Iterative Unsupervised Skill Learning

Eric Lin
Harvard University
eric_lin@college.harvard.edu

Catherine Zeng
Harvard University
catherinezeng@college.harvard.edu

Abstract

In environments with no or sparse rewards, unsupervised skill learning can learn policies unattainable by reinforcement learning (RL) approaches. However, unsupervised skill learning methods often require a pre-specified number of skills. One such method is Diversity is All You Need (DIAYN), which combines theoretical motivations and empirical success. We introduce various approaches for *iteratively* learning the number of skills for DIAYN. In doing so, we automate the number of skills selected and we improve training efficiency (since fewer skills are trained averaged over the number of episodes). We demonstrate our results on two challenging continuous control tasks, and we conclude with an analysis of the approach's limitations and potential future work.

1 Introduction

In reinforcement learning, agents learn policies to maximize total expected rewards. Intuitively defined rewards can often be sparse – a robot might receive a +1 reward for picking up the correct object, but this reward can only be obtained in a small part of a very large state space. To facilitate reinforcement learning, we can define shaped rewards. For example, the robot might receive progressively increasing rewards as it moves closer to the correct object, and then another set of rewards for opening its gripper next to the object. However, shaped rewards require careful human engineering [8][4].

Unsupervised skill learning addresses the question of what agents can learn with few or no specified rewards. One challenge in unsupervised skill learning is determining the number of skills that the agent should learn. In many approaches, this number is pre-specified based on experimentation or prior knowledge. In our study, we explore iteratively increasing the number of skills during training to find reasonable stopping points. Additionally, we show that iterative learning trains faster, since fewer skills are learned on average each episode.

Specifically, we extend an approach on unsupervised skill learning called Diversity is All You Need (DIAYN) [6]. We modify DIAYN to gradually learn more skills over time, rather than require a pre-specified number of skills. We propose multiple methods for determining when new skills should be learned, and we empirically compare these methods on two environments.

2 Related Work

Our work builds on existing methods for unsupervised skill learning, particularly DIAYN [6]. DIAYN is motivated by *diversity*, the idea that different useful policies should reach diverse parts of the state space. Specifically, DIAYN maximizes an information-theoretic objective with a maximum entropy policy.

Prior to DIAYN, diversity maximization was used in neuroevolution and evolutionary algorithms to learn complex behaviors [10][16][14]. Rather than focusing on complexity, DIAYN attempts to improve efficiency of skill acquirement. DIAYN was also motivated by work in intrinsic motivation

[13][1][7][12]. While these previous methods studied learning a single policy, DIAYN's objective is capable of learning multiple policies at once.

DIAYN has also inspired more recent works on unsupervised skill discovery. [11] follows DIAYN's approach of using a categorical uniform prior over the latent skills, but leverages contrastive techniques for maximizing mutual information. [9] also defines skills as latent-conditioned policies and introduces a dynamics-simplifying linearizer. [3] optimizes the same information-theoretic objective using different optimization techniques.

To the best of our knowledge, there is no other work on attempting to iteratively learn skills in an unsupervised manner. We explore our method in the context of the diversity objective, but our approach may also be applied to other unsupervised skill learning methods.

3 Preliminaries

We consider the reinforcement learning context of a Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$ with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, and rewards $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. In reinforcement learning, the agent receives rewards through interactions with the environment, and the objective is to learn a policy that maximizes $\mathbb{E}(\sum_{t=1}^{T} r_t)$. In our setting of unsupervised skill learning, we modify the environment so that our agent never receives rewards, and our goal is to learn a diverse set of policies which may be applied towards future downstream tasks. Note that the original environment reward can be seen as a task, and we expect some of our unsupervised policies to obtain high reward when executed in the original environment.

DIAYN introduces additional notation for their information-theoretic approach. S and A represent random variables for states and actions, and $Z \sim p(z)$ is a latent variable on which the policy is conditioned. Policies are conditioned on a fixed Z, and a latent-conditioned policy is called a "skill". Mutual information and Shannon entropy are notated as $I(\cdot;\cdot)$ and $\mathcal{H}[\cdot]$.

Then, DIAYN constructs encodes an objective following three core ideas: (1) Skills should control which states the agent visit by maximizing the mutual information between skills and states $I(S;Z)$; (2) States (not actions) are used to distinguish skills by minimizing the mutual information between skills and actions given the state $I(A;Z|S)$; and (3) Viewing all skills as a mixture of policies, the mixture policy should have maximum entropy $H[A|S]$. We encourage the reader to refer to the original paper [6] for more intuition on these three terms of the objective.

Combining the three terms, DIAYN maximizes:

$$\begin{aligned}\mathcal{F}(\theta) &\triangleq I(S;Z) + \mathcal{H}[A|S] - I(A;Z|S) \\ &= (\mathcal{H}[Z] - \mathcal{H}[Z|S]) + \mathcal{H}[A|S] - (\mathcal{H}[A|S] - \mathcal{H}[A|S,Z]) \\ &= \mathcal{H}[Z] - \mathcal{H}[Z|S] + \mathcal{H}[A|S,Z]\end{aligned}$$

DIAYN then fixes $p(z)$ to be uniform to guarantee maximizing the first term. The second term is approximated using a learned discriminator $q_\phi(z|s)$ to obtain:

$$\begin{aligned}\mathcal{F}(\theta) &= \mathcal{H}[A|S,Z] + \mathbb{E}_{z\sim p(z), s\sim\pi(z)}[\log p(z|s)] - \mathbb{E}_{z\sim p(z)}[\log p(z)] \\ &\geq \mathcal{H}[A|S,Z] + \mathbb{E}_{z\sim p(z), s\sim\pi(z)}[\log q_\phi(z|s) - \log p(z)]\end{aligned}$$

DIAYN is implemented with soft actor critic, which maximizes the policy's entropy over actions and thus maximizes $\mathcal{H}[A|S,Z]$. To maximize the expectation term, DIAYN replaces the task reward with a pseudo-reward:

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

Our work builds on DIAYN's notations and intuitions, and we refer the reader to [6] for further details on implementation.

4 Approach

DIAYN requires the user to pre-specify the number of skills n, which is the dimension of the categorical latent variable Z. However, DIAYN provides no intuition towards how the user might specify n, and a hyperparameter search over n can be costly. Namely, overestimating n incurs

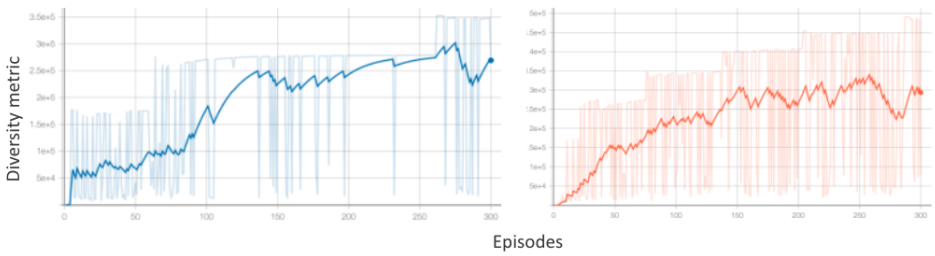

Figure 1: The Diverse1 metric (left) and Diverse2 metric (right) during training of iterative unsupervised skill learning on BipedalWalker.

exponentially larger training costs, while underestimating n throws away DIAYN's performance guarantees and may lead to severely poor performance.

To address these challenges, we modify DIAYN to incrementally increase the number of skills trained based on various metrics. Our method has two advantages: 1) the metrics provide insight towards when the user should stop training and thus the choice of n, and 2) training is faster since fewer skills are trained on average over the episodes.

We refer to the original DIAYN algorithm where the number of skills is held constant as the *baseline* approach. In the following subsections, we present four original approaches for iterative skill learning.

4.1 Naive

We first present *Naive* as the simplest version of iterative skill learning, where the number of skills is incremented by linearly. Specifically, every $j = 20$ rounds the number of skills is incremented by k. This is used primarily as a proof-of-concept to demonstrate the potential benefits of iterative skill learning. However, if it works, its simplicity is appealing in that there are relatively few things to tune with the *Naive* algorithm when applying it to different environments. One limitation of the *Naive* approach is that it doesn't consist of a natural method for determining when to stop adding skills on its own.

4.2 EnvInformed

An alternative algorithm we propose is *EnvInformed*, which uses environmental rewards as cues for when to increment the number of skills to be learned. Specifically, under the *EnvInformed* paradigm, the number of skills n is incremented by k whenever a new maximum environmental reward is set.

Crucially, the maximum environmental reward is different from the diversity reward function that DIAYN uses for training. The intuition behind this approach is to use environmental cues as an oracle for understanding when the diversity approach has trained enough on a current set of skills to warrant adding another one. In the course of training, the maximum environmental reward will naturally converge and as such serves as a natural stopping point.

Note that *EnvInformed*, since it accesses environmental rewards, branches off from the fundamental principles of DIAYN which states that diversity metrics are the only things needed during training. In the following two subsections, we propose two approaches that stay in the DIAYN framework of not accessing environmental rewards and only training based on diversity metrics.

4.3 Diverse1

Our first diversity-based iterative skill learning approach, *Diverse1*, uses the diversity reward included in the original DIAYN paper:

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

Diverse1 uses the diversity reward function $r_z(s, a)$ in a similar manner to *EnvInformed*. During each training episode, *Diverse1* sums up the total r_z from each step (up to 1000 steps per episode). This cumulative sum is referred to as the Diverse1 metric. *Diverse1* keeps a moving average of the

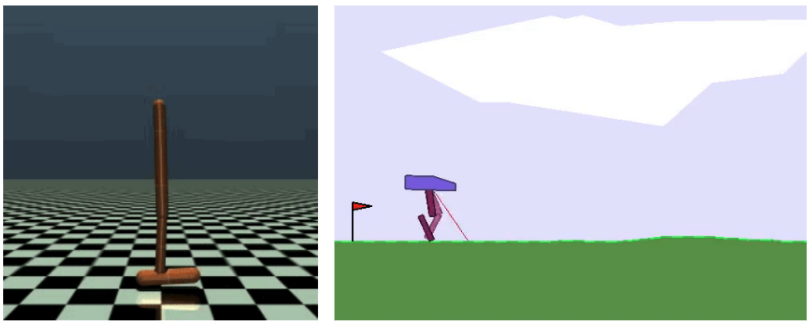

Figure 2: Hopper (left) and Bipedal Walker (right)

Diverse1 metric over the last w rounds. The number of skills is incremented by k whenever the Diverse1 metric increases by greater than or equal to $diverse1\_thresh$ (in our experiments, we set this to be 70%).

4.4 Diverse2

In the *Diverse2* approach, we try to measure the diversity of our current skills based on the difference in the actions they produce. Note that this is different from the DIAYN notion of diversity, which is based on the states visited by the skills.

More specifically, we obtain our Diverse2 metric by comparing the action generated by the most recently learned skill with actions generated by all other skills. Our metric is the sum of the norm of the differences between each pair of actions, divided by n - 1 (the number of pairs of actions). When the number of skills is held constant, we empirically observe that the Diverse2 metric decreases in an exponential shape, experiencing a steeper decline before gradually leveling out. We reason that policies must converge in their actions to achieve diverse parts of their state space. For example, a human randomly moving their joints may end up flat on the ground. Walking, running, and jumping may achieve different states but require similar joint movements. As the number of skills is increased, we observe that the Diverse2 metric increases before leveling off; intuitively, permitting a larger number of diverse skills to be learned increases the potential difference in actions until the skills start overlapping each other.

Therefore, we increment the number of skills when the rate of decrease of the Diverse2 metric significantly slows, which should reflect that the current set of skills have been sufficiently trained. The metric also helps determine a good stopping point for adding new skills – when the Diverse2 metric stops increasing as new skills are added, we know that we have learned enough skills for the environment.

4.5 Decaying increment

Our final contribution in our iterative skill learning approaches is the introduction of a decay in our increment mechanism. As mentioned above, each of our approaches decides to increment the number of skills to be learned by k every time a certain criteria is met. Here, we add another stipulation that incrementing can only occur if at least R rounds have passed since the last increment. Moreover, R is multiplied by a constant factor after every increment to induce a decay:

$$R' = R * R\_mult$$

This is used for all of our approaches (*Naive*, *EnvInformed*, *Diverse1*, and *Diverse2*). The reasoning behind decaying increment is bipartite – (1) it takes more rounds to train the growing the number of skills with each increment and (2) this leads to a natural convergence in training. We set $R = 20$ and $R = 1.3$ in all our experiments (we arrive at these numbers from some simple hyperparameter tuning steps described below).

5 Experiments

We evaluate our proposed iterative skill learning approaches using similar experiments employed by the original DIAYN paper [6]. First, we train their original algorithm to establish a baseline comparison for our approaches. Then, we conduct a set of controlled experiments to evaluate how iterative skill learning affects training performance. In particular, we not only analyze terminal max rewards achieved by each approach but also training convergence and similarity metrics between different skill policies. We encourage readers to view code and GIFs of our experiments[1].

5.1 Environments

We perform experiments using two challenging continuous control tasks: *Hopper-v3* and *BipedalWalker-v3*. These are based on the OpenAI Gym and MuJoCo engines [2, 15]. *Hopper-v3* consists of a two-dimensional one-legged robot where the reward function is tied to forward progress [5]. *BipedalWalker-v3* extends this to a robot with a hull, two legs, and hip and leg joints. The bipedal robot has access to 10 LIDAR rangefinder measurements to evaluate the terrain in front of it (reaching the end accumulates to 300+ in reward whilst falling decreases the reward by 100).

We selected these environments based on their more complex characteristics that are amenable for evaluating skill learning. Compared to other environments like *MountainCar* or *LunarLander*, the skills necessary to achieve high rewards in *Hopper-v3* and *BipedalWalker-v3* are less obvious and more numerous. For example, whereas learning to control a car or lander has directional action inputs that translate to a set of relatively straightforward skills, there are many successful methods to move forward in *BipedalWalker-v3* that utilize completely different skills. Consequently, there are a greater number of possible combinations of skills a successfully trained agent may learn.

In these situations, it is difficult to determine how many skills to learn prior to training. Hence, we deliberately select these environments to study the efficacy of our suite of iterative skill learning approaches.

5.2 Methods

A major objective of our work is to propose a method for diversity-based learning that (1) doesn't require large amounts of tuning based on pre-determining the number of skills to learn and (2) is more efficient in terms of faster convergence when compared to the original DIAYN paper (which achieves the highest rewards when the number of pre-determined skills is set very high and effectively brute forces finds a skill that performs well).

Working towards these goals, we set a constant number of training episodes to be 300 for all approaches (each run took several hours to complete). We then conducted a few runs of each approach on a few validation environments. This mainly served as a sanity check for our methods and resulted in some brief hyperparameter tuning. However, we actually discovered that changing the parameters in our approaches didn't lead to significantly different results (you can see one example plot in the appendix). Our experimental results come from 3 randomized trials of each approach in both environments, where each approache used the same three random seeds.

The lack of a need for lengthy hyperparameter tuning is one component of our work. We set the starting number of skills $n_0 = 2$, skill increment $k = 1$, $R = 20$, and $R\_mult = 1.3$ for all experiments across both environments.

6 Results

6.1 *Hopper-v3*

We compare our approaches based on number of skills learned, the pseudo-rewards obtained by the mixture policy, and the environment rewards obtained by the best skills. As shown in Figure 3, each of the iterative approaches learns much fewer skills (up to 7) compared to the baseline approach which learns 15 skills. *Diverse2* increments skills the most slowly, *Diverse1* and *Naive* both increment skills

[1] https://github.com/eric-z-lin/cs282-iterative-diverse-RL

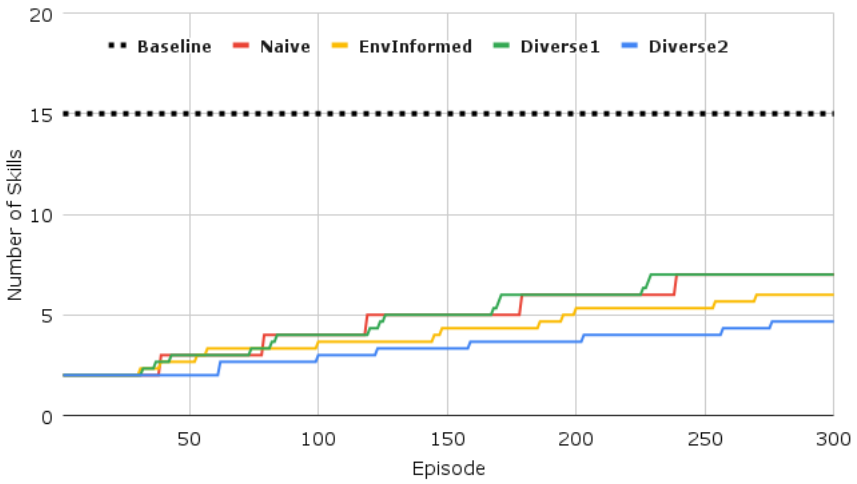

Figure 3: The number of skills learned by different approaches on the Hopper environment.

quickly, and *Envinformed* increments at an in-between pace. While *Naive* regularly increments the number of skills, all the other approaches increment skills more slowly over time.

A direct result of fewer skills learned is that each skill is trained more on average. Since DIAYN uses a uniform distribution $p(z)$ over the set of skills, this means that each of the skills in our proposed approaches were selected for training at least twice as many times on average than baseline DIAYN.

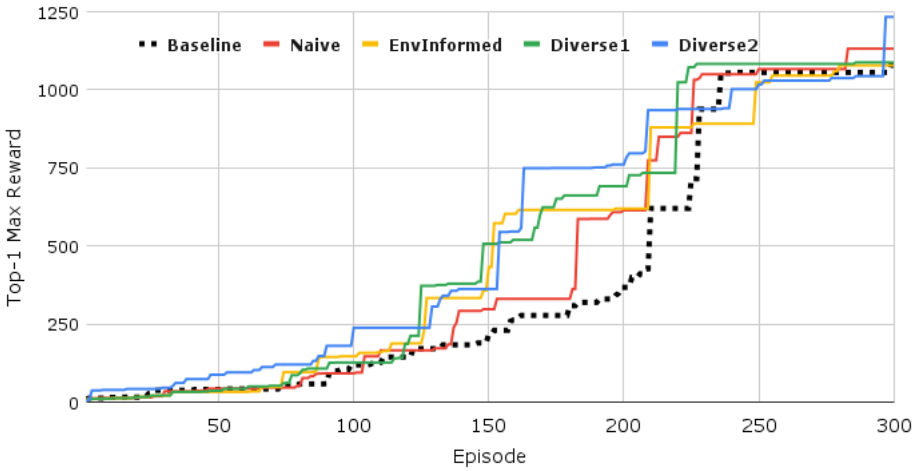

Figure 4: The environment rewards achieved by the best skill in the currently learned set of skills, shown for various approaches on Hopper.

Figure 4 shows the environment reward obtained by the best skill in the current set of learned skills. Although the performance of the best skill is not necessarily perfectly correlated with our true goal of obtaining a small but very diverse set of skills, it gives us an idea of the utility of the learned skills (at least on one task). We see that, despite learning fewer skills, all iterative approaches learn a best skill that performs better than the Baseline method's best skill. *Diverse1* and *Diverse2* seem to perform especially well, each learning best policies obtaining higher rewards at a lower number of training episodes.

Figure 5 shows the pseudo-reward obtained by the mixture policy (i.e. the training policy which switches skills every episode). Recall that the pseudo-reward is part of DIAYN's objective and should reflect the diversity of the learned skills. Here we see that the *Baseline* method achieves the lowest

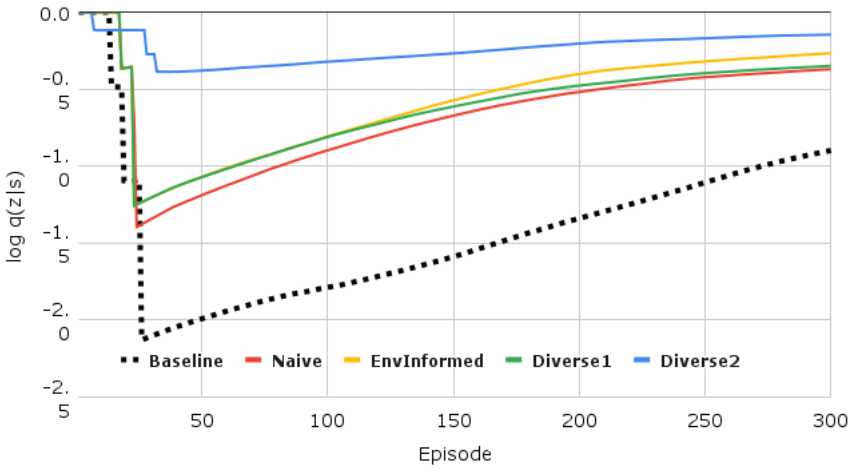

Figure 5: The DIAYN pseudo-reward obtained by different approaches on *Hopper-v3*.

pseudo-reward, and all iterative approaches have higher pseudo-reward. Naive, *EnvInformed*, and *Diverse1* perform comparatively, while *Diverse2* performs significantly better than the rest.

Approach	Top-1 Max Reward	Top-3 Max Reward	Avg Max Reward
Baseline	1062.7	877.3	422.8
Naive	<u>1113.3</u>	**949.0**	653.5
EnvInformed	1063.3	903.8	<u>725.8</u>
Diverse1	1063.3	<u>941.8</u>	722.4
Diverse2	**1200.7**	886.7	**771.8**

Table 1: Max Reward statistics averaged over three 300-episode runs of each approach on *Hopper-v3*. Top-t max reward refers to the average maximum reward for the t best performing skills. **Bolded** numbers represent best performance and <u>underlined</u> numbers second-best.

Table 1 shows the aggregated statistics for maximum rewards at the end of 300 episodes of traiing (averaged over the same 3 randomly seeded runs for each approach). Although achieving the best max reward was not the main objective of our work, we see that *Diverse2* performed particularly well with a significantly higher Top-1 and average maximum reward compared to other approaches. The Top-1 maximum reward was fairly similar across *Baseline*, *EnvInformed*, and *Diverse1*, but the average maximum reward for *Baseline* was significantly lower than all other approaches. This indicates that many skills learned in the *Baseline* approach resulted in poor environmental rewards.

6.2 *BipedalWalker-v3*

Next, we ran the same set of approaches and experimental settings on the *BipedalWalker-v3* environment.

In Figure 6, we see that the approaches follow the same trends as seen in the *Hopper-v3* experiments for the diversity pseudo-reward. It is interesting to note that even though all of our (non-*baseline*) approaches started at $n = 2$ skills, the *Diverse2* approach performed far better than all other approaches by round 30. This is contrasted with *Naive*, which initially performs worse than *Baseline*.

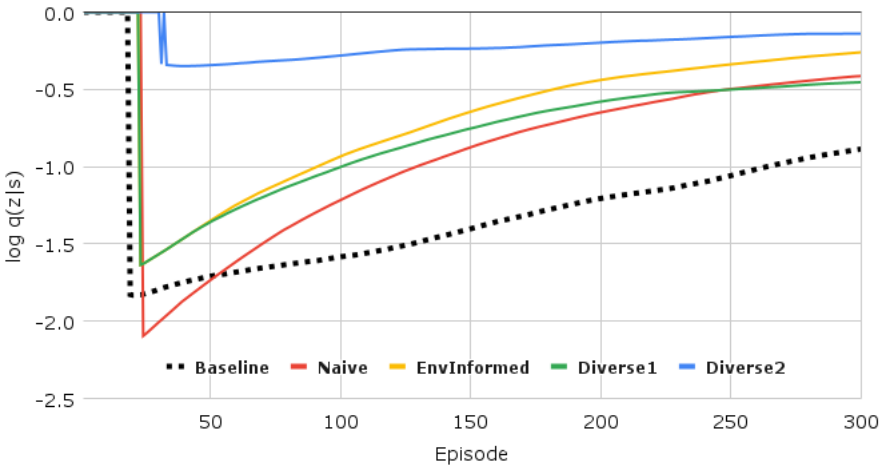

Figure 6: DIAYN pseudo-reward obtained by different approaches on *BipedalWalker-v3*.

Approach	Top-1 Max Reward	Top-3 Max Reward	Avg Max Reward
Baseline	**-1.9**	**-15.1**	-47.7
Naive	-12.7	-25.8	-39.8
EnvInformed	<u>-10.9</u>	<u>-20.4</u>	<u>-39.0</u>
Diverse1	-20.2	-27.7	**-36.4**
Diverse2	-13.7	-32.9	-39.5

Table 2: Max Reward statistics averaged over three 300-episode runs of each approach on *BipedalWalker-v3*. Top-t max reward refers to the average maximum reward for the t best performing skills. **Bolded** numbers represent best performance and <u>underlined</u> numbers second-best.

In Table 2, we see a reversal in the trend when compared to the previous table for *Hopper-v3*. Here, *Baseline* posts better top-1 and top-3 maximum reward statistics than our proposed approaches. This may be due to the higher complexity of *BipedalWalker-v3* (with multiple joints and LIDAR sensors) necessitating approaches that employ number of skills. In our experiments, we capped the number of training episodes at 300 due to computational limits, but more difficult environments like *BipedalWalker-v3* may benefit from many more rounds of training (where our approaches would also add more skills over time).

7 Discussion

From our experiments, we observe that iterative unsupervised skill learning approach can learn more diverse sets of skills in fewer episodes than the baseline non-iterative approach in DIAYN. Although the iterative approach learns a smaller set of skills, the best policy within that set still achieves better performance on the original environment's task. In addition, our Diverse1 and Diverse2 metrics help visualize appropriate stopping points for training and thus automatically detect a good number of skills to learn for the environment.

Some of our results were fairly surprising. Diverse1 and Diverse2 learned slightly different numbers of skills, but they performed similarly in terms of the environment reward achieved by the best policy. In contrast, Diverse1 and Naive learned a similar number of skills, but Diverse1 performed better than Naive in terms of the max environment reward. This seems to imply that not only is the number of skills learned important, but also the timing of when new skills are added. In addition, recall that the Diverse1 metric measures diversity based on the states visited by the skills, while the Diverse2 metric measures diversity based on the actions sampled from the skills. We initially expected Diverse1 to perform better than Diverse2 since the Diverse1 metric is more aligned with the original DIAYN

objective. We are uncertain why Diverse2 appears to outperform Diverse1, particularly when looking at the significantly better performance measured by the DIAYN-pseudo-reward.

It is clear that starting the learning process with a significantly lower number of skills can accelerate convergence. In some environments, perhaps easier ones such as *Hopper-v3*, a lower number of skills may translate to higher environmental rewards as well. However, more investigation is required to understand how to optimize iterative skill learning for more difficult environments without spending too much time on hyperparameter tuning.

8 Conclusion

In this work, we introduced iterative unsupervised skill learning, in which new skills are gradually discovered when the current skills have been sufficiently trained. We found that on various metrics in the MuJoCo environment *Hopper-v3*, our iterative approach outperforms the baseline approach of attempting to learn all new skills simultaneously. Our results on the OpenAI Gym environment *BipedalWalker-v3* were not as conclusive due to limited compute, but still showed that the iterative approach is promising. Overall, iterative learning for unsupervised skill discovery can achieve higher utility skills with more efficient training.

8.1 Limitations

Our method shares limitations with DIAYN. Although DIAYN empirically performs well and inspired other work using the same information-theoretic objective, it is important to note that their objective was built on intuitions rather than concrete theory. It is not clear to us whether there is a better objective to optimize in unsupervised skill learning. Notably, the idea that diversity of policies should be based on states rather than actions is not necessarily intuitive, and in our work we found that measuring diversity using actions was useful towards skill incrementation.

Our method also requires an extra hyperparameter for the Diverse1 and Diverse2 approaches, which is the threshold of change at which we determine the metric has stabilized and a new skill can be learned. We found in general that performance was fairly robust to this hyperparameter and performed minimal tuning. However, we think that tuning this hyperparameter (which can be done within a small number of episodes to check for soundness) is much easier than tuning the number of skills.

8.2 Future Work

In the future, with more compute and time, we would like to run our BipedalWalker experiments for more episodes for a more fair comparison between the iterative approach and baseline. We would also be interested in investigating the theoretical fundamentals of unsupervised skill learning, as we believe there might be provable differences in the iterative and baseline approaches. Finally, we would be interested in further investigating why Diverse2 empirically outperformed Diverse1.

References

[1] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in Neural Information Processing Systems*, 2016.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giro-I-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. *International Conference on Machine Learning*, 2020.

[4] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Inverse reward design. *International Conference on Machine Learning*, 2019.

[5] Tom Erez, Yuval Tassa, and Emanuel Todorov. Infinite horizon model predictive control for nonlinear periodic tasks. *Manuscript under review*, 4, 2011.

[6] Benjamin Eysenbach, Abhishek Gupta Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations (ICLR)*, 2019.

[7] Justin Fu, John Co-Reyes, and Sergey Levine. Ex2: Exploration with exemplar models for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 2017.

[8] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Challenges of real-world reinforcement learning. *Advances in Neural Information Processing Systems*, 2017.

[9] Jaekyeom Kim, Seohong Park, and Gunhee Kim. Unsupervised skill-discovery with bottleneck option learning. *International Conference on Machine Learning*, 2021.

[10] Joel Lehman and Kenneth Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 2011.

[11] Juan Jose Nieto, Roger Creus, and Xavier Giro-i Nieto. Unsupervised skill-discovery and skill-learning in minecraft. *International Conference on Machine Learning*, 2021.

[12] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *International Conference on Machine Learning*, 2017.

[13] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 2000.

[14] Kenneth Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 2002.

[15] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[16] Brian G Woolley and Kenneth Stanley. On the deleterious effects of a priori objectives on evolution and representation. *Conference on Genetic and evolutionary computation*, 2011.

A Project details

Eric and Catherine worked together on all parts of the project, including brainstorming, environment setup, code implementation, experiments, and paper write-up. For the slightly more complex metrics, Eric focused on EnvInformed and Diverse1, and Catherine focused on Diverse2. Eric compiled experiment results and generated plots and tables, and Catherine wrote more of the introduction and related work.

Catherine's main research focus has been in robotics and Eric's in federated learning. Both of us are interested in exploring reinforcement learning and diversity, which led to this collaborative effort. However, this project does not have a strong overlap with either of our ongoing research works.

This project was Eric's first exposure to implementing single-agent reinforcement learning, and we both learned a lot about implementing state-of-the-art reinforcement learning algorithms. We were also both fairly new to information-theoretic objectives, and it was interesting to not only reason about DIAYN's objective, but also work on its implementation.

Thank you so much for teaching us this semester – we learned a lot, and it has been inspirational for our future research endeavors!

B Additional Results

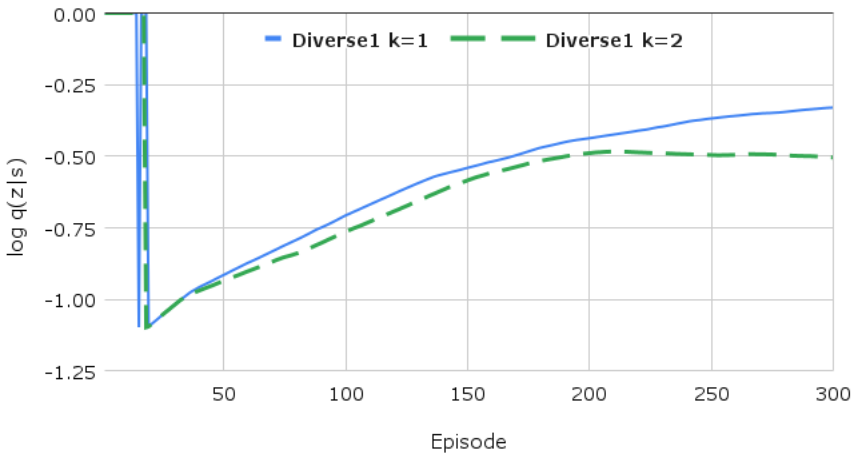

Figure 7: DIAYN pseudo-reward for *Diverse1* approach using different parameters for increment ($k = 1, 2$).

As shown in Figure 7, tuning parameters like the number of skills to increment each time resulted in relatively minor changes in performance.