Landmark-Guided Subgoal Generation in Hierarchical Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Goal-conditioned hierarchical reinforcement learning (HRL) has shown promising results for solving complex and long-horizon RL tasks. However, the action space of high-level policy in the goal-conditioned HRL is often large, so it results in poor exploration, leading to inefficiency in training. In this paper, we present HIerarchical reinforcement learning Guided by Landmarks (HIGL), a novel framework for training a high-level policy with a reduced action space guided by *landmarks*, i.e., promising states to explore. The key component of HIGL is twofold: (a) sampling landmarks that are informative for exploration and (b) encouraging the high-level policy to generate a subgoal towards a selected landmark. For (a), we consider two criteria: coverage of the entire visited state space (i.e., dispersion of states) and novelty of states (i.e., prediction error of a state). For (b), we select a landmark as the very first landmark in the shortest path in a graph whose nodes are landmarks. Our experiments demonstrate that our framework outperforms prior-arts across a variety of control tasks, thanks to efficient exploration guided by landmarks.

1 Introduction

Deep reinforcement learning (RL) has demonstrated wide success in a variety of sequential decision-making problems, i.e., board games [27, 29], video games [1, 18, 27], and robotic control tasks [11, 21, 36]. However, solving complex and long-horizon tasks has still remained a major challenge in RL, where hierarchical reinforcement learning (HRL) provides a promising direction by enabling control at multiple time scales via a hierarchical structure. Among HRL frameworks, goal-conditioned HRL has long been recognized as an effective paradigm [3, 15, 20, 26], showing significant success in a variety of long and complex tasks, e.g., navigation with locomotion [16, 37]. The framework comprises a high-level policy and a low-level policy; the former breaks the original task into a series of subgoals, and the latter aims to reach those subgoals.

The effectiveness of goal-conditioned HRL depends on the acquisition of effective and semantically meaningful subgoals. To this end, several strategies have been proposed, e.g., learning the subgoal representation space [4, 9, 16, 20, 22–24, 30, 34] or utilizing domain-specific knowledge for pre-defining subgoal space [19, 37]. However, the learned or pre-defined subgoal space is often too large, which results in poor high-level exploration, leading to inefficient training. To address this issue, Zhang et al. [37] recently proposed to reduce the high-level action space into the k-step adjacency region around the current state. This approach, however, is limited in that it considers all the states within the k-step adjacency region equally as the candidate actions for the high-level policy, without considering the *novelty* of states, which is crucial for exploration.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

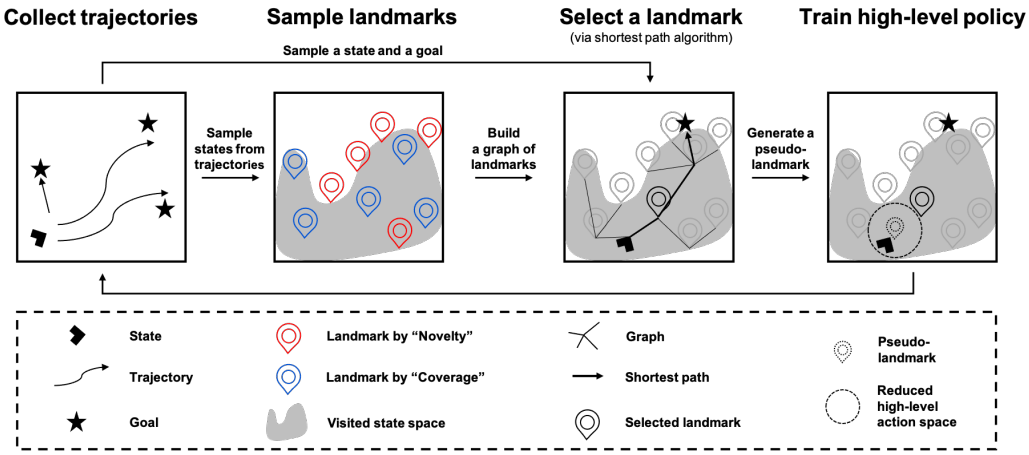


Figure 1: Illustration of **HI**erarchical reinforcement learning **G**uided by **L**andmarks (HIGL). (1) We collect trajectories using a high-level and a low-level policy. (2) Sample landmarks from visited states based on "Coverage" and "Novelty" criteria, respectively, and merge them. (3) Select a single landmark among the sampled landmarks in a graph constructed by landmarks, a goal, and a current state. (i.e., select the very first landmark in the shortest path to the goal). (4) Train a high-level policy to generate a subgoal toward the selected landmark.

Contribution. In this paper, we present **HI**erarchical reinforcement learning **G**uided by **L**andmarks (HIGL), a novel framework for training a high-level policy that generates a subgoal toward *landmarks*, i.e., promising states to explore. HIGL consists of the following key ingredients (see Figure 1):

- **Landmark sampling:** To effectively sample landmarks that represent promising states to explore, it is important to sample landmarks that cover a wide area of state space and contain novel states. To this end, we propose two sampling schemes: (a) coverage-based sampling scheme that samples located as far away from each other as possible and (b) novelty-based sampling scheme that stores novel states encountered during training and utilizes them as landmarks. We find that our method successfully samples diverse and novel landmarks.
- **Landmark-guided subgoal generation:** Among the sampled landmarks, we select the most *urgent* landmark by our landmark selection scheme with the shortest path planning algorithm. Then we propose to *shift* the action space of a high-level policy toward a selected landmark. Because HIGL constructs a high-level action space that is both (a) reachable from the current state and (b) shifted towards a promising landmark state, we find that the proposed method can effectively guide the subgoal generation of a high-level policy.

We demonstrate the effectiveness of HIGL on various long-horizon continuous control tasks based on MuJoCo simulator [32], which is widely-used in the HRL literature [7, 12, 19, 20]. In our experiments, HIGL significantly outperforms the prior state-of-the-art method, i.e., HRAC [37], especially in complex environments with sparse reward signals. For example, HIGL achieves the success rate of 65.1% in Ant Maze (sparse) environment, where HRAC only achieves 17.6%.

2 Related work

Goal-conditioned HRL. By introducing a hierarchy consisting of a high-level policy and a low-level policy, goal-conditioned HRL has been successful in a wide range of tasks. Notably, Nachum et al. [19] proposed an off-policy correction method for goal-conditioned HRL, and Levy et al. [14] successfully trained a policy with multiple hierarchies. However, efficient training of a high-level policy still remains a challenge. Several approaches have been proposed to address this problem, including learning-based approaches [4, 9, 16, 20, 22–24, 30, 34] and approaches that utilize a domain knowledge [19, 37]. The work closest to ours is Zhang et al. [37] that reduces the high-level action space to k-adjacent region of the current state. Our work differs in that we explicitly consider the *novelty* of each state within the region instead of treating all states in an equal manner.

Planning on an environmental graph. This line of works constructs an environmental graph and offers a reachable subgoal for a goal-conditioned policy via planning in the graph [6, 10, 13, 17, 25, 28, 35]. Recent advances in this field focus on building a 'good' graph representing the entire map with a limited number of nodes, i.e., farthest point sampling [10] and sparsification [13]. Typically, an weight of an edge is set to an estimated distance between two nodes; the distance can be calculated by the goal-conditioned value function [6, 10]. After building a graph, they run a (traditional) planning algorithm, e.g., the Bellman-Ford algorithm, to provide a subgoal for a policy. These works have shown promising results in complex RL tasks but have a scalability issue due to increasing planning time with a larger map. Meanwhile, our framework differs from this line of works since ours 'train' high-level policy rather than hard-coded high-level planning to generate a subgoal.

3 Preliminaries

We formulate a control task with a finite-horizon, goal-conditioned Markov decision process (MDP) [31] defined as a tuple $(\mathcal{S}, \mathcal{G}, \mathcal{A}, p, r, \gamma, H)$, where $\mathcal{S}$ is the state space, $\mathcal{G}$ is the goal space, $\mathcal{A}$ is the action space, $p\left(s'|s,a\right)$ is the transition dynamics, $r\left(s,a\right)$ is the reward function, $\gamma \in [0,1)$ is the discount factor, and H is the horizon.

Goal-conditioned HRL. We consider a framework that consists of two hierarchies: a high-level policy $\pi(g|s;\theta_{\texttt{high}})$ and a low-level policy $\pi(a|s,g;\theta_{\texttt{low}})$, where each policy is modeled as a Gaussian with mean and covariance given by neural networks whose parameters are $\theta_{\texttt{high}}$ and $\theta_{\texttt{low}}$, respectively. At each timestep t, The high-level policy generates a high-level action, i.e., subgoal $g_t \in \mathcal{G}$, by either sampling from its policy $g_t \sim \pi(g|s_t;\theta_{\texttt{high}})$ when $t \equiv 0 \pmod{k}$, or otherwise using a pre-defined goal transition process $g_t = h(g_{t-1}, s_{t-1}, s_t)$. The low-level policy observes the state s_t and goal g_t, and performs a low-level atomic action $a_t \sim \pi(a|s_t, g_t;\theta_{\texttt{low}})$. Then, the reward function for the high-level policy is given as the sum of m external rewards from the environment as follows:

$$r^{\texttt{high}}(\tau_{t,m}) = \sum_{i=0}^{m-1} r(s_i, a_i), \tag{1}$$

where $\tau_{t,m} = \{(s_t, g_t, a_t), \cdots, (s_{t+m-1}, g_{t+m-1}, a_{t+m-1})\}$ denotes a trajectory segment of size m. The goal of a high-level policy is to maximize the expected sum of $r^{\texttt{high}}$ by providing the low-level policy with an intrinsic reward proportional to the distance in the goal space $\mathcal{G}$. Specifically, the reward function for a low-level policy is defined as:

$$r^{\texttt{low}}(s_t, g_t, a_t, s_{t+1}) = -\|g_t, \varphi(s_{t+1} - s_t)\|_2, \tag{2}$$

where $\varphi : \mathcal{S} \to \mathcal{G}$ is a goal mapping function that maps a state to a goal.

Adjacency network. Let $d_{\texttt{st}}(s, s')$ be the shortest transition distance from state s to state s', i.e., $d_{\texttt{st}}(s, s')$ is the expected number of steps an optimal agent should take to reach the state s' from s. To estimate $d_{\texttt{st}}$, Zhang et al. [37] introduce an *adjacency network* ψ parameterized by ϕ, that discriminates whether two states are k-step adjacent or not. The network is trained to learn a mapping from a goal space to an adjacency space by minimizing the following contrastive-like loss:

$$\begin{aligned}\mathcal{L}_{\texttt{adj}}(\phi) = \mathbb{E}_{s_i, s_j \in \mathcal{S}}[&l \cdot \max(\|\psi_\phi(g_i) - \psi_\phi(g_j)\|_2 - \varepsilon_k, 0) \\ &+ (1-l) \cdot \max(\varepsilon_k + \delta - ||\psi_\phi(g_i) - \psi_\phi(g_j)||_2, 0)],\end{aligned} \tag{3}$$

where $\delta > 0$ is a margin between embeddings, ε_k is a scaling factor, and $l \in \{0, 1\}$ represents the label indicating k-step adjacency derived from the k-step adjacency matrix $\mathcal{M}$ that stores the adjacency information of the explored states. Then the adjacency network can be used to estimate the shortest transition distance as follows:

$$\widehat{d}_{\texttt{st}}(s, s'; \phi) = \frac{k}{\varepsilon_k} ||\psi_\phi(g_1), \psi_\phi(g_2)||_2 \approx d_{\texttt{st}}(s, s'). \tag{4}$$

Random network distillation. Random Network Distillation (RND) [2] is an exploration method that utilizes the prediction error of a neural network as a novelty score. Specifically, let f be a neural network with fixed parameters $\bar{\theta}$ and $\hat{f}$ be a neural network parameterized by θ. RND updates θ by minimizing the expected mean squared prediction error of f, $\mathbb{E}_{s \sim \mathcal{B}} \|\hat{f}(s;\theta) - f(s)\|_2$. Then the novelty score of a state s is defined as:

$$n(s) = \|\hat{f}(s;\theta) - f(s)\|_2. \tag{5}$$

4 Hierarchical reinforcement learning guided by landmarks (HIGL)

In this section, we propose HIGL: **HI**earchical reinforcement learning **G**uided by **L**andmarks, a novel framework for training a high-level policy with reduced action space guided by landmarks. We describe HIGL with three parts sequentially: (1) landmark sampling in Section 4.1, (2) landmark selection in Section 4.2, and (3) training in Section 4.3. We provide an illustration and an overall description of our framework in Figure 1 and Algorithm 1, respectively.

4.1 Landmark sampling

To effectively guide the subgoal generation of a high-level policy, it is important to construct a set of landmarks that covers a wide area of a state space and contains novel states promising to explore. To this end, we consider two criteria for landmark selection: (1) the coverage of the entire visited state space and (2) the novelty of a state.

Coverage-based sampling. We propose to sample landmarks that cover a wide range of visited states from a replay buffer $\mathcal{B}$. To this end, we utilize Farthest Point Sampling (FPS) [33], which samples a *pool* of states from $\mathcal{B}$ and chooses states which are as far as possible from each other in the pool. Specifically, we sample a set of *coverage-based* landmarks $L^{\texttt{cov}} = \{l_i^{\texttt{cov}}\}_{i=1}^{M_{\texttt{cov}}}$ by applying FPS where the distance between two states s and s' is measured in the goal space as $||\varphi(s) - \varphi(s')||_2$, following Huang et al. [10]. We note that FPS implicitly samples states at the frontier of visited state space, which implies that coverage-based landmarks implicitly mean promising states to explore as well (see Figure 6 for supporting experimental results).

Novelty-based sampling. To explicitly sample novel landmarks, we propose to store the novel states encountered during the environment interaction and utilize them as landmarks. To this end, we introduce a priority queue $\mathcal{Q}$ of a fixed size K where the priority of each element (state) is defined as the novelty of a state $n(s)$ in (5), and store a state s in $\mathcal{Q}$ with a priority of $n(s)$. One important here is that the novelty of a state s decreases as the exploration proceeds, so the priority of stored states in $\mathcal{Q}$ should be constantly updated. For this reason, we propose a similarity-based update scheme that discards previously stored samples that are similar to the newly encountered state. Specifically, when we encounter a state s, we measure the similarity of s between all stored states $s' \in \mathcal{Q}$ and discard similar states, i.e., $\{s' \in \mathcal{Q} : ||\varphi(s) - \varphi(s')||_2 < \lambda\}$, where λ is a similarity threshold. Then we store a state s in $\mathcal{Q}$, and sample a set of *novelty-based* landmarks $L^{\texttt{nov}} = \{l_i^{\texttt{nov}}\}_{i=1}^{M_{\texttt{nov}}}$ from $\mathcal{Q}$.

4.2 Landmark selection

Since all the landmarks in $L = L^{\texttt{cov}} \cup L^{\texttt{nov}}$ are not equally valuable to reach a goal g from a current state s, i.e., some landmarks may be irrelevant or even impeditive to arrive at the goal, we propose a landmark selection scheme to select the most *urgent* landmark among the sampled landmarks. To this end, we introduce a two-stage scheme: (a) we first build a graph of landmarks, and (b) we run the shortest path planning to a goal in the graph.

Building a graph. For landmark selection, we build a graph whose nodes consist of a current state s_t, a (final) goal g, and landmarks L. First, we connect all the nodes and assign the weight of each edge with a distance between two nodes, where distance is estimated by a low-level (goal-conditioned) value function, i.e., $-V(s_1, \varphi(s_2 - s_1))$ for state s_1, s_2, following prior works [6, 10, 23]. As the distance estimation via value function is locally accurate but unreliable for far states, we disconnect two nodes when the weight of corresponding edge is larger than a preset threshold $\gamma_{\texttt{dist}}$ following Huang et al. [10].

Planning. After building a graph, we run the shortest path planning algorithm to select the most urgent state to visit, $l_t^{\texttt{sel}}$, from a current state s_t to a goal g. Since the general value iteration for RL problems is exactly the shortest path algorithm on the graph, we utilize the value iteration as the shortest path planning, following the prior work of Huang et al. [10]. By sampling promising landmarks and selecting the most urgent landmark, our method can effectively find the most useful landmark from the high-dimensional goal space.

Algorithm 1 Hierarchical reinforcement learning guided by landmarks (HIGL)

Input: Goal transition function h, state-goal mapping function φ, high-level action frequency m, the number of training episode N, adjacency learning frequency C, replay buffer $\mathcal{B}$, training batch size B and the number of landmarks $M_{\texttt{cov}}$, $M_{\texttt{nov}}$
Initialize the parameters of high-level policy $\theta_{\texttt{high}}$, low-level policy $\theta_{\texttt{low}}$, adjacency network ϕ.
Initialize empty adjacency matrix $\mathcal{M}$
Initialize priority queue $\mathcal{Q}$
for $n = 1, \ldots, N$ **do**
  Reset the environment and sample the initial state s_0.
  $t = 0$.
  repeat
    if $t \equiv 0 \pmod{m}$ **then**
      Sample subgoal $g_t \sim \pi(g|s_t; \theta_{\texttt{high}})$.
    else
      Perform subgoal transition $g_t = h(g_{t-1}, s_{t-1}, s_t)$
    end if
    Collect a transition (s_t, a_t, s_{t+1}, r_t) using low-level policy $\theta_{\texttt{low}}$.
    Calculate novelty of the state s_t, and update the priority queue $\mathcal{Q}$.
    Sample episode end signal $done$.
    $t = t + 1$
  until $done$ is $true$
  Store the sampled trajectory in $\mathcal{B}$.
  for $j = 1, \ldots, B$ **do**
    Sample a state and a corresponding goal from $\mathcal{B}$.
    Sample $M_{\texttt{cov}}$ landmarks from $\mathcal{B}$ and $M_{\texttt{nov}}$ landmarks from $\mathcal{Q}$, and merge them.
    Build a graph with the sampled landmarks, a state and a goal.
    Select a landmark in the graph. (i.e., the very first landmark in the shortest path to a goal.)
    Train high-level policy $\theta_{\texttt{high}}$ according to equation 8.
    Train low-level policy $\theta_{\texttt{low}}$.
  end for
  if $n \equiv 0 \pmod{C}$ **then**
    Update the adjacency matrix $\mathcal{M}$ using trajectories in $\mathcal{B}$.
    Train ϕ using $\mathcal{M}$ by minimizing equation 3.
  end if
end for

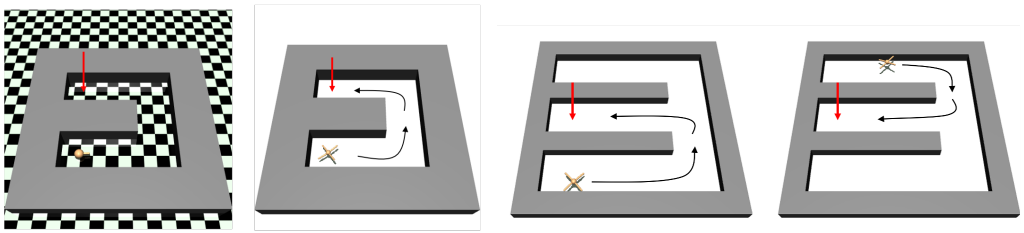
4.3 Training high-level policy guided by landmark

To effectively guide the subgoal generation of a high-level policy, we propose to *shift* the high-level action action space toward the selected landmark. Our intuition is that our scheme allows for efficient exploration by enabling to generate a subgoal within a meaningful high-level action space which is (1) reachable from the current state and (2) directed toward the urgent landmark. Specifically, we generate a *pseudo*-landmark $l_t^{\texttt{pseudo}}$ whose corresponding goal $g_t^{\texttt{pseudo}}$ is located between $\varphi(s_t)$ and $\varphi(l_t^{\texttt{sel}})$ as follows:

$$\varphi(l_t^{\texttt{pseudo}}) = g_t^{\texttt{pseudo}} := g_t^{\texttt{cur}} + \delta_{\texttt{pseudo}} \cdot \frac{g_t^{\texttt{sel}} - g_t^{\texttt{cur}}}{||g_t^{\texttt{sel}} - g_t^{\texttt{cur}}||_2}, \tag{6}$$

where $\delta_{\texttt{pseudo}}$ is the shift magnitude, $g_t^{\texttt{cur}} = \varphi(s_t)$, and $g_t^{\texttt{sel}} = \varphi(l_t^{\texttt{sel}})$. Then, the high-level policy is encouraged to generate a subgoal within k-step adjacent region to the pseudo-landmark $l_t^{\texttt{pseudo}}$. To discriminate whether generated subgoal is k-step adjacent to the pseudo-landmark, we employ the adjacent network proposed in Zhang et al. [37]. Instead of strict adjacency constraint, which may cause instability in training, we encourage high-level policy to generate a subgoal near pseudo-landmark via *landmark loss*, motivated by the prior work [37]. The landmark loss is calculated using the adjacency network as follows:

$$\mathcal{L}_{\texttt{landmark}}(\theta_{\texttt{high}}) = \max(||\psi_\phi(g_t^{\texttt{pseudo}}) - \psi_\phi(g_t)||_2 - \varepsilon_k, 0), \tag{7}$$



(a) Point Maze (b) Ant Maze (U-shape) (c) Ant Maze (W-shape)

Figure 2: Environments used in our experiments. The red arrow indicates the goal in each task. The black line represents a desired trajectory from the current state to the goal. In (a) Point Maze and (b) Ant Maze (U-shape) an agent is born at the bottom-left corner at the start of the episode. In (c) Ant Maze (W-shape), an agent is born at random point in the maze except the goal point.

where $g_t \sim \pi(g|s_t; \theta_{\texttt{high}})$ is a generated subgoal by the high-level policy. Then, we train high-level policy by incorporating $\mathcal{L}_{\texttt{landmark}}$ into the goal-conditioned HRL framework:

$$\mathcal{L}_{\texttt{high}}(\theta_{\texttt{high}}) = -\mathbb{E}_{\theta_{\texttt{high}}} \sum_{t=0}^{T-1} (\gamma^t r^{\texttt{high}}(\tau_{t,m}) - \eta \cdot \mathcal{L}_{\texttt{landmark}}), \tag{8}$$

where η is the balancing coefficient. In practice, we plug $\mathcal{L}_{\texttt{landmark}}$ as an extra loss term into the original policy loss term of a specific high-level RL algorithm, e.g., TD error for temporal-difference learning methods.

5 Experiments

In this section, we designed our experiments to answer the following questions:

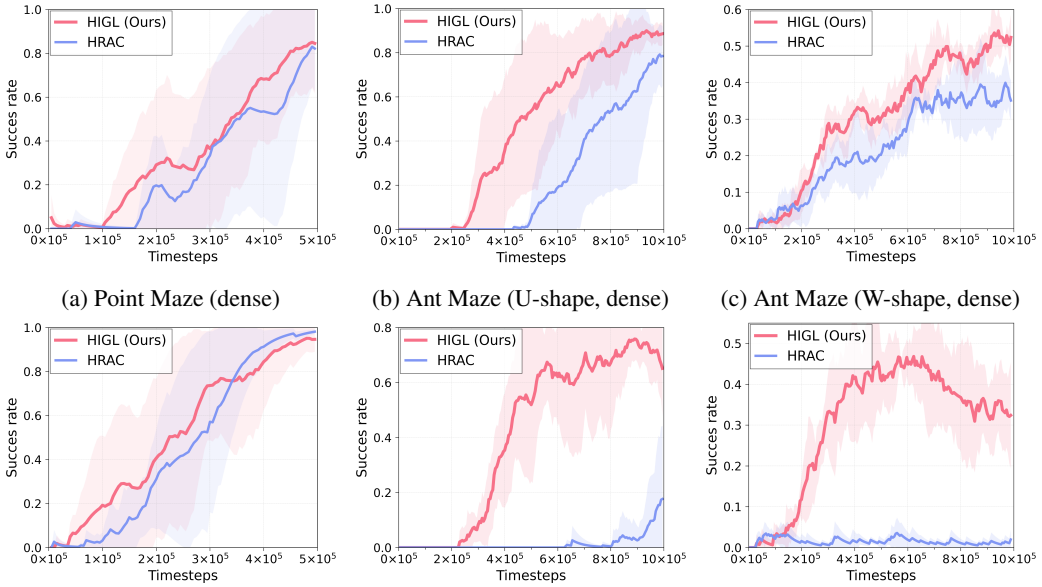
- How does our method compare to the state-of-the-art HRL method [37] across various long-horizon continuous control tasks (see Figure 3)?
- How does the number of landmarks affect the performance (see Figure 4a)?
- How does the shift magnitude $\delta_{\texttt{pseudo}}$ affect the performance (see Figure 4b)?
- How does the adjacency degree k affect the performance (see Figure 4c)?
- Does considering the novelty for landmark sampling improves the performance (see Figure 5)?

5.1 Experimental setup

Environments. We conduct our experiments on a set of challenging long-horizon continuous control tasks based on MuJoCo simulator [32]. Specifically, we consider the following three environments to evaluate our framework (see Figure 2 for the visualization of environments).

- **Point Maze** [5]: A simulated ball starts at the bottom left corner in a '$\supset$'-shaped maze and aims to reach the top left corner.
- **Ant Maze (U-shape)** [5]: A simulated ant starts at the bottom left corner in a '$\supset$'-shaped maze and aims to reach the top left corner.
- **Ant Maze (W-shape)** [37]: A simulated ant starts from a random position in the '$\exists$'-shaped maze and aims to reach the target position located at the middle left corner.

We evaluate HIGL on the three environments with two different reward shaping: *dense* and *sparse*. In dense reward shapings, the reward is the negative of L2 distance from the current (x, y) position to the goal. In the sparse setting, the reward is 0 if the distance to the goal is lower than pre-set threshold $\gamma_{\texttt{ext_rew}}$, otherwise -1. In all environments, we use a pre-defined 2-dimensional goal space that represents the (x, y) position of the agent following prior works [19, 37]. We provide more details of the environments in the supplementary material.



(a) Point Maze (dense) (b) Ant Maze (U-shape, dense) (c) Ant Maze (W-shape, dense)

(d) Point Maze (sparse) (e) Ant Maze (U-shape, sparse) (f) Ant Maze (W-shape, sparse)

Figure 3: The average success rate in various continuous control tasks of HIGL and HRAC [37]. We observe that HIGL consistently outperforms HRAC, especially in more complex and long-horizon environments. The solid lines and shaded regions represent mean and standard deviation, respectively, across four runs. All curves are smoothed for visual clarity.

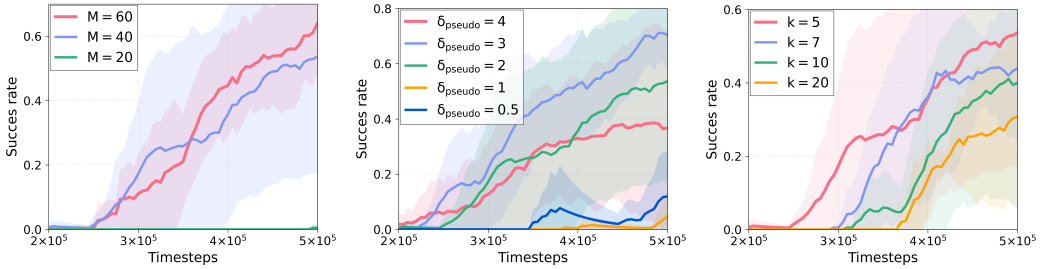
Implementation. We use TD3 algorithm [8] as the underlying algorithm for training both high-level policy and low-level policy for all considered methods. For the number of coverage-based landmarks $M_{\texttt{cov}}$ and the number of novelty-based landmarks $M_{\texttt{cov}}$, we use $M_{\texttt{cov}} = 20$ and $M_{\texttt{nov}} = 20$ in the Point Maze and Ant Maze (U-shape) environments, and $M_{\texttt{cov}} = 40$ and $M_{\texttt{nov}} = 40$ in the more complex Ant Maze (W-shape) environment. In order to avoid the instability in training due to the noisy pseudo-landmark in the early phase of training, we use $\delta_{\texttt{pseudo}} = 0$ for the initial 60K timesteps, i.e., k-step adjacent region to the current state instead of pseudo-landmark. We find that this stabilizes the training by avoiding the inaccurate planning at the early phase; under-trained value function causes inaccurate distance estimation at landmark selection. We provide all the hyperparameters used for the experiments in the supplementary materials with details. All of the experiments were processed using a single GPU (NVIDIA TITAN Xp) and 8 CPU cores (Intel Xeon E5-2630 v4).

5.2 Comparative evaluation

We compare HIGL to the prior state-of-the-art method HRAC, that encourages a high-level policy to generate a subgoal within the k-step adjacent region of the current state. We observe that HIGL outperforms HRAC significantly in difficult tasks such as Ant Maze (U-shape and W-shape). In particular, HIGL achieves a success rate of 65.1%, whereas HRAC performs about 17.6% at timesteps 10×10^5 in Ant Maze (U-shape, sparse) task. This demonstrates that efficient high-level exploration by landmarks is indeed effective. We also observe that HIGL is more sample-efficient when the task is much difficult; HIGL shows a larger margin in performance in (1) AntMaze (U-shape and W-shape) than PointMaze, and (2) sparse reward setting than dense reward setting.

5.3 Ablation studies

We conduct ablation studies on HIGL to investigate the effect of (1) the number of landmarks $M = M_{\texttt{cov}} + M_{\texttt{nov}}$, (2) shift magnitude $\delta_{\texttt{pseudo}}$, (3) adjacency degree k, and (4) novelty-based sampling in Figure 4 and Figure 5. For all the experiments for ablation studies, we employed the Ant Maze (U-shape, dense) task.



(a) Number of landmarks (b) Shift magnitude $\delta_{\texttt{pseudo}}$ (c) Adjacency degree k

Figure 4: Performance of HIGL on Ant Maze (dense) environment with varying (a) the number of landmarks, (b) shift magnitude, and (c) adjacency degree. The solid lines and shaded regions represent mean and standard deviation, respectively, across four runs. All curves are smoothed for visual clarity.

Number of landmarks. To demonstrate the effectiveness of number of landmarks, we conduct experiments using the different number of landmarks $M = M_{\texttt{cov}} + M_{\texttt{nov}}$ in Figure 4a. We sample same number of landmarks for each criteria, i.e., $M_{\texttt{cov}} = M_{\texttt{nov}}$. The results show that the performance of HIGL is improved with the increased number of landmarks since landmark selection becomes more reliable due to accurate planning in the landmark selection.

Shift magnitude $\delta_{\texttt{pseudo}}$. In Figure 4b, we conduct experiments with varying values of $\delta_{\texttt{pseudo}}$, which determines the location of pseudo-landmark. The results demonstrate the location of the pseudo landmark can affect the performance. If $\delta_{\texttt{pseudo}}$ is small, the high-level policy tends to be trained to generate a subgoal near the current state rather than the selected landmark; which may cause the high-level policy to not fully enjoy benefits of efficient exploration guided by the selected landmark. On the other hand, if $\delta_{\texttt{pseudo}}$ is too large, the high-level policy is promoted to generate a subgoal which is unreachable from the currents state, which leads to performance degradation, i.e., $\delta_{\texttt{pseudo}} = 4$ in Figure 4b.

Adjacency degree k. We investigate the effectiveness of the adjacency degree k, which determines the size of the region where high-level policy is encouraged to generate a subgoal in the Figure 4c. If the adjacency degree k is large, it is allowed for high-level policy to generate a subgoal, which is quite far from the pseudo landmark. As shown in the Figure 4c, if k is too large, performance degradation occur since an area which is far from the pseudo-landmark is uncertain to be a promising region to explore.

Novelty based sampling. To investigate whether if our proposed novelty sampling is effective, we evaluate HIGL with varying numbers of samples from novelty-based sampling in the Figure 5. Specifically, we report the performance of HIGL on AntMaze (U-shape, dense) environment with $M_{\texttt{cov}} = 20$ and varying $M_{\texttt{nov}} \in \{0, 10, 20\}$. We observe that utilizing our proposed novelty-based sampling improves the sample-efficiency. For example, we emphasize that HIGL with $M_{\texttt{nov}} = 10$ novelty-based landmarks significantly improves over $M_{\texttt{nov}} = 0$, which corresponds to HIGL with only coverage-based sampling. This demonstrates the importance of considering the novelty of each state is crucial for efficient exploration.

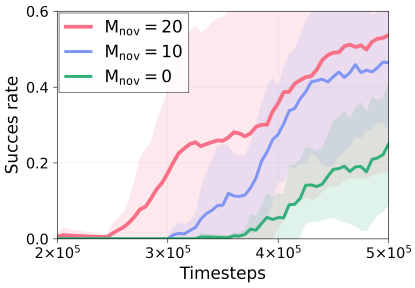


Figure 5: Performance of HIGL on Ant Maze (U-shape, dense) environment with varying novelty-based landmarks.

5.4 Qualitative analysis on landmark sampling

In Figure 6, we qualitatively analyze how our landmarks sampling method works in the Ant Maze (U-shape, dense) task. We sample 20 coverage-based landmarks (blue dots) and 20 novelty-based landmarks (red dots). As shown in the figure, we verify that coverage-based landmarks are dispersed across visited space, and novelty-based landmarks are concentrated to the frontier of the visited

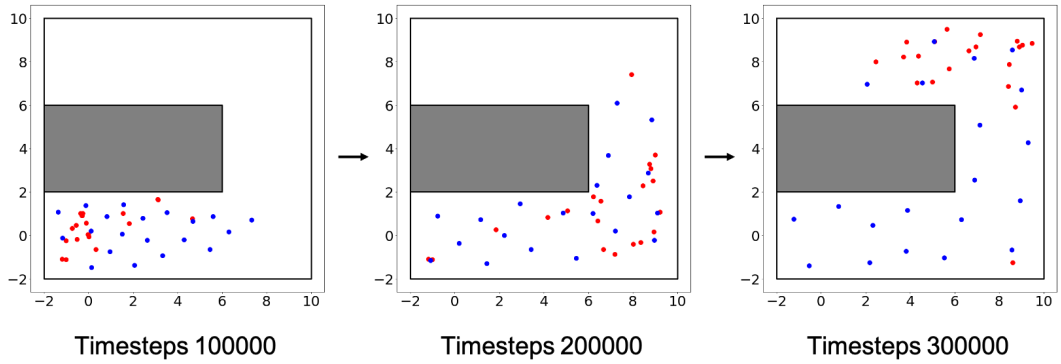


Figure 6: Qualitative analysis on the landmark sampling scheme in HIGL. The blue dots denote the landmarks sampled by the coverage criteria, and the red dots denote the landmarks sampled by the novelty criteria. As shown in the figure, the blue dots are dispersed across the visited state space, and red dots are concentrated at the frontier of the visited space, which is likely to be novel. The analysis is conducted on the Ant Maze (U-shape, dense).

space. In particular, at the early phase of training, the coverage-based landmarks are scattered in the bottom-left region of the maze since an agent is born at the bottom-left corner in the beginning of an episode. As training proceeds, the agent visits wider regions, so coverage based-landmarks are more scattered across the map. Remarkably, the novelty-based landmarks are concentrated at the frontier of the visited space throughout the training phase.

6 Discussion and conclusion

We present HIGL, a new framework for training a high-level policy with reduced action space guided by landmarks. Our main idea is (a) sampling landmarks that are informative for exploration and (b) training the high-level policy to be encouraged to generate a subgoal toward a selected landmark. Experiments show that HIGL outperforms the prior state-of-the-art method thanks to efficient exploration by landmarks. We believe that our framework would guide new interesting direction in the HRL: high-level action space reduction into the promising region to explore.

Limitation. While our experiments demonstrate that HIGL is effective for solving the complex, long-horizon continuous control tasks, our work might be limited in that we only consider the setup where the final goal of a task is given by the environment. An interesting future direction is to develop a landmark selection scheme that works without the final goal, i.e., selecting a landmark which is the most novel among nearby landmarks, or selecting the very first landmark in the shortest path to a *pseudo* final goal, (i.e., hindsight goal). Also, one may point out that the cost consumed for high-level planning may cause a scalability issue, as we perform the shortest path planning at every training step. However, given that (i) our method does not utilize planning at deployment time where the execution response time is important, and (ii) environment interaction for sample collection is often dangerous and expensive, we believe that incurring such costs to improve the response time and the sample-efficiency of the algorithm is a reasonable and appropriate direction.

Potential negative impacts. This work would promote the research in the field of HRL and RL, and has potential real-world applications such as robotics. However, there could be unfortunately the potential negative consequences of developing an algorithm for autonomous agents. For example, if a malicious user specifies a reward function that corresponds to harmful behavior for a society, an RL agent would just learn such behaviors without considering the expected results from such behaviors. Specifically, developing a HRL agent for solving a complex and long-term task would facilitate the development of the real-world deployment of malicious robots, which could perform a long-horizon operation in an autonomous way without the direction of a human. For this reason, in addition to developing a HRL and RL algorithms for improving the sample-efficiency and performance, it is important to develop a method that could consider the consequence of its own behaviors to a society for learning behaviors.

References

[1] Badia, Adrià Puigdomènech, Piot, Bilal, Kapturowski, Steven, Sprechmann, Pablo, Vitvitskyi, Alex, Guo, Zhaohan Daniel, and Blundell, Charles. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pp. 507–517. PMLR, 2020.

[2] Burda, Yuri, Edwards, Harrison, Storkey, Amos, and Klimov, Oleg. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

[3] Dayan, Peter and Hinton, Geoffrey E. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*, 1993.

[4] Dilokthanakul, Nat, Kaplanis, Christos, Pawlowski, Nick, and Shanahan, Murray. Feature control as intrinsic motivation for hierarchical reinforcement learning. *IEEE transactions on neural networks and learning systems*, 30(11):3409–3418, 2019.

[5] Duan, Yan, Chen, Xi, Houthooft, Rein, Schulman, John, and Abbeel, Pieter. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

[6] Eysenbach, Ben, Salakhutdinov, Russ R, and Levine, Sergey. Search on the replay buffer: Bridging planning and reinforcement learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5c48ff18e0a47baaf81d8b8ea51eec92-Paper.pdf.

[7] Florensa, Carlos, Duan, Yan, and Abbeel, Pieter. Stochastic neural networks for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2017.

[8] Fujimoto, Scott, Hoof, Herke, and Meger, David. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

[9] Ghosh, Dibya, Gupta, Abhishek, and Levine, Sergey. Learning actionable representations with goal conditioned policies. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hye9lnCct7.

[10] Huang, Zhiao, Liu, Fangchen, and Su, Hao. Mapping state space using landmarks for universal goal reaching. *Advances in Neural Information Processing Systems*, 32:1942–1952, 2019.

[11] Kalashnikov, Dmitry, Irpan, Alex, Pastor, Peter, Ibarz, Julian, Herzog, Alexander, Jang, Eric, Quillen, Deirdre, Holly, Ethan, Kalakrishnan, Mrinal, Vanhoucke, Vincent, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[12] Kulkarni, Tejas D, Narasimhan, Karthik R, Saeedi, Ardavan, and Tenenbaum, Joshua B. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3682–3690, 2016.

[13] Laskin, Michael, Emmons, Scott, Jain, Ajay, Kurutach, Thanard, Abbeel, Pieter, and Pathak, Deepak. Sparse graphical memory for robust planning. *arXiv preprint arXiv:2003.06417*, 2020.

[14] Levy, Andrew, Konidaris, George, Platt, Robert, and Saenko, Kate. Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*, 2018.

[15] Li, Siyuan, Wang, Rui, Tang, Minxue, and Zhang, Chongjie. Hierarchical reinforcement learning with advantage-based auxiliary rewards. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/81e74d678581a3bb7a720b019f4f1a93-Paper.pdf.

[16] Li, Siyuan, Zheng, Lulu, Wang, Jianhao, and Zhang, Chongjie. Learning subgoal representations with slow dynamics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wxRwhSdORKG.

[17] Liu, Kara, Kurutach, Thanard, Tung, Christine, Abbeel, Pieter, and Tamar, Aviv. Hallucinative topological memory for zero-shot visual planning. In *International Conference on Machine Learning*, pp. 6259–6270. PMLR, 2020.

[18] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[19] Nachum, Ofir, Gu, Shixiang, Lee, Honglak, and Levine, Sergey. Data-efficient hierarchical reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3307–3317, 2018.

[20] Nachum, Ofir, Gu, Shixiang, Lee, Honglak, and Levine, Sergey. Near-optimal representation learning for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2019.

[21] Nagabandi, Anusha, Clavera, Ignasi, Liu, Simin, Fearing, Ronald S, Abbeel, Pieter, Levine, Sergey, and Finn, Chelsea. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *ICLR*, 2019.

[22] Nair, Suraj and Finn, Chelsea. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*, 2019.

[23] Nasiriany, Soroush, Pong, Vitchyr, Lin, Steven, and Levine, Sergey. Planning with goal-conditioned policies. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/c8cc6e90ccbff44c9cee23611711cdc4-Paper.pdf.

[24] Péré, Alexandre, Forestier, Sébastien, Sigaud, Olivier, and Oudeyer, Pierre-Yves. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations*, 2018.

[25] Savinov, Nikolay, Dosovitskiy, Alexey, and Koltun, Vladlen. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SygwwGbRW.

[26] Schmidhuber, Jürgen and Wahnsiedler, Reiner. Planning simple trajectories using neural subgoal. In *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, volume 2, pp. 196. MIT Press, 1993.

[27] Schrittwieser, Julian, Antonoglou, Ioannis, Hubert, Thomas, Simonyan, Karen, Sifre, Laurent, Schmitt, Simon, Guez, Arthur, Lockhart, Edward, Hassabis, Demis, Graepel, Thore, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

[28] Shang, Wenling, Trott, Alex, Zheng, Stephan, Xiong, Caiming, and Socher, Richard. Learning world graphs to accelerate hierarchical reinforcement learning. *arXiv preprint arXiv:1907.00664*, 2019.

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

[30] Sukhbaatar, Sainbayar, Denton, Emily, Szlam, Arthur, and Fergus, Rob. Learning goal embeddings via self-play for hierarchical reinforcement learning. *arXiv preprint arXiv:1811.09083*, 2018.

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

[32] Todorov, Emanuel, Erez, Tom, and Tassa, Yuval. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

[33] Vassilvitskii, Sergei and Arthur, David. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2006.

[34] Vezhnevets, Alexander Sasha, Osindero, Simon, Schaul, Tom, Heess, Nicolas, Jaderberg, Max, Silver, David, and Kavukcuoglu, Koray. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549. PMLR, 2017.

[35] Zhang, Amy, Sukhbaatar, Sainbayar, Lerer, Adam, Szlam, Arthur, and Fergus, Rob. Composable planning with attributes. In *International Conference on Machine Learning*, pp. 5842–5851. PMLR, 2018.

[36] Zhang, Marvin, Vikram, Sharad, Smith, Laura, Abbeel, Pieter, Johnson, Matthew J, and Levine, Sergey. Solar: deep structured representations for model-based reinforcement learning. In *ICML*, 2019.

[37] Zhang, Tianren, Guo, Shangqi, Tan, Tian, Hu, Xiaolin, and Chen, Feng. Generating adjacency-constrained subgoals in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5, and the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5 and the supplementary material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5 and the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]