Enhancing autonomy and efficiency in goal-conditioned reinforcement learning

Yang, Z.

Chapter 2

Preliminaries

In this chapter, we explain the necessary concepts, definitions, and notations for understanding the entire thesis. Chapter-specific concepts are explained within each chapter. First, we introduce reinforcement learning, a framework used to solve decision-making problems. Then, reinforcement learning will be extended to the goal-conditioned setting, where the agent needs to learn policies conditioned on different goals. In the end, related work on four fundamental phases of the goal-conditioned reinforcement learning framework is briefly discussed.

2.1 Reinforcement learning

Reinforcement learning (RL) is a framework to solve sequential decision-making problems, which are formalized as Markovian decision processes (MDPs). In this thesis, we follow the definition and notations of MDP introduced by Sutton and Barto, 2018. A MDP is a 5-tuple, $< \mathcal{S}, \mathcal{A}, p, r, \gamma >$, where:

- $\mathcal{S}$ is a set of states, called the state space.
- $\mathcal{A}$ is a set of actions, called the action space.
- $p(s'|s,a) \doteq Pr(S_t = s'|S_{t-1} = s, A_{t-1} = a)$ is the transition dynamic, which defines the probability of transitioning from state s to s' after taking the action a at time step t. $s, s' \in \mathcal{S}$ are states and S_t is the state at time t. Similarly, $a \in \mathcal{A}$ is the action and A_t is the action at time t.

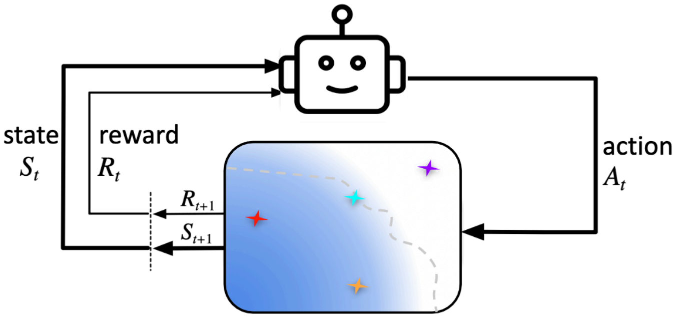

Figure 2.1: The agent-environment interaction in a MDP (Sutton & Barto, 2018). The agent observes the state S_t and takes the action A_t in the environment. The environment returns the next state S_{t+1} and reward R_{t+1}. Then the process repeats.

- $r(s, a, s')$ is the reward function, which produces the intermediate reward $R_t \doteq r(s, a, s'|S_{t-1} = s, A_{t-1} = a, S_t = s')$, i.e. after transitioning from s to s' by taking action a.
- γ is the discount factor ranging from $[0, 1]$.

The agent-environment interaction is shown in Figure 2.1. At time step t, the agent observes the state S_t. Then, according to a policy $\pi : \mathcal{S} \to \Pr(\mathcal{A})$, the agent chooses an action $A_t \sim \pi(A_t|S_t)$. Next, the action A_t is executed and influences the environment to transition from S_t to S_{t+1} based on the transition dynamic p. At the same time, the agent gets a reward R_{t+1}. The agent observes the new state S_{t+1}, then the process repeats. We are interested in the long-term performance of the agent, thus we would like to maximize the cumulative future reward that the agent can get. The cumulative future reward (also called return G) at the time step t can be written as:

$$G_t \doteq R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k+1}. \tag{2.1}$$

We define the value function $V_\pi(s)$ of the state s as the expected return starting from the state s, following the policy π and transition dynamic p:

$$V^\pi(s) \doteq \mathbb{E}_{\pi,p}[G_t|S_t = s]. \tag{2.2}$$

The equation can be written in a recursive format, known as the Bellman equation:

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim p(\cdot|s,a)}[r(s, a, s') + \gamma \cdot V^\pi(s')]. \tag{2.3}$$

The state-action value function $Q^\pi(s,a)$ is defined as the expected return starting from the state s and taking the action a, and following the policy π afterwards, based on the transition dynamic p:

$$Q^\pi(s,a) \doteq \mathbb{E}_{\pi,p}[G_t|S_t = s, A_t = a]. \tag{2.4}$$

Similarly, the state-action value function can be rewritten in a recursive format as well:

$$Q^\pi(s,a) = \mathbb{E}_{s'\sim p(\cdot|s,a)}[r(s,a,s') + \gamma \cdot \mathbb{E}_{a'\sim\pi(\cdot|s')}[Q^\pi(s',a')]]. \tag{2.5}$$

The objective of RL is to find a policy π^* that maximizes the state-action value function:

$$\pi^* \doteq \operatorname*{argmax}_\pi Q^\pi(s,a). \tag{2.6}$$

2.2 Goal-conditioned reinforcement learning

Goal-conditioned reinforcement learning (GCRL) extends RL to a multiple-goal setting to solve goal-conditioned MDP, which is defined as $< \mathcal{S}, \mathcal{A}, p, r, \gamma, \mathcal{G} >$, where $\mathcal{G}$ is a set of goals called the goal space. For any goal $g \in \mathcal{G}$, the reward function will also take the goal g as input, denoted as $r(s,a,s',g)$ (Schaul, Horgan, et al., 2015), which provides the reward for achieving the goal g. We can then rewrite the goal-conditioned state value function and state-action value function in a similar way as Equations (2.3) and (2.5):

$$\begin{aligned} V^\pi(s,g) &= \mathbb{E}_{a\sim\pi(\cdot|s,g),s'\sim p(\cdot|s,a)}[r(s,a,s',g) + \gamma \cdot V_g^\pi(s',g)], \\ Q^\pi(s,a,g) &= \mathbb{E}_{s'\sim p(\cdot|s,a)}[r(s,a,s',g) + \gamma \cdot \mathbb{E}_{a'\sim\pi(\cdot|s',g)}[Q^\pi(s',a',g)]], \end{aligned} \tag{2.7}$$

where π now takes an additional input g to be a goal-conditioned policy. The objective of GCRL is to find a policy π^* that maximizes the state-action value function over all goals (Liu et al., 2022):

$$\pi^* \doteq \operatorname*{argmax}_\pi \mathbb{E}_{g\sim p_g} Q^\pi(s,a,g), \tag{2.8}$$

where p_g represents the desired goal distribution.

As we discussed in Chapter 1, a typical GCRL framework comprises four phases (as detailed in Table 2.1): 1) defining the goal space; 2) selecting an interesting goal for the agent; 3) the agent learning to reach the goal; 4) the agent post-exploring. Each of

Table 2.1: Commonly used methods for each step in the GCRL framework (see Figure 1.1). Representative references are added in the table and more extensive references can be found in the text. GCP is short for goal-conditioned policy and MB is short for model-based.

1. Define the goal space	2. Select a goal	3. Reach the goal	4. Post-explore
State space (Ecoffet et al., 2021; Plappert et al., 2018)	Frontier states (Pitis et al., 2020; Pong et al., 2019)	GCP (Hafner et al., 2023; Schaul, Horgan, et al., 2015)	Random (Ecoffet et al., 2021)
Learned (Eysenbach et al., 2022; Mendonca et al., 2021)	Learning progress (Florensa et al., 2018; Portelas et al., 2020)	MB planning (Hansen et al., 2023)	Intrinsic reward (Sekar et al., 2020; Zhu et al., 2020)
Others (Fu et al., 2019; Jiang et al., 2022)	Others (E. S. Hu et al., 2023; C. Lu et al., 2024)	Others (Blundell, Uria, Pritzel, Li, Ruderman, Leibo, Rae, et al., 2016; Ecoffet et al., 2021)	Others (Klissarov et al., 2023)

these steps has been studied either independently or in conjunction with others. We now provide an overview of commonly used methods for each step. A summary can be found in Table 2.1. It is important to note that while some of these works may not have been explicitly studied within the GCRL framework, they can essentially be adapted and integrated into the GCRL framework with minimal modifications.

2.2.1 Define the goal space

To ensure that the training goals align with the evaluation goals, it is essential to define a goal space that matches or covers the area of interest for evaluation. Typically, evaluation focuses on a sub-space of the entire state space. For instance, in the Fetch-Push task [1], although the state space of the block covers the whole table, the robot is only required to push the block to goals sampled from the center area of the table during evaluation, rather than the entire table. Thus, a pre-defined sub-space of the state space is generally used for sampling goals in tasks with a continuous state space (Plappert et al., 2018; X. Yang et al., 2021; Yu et al., 2019). In tasks with a discrete state space, specific states are selected (e.g. states that can lead to more exploration potential) to form a pool representing the goal space (Ecoffet et al., 2021; Kompella

[1] The Fetch-Push task can be found in https://robotics.farama.org/envs/fetch/push/

et al., 2022).

When dealing with high-dimensional state spaces, such as images, the original state space may contain excessive distracting information. Therefore, a more compact latent space can be learned using techniques such as a self-predictive objective (Hansen et al., 2023; Schwarzer et al., 2020), contrastive loss (Eysenbach et al., 2022; Poudel et al., 2024), or reconstruction loss (Hafner et al., 2023; Mendonca et al., 2021), which can then be utilized as the goal space.

Besides using the original state space or a learned latent space thereof, language itself offers a compact and abstract representation that can be used as a goal space (Chevalier-Boisvert et al., 2024; Fu et al., 2019). Moreover, the integration of multimodal goals, which combines both languages and images (Jiang et al., 2022), can further provide a richer goal space.

Although our thesis does not focus on how to better define a goal space, it is a crucial aspect of the GCRL framework in general.

2.2.2 Select a goal

The exploration of the GCRL agent largely depends on the selected goals, which serve as guidance to indicate the next area of interest for the agent. Once the goal space is defined, the most straightforward approach to select goals from the goal space is uniform sampling, where each goal in the goal space is sampled with equal probability. However, as learning progresses, certain goals may become more relevant than others. For instance, after the agent masters nearby goals, it may be more beneficial to focus on goals that are slightly farther away, but not too distant (e.g. the step 2 in Figure 1.1).

Various strategies have been developed to select goals dynamically. Skew-fit (Pong et al., 2019) and MEGA (Pitis et al., 2020) focus on selecting frontier states as goals by first estimating state density and then choosing states with lower densities. These methods encourage the agent to explore less-visited areas of the state space. AMIGo (Campero et al., 2021) trains two agents in an adversarial fashion: one agent is trained to select challenging goals for another agent, which then attempts to reach these selected goals. This approach ensures continuous improvement for both learning agents, encouraging the goal-selecting agent to select interesting goals progressively. Goal-GAN (Florensa et al., 2018) selects goals based on their difficulty, and goals that are neither too easy nor too difficult are selected for the agent to achieve. Similarly, selecting goals where the agent demonstrates learning progress (Portelas et al., 2020) has also shown to be effective. PEG (E. S. Hu et al., 2023) focuses on goals that offer the most exploration potential, driving the agent toward the most informative parts of the state space.

Recently, foundation models have demonstrated a profound understanding of the real world, characterized by their ability to master a wide range of tasks and domains. Consequently, they can be directly utilized to inform the agent about which goals to pursue (C. Lu et al., 2024).

Unlike the goal selection strategy we propose in Chapter 3, which is designed especially for a more realistic and autonomous RL scenario where the expensive reset is eliminated, the methods discussed here are tailored for the conventional episodic RL setting and often fail in the more challenging reset-free setting. More specifically, these methods cannot deal with over-exploration, a challenge that is introduced by the reset-free setting.

2.2.3 Reach the goal

After the goal is selected, goal-conditioned policies / value functions (Schaul, Horgan, et al., 2015) are usually deployed to learn to reach the given goal. On the other hand, since the process of learning to reach the goal is a decision-making problem formulated as a MDP, any methods that solve an MDP can be used here. Meanwhile, RL methods might not always be the best choice due to their poor performance under certain scenarios, such as low data regime, or in the setting where a dynamic model is given or learned, methods beyond reinforcement learning are also used to fulfill the given goal.

If access to the underlying tasks / environments is available, then the agent can be directly teleported to the selected goal and then post-explores (Ecoffet et al., 2021). It saves a massive amount of learning time but having such oracle access is impossible and unrealistic in real life. When a dynamic model is given / learned, besides training goal-conditioned policies/value functions from synthetic data generated by the dynamic model (Hafner et al., 2023), model-based planning can also be utilized to find the solution to the given goal (Hansen et al., 2023; Schrittwieser et al., 2020). Other methods like imitation learning (Ding et al., 2019; Reuss et al., 2023), evolution strategies (Salimans et al., 2017) or episodic control (Blundell, Uria, Pritzel, Li, Ruderman, Leibo, Rae, et al., 2016; H. Hu et al., 2021) can also be used to learn to reach the goal.

In Chapters 4 and 5, we explore the application of episodic control, a non-parametric approach known for its superior learning speed (Blundell, Uria, Pritzel, Li, Ruderman, Leibo, Rae, et al., 2016), for reaching the goal. We first extend the previous episodic control to tasks with a continuous action space in Chapter 4. Then, to address its inherent limitation (i.e. non-optimality), we propose to combine episodic control with deep RL Chapter 5, achieving both fast learning and better final performance.

2.2.4 Post-explore

When the selected goal or the allocated budget is reached, the agent may post-explore. This differs from the exploration in typical RL paradigms, where the exploration and exploitation are alternated (e.g. ϵ-greedy) or integrated (e.g. adding policy entropy regularization). In contrast, post-exploration is pure exploration, focusing entirely on exploring the environment without considering any task-specific rewards.

Essentially, any exploration approaches can be used for post-exploration. The most straightforward approach is random exploration (Ecoffet et al., 2021; Pong et al., 2019), which, despite its simplicity, has demonstrated strong performance. More sophisticated exploration mechanisms that seek novel states based on intrinsic reward (distinct from extrinsic reward provided by the environment) are also applicable.

For instance, Random Network Distillation (RND) (Burda et al., 2018) encourages the agent to seek novel states by maximizing the prediction error between a fixed randomly initialized network and a predictor network. In this setup, the predictor network is trained to predict the random network's output, with larger prediction errors indicating more novel states. Another method, Intrinsic Curiosity Module (ICM) (Pathak et al., 2017), rewards the agent based on the prediction error between the predicted and the real next state. ICM stimulates the agent to collect states that lead to more accurate prediction, effectively targeting less frequently encountered states. Plan To Explore (P2E) (Sekar et al., 2020) is a model-based exploration method that is trained to maximize the disagreement among outputs of an ensemble of world models, encouraging the agent to visit states that can improve world models the most.

Foundation models, as we described in Section 2.2.3, show profound understandings of the world. They can be used to provide preferences over states (Klissarov et al., 2023), which are then used to guide the exploration in a manner that aligns with the heuristics and insights of the used foundation models.

Whereas the paper that first introduces the idea of post-exploration illustrates that RL agents equipped with post-exploration can solve extremely challenging exploration tasks (Ecoffet et al., 2021), it lacks a direct comparison with agents that do not employ post-exploration. The underlying reasons why post-exploration enhances performance remains unclear. Therefore, in Chapter 6, we conduct a systematic investigation on post-exploration by turning it on and off within the same RL agent and qualitatively analyse why post-exploration facilitates stronger performance.

Four primary steps of the GCRL framework in Figure 1.1 are discussed above and the chapters presented later focus on the step 2, 3 and 4. In Chapter 3, we work on the step 2 where a goal sampling strategy is proposed to tackle the reset-free RL setting. Chapter 4 and Chapter 5 study the step 3, and propose two non-parametric methods to learn the given task faster. Post-exploration (the step 4) is investigated in Chapter 6.