From Single and Short to Multiple and Long: Goal Representation in Reinforcement Learning

Feiyang Xie
Yuanpei College
Peking University
2100017837@stu.pku.edu.cn

Abstract

Goals can be expressed in many ways. For humans, in daily life, goals can be expressed as short sentences or specific future states. As external manifestations of beliefs and desires, goals can guide action selection and are thus critical for planning problems. In reinforcement learning (RL), goal representation has been an active research area. Effective goal representations can enable agents to efficiently solve tasks. However, current goal representation research often focuses on particular tasks and environments, lacking generalizability and versatility. This essay will first explain goal usage in hierarchical RL. Then, it will introduce goal representation methods from simple to complex, discussing their advantages and disadvantages. Finally, it will consider more general goal representations given modern pretrained models.

1 Introduction

Goals, arising from desires and beliefs, are critical for humans facing complex tasks. By guiding action selection, sequences of goals enable completing multi-step tasks. In the field of AI, related research has conducted a lot of exploration on goal representation, and many effective goal representation methods have been proposed. In the field of reinforcement learning, an agent faces the current state and a given task and needs to choose the action to be performed according to its own strategy. However, when faced with complex tasks, the above conditions are often not enough to give enough information required for reasoning. Therefore, many studies try to split the task into a series of goals and introduce the goals information into the agent. Since for humans, goals are abstract concepts, it is difficult for current agents to understand abstract goals, so we need to express the goals concretely. Depending on the task scenarios faced, there are various methods of goal representation, such as using a single future state in the trajectory as the goal, using the reward value as the goal, etc. Different goal representation methods often lead to different modeling methods of the problem. In the following, we will first introduce the general framework for introducing goals in RL problems - hierarchical reinforcement learning (HRL), and introduce how to use goals. In Sec. 3 and Sec. 4, we will introduce various goal representation methods in order from single and short to multiple and long, and analyze their respective advantages and disadvantages. Finally, I will briefly discuss how to build a more general form of goal representation given the variety of pre-trained large models available today.

2 Use goals through HRL

In practice, reinforcement learning struggles with scaling. HRL addresses these challenges using temporal abstraction [1]. HRL typically consists of high-level and low-level policies. The high-level policy decomposes tasks into subtasks and plans subgoals. Compared to standard reinforcement learning, HRL has advantages:

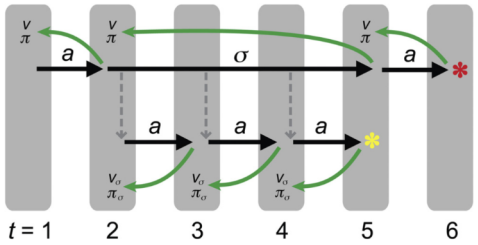

Figure 1: Hierarchical learning dynamics. According to [2].

1. **Sample efficiency**: data generation is often a bottleneck and current RL methods are data inefficient. With HRL, sub-tasks and abstract actions can be used in different tasks on the same domain (transfer learning).
2. **Scaling up**: the application of classic RL to the problems with large action and/or state space is infeasible (curse of dimensionality). HRL aims to decompose large problems into smaller ones (efficient learning).
3. **Generalization**: trained agents can solve complex tasks, but if we want them to transfer their experience to new (even similar) environments, most state of the art RL algorithms will fail. However, hierarchical reinforcement learning can try to stitch different trajectories to obtain a certain generalization ability.
4. **Abstraction**: state and temporal abstractions allow to simplify the problem since resulting sub-tasks can effectively be solved by RL approaches (better knowledge representation).

Formally, we can see HRL as:

$$\pi(a|s) = \int \pi^h(g|s)\pi^l(a|s,g)dg$$

where π^h is the high-level policy and π^l is the low-level policy, g can be any form of goal representation. In HRL, high-level policy understands the dynamics of the environment and the trajectories of completing the task through learning the goal, and guides the actions of lower-level policies by predicting goals. In this understanding, goal is an abstract intermediate representation of environmental information and task information.

3 Single and short goal representation

3.1 Future state as goal

In this part, we mainly discuss the method of using a certain future state in the trajectory as the goal of the current state. Unlike methods that predict actions given the current state, there are many algorithms that use information about future states in both learning and prediction, such as behavior cloning (BC). BC takes current state s_t and next state s_{t+1} as input, the corresponding action a_t as output. So in this case, we can regard a certain future state as the goal representation of the current state. There is a lot of related work in this area. STEVE-1 randomly select timesteps from episodes and use hindsight relabeling to set the intermediate goals for the trajectory segments [3]. PTGM clusters the states in the data set to form a codebook, and uses the code that is most similar to a certain future state as the goal [4].

Figure 2: STEVE-1's goal representation. According to [3].

The advantages of a certain future state as goal representation are that it is easy to implement and very intuitive, and it can show good results when facing most tasks. However, the disadvantage is that

this method is very dependent on the data set. When the trajectories in the data set are sub-optimal trajectories, it is difficult for this method to learn better strategies and has weak generalization.

3.2 Reward as goal

In RL, a very important link is to learn the value function, Q-function and policy based on rewards. Therefore, sparse rewards often make it difficult to learn a good policy. Since most tasks in real life are sparsely rewarded, it is very important to design algorithms that can learn good policies under sparse reward conditions. Therefore, some research work attempts to use the reward itself as a representation of the goal. Decision transformer (DT) uses sequence modeling methods to solve reinforcement learning problems. [5] DT uses return-to-go in the current state as the target representation of the current state, and DT can be viewed as learning a model that predicts what action should be taken at a given state in order to make so many returns. The expectation of this modeling method is that the reward corresponding to each trajectory is used as the goal during training, and a larger goal is artificially set during inference, so that the model can stitch different trajectories together to obtain better performance.

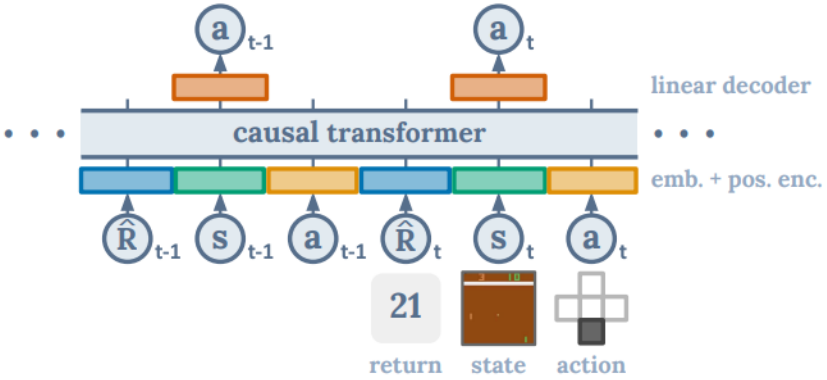

Figure 3: Decision transformer's goal representation. According to [5].

However, in experiments, a notable limitation of DT is its reliance on recalling trajectories from datasets, losing the capability to seamlessly stitch sub-optimal trajectories together. Obviously, the disadvantage of using reward as goal is that reward-to-go contains too little information, so memorizing trajectory information is still not enough for obtaining the stitching ability. The advantage of this method is that When the quality of the data set is high, this method can simply handle the problem as sequence modeling and can perform better.

4 Multiple and Long goal representation

4.1 Future trajectory as goal

Since single and short goal representation is difficult to provide sufficient information to guide action selection, multiple and long goal representation has become a research trend. Given a trajectory $T = (s_t, a_t, s_{t+1}, a_{t+1}, \dots)$, this method maps the subsequent trajectory of the current state s_t into a hidden state and uses this hidden state as the goal representation of the current state. Some studies use future k action sequences as goal representations and understand such goals as skills [6]. In inference, the prior learned during training is used to guide the selection of subsequent actions. MineClip uses the embedding of 16 frames as the goal representation and matches it with the language, so that the similarity can be used as a reward function to help RL training [7].

The advantage of using future trajectory as goal is that this goal representation can contain richer information than other methods, and this goal can be understood as a skill with stronger generalization ability in similar environments. Again, this representation has a number of disadvantages. First of all, this kind of goal is based on a data set. The quality of the data set determines to a large extent whether the goal can be effective. Secondly, the interpretability of this hidden state is weak. When facing complex tasks and a larger goal space, it is difficult for high-level strategies to learn how to predict better goals in the current state.

4.2 Prompt as goal

In current large language models, prompt often determines the output content and quality of the model. Due to the autoregressive structure of the transformer, we can also regard this prompt as a goal representation. In RL, some research work also uses prompt as the goal of high-level policy. DEPS uses a LLM-based method for planning, that is, the output of LLM is used as the goal of the current state [8]. Prompt decision transformer (PDT) is also another research direction that uses prompt as goal. Some studies show that PDT is good at few-shot policy generalization [9].

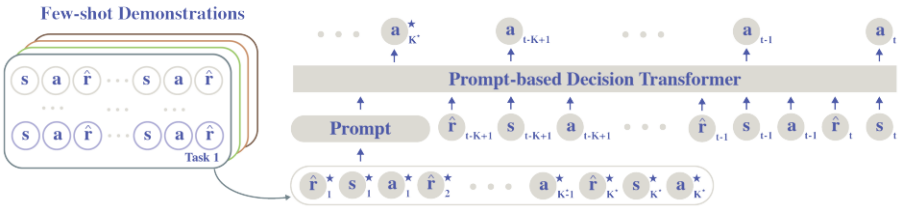

Figure 4: Prompting decision transformer's goal representation. According to [9].

As the main way of human communication, the goals expressed using language are more abstract and can contain more information. And due to the composability and versatility of the language, this method is highly generalizable and can be used in different environments. Due to the strong capabilities of LLMs, using LLM as a planner can further improve the agent's capabilities. The disadvantage of this method is that when faced with an expert domain or a specific environment, it is difficult for current LLMs to give a prompt that can correctly complete the task. In addition, due to the diversity and complexity of prompts themselves, it is difficult for lower-level policy to learn the correspondence between a given prompt and future actions.

5 Conclusion

This essay examined four goal representation approaches: future states, rewards, future trajectories, and prompts. Analyzing their tradeoffs shows a trend from simple, short goals toward complex, long ones. More goal information expands the goal space and increases abstraction. Overly simple goals inadequately guide low-level policies, while overly complex goals impede learning. Future methods may use more language-based goals, given language's power and generalization. Modeling beliefs, desires, and goals holistically could also enable solving more complex tasks.

References

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

[2] Jose JF Ribas-Fernandes, Alec Solway, Carlos Diuk, Joseph T McGuire, Andrew G Barto, Yael Niv, and Matthew M Botvinick. A neural signature of hierarchical reinforcement learning. *Neuron*, 71(2):370–379, 2011. 2

[3] Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *arXiv preprint arXiv:2306.00937*, 2023. 2

[4] Anonymous. Pre-training goal-based models for sample-efficient reinforcement learning. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. under review. 2

[5] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021. 3

[6] Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pages 188–204. PMLR, 2021. 3

[7] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge, 2022. 3

[8] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents, 2023. 4

[9] Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pages 24631–24645. PMLR, 2022. 4