Iterative Unsupervised Skill Learning

Eric Lin and Catherine Zeng

Harvard University CS 282r Final Project

Introduction and Motivation

- Reinforcement learning (RL) approaches generally fail in environments with no or sparse rewards. We explore learning skills without supervision.
- Unsupervised skill learning methods often require a pre-specified number of skills.
- We experiment with iterative skill learning, where we automatically detect when we have learned a sufficient number of skills.

Iterative learning aims to:

1. eliminate need to finetune number of skills
2. speed up training, since averaged over episodes, a fewer number of skills are trained

Notation

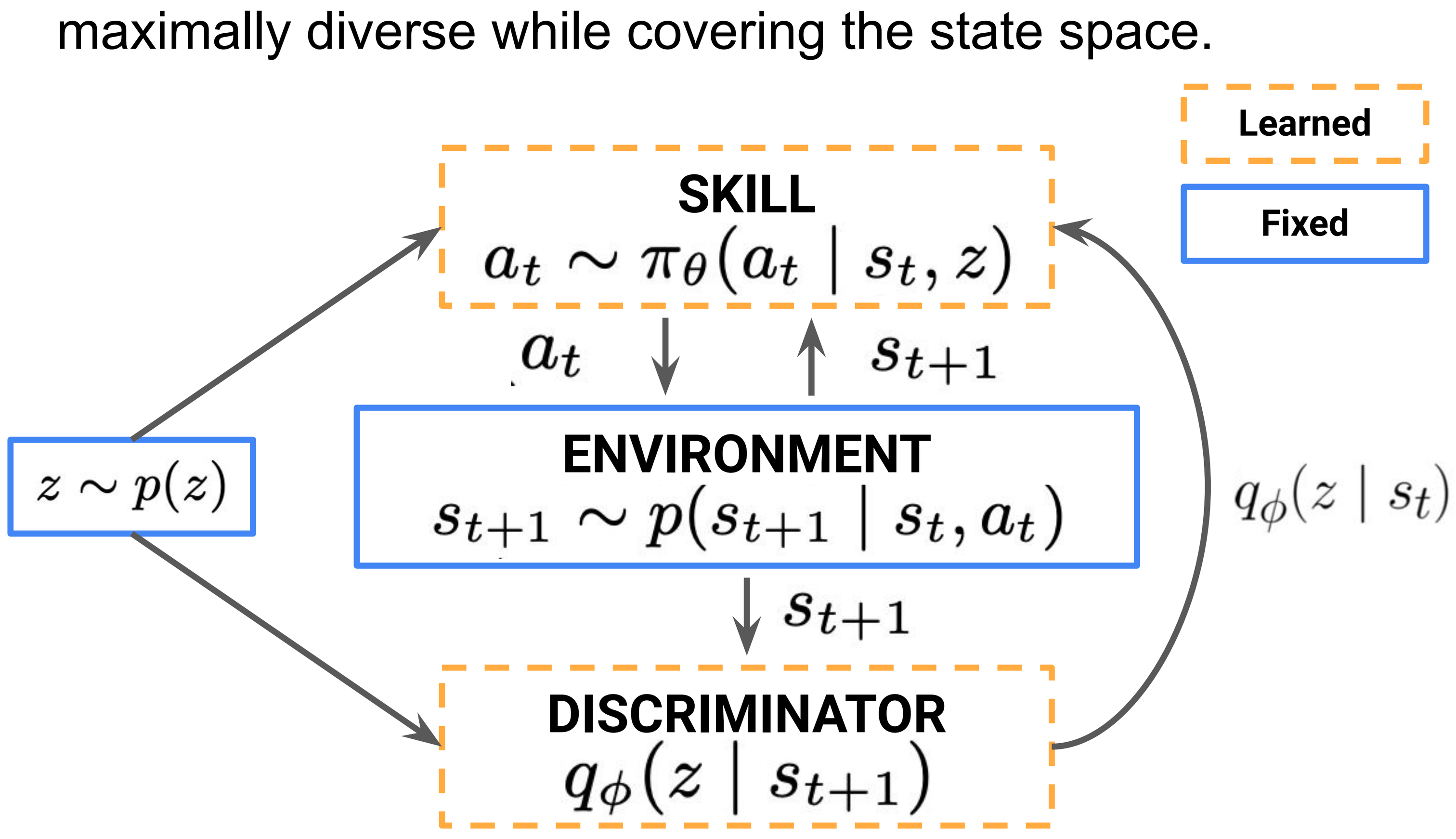
S, A	random variables for states and actions
$Z \sim p(z)$	latent variable on which we condition policies ('skills')
$\mathcal{I}(\cdot\,;\cdot)$	mutual information
$\mathcal{H}[\cdot]$	Shannon entropy

Background

- We base our method primarily on Diversity is All You Need (*DIAYN*, Eysenbach et al.), which maximizes an information-theoretic objective with a maximum entropy policy:

$$\mathcal{F}(\theta) = \mathcal{H}[Z] - \mathcal{H}[Z|S] + \mathcal{H}[A|S, Z]$$

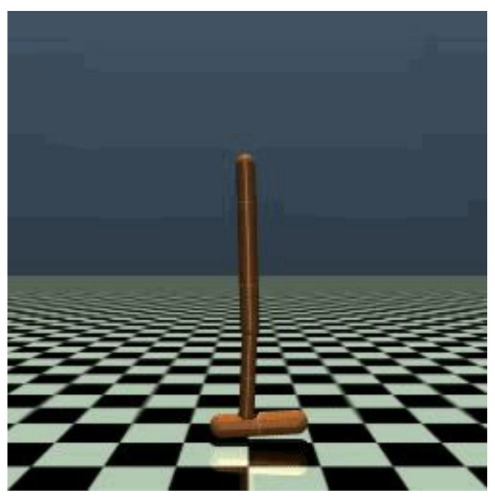
- *DIAYN* encourages skills (latent-conditioned policies) to be maximally diverse while covering the state space.

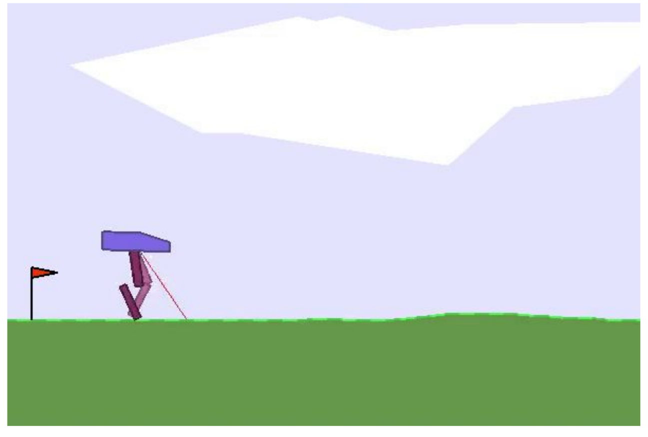


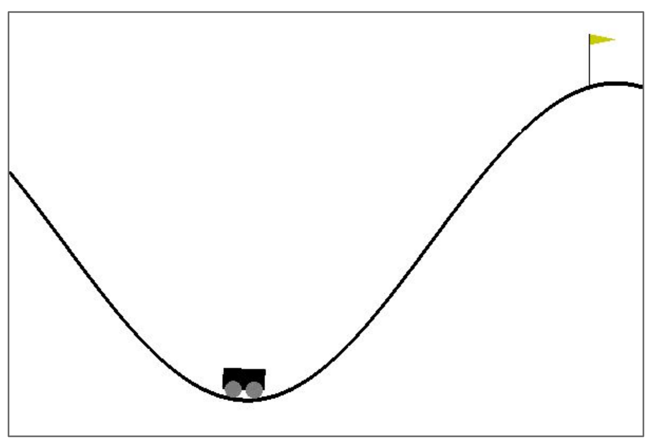
We extend *DIAYN* by proposing and comparing a number of iterative skill learning approaches.

Environments

We test our approaches on *Hopper* (right), *BipedalWalker* (bottom left), and *MountainCar Continuous* (bottom right).







Approach

- We begin with DIAYN's skill-learning technique, which uses Soft Actor-Critic (SAC) with a diversity reward:

$$r_z(s, a) = \log q_\phi(z|s) - \log p(z)$$

- In iterative learning, we must choose (i) when to increase the number of skills and (ii) how much to increase it by.

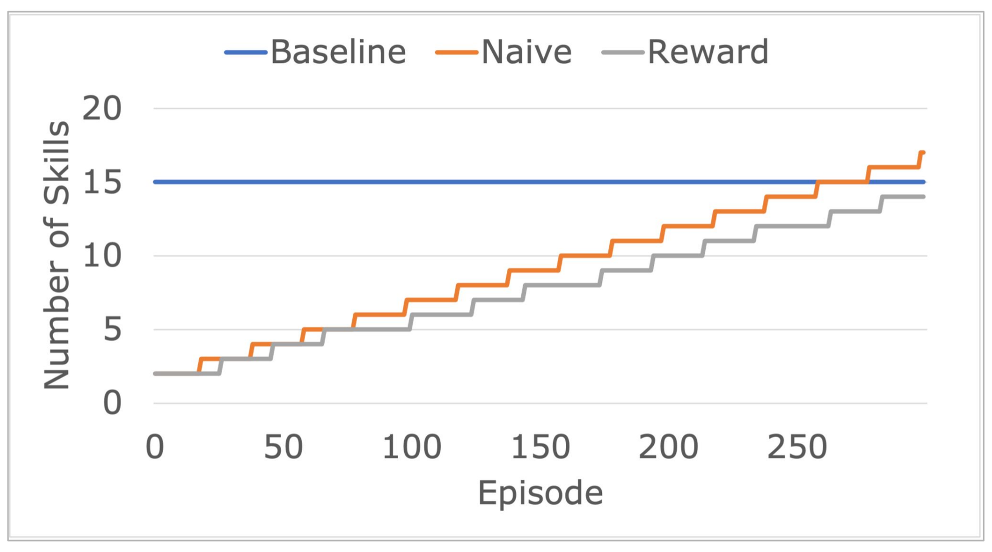
For (i), we compare five methods:

1. *Baseline*: specifying a constant number of skills up-front
2. *Naive*: increment the number of skills every *n* episodes
3. *Reward*: increment the number of skills when the max reward has stayed constant for the past *n* episodes
4. *Diverse1*: increment once the *DIAYN* diversity reward has reached a certain threshold
5. *Diverse2*: increment once the skills differ enough from each other, as determined by the differences in the actions that are sampled

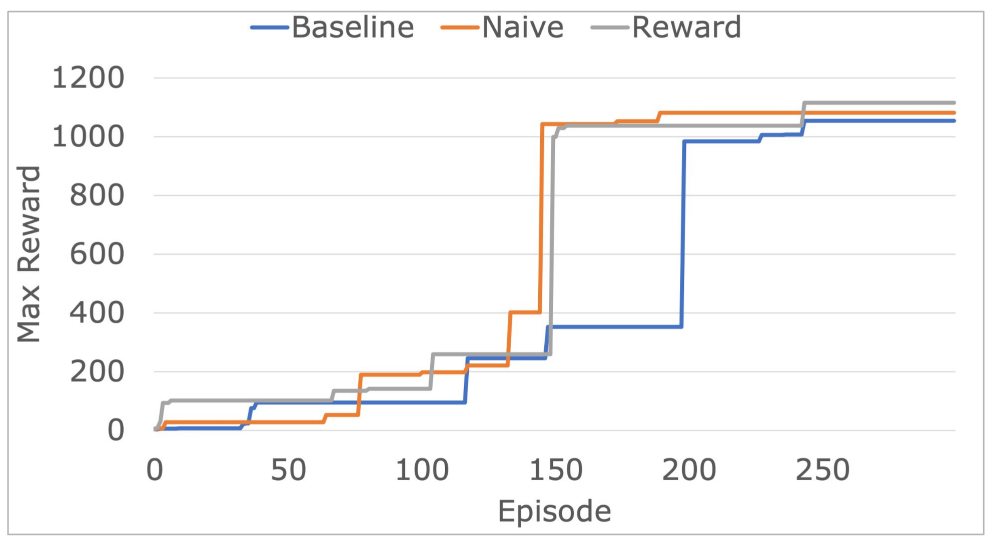
For (ii), we compare two increment styles:

1. *Constant*: increasing the number of the skills by a constant increment *k*
2. *Decay*: decreasing the skill increment *k* over time

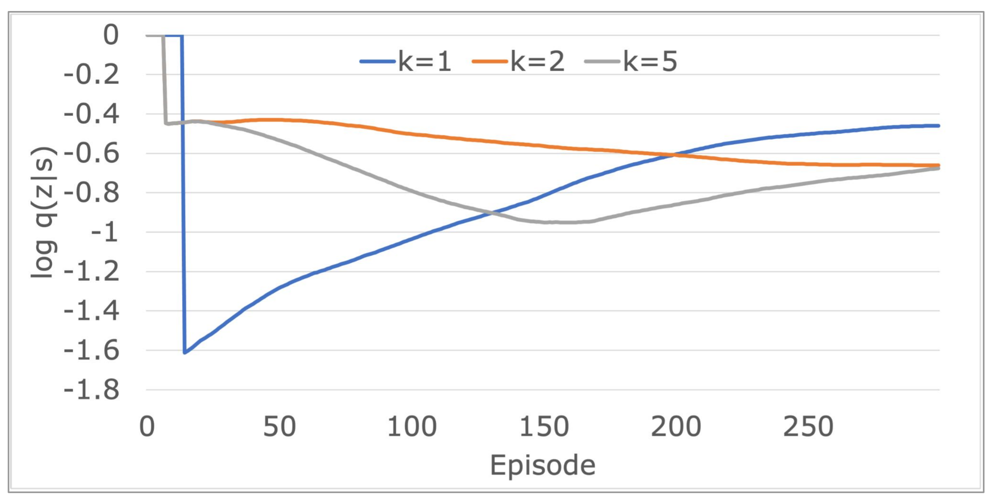
Preliminary Experiments



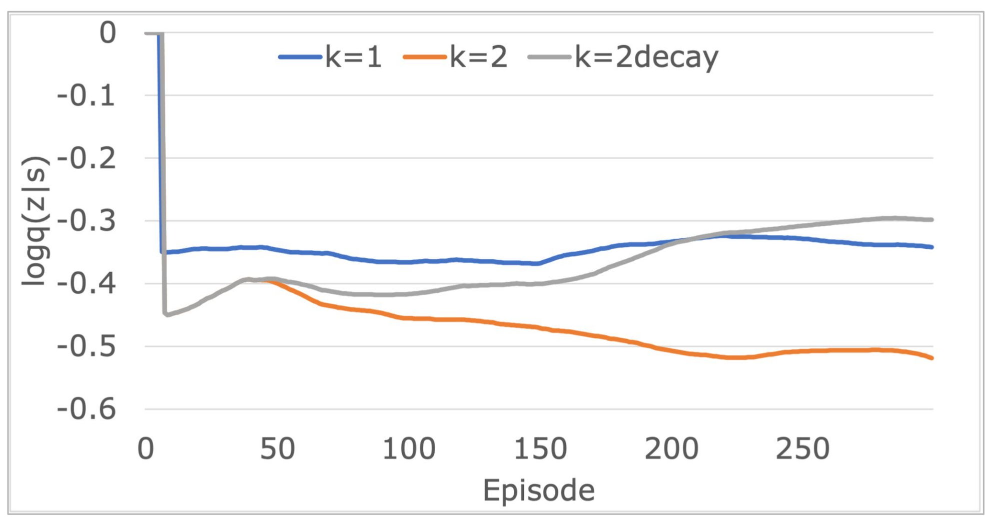
Visualization of the number of skills learned for different approaches on *Hopper*. Naive and reward approaches use *k*=1 increment.



Comparison of the maximum reward achieved on *Hopper* by the three approaches mentioned above.



Comparison of the diversity award for the naive method at different *k* on *Hopper*.



Comparison of the diversity award for the reward method using varying *k* on *Hopper*. *k* = 2decay refers to starting with *k*=2 and multiplying *n* by 1.3 everytime skill is incremented.

Conclusions

- Iterative reward learning can achieve the same level of max reward with fewer skills and less training time.
- Decaying the skill increment is a promising approach.
- We will continue to test different methods and run more experiments for our final paper.