LEARNING DIVERSE GOAL-SPECIFIC SKILLS VIA LATENT EMBEDDING

Changzhi Yan*

ABSTRACT

To be added...

1 Introduction

To be added...

2 Background

MDP Consider the standard reinforcement learning formalism in the infinite-horizon discounted Markov Decision Process (MDP) setting specified by its state space $\mathcal{S}$, action space $\mathcal{A}$, transition probabilities $p(s_{t+1}|s_t, a_t)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in [0, 1)$, and initial state distribution μ. A stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping from states to probability distributions over actions, where $\Delta(\mathcal{A})$ is the space of probability distributions over $\mathcal{A}$. Every episode starts with sampling an initial state $s_0 \sim \mu$. At every timestep t, the agent takes an action based on the current state $a_t \sim \pi(\cdot \mid s_t)$ and gets a reward $r_t = r(s_t, a_t)$, and then steps into a new state sampled from the transition distribution $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. A discounted sum of future rewards is called a return, $R_t \coloneqq \sum_{i=t}^{\infty} \gamma^{i-t} r_i$. The value function of state s under policy π is defined as the expected return: $V^\pi(s) \coloneqq \mathbb{E}_\pi[R_0 \mid s_0 = s]$, where $\mathbb{E}_\pi[\cdot]$ refers to the expectation under the distribution of trajectories induced by policy π. Similarly, the state-action value (or Q-value) function is given as $Q^\pi(s, a) \coloneqq \mathbb{E}_\pi[R_0 \mid s_0 = s, a_0 = a]$. Denote $V^\pi(\mu) \coloneqq \mathbb{E}_{s\sim\mu}[V^\pi(s)]$. Given the initial state distribution μ, there exists an optimal policy $\pi^\star$ such that $V^\star(\mu) \coloneqq V^{\pi^\star}(\mu) = \max_\pi V^\pi(\mu)$.

UVFA Universal Value Function Approximators (UVFA) (Schaul et al. [2015]) is an extension of Deep Q-Networks (DQN) (Mnih et al. [2015]) to the setup where the agent tries to achieve multiple different goals, which we refer to as the *multi-goal* setup. Let $\mathcal{G}$ be the space of all possible goals. Every goal $g \in \mathcal{G}$ corresponds to a goal-specific reward function, $r_g : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. In this setting, the policy gets as input not only a state but also a goal, $\pi : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$, called the *universal* policy. At the beginning of each episode, a goal is sampled from a distribution $p_g(\cdot)$ and remains fixed throughout the entire episode. At every timestep, the agent takes an action based on the current state and goal, $a_t \sim \pi(\cdot \mid s_t, g)$, and receives a reward $r_t = r_g(s_t, a_t)$. Correspondingly, the value function now depends not only on a state but also on a goal: $V^\pi(s, g) = \mathbb{E}_\pi[R_0 \mid s_0 = s, g]$, called the *universal* value function. The universal Q-value function can be similarly defined as $Q^\pi(s, a, g) = \mathbb{E}_\pi[R_0 \mid s_0 = s, a_0 = a, g]$. It is possible to train an approximator to the universal Q-value function using direct bootstrapping from the Bellman equation (i.e., fitted-Q iteration), and a greedy policy derived from it can generalize to previously unseen states.

HER Hindsight Experience Replay (HER) (Andrychowicz et al. [2017]) is a replay technique that allows for sample-efficient learning from sparse and binary rewards in the aforementioned multi-goal setting. The primary difficulty of learning in a sparse reward environment is that the agent rarely visits a goal state. However, a trajectory that fails to reach the goal can be considered successful if the goal state is replaced by its final state itself. HER performs exactly this kind of reasoning by replaying each episode with a different goal than the one the agent tries to achieve, e.g., one of the goals that has been achieved in the episode, thus making the agent receive reward signals much more frequently. For simplicity, let $\mathcal{G} = \mathcal{S}$, and let the reward be 0 if the goal is reached and -1 otherwise. Consider a trajectory with

*This paper comes from a chapter in my M.E. thesis. (In this paper, all the expressions "we" actually mean "I.")

a state sequence $s_0, \ldots, s_T$ and a goal $g \neq s_0, \ldots, s_T$, which implies that the agent receives a reward of -1 at every timestep. The key idea behind HER is to re-examine this trajectory with a different goal — while this trajectory tells us little about how to reach the state g, it definitely provides useful information about how to achieve the state s_T. This information can be harvested by using an off-policy RL algorithm with experience replay, where we replace g by s_T for that trajectory in the replay buffer. The raw trajectory without goal replacement will also be replayed. In this way, at least half of the trajectories in the replay buffer contain rewards other than -1 and thus make learning much simpler. Note that the goal achieved in the final state of the episode is just one choice of additional goals used for replay, and there are other strategies for choosing goals to replay trajectories with. HER can be combined with any off-policy RL algorithm, and with the integration of function approximators, it allows the agent to learn how to reach goals that are not observed during training.

DIAYN Diversity is All You Need (DIAYN) (Eysenbach et al. [2018]) is a method for learning useful skills without a reward function. A skill is a latent-conditioned policy whose behavior is altered by a latent variable. DIAYN learns task-agnostic (i.e., without the prior knowledge of the task) skills without supervision[1], which can serve as primitives for hierarchical RL to solve many different tasks, as the hierarchical composition of such skills can not only shorten the planning horizon but also increase the agent's exploration ability. Desired skills should have maximal coverage over the set of possible behaviors. For that purpose, DIAYN proposes an information-theoretic objective with a maximum entropy policy, maximizing which yields skills that are not only *distinguishable* but also as *diverse* as possible. While one particular skill may exhibit a useless behavior, other skills should behave differently from it, thus being more useful. To enable discriminability between skills, one can explicitly maximize the mutual information between states and latent variables. By learning distinguishable skills that act as randomly as possible, we can "push" the skills away from each other, making them as diverse as possible, so that collectively they can explore the state space effectively. This can be achieved by using maximum entropy RL to train skills with the mutual information mentioned above as the surrogate reward.

3 Goal-Specific Skill Discovery and Transfer

We consider skill discovery in the multi-goal RL setting and explore how discovered skills can be utilized for complex, goal-specific tasks. We first learn a set of admissible skills in the "source domain," and then compose these skills under a hierarchical mechanism to solve downstream tasks in the "target domain." Conceivably, the environments in these two domains should share some structural commonalities.

3.1 Diverse Goal-Specific Skill Learning

We aim to acquire useful skills in multi-goal, sparse-reward environments (i.e., source domain), which can be leveraged for hierarchical RL to solve challenging downstream tasks in similar environments. To that end, we propose a method called "Diverse Goal-Specific Skill Learning" (DGSL) to learn such admissible skills.

3.1.1 Principle

For skills to be useful, the learned skills should encompass as many different behaviors as possible. At the state level, each skill should be distinct, while the skills collectively should explore a significant portion of the state space. To that end, DGSL builds on three ideas. First, different skills should induce distinct trajectories, thus being *distinguishable*. Second, we incentivize skills to be as *diverse* as possible by learning skills that not only are distinguishable but also act as randomly as possible, as skills with high entropy while remaining discriminable must explore parts of the state space far away from each other, lest the randomness in their actions lead them to states where they cannot be distinguished. Note that discriminability does not necessarily imply diversity — a slight difference in visited states makes two skills distinguishable, but not necessarily diverse in a semantically meaningful way. Such distinguishable and diverse skills can be understood as the basis of a skill space[2], as their "combination" can represent any possible behavior. Finally, each skill should have the potential to achieve goals in sparse reward environments, thereby providing prior knowledge for goal-specific downstream tasks. We believe that such skills can serve as desirable motion primitives for hierarchical RL in the multi-goal setup, and the prior knowledge of achieving goals obtained in the source domain can be transferred into the target domain to accelerate learning via hierarchical composition of these skills.

[1] In the context of skill discovery, supervision refers to learning from the reward of human design. By maximizing the reward, the agent is implicitly learning from human supervision. Learning skills without a reward function can be understood as unsupervised skill discovery.

[2] This is similar to the basis of a vector space, which consists of the maximal linearly independent vectors.

While it is straightforward to use the universal value function approximators and hindsight experience replay to train goal-conditioned policies to achieve goals in sparse reward environments, there is no guarantee that $\pi(a \mid s, g)$ should be diverse and distinguishable. To combat this issue, we map a goal g into a conditional distribution of latent variables: $g \to p(\cdot|g)$, and require the latent-conditioned policies (i.e., skills) $\pi(a \mid s, z)$ to achieve the goal g instead, where z is sampled from $p(\cdot|g)$. In this way, $\pi(a \mid s, z)$ can be diverse and distinguishable, if we find a proper embedding network, parameterized by ψ, from the goal space $\mathcal{G}$ to the latent space $\mathcal{Z}$. Let $p_\psi(\cdot \mid g)$ denote the distribution of latent variables embedded from the given goal g. $\mathcal{H}(\cdot)$ and $\mathcal{I}(\cdot;\cdot)$ refer to Shannon entropy and mutual information, respectively, both computed with base e. We construct a simple task-agnostic objective based on several information-theoretic terms:[3]

$\max \mathcal{I}(Z; G)$ We want latent variables and goals to be highly correlated. Distinct goals should be mapped to different sets of latent variables. On the one hand, for skills to be more diverse, our algorithm is expected to frequently sample as many different skills as possible, which leads to maximizing $\mathcal{H}(Z)$. On the other hand, minimizing $\mathcal{H}(Z \mid G)$ would incentivize the embedding distribution $p_\psi(\cdot \mid g)$ to be sharp enough to diminish the potential overlap between sets of embedding latent variables mapped from different goals. In aggregate, we maximize $\mathcal{H}(Z) - \mathcal{H}(Z \mid G) = \mathcal{I}(Z; G)$.

$\max \mathcal{I}(S; Z \mid G)$ Given a goal, we maximize the mutual information between states and latent variables to encode the idea that the skill should determine the states visited, and in turn, that the skill can be inferred from the states the agent visits. This term imposes discriminability between skills.

$\min \mathcal{I}(A; Z \mid S, G)$ We want to use states, not actions, to distinguish skills. Actions that do not affect the environment are not visible to the outside observer. For example, an outside observer cannot determine the force a robotic arm applies when grasping a cup if the cup remains stationary.

$\max \mathcal{H}(A \mid S, G)$ Since $\pi(a \mid s, g)$ can be viewed as a mixture of skills $\pi(a \mid s, z)$ weighted by $p_\psi(z \mid g)$, we maximize the entropy of this mixture policy to encourage skills to develop different patterns of behaviors, leading them to be as diverse as possible.

In summary, we maximize

$$\begin{aligned}
&\mathcal{I}(Z; G) + \mathcal{I}(S; Z \mid G) - \mathcal{I}(A; Z \mid S, G) + \mathcal{H}(A \mid S, G) \\
=&(\mathcal{H}(Z) - \mathcal{H}(Z \mid G)) + (\mathcal{H}(Z \mid G) - \mathcal{H}(Z \mid S, G)) - (\mathcal{H}(A \mid S, G) - \mathcal{H}(A \mid S, G, Z)) + \mathcal{H}(A \mid S, G) \\
=&\mathcal{H}(Z) - \mathcal{H}(Z \mid S, G) + \mathcal{H}(A \mid S, Z)
\end{aligned} \tag{1}$$

Equation 1 presents an alternative interpretation of how diverse and distinguishable skills can be learned in the multi-goal RL setup. The first term encourages high entropy of the latent variable distribution $p_\psi(z) = \int p_\psi(z|g)p(g)\, dg$, from which diverse skills would be sampled more frequently. Notice that $\exp(\mathcal{H}(Z))$ can serve as a measurement of the effective number of skills. The second term suggests that it should be straightforward to infer the skill the agent is using from the current state and goal. This term essentially reflects the discriminability between skills embedded from the goal g. The third term represents the expected entropy of skills, showing that each skill should behave as randomly as possible. The random variable G is omitted in its conditional terms, as the latent variable itself can fully determine a skill. This interpretation aligns with our aforementioned observation that diverse skills can be obtained by learning skills that are not only distinguishable but also as random as possible.

Let the skill be parameterized by θ. Denote $\mathbb{E}[\cdot] = \mathbb{E}_{g\sim \mathrm{U}(\mathcal{G}), z\sim p_\psi(\cdot|g), s\sim p_\theta(\cdot|z), a\sim\pi_\theta(\cdot|s,z)}[\cdot]$, where $\mathrm{U}(\mathcal{G})$ is the uniform distribution over the goal space $\mathcal{G}$, and $p_\theta(\cdot \mid z)$ is the state distribution induced by the skill $\pi_\theta(a \mid s, z)$. In addition to Equation 1, we maximize $\mathbb{E}[r(s, a, g)]$ to ensure that each skill has the potential[4] to achieve the goal. At the start of each episode, we uniformly sample a goal in the goal space, $g \sim \mathrm{U}(\mathcal{G})$, based on which we sample a latent variable from the embedding distribution, $z \sim p_\psi(\cdot \mid g)$, and run the corresponding skill $\pi_\theta(a \mid s, z)$ throughout the episode in the hope that the skill can lead the agent to reach the goal. Putting everything together, we maximize

$$\mathcal{F}(p_\psi, \pi_\theta) \coloneqq \mathcal{H}(Z) - \mathcal{H}(Z \mid S, G) + \alpha\mathcal{H}(A \mid S, Z) + \beta\mathbb{E}[r(s, a, g)] \tag{2}$$

where two scale factors are added to regulate the behavior of learned skills. α reflects the trade-off between exploration and discriminability. Large α inclines the skills towards exploration, yet making discriminability harder. β influences the degree to which skills can perform the goal-specific task. Large β enforces skills to focus on goal achieving, but it will pose challenges to diversity. Next, we come up with a practical method to maximize $\mathcal{F}(p_\psi, \pi_\theta)$.

[3]For variables, we use the capital and calligraphic letter to denote the random variable and the space of variables, respectively. For example, Z is the random variable for the latent, while $\mathcal{Z}$ represents the latent space.

[4]We only require that the skills can perform the goal-specific tasks to some extent, not necessarily that they can perfectly achieve the goal.

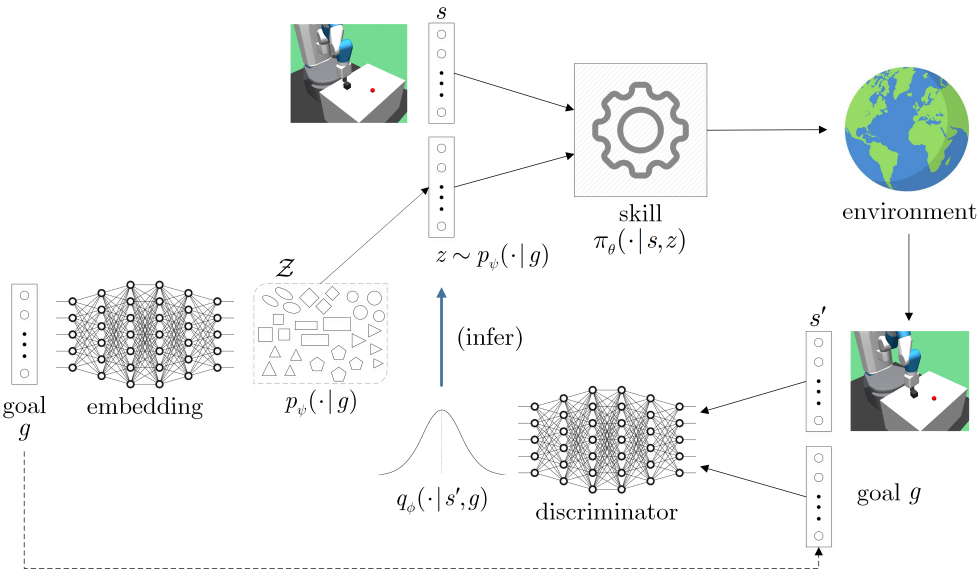

Figure 1: DGSL mechanism.

3.1.2 Implementation

The only thing that remains unknown is the posterior $p(z \mid s, g)$ for computing $\mathcal{H}(Z \mid S, G)$. We approximate it with a learned discriminator $q_\phi(z \mid s, g)$, using a neural network parameterized by ϕ, that takes the state and goal as input and outputs the approximated posterior distribution $q_\phi(\cdot \mid s, g)$. The following lemma presents a tractable lower bound on $-\mathcal{H}(Z \mid S, G)$ with respect to q_ϕ.

Lemma 1. *For any distribution $q_\phi(\cdot)$, we have*

$$-\mathcal{H}(Z \mid S, G) \geq \mathbb{E}_{g \sim \mathrm{U}(\mathcal{G}), z \sim p_\psi(\cdot \mid g), s \sim p_\theta(\cdot \mid z)}[\log q_\phi(z \mid s, g)]. \tag{3}$$

Proof. Since $D_{\mathrm{KL}}(p \,\|\, q_\phi) \geq 0$, we have $\int p(z \mid s, g) \log p(z \mid s, g)\, dz \geq \int p(z \mid s, g) \log q_\phi(z \mid s, g)\, dz,\ \forall p, q_\phi$.

$$\begin{aligned}
-\mathcal{H}(Z \mid S, G) &= \iint p(s, g) \int p(z \mid s, g) \log p(z \mid s, g)\, dz\, ds\, dg \\
&\geq \iint p(s, g) \int p(z \mid s, g) \log q_\phi(z \mid s, g)\, dz\, ds\, dg \\
&= \int p(g) \iint p(s, z \mid g) \log q_\phi(z \mid s, g)\, dz\, ds\, dg \\
&= \int p(g) \int p(z \mid g) \int p(s \mid z, g) \log q_\phi(z \mid s, g)\, ds\, dz\, dg \\
&= \mathbb{E}_{g \sim \mathrm{U}(\mathcal{G})}\big[\mathbb{E}_{z \sim p_\psi(\cdot \mid g)}\big[\mathbb{E}_{s \sim p_\theta(\cdot \mid z)}[\log q_\phi(z \mid s, g)]\big]\big]\,.
\end{aligned}$$

The last step follows due to $p(s \mid z, g) = p_\theta(s \mid z)$. □

This lemma gives us a variational lower bound $\mathcal{L}(p_\psi, q_\phi, \pi_\theta)$ on $\mathcal{F}(p_\psi, \pi_\theta)$:

$$\begin{aligned}
\mathcal{F}(p_\psi, \pi_\theta) &\geq \mathcal{H}(Z) + \mathbb{E}_{g \sim \mathrm{U}(\mathcal{G}), z \sim p_\psi(\cdot \mid g), s \sim p_\theta(\cdot \mid z)}[\log q_\phi(z \mid s, g)] \\
&\quad + \alpha \mathbb{E}_{z \sim p_\psi(\cdot \mid g), s \sim p_\theta(\cdot \mid z)}[\mathcal{H}(\pi_\theta(\cdot \mid s, z))] + \beta \mathbb{E}[r(s, a, g)] \\
&= \mathcal{H}(Z) + \mathbb{E}[\log q_\phi(z \mid s, g) + \beta r(s, a, g) + \alpha \mathcal{H}(\pi_\theta(\cdot \mid s, z))] \\
&:= \mathcal{L}(p_\psi, q_\phi, \pi_\theta)
\end{aligned} \tag{4}$$

Since $\mathcal{F}$ is intractable to compute, we maximize its lower bound $\mathcal{L}$ instead, which consists of two terms that can be dealt with separately. Being a function of p_ψ, the first term $\mathcal{H}(Z)$ is independent of π_θ, meaning that the embedding network can be optimized as a preparation step before the agent interacts with the environment. The second term can be handled by the maximum entropy RL, which maximizes the policy's entropy over actions and thus copes with the

Algorithm 1 Diverse Goal-Specific Skill Learning (DGSL)

Train the embedding network: $p^\star_\psi \leftarrow \arg\max \mathcal{H}(Z)$ ▷ w.r.t. the goal space $\mathcal{G}$
while not converged **do**
  Sample a goal uniformly from the goal space: $g \sim \mathrm{U}(\mathcal{G})$
  for episode $n = 1$ to N **do**
    Sample a latent variable $z \sim p^\star_\psi(\cdot \mid g)$ and an initial state $s_0 \sim \mu$
    for timestep $t = 0$ to $T - 1$ **do**
      Sample an action from the skill: $a_t \sim \pi_\theta(\cdot \mid s_t, z)$
      Step the environment: $s_{t+1} \sim p(\cdot \mid s_t, a_t)$
      Compute the surrogate reward: $\widetilde{r}_t = \log q_\phi(z \mid s_{t+1}, g) + \beta r(s_t, a_t, g)$
      Store the transition $(s_t \| z, a_t, \widetilde{r}_t, s_{t+1} \| z)$ in the replay buffer $\mathcal{R}$ ▷ $\|$ denotes concatenation
      for gradient step $m = 1$ to M **do**
        Sample a minibatch $\mathcal{B}$ from the replay buffer $\mathcal{R}$
        Perform one gradient step on θ to maximize $\widetilde{r}_t$ using SAC and minibatch $\mathcal{B}$
        Perform one gradient step on ϕ to maximize $q_\phi(z \mid s_{t+1}, g)$ with SGD
      end for
    end for
  end for
end while

$\mathcal{H}(\pi_\theta)$ term in our objective $\mathcal{L}$. Specifically, we use Soft Actor-Critic (Haarnoja et al. [2018]) to optimize the second term in Equation 4 by replacing the task reward with the following pseudo-reward:

$$r_g(s, a, z) \coloneqq \log q_\phi(z \mid s, g) + \beta r(s, a, g) \tag{5}$$

which indicates that the agent is rewarded not only for reaching the goal, but also for visiting states that are easy to discriminate. (This is why q_ϕ gets the name "discriminator.") Note that the pseudo-reward is continuous, even though the task reward $r(s, a, g)$ is sparse. The discriminator is updated to better infer the skill z from its induced states and the goal it is embedded from, meaning that $q_\phi(z \mid s, g)$ is maximized (with respect to ϕ) for the skill z that the agent is currently using.

A schematic diagram of the learning mechanism is shown in Figure 1. The corresponding algorithm for optimizing the variational lower bound $\mathcal{L}$, called Diverse Goal-Specific Skill Learning (DGSL), is presented in Algorithm 1. Similar to UVFA, we augment the states stored in the replay buffer by concatenating them with latent variables to incorporate information from the latent variables into the agent. DGSL returns a policy network $\pi^\star_\theta(a \mid s, z)$ that represents diverse and distinguishable skills for different z, which we will use as a building block for hierarchical RL to solve downstream tasks in the next section. As a remark, the two key optimization steps in DGSL (updating θ and ϕ) form a cooperative game, thereby avoiding the instability of adversarial saddle-point formulations in many other adversarial unsupervised RL methods.

3.2 Using Skills for Hierarchical RL

A common practice of using hierarchical RL to solve downstream tasks is to compose and interpolate skills by running different skills at different timesteps in an episode. In theory, hierarchical RL should decompose a complex task into motion primitives that can be reused for multiple tasks. In practice, hierarchical RL algorithms may face numerous challenges: (1) the hierarchical policy only samples a single motion primitive (Gregor et al. [2016]), or (2) all motion primitives endeavor to do the entire task. In contrast, DGSL identifies diverse, goal-specific skills, which have the potential to serve as building blocks for hierarchical RL in addressing multi-goal tasks.

To use the learned skills in downstream tasks, the environment must exhibit some similarities to that in the pretraining stage (i.e., the skill discovery stage), which we refer to as the *structural assumptions* of MDPs between the source and target domains. We use $\mathcal{M}$ to denote the set of downstream task MDPs. For any $M \in \mathcal{M}$, its state space $\mathcal{S}_M$ consists of two parts, $\mathcal{S}_M^{\text{agent}}$ and $\mathcal{S}_M^{\text{goal}}$, where $\mathcal{S}_M^{\text{agent}}$ represents the state of the agent itself, independent of the task, and $\mathcal{S}_M^{\text{goal}}$ identifies the goal-related part of the state space. In principle, downstream tasks should also be multi-goal with sparse rewards, yet more complex, and consequently, they share the same $\mathcal{S}_M^{\text{agent}}$ if we intend to reuse skills across multiple downstream tasks.

structural assumptions For any downstream task $M \in \mathcal{M}$, $\mathcal{S}_M^{\text{agent}}$ is identical to the state space of the pretraining environment. Moreover, the action spaces are identical between the pretraining and downstream task environments.

To utilize the discovered skills for hierarchical RL, we learn a meta-controller, parameterized by φ, whose action is to select which skill to execute for the next k steps. For a downstream task M, the meta-controller π_φ takes as input the current state $s_M = (s_M^{\text{agent}}, s_M^{\text{goal}})$, and outputs a latent variable $z \sim \pi_\varphi(\cdot \mid s_M)$, feeding it to the pretrained skill network to form the current skill $\pi_\theta(\cdot \mid s_M^{\text{agent}}, z)$, and run this skill for the subsequent k timesteps. Note that the skill network is learned in the pretraining stage and, by the structural assumptions, $s_M^{\text{agent}} = s$. With the skill network fixed, the meta-controller is trained to maximize the task reward, which is a standard RL problem. In this manner, the hierarchical policy π_φ learns to compose and interpolate discovered skills for downstream tasks, which possess certain capabilities to achieve goals, thereby enhancing sample efficiency.

4 Experiments

To be added...

5 Conclusions and Future Work

To be added...

References

M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, page 5055–5065, 2017.

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

K. Gregor, D. J. Rezende, and D. Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1861–1870, 2018.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1312–1320, 2015.

A Experimental Details

A.1 appendixA1

To be added...

A.2 appendixA2

To be added...