Learn Goal-Conditioned Policy with Intrinsic Motivation for Deep Reinforcement Learning

Jinxin Liu[1,2,3], **Donglin Wang**[2,3*], **Qiangxing Tian**[1,2,3], **Zhengyu Chen**[1,2,3]

[1]Zhejiang University. [2]Westlake University. [3]Westlake Institute for Advanced Study.
{liujinxin, wangdonglin, tianqiangxing, chenzhengyu}@westlake.edu.cn

Abstract

It is of significance for an agent to autonomously explore the environment and learn a widely applicable and general-purpose goal-conditioned policy that can achieve diverse goals including images and text descriptions. Considering such perceptually-specific goals, one natural approach is to reward the agent with a prior non-parametric distance over the embedding spaces of states and goals. However, this may be infeasible in some situations, either because it is unclear how to choose suitable measurement, or because embedding (heterogeneous) goals and states is non-trivial. The key insight of this work is that we introduce a latent-conditioned policy to provide goals and intrinsic rewards for learning the goal-conditioned policy. As opposed to directly scoring current states with regards to goals, we obtain rewards by scoring current states with associated latent variables. We theoretically characterize the connection between our unsupervised objective and the multi-goal setting, and empirically demonstrate the effectiveness of our proposed method which substantially outperforms prior techniques in a variety of tasks.

Introduction

Deep reinforcement learning (RL) makes it possible to drive agents to achieve sophisticated goals in complex and uncertain environments, from computer games (Badia et al. 2020; Berner et al. 2019) to real robot control (Lee et al. 2018; Lowrey et al. 2019; Vecerik et al. 2019; Popov et al. 2017), which usually involves learning a specific policy for individual task relying on hand-specifying reward function. However, autonomous agents are expected to exist persistently in the world and have the ability to reach diverse goals. To achieve this, one needs to design a mechanism to spontaneously generate diverse goals and the associated rewards, over which the goal-conditioned policy is trained.

Based on the space of goal manifold, previous works can be divided into two categories: *perceptually-specific goal based approaches* and *latent variable based methods*. In the former, previous approaches normally assume the spaces of perceptual goals and states are same, and sample goals from the historical trajectories of the policy to be trained. It is convenient to use a prior non-parametric measure function, such as L2 norm, to provide rewards (current states vs. goals) over the state space or the embedding space (Higgins et al. 2017; Nair et al. 2018; Sermanet et al. 2018; Warde-Farley et al. 2019). However, these approaches taking the prior non-parametric measure function may limit the repertoires of behaviors and impose manual engineering burdens.

On the contrary, *latent variable based methods* assume that goals (latent variables) and states come from different spaces and the distribution of goals (latent variables) is known a priori. In parallel, such methods autonomously learn a reward function and a latent-conditioned policy through the lens of empowerment (Salge, Glackin, and Polani 2014; Eysenbach et al. 2018; Sharma et al. 2020). However, such policy is conditioned on latent variables rather than perceptually-specific goals. Applying this procedure to goal-reaching tasks, similar to the parameter initialization or hierarchical RL, needs an external reward function for new tasks; otherwise the learned latent-conditioned policy cannot be applied directly to perceptually-specific goals.

In this paper, we incorporate a latent variable based objective into the perceptual goal-reaching tasks. Specifically, we decouple the task generation (including perceptual goals and associated reward functions) and goal-conditioned policy optimization, which are often intertwined in prior approaches. For the task generation, we employ a latent variable based objective (Eysenbach et al. 2018) to learn a latent-conditioned policy, run to generate goals, and a discriminator, serve as the reward function. Then our goal-conditioned policy is rewarded by the discriminator to imitate the trajectories, relabeled as goals, induced by the latent-conditioned policy. This procedure enables the acquired discriminator as a proxy to reward the goal-conditioned policy for various relabeled goals. *In essence, the latent-conditioned policy can reproducibly influence the environment, and the goal-conditioned policy perceptibly imitates these influences.*

The main contribution of our work is an unsupervised RL method that can learn a perceptual goal-conditioned policy via intrinsic motivation (GPIM). Our training procedure decouples the task (goals and rewards) generation and policy optimization, which makes the obtained reward function universal and effective for various relabeled goals, including images and texts. We formally analyze the effectiveness of our relabeling procedure, and empirically find that our intrinsic reward is well shaped by environment's dynamics and as a result benefits the training efficiency on extensive tasks.

[*]Corresponding author.

Preliminaries

The goal in a reinforcement learning problem is to maximize the expected return in a Markov decision process (MDP) $\mathcal{M}$, defined by the tuple (S, A, p, r, γ), where S and A are state and action spaces, $p(s_{t+1}|s_t, a_t)$ gives the next-state distribution upon taking action a_t in state s_t, $r(s_t, a_t, s_{t+1})$ is the reward received at transition $s_t \xrightarrow{a_t} s_{t+1}$, and γ is a discount factor. The objective is to learn the policy $\pi_\theta(a_t|s_t)$ by maximizing $\mathbb{E}_{p(\tau;\theta)}[R(\tau)] = \mathbb{E}_{p(\tau;\theta)}[\sum_t \gamma^t r(s_t, a_t, s_{t+1})]$, where $p(\tau;\theta)$ denotes the induced trajectories by policy π_θ in the environment: $p(\tau;\theta) = p(s_0) \cdot \prod_{t=0}^{t=T-1} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t)$.

Multi-goal RL augments the above optimization with a goal g by learning a goal-conditioned policy $\pi_\theta(a_t|s_t, g)$ and optimizing $\mathbb{E}_{p(g)}\mathbb{E}_{p(\tau|g;\theta)}[R(\tau)]$ with reward $r(s_t, a_t, s_{t+1}, g)$. Such optimization can also be interpreted as a form of mutual information between the goal g and agent's trajectory τ (Warde-Farley et al. 2019):

$$\max \mathcal{I}(\tau; g) = \mathbb{E}_{p(g)p(\tau|g;\theta)}[\log p(g|\tau) - \log p(g)]. \quad (1)$$

If $\log p(g|\tau)$ is unknown and the goal g is a latent variable, latent variable based models normally maximize the mutual information between the latent variable ω and agent's behavior b, and lower-bound this mutual information by approximating the posterior $p(\omega|b)$ with a learned $q_\phi(\omega|b)$: $\mathcal{I}(b;\omega) \geq \mathbb{E}_{p(\omega,b;\mu)}[\log q_\phi(\omega|b) - \log p(\omega)]$, where the specific manifestation of agent's behavior b can be an entire trajectory τ, an individual state s or a final state s_T. It is thus applicable to train $\pi_\mu(a_t|s_t, \omega)$ with learned q_ϕ (as reward).

Several prior works have sought to incorporate the latent-conditioned $\pi_\mu(a_t|s_t, \omega)$ (as low-level skills) into hierarchical RL (Zhang, Yu, and Xu 2021) or reuse the learned q_ϕ (as predefined tasks) in meta-RL (Gupta et al. 2018), while we claim to reuse *both* π_μ and q_ϕ with our relabeling procedure.

The Method

In this section, we first formalize the problem and introduce the framework. Second, we illustrate our GPIM objective and elaborate on the process of how to jointly learn the latent-conditioned policy and a goal-conditioned policy. Third, we formally verify our (unsupervised) objective and understand how GPIM relates to the standard multi-goal RL.

Overview

As shown in Figure 1 (right), our objective is to learn a goal-conditioned policy $\pi_\theta(a|\tilde{s}, g)$ that inputs state $\tilde{s}$ and perceptually-specific goal g and outputs action a. To efficiently generate tasks for training the goal-conditioned policy π_θ, we introduce another latent-conditioned policy $\pi_\mu(a|s, \omega)$, which takes as input a state s and a latent variable ω and outputs action a to generate goals, and the associated discriminator q_ϕ (i.e., generating tasks). Additionally, we assume that we have access to a procedural relabeling function f_κ (we will discuss this assumption latter), which can relabel states s as goals g for training π_θ. On this basis, $\pi_\theta(a|\tilde{s}, g)$ conditioned on the relabeled goal g interacts with the reset environment under the instruction of the associated q_ϕ. We use the non-tilde s and the tilde $\tilde{s}$ to distinguish between the states of two policies respectively. Actually, $\tilde{s}$ and s come from the same state space. In the following, if not specified, goal g refers to the perceptually-specific goal, and no longer includes the case that goal is a latent variable.

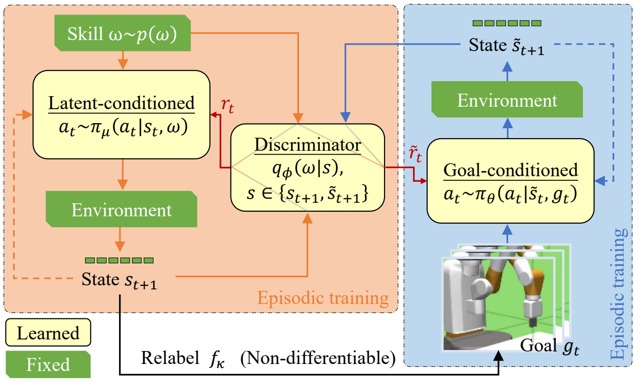


Figure 1: Framework of GPIM. We jointly train the latent-conditioned policy π_μ and the discriminator q_ϕ to understand skills which specify task objectives (e.g., trajectories, the final goal state), and use such understanding to reward the goal-conditioned policy π_θ for completing such tasks (relabeled states). In this diagram, state s_{t+1} (e.g., joints) induced by π_μ is converted into the perceptually-specific goal g_t (e.g., images or text descriptions) for π_θ. Note that the two environments above are same, and the initial states s_0 of π_μ and $\tilde{s}_0$ of π_θ are sampled from the same (fixed) distribution.

To ensure the generated tasks (by π_μ) are reachable for π_θ, we explicitly make the following assumption[1]:

Assumption 1 *The initial state of the environment is fixed.*

Proposed GPIM Method

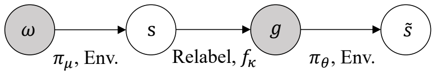


Figure 2: Latent-conditioned policy π_μ provides goals and the associated reward for the goal-conditioned policy π_θ.

In order to jointly learn the latent-conditioned $\pi_\mu(a|s, \omega)$ and goal-conditioned $\pi_\theta(a|\tilde{s}, g)$, we maximize the mutual information between the state s and latent variable w for π_μ, and simultaneously maximize the mutual information between the state $\tilde{s}$ and goal g for π_θ. Consequently, the overall objective to be maximized can be expressed as follows [2]

$$\mathcal{F}(\mu, \theta) = \mathcal{I}(s; \omega) + \mathcal{I}(\tilde{s}; g). \quad (2)$$

For clarification, Figure 2 depicts the graphical model for the latent variable ω, state s induced by π_μ, goal g relabeled

[1] In appendix, we empirically find the assumption can be lifted.

[2] To further clarify the motivation, we conduct the ablation study to compare our method (maximizing $\mathcal{I}(s;\omega) + \mathcal{I}(\tilde{s}; g)$) with that just maximizing $\mathcal{I}(\tilde{s};\omega)$ and that maximizing $\mathcal{I}(\tilde{s}; g)$ in appendix.

from s, and state $\tilde{s}$ induced by π_θ. As seen, the latent variable $\omega \sim p(\omega)$ is firstly used to generate state s via the policy π_μ interacting with the environment. Then, we relabel the generated state s to goal g. After that, π_θ conditioned on g interacts with the environment to obtain the state $\tilde{s}$ at another episode. In particular, π_μ is expected to generate diverse behavior modes by maximizing $\mathcal{I}(s;\omega)$, while π_θ behaving like $\tilde{s}$ is to "imitate" (see next q_ϕ) these behaviors by taking as input the relabeled goal (indicated by Figure 2).

Based on the context, the correlation between $\tilde{s}$ and g is no less than that between $\tilde{s}$ and w: $\mathcal{I}(\tilde{s};g) \geq \mathcal{I}(\tilde{s};\omega)$ (Beaudry and Renner 2011). Thus, we can obtain the lower bound:

$$\begin{aligned}\mathcal{F}(\mu,\theta) &\geq \mathcal{I}(s;\omega) + \mathcal{I}(\tilde{s};\omega) \qquad (3)\\ &= 2\mathcal{H}(\omega) + \mathbb{E}_{p_m(\cdot)}\left[\log p(\omega|s) + \log p(\omega|\tilde{s})\right],\end{aligned}$$

where $p_m(\cdot) \triangleq p(\omega, s, g, \tilde{s}; \mu, \kappa, \theta)$ denotes the joint distribution of ω, s, g and $\tilde{s}$ specified by the graphic model in Figure 2. Since it is difficult to exactly compute the posterior distributions $p(\omega|s)$ and $p(\omega|\tilde{s})$, Jensen's Inequality (Barber and Agakov 2003) is further applied for approximation by using a learned discriminator network $q_\phi(\omega|\cdot)$. Thus, we have $\mathcal{F}(\mu,\theta) \geq \mathcal{J}(\mu,\phi,\theta)$, where

$$\mathcal{J}(\mu,\phi,\theta) \triangleq 2\mathcal{H}(\omega) + \mathbb{E}_{p_m(\cdot)}\left[\log q_\phi(\omega|s) + \log q_\phi(\omega|\tilde{s})\right].$$

It is worth noting that the identical discriminator q_ϕ is used for the variational approximation of $p(\omega|s)$ and $p(\omega|\tilde{s})$. For the state s induced by skill $\pi_\mu(\cdot|\cdot,\omega)$ and $\tilde{s}$ originating from $\pi_\theta(\cdot|\cdot,g)$, the shared discriminator q_ϕ assigns a similarly high probability on ω for both states s and $\tilde{s}$ associated with the same ω. Therefore, q_ϕ can be regarded as a reward network shared by the latent-conditioned π_μ and goal-conditioned π_θ. Intuitively, we factorize acquiring the goal-conditioned policy π_θ and learn it purely in the space of the agent's embodiment (i.e., the latent ω) — separate from the perceptually-specific goal g (e.g., states, images and texts), where the latent ω and the perceptual goal g have different characteristics due to the underlying manifold spaces.

According to the surrogate objective $\mathcal{J}(\mu,\phi,\theta)$, we propose an alternating optimization between π_μ, q_ϕ and π_θ:

Step I: Fix π_θ and update π_μ and q_ϕ. In this case, θ is not a variable to update and thus $\mathcal{J}(\mu,\phi,\theta)$ becomes

$$\begin{aligned}\mathcal{J}(\mu,\phi) = \;&\mathbb{E}_{p(\omega,s;\mu)}\left[\log q_\phi(\omega|s)\right]\\ &+ \underbrace{\mathbb{E}_{p_m(\cdot)}\left[\log q_\phi(\omega|\tilde{s}) - 2\log p(\omega)\right]}_{\text{Variable independent term}}. \qquad (4)\end{aligned}$$

According to Equation 4, π_μ can be thus optimized by setting the intrinsic reward at time step t as

$$r_t = \log q_\phi(\omega|s_{t+1}) - \log p(\omega), \qquad (5)$$

where the term $-\log p(\omega)$ is added for agents to avoid artificial termination and reward-hacking issues (Amodei et al. 2016; Eysenbach et al. 2018). We implement this optimization with SAC. In parallel, the reward network (discriminator) q_ϕ can be updated with SGD by maximizing

$$\mathbb{E}_{p(\omega)p(s|\omega;\mu)}\left[\log q_\phi(\omega|s)\right]. \qquad (6)$$

Algorithm 1: Learning process of our proposed GPIM

```
 1: while not converged do
 2:     # Step I: generate goals and reward functions.
 3:     Sample the latent variable: $\omega \sim p(\omega)$.
 4:     Reset Env. & sample initial state: $s_0 \sim p_0(s)$.
 5:     for $t = 0, 1, ..., T-1$ steps do
 6:         Sample action: $a_t \sim \pi_\mu(a_t|s_t,\omega)$.
 7:         Step environment: $s_{t+1} \sim p(s_{t+1}|s_t,a_t)$.
 8:         Relabel: $g_t = f_\kappa(s_{t+1})$.    ▷ Record.
 9:         Compute reward $r_t$ for policy $\pi_\mu$ using (5).
10:         Update policy $\pi_\mu$ to maximize $r_t$ with SAC.
11:         Update discriminator ($q_\phi$) to maximize (6) with SGD.
12:     end for
13:     # Step II: $\pi_\theta$ imitates $\pi_\mu$ with the relabeled goals and the
        associated rewards (for the same $\omega$).
14:     Reset Env. & sample initial state: $\tilde{s}_0 \sim p_0(\tilde{s})$.
15:     for $t = 0, 1, ..., T-1$ steps do
16:         Recap dynamic (time-varying) goal $g_t$ from the recorded
            goals in line 8.  # Note: $g_t = g_T$ for static (fixed) goals.
17:         Sample action: $a_t \sim \pi_\theta(a_t|\tilde{s}_t,g_t)$.
18:         Step environment: $\tilde{s}_{t+1} \sim p(\tilde{s}_{t+1}|\tilde{s}_t,a_t)$.
19:         Compute reward $\tilde{r}_t$ for policy $\pi_\theta$ using (8).
20:         Update policy $\pi_\theta$ to maximize $\tilde{r}_t$ with SAC.
21:     end for
22: end while
```

Step II: Fix π_μ and q_ϕ to update π_θ. In this case, μ and ϕ are not variables to update, and $\mathcal{J}(\mu,\phi,\theta)$ can be simplified as

$$\begin{aligned}\mathcal{J}(\theta) = \;&\mathbb{E}_{p_m(\cdot)}\left[\log q_\phi(\omega|\tilde{s})\right]\\ &+ \underbrace{\mathbb{E}_{p(\omega,s;\mu)}\left[\log q_\phi(\omega|s) - 2\log p(\omega)\right]}_{\text{Variable independent term}}. \qquad (7)\end{aligned}$$

According to Equation 7, π_θ can thus be optimized by setting the intrinsic reward at time step t as

$$\tilde{r}_t = \log q_\phi(\omega|\tilde{s}_{t+1}) - \log p(\omega), \qquad (8)$$

where the term $-\log p(\omega)$ is added for the same reason as above and we also implement this optimization with SAC. Note that we do not update q_ϕ with the data induced by π_θ.

These two steps are performed alternately until convergence (see Algorithm 1). In summary, we train the goal-conditioned π_θ along with an extra latent-conditioned π_μ an a procedural relabel function f_κ, which explicitly decouples the procedure of unsupervised RL into task generation (including goals and reward functions) and policy optimization.

For clarity, we state three different settings for f_κ: (1) if $f_\kappa = q_\phi$, our objective is identical to the latent variable based models, maximizing $\mathcal{I}(\omega;s)$ to obtain the *latent-conditioned* policy (Eysenbach et al. 2018); (2) if $f_\kappa(s) = s$, this procedure is consistent with the hindsight relabeling (Andrychowicz et al. 2017); (3) if $f_\kappa(s)$ and s have different spaces (not latent spaces), this relabeling is also a reasonable belief under the semi-supervised setting, e.g., the social partner in Colas et al. (2020). In our experiment, we will consider (2) and (3) to learn $\pi_\theta(\cdot|\cdot,g)$ that is conditioned *perceptually-specific* goals. For (3), it is easy to procedurally generate the image-based goals from (joint-based) states with the MuJoCo Physics Engine's (Todorov, Erez, and Tassa 2012) built-in renderer. Facing high-dimensional goals, we also incorporate a self-supervised loss over the perception-level (Hafner et al. 2020; Lu et al. 2020) for π_θ.

Theoretical Analysis

Normally, multi-goal RL seeks the goal-conditioned policy $\pi_\theta(a|\tilde{s}, g)$ that maximizes $\mathcal{I}(\tilde{s}; g)$ with prior goal-distribution $p'(g)$ and the associated reward $p'(g|\tilde{s})$, while GPIM learns $\pi_\theta(a|\tilde{s}, g)$ by maximizing $\mathcal{I}(s; \omega) + \mathcal{I}(\tilde{s}; \omega)$ without any prior goals and rewards. Here, we characterize the theoretical connection of returns between the two objectives under deterministic π_μ and Assumption 2.

Assumption 2 *The relabeling function f_κ is bijective and the environment is deterministic.*

Let $\eta(\pi_\theta) \triangleq \mathcal{I}(\tilde{s}; g) = \mathbb{E}_{p'(g,\tilde{s};\theta)} \left[\log p'(g|\tilde{s}) - \log p'(g)\right]$, where the expectation is taken over the rollout $p'(g, \tilde{s}; \theta) = p'(g)p(\tilde{s}|g; \theta)$, and $\hat{\eta}(\pi_\theta) \triangleq \mathbb{E}_{p^*_m(\cdot)} \left[\log p(\omega|\tilde{s}) - \log p(\omega)\right]$, where the joint distribution $p^*_m(\cdot) \triangleq p(\omega, s, g, \tilde{s}; \mu^*, \kappa, \theta) = p(\omega)p(s|\omega; \mu^*)p(g|s; \kappa)p(\tilde{s}|g; \theta)$, $\mu^* = \arg\max_\mu \mathcal{I}(s; \omega)$, and $\mathbb{E}_{p(\omega,s;\mu^*)} \log p(s|\omega; \mu^*) = 0$. According our training procedure (π_μ is not affected by π_θ) in Algorithm 1, it is trival to show that $\hat{\eta}(\pi_\theta)$ is a surrogate for our $\mathcal{I}(s; \omega) + \mathcal{I}(\tilde{s}; \omega)$ in Equation 3. We start by deriving that the standard multi-goal RL objective $\eta(\pi_\theta)$ and our (unsupervised) objective $\hat{\eta}(\pi_\theta)$ are equal under some mild assumptions and then generalize this connection to a general case.

Special case: We first assume the prior goal distribution $p'(g)$ for optimizing $\eta(\pi_\theta)$ matches the goal distribution $\mathbb{E}_\omega \left[p(g|\omega; \mu^*, \kappa)\right]$ induced by π_{μ^*} and f_κ for optimizing $\hat{\eta}(\pi_\theta)$. Then, we obtain:

$$
\begin{aligned}
&\hat{\eta}(\pi_\theta) - \eta(\pi_\theta) \\
&= \mathbb{E}_{p^*_m(\cdot)} \left[\log p(\omega|\tilde{s}) - \log p(\omega) - \log p'(g|\tilde{s}) + \log p'(g)\right] \\
&= \mathbb{E}_{p^*_m(\cdot)} \left[\log p(\tilde{s}|\omega; \mu^*, \kappa, \theta) - \log p(\tilde{s}|g; \theta)\right] = 0. \qquad (9)
\end{aligned}
$$

Equation 9 comes from our relabeling procedure (with deterministic π_{μ^*} and Assumption 2), specifying that $p(\tilde{s}|\omega; \mu^*, \kappa, \theta) = \mathbb{E}_{s,g} \left[p(s|\omega; \mu^*)p(g|s; \kappa)p(\tilde{s}|g; \theta)\right]$, $\mathbb{E}_{p^*_m(\cdot)} \left[\log p(s|\omega; \mu^*)\right] = 0$ and $\mathbb{E}_{p^*_m(\cdot)} \left[\log p(g|s; \kappa)\right] = 0$. Essentially, this special case shows that without inductive bias on the self-generated goal distribution, our learning procedure leads to the desired goal-conditioned policy π_θ.

General case: Suppose that we do not have any prior connection between the goal distributions $p'(g)$ wrt optimizing $\eta(\pi_\theta)$ and the self-generated $\mathbb{E}_\omega \left[p(g|\omega; \mu^*, \kappa)\right]$ wrt optimizing $\hat{\eta}(\pi_\theta)$. The following theorem provides such a performance guarantee (Please see appendix for a full derivation):

Theorem 1 *Let $\eta(\pi_\theta)$ and $\hat{\eta}(\pi_\theta)$ be as defined above, and assume relabeling f_κ is bijective, then,*

$$\hat{\eta}(\pi_\theta) - \eta(\pi_\theta) \leq 2R_{max}\sqrt{\epsilon/2},$$

where $R_{max} = \max_{p'(g)p(\tilde{s}|g;\theta)} \log p'(g|\tilde{s}) - \log p'(g)$ and $\epsilon = \mathbb{E}_{p(\omega)} \left[D_{KL}(p(s|\omega; \mu^*) \| p'(s))\right]$.

This theorem implies that as long as we improve the return wrt $\hat{\eta}(\pi_\theta)$ by more than $2R_{max}\sqrt{\epsilon/2}$, we can guarantee improvement wrt the return $\eta(\pi_\theta)$ of standard multi-goal RL.

Note that the analysis presented above makes an implicit requirement that the goal distribution is valid for training π_θ (i.e., there is the corresponding target in the environment for the agent to pursue). This requirement is well satisfied for the multi-goal RL. However, ambiguity appears when the goal distribution $\mathbb{E}_\omega \left[p(g|\omega; \mu^*, \kappa)\right]$, induced by π_{μ^*} and f_κ, and the existed target $p'(g)$ in the reset environment for training π_θ have different supports. For example, the generated goal is "reaching red square", while such target does not exist in the reset environment for training π_θ (Algorithm 1 *line 14*). Thus, we introduce relabeling over the environment (see appendix) for granting valid training.

Related Work

Investigating the goal distribution: For goal-reaching tasks, many prior methods (Schaul et al. 2015; Andrychowicz et al. 2017; Levy et al. 2017; Pong et al. 2018; Hartikainen et al. 2019) assume an available distribution of goals during the exploration. In the unsupervised RL setting, the agent needs to automatically explore the environment and discover potential goals for learning the goal-conditioned policy. Several works (Colas, Sigaud, and Oudeyer 2018; Péré et al. 2018; Warde-Farley et al. 2019; Pong et al. 2019; Kovač, Laversanne-Finot, and Oudeyer 2020) also adopt heuristics to acquire the goal distribution based on previously visited states, which is orthogonal to our relabeling procedure.

Learning the goal-achievement reward function: Building on prior works in standard RL algorithms (Schaul et al. 2015; Schulman et al. 2017; Haarnoja et al. 2018) that learn policies with prior goals and rewards, unsupervised RL faces another challenge — automatically learning the goal-achievement reward function. Two common approaches to obtain rewards are (1) applying *the pre-defined function* on the learned goal representations, and (2) *directly learning a reward function*. Estimating the reward with *the pre-defined function* typically assumes the goal space is the same as the state space, and learns the embeddings of states and goals with various auxiliary tasks (self-supervised loss in perception-level (Hafner et al. 2020)): Sermanet et al. (2018); Warde-Farley et al. (2019); Liu et al. (2021) employ the contrastive loss to acquire embeddings for high-dimensional inputs, and Nair et al. (2018); Florensa et al. (2019); Nair et al. (2019); Pong et al. (2019) elicit the features with the generative models. Over the learned representations, these approaches apply a prior non-parametric measure function (e.g., the cosine similarity) to provide rewards. This contrasts with our decoupled training procedure, where we acquire rewards by scoring current states with their associated latent variables, instead of the perceptual goals. Such procedure provides more flexibility in training the goal-conditioned policy than using pre-defined measurements, especially for the heterogeneous states and goals.

Another approach, *directly learning a reward function*, aims to pursues skills (the latent-conditioned policy) by maximizing the empowerment (Salge, Glackin, and Polani 2014), which draws a connection between option discovery and information theory. This procedure (Achiam et al. 2018; Eysenbach et al. 2018; Gregor, Rezende, and Wierstra 2017; Campos et al. 2020; Sharma et al. 2020; Tian et al. 2021) typically maximizes the mutual information between a latent variable and the induced behaviors (states or trajectories), which is optimized by introducing a latent-conditioned reward function. We explicitly relabels the states induced by the latent-conditioned policy and reuses the learned discriminator for instructing ”imitation”.

Hindsight, self-play and knowledge distillation: Our method is similar in spirit to goal relabeling methods like hindsight experience replay (HER) (Andrychowicz et al. 2017) which replays each episode with a different goal in addition to the one the agent was trying to achieve. By contrast, our GPIM relabels the task for another policy while keeping behavior invariant. The self-play (Sukhbaatar et al. 2017, 2018) and knowledge distillation (Xu et al. 2020) are also related to our relabeling scheme, aiming to refine the training of one task with another associated task.

Experiments

Extensive experiments are conducted to evaluate our proposed GPIM method, where the following four questions will be considered in the main paper: (1) By using the ”archery” task, we clarify whether q_ϕ can provide an effective reward function on learning the goal-conditioned policy π_θ. Furthermore, more complex tasks including navigation, object manipulation, atari games, and mujoco tasks are introduced to answer: (2) Does our model learn effective behaviors conditioned on a variety of goals (with different procedural relabeling f_κ), including high-dimensional images and text descriptions that are heterogeneous to states? (3) Does the proposed GPIM on learning the goal-conditioned policy outperform baselines? (4) Does the learned reward function produce better expressiveness of tasks, compared to the prior non-parametric function in the embedding space? For more experimental questions, analysis and results, see appendix[3] and https://sites.google.com/view/gpim (videos).

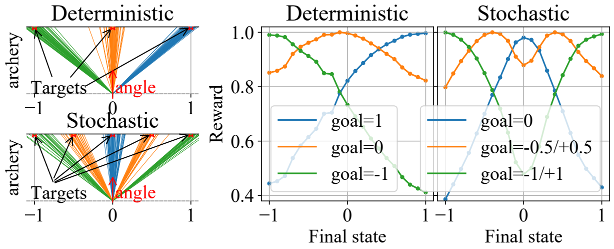


Figure 3: ”Archery” tasks (left) and the learned rewards (right) on both deterministic and stochastic environments.

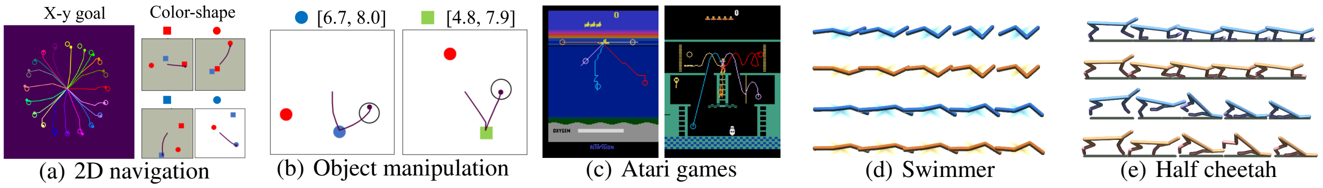
Visualizing the learned reward function. We start with simple ”archery” task to visualize how the learned reward function (discriminator q_ϕ) accounts for goal-conditioned behaviors in environment. The task shown in Figure 3 requires choosing an angle at which we shoot an arrow to the target. The left upper subfigure shows that in a deterministic environment, given three different but fixed targets (with different colors), the arrow reaches the corresponding target successfully under the learned reward function q_ϕ. The reward as a function of the final location of arrows in three tasks is shown on the right. We can find that the learned reward functions resemble convex in terms of the distance between final states to targets. Specifically, the maximum value of the learned reward function is achieved when the final state is close to the given target. The farther away the agent’s final state is from the target, the smaller this reward value is. Similarly, the same conclusion can be drawn from the stochastic environment in the left lower subfigure, where the angle of the arrow has a 50% chance to become a mirror symmetric angle. We see that the learned reward function substantially describes the environment’s dynamics and the corresponding tasks, both in deterministic and stochastic environments. This answers our *first* question.

Scaling to more complex tasks. To answer our *second* question, we now consider more complex tasks as shown in Figure 4. (1) In *2D navigation tasks*, an agent can move in each of the four cardinal directions. We consider the following two tasks: moving the agent to a specific coordinate named *x-y goal* (see appendix for details) and moving the agent to a specific object with certain color and shape named *color-shape goal*. (2) *Object manipulation* considers a moving agent in 2D environment with one block for manipulation, and the other block as a distractor. The agent first needs to reach the block and then move the block to the target location, where the block is described using color and shape. In other words, the description of the goal contains the *color-shape goal* of the true block and the *x-y goal* of the target coordinate. (3) Three *atari games* including seaquest, berzerk and montezuma revenge require an agent to reach the given final states. (4) We use three *mujoco tasks* (swimmer, half cheetah, and fetch) taken from OpenAI GYM (Brockman et al. 2016) to fast imitate given expert trajectories. Specifically, the *static* goals for π_θ in 2D navigation, object manipulation and atari games are the relabeled final state s_T induced by the latent-conditioned policy π_μ: $g_t = f_\kappa(s_T)$, and the *dynamic* goals for π_θ in mujoco tasks are the relabeled states induced by π_μ at each time step: $g_t = f_\kappa(s_{t+1})$ for $0 \leq t \leq T-1$.

The left subfigure of Figure 4(a) shows the learned behavior of navigation in continuous action space given the x-y goal which is denoted as the small circle, and the right subfigure shows the trajectory of behavior with the given color-shape goal. As observed, the agent manages to learn navigation tasks by using GPIM. Further, 2D navigation with color-shape goal (Figure 4(a) *right*) and object manipulation tasks (Figure 4(b)) show the effectiveness of our model facing heterogeneous goals and states. Specifically, Figure 4(b) shows the behaviors of the agent on object manipulation, where the agent is asked to first arrive at a block (i.e., blue circle and green square respectively) and then push it to the target location inside a dark circle (i.e., [6.7, 8.0] and [4.8, 7.9] respectively), where the red object exists as a distractor. Figure 4(c) shows the behaviors of agents that reach the final states in a higher dimensional (action, state and goal) space on *seaquest* and *montezuma revenge* respectively. Figure 4(d-e) shows how the agent imitates expert trajectories (dynamic goals) of *swimmer* and *half cheetah*. We refer the reader to appendix for more results (task *berzerk* and *fetch*).

By learning to reach diverse goals generated by the latent-conditioned policy and employing the self-supervised loss

[3] We refer the reader to https://arxiv.org/abs/2104.05043.



(a) 2D navigation (b) Object manipulation (c) Atari games (d) Swimmer (e) Half cheetah

Figure 4: Goals and learned behaviors: Dots in 2D navigation (x-y goal) and atari games denote different final (*static*) goal states, and curves with same color represent corresponding trajectories; Goals in 2D navigation (color-shape goal) and object manipulation are described using the text at the top of the diagram, where the purple lines imply the behaviors; In the swimmer and half cheetah tasks, the first and third rows represent the *dynamic* goals, and each row below represents the learned behaviors.

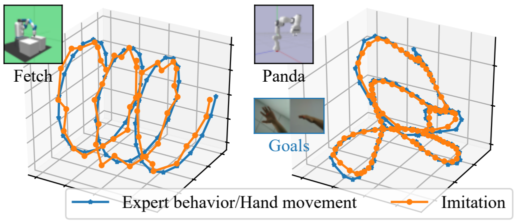


Figure 5: Dynamic goals on Fetch and Panda.

over the perception-level to represent goals, the agent learns the ability to infer new goals later encountered by the agent. For example, as in Figure 4(a) (*right*), learning three behaviors with the goal of red-square, red-circle or blue-square (gray background) makes the agent accomplish the new goal of blue-circle (white background). In appendix, we also conduct the ablation study to show how the self-supervised loss (in the perception-level) affects behaviors, and provide more experiments to show the generalization to unseen goals.

Compared to the usual relabeling procedure (e.g., HER with static goals) or latent variable base methods (e.g., DIAYN, equivalent to $f_\kappa = q_\phi$ in our GPIM), our approach scales to dynamic goals. Considering the setting of dynamic relabeling in *fetch* task, we further demonstrate the ability of GPIM on temporally-extended tasks, where the 3D-coordinates of the gripper of the robotic arm is relabeled as goals for ”imitation” in the training phase. During test, we employ a parameterized complex curve, $(x, y, z) = (t/5, \cos(t)/5 - 1/5, \sin(t)/5)$, for the gripper to follow and show their performance in Figure 5 (left). It is worth noting that during training the agent is required to imitate a large number of simple behaviors and has never seen such complex goals before testing. We also validate GPIM on a Franka Panda robot (Figure 5 right)), purposing tracking of hand movement, with MediaPipe (Lugaresi et al. 2019) to capturing features of images. It is observed from Figure 5 that the imitation curves are almost overlapping with the given desired trajectories, indicating that the agent using GPIM framework has the potential to learn such compositional structure of goals during training and generalize to new composite goals during test.

Comparison with baselines. For the *third* question, we mainly compare our method to three baselines: **RIG** (Nair et al. 2018), **DISCERN** (Warde-Farley et al. 2019), and **L2 Distance**. L2 Distance measures the distance between states and goals, where the $L2$ distance $-||\tilde{s}_{t+1} - g_t||^2/\sigma_{pixel}$ is considered with a hyperparameter σ_{pixel}. Note that 2D navigation with the color-shape goal and object manipulation using text description makes the dimensions of states and goals different, so L2 cannot be used in these two tasks. In RIG, we obtain rewards by using the distances in two embedding spaces and learning two independent VAEs, where one VAE is to encode states and the other is to encode goals. For this heterogeneous setting, we also conduct baseline **RIG**$^+$ by training one VAE only on goals and then reward agent with the distance between the embeddings of goals and relabeled states (i.e., g_t vs. $f_\kappa(\tilde{s}_{t+1})$). We use the normalized distance to goals as the evaluation metric, where we generate 50 goals (tasks) as validation.

We show the results in Figure 6 by plotting the normalized distance to goals as a function of the number of actor's steps, where each curve considers 95% confidence interval in terms of the mean value across three seeds. As observed, our GPIM consistently outperforms baselines in almost all tasks except for the RIG in 2D navigation (x-y goal) due to the simplicity of this task. Particularly, as the task complexity increases from 2D navigation (x-y goal) to 2D navigation (color-shape goal) and eventually object manipulation (mixed x-y goal and color-shape goal), GPIM converges faster than baselines and the performance gap between our GPIM and baselines becomes larger. Moreover, although RIG learns fast on navigation with x-y goal, it fails to accomplish complex navigation with color-shape goal because the embedding distance between two independent VAEs has difficulty in capturing the correlation of heterogeneous states and goals. Even with a stable VAE, RIG$^+$ can be poorly suited for training the goal-reaching policy. Especially in high-dimensional action space and on more exploratory tasks (atari and mujoco tasks), our method substantially outperforms the baselines.

To gain more intuition for our method, we record the distance (Δd) between the goal induced by π_μ and the final state induced by π_θ throughout the training process of the 2D navigation (x-y goal). In this specific experiment, we update π_μ and q_ϕ but ignore the update of π_θ before 200 k steps to show the exploration of π_μ at the task generation phase. As shown in Figure 7, Δd steadily increases during the first 200 k steps, indicating that the latent-conditioned policy π_μ explores the environment (i.e., goal space) to dis-

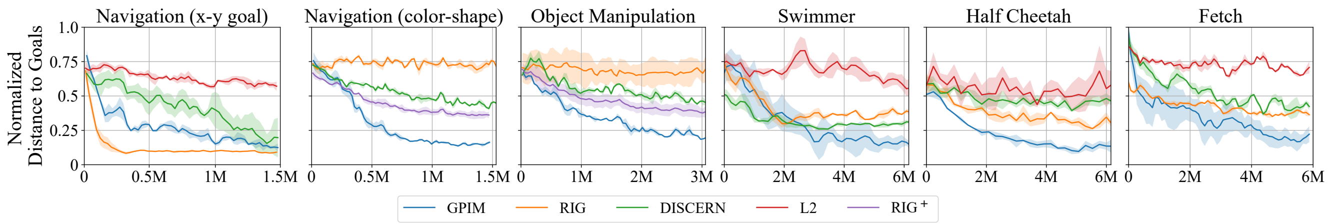


Figure 6: Performance (normalized distance to goals vs. actor steps) of our GPIM and baselines (RIG, DISCERN, L2, RIG$^+$).

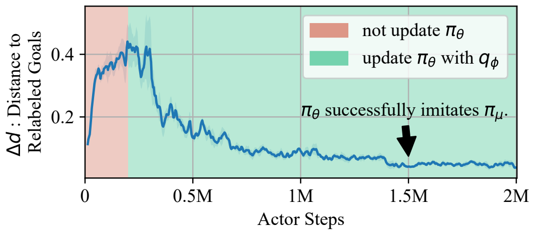


Figure 7: *Pink slice*: Latent-conditioned π_μ gradually explores environment, generating more difficult goals. *Mint green*: Learned discriminator q_ϕ encourages π_θ to mimic π_μ.

tinguish skills more easily (with q_ϕ), and as a result, generates diverse goals for training goal-conditioned policy π_θ. After around 1.5 M steps, Δd almost comes to 0, indicating that goal-conditioned π_θ has learned a good strategy to reach the relabeled goals. In appendix, we visually show the generated goals in more complex tasks, which shows that our straightforward framework can effectively explore without additional sophisticated exploration strategies.

Expressiveness of the reward function. Particularly, the performance of unsupervised RL methods depend on the diversity of generated goals and the expressiveness of the learned reward functions that are conditioned on the goals. We show that our straightforward framework can effectively explore environments in appendix (though it is not our focus). The next question is that: with the same exploration capability to generate goals, does our model achieve competitive performance against the baselines? Said another way, will the obtained rewards (over embedding space) of baselines taking prior non-parametric functions limit the repertoires of learning tasks in some environments? For better graphical interpretation and comparison with baselines, we simplify the complex Atari games to a maze environment shown in Figure 8, where the middle wall poses a bottleneck state. At the same time, as an example to show the compatibility of our objective with existing exploration strategies (Jabri et al. 2019; Lee et al. 2019), we set the reward for the latent-conditioned policy π_μ as $r'_t = \lambda r_t + (\lambda - 1) \log q_\nu(s_{t+1})$, where q_ν is a density model, and $\lambda \in [0, 1]$ can be interpreted as trade off between discriminability of skills and task-specific exploration (here we set $\lambda = 0.5$). Note that we modify r'_t for improving the exploration on generating goals and we do not change the reward for training goal-conditioned π_θ. To guarantee the generation of same diverse goals for goal-conditioned policies of baselines, we adopt π_μ taking the modified r'_t to generate goals for RIG and DISCERN.

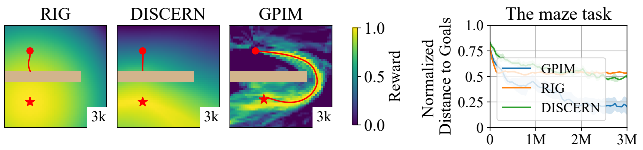


Figure 8: (Left) Reward functions, where heatmaps depict the reward conditioned on the bottom-left goal, reaching the left-bottom star. (Right) Learning curves on the left maze.

In Figure 8, we visualize the learned reward on a specific task reaching the left-bottom star, and the learning curves on the maze task, where the testing-goals are random sampled. We can see that the learned rewards of RIG and DISCERN produce poor signal for the goal-conditioned policy, which makes learning vulnerable to local optima. Our method acquires the reward function q_ϕ after exploring the environment, dynamics of which itself further shapes the reward function. In Figure 8 (left), we can see that our model provides the reward function better expressiveness of the task by compensating for the dynamics of training environment. This produces that, even with the same exploration capability to generate diverse goals, our model sufficiently outperforms the baselines, as shown in Figure 8 (right).

Conclusion

In this paper, we propose GPIM to learn a goal-conditioned policy in an unsupervised manner. The core idea of GPIM lies in that we introduce a latent-conditioned policy with a procedural relabeling procedure to generate tasks (goals and the associated reward functions) for training the goal-conditioned policy. For goal-reaching tasks, we theoretically describe the performance guarantee of our (unsupervised) objective compared with the standard multi-goal RL. We also conduct extensive experiments on a variety of tasks to demonstrate the effectiveness and efficiency of our method.

There are several potential directions for future in our unsupervised relabeling framework. One promising direction would be developing a domain adaptation mechanism when the interaction environments (action/state spaces, dynamics, or initial states) wrt learning π_μ and π_θ are different. Additionally, GPIM can get benefits from more extensive exploration strategies to control the exploration-exploitation trade-off. Finally, latent-conditioned π_μ (generating goals and reward functions) is not affected by the goal-conditioned π_θ in GPIM. One can develop self-paced (curriculum) learning over the two policies under the unsupervised RL setting.

Acknowledgments

The authors would like to thank Yachen Kang for helpful discussions and the anonymous reviewers for the valuable comments. This work is supported by National Science Foundation of China (62176215).

References

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational Option Discovery Algorithms. *CoRR*, abs/1807.10299.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P. F.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *CoRR*, abs/1606.06565.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, O. P.; and Zaremba, W. 2017. Hindsight experience replay. In *Advances in neural information processing systems*, 5048–5058.

Badia, A. P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, D.; and Blundell, C. 2020. Agent57: Outperforming the atari human benchmark. *arXiv preprint arXiv:2003.13350*.

Barber, D.; and Agakov, F. V. 2003. The IM algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, None.

Beaudry, N. J.; and Renner, R. 2011. An intuitive proof of the data processing inequality. *arXiv preprint arXiv:1107.0740*.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *ArXiv*, abs/1606.01540.

Campos, V. A.; Trott, A.; Xiong, C.; Socher, R.; i Nieto, X. G.; and Torres, J. 2020. Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills. *ArXiv*, abs/2002.03647.

Colas, C.; Karch, T.; Lair, N.; Dussoux, J.; Moulin-Frier, C.; Dominey, P. F.; and Oudeyer, P. 2020. Language as a Cognitive Tool to Imagine Goals in Curiosity Driven Exploration. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Colas, C.; Sigaud, O.; and Oudeyer, P.-Y. 2018. Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2018. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

Florensa, C.; Degrave, J.; Heess, N.; Springenberg, J. T.; and Riedmiller, M. 2019. Self-supervised learning of image embedding for continuous control. *arXiv preprint arXiv:1901.00943*.

Gregor, K.; Rezende, D. J.; and Wierstra, D. 2017. Variational Intrinsic Control. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Gupta, A.; Eysenbach, B.; Finn, C.; and Levine, S. 2018. Unsupervised Meta-Learning for Reinforcement Learning. *CoRR*, abs/1806.04640.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Hafner, D.; Ortega, P. A.; Ba, J.; Parr, T.; Friston, K. J.; and Heess, N. 2020. Action and Perception as Divergence Minimization. *CoRR*, abs/2009.01791.

Hartikainen, K.; Geng, X.; Haarnoja, T.; and Levine, S. 2019. Dynamical Distance Learning for Semi-Supervised and Unsupervised Skill Discovery. In *International Conference on Learning Representations*.

Higgins, I.; Pal, A.; Rusu, A.; Matthey, L.; Burgess, C.; Pritzel, A.; Botvinick, M.; Blundell, C.; and Lerchner, A. 2017. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1480–1490. JMLR. org.

Jabri, A.; Hsu, K.; Gupta, A.; Eysenbach, B.; Levine, S.; and Finn, C. 2019. Unsupervised curricula for visual meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, 10519–10531.

Kovač, G.; Laversanne-Finot, A.; and Oudeyer, P.-Y. 2020. Grimgep: learning progress for robust goal sampling in visual deep reinforcement learning. *arXiv preprint arXiv:2008.04388*.

Lee, L.; Eysenbach, B.; Parisotto, E.; Xing, E. P.; Levine, S.; and Salakhutdinov, R. 2019. Efficient Exploration via State Marginal Matching. *CoRR*, abs/1906.05274.

Lee, Y.; Sun, S.-H.; Somasundaram, S.; Hu, E. S.; and Lim, J. J. 2018. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*.

Levy, A.; Konidaris, G.; Platt, R.; and Saenko, K. 2017. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*.

Liu, G.; Zhang, C.; Zhao, L.; Qin, T.; Zhu, J.; Li, J.; Yu, N.; and Liu, T. 2021. Return-Based Contrastive Representation Learning for Reinforcement Learning. *CoRR*, abs/2102.10960.

Lowrey, K.; Rajeswaran, A.; Kakade, S. M.; Todorov, E.; and Mordatch, I. 2019. Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Lu, X.; Lee, K.; Abbeel, P.; and Tiomkin, S. 2020. Dynamics Generalization via Information Bottleneck in Deep Reinforcement Learning. *CoRR*, abs/2008.00614.

Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.; Yong, M. G.; Lee, J.;

Chang, W.; Hua, W.; Georg, M.; and Grundmann, M. 2019. MediaPipe: A Framework for Building Perception Pipelines. *CoRR*, abs/1906.08172.

Nair, A.; Bahl, S.; Khazatsky, A.; Pong, V.; Berseth, G.; and Levine, S. 2019. Contextual Imagined Goals for Self-Supervised Robotic Learning. *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, 100: 530–539.

Nair, A. V.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; and Levine, S. 2018. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, 9191–9200.

Péré, A.; Forestier, S.; Sigaud, O.; and Oudeyer, P.-Y. 2018. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. *arXiv preprint arXiv:1803.00781*.

Pong, V.; Gu, S.; Dalal, M.; and Levine, S. 2018. Temporal difference models: Model-free deep rl for model-based control. *arXiv preprint arXiv:1802.09081*.

Pong, V. H.; Dalal, M.; Lin, S.; Nair, A.; Bahl, S.; and Levine, S. 2019. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*.

Popov, I.; Heess, N.; Lillicrap, T.; Hafner, R.; Barth-Maron, G.; Vecerik, M.; Lampe, T.; Tassa, Y.; Erez, T.; and Riedmiller, M. 2017. Data-efficient deep reinforcement learning for dexterous manipulation. *arXiv preprint arXiv:1704.03073*.

Salge, C.; Glackin, C.; and Polani, D. 2014. Empowerment–an introduction. In *Guided Self-Organization: Inception*, 67–114. Springer.

Schaul, T.; Horgan, D.; Gregor, K.; and Silver, D. 2015. Universal value function approximators. In *International conference on machine learning*, 1312–1320.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR*, abs/1707.06347.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; Levine, S.; and Brain, G. 2018. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1134–1141. IEEE.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2020. Dynamics-Aware Unsupervised Discovery of Skills. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.*

Sukhbaatar, S.; Denton, E.; Szlam, A.; and Fergus, R. 2018. Learning goal embeddings via self-play for hierarchical reinforcement learning. *arXiv preprint arXiv:1811.09083*.

Sukhbaatar, S.; Lin, Z.; Kostrikov, I.; Synnaeve, G.; Szlam, A.; and Fergus, R. 2017. Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*.

Tian, Q.; Liu, J.; Wang, G.; and Wang, D. 2021. Unsupervised Discovery of Transitional Skills for Deep Reinforcement Learning. *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, 5026–5033. IEEE.

Vecerik, M.; Sushkov, O.; Barker, D.; Rothörl, T.; Hester, T.; and Scholz, J. 2019. A practical approach to insertion with variable socket position using deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, 754–760. IEEE.

Warde-Farley, D.; de Wiele, T. V.; Kulkarni, T. D.; Ionescu, C.; Hansen, S.; and Mnih, V. 2019. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Xu, G.; Liu, Z.; Li, X.; and Loy, C. C. 2020. Knowledge Distillation Meets Self-Supervision. *arXiv preprint arXiv:2006.07114*.

Zhang, J.; Yu, H.; and Xu, W. 2021. Hierarchical Reinforcement Learning By Discovering Intrinsic Options. *CoRR*, abs/2101.06521.