

# Goal-Conditioned Reinforcement Learning: Problems and Solutions

Minghuan Liu\*, Menghui Zhu\*, Weinan Zhang<sup>†</sup>

Shanghai Jiao Tong University

{minghuanliu,zerozmi7,wnzhang}@sjtu.edu.cn

## Abstract

Goal-conditioned reinforcement learning (GCRL), related to a set of complex RL problems, trains an agent to achieve different goals under particular scenarios. Compared to the standard RL solutions that learn a policy solely depending on the states or observations, GCRL additionally requires the agent to make decisions according to different goals. In this survey, we provide a comprehensive overview of the challenges and algorithms for GCRL. Firstly, we reveal the basic problems studied in this field. Then, we explain how goals are represented and present how existing solutions are designed from different points of view. Finally, we conclude and discuss potential future prospects that recent researches focus on.<sup>1</sup>

## 1 Introduction

One of the remarkable behaviors of human intelligence is that humans learn to reach one-by-one goals. Such ability allows generalizing their skills across different tasks while learning effectively. Building upon the recent successes of deep reinforcement learning (RL), which aims to induce a unique optimal solution for a specific task [Sutton and Barto, 2018], researchers extend the problem towards tackling multiple goals simultaneously [Plappert *et al.*, 2018], known as **goal-conditioned RL**, or **goal-oriented RL**, **multi-goal RL**, and we will refer as GCRL in this paper for brevity. Technically, goal is a core data structure widely used in many related domains, for describing either achieving or maintaining some condition, e.g., **goal reasoning** [Roberts *et al.*, 2018; Niemueller *et al.*, 2019] and **robotics** [Rolf and Steil, 2013; Santucci *et al.*, 2016]. In this paper, we mainly focus on the usage of goals in RL literature. Different from standard RL, GCRL augments the observation with an additional goal that the agent is required to achieve when making a decision in an episode [Schaul *et al.*, 2015; Andrychowicz *et al.*, 2017].

The reward function is known and usually defined as a binary bonus of reaching the goal. After training, the agent should be able to achieve arbitrary goals as the task specifies. Such a problem raises additional challenges, for example, the simple but sparse reward function and the sample efficiency for multi-task learning.

From the task perspective, various solutions are designed for different categories of goals. A typical goal space is a sub-space of the state space, such as images or feature vectors, which is natural for describing the desired condition that the agents are required to meet. In addition, language, rewards, commands and intrinsic skills are also used as the representation of goals for different kinds of tasks.

From the learning perspective, researchers have proposed bags of tricks to alleviate the challenges in learning efficiency and generalization ability. Specifically, during a training loop, the agent first receives a desired goal from the environment or it should choose its own goal that is easier to achieve at the current stage [Ren *et al.*, 2019; Florensa *et al.*, 2018]. Then to pursue the goal, the agent interacts with the environment to generate trajectories [Trott *et al.*, 2019]. The sampled trajectories are further post-processed for the training, for example, Andrychowicz *et al.* [2017] chose to relabel the goals to increase the effectiveness of the samples. The final step in the loop is the optimization of the policy and value networks, where the goal-augmented extensions of the state and state-action value function are proposed as the foundation for solving the goal-conditioned policy [Schaul *et al.*, 2015].

This survey focuses on the algorithms designed for GCRL and is organized as follows:

1. **Background and basic challenges.** We firstly formulate the basic problem in GCRL, and list all the challenges that should be addressed (Section 2);
2. **Goals representations.** We discuss how the goals are defined and represented in different task specifications (Section 3);
3. **Solutions towards learning phases.** We categorize existing researches from the perspective of different phases, explain how the challenges of GCRL are tackled from different points of views (Section 4);
4. **Future prospects.** We summarize and present future prospects that are potential and expected to be studied in this field (Section 5).

\*Equal contribution.

<sup>†</sup>Corresponding author.

<sup>1</sup>We collect and summarize related algorithms and widely used benchmark environments in <https://github.com/apexrl/GCRL-Collection>.

## 2 Problems and Challenges

### 2.1 Standard RL Problems as MDPs

Standard RL tasks are usually modeled as Markov Decision Process (MDP)  $\mathcal{M}$ , denoted as a tuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma, \rho_0 \rangle$ , where  $\mathcal{S}$ ,  $\mathcal{A}$ ,  $\gamma$  and  $\rho_0$  denote the state space, action space, discount factor and the distribution of initial states, respectively.  $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$  is the dynamics transition function, and  $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is the reward function. Formally, the problem requires the agent to learn a decision-making policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  to maximize the expected cumulative return:

$$J(\pi) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot | s_t) \\ s_{t+1} \sim \mathcal{T}(\cdot | s_t, a_t)}} \left[ \sum_t \gamma^t r(s_t, a_t) \right]. \quad (1)$$

### 2.2 Goal-Augmented MDPs

Standard RL only requires the agent to finish one specific task defined by the reward function while GCRL focuses on a more general and complex scenario. As depicted in Fig. 1, the agent in GCRL is required to either learn to master multiple tasks simultaneously, or decompose the long-term and hard-reaching goals into intermediate sub-goals while learning to reach them with a unified policy, or achieve both of them. For example, to navigate a robot or manipulate an object, the goals are typically defined as the target positions to reach. In Section 3, we will further discuss the detailed representation of goals case-by-case.

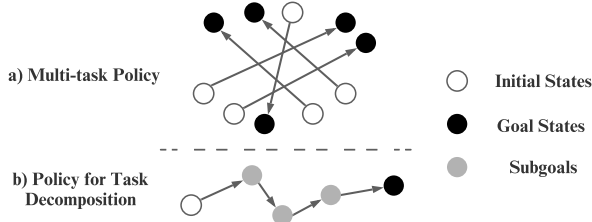


Figure 1: The mission of policy in complex RL problems. a) Learn to achieve multiple tasks with one single policy; b) Decompose the long-term, hard-reaching goals into easily obtained sub-goals.

To tackle such a challenge, the formulation of GCRL augments the MDP with an extra tuple  $\langle \mathcal{G}, p_g, \phi \rangle$  as a goal-augmented MDP (GA-MDP), where  $\mathcal{G}$  denotes the space of goals describing the tasks,  $p_g$  represents the desired goal distribution of the environment, and  $\phi : \mathcal{S} \rightarrow \mathcal{G}$  is a tractable mapping function that maps the state to a specific goal. For many scenarios,  $\mathcal{S}$  and  $\mathcal{G}$  can be the same, in which case  $\phi$  is an identity function. In GA-MDP, the reward function  $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$  is defined with the goals, and therefore the objective of GCRL is to reach goal states via a goal-conditioned policy  $\pi : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \rightarrow [0, 1]$  that maximizes the expectation of the cumulative return over the goal distribution:

$$J(\pi) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot | s_t, g), g \sim p_g \\ s_{t+1} \sim \mathcal{T}(\cdot | s_t, a_t)}} \left[ \sum_t \gamma^t r(s_t, a_t, g) \right]. \quad (2)$$

To better understand the agent behaviors in a GA-MDP, some common definitions of goal used in previous works [Pitis et al., 2020; Zhu et al., 2021], are listed below.

**Definition 1** (Desired Goal). *A desired goal is a required task to solve which can either be provided by the environments or generated intrinsically.*

**Definition 2** (Achieved Goal). *An achieved goal is the corresponding goal achieved by the agent at the current state.*

**Definition 3** (Behavioral Goal). *A behavioral goal represents the targeted task for sampling in a rollout episode.*

We note that the behavioral goal should be equivalent to the desired goal if there is no sub-goal replacement.

### 2.3 Basic Challenges

The mission of the agent as mentioned in Fig. 1 can be summarized as learning a multifunctional policy. In the sequel, the basic challenges are mainly the problem of generalization and sample efficiency.

#### Generalization: Learning over Multiple Tasks

In multi-task learning problems, denoting the loss function of the task  $i$  as  $L_i$ , the weighted objective for learning  $k$  tasks with a function  $f$  simultaneously is defined as

$$L(f) = \sum_{i=1}^k w_i \cdot L_i(f). \quad (3)$$

This requires generalization and feature sharing across tasks that boost the learning efficiency of the function. If we further regard the weight  $w_i$  for each task  $i$  as the occurring probability such that  $i \sim p$  and  $p(i) = w_i$ , we can further rewrite Eq. (3) as an objective of expectation:

$$L(f) = \mathbb{E}_p[L_i(f)]. \quad (4)$$

Notice that this generalizes the objective Eq. (2) of GCRL. Correspondingly, the task variety  $p$  is exactly the desired goal distribution  $p_g$ ; and different goals  $g \sim p_g$  implies different reward functions under the same environment. Since the agent is required to learn a unified policy to perform under multiple goals simultaneously, one of the main challenges in a complex RL problem can be naturally regarded as a multi-task learning problem towards generalization. And in result, we get to know that GCRL focuses on **multi-task learning** where the task variation comes only from the difference of the reward function induced by different desired goals under the same dynamics.

Fortunately, different goals in the same environment always possess similar properties (e.g., the dimension, the structure, and even the way to compute rewards are the same). This makes it possible for the goal-conditioned policy to share parameters across various tasks (i.e., goals  $g$ ) to derive a generalized solution in GCRL.

#### Sample Efficiency: Towards Sparse Rewards

In GA-MDP, the reward function  $r$  is typically a simple unshaped binary signal, indicating whether the task is completed, i.e.,

$$r_g(s_t, a_t, g) = \mathbb{1}(\text{the goal is reached}), \quad (5)$$

where  $\mathbb{1}$  is the indicator function. The principle for completing the goal is usually a known criteria, for example, the distance between the achieved goal and the desired goal is less

than a small number  $\epsilon$  [Andrychowicz et al., 2017]:

$$\mathbb{1}(\text{the goal is reached}) = \mathbb{1}(\|\phi(s_{t+1}) - g\| \leq \epsilon). \quad (6)$$

Therefore, without shaping, the reward is rather sparse, so the agent can gain nothing from unsuccessful episodes. A straightforward idea to prevent the problem is to reshape the reward with a distance measure  $d$  between the achieved goal and the desired goal without any domain knowledge [Trott et al., 2019], i.e.,

$$\tilde{r}_g(s_t, a_t, g) = -d(\phi(s_{t+1}), g). \quad (7)$$

Such reshaped signals are so dense that the sparse problem can be alleviated in many cases. However, it may cause additional local optima challenges due to the structure of the environment. For instance, the reshaped reward prevents learning from a behavior that the agent must first increase the distance to the goal before finally reaching it [Trott et al., 2019].

### 3 What are Goals?

In this section, we characterize the basic element in GCRL – what exactly the goals are, and their typical representations. Indeed, there is no unique answer for the structure of goals, which highly depends on the task specifications. For research convenience, we further summarize the widely-used benchmark environments from previous works in Tab. 1.

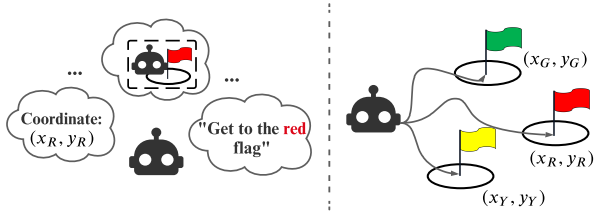


Figure 2: Typical representations of goals in GCRL: vectors, images and languages.

#### 3.1 Desired Property and Feature Vectors

Typically, goals are defined as desired properties or features of an agent. A task is therefore asking the agent to achieve some states that suffices specific features, usually represented as vectors. A wide variety of simulated robotics control tasks [Plappert et al., 2018] are commonly used benchmarks for evaluating vector GCRL algorithms. For instance, asking the robotic arm to push, pick, or slide a block to a specific location, like *FetchPush*, *FetchPickAndPlace* and *FetchSlide*. Additionally, a robotic hand is required to manipulate the block, or the pen to a particular position and rotation, e.g., *HandManipulateBlock* and *HandManipulatePen*. In such cases, the representation of the goal is a set of vectors of the desired location. On the other hand, setting the goal as a desired moving speed and ask the cheetah agent to run up at the speed (*HalfCheetah*) [Zhu et al., 2021]; or a vector of the target position to reach, as is used in *AntLocomotion* [Florensa et al., 2018], *Reacher* [Tang and Kucukelbir, 2021], and *N-D Mass* [Fang et al., 2019; Pitis et al., 2020].

For tasks with vector goals, the goal space is usually a subspace of the state space. In other words, the mapping function  $\phi$  is either an identical function such that matching a goal is exactly matching a target state, or a dimension reduction function that choosing specific dimensions from the state.

#### 3.2 Image Goals

Comparing to vectors describing simple properties and features, image is a more straightforward representation for complex tasks, especially in practical applications. Researchers used to evaluate their image algorithms on simulation environments, as in [Warde-Farley et al.; Lee et al.; Chane-Sane et al. [2019; 2020; 2021], and also in real-world robotics applications [Nair et al., 2018b; Nair et al., 2020b].

Simulation environments include the robotics benchmarking tasks discussed in Section 3.1 with image state / goal space instead; moreover, DeepMind Lab [Beattie et al., 2016], Deepmind Control Suite (DMC) [Tassa et al., 2020], Atari games [Brockman et al., 2016] and Minecraft [Johnson et al., 2016] are also popular testbeds. In these tasks, the goal is either the image of the target position to approach, or a imagined image of winning the game. Beyond simulation, researchers also bring GCRL into real-world applications. For instance, [Nair et al.; Nair et al. [2018b; 2020b]] first simulated a 7-dof Sawyer arm to reach goal positions, and then applied the agent to real world robotics control tasks with camera images.

To solve image tasks, the state and goal images can be directly fed into the policy and value networks and trained in an end-to-end manner, as in [Chane-Sane et al.; Eysenbach et al.; Trott et al. [2021; 2019b; 2019]. However, due to the high dimensionality of images, a more general way is to work on a compact vector representation, using an encoding model  $E$  to map the image observation  $o$  to a latent goal  $g_z = E(o)$ . A common choice of the encoder is variational auto-encoder (VAE) [Nair et al., 2018b; Khazatsky et al., 2021]; or a context-conditioned VAE [Nair et al., 2020a] for generalizing to different environments. The encoding can also be derived via training a weak supervised model for learning disentangled representation [Lee et al., 2020]; and even an end-to-end supervised imitation learning objective [Srinivas et al., 2018]. In such tasks, the goal mapping function is an identical function since the state and goal are in the same image space.

#### 3.3 Language Goals

Natural languages are also used to represent the goals. In fact, language-conditioned RL is a wide research direction, which combines the techniques of GCRL and the development of neural language processing (NLP). Since our work focus on the essential of GCRL, we will only cover how language forms the goals, and readers may refer to [Luketina et al., 2019] for a detailed review of language usage in RL.

The advantage of language goals, i.e., sequences of discrete symbols, is that they can easily be understood by human. In most cases, the goal is an instruction sentence containing explicit verbs and objects, for instance, “go to blue torch” [Chan et al., 2019]; or question words “there is a red ball; are there any matte cyan sphere right of it?” [Jiang et al.,

Environment / Task	Benchmark Type	Description
FetchReach	Gym Robotics/Mujoco	A robotic arm reaches a target position
FetchPush	Gym Robotics/Mujoco	A robotic arm pushes a block to a target position
FetchPickAndPlace	Gym Robotics/Mujoco	A robotic arm first picks up the block then places it to a target position
FetchSlide	Gym Robotics/Mujoco	A robotic arm slides a block to a goal position
HandManipulateBlock	Gym Robotics/Mujoco	Orient a block using a robot hand
HandManipulateEgg	Gym Robotics/Mujoco	Orient an egg using a robot hand
HandManipulatePen	Gym Robotics/Mujoco	Orient a pen using a robot hand
HalfCheetah	Mujoco	Make a 2D cheetah robot run and keep a specific speed
AntLocomotion / AntMaze	Mujoco	Make a 3D four-legged robot walk to a target position
Sawyer	Mujoco	A sawyer robot reaches a target position
Reacher	Mujoco	A reacher robot reaches a target position
N-D Mass	Mujoco	A ball reaches a target position
DoorOpening	Mujoco	A sawyer robot opens a door by latching onto the handle
Dy-Reaching	Mujoco	A robotic arm reaches a target position moving along a straight line
Dy-Circling	Mujoco	A robotic arm reaches a target position moving along a circle
Dy-Pushing	Mujoco	A robotic arm moves the box to a moving target position
Dy-Pouring	Mujoco	A robotic arm grips the can and pours the water into a cup
WaterMaze	DeepMind Lab	Get to a target position described by an image
Seaquest	Atari	Get to a target position described by an image
Montezuma’s Revenge	Atari	Get to a target position described by an image
Reacher	DeepMind Control Suit	A two-link planar reacher poses an aimed shape
Manipulator	DeepMind Control Suit	A 3-DoF finger robot rotates the body into a target orientation
Finger	DeepMind Control Suit	A planar manipulator brings an object in a target location
Place Object in Tray	PyBullet	A robotic arm puts toys into a tray
Opening Drawer	PyBullet	A robotic arm opens the drawer
PickandPlace	PyBullet	A robotic arm first picks up the block then places it to a target position
Five Object Manipulation	CLEVR-Robot/Mujoco	A point mass agent arranges, orders and sorts 5 fixed objects
Diverse Object Manipulation	CLEVR-Robot/Mujoco	A point mass agent orders objects in different shapes and colors
Playground	Playground	A 2D hand moves and grasps/releases different objects
VizDoom	ViZDoom	AI bots reach specific place in the first person shooter 3D game Doom

Table 1: Summary of widely used benchmark environments. Note almost all environments support various kinds of goal structures, i.e., vector, image, language, etc. More detailed information can be referred to the [repository](#) we maintained.

2019]; and simply, a predefined set of semantic predicates, like “close”, “above” [Akakzia *et al.*, 2021]. To train RL algorithms, those words must be either translated into embeddings via pre-training models [Chan *et al.*, 2019], or taken into a recurrent neural network (RNN) as learning the embedding jointly with the RL agent [Colas *et al.*, 2020a]. If the language space is constrained such that the discrete predicates or objects are limited, a simple semantic mapping function can be learned to translate the handcrafted encodings into vectors [Akakzia *et al.*, 2021]. In this case, the goal mapping is a function from the state space to the embedding space.

### 3.4 Other Forms of Goals

Not limited in forms above, researchers have used the conditioned policy on various goal types, which may lead to different unresolved problems.

**Expected return as goals.** Kumar *et al.* [2019] proposed to learn a return-conditioned policy  $\pi(a|s, R)$  where  $R$  is an estimated return of the whole trajectory. The basic idea is to reuse sub-optimal trajectories for training as these trajectories are optimal supervision for a policy of certain quality,

and thus the **return  $R$  can be seen as a desired goal to reach**. Specifically,  $R$  is modeled as a dynamic distribution as the learning proceeds. Recently, Chen *et al.* [2021] proposed decision transformer (DT) to solve offline RL problems, which treats RL as a sequence modeling problem by generating actions from offline datasets. In detail, the trajectories are represented as a sequence of  $(\hat{R}_1, s_1, a_1, \dots, \hat{R}_T, s_T, a_T)$ , where  $\hat{R}_t = \sum_{t'=t}^T r_{t'}$  is the target reward that requires the agent to reach, and resembles the desired goal in GCRL.

**Upside-down RL: commands as goals.** Concurrently, Schmidhuber [2019] and Srivastava *et al.* [2019] also exploited reward-form goals on standard RL problems. Different from the works discussed above, they **regard the goals as a set of commands**, such as asking the agent to reach a desired return  $R$  in a desired horizon  $H$ , leading to a conditioned policy  $\pi(a|s, R, H)$ . These works do not adopt any training techniques from RL literature, but learn the policy from replay buffer in a supervised style instead.



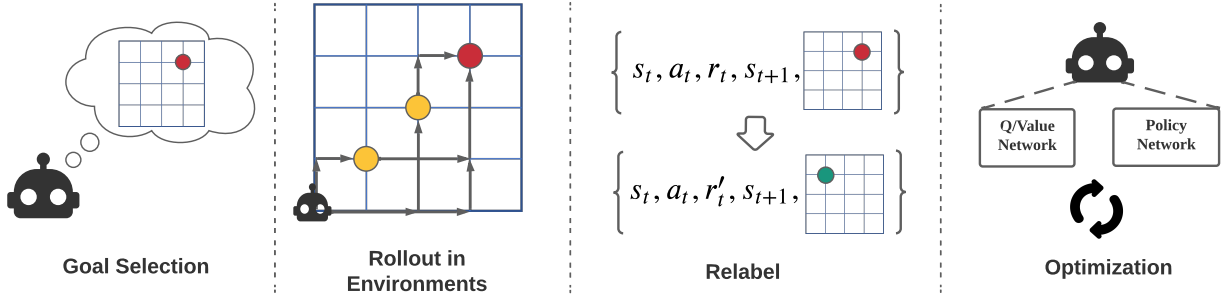


Figure 3: Typical phases in a GCRL algorithm. The agent is provided (or selects by itself) a behavior goal as the target, then interacts with the environments and collects experiences. Before optimizing the policy, the historical data can be relabeled by changing the desired goals.

## 4 Algorithms to GCRL

In this section, we discuss the solutions for GCRL tasks from the major perspectives of algorithms, including optimization, sub-goal selection and relabeling, as shown in Fig. 3, which is also summarized in Tab. 2.

### 4.1 Optimization

The essential difference between GCRL and traditional RL algorithms on the stage of optimization is learning a conditioned policy towards a set of goals. Therefore, both universal and specific solutions are proposed for resolving GCRL.

**Universal value function.** To solve GCRL, [Schaul et al. \[2015\]](#) first proposed the universal value function approximation (UVFA), which provides a straightforward way to characterize Goal-Augmented MDP (GA-MDP) by generalizing the standard value function as a goal-conditioned value function, and is defined as

$$V^\pi(s_t, g) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot | s_t, a_t) \\ s_{t+1} \sim \mathcal{T}(\cdot | s_t, a_t)}} \left[ \sum_t \gamma^t r(s_t, a_t, g) \right]. \quad (8)$$

UVFA is widely applied in the following research works serving as a basic framework.

**Reward shaping.** One of the basic challenges of GCRL is the extremely sparse reward, so a corresponding solution should be reward shaping. In the previous context, we have explained that a lazy distance-based shaping method can lead to additional local optima. To alleviate the problem, [Trott et al. \[2019\]](#) proposed a **sibling rivalry (SR) mechanism**, which samples two sibling rollouts simultaneously, uses each others' achieved goals as anti-goals  $\bar{g}$  and reward an anti-goal bonus  $d(s, \bar{g})$  to improve exploring, where  $d$  is the distance metric, such as L2-norm. In addition, many works proposed different distance measures for evaluating the progress of achieving goals. For instance, [Hartikainen et al. \[2019\]](#) calculated the average number of actions required to taken to reach the goal and used it as the penalty on reward; [Wu et al. \[2018\]](#) learned the Laplacian matrix then used its eigenvectors as the representation of the states and goals, and the reward function is defined as the Euclidean distance between the state and goal vectors; [Durugkar et al. \[2021\]](#) defined the target distribution in GCRL as a Dirac measure at the goal state and estimated Wasserstein-1 distance of the state visitation distribution to the target distribution as the reward bonus.

In order to encourage exploration, [McCarthy and Redmond \[2021\]](#) trained an ensemble of forward models and adopted the variance between the ensemble predictions of the next state as the intrinsic reward.

**Self-imitation learning.** Although reaching a specific goal state is agnostic to the agent at the beginning of training, we can treat any trajectory as a successful trial for reaching its final state. Drawn intuition from such an observation, [Ghosh et al. \[2021\]](#) proposed a supervised algorithm named goal-conditioned supervised learning (GCSL), which learns the policy by maximizing the likelihood of the actions for a reached goal within a number of steps:

$$J(\pi) = \mathbb{E}_{\mathcal{D}} [\log \pi(a|s, g, h)] , \quad (9)$$

where  $\mathcal{D}$  is the replay buffer and  $h$  represents the left steps (horizon) to reach the goal. Similar to the idea of GCSL, [Nair et al. \[2018a\]](#) filtered the actions from demonstrations by Q values and added a supervised auxiliary loss on the RL objective to improve the learning efficiency.

**Planning and model-based RL.** In RL tasks, agents must interact with the environment to collect experience for policy learning, which results in considerable sampling costs. However, as humans have the ability to infer future outcomes in mind, agents can also keep an environment dynamics model for planning or imagining with less sampling in the environment. Formally, the probabilistic parameterized dynamics model  $M_\theta$  can be trained from experience data  $\mathcal{D}$  by supervised objectives, e.g., by maximum likelihood estimation:

$$\arg \max_{\theta} \mathbb{E}_{\{s_t, a_t, s_{t+1}\} \sim \mathcal{D}} [\log \Pr(s_{t+1} | s_t, a_t; \theta)] . \quad (10)$$

Generally speaking, algorithms utilizing a dynamics model are realized as model-based methods. The typical usage of the model contains planning and generating more interaction data for training to improve the sample efficiency. For example, [Nair et al. \[2020b\]](#) tried to plan to reach the goal states in the latent space for image-based inputs; they trained an encoder-decoder model and a goal-conditioned dynamics model on the latent space to learn the state representation function. [Charlesworth and Montana \[2020\]](#) utilized a dynamics model for predicting consecutive states to regularize the consistency of the trajectories generated by a GAN model. With an ensemble of well-trained GANs, the agent generate

trajectories and choose the next actions from the trajectory that spends less time achieving the goal. Besides, to decrease the sampling cost, [Zhu et al. \[2021\]](#) brought model-based optimization techniques into GCRL while also planning in the learned model to infer future goals for relabeling; [Zhang et al. \[2021\]](#) learned a representation model from the goal space to a compact latent space in a self-supervised manner, and tried to find a sparse set of latent landmarks scattered across goal space in terms of reachability by clustering and planning with the landmarks; [Mendonca et al. \[2021\]](#) took an ensemble of dynamics model for training an explorer to discover unseen surprising states by defining exploration bonus as the variance of the ensemble predictions, and they further train the policy to reach randomly sampled goals from the replay buffer.

## 4.2 Sub-goal Selection / Generation

In most environments, the desired goal asked by the task for the agent is a long-term target and hard to achieve. Nevertheless, building on the insight that humans tend to split the long-term target into the short-term goals that are easier to reach, researchers have proposed to generate or select sub-goals to help the agent reach the target goals. Formally, **collecting experience following a replaced desired goal is changing the goal distribution of the environment from  $p_g$  into  $f$** , and the optimization in Eq. (2) is now rewritten as

$$J(\pi) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot | s_t, g), g \sim f \\ s_{t+1} \sim \mathcal{T}(\cdot | s_t, a_t)}} \left[ \sum_t \gamma^t r(s_t, a_t, g) \right], \quad (11)$$

where  $f$  can either be specific rules that select  $g$  from past experience, or a learned function depending on the principle.

**Intermediate difficulty.** Accurately evaluating the difficulty to reach a goal helps determine what to learn within the capability of the agent. An appropriate goal should be just beyond the scope of the agent in order to improve learning efficiency. Accordingly, the intermediate difficulty can be used as a critic principle for choosing sub-goals. For instance, [Florensa et al. \[2018\]](#) scored the difficulty with the received reward, which is used to label the goals as positive/negative samples; then, they trained a GAN model to generate goals with appropriate difficulty scores. [Campero et al. \[2021\]](#) proposed a **teacher-student** framework, where the **teacher serves as a goal producer that tries to propose increasingly challenging goals. The teacher learns from the student’s performance on the produced goal, while the student learns from both extrinsic rewards from the environment and intrinsic rewards from the teacher.** Moreover, [Zhang et al. \[2020\]](#) treated the epistemic uncertainty of the Q-function as a measurement of the difficulty for reached goals, which is further used for constructing a distribution to sample goals. When the uncertainty is high, the goal should be lied at the knowledge frontier of the policy, and is thus at an appropriate difficulty to learn.

**Exploration awareness.** Notice that in GCRL, the essential challenge making desired goals hard to reach is the sparse reward function which limits the learning efficiency of the agent. Therefore, we should boost its exploration ability to cover unseen goals and states as many as possible, so that the agent can learn far more from experience replay. To this

end, in GCRL, many researchers propose to sample with the sub-goals that lead to a better exploring ability. [Pitis et al. \[2020\]](#) proposed to sample the goals which could maximize the entropy of the historically achieved goals, and in result, enhance the coverage of the goal space; [Pong et al. \[2020\]](#) trained a self-supervised generative model to generate behavioral goals that are skewed to be approximately uniform over achieved goals from past experience, which is shown as equivalent to maximizing the entropy of the desired goal distribution. Likewise, to provide with reachable and diverse goals, [Warde-Farley et al. \[2019\]](#) and [Nair et al. \[2018b\]](#) both chose to sample past achieved goals uniformly from experience replay as the behavioral goals, while [Warde-Farley et al. \[2019\]](#) sampled goals from a diversified replay buffer in which the distance between the stored goals are held as far as possible, and [Nair et al. \[2018b\]](#) generate goals by fitting a generative model for historical achieved goals. Furthermore, [Ren et al. \[2019\]](#) generated a set of hindsight goals for exploration by formulating a Wasserstein Barycenter problem. The generated sub-goals serve as an implicit curriculum that is effective to learn using the current value function. [Hartikainen et al. \[2019\]](#) chose to learn a distance function and take the goals from the replay buffer which have maximum distance from the initial state for exploration efficiency.

**Searching from experience.** Looking back upon past experience, both humans and agents can benefit since the history contains the way-points leading us to reach some goals. Motivated by such an idea, [Lai et al. \[2020\]](#) chose  $k$  critical middle point goals from a sampled episode, and learned an RNN to generate the sequence given the starting and the ultimate point; thereafter, they apply the generated middle points as a sequence of desired goals when interacting with the environment. On the other hand, [Eysenbach et al. \[2019b\]](#) built a graph on the replay buffer, taking the states as nodes, and distance from the start state to the goal state as the edge weights. In their work, they utilize the intuition that the value of states can approximate the distance to goals under the binary reward function (Eq. (5)); then, the proposed algorithm runs a graph search to find the sequence of way-points to achieve the target and takes actions by a learned policy conditioned on the way-points one after another.

**Model-based planning.** Human has the ability to plan a long-term goal by decomposing it into several short-term segments. To this end, [Nair and Finn \[2020\]](#) trained a dynamics model for planning a sequence of sub-goals from the initial state to the final target in image-based tasks. Particularly, they performed visual model-predictive control (MPC) to measure the predicted cost of the sub-goal segments and sought for the minimal-cost one as the current goal to reach. When the sub-goals were planned, they simply used visual MPC to plan to reach sub-goals one by one.

**Learn from experts.** The expert demonstration can provide useful guidance for reaching a hard goal, since it directly contains the way-points for decomposing such a target into smaller sub-goals. Such an intuition motivates the work of [Paul et al. \[2019\]](#), which made use of the order of sub-goal indices along the expert trajectory as the evidence of difficulty, and learned to predict the sub-goals in the equipartitions from

the expert trajectory conditioned on the given state.

### 4.3 Relabeling

Similar to sub-goal selection, the relabeling step also replaces the initial desired goal provided by the task to enhance the learning efficiency. However, relabeling focuses on replacing the history data in the experience buffer before training the agent, instead of changing the distribution for sampling experience. Formally, it requires a relabel function  $h$  to replace the goal of the samples tuple  $(s_t, a_t, g_t, r_t)$  from the replay buffer  $\mathcal{B}$  and re-compute the reward:

$$(s_t, a_t, g_t, r_t) \leftarrow (s_t, a_t, h(\cdot), r(s_t, a_t, h(\cdot))), \quad (12)$$

where  $h$  can be designed as taking any particular features into getting new goals.

**Hindsight relabeling.** The relabeling strategy can be traced back to the famous Hindsight Experience Replay (HER) [Andrychowicz *et al.*, 2017], which is motivated by the fact that human learns from failure experience. HER relabels the desired goals in the buffer with achieved goals in the same trajectories and alleviates the sparse reward signal problem. It shows that off-policy algorithms can reuse the data in the replay buffer by relabeling goals in the previous trajectory. HER is widely used as the backbone in many related works [Ren *et al.*, 2019; Pitis *et al.*, 2020; Durugkar *et al.*, 2021], and researchers also take efforts on extending the idea of HER on designing advanced relabeling strategies. As an improvement, Fang *et al.* [2019] proposed CHER, a curriculum relabeling method which adaptively selects the relabeled goals from failed experiences. The curriculum is evaluated according to the likeness to desired goals and the diversity of goals. To increase the variety of relabeled goals and not restricted in the same trajectory, Pitis *et al.* [2020] generalized HER by additionally relabeling goals that are randomly sampled from buffers of desired goals, achieved goals and behavioral goals, which allows the agent to learn different skills at a time. To improve the sample efficiency by exploring on rare visited experiences, Kuang *et al.* [2020] learned a density model for achieved goals and relabeled them by prioritizing the rarely seen ones using the density model.

Recently, Eysenbach *et al.* [2020] revealed the relationship between HER and inverse RL which studies how to recover the reward function on a set of offline expert trajectories. Specifically, hindsight relabeling can be seen as a special case of maximum-entropy inverse RL with a pre-defined reward function, and thus inverse RL method is used for relabeling goals in their work.

**Relabeling by learning.** Selecting the relabeling goals from past experiences limit the diversity of relabeled goals. To alleviate such a problem and allows relabeling unseen goals, Bai *et al.* [2019] trained a conditional generative RNN by explicitly modeling the relationship between the policy performance and the achieved goals, which is further used for generating relabeled goals according to the averaged return obtained by the current policy.

**Foresight.** Motivated by the fact that humans can learn from failure but are also capable of planning for the future based on the current situations, Zhu *et al.* [2021] proposed to

learn and utilize a dynamics model to generate virtual trajectories for relabeling. Foresight goal relabeling can also prevent labeling homogeneous goals limited in historical data, which plans new goals that the current policy can achieve.

## 5 Summary and Future Prospects

This paper provides a brief review of the development of goal-conditioned reinforcement learning (GCRL). We show the formulation in the beginning, explain the problem and summarize the basic challenges of in GCRL literature. Further, we investigate the critical element – goal in previous works and present its main types. In the later part, we discuss key ways for solving GCRL from the perspective of optimization, sub-goal generation and relabeling, respectively.

Although we try to cover most of the topic, GCRL formulation and its variant are not limited in the discussed material. Interested readers can also refer to another survey paper [Colas *et al.*, 2020b] that concluded GCRL from a different point of view. As is known to all, RL is hard for large-scale applications, due to its thorniest limitation of learning from a single reward function and the bottleneck of the online interaction cost. Therefore, we note four potential prospects for GCRL, namely learning totally intrinsic skills, learning from offline datasets, and large decision models.

**Learning totally intrinsic skills.** The formulation of GCRL provides the considerable potential to incorporate self-supervised training to improve the model ability instead of learning from a simple reward signal. Recently, there are more research works that break the circle of reaching a meaningful goal state (we have discussed some examples like reward and commands in Section 3.4) and ask the agents to explore all possible goals through itself, as a trend of developing totally intrinsic skill learning [Eysenbach *et al.*, 2019a; Sharma *et al.*, 2020]. Specifically, a goal-conditioned policy can be only with intrinsic rewards targeted to maximizing the mutual information between the skill and the behaviors.

**Learning from offline datasets.** Since the online data collection in many realistic applications is always infeasible, many researches pay attention to offline RL methods, which shows potential to learning sequential decision policies solely from static datasets without interacting with the environment. This inspires taking advantage of GCRL for learning generalized goal-conditioned decision models from the offline dataset, like in Tian *et al.* [2021] and Chebotar *et al.* [2021].

**Potential in large decision models.** An appealing direction for utilizing GCRL is to develop general pre-trained goal-conditioned control policies. Combined with other pre-trained models, such a large decision model can be used to command the robots to handle a variety of tasks with kinds of goal elements, such as a sentence of text description [Ahn *et al.*, 2022], a goal image of the order of object placement [Chebotar *et al.*, 2021], a goal image of the targeted area in the open world [Shah *et al.*, 2021], etc.

**Other prospects.** Apart from directions mentioned above, we note that there are more intriguing future directions worth explored in the formulation of GCRL, such as time-extended goals [Forestier and Oudeyer, 2016; Brafman *et al.*, 2018],

Approach	Resolved Goal Type	Sub-Goal	Relabel	Optimization
UVFA [Schaul <i>et al.</i> , 2015]	Vector/Image			✓
HER [Andrychowicz <i>et al.</i> , 2017]	Vector/Image		✓	
Laplacian Representation Learning [Wu <i>et al.</i> , 2018]	Vector/Image			✓
SR [Trott <i>et al.</i> , 2019]	Vector/Image			✓
DDL [Hartikainen <i>et al.</i> , 2019]	Vector/Image	✓		✓
SoRB [Eysenbach <i>et al.</i> , 2019b]	Vector/Image	✓		
Skew-Fit [Pong <i>et al.</i> , 2020]	Vector/Image	✓		
HER with Demonstrations [Nair <i>et al.</i> , 2018a]	Vector			✓
GoalGAN [Florensa <i>et al.</i> , 2018]	Vector	✓		
Sub-goal Discovery [Paul <i>et al.</i> , 2019]	Vector	✓		
HGG [Ren <i>et al.</i> , 2019]	Vector	✓		
CHER [Fang <i>et al.</i> , 2019]	Vector		✓	
G-HER [Bai <i>et al.</i> , 2019]	Vector		✓	
Hindsight Planner [Lai <i>et al.</i> , 2020]	Vector	✓		
GDP [Kuang <i>et al.</i> , 2020]	Vector		✓	
VDS [Zhang <i>et al.</i> , 2020]	Vector	✓		
MEGA [Pitis <i>et al.</i> , 2020]	Vector	✓	✓	
PlanGAN [Charlesworth and Montana, 2020]	Vector			✓
AIM [Durugkar <i>et al.</i> , 2021]	Vector			✓
I-HER [McCarthy and Redmond, 2021]	Vector			✓
GCSL [Ghosh <i>et al.</i> , 2021]	Vector			✓
MapGo [Zhu <i>et al.</i> , 2021]	Vector		✓	✓
L3P [Zhang <i>et al.</i> , 2021]	Vector			✓
RIG [Nair <i>et al.</i> , 2018b]	Image	✓		
DISCERN [Warde-Farley <i>et al.</i> , 2019]	Image	✓		
HVF [Nair and Finn, 2020]	Image	✓		
AMIGo [Campero <i>et al.</i> , 2021]	Image	✓		
GAP [Nair <i>et al.</i> , 2020b]	Image			✓
LEXA [Mendonca <i>et al.</i> , 2021]	Image			✓

Table 2: Summary of the algorithms mentioned in Section 4.

constraint-based goals [Colas *et al.*, 2021], composition of goals [Chitnis *et al.*, 2021] etc.

## Acknowledgements

The authors are supported by “New Generation of AI 2030” Major Project (2018AAA0100900), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), NSFC (62076161). Minghuan Liu is also supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, SJTU. We thank Mingcheng Chen, Xihuai Wang and Zhengyu Yang for helpful discussions.

## References

- [Ahn *et al.*, 2022] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- [Akakzia *et al.*, 2021] Ahmed Akakzia, Cédric Colas, Pierre-Yves Oudeyer, Mohamed Chetouani, and Olivier Sigaud. Grounding language to autonomously-acquired skills via goal generation. *ICLR*, 2021.
- [Andrychowicz *et al.*, 2017] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *NeurIPS*, 2017.
- [Bai *et al.*, 2019] Chenjia Bai, Peng Liu, Wei Zhao, and Xianglong Tang. Guided goal generation for hindsight multi-goal reinforcement learning. *Neurocomputing*, 2019.
- [Beattie *et al.*, 2016] Charles Beattie, Joel Z Leibo, et al. Deepmind lab. *CoRR abs/1612.03801*, 2016.
- [Brafman *et al.*, 2018] Ronen Brafman, Giuseppe De Giacomo, and Fabio Patrizi. Ltlf/ldlf non-markovian rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, et al. Openai gym. *CoRR abs/1606.01540*, 2016.
- [Campero *et al.*, 2021] Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. Learning with amigo: Adversarially motivated intrinsic goals. *ICLR*, 2021.
- [Chan *et al.*, 2019] Harris Chan, Yuhuai Wu, Jamie Kiros, Sanja Fidler, and Jimmy Ba. Actrce: Augmenting expe-



- rience via teacher’s advice for multi-goal reinforcement learning. *CoRR abs/1902.04546*, 2019.
- [Chane-Sane *et al.*, 2021] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *ICML*, 2021.
- [Charlesworth and Montana, 2020] Henry Charlesworth and Giovanni Montana. Plangan: Model-based planning with sparse rewards and multiple goals. *NeurIPS*, 2020.
- [Chebotar *et al.*, 2021] Yevgen Chebotar, Karol Hausman, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *ICML*, 2021.
- [Chen *et al.*, 2021] 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. *CoRR abs/2106.01345*, 2021.
- [Chitnis *et al.*, 2021] Rohan Chitnis, Tom Silver, Joshua Tenenbaum, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Glib: Efficient exploration for relational model-based reinforcement learning via goal-literal babbling. *AAAI*, 2021.
- [Colas *et al.*, 2020a] Cédric Colas, Tristan Karch, Nicolas Lair, Jean-Michel Dussoux, Clément Moulin-Frier, Peter Ford Dominey, and Pierre-Yves Oudeyer. Language as a cognitive tool to imagine goals in curiosity-driven exploration. *NeurIPS*, 2020.
- [Colas *et al.*, 2020b] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *arXiv preprint arXiv:2012.09830*, 2020.
- [Colas *et al.*, 2021] Cédric Colas, Boris Hejblum, Sébastien Rouillon, et al. Epidemiotrim: A toolbox for the optimization of control policies in epidemiological models. *JAIR*, 2021.
- [Durugkar *et al.*, 2021] Ishan Durugkar, Mauricio Tec, Scott Niekum, and Peter Stone. Adversarial intrinsic motivation for reinforcement learning. *CoRR abs/2105.13345*, 2021.
- [Eysenbach *et al.*, 2019a] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *ICLR*, 2019.
- [Eysenbach *et al.*, 2019b] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. *NeurIPS*, 2019.
- [Eysenbach *et al.*, 2020] Benjamin Eysenbach, Xinyang Geng, Sergey Levine, and Ruslan Salakhutdinov. Rewriting history with inverse rl: Hindsight inference for policy improvement. *NeurIPS*, 2020.
- [Fang *et al.*, 2019] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *NeurIPS*, 2019.
- [Florensa *et al.*, 2018] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. *ICML*, 2018.
- [Forestier and Oudeyer, 2016] Sébastien Forestier and Pierre-Yves Oudeyer. Modular active curiosity-driven discovery of tool use. *IROS*, 2016.
- [Ghosh *et al.*, 2021] Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *ICLR*, 2021.
- [Hartikainen *et al.*, 2019] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. *ICLR*, 2019.
- [Jiang *et al.*, 2019] Yiding Jiang, Shixiang Gu, Kevin Murphy, and Chelsea Finn. Language as an abstraction for hierarchical deep reinforcement learning. *NeurIPS*, 2019.
- [Johnson *et al.*, 2016] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. 2016.
- [Khazatsky *et al.*, 2021] Alexander Khazatsky, Ashvin Nair, Daniel Jing, and Sergey Levine. What can i do here? learning new skills by imagining visual affordances. *ICRA*, 2021.
- [Kuang *et al.*, 2020] Yingyi Kuang, Abraham Itzhak Weinberg, George Vogiatzis, and Diego R Faria. Goal density-based hindsight experience prioritization for multi-goal robot manipulation reinforcement learning. *RO-MAN*, 2020.
- [Kumar *et al.*, 2019] Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies. *CoRR abs/1912.13465*, 2019.
- [Lai *et al.*, 2020] Yaqing Lai, Wufan Wang, Yunjie Yang, Jihong Zhu, and Minchi Kuang. Hindsight planner. *AAMAS*, 2020.
- [Lee *et al.*, 2020] Lisa Lee, Benjamin Eysenbach, Ruslan Salakhutdinov, Shixiang Shane Gu, and Chelsea Finn. Weakly-supervised reinforcement learning for controllable behavior. *NeurIPS*, 2020.
- [Luketina *et al.*, 2019] Jelena Luketina, Nantas Nardelli, et al. A survey of reinforcement learning informed by natural language. *IJCAI*, 2019.
- [McCarthy and Redmond, 2021] Robert McCarthy and Stephen J Redmond. Imaginary hindsight experience replay: Curious model-based learning for sparse reward tasks. *CoRR abs/2110.02414*, 2021.
- [Mendonca *et al.*, 2021] Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *CoRR abs/2110.09514*, 2021.
- [Nair and Finn, 2020] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *ICLR*, 2020.
- [Nair *et al.*, 2018a] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel.

- Overcoming exploration in reinforcement learning with demonstrations. *ICRA*, 2018.
- [Nair *et al.*, 2018b] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *NeurIPS*, 2018.
- [Nair *et al.*, 2020a] Ashvin Nair, Shikhar Bahl, Alexander Khazatsky, Vitchyr Pong, Glen Berseth, and Sergey Levine. Contextual imagined goals for self-supervised robotic learning. *CoRL*, 2020.
- [Nair *et al.*, 2020b] Suraj Nair, Silvio Savarese, and Chelsea Finn. Goal-aware prediction: Learning to model what matters. *ICML*, 2020.
- [Niemueller *et al.*, 2019] Tim Niemueller, Till Hofmann, and Gerhard Lakemeyer. Goal reasoning in the clips executive for integrated planning and execution. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 754–763, 2019.
- [Paul *et al.*, 2019] Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. *NeurIPS*, 2019.
- [Pitis *et al.*, 2020] Silviu Pitis, Harris Chan, Stephen Zhao, Bradley Stadie, and Jimmy Ba. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. *ICML*, 2020.
- [Plappert *et al.*, 2018] Matthias Plappert, Marcin Andrychowicz, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *CoRR abs/1802.09464*, 2018.
- [Pong *et al.*, 2020] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *ICML*, 2020.
- [Ren *et al.*, 2019] Zhizhou Ren, Kefan Dong, Yuan Zhou, Qiang Liu, and Jian Peng. Exploration via hindsight goal generation. *NeurIPS*, 2019.
- [Roberts *et al.*, 2018] Mark Roberts, Daniel Borrajo, Michael Cox, and Neil Yorke-Smith. Special issue on goal reasoning. *AI Communications*, 31(2):115–116, 2018.
- [Rolf and Steil, 2013] Matthias Rolf and Jochen J Steil. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE transactions on neural networks and learning systems*, 25(6):1147–1160, 2013.
- [Santucci *et al.*, 2016] Vieri Giuliano Santucci, Gianluca Baldassarre, and Marco Mirolli. Grail: a goal-discovering robotic architecture for intrinsically-motivated learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8(3):214–231, 2016.
- [Schaul *et al.*, 2015] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. *ICML*, 2015.
- [Schmidhuber, 2019] Juergen Schmidhuber. Reinforcement learning upside down: Don’t predict rewards—just map them to actions. *CoRR abs/1912.02875*, 2019.
- [Shah *et al.*, 2021] Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.
- [Sharma *et al.*, 2020] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *ICLR*, 2020.
- [Srinivas *et al.*, 2018] Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. *ICML*, 2018.
- [Srivastava *et al.*, 2019] Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz, Wojciech Jaśkowski, and Jürgen Schmidhuber. Training agents using upside-down reinforcement learning. *CoRR abs/1912.02877*, 2019.
- [Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [Tang and Kucukelbir, 2021] Yunhao Tang and Alp Kucukelbir. Hindsight expectation maximization for goal-conditioned reinforcement learning. In *AISTATS*, pages 2863–2871, 2021.
- [Tassa *et al.*, 2020] Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm\_control: Software and tasks for continuous control, 2020.
- [Tian *et al.*, 2021] Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Model-based visual planning with self-supervised functional distances. *ICLR*, 2021.
- [Trott *et al.*, 2019] Alexander Trott, Stephan Zheng, Caiming Xiong, and Richard Socher. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *NeurIPS*, 2019.
- [Warde-Farley *et al.*, 2019] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *ICLR*, 2019.
- [Wu *et al.*, 2018] Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *ICLR*, 2018.
- [Zhang *et al.*, 2020] Yunzhi Zhang, Pieter Abbeel, and Lerrel Pinto. Automatic curriculum learning through value disagreement. *NeurIPS*, 2020.
- [Zhang *et al.*, 2021] Lunjun Zhang, Ge Yang, and Bradley C Stadie. World model as a graph: Learning latent landmarks for planning. *ICML*, 2021.
- [Zhu *et al.*, 2021] Menghui Zhu, Minghuan Liu, Jian Shen, Zhicheng Zhang, Sheng Chen, Weinan Zhang, Deheng Ye, Yong Yu, Qiang Fu, and Wei Yang. Mapgo: Model-assisted policy optimization for goal-oriented tasks. *IJCAI*, 2021.