

## A NOTATIONS

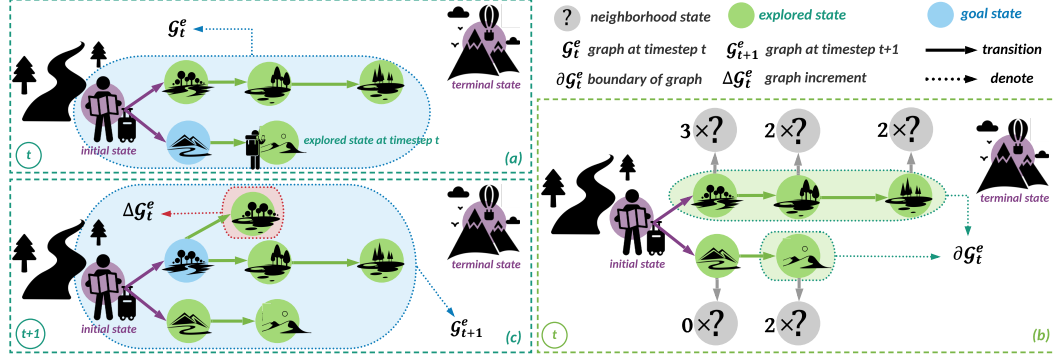


Figure 7: An illustrated example of notations in Graph Structured Reinforcement Learning (GSRL).

In this paper, we construct a state-transition graph on top of the replay buffer. As Figure 7(a) shows, we build the graph  $\mathcal{G}_t^e$  based on historical explored trajectories at timestep  $t$  in episode  $e$ . For any not fully explored state-transition graph, there exist many not well explored states. We measure the exploration (*i.e.*, certainty in Definition 1) of these states according to the number of their untaken candidate actions. As illustrated in Figure 7(b), we define the boundary of  $\mathcal{G}_t^e$  as a set of states, at least one of whose candidate actions is not ready been taken. Each untaken action may lead to unvisited states (denoted as ? icon). We denote the boundary as  $\partial\mathcal{G}_t^e$ . As illustrated in Figure 7(c), after each timestep  $t + 1$ , the agent explored a new state denoted as  $\Delta\mathcal{G}_t^e$ . Then,  $\mathcal{G}_t^e$  and  $\Delta\mathcal{G}_t^e$  together make up the dynamic graph at timestep  $t + 1$  denoted as  $\mathcal{G}_{t+1}^e$ .

## B ALGORITHM

---

### Algorithm 1 Graph Structured Reinforcement Learning (GSRL)

---

```

1: Initialize replay buffer  $\mathcal{D} = \{s_0\}$  and state-transition graph  $\mathcal{G} = \{s_0\}$ 
2: for episode number  $e = 1, 2, \dots, E$  do
3:   Select an appropriate group for exploration according to Eq. (2)
4:   Generate goal  $g_e$  according to Eq. (3)
5:   for interaction  $t = 0, 1, 2, \dots, T - 1$  do
6:     Receive observation  $s_t$  from environment
7:      $a_t \leftarrow \epsilon$ -greedy policy based on  $Q(s_t, a, g)$ 
8:     Take action  $a_t$ , receive reward  $r_t$  and next state  $s_{t+1}$ 
9:     Append  $(s_t, a_t, r_t, s_{t+1}, g_e)$  to  $\mathcal{D}$ 
10:    Relabel rewards  $r_g$  with  $g_e$ 
11:    Append  $(s_t, a_t, s_{t+1})$  to  $\mathcal{G}$  if  $(s_t, a_t, s_{t+1}) \notin \mathcal{G}$ 
12:    if  $t \bmod \text{update\_interval} == 0$  then
13:      Sample related experience  $(s_t, a_t, s_{t+1}, r_t, g_e)$  to  $\mathcal{D}_{\text{related}}$ 
14:      Update parameter  $\theta$  using Eq. (6)
15:    end if
16:  end for
17:  Compute optimal goal  $g_e^*$  according to Definition 2
18:  Update parameter  $\phi$  using Eq. (4)
19: end for

```

---

We provide the overall algorithm in Algorithm 1. The key contribution of our paper is to leverage structured information in the state-transition graph for efficient goal generation and value estimation, which is represented in line 4 and 13, respectively. We then describe the overall procedure of GSRL according to Algorithm 1 as follows:

There is no graph structure for agent to support when the task starts. Hence, the agent initializes the replay buffer  $\mathcal{D}$  and the state-transition graph  $\mathcal{G}$  in line 1. At the beginning timestep of each episode

$e$ , we divide the boundary of the state-transition graph  $\partial\mathcal{G}_0^e$  into  $N$  groups and adopt attention mechanism to select an appropriate one for exploration in line 3. Within selected group, we choose the state with the highest value as the generated goal  $g_e$  in line 4. The agent try to reach the goal state through current policy based  $Q$  value in line 7 and record interaction history in the replay rebuffer in line 9. As goal-oriented RL provides the agent intrinsic reward conditioned on current goal, the agent is required to relabel reward with  $r_g$  conditioned on  $g_e$  in line 10. Then the agent updates the state-transition graph in line 11. In order to efficiently update policy, the agent sample related trajectories that contains at least one neighbor states of current state in line 13. In line 14, the policy is updated with DDPG (Lillicrap et al., 2015). At the end of each episode  $e$ , the state-transition graph is actually built for episode  $e + 1$  denoted as  $\mathcal{G}_0^{e+1}$ . The agent is able to find optimal goal  $g_e^*$  through planning algorithm on  $\mathcal{G}_0^{e+1}$  in line 17. The attention mechanism is updated by supervised learning in line 18.

## C PROOFS

### C.1 PROOF OF PROPOSITION 1

**Proposition 1.** *Given the full state-transition graph  $\mathcal{G}_{full}$ , we assume that the probability of degree of an arbitrary state being less than or equal to  $d$  is larger than  $p$  (i.e.,  $P(\deg(s) \leq d) > p, \forall s \in S_{\mathcal{G}}$ ). Considering a sequence of consecutively expanding sub-graphs  $(\mathcal{G}_0^0, \mathcal{G}_1^0, \dots, \mathcal{G}_{T-1}^{E-1})$ , starting with  $\mathcal{G}_0^0 = \{s_0\}$ , for all  $t \geq 1, e \geq 0$ , we can ensure that  $P(|S_{\mathcal{G}_t^e}| \leq \epsilon) > p^\epsilon$ , where  $\epsilon = \frac{d \cdot (d-1)^{T \cdot e + t - 2}}{d-2}$  when  $d > 2$  and  $\epsilon = 1, 3$  when  $d = 1, 2$  respectively.*

*Proof.* We consider the extreme case of greedy consecutive expansion at each timestep  $t$  in any episode  $e$ , where  $\mathcal{G}_{t+1}^e = \mathcal{G}_t^e \cup \Delta\mathcal{G}_t^e = \mathcal{G}_t^e \cup \partial\mathcal{G}_t^e$ , since if this case satisfies the inequality, any case of consecutive expansion can also satisfy it. By definition, all the subgraphs  $\mathcal{G}_t^e$  are a connected graph. Here, we use  $\Delta S_t^e$  to denote  $S_{\Delta\mathcal{G}_t^e}$  for short. In each episode, we can ensure that the newly added nodes  $\Delta S_t^e$  at timestep  $t$  only belong to the neighborhood of the last added nodes  $\Delta S_{t-1}^e$ .

Within each episode  $e$ , we study the sequence  $\{\Delta\mathcal{G}_0^e, \Delta\mathcal{G}_1^e, \dots, \Delta\mathcal{G}_{T-1}^e\}$ , where  $T$  is the episode length. In this case, each node in  $\Delta S_t^e$  already has at least one edge within  $\Delta\mathcal{G}_{t-1}^e$  due to the definition of connected graphs, we can have

$$P(|\Delta S_t^e| \leq |\Delta S_{t-1}^e| \cdot (d-1)) > p^{|\Delta S_{t-1}^e|}. \quad (7)$$

For  $e = 1$  and  $t = 0$ , we have  $P(|\Delta S_1^1| \leq d) > p$  and thus

$$P(|S_{\mathcal{G}_0^1}| \leq 1 + d) > p. \quad (8)$$

For  $e \geq 1$  and  $t \geq 1$ , we analyze the consecutive expansion of the state-transition graph  $\mathcal{G}$  as

$$\begin{aligned} & \mathcal{G}^1 \rightarrow \mathcal{G}^2 \rightarrow \dots \rightarrow \mathcal{G}^E \\ \Rightarrow & \underbrace{\mathcal{G}_0^1 \rightarrow \mathcal{G}_1^1 \rightarrow \dots \rightarrow \mathcal{G}_{T-1}^1}_{\mathcal{G}^1} \rightarrow \underbrace{\mathcal{G}_0^2 \rightarrow \mathcal{G}_1^2 \rightarrow \dots \rightarrow \mathcal{G}_{T-1}^2}_{\mathcal{G}^2} \rightarrow \dots \rightarrow \underbrace{\mathcal{G}_0^E \rightarrow \mathcal{G}_1^E \rightarrow \dots \rightarrow \mathcal{G}_{T-1}^E}_{\mathcal{G}^E}. \end{aligned} \quad (9)$$

Given that  $|\Delta S_{\mathcal{G}_t^e}| \geq 1, \forall t \in [0, T-1]$ , we consider the extreme case that  $|\Delta S_{\mathcal{G}_t^e}| = 1, \forall t \in [0, T-1]$ , which means that every exploration will result in a new explored state and should be responses to the upper bound of the explosion. Based on  $|\Delta S_{\mathcal{G}_t^e}| = 1 + |\Delta S_{\mathcal{G}_0^1}| + |\Delta S_{\mathcal{G}_1^1}| + \dots + |\Delta S_{\mathcal{G}_{T-1}^1}| + |\Delta S_{\mathcal{G}_0^2}| + |\Delta S_{\mathcal{G}_1^2}| + \dots + |\Delta S_{\mathcal{G}_{T-1}^2}| + \dots + |\Delta S_{\mathcal{G}_0^E}| + \dots + |\Delta S_{\mathcal{G}_{T-1}^E}|$ , we have

$$P(|S_{\mathcal{G}^e}| \leq 1 + d + d \cdot (d-1) + \dots + d \cdot (d-1)^{e \cdot T + t - 1}) > p^{1 + d + d \cdot (d-1) + \dots + d \cdot (d-1)^{e \cdot T + t - 2}}. \quad (10)$$

When  $d = 1$ , there can be only one node, so in this case,  $\epsilon = 1$ . When  $d = 2$ , we follow Eq. (10) and derive that in this case,  $\epsilon = 3$ . When  $d > 2$ , it holds that

$$P(|S_{\mathcal{G}^t}| \leq \frac{d \cdot (d-1)^{e \cdot T + t} - 2}{d-2}) > p^{\frac{d \cdot (d-1)^{e \cdot T + t - 1} - 2}{d-2}}. \quad (11)$$

We can find that  $t = 0$  also satisfies this inequality.  $\square$

Notice that here we share some mathematical derivations with Xu et al. (2020).

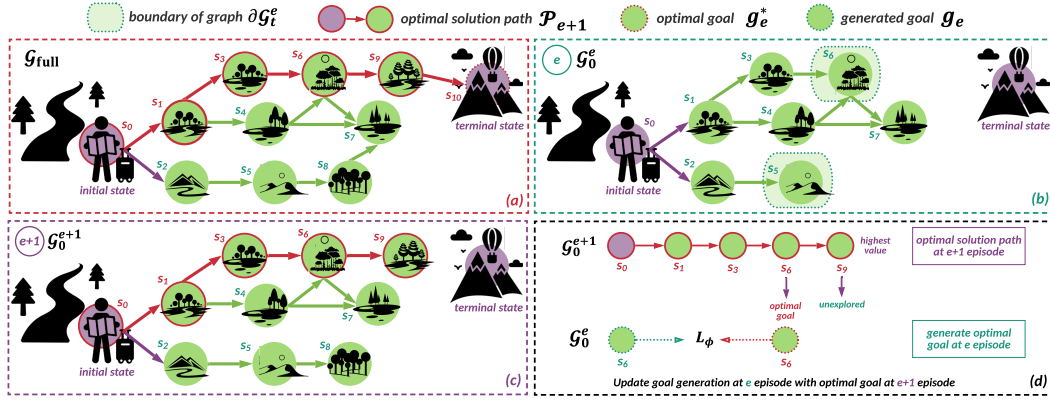


Figure 8: An illustrated example for relationship between optimal goal and boundary. In the fully explored graph  $\mathcal{G}_{\text{full}}$ , the red circled states together show the optimal solution path ( $\mathcal{P}_{\text{full}} = \langle s_0, s_1, s_3, s_6, s_9, s_{10} \rangle$ ) with terminal one ( $s_{10}$ ) for the optimal goal in (a). In any other not fully explored state-transition graph  $\mathcal{G}_0^e$  at the beginning timestep of any episode  $e$  in (b), we regard the reachable state in the dashed line circle ( $s_6$ ) through planning in the next episode  $\mathcal{G}_0^{e+1}$  in (c) as the optimal goal in (d).

## C.2 PROOF OF PROPOSITION 2

**Proposition 2.** Assume that  $Q$ -value of each state is well estimated (i.e.,  $Q = Q^*$ ), then optimal goal  $g_e^*$  at the beginning timestep of any episode  $e$  is always included in the boundary of the state-transition graph  $\mathcal{G}_0^e$  (i.e.,  $\partial\mathcal{G}_0^e$ ).

*Proof.* According to Definition 2 as shown in Figure 8(a), in the fully explored graph  $\mathcal{G}_{\text{full}}$ , the optimal goal  $g_{\text{full}}^*$  is the terminal state in the optimal solution  $\mathcal{P}_{\text{full}}$ , which is also the terminal state in the environment, i.e.,  $s_{10}$ . The intuitive explanation behind this is very natural, where the environment in this case is fully explored, thus the agent is ready to target at the terminal state. In the other cases, we generate the optimal goal  $g_e^*$  of episode  $e$  at the episode  $e + 1$ . Specially, we find the shortest path to the highest value state in  $\mathcal{G}_0^{e+1}$  as the optimal solution path  $\mathcal{P}_{e+1}$ . As Figure 8 illustrates, in the episode  $e + 1$ , the highest value is  $s_9$  and the optimal solution path in this case is  $\mathcal{P}_{e+1} = \langle s_0, s_1, s_3, s_6, s_9 \rangle$ . We then compare the explored states in  $\mathcal{G}_0^e$  with the states in  $\mathcal{P}_{e+1}^{\text{inverse}}$ , where  $\mathcal{P}_{e+1}^{\text{inverse}} = \langle s_9, s_6, s_3, s_1, s_0 \rangle$  is the inverse order of  $\mathcal{P}_{e+1}$ . As Figure 8(d) shows, finally we obtain  $s_6$  as the optimal goal  $g_e^*$ . As stated above, it's easy to find that there are two cases in the optimal goal generation. One is the last node of solution path  $\mathcal{P}_{e+1}$ . The other is one of the rest nodes in  $\mathcal{P}_{e+1}$  except the last one. We then prove that in the both of these cases, optimal goal  $g_e^*$  is always included in the boundary of the state-transition graph  $\partial\mathcal{G}_0^e$ .

**Case I: Node at Last.** If  $Q$ -value of each state is well estimated, i.e.,  $Q = Q^*$ , then the optimal solution path  $\mathcal{P}_{e+1}$  at episode  $e + 1$  should be close to the optimal solution path  $\mathcal{P}_{\text{full}}$  in the full graph  $\mathcal{G}_{\text{full}}$ , and the last state of the path  $\mathcal{P}_{e+1}$  should be closest to the terminal state. Hence, if  $g_e^*$  is not in the boundary, there must be one neighbor node closer to the terminal state. Otherwise,  $g_e^*$  is the dead end, thus should not be regarded as the optimal goal. And if there is one neighbor node closer to the terminal state, then this state should be regarded as the optimal goal. Therefore, we obtain the contradiction.

**Case II: Node Not at Last.** If the optimal goal is not the last state, then there must exist the state unexplored at episode  $e$ . Take Figure 8 as an example, if we take  $s_6$  as the optimal goal  $g_e^*$  in (d), state  $s_9$  must be unexplored in  $\mathcal{G}_0^e$  in (c) and explored in  $\mathcal{G}_0^{e+1}$  in (b). If  $g_e^*$  is not included in  $\partial\mathcal{G}_0^e$ , then there should not exist any unexplored state that is included in its neighborhood. According to the definition of the boundary of graph, we have proved the proposition by contradiction.

In summary, we have proved the proposition in both two cases by the contradiction.  $\square$

## C.3 PROOF OF PROPOSITION 3

**Proposition 3.** Denote the Bellman backup operator in Eq. (5) as  $\mathcal{B} : \mathbb{R}^{|S| \times |A| \times |G|} \rightarrow \mathbb{R}^{|S| \times |A| \times |G|}$  and a mapping  $Q : S \times A \times G \rightarrow \mathbb{R}^{|S| \times |A| \times |G|}$  with  $|S| < \infty$  and  $|A| < \infty$ . Repeated application of the operator  $\mathcal{B}$  for our graph-based state-action value estimate  $\hat{Q}_G$  converges to a unique optimal value  $\hat{Q}_{G^*}^*$  with well explored graph  $\mathcal{G}^*$  including optimal solution path.

*Proof.* The proof of Proposition 3 is done in two main steps. The first step is to show that our state-transition graph  $\mathcal{G}$  can converge to well explored graph  $\mathcal{G}^*$ . Here, we define  $\mathcal{G}^*$  as the graph that includes the optimal path (i.e.  $\mathcal{P}_{\text{full}}$  in Definition 2). In the second step, we prove that given graph  $\mathcal{G}$ , our graph-based method can converge to unique optimal value  $Q_{\mathcal{G}}^*$ .

**Step I.** Since  $|S| < \infty$  and  $|A| < \infty$ , we can obtain that  $\mathcal{V}_{\mathcal{G}} < \infty$  and  $\mathcal{E}_{\mathcal{G}} < \infty$ . Note that the state-transition graph  $\mathcal{G}$  is a dynamic graph and goals  $g$  generated on  $\mathcal{G}$  are updated at the beginning timestep of each episode. Hence, there is a sequence of goals denoted as  $(g_1, g_2, \dots, g_E)$  and corresponding sequence of graphs denoted as  $(\mathcal{G}_0^1, \mathcal{G}_0^2, \dots, \mathcal{G}_0^E)$ , where  $E$  here is the number of episodes. Given that  $|S| < \infty$  and  $|A| < \infty$ , the number of nodes and edges in the full graph  $\mathcal{G}_{\text{full}}$  is also bounded. Based on the explore strategy introduced in Section 4, we know that goal-oriented RL will first search for a path leading to the terminal state. After that, the terminal state will be included in  $\mathcal{G}$ . Then the agent will seek for the shortest path to the terminal state, because the agent is given with a negative reward at each timestep. Hence, the optimal solution path  $\mathcal{P}_{\text{full}}$  will be involved. Hence, we can obtain that

$$\mathcal{G}_0^1 \subseteq \mathcal{G}_0^2 \subseteq \dots \subseteq \mathcal{G}^* \Rightarrow \mathcal{G} \rightarrow \mathcal{G}^*. \quad (12)$$

Assume that  $E$  is large enough, our state-transition graph  $\mathcal{G}$  can finally converge to well explored graph  $\mathcal{G}^*$ .

**Step II.** Note that the proof of convergence for our graph-based goal-oriented RL is quite similar to  $Q$ -learning (Bellman, 1966; Bertsekas et al., 1995; Sutton & Barto, 2018). The differences between our approach and  $Q$ -learning are that  $Q$  value  $Q(s, a, g)$  is also conditioned on goal  $g$ , and that the state-transition probability  $P_{\mathcal{G}}(s'|s, a)$  can be reflected by graph  $\mathcal{G}$ . We provide detailed proof as follows:

For any state-transition graph  $\mathcal{G}$ , we can obtain goal  $g \in G$  conditioned on  $\mathcal{G}$  from Step I. Based on that, our estimated graph-based action-value function  $\hat{Q}_{\mathcal{G}}$  can be defined as

$$\mathcal{B}\hat{Q}_{\mathcal{G}}(s, a, g) = R(s, a, g) + \gamma \cdot \max_{a' \in A} \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot \hat{Q}_{\mathcal{G}}(s', a', g). \quad (13)$$

For any action-value function estimates  $\hat{Q}_{\mathcal{G}}^1, \hat{Q}_{\mathcal{G}}^2$ , we study that

$$\begin{aligned} & |\mathcal{B}\hat{Q}_{\mathcal{G}}^1(s, a, g) - \mathcal{B}\hat{Q}_{\mathcal{G}}^2(s, a, g)| \\ &= \gamma \cdot \left| \max_{a' \in A} \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot \hat{Q}_{\mathcal{G}}^1(s', a', g) - \max_{a' \in A} \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot \hat{Q}_{\mathcal{G}}^2(s', a', g) \right| \\ &\leq \gamma \cdot \max_{a' \in A} \left| \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot \hat{Q}_{\mathcal{G}}^1(s', a', g) - \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot \hat{Q}_{\mathcal{G}}^2(s', a', g) \right| \\ &= \gamma \cdot \max_{a' \in A} \sum_{s' \in S} P_{\mathcal{G}}(s'|s, a) \cdot |\hat{Q}_{\mathcal{G}}^1(s', a', g) - \hat{Q}_{\mathcal{G}}^2(s', a', g)| \\ &\leq \gamma \cdot \max_{s \in S, a \in A} |\hat{Q}_{\mathcal{G}}^1(s, a, g) - \hat{Q}_{\mathcal{G}}^2(s, a, g)| \end{aligned} \quad (14)$$

So the contraction property of Bellman operator holds that

$$\max_{s \in S, a \in A} |\mathcal{B}\hat{Q}_{\mathcal{G}}^1(s, a, g) - \mathcal{B}\hat{Q}_{\mathcal{G}}^2(s, a, g)| \leq \gamma \cdot \max_{s \in S, a \in A} |\hat{Q}_{\mathcal{G}}^1(s, a, g) - \hat{Q}_{\mathcal{G}}^2(s, a, g)| \quad (15)$$

For the fixed point  $Q_{\mathcal{G}}^*$ , we have that

$$\max_{s \in S, a \in A} |\mathcal{B}\hat{Q}_{\mathcal{G}}(s, a, g) - \mathcal{B}Q_{\mathcal{G}}^*(s, a, g)| \leq \gamma \cdot \max_{s \in S, a \in A} |\hat{Q}_{\mathcal{G}}(s, a, g) - Q_{\mathcal{G}}^*(s, a, g)| \Rightarrow \hat{Q}_{\mathcal{G}} \rightarrow Q_{\mathcal{G}}^*. \quad (16)$$

Combining Step I and II, we can conclude that our graph-based estimated state-action value  $\hat{Q}_{\mathcal{G}}$  can converge to unique optimal value  $Q_{\mathcal{G}}^*$ .  $\square$

## D DISCUSSIONS

### D.1 DISCUSSION ON CERTAINTY OF STATE

In this section, we further discuss the relationship between certainty of state and number of state. In the previous exploration RL literatures (Ostrowski et al., 2017; Bellemare et al., 2016), the performance of exploration often is measured by the number of the visited states. Namely, given fixed

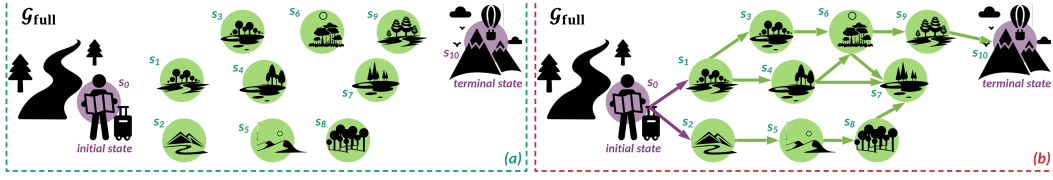


Figure 9: An illustrated example for relationship between certainty and number of visited states.

number of episodes, more visited states, better performance. In this paper, we propose to utilize a new measurement, *i.e.*, certainty of state as illustrated in Definition 1. We conclude the relations between certainty and number of visited states as Proposition 4.

**Proposition 4.** *Given a whole state-transition graph  $\mathcal{G}_{full}$ , we can regard the certainty of states as the local measurement and the number of states as the global measurement for exploration, which share the similar trend during agent exploration.*

*Proof.* We illustrate and prove the proposition hindsightly. If we have the fully observation for states as shown in Figure 9(a), we can model the agent finding new states as connecting new states with visited states. In other words, since the state-transition graph  $\mathcal{G}_t^e$  must keep to be a fully connected graph at any timestep  $t$  in any episode  $e$ . Hence, adding new states into the visited state set can always be regarded as finding new edge between new states and the visited state set. And each directed edge in the state-transition graph as shown in Figure 9(b) is determined by action and state-transition function. If the environment is determined, we can roughly regard the number of edges as the approximate measurement for exploration. The certainty of states is the local perspective for this measurement.  $\square$

## D.2 DISCUSSION ON OPTIMAL GOAL

In the previous goal-oriented RL literatures (Andrychowicz et al., 2017; Ren et al., 2019), what kind of generated goals is helpful for the agent to efficiently learn a well performed policy is one of the key question to be answered. The basic idea of goal-oriented RL architecture is to generate goals to decompose the complex task into several goal-oriented tasks. In this paper, we analyze our generated goals in two perspectives, namely reachability and curriculum.

**Reachability.** The first property required in the optimal goal is that the generated goal is guaranteed to be reachable for the agent. To this end, in this paper, the candidate goal set is constrained into the visited states. In other words, the goal generated in the episode  $e$  must be visited before the episode  $e$ . Therefore, we can guarantee that the generated goal is reachable.

**Curriculum.** The second property is the curriculum, which means that our optimal goals are required to approach the terminal state during the exploration. If the  $Q$ -value of each states is well estimated, our goal generation under the supervision of forward-looking planning at the next episode will focus on the potential highest value states in the future, which is actually the terminal state when the agent has the full observation of states.

## D.3 DISCUSSION ON GROUP DIVISION

**Motivation.** The intuitive motivation behind the group division is very natural. Proposition 1 implies that exploration on the state-transition graph  $\mathcal{G}_t^e$  at timestep  $t$  in episode  $e$  without any constraint may lead to explosion of graph and inefficiency of exploration. Therefore, the agent is expected to do exploration within a limited domain. Considering that  $\mathcal{G}_t^e$  is always changing and the number of nodes (*i.e.*,  $|\mathcal{S}_{\mathcal{G}_t^e}|$ ) keeps increasing, it is non-trivial for the agent to learn to select state as the goal for further exploration. Hence, we first restrict the exploration within the boundary of state-transition graph  $\partial\mathcal{G}_t^e$  according to Proposition 2. We then consider to partition  $\partial\mathcal{G}_t^e$  into several groups.

We set the last visited state  $s_{last}$  as the original point, because  $s_{last}$  is likely to be close to the target state and reachable for current policy. As introduced in Section 4, we propose to extend groups from  $s_{last}$  following two possible perspectives, namely neighbor and uncertain nodes.

**Complexity.** Let  $d_{\partial\mathcal{G}_t^e}$  denote the maximum degree of states in  $\partial\mathcal{G}_t^e$ , and  $|\mathcal{S}_{\partial\mathcal{G}_t^e}|$  denote the number of states in  $\partial\mathcal{G}_t^e$ . Note that  $\partial\mathcal{G}_t^e$  is always a directed fully connected graph. If we want to find the  $n$ -hop neighbors of  $s_{last}$ , we need to iteratively go through related nodes' neighborhood. In other words,

the computation complexity should be  $\mathcal{O}(d_{\partial\mathcal{G}_t^e}^n)$ . Hence, the complexity to construct  $\mathcal{C}_1, \dots, \mathcal{C}_N$  by extending from neighbor nodes is  $\mathcal{O}(d_{\partial\mathcal{G}_t^e}^1) + \mathcal{O}(d_{\partial\mathcal{G}_t^e}^2) + \dots + \mathcal{O}(d_{\partial\mathcal{G}_t^e}^{N-1}) = \mathcal{O}(d_{\partial\mathcal{G}_t^e}^{N-1})$ . If we want to find nodes whose uncertainty equals  $n$ , we need to go through the graph once. In this case, the computation complexity should be  $\mathcal{O}(|S_{\partial\mathcal{G}_t^e}|)$ . Hence, the complexity to construct  $\mathcal{C}_1, \dots, \mathcal{C}_N$  extending from uncertain nodes is  $\mathcal{O}(|S_{\partial\mathcal{G}_t^e}|)$ .

## E EXPERIMENTS

### E.1 ENVIRONMENT CONFIGURATION

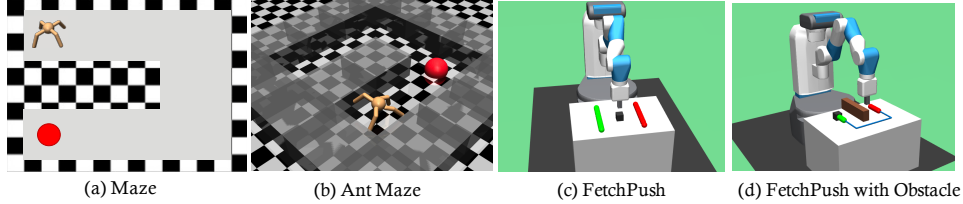


Figure 10: Visualization of robotic manipulation environments.

**Maze.** As shown in Figure 10(a), in the maze environment, a point in a 2D U-maze aims to reach a goal represented by a red point. The size of maze is  $15 \times 15$ , the state space and is in this 2D U-maze, and the goal is uniformly generated on the segment from  $(0, 0)$  to  $(15.0, 15.0)$ . The action space is from  $(-1.0, -1.0)$  to  $(1.0, 1.0)$ , which represents the movement in  $x$  and  $y$  directions.

**AntMaze.** As shown in Figure 10(b), in the AntMaze environment, an ant is put in a U-maze, and the size of maze is  $12 \times 12$ . The ant is put on a random location on the segment from  $(-2.0, -2.0)$  to  $(10.0, 10.0)$ , and the goal is uniformly generated on the segment from  $(-2.0, -2.0)$  to  $(10.0, 10.0)$ . The state of ant is 30-dimension, including its positions and velocities.

**FetchPush.** As shown in Figure 10(c), in the fetch environment, the agent is trained to fetch object from initial position (rectangle depicted in green) to distant position (rectangle depicted in red). Let the origin  $(0, 0, 0)$  denote the projection of gripper’s initial coordinate on the table. The object is uniformly generated on the segment from  $(-0.0, -0.0, 0)$  to  $(8, 8, 0)$ , and the goal is uniformly generated on the segment from  $(-0.0, -0.0, 0)$  to  $(8, 8, 0)$ .

**FetchPush with Obstacle.** As shown in Figure 10(d), in the fetch with obstacle environment, we create an environment based on FetchPush with a rigid obstacle, where the brown block is a static wall that can’t be moved. The object is uniformly generated on the segment from  $(-0.0, -0.0, 0)$  to  $(8, 8, 0)$ , and the goal is uniformly generated on the segment from  $(-0.0, -0.0, 0)$  to  $(8, 8, 0)$ .

**AntMaze with Obstacle.** This environment is an extension version of AntMaze, where an  $1 \times 1$  rigid obstacle is put in U-maze.

### E.2 EVALUATION DETAILS

- All curves presented in this paper are plotted from 10 runs with random task initialization and seeds.
- Shaded region indicates 60% population around median.
- All curves are plotted using the same hyper-parameters (except ablation section).
- Following (Andrychowicz et al., 2017), an episode is considered successful if  $\|g - s_{\text{object}}\|_2 \leq \delta_g$  is achieved, where  $s_{\text{object}}$  is the object position at the end of the episode.  $\delta_g$  is the threshold.
- The max timestep for each episode is set as 200 for training and 500 for tests.
- The average success rate using in the curve is estimated by  $10^2$  samples.

### E.3 HYPER-PARAMETERS

Almost all hyper-parameters using DDPG (Lillicrap et al., 2015) and HER (Andrychowicz et al., 2017) are kept the same as benchmark results, except these:

- Number of MPI workers: 1;
- Actor and critic networks: 3 layers with 256 units and ReLU activation;



- Adam optimizer with  $5 \times 10^{-4}$  learning rate;
- Polyak-averaging coefficient: 0.98;
- Action  $l_2$ -norm penalty coefficient: 0.5;
- Batch size: 256;
- Probability of random actions: 0.2;
- Scale of additive Gaussian noise: 0.2;
- Probability of HER experience replay: 0.8;
- Number of batches to replay after collecting one trajectory: 50.

Hyper-parameters in goal generation:

- Adam optimizer with  $1 \times 10^{-3}$  learning rate;
- $K$  of  $K$ -bins discretization: 20;
- Number of groups to depart the graph: 3.

#### E.4 COMPARISON ON SAMPLE EFFICIENCY

We show sample efficiency with comparisons according to the number of states visited and actions taken. We report the log files of GSRL and HER in Maze environment here at 10, 50, 100 episodes, which contain the number of visited nodes and actions taken.

```
===== Graph Structured Reinforcement Learning (GSRL) =====
episode is: 10
nodes: [22, 21, 11, 31, 42, 32, 41, 23, 43, 33, 44, 34, 13, 14, 15, 26, 25, 36, 35, 45, 16, 24, 37, 47, 38, 48, 46, 12, 49,
59, 69, 79, 80, 78, 90, 89, 99, 109, 110, 100, 27, 28, 39, 50, 40]
number of nodes: 45
edges: [(22, 22), (22, 21), (22, 23), (22, 32), (22, 33), (22, 13), (22, 31), (22, 12), (21, 21), (21, 11), (21, 31), (21,
32), (21, 22), (11, 11), (11, 21), (11, 12), (31, 31), (31, 42), (31, 41), (31, 22), (31, 32), (31, 21), (42, 42), (42,
32), (42, 43), (42, 41), (42, 31), (32, 31), (32, 32), (32, 42), (32, 43), (32, 33), (32, 23), (32, 22), (41, 41),
(41, 31), (23, 22), (23, 23), (23, 32), (23, 34), (23, 33), (23, 24), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44),
(43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 23), (33, 34), (44, 44), (44, 33), (44, 43), (44, 35), (44, 34),
(44, 45), (34, 43), (34, 33), (34, 34), (34, 24), (34, 35), (34, 44), (34, 45), (13, 13), (13, 14), (13, 23), (14,
15), (14, 25), (14, 14), (15, 15), (15, 26), (15, 25), (15, 14), (15, 16), (26, 26), (26, 25), (26, 36), (26, 16), (26,
37), (26, 27), (25, 26), (25, 25), (25, 15), (25, 36), (36, 35), (36, 26), (36, 36), (36, 37), (36, 27), (35, 35),
(35, 45), (35, 26), (35, 34), (35, 36), (35, 44), (35, 25), (45, 35), (45, 44), (45, 45), (16, 26), (24, 35), (24, 34),
(24, 25), (37, 47), (37, 37), (37, 38), (37, 48), (47, 37), (47, 47), (47, 48), (47, 46), (38, 47), (38, 48), (38, 37),
(38, 49), (38, 28), (38, 39), (48, 38), (48, 48), (48, 49), (46, 47), (46, 46), (12, 11), (12, 21), (12, 23), (12,
22), (12, 13), (49, 48), (49, 59), (49, 50), (59, 69), (69, 69), (69, 79), (79, 80), (79, 78), (79, 79), (79, 90), (80,
79), (78, 79), (90, 89), (89, 99), (99, 99), (99, 109), (109, 110), (109, 109), (109, 100), (110, 100), (100, 109),
(27, 36), (27, 27), (27, 38), (28, 28), (28, 38), (39, 50), (39, 40), (39, 39), (50, 39), (50, 40), (50, 50), (40, 49),
(40, 39), (40, 50)]
number of edges: 166
episode is: 50
nodes: [22, 21, 11, 31, 42, 32, 41, 23, 43, 33, 44, 34, 13, 14, 15, 26, 25, 36, 35, 45, 16, 24, 37, 47, 38, 48, 46, 12, 49,
59, 69, 79, 80, 78, 90, 89, 99, 109, 110, 100, 27, 28, 39, 50, 40, 29, 30, 51, 57, 18, 19, 58, 68, 17, 60, 20, 67, 70,
71, 61]
number of nodes: 60
edges: [(22, 22), (22, 21), (22, 23), (22, 32), (22, 33), (22, 13), (22, 31), (22, 12), (22, 11), (21, 21), (21, 11), (21,
31), (21, 32), (21, 22), (21, 12), (11, 11), (11, 21), (11, 12), (11, 22), (31, 31), (31, 42), (31, 41), (31, 22), (31,
32), (31, 21), (31, 40), (31, 30), (42, 42), (42, 32), (42, 43), (42, 41), (42, 31), (32, 31), (32, 32), (32, 42),
(32, 43), (32, 33), (32, 23), (32, 22), (32, 41), (32, 21), (41, 41), (41, 31), (41, 40), (41, 32), (23, 22), (23, 23),
(23, 32), (23, 34), (23, 33), (23, 24), (23, 13), (23, 14), (23, 12), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44),
(43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 23), (33, 34), (33, 22), (33, 42), (33, 24), (44, 44), (44,
33), (44, 43), (44, 35), (44, 34), (44, 45), (34, 43), (34, 33), (34, 34), (34, 24), (34, 35), (34, 44), (34, 45), (34,
25), (13, 13), (13, 14), (13, 23), (13, 12), (13, 22), (13, 24), (14, 15), (14, 25), (14, 14), (14, 24), (14, 13),
(14, 23), (15, 15), (15, 26), (15, 25), (15, 14), (15, 16), (26, 26), (26, 25), (26, 36), (26, 16), (26, 37), (26, 27),
(26, 35), (26, 17), (25, 26), (25, 25), (25, 15), (25, 36), (25, 24), (25, 35), (25, 16), (25, 34), (36, 35), (36, 26),
(36, 36), (36, 37), (36, 27), (36, 46), (36, 47), (36, 45), (35, 35), (35, 45), (35, 26), (35, 34), (35, 36), (35,
44), (35, 25), (35, 46), (35, 24), (45, 35), (45, 44), (45, 45), (45, 46), (45, 36), (45, 34), (16, 26), (16, 16), (16,
27), (16, 17), (16, 15), (16, 25), (24, 35), (24, 34), (24, 25), (24, 24), (24, 14), (24, 23), (24, 15), (24, 33),
(37, 47), (37, 37), (37, 38), (37, 48), (37, 36), (37, 46), (37, 27), (47, 37), (47, 47), (47, 48), (47, 46),
(47, 38), (47, 58), (47, 57), (47, 36), (38, 47), (38, 48), (38, 37), (38, 49), (38, 28), (38, 39), (38, 38), (38, 27),
(48, 38), (48, 48), (48, 49), (48, 57), (48, 47), (48, 58), (48, 59), (46, 47), (46, 46), (46, 37), (46, 45), (46,
36), (46, 35), (12, 11), (12, 21), (12, 23), (12, 22), (12, 13), (12, 12), (49, 48), (49, 59), (49, 50), (49, 39), (49,
49), (49, 60), (59, 69), (59, 59), (59, 48), (59, 50), (59, 60), (59, 49), (59, 58), (69, 69), (69, 79), (69, 78),
(69, 80), (79, 80), (79, 78), (79, 79), (79, 90), (79, 68), (80, 79), (80, 69), (80, 80), (80, 90), (78, 79), (78, 78),
(78, 68), (78, 69), (78, 89), (90, 89), (90, 79), (90, 60), (89, 99), (89, 79), (99, 99), (99, 109), (109, 110), (109,
109), (109, 100), (110, 100), (100, 109), (27, 36), (27, 27), (27, 38), (27, 28), (27, 26), (27, 37), (27, 18), (27,
16), (27, 17), (28, 28), (28, 38), (28, 27), (28, 18), (28, 19), (28, 37), (28, 39), (28, 29), (39, 50), (39, 40), (39,
39), (39, 29), (39, 38), (50, 39), (50, 40), (50, 50), (50, 49), (50, 59), (50, 60), (50, 51), (40, 49), (40, 39),
(40, 50), (40, 40), (40, 51), (40, 41), (40, 29), (40, 31), (40, 30), (29, 30), (29, 39), (29, 29), (29, 19), (29, 28),
(30, 29), (30, 40), (30, 31), (30, 30), (30, 20), (30, 39), (51, 40), (57, 57), (57, 68), (57, 47), (57, 58), (57, 48),
(18, 19), (18, 27), (18, 28), (18, 18), (18, 17), (18, 29), (19, 28), (19, 19), (19, 18), (19, 30), (19, 59), (58,
48), (58, 58), (58, 57), (58, 59), (58, 49), (58, 47), (58, 67), (68, 69), (68, 78), (68, 79), (17, 17), (17, 18), (17,
28), (17, 16), (17, 27), (60, 50), (60, 60), (60, 49), (60, 70), (60, 61), (60, 59), (20, 19), (67, 67), (67, 58),
(70, 71), (70, 70), (70, 60), (70, 69), (71, 71), (71, 70), (61, 70)]
```

number of edges: 336

episode: 100

nodes: [22, 21, 11, 31, 42, 32, 41, 23, 43, 33, 44, 34, 13, 14, 15, 26, 25, 36, 35, 45, 16, 24, 37, 47, 38, 48, 46, 12, 49, 59, 69, 79, 80, 78, 90, 89, 99, 109, 110, 100, 27, 28, 39, 50, 40, 29, 30, 51, 57, 18, 19, 58, 68, 17, 60, 20, 67, 70, 71, 61, 88, 87, 96, 106, 105, 104, 114, 115, 81, 77, 97, 107, 86, 98, 108, 95, 85, 94, 103]

number of nodes: 79

edges: [(22, 22), (22, 21), (22, 23), (22, 32), (22, 33), (22, 13), (22, 31), (22, 12), (22, 11), (21, 21), (21, 11), (21, 31), (21, 32), (21, 22), (21, 12), (21, 20), (21, 30), (11, 11), (11, 21), (11, 12), (11, 22), (31, 31), (31, 42), (31, 41), (31, 22), (31, 32), (31, 21), (31, 40), (31, 30), (42, 42), (42, 32), (42, 43), (42, 41), (42, 31), (42, 33), (32, 31), (32, 32), (32, 42), (32, 43), (32, 33), (32, 23), (32, 22), (32, 41), (32, 21), (41, 41), (41, 31), (41, 40), (41, 32), (41, 42), (41, 50), (41, 30), (23, 22), (23, 23), (23, 32), (23, 34), (23, 33), (23, 24), (23, 13), (23, 14), (23, 12), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44), (43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 23), (33, 34), (33, 22), (33, 42), (33, 24), (44, 44), (44, 33), (44, 43), (44, 35), (44, 34), (44, 45), (34, 43), (34, 33), (34, 34), (34, 24), (34, 35), (34, 44), (34, 45), (34, 25), (34, 23), (13, 13), (13, 14), (13, 23), (13, 12), (13, 22), (13, 24), (14, 15), (14, 25), (14, 14), (14, 24), (14, 13), (14, 23), (15, 15), (15, 26), (15, 25), (15, 14), (15, 16), (15, 24), (26, 26), (26, 25), (26, 36), (26, 16), (26, 37), (26, 27), (26, 35), (26, 17), (26, 15), (25, 26), (25, 25), (25, 15), (25, 36), (25, 24), (25, 35), (25, 16), (25, 34), (25, 14), (36, 35), (36, 26), (36, 36), (36, 37), (36, 27), (36, 46), (36, 47), (36, 45), (36, 25), (35, 35), (35, 45), (35, 26), (35, 34), (35, 46), (35, 44), (35, 25), (35, 46), (35, 24), (45, 35), (45, 44), (45, 45), (45, 46), (45, 36), (45, 34), (16, 26), (16, 16), (16, 27), (16, 17), (16, 15), (16, 25), (24, 35), (24, 34), (24, 25), (24, 24), (24, 14), (24, 23), (24, 15), (24, 33), (24, 13), (37, 47), (37, 37), (37, 38), (37, 48), (37, 28), (37, 36), (37, 46), (37, 27), (37, 26), (47, 47), (47, 48), (47, 46), (47, 38), (47, 58), (47, 57), (47, 36), (38, 47), (38, 48), (38, 37), (38, 49), (38, 28), (38, 39), (38, 38), (38, 27), (38, 29), (48, 38), (48, 48), (48, 49), (48, 57), (48, 47), (48, 58), (48, 59), (48, 37), (48, 39), (46, 47), (46, 46), (46, 37), (46, 45), (46, 36), (46, 35), (12, 11), (12, 21), (12, 23), (12, 22), (12, 13), (12, 12), (49, 48), (49, 59), (49, 50), (49, 39), (49, 49), (49, 60), (49, 58), (49, 40), (49, 38), (59, 69), (59, 59), (59, 48), (59, 50), (59, 60), (59, 49), (59, 58), (59, 68), (59, 70), (69, 69), (69, 79), (69, 78), (69, 80), (69, 70), (69, 68), (69, 59), (79, 80), (79, 78), (79, 79), (79, 90), (79, 68), (79, 88), (79, 89), (79, 69), (80, 79), (80, 69), (80, 80), (80, 90), (80, 89), (80, 81), (80, 70), (80, 71), (78, 79), (78, 78), (78, 68), (78, 69), (78, 89), (78, 87), (78, 67), (78, 77), (78, 88), (90, 89), (90, 79), (90, 90), (90, 80), (89, 99), (89, 79), (89, 80), (89, 89), (89, 88), (89, 90), (89, 78), (89, 98), (99, 99), (99, 109), (99, 88), (99, 89), (99, 98), (99, 100), (109, 110), (109, 109), (109, 100), (110, 100), (100, 109), (100, 99), (27, 36), (27, 27), (27, 38), (27, 28), (27, 26), (27, 37), (27, 18), (27, 16), (27, 17), (28, 28), (28, 38), (28, 27), (28, 18), (28, 19), (28, 37), (28, 39), (28, 29), (28, 17), (39, 50), (39, 40), (39, 39), (39, 29), (39, 38), (39, 49), (39, 48), (39, 30), (50, 39), (50, 40), (50, 50), (50, 49), (50, 59), (50, 60), (50, 51), (50, 61), (50, 41), (40, 49), (40, 39), (40, 50), (40, 40), (40, 51), (40, 41), (40, 29), (40, 31), (40, 30), (29, 30), (29, 39), (29, 29), (29, 19), (29, 28), (29, 18), (29, 40), (29, 38), (29, 20), (30, 29), (30, 40), (30, 31), (30, 30), (30, 20), (30, 39), (30, 19), (30, 21), (51, 40), (51, 51), (51, 60), (51, 50), (57, 57), (57, 68), (57, 47), (57, 58), (57, 48), (57, 67), (18, 19), (18, 27), (18, 28), (18, 18), (18, 17), (18, 29), (19, 28), (19, 19), (19, 18), (19, 30), (19, 29), (19, 20), (58, 48), (58, 58), (58, 57), (58, 59), (58, 49), (58, 47), (58, 67), (58, 69), (58, 68), (68, 69), (68, 78), (68, 79), (68, 68), (68, 57), (68, 67), (68, 77), (68, 58), (17, 17), (17, 18), (17, 28), (17, 16), (17, 27), (60, 50), (60, 60), (60, 49), (60, 70), (60, 61), (60, 59), (60, 51), (60, 69), (60, 71), (20, 19), (20, 30), (20, 20), (20, 21), (20, 29), (67, 67), (67, 58), (67, 68), (67, 78), (67, 77), (67, 57), (70, 71), (70, 70), (70, 60), (70, 69), (70, 79), (70, 80), (70, 59), (71, 71), (71, 70), (71, 80), (61, 70), (61, 61), (61, 50), (61, 60), (88, 87), (88, 79), (88, 89), (88, 88), (88, 99), (88, 98), (88, 78), (88, 97), (87, 78), (87, 88), (87, 96), (87, 87), (87, 97), (87, 77), (87, 86), (96, 106), (96, 97), (96, 87), (96, 86), (96, 96), (96, 95), (106, 105), (106, 107), (106, 96), (105, 105), (105, 104), (105, 114), (105, 115), (104, 114), (104, 104), (104, 105), (114, 114), (114, 104), (114, 105), (115, 105), (81, 80), (77, 77), (77, 67), (77, 68), (77, 88), (77, 78), (97, 96), (97, 106), (97, 107), (97, 87), (107, 96), (107, 106), (107, 107), (107, 108), (86, 96), (86, 86), (98, 99), (98, 89), (98, 98), (95, 95), (95, 85), (95, 94), (85, 85), (85, 95), (94, 103), (103, 103)]

number of edges: 486

===== Hindsight Experience Replay (HER) =====

episode is: 10

nodes: [22, 21, 31, 41, 42, 32, 23, 43, 33, 44, 34, 35, 45, 46, 36, 37, 47, 13, 12, 14, 15, 16, 17, 26, 25, 24, 48, 11, 27]

number of nodes: 29

edges: [(22, 21), (22, 23), (22, 22), (22, 32), (22, 33), (22, 12), (21, 21), (21, 31), (21, 22), (31, 31), (31, 41), (31, 21), (31, 42), (41, 42), (41, 41), (41, 32), (41, 31), (42, 41), (42, 42), (42, 32), (42, 43), (42, 31), (32, 31), (32, 32), (32, 42), (32, 33), (32, 43), (32, 23), (32, 22), (32, 41), (23, 22), (23, 13), (23, 14), (23, 33), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44), (43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 34), (44, 44), (44, 43), (44, 34), (44, 35), (44, 45), (34, 43), (34, 33), (34, 35), (34, 34), (34, 45), (35, 45), (35, 46), (35, 35), (35, 34), (35, 25), (45, 46), (45, 45), (45, 35), (45, 44), (45, 34), (46, 46), (46, 45), (46, 35), (46, 36), (46, 37), (36, 36), (36, 37), (36, 47), (36, 26), (36, 46), (37, 47), (37, 37), (37, 36), (37, 48), (47, 47), (47, 37), (47, 36), (47, 46), (13, 13), (13, 12), (13, 14), (12, 13), (12, 11), (12, 12), (12, 23), (14, 14), (14, 13), (14, 15), (15, 15), (15, 16), (15, 26), (15, 25), (16, 16), (16, 17), (26, 25), (26, 26), (26, 37), (26, 36), (26, 27), (25, 24), (25, 36), (25, 26), (24, 15), (48, 37), (11, 12), (11, 11), (27, 27)]

number of edges: 112

episode is: 50

nodes: [22, 21, 31, 41, 42, 32, 23, 43, 33, 44, 34, 35, 45, 46, 36, 37, 47, 13, 12, 14, 15, 16, 17, 26, 25, 24, 48, 11, 27, 18, 19, 20, 58, 57, 49, 39, 60, 50, 59, 40, 38, 28, 29, 30, 69, 70, 80, 90, 101, 100, 67, 68, 77, 78, 61]

number of nodes: 55

edges: [(22, 21), (22, 23), (22, 22), (22, 32), (22, 33), (22, 12), (22, 13), (22, 11), (21, 21), (21, 31), (21, 22), (21, 11), (21, 12), (21, 32), (31, 31), (31, 41), (31, 21), (31, 42), (31, 32), (41, 42), (41, 41), (41, 32), (41, 31), (42, 41), (42, 42), (42, 32), (42, 43), (42, 31), (42, 33), (32, 31), (32, 32), (32, 42), (32, 33), (32, 43), (32, 23), (32, 22), (32, 21), (23, 22), (23, 13), (23, 14), (23, 33), (23, 34), (23, 24), (23, 23), (23, 12), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44), (43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 34), (33, 23), (33, 22), (33, 42), (33, 24), (44, 44), (44, 33), (44, 43), (44, 34), (44, 35), (44, 45), (34, 43), (34, 33), (34, 35), (34, 34), (34, 45), (34, 23), (34, 44), (34, 25), (34, 24), (35, 45), (35, 46), (35, 35), (35, 34), (35, 25), (35, 44), (35, 36), (45, 46), (45, 45), (45, 35), (45, 44), (45, 34), (45, 36), (46, 46), (46, 45), (46, 35), (46, 36), (46, 37), (46, 47), (36, 36), (36, 37), (36, 47), (36, 26), (36, 46), (36, 45), (36, 35), (36, 27), (37, 47), (37, 37), (37, 36), (37, 48), (37, 46), (37, 27), (47, 47), (47, 37), (47, 36), (47, 46), (47, 48), (47, 38), (47, 57), (47, 58), (13, 13), (13, 12), (13, 14), (13, 22), (13, 23), (12, 13), (12, 11), (12, 12), (12, 23), (12, 22), (12, 21), (14, 14), (14, 13), (14, 15), (14, 23), (14, 25), (15, 15), (15, 16), (15, 26), (15, 25), (15, 14), (16, 16), (16, 17), (16, 15), (16, 26), (17, 17), (17, 18), (26, 25), (26, 26), (26, 37), (26, 36), (26, 27), (26, 15), (26, 16), (25, 24),



(25, 36), (25, 26), (25, 16), (25, 15), (25, 14), (25, 25), (25, 35), (24, 15), (24, 35), (24, 24), (24, 25), (24, 33), (24, 23), (24, 34), (48, 37), (48, 47), (48, 48), (48, 58), (48, 49), (48, 59), (48, 38), (11, 12), (11, 11), (11, 21), (27, 27), (27, 28), (27, 38), (27, 37), (18, 18), (18, 19), (19, 19), (19, 20), (20, 20), (20, 19), (20, 30), (58, 57), (58, 69), (58, 58), (58, 67), (58, 59), (58, 48), (57, 48), (57, 57), (57, 58), (57, 67), (49, 39), (49, 60), (49, 50), (49, 59), (49, 49), (39, 49), (39, 29), (39, 39), (60, 49), (60, 50), (60, 70), (50, 60), (50, 49), (50, 40), (59, 49), (59, 59), (59, 69), (59, 58), (59, 60), (40, 49), (38, 47), (38, 28), (38, 38), (38, 39), (38, 48), (28, 29), (28, 38), (29, 29), (29, 30), (29, 39), (30, 30), (30, 20), (30, 29), (69, 69), (69, 70), (69, 58), (69, 68), (69, 59), (70, 70), (70, 80), (70, 61), (80, 80), (80, 90), (90, 90), (90, 101), (101, 101), (101, 100), (100, 101), (100, 100), (100, 90), (67, 68), (67, 67), (67, 77), (68, 58), (68, 69), (77, 77), (77, 78)]

number of edges: 255  
episode: 100

nodes: [22, 21, 31, 41, 42, 32, 23, 43, 33, 44, 34, 35, 45, 46, 36, 37, 47, 13, 12, 14, 15, 16, 17, 26, 25, 24, 48, 11, 27, 18, 19, 20, 58, 57, 49, 39, 60, 50, 59, 40, 38, 28, 29, 30, 69, 70, 80, 90, 101, 100, 67, 68, 77, 78, 61, 88, 87, 97, 96, 106, 117, 107, 71, 79, 89, 86, 85, 95, 51, 99, 110]

number of nodes: 71

edges: [(22, 21), (22, 23), (22, 22), (22, 32), (22, 33), (22, 12), (22, 13), (22, 11), (22, 31), (21, 21), (21, 31), (21, 22), (21, 11), (21, 12), (21, 32), (21, 20), (31, 31), (31, 41), (31, 21), (31, 42), (31, 32), (31, 30), (31, 40), (31, 22), (41, 42), (41, 41), (41, 32), (41, 31), (42, 41), (42, 42), (42, 32), (42, 43), (42, 31), (42, 33), (32, 31), (32, 32), (32, 42), (32, 33), (32, 43), (32, 23), (32, 41), (32, 22), (32, 21), (23, 22), (23, 13), (23, 14), (23, 33), (23, 34), (23, 24), (23, 23), (23, 12), (43, 32), (43, 43), (43, 42), (43, 33), (43, 44), (43, 34), (33, 33), (33, 44), (33, 32), (33, 43), (33, 34), (33, 23), (33, 22), (33, 42), (33, 24), (44, 44), (44, 33), (44, 43), (44, 34), (44, 35), (44, 45), (34, 43), (34, 33), (34, 35), (34, 34), (34, 45), (34, 23), (34, 44), (34, 25), (34, 24), (35, 45), (35, 46), (35, 35), (35, 34), (35, 25), (35, 44), (35, 36), (35, 24), (35, 26), (45, 46), (45, 45), (45, 35), (45, 44), (45, 34), (45, 36), (46, 46), (46, 45), (46, 35), (46, 36), (46, 37), (46, 47), (36, 36), (36, 37), (36, 47), (36, 26), (36, 46), (36, 45), (36, 35), (36, 27), (36, 25), (37, 47), (37, 37), (37, 36), (37, 48), (37, 46), (37, 27), (37, 38), (37, 26), (47, 47), (47, 37), (47, 36), (47, 46), (47, 48), (47, 38), (47, 57), (47, 58), (13, 13), (13, 12), (13, 14), (13, 22), (13, 23), (13, 24), (12, 13), (12, 11), (12, 12), (12, 23), (12, 22), (12, 21), (14, 14), (14, 13), (14, 15), (14, 23), (14, 25), (14, 24), (15, 15), (15, 16), (15, 26), (15, 25), (15, 14), (15, 24), (16, 16), (16, 17), (16, 15), (16, 26), (16, 27), (17, 17), (17, 18), (17, 16), (17, 28), (17, 27), (17, 26), (26, 25), (26, 26), (26, 37), (26, 36), (26, 27), (26, 15), (26, 16), (26, 35), (26, 17), (25, 24), (25, 36), (25, 26), (25, 16), (25, 15), (25, 14), (25, 25), (25, 35), (24, 15), (24, 35), (24, 24), (24, 25), (24, 33), (24, 23), (24, 34), (24, 14), (48, 37), (48, 47), (48, 48), (48, 58), (48, 49), (48, 59), (48, 38), (48, 39), (48, 57), (11, 12), (11, 11), (11, 21), (27, 27), (27, 28), (27, 38), (27, 37), (27, 18), (27, 26), (27, 17), (18, 18), (18, 19), (18, 17), (19, 19), (19, 20), (19, 29), (19, 18), (19, 28), (20, 20), (20, 19), (20, 30), (20, 29), (20, 21), (58, 57), (58, 69), (58, 58), (58, 67), (58, 59), (58, 48), (58, 68), (58, 49), (57, 48), (57, 57), (57, 58), (57, 67), (57, 68), (49, 39), (49, 60), (49, 50), (49, 59), (49, 49), (49, 58), (49, 48), (39, 49), (39, 29), (39, 39), (39, 38), (39, 30), (39, 40), (60, 49), (60, 50), (60, 70), (60, 60), (60, 69), (60, 59), (60, 61), (50, 60), (50, 49), (50, 40), (50, 61), (50, 50), (50, 51), (59, 49), (59, 59), (59, 69), (59, 58), (59, 60), (59, 68), (40, 49), (40, 40), (40, 39), (40, 29), (40, 30), (40, 31), (40, 50), (38, 47), (38, 28), (38, 38), (38, 39), (38, 48), (38, 37), (38, 29), (38, 49), (28, 29), (28, 38), (28, 28), (28, 39), (28, 18), (28, 19), (29, 29), (29, 30), (29, 39), (29, 19), (29, 28), (29, 38), (29, 40), (30, 30), (30, 20), (30, 29), (30, 31), (30, 40), (69, 69), (69, 70), (69, 58), (69, 68), (69, 59), (69, 78), (69, 80), (69, 79), (70, 70), (70, 80), (70, 61), (70, 71), (70, 60), (80, 80), (80, 90), (80, 79), (80, 70), (80, 69), (80, 89), (90, 90), (90, 101), (90, 79), (90, 89), (101, 101), (101, 100), (100, 101), (100, 100), (100, 90), (100, 110), (67, 68), (67, 67), (67, 77), (67, 57), (68, 58), (68, 69), (68, 68), (68, 67), (68, 78), (77, 77), (77, 78), (77, 67), (77, 87), (78, 88), (78, 77), (78, 68), (78, 69), (61, 61), (61, 60), (61, 50), (61, 70), (88, 87), (87, 97), (87, 86), (97, 96), (96, 106), (106, 117), (106, 107), (117, 106), (107, 107), (71, 71), (71, 70), (79, 79), (79, 68), (79, 80), (79, 69), (79, 78), (89, 90), (89, 99), (86, 85), (85, 95), (51, 60), (99, 99), (99, 100)]

number of edges: 370