following HESS [15], In order to reduce the computational cost, we discretize the state space into cells and estimate $n(\phi(s))$ by counting the number of times each cell is visited.

B. Stable Low-level Policy Learning

Goal-conditioned hierarchical reinforcement learning is an effective paradigm for solving complex sequential decision-making tasks. However, the dynamic changes in low-level policy introduce non-stationarity in the high-level state transitions. Over time, taking the same high-level action in the

same state can lead to completely different state transitions. This is because high-level state transitions are not solely determined by the environmental dynamics. To mitigate the non-stationarity in learning the low-level policy, inspired by the stable learning of subgoal representations in HESS [15], we introduce a state-specific regularization term L_r in the loss of the low-level policy. L_r is designed to limit the estimation differences of the low-level Q-function at different time steps without compromising learning efficiency. The definition of L_r is as follows:

$$L_r(\theta) = \mathbb{E}_{s,g\sim\mathcal{B},a\sim\pi_l}[\lambda(s,g)\left\|Q_\theta(s,g,a) - Q_{\theta_{old}}(s,g,a)\right\|_2] \tag{5}$$

where $\mathcal{B}$ is the replay buffer, π_l is the low-level policy and $\lambda(s,g)$ is the regularization weight of the state-subgoal pair (s,g). In practice, we first calculate the loss $L_Q(\theta)$ for different state-subgoal pairs using Equation 1. Then, we set λ to 1 for the $k\%$ smallest state-subgoal pairs and 0 for the rest. This is to limit the variation of the value function for states where the agent has already explored sufficiently, while maintaining the learning efficiency of the low-level policy in unknown states. So the overall loss for the value function of the low-level policy is represented by $L_Q(\theta) + L_r(\theta)$.

C. Hierarchical Exploration Strategy

Active exploration strategy can directly influence the behavioral policy, avoiding the introduction of additional non-stationarity through intrinsic rewards. Selecting subgoals with high potential and novelty can facilitate the intelligent agent in expanding its exploration area. However, blindly exploring unknown region may lead to accumulation of ineffective experiences. In order to further enhance the efficiency of training the hierarchical policy, we propose an active hierarchical exploration strategy that takes into account the influence of the final goal. The proposed exploration strategy aims to maintain the agent's ability to explore unknown regions while prioritizing subgoal g_t with high prospect, specified as follows:

$$\begin{aligned} g_t &= \underset{\phi(s)}{\operatorname{argmax}}\, N(\phi(s)) + \alpha P(\phi(s)) \\ &\text{subject to} \begin{cases} D(\phi(s),\phi(s_t)) \le r_g \\ s \in \mathcal{B} \end{cases} \end{aligned} \tag{6}$$

where N and P are normalized measures of novelty and prospect, respectively, D represents the Euclidean distance, $\mathcal{B}$ is the replay buffer, ϕ is the subgoal representation function learned online, and α is the balancing coefficient. Algorithm 2 describes our active hierarchical exploration strategy. In LESSON [13], it has been demonstrated that the triplet loss based on slow features can capture the relative positional relationships in the state space. Therefore, following LESSON, we train the subgoal representation function ϕ using the triplet loss, as follows:

$$\begin{aligned} L_\phi = \mathbb{E}_{(s_t,s_{t+1},s_{t+c})\sim\mathcal{B}}[&\left\|\phi(s_t) - \phi(s_{t+1})\right\|_2 \\ &+max(0, \delta - \left\|\phi(s_t) - \phi(s_{t+c})\right\|_2)] \end{aligned} \tag{7}$$

where $\mathcal{B}$ is the replay buffer, δ is the margin parameter of triplet loss, c is the low level policy length of the hierarchical framework.

Algorithm 2 LESP algorithm

Initialize: $\pi_h(g|s), \pi_l(a|s,g)$ and $\phi(s)$
for $i = 1..episodeNum$ **do**
  for $t = 0..T-1$ **do**
    if $t \equiv 0(\text{mod}c)$ **then**
      Uniformly sample a number n in the range (0,1)
      if $n < p\,(p \in (0,1))$ **then**
        Select landmark l_{sel} with Algorithm 1
        Generate a candidate set of subgoals
        Calculate the prospect and novelty of subgoals by Eq.3 and Eq.4
        Select subgoal g_t by Eq.6
      else
        Execute $g_t \sim \pi_h(\cdot|s_t)$
      Update π_h
    else
      $g_t = g_{t-1}$
    end if
    Execute $a_t \sim \pi(\cdot|s_t, g_t)$
    Store experiences in replay buffer $\mathcal{B}$
    Update π_l by Eq.1 and Eq.5
  end for
  if $i \equiv 0(\text{mod}I)$ **then**
    Update ϕ using the triplet loss defined by Eq.7
  end if
end for
return: π_h, π_l, ϕ

V. Experiments

In order to evaluate the performance of the landmark-guided hierarchical exploration scheme, we conduct experiments on challenging sparse reward tasks in the Mujoco environment. The experimental design primarily aims to answer the following questions:

1) How does the landmark-guided active exploration strategy compare to the state-of-the-art active exploration method HESS in terms of performance?

2) How does the Prospect measure guide exploration and improve performance?

3) What is the importance of each component in the proposed hierarchical exploration strategy, as well as the impact of the hyper-parameters on the experiments?

A. Experimental Setup

We conduct experiments on several long-horizon decision making tasks based on the Mujoco engine, as illustrated in Figure 3.

Point Maze: Simulate a ball initially positioned in the bottom-left corner of a U-shaped maze, with the objective of reaching the top-left corner of the maze.

Ant Maze: Similar to Point Maze, but the agent is a simulated ant navigating through the maze environment.

Ant FourRooms: The task environment is a maze with four rooms. A simulated ant starts from the bottom-left corner and aims to reach a distant goal located in the top-right corner.

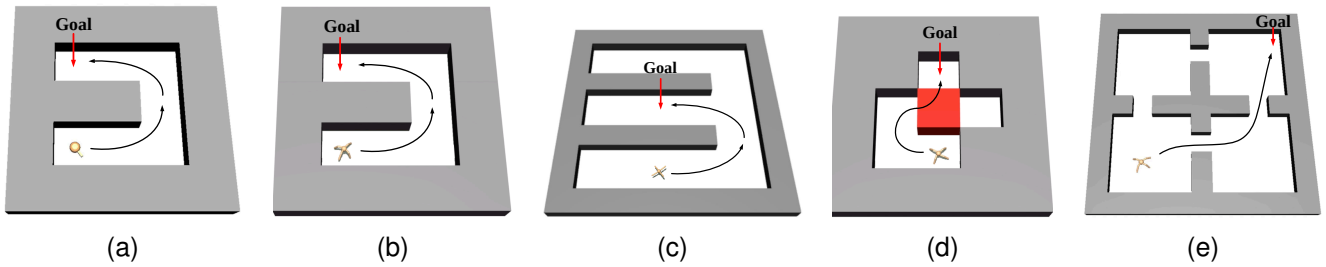

Fig. 3. The Mujoco environments we used for hierarchical exploration experiment. (a) Point Maze. (b) Ant Maze. (c) Ant Maze (W-shape). (d) Ant Push. (e) Ant FourRooms. In our experiments, all the task environments are designed with sparse reward. This setup adds to the challenge of the task, as the agents must explore and discover effective strategies to achieve the desired goal despite the scarcity of rewards.

Ant Push: In this task, there is a movable block obstructing the path of the ant towards the target. The ant needs to move the block aside in order to reach the goal.

Ant Maze (W-shape): In this environment, the maze structure is larger, 32 × 32. The start position is (14,0), and the goal is (14, 14).

In the experiments, all of these task environments are designed with sparse rewards, making the tasks more challenging and suitable for evaluating the exploratory capabilities of the algorithms. The adoption of sparse rewards ensures that the agents must actively explore and discover effective strategies to achieve their goals. This setup provides a rigorous evaluation of the algorithms' performance in handling tasks with limited feedback and encourages the development of efficient exploration techniques.

In all the experiments, the actor network for both the low-level and the high-level policies is a Multi-Layer Perceptron (MLP) with two hidden layers of dimension 256. The Critic network structure is the same as that of the actor network. The subgoal representation function is a MLP with one hidden layer of dimension 100. The activation function of the MLP is the ReLU function. All the experiments are carried out on NVIDIA GTX 2080 Ti GPU and optimized using the Adam optimizer. In all the experiments, the radius r_g of subgoal selection is set to 20. The discount factor γ is set to 0.99. The subgoal representation function is updated every 100 episodes. The batch size for both level policies is set to 128.

B. Comparative Analysis

We conduct a comparative analysis of our proposed algorithm with several existing hierarchical reinforcement learning methods, including HISM, HSR, HESS, LESSON, and SAC. (i) H-ICM utilizes curiosity as intrinsic rewards to guide the agent's exploration in the environment and learn useful skills. (ii) HESS introduces a potential measure for subgoal and proposes an active exploration strategy with stable subgoal representation learning. (iii) HSR introduces a count-based exploration method that utilizes implicit state visit counting to guide the exploration process. (iv) LESSON utilizes triplet loss to learn subgoal representations online and achieves effective hierarchical exploration by selecting slow features as the subgoal space representation. (v) SAC is a reinforcement learning algorithm based on the actor-critic mechanism. In the experiment, both the high-level policy and low-level policy of the hierarchical framework are trained by the SAC algorithm.

The experimental results depicted in Figure 4 demonstrate that, our proposed exploration strategy LESP outperforms all the baseline methods in handling hard-exploration tasks. The superiority of LESP can be attributed to its comprehensive consideration of both the novelty and the prospect in the exploration strategy. The calculation of prospect involves planning feasible trajectory based on reachability between landmarks, which takes into account the influence of task goal on exploration. Unlike the potential-based exploration approach in HESS, which only focuses on expanding the exploration area, LESP effectively guides the agent towards regions that have a positive impact on reaching the task goal. Additionally, we find that, in more complex tasks such as Ant FourRooms and Ant Maze (W-shape), which place greater demands on the exploration capability of the agent, the advantage of LESP in terms of sample utilization efficiency becomes more pronounced. It can be observed that, HICM and HSR fall short compared to our proposed method. HICM's performance is dependent on the learning error of the dynamic model; the instability of the intrinsic rewards restricts the learning of the high-level policy. HSR calculates state visit counts using the L_1 norm of the successor representation, but relying solely on the successor representation, proves inadequate in promoting effective exploration. The experimental results demonstrate that, SAC exhibits the poorest performance when compared to the goal-conditioned hierarchical framework LESSON, which incorporates online learning of subgoal representations. This finding serves as strong evidence supporting the effectiveness of the goal-conditioned hierarchical framework.

C. Qualitative Analysis on Measures for Subgoals

To assess the impact of subgoal measures on subgoal selection, we visualize the Prospect and Novelty measures in the AntMaze task, as shown in Figure 5. The visualization reveals that, subgoals with high Novelty are distributed throughout the candidate subgoal set, indicating that the agent can explore in various directions to expand the explored area. Figure 5 provides insightful observations regarding the Prospect measure. It indicates that subgoals guiding the agent closer to the goal exhibit higher Prospect. In contrast, subgoals that do not contribute significantly to task completion have lower Prospect. This suggests that the Prospect measure effectively

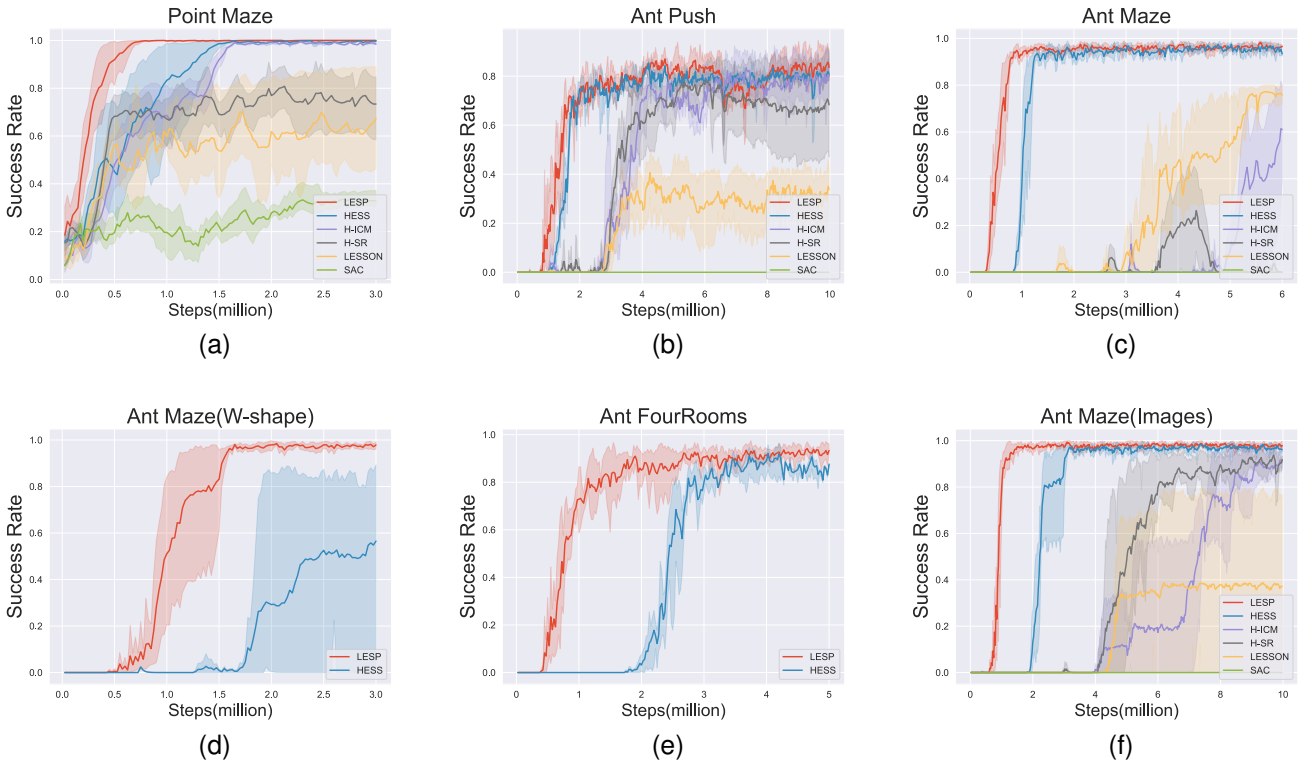

Fig. 4. Learning curves of LESP and baselines on all environments. (a) Point Maze. (b) Ant Maze. (c) Ant Maze (W-shape). (d) Ant Push. (e) Ant FourRooms. (f) Ant Maze (Images). The x-axis represents the training time steps, while the y-axis represents the average success rate over 50 episodes. The experiments are evaluated for each algorithm using five different random seeds. The shaded area represents the 95% confidence interval.

captures the potential impact of subgoals on the agent's ability to reach the ultimate goal. As depicted in Figure 5, after considering both novelty and prospect measures comprehensively, the exploration strategy tends to prioritize subgoals with high Prospect. Meantime, the exploration strategy also maintains the ability to explore unknown region, thereby striking a balance between exploiting promising subgoals and continuing to explore uncharted areas. After training 300,000 time steps, the agent has made progress and could reach positions closer to the target. Furthermore, as the buffer expands, the novelty measure based on counting becomes more accurate. Subgoals with high novelty tend to concentrate in directions where exploration is insufficiently conducted.

D. Ablation Studies

Ablative analysis of various components: In order to investigate the importance of different components in LESP, we conduct an ablation study on the proposed exploration strategy. We evaluate the performance of the algorithm under various experimental settings, including (i) the original exploration strategy (LESP) proposed in our work. (ii) LESP without stable value function, which means employing Equation 2 as the loss function to update the parameters of the value network in the low-level policy, without incorporating state-specific regularization. (iii) LESP without prospect, which means that the weight coefficient in Equation 6 is set to 0. (iv) replace prospect with potential, which means the prospect measure is replaced with the potential measure used in HESS and (v)HESS, state-of-the-art active exploration method. We conduct experiments in three different tasks: AntMaze (Images), AntMaze (W-shape), and AntFourRooms. The experimental results are depicted in Figure 6. As observed in Figure 6, replacing the prospect measure with potential leads to a significant decline in the performance of the exploration strategy. This highlights the superiority of the prospect measure over potential. While potential measure encourages the agent to expand the exploration area, prospect effectively guides the agent towards regions that positively contribute to accomplishing the task. In the experimental setting without prospect, the algorithm performs poorly across multiple complex tasks. This emphasizes the crucial role of the prospect measure in the exploration process. LESP demonstrates better performance compared to the setting without a stable value function, especially in the AntMaze (W-shape) task, LESP discovers effective trajectories at an earlier stage. Furthermore, even after replacing prospect with potential, the performance of the exploration strategy remains superior to that of HESS. This further highlights the importance of stable learning of the low-level state-action value function in promoting the stability of hierarchical policies.

Ablation studies on the hyper-parameters selection: To evaluate the effectiveness of the hyper-parameters, including the number of landmark samples (n_{cov}), the balance coefficient (α), and the low-level policy length (c), we conduct an ablation study. All tests are performed in the challenging Ant FourRooms task, and the experimental results are presented in

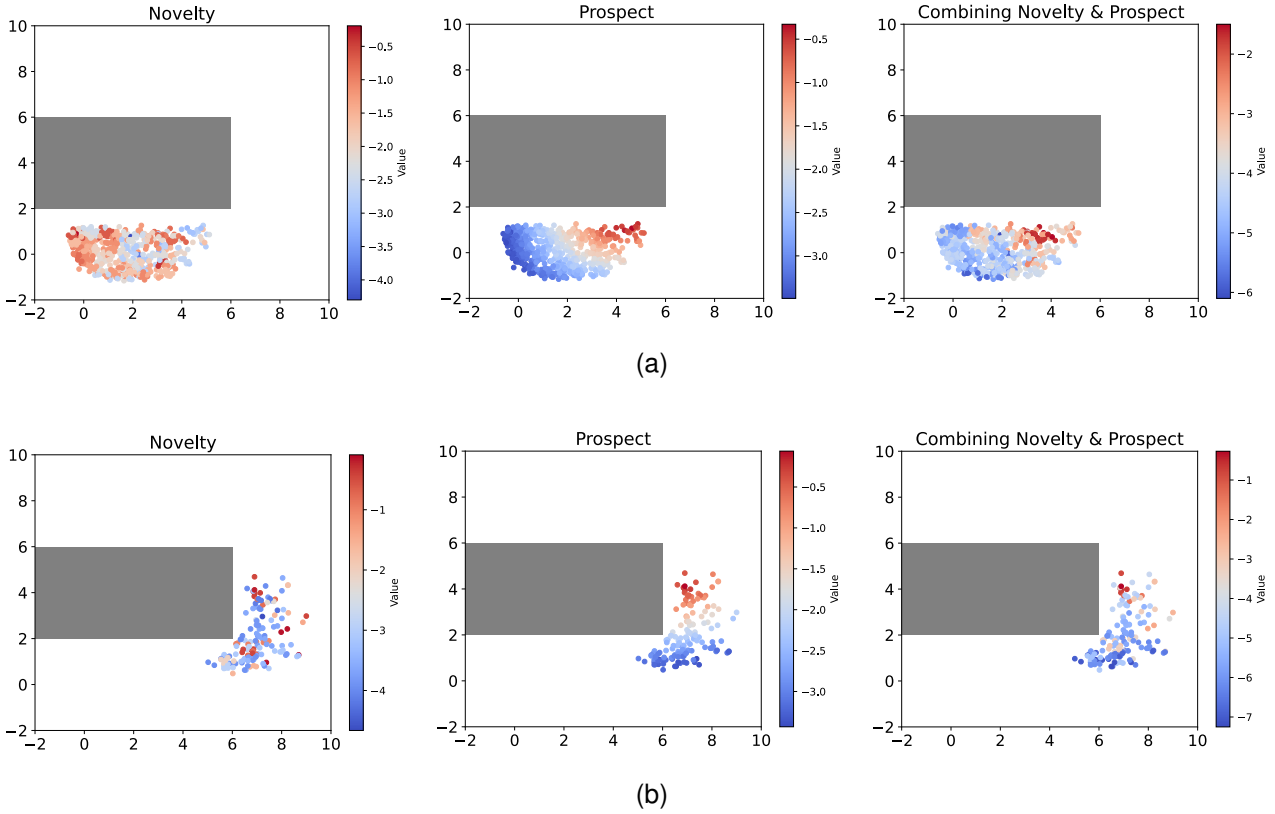

Fig. 5. The visualization of subgoal measures in the AntMaze task. (a) visualization at time steps 200000. (b) visualization at time steps 300000. The circular markers represent the candidate subgoal set sampled by the agent. The color intensity of the markers, ranging from red to blue, indicates the corresponding measure values.

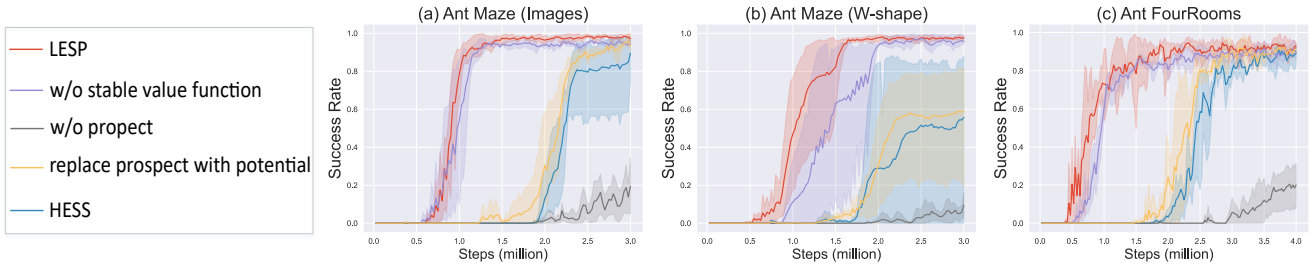

Fig. 6. Ablation studies on the components of proposed exploration strategy. In the conducted ablation experiments, three task scenarios are examined, namely: (a) Ant Maze (Images). (b) Ant Maze (W-shape). (c) Ant FourRooms. For each of these tasks, the experiments are evaluated using five different random seeds.

Figure 7.

Balance coefficient α: In the proposed exploration strategy, the balance coefficient α is used to balance the importance of novelty and prospect measure. A larger α value indicates that the exploration strategy prioritizes subgoals that have a more positive impact on guiding the agent towards the goal, while potentially diminishing the ability to explore new states. In complex scenarios, choosing an appropriate balance coefficient is crucial. From Figure 7 (a), it can be observed that larger values of α in a reasonable range tend to yield better results. For all the experiments in Section 5.2, we set α to 0.1

Number of landmark samples n_{cov}: LESP utilizes the FPS algorithm to sample n_{cov} candidate landmarks from the initial set of landmarks. It then selects the nearest landmark on the planned trajectory as the l_{sel} (selected landmark). The number of sampled landmarks, n_{cov}, influences the efficiency of trajectory planning. If n_{cov} is too small, it may result in the inability to plan feasible path to the goal. On the other hand, if n_{cov} is too large, the selected l_{sel} may be too close to the current state s_t. This can hinder the effectiveness of prospect in guiding the exploration process. So selecting an appropriate n_{cov} is indeed crucial for choosing subgoals that provide meaningful guidance during exploration. In the experiments conducted in Section 5.2, for the Ant FourRooms task, we set n_{cov} to 60, indicating a larger number of candidate landmarks to ensure comprehensive coverage of the environment. For the Ant Maze (W-shape) task, n_{cov} is set to 40. For other tasks, n_{cov} is set to 20.

Low-level policy length c: Hierarchical reinforcement learning (HRL) decomposes long-horizon tasks into sub-tasks

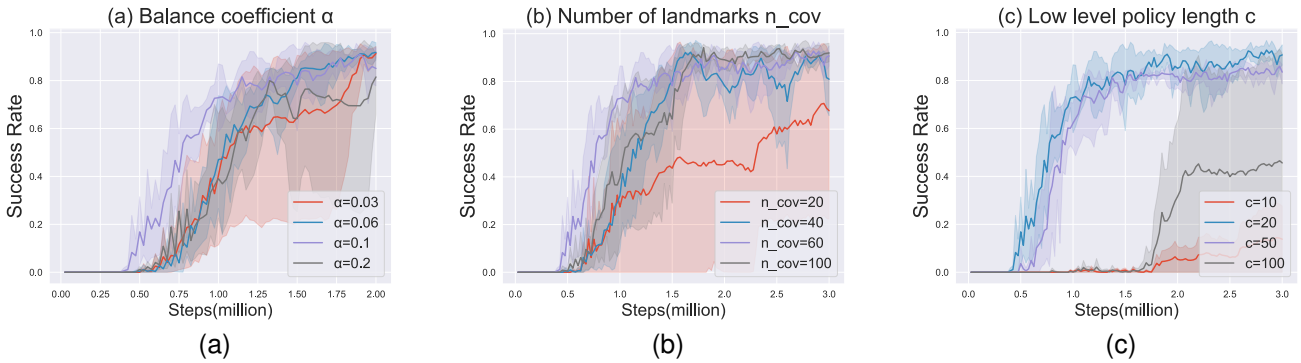

Fig. 7. Ablation studies on the hyper-parameters. (a) Balance coefficient α. (b) Number of landmarks n_{cov}. (c) Low level policy length c. All the ablation experiments regarding hyper-parameters are conducted specifically on the Ant FourRooms task

with finite horizon, denoted as c. Selecting an appropriate value for c is advantageous as it allows for proper allocation of task difficulty between the high-level policy and low-level policy. In active exploration approaches, the selected sub-goals are fed to the low-level policy with a finite horizon. Therefore, selecting an inappropriate value for c can result in the selected sub-goals being unable to effectively guide exploration, leading to the failure of the active exploration strategy. The experimental results are shown in Figure 7 (c). For all the experiments in Section 5.2,we set c to 20.

VI. Conclusion

Goal-conditioned hierarchical reinforcement learning (HRL) is a paradigm that effectively addresses complex long-horizon problems. However, improving sample efficiency is a crucial and yet unresolved issue in reinforcement learning. To tackle challenging sparse reward tasks, we proposed an active exploration strategy LESP, which takes into account prospect and novelty measures for subgoals. We designed a prospect measure for subgoals in LESP. LESP generated promising trajectories by planning landmarks in the goal space. It then computed prospect measures for subgoals based on the selected landmark. Unlike HIGL, which guided exploration by constraining the high-level policy, LESP sampled subgoals in the vicinity of the current state to guild exploration. This active exploration approach avoided introducing additional non-stationarity to the high-level policy. In addition, to mitigate the impact of dynamic changes in the low-level policy on the high-level state transitions, we incorporated state-specific regularization into the training of the low-level policy. Experimental results demonstrated that LESP outperformed the state-of-the-art approach HESS. Additionally, some people may point out that LESP requires extra buffer to store samples of prior state space, as well as a goal-condition value function that reflects the reachability between two states. Our view on this is that, since an expert policy is not required for randomly walk in the environment and the goal-conditioned value function only focuses on the neighboring states, the additional computation cost brought by LESP is acceptable. For some complex, long-term tasks such as robot navigation and robotic arm control, LESP is of significant importance for efficient interaction between agents and the environment.

References

[1] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[2] V. Micheli, E. Alonso, and F. Fleuret, "Transformers are sample efficient world models," *arXiv Preprint arXiv:2209.00588*, 2022.

[3] L. Kästner, X. Zhao, T. Buiyan, J. Li, Z. Shen, J. Lambrecht, and C. Marx, "Connecting deep-reinforcement-learning-based obstacle avoidance with conventional global planners using waypoint generators," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1213–1220.

[4] D. Dugas, J. Nieto, R. Siegwart, and J. J. Chung, "Navrep: Unsupervised representations for reinforcement learning of robot navigation in dynamic human environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7829–7835.

[5] H. Li, X. Yang, Y. Yang, S. Mei, and Z. Zhang, "Memonav: Selecting informative memories for visual navigation," *arXiv Preprint arXiv:2208.09610*, 2022.

[6] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 5, 1992, pp. 271–178.

[7] O. Nachum, S. Gu, H. Lee, and S. Levine, "Near-optimal representation learning for hierarchical reinforcement learning," *arXiv Preprint arXiv:1810.01257*, 2018.

[8] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *Advances in Neural Information Processing Systems*, vol. 29, 2016, pp. 1–9.

[9] V. H. Wang, J. Pajarinen, T. Wang, and J.-K. Kämäräinen, "Hierarchical reinforcement learning with adversarially guided subgoals." *CoRR*, 2022.

[10] T. Zhang, S. Guo, T. Tan, X. Hu, and F. Chen, "Generating adjacency-constrained subgoals in hierarchical reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21579–21590.

[11] J. Kim, Y. Seo, and J. Shin, "Landmark-guided subgoal generation in hierarchical reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 28336–28349.

[12] A. Péré, S. Forestier, O. Sigaud, and P.-y. Oudeyer, "Unsupervised learning of goal spaces for intrinsically motivated goal exploration," in *International Conference on Learning Representations*, 2018, pp. 1–26.

[13] S. Li, L. Zheng, J. Wang, and C. Zhang, "Learning subgoal representations with slow dynamics," in *International Conference on Learning Representations*, 2021, pp. 1–18.

[14] S. Sukhbaatar, E. Denton, A. Szlam, and R. Fergus, "Learning goal embeddings via self-play for hierarchical reinforcement learning," *arXiv Preprint arXiv:1811.09083*, 2018.

[15] S. Li, J. Zhang, J. Wang, Y. Yu, and C. Zhang, "Active hierarchical exploration with stable subgoal representation learning," in *International Conference on Learning Representations*, 2022, pp. 1–18.

[16] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 1–11.

[17] A. Levy, G. Konidaris, R. Platt, and K. Saenko, "Learning multi-level hierarchies with hindsight," in *International Conference on Learning Representations*, 2019, pp. 1–15.

[18] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[19] M. Eppe, C. Gumbsch, M. Kerzel, P. D. Nguyen, M. V. Butz, and S. Wermter, "Intelligent problem-solving as integrated hierarchical reinforcement learning," *Nature Machine Intelligence*, vol. 4, no. 1, pp. 11–20, 2022.

[20] G. Wan, S. Pan, C. Gong, C. Zhou, and G. Haffari, "Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning," in *Proceedings of the Twenty-ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 1926–1932.

[21] Y. Takubo, H. Chen, and K. Ho, "Hierarchical reinforcement learning framework for stochastic spaceflight campaign design," *Journal of Spacecraft and rockets*, vol. 59, no. 2, pp. 421–433, 2022.

[22] M. Rohmatillah and J.-T. Chien, "Hierarchical reinforcement learning with guidance for multi-domain dialogue policy," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 748–761, 2023.

[23] S. Notter, F. Schimpf, G. Müller, and W. Fichter, "Hierarchical reinforcement learning approach for autonomous cross-country soaring," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 1, pp. 114–126, 2023.

[24] K. Gregor, D. J. Rezende, and D. Wierstra, "Variational intrinsic control," *arXiv Preprint arXiv:1611.07507*, 2016.

[25] R. Fox, S. Krishnan, I. Stoica, and K. Goldberg, "Multi-level discovery of deep options," *arXiv Preprint arXiv:1703.08294*, 2017.

[26] J. Duan, S. Eben Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020.

[27] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *International Conference on Machine Learning*, 2017, pp. 3540–3549.

[28] D. Ghosh, A. Gupta, and S. Levine, "Learning actionable representations with goal conditioned policies," in *International Conference on Learning Representations*, 2019, pp. 1–18.

[29] V. Pong, S. Gu, M. Dalal, and S. Levine, "Temporal difference models: Model-free deep rl for model-based control," in *International Conference on Learning Representations*, 2018, pp. 1–14.

[30] S. Nasiriany, V. Pong, S. Lin, and S. Levine, "Planning with goal-conditioned policies," *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 1–12.

[31] E. Chane-Sane, C. Schmid, and I. Laptev, "Goal-conditioned reinforcement learning with imagined subgoals," in *International Conference on Machine Learning*, 2021, pp. 1430–1440.

[32] L. Zhang, G. Yang, and B. C. Stadie, "World model as a graph: Learning latent landmarks for planning," in *International Conference on Machine Learning*, 2021, pp. 12611–12620.

[33] J. Li, C. Tang, M. Tomizuka, and W. Zhan, "Hierarchical planning through goal-conditioned offline reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10216–10223, 2022.

[34] L. Kästner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, "Arena-rosnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6456–6463.

[35] L. Kästner, X. Zhao, Z. Shen, and J. Lambrecht, "Obstacle-aware waypoint generation for long-range guidance of deep-reinforcement-learning-based navigation approaches," *arXiv Preprint arXiv:2109.11639*, 2021.

[36] R. S. Sutton and A. G. Barto, *Reinforcement earning: An introduction*. MIT press, 2018.

[37] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016, pp. 2094–2100.

[38] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1995–2003.

[39] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv Preprint arXiv:1812.05905*, 2018.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv Preprint arXiv:1707.06347*, 2017.

[41] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[42] Z. Huang, F. Liu, and H. Su, "Mapping state space using landmarks for universal goal reaching," *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 1942–1952.

[43] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035.