Curriculum Reinforcement Learning for Hard Exploration Tasks

Anonymous NIPS Submission
Paper ID 251

Abstract

Learning goal-oriented behavior in environments with sparse rewards is a notorious challenge in reinforcement learning (RL). Most of existing RL algorithms with simple heuristics for exploration can not learn useful knowledge to achieve the desired goal when no reward is obtained, and thus suffer from poor sample efficiency. This work proposes Curriculum Reinforcement Learning (CRL), a new exploration diagram for RL which effectively drives the agent to learn stagelized and hierarchical sub-goals to achieve the final goal. The goal selection mechanism developed in CRL is automatically scheduled by a curriculum derived from the past states in the experience replay using information history. In addition, CRL can be deployed into any off-policy RL algorithms. Extensive experimental results have verified its effectiveness and efficiency. The source code will be released.

1 Introduction

A fundamental challenge in reinforcement learning (RL) is how to trade-off between exploration and exploitation. The agent has to decide whether to greedily exploit what it already knows in order to maximize expected cumulative reward, or explore the unknown environment to gather more information and find a potentially better policy. Though simple exploration strategies such as ϵ-greedy action selection Mnih *et al.* [2015] or Gaussian control noise Mnih *et al.* [2016] works well on a wide range of tasks, they are inefficient in hard-exploration tasks with sparse-rewards such as the Atari games *Montezuma's Revenge*, *Gravitar*, *Private Eye*, *Freeway*, and *Venture*Bellemare *et al.* [2016]. For these hard-exploration tasks, standard RL algorithms perform poorly without even finding a single positive reward.

For reinforcement learning the hard exploration task, many previous works Aytar *et al.* [2018]; Pohlen *et al.* [2018] seek to imitation learning with expert demonstrations. However, replays from expert demonstration are often not available in practical applications. In this case, curiosity based methods Bellemare *et al.* [2016]; Pathak *et al.* [2017] introduce some notions of curiosity or uncertainty as an exploration bonus to guide the learning process. However, such exploration methods still struggle with hard exploration games such as *Montezuma's Revenge*. In another way, curriculum learning Bengio *et al.* [2009a] is a promising way to help the agent learn gradually from easy to hard. However, these curricula are usually hand-designed and require domain-specific knowledge. The recently proposed Hindsight experience replay (HER) Andrychowicz *et al.* [2017] method can be seen as one kind of the goal-conditioned curriculum learning. HER tackles the sparse-reward problem by learning from virtual goals, which are sampled randomly from the experience replay. The goals are selected randomly without considering the importance of the samples in the experience replay.

To deal with the above problems, we argue that experience replay should not be treated equally and goals should be dynamically adjusted and set at the appropriate level of difficulty for the current policy. To carry out the above argument, one thing the agent must know is how to select appropriate goals from the experience replay, and another thing is how to learn to achieve the goals efficiently. Based on these considerations, we introduce the curriculum learning into reinforcement learning and

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

propose a new learning framework, referred to as Curriculum Reinforcement Learning (CRL) for hard exploration tasks. The proposed CRL framework is inspired by the way human self-taught curricula. Through CRL framework, agents can automatically generate learning goals based on existing models and random exploration. To summarize, the contributions of this paper is threefold:

- We introduce the curriculum learning into the reinforcement learning problem and propose the curriculum reinforcement learning framework to deal with hard exploration tasks.
- We propose an information entropy based model, which is proved to satisfy the curriculum definition and applicable to a wide range of hard-exploration tasks with sparse rewards.
- We develop two instantiated learning algorithms under the CRL framework, single curriculum reinforcement learning and elective curriculum reinforcement learning, to demonstrate the feasibility and extendability of the framework.

The proposed framework is general and can be combined to any off-policy RL algorithms. In addition, self-imitation learning(SIL) Oh *et al.* [2018] can better help agents achieve their goals in CRL framework. We verify CRL framework on hard exploration tasks like *Montezuma's Revenge* and *Gravitar*. Extensive experimental analyses and comparisons demonstrate the effectiveness of the instantiated learning algorithms. To facilitate further studies on reinforcement learning of hard exploration tasks, the source code, trained models, and all the experimental results of this work will be released upon its publication.

2 Related work

Exploration. Various approaches have been proposed to improve exploration in sparse-reward RL, including count-based exploration Bellemare *et al.* [2016]; Tang *et al.* [2017] and prediction-based exploration Pathak *et al.* [2017]; Savinov *et al.* [2018]; Burda *et al.* [2018]. State visitation counts have been investigated to reduce the agent's uncertainty by visiting states or state-action pairs with low visit-counts. For RL in large domains, a "pseudo count" can be constructed using a density model over the state space Bellemare *et al.* [2016], or cluster occurrence counts with locality-sensitive hashing to cluster states Tang *et al.* [2017]. Another class of exploration methods are based on prediction error for a problem related to the agent's transitions. Directly predicting the next observations or their embedding Pathak *et al.* [2017]; Stadie *et al.* [2015] will be subject to the "noisy TV" problem in stochastic environment or partially observable environment. Recently, some works addresses the noisy TV problem in prediction-based exploration, the notable ones are episodic curiosity through reach ability Savinov *et al.* [2018] and random network distillation Burda *et al.* [2018].

Experience replay. The success of DQN and its variations own much to the usage of experience replay buffer. Prioritized experience replay Schaul *et al.* [2015] improves the sample efficiency by prioritizing past experiences based on temporal-difference error. Actor-critic RL algorithms such as A2C/A3C Mnih *et al.* [2016] are known for their sampling inefficiency. In fact, these algorithms can also utilize experience replay. Many existing methods either require importance sampling Wang *et al.* [2016]; Gruslys *et al.* [2017] or are limited to continuous control Lillicrap *et al.* [2015]. Oh *et al.* [2018] used a replay buffer filled with past good experience and proposed self-imitation-learning (SIL) method to imitate what the agent has experienced but has not yet learned. In this work, we use the actor-critic RL framework with self-imitation learning to accelerate our learning process.

Hierarchical RL. Hierarchical RL is a promising approach to extend traditional RL methods to solve complex tasks. Options framework (Fruit and Lazaric [2017]) is one of the most important hierarchical RL methods. There is a useful set of options which are fully defined as a priori. Planning and learning only occurs at the higher level. Hierarchical Imitation and Reinforcement Learning (Le *et al.* [2018]) does not have access to the policy of sub-goals and must learn them via expert or reinforcement feedback. Kulkarni *et al.* [2016] uses a similar hierarchical structure, but has no high-level expert and hierarchical guidance. Unlike the traditional hierarchical approach, we simplify the problem into sub-tasks by accessing the internal rewards, but not with policy.

Curriculum learning. Curriculum learning Bengio *et al.* [2009a] have been explored in supervised learning to split complex tasks into smaller, easier-to-solve sub-problems. Intensive research has investigated curriculum-based approaches in robotics tasks Forestier and Oudeyer [2016] and reinforcement learning Florensa *et al.* [2017]; Graves *et al.* [2017]. Most practical curriculum approaches in RL require pre-specified task sequences. Some general frameworks have been proposed to generate increasingly hard problems Schmidhuber [2013], while the idea still struggles to solve the complex robotics tasks. Recently more research interests focus on the automatic curriculum learning Graves *et al.* [2017] or reverse curriculum generation Florensa *et al.* [2017] in RL. On the other side, some methods such as hindsight experience replay Andrychowicz *et al.* [2017] can also be regarded as one kind of curriculum learning.

3 Curriculum reinforcement learning

3.1 Framework formulation

We consider the standard reinforcement learning formalism consisting of an agent interacting with an environment to fulfill some tasks. To simplify the exposition we assume that the environment is fully observable. An environment is described by a set of states S, a set of actions A, and a reward function $r : S \times A \rightarrow R$. The input of the problem is a sequence of task states $\mathcal{S} = \{S_1, S_2, \ldots, S_t, \ldots\}$, The agent needs choose from a set of predefined actions $\mathcal{A} = \{a_k\}_{k=1}^{K}$ to get through different task levels, where K is the number of predefined actions. Different games may have different actions. After executing an action a_k at each task state S_t, the agent will get a reward $r(S'|S_t, a_k)$, and the objective of the agent is to maximize the cumulative task rewards $R = \sum_{t=1}^{\infty} \lambda^{t-1} r(S_{t+1}|S_t, a_k)$, where λ is the discount factor to favor more recent rewards.

For hard exploration tasks with sparse rewards, *i.e.*, $\{r(S'|S_t, a_k)\}$ are only defined at a few of states to express the task goal and set to 0 at all the other states. Standard RL algorithms like DQN Mnih *et al.* [2015], A3C Mnih *et al.* [2016], A2C Dhariwal *et al.* [2017], and PPO Schulman *et al.* [2017] perform poorly under this setting. The proposed approach is thus motivated to improve the learning efficiency over existing algorithm with sparse rewards.

3.2 Entropy based curriculum modeling

Curriculum is a corresponding sequence of distributions Q_λ in which the entropy of the distributions $H(Q_\lambda)$ and the weight applied to example z at step λ is incremental Bengio *et al.* [2009b].

We define the random exploration of the game as follows: When the agent does some turns experiment, some states are obtained in experience replay buffer. Sorting the magnitude of the frequency of occurrence of different states and performing statistical statistics on them, we get the following states as S = $(s_0, s_1, ...s_k)$. For the sparse reward problem, the agent should explore the environment more when there is not any reward obtained. So it is necessary for a model to be able to achieve all the states S for the current agent. We define z as a random example $(s_t, a_k, 0, s_{t+1})$ in experience buffer, $W_\lambda(z)$ as the weight applied to example z at step $0 \leq \lambda \leq 1$, also λ can represent the degree of completion of learning. The target training distribution is $P(z)$ and the corresponding training distribution at step λ is Q_λ, which is proportional to $W_\lambda(z)P(z)$. For a RL algorithm, when we use some experience Z to train model at step λ for the step $\lambda + \epsilon$, it is obvious that:

$$W_{\lambda+\epsilon}(z) \geq W_\lambda(z) \qquad \forall\epsilon, \forall z > 0, \tag{1}$$

$$H(Q_\lambda) < H(Q_{\lambda+\epsilon}) \qquad \forall\epsilon > 0. \tag{2}$$

For further verification, we define the random exploration of the game as follows: When the agent does n turns, m steps experiment, $n * m$ states are obtained in experience replay buffer. Sorting the magnitude of the frequency of occurrence of different states and performing statistical statistics on them, we get the following states and probabilities as $S = (s_0, s_1, ...s_k)$, $P = (p_0, p_1, ...p_k)$. For different states, it is obvious that:

$$-\sum p_{i\in\lambda} \log p_{i\in\lambda} < -\sum p_{i\in\lambda+\epsilon} \log p_{i\in\lambda+\epsilon}, \tag{3}$$

$$\sum p_{i\in\lambda} < \sum p_{i\in\lambda+\epsilon} \tag{4}$$

in which $p_{i\in\lambda}$ refers to probability of state s_i updated in λ step in one RL algorithm. The above four formulas prove that for hard exploration tasks, when there is no reward obtained, a reinforcement learning model used to help the agent go through all the possible states is a matter of curriculum learning. In other words, it is a learning process from easy to difficult. But for sparse reward problems, it is not necessary to experience some states in some special ways.

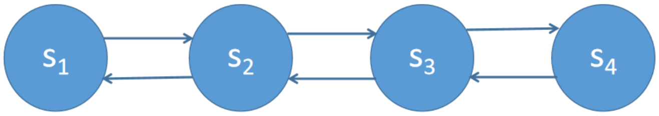

Figure 1: A simple example. S_1, S_2, S_3, and S_4 are four different states of the exploration task.

Here is a simple example as fig.1. When agents need get S_4 to get the only reward in this exploration task. RL algorithms need to learn the following three strategies ($S_1 \rightarrow S_2, S_2 \rightarrow S_3, S_3 \rightarrow S_4$). However, there are two other ways that RL algorithms don't need to learn ($S_3 \rightarrow S_2, S_4 \rightarrow S_3$). In order to solve this problem, we put forward two algorithms: single curriculum reinforcement learning and elective curricula reinforcement learning.

3.3 Goal Definition of Single Curriculum Reinforcement Learning(SCRL)

In order to solve hard exploration tasks, we propose single curriculum reinforcement learning algorithm. RL algorithms should not learn strategies that have little value in exploring the states of the environment. Following the process of human learning, we sample previously experienced but unlearned states to form a state replay pool, sorting the probability of the occurrence of states in the state replay pool and select a state as the target. According to Law of Large Numbers, the probability of the occurrence of the states in state replay pool can represent the probability of the occurrence of the states. At the same time, these probabilities can also represent the difficulty of occurrence for the existing model. When the agent gets the goal state, the environment gives the agent an additional reward. In this way, the agent can effectively reduce the learning of unnecessary strategies, and then help the agent to explore the environment more quickly. See Alg.1 for a more formal description of the algorithm.

Algorithm 1: Single Curriculum Defintion

```
Input: Existing RL model $\mathcal{M}$, Old Replay buffer $R_{OldState}$
Output: Single Curriculum goal $S_{curriculum}$
Initialize New Replay buffer $R_{NewState}$
while Turn$< N$ do
    while step$< M$ do
        Execute an action $s_{t,i}, a_{t,i}, r_{t,i}, s_{t,i+1} \sim \pi_{\mathcal{M}}(a_{t,i}|s_{t,i})$
        if $s_{t,i+1}$ not in $R_{OldState}$ then
            $R_{NewState} \leftarrow s_{t,i+1}$
        if $r_{t,i}! = 0$ then
            $S_{curriculum} \leftarrow s_{t,i}$
            break
```

3.4 Goal Definition of Elective Curricula Reinforcement Learning(ECRL)

Although SCRL can help agents learn more useful strategies, there are also some complex exploration problems can not be solved through SCRL. Affected by the process of students'self-learn we put forward elective curricula reinforcement learning. The reward sparsity problem is like the process of human growth. Only a few important examinations (eg: the entrance exam for university) can detect how we learn and what kind of schools we will go next. But for human growth, teachers guide us to learn some subjects, help us learn curricula, and arrange various quizzes to check our mastery of knowledge in the process when we are students. However, for the reward sparsity problem, agents have no reward until they get rewards. We analogize the rewarded status for agents to an important test in human learning process.

The purpose of elective curricula reinforcement learning is to help agents form their own curricula, to use their past experience to teach agents and check it regularly. For human beings, we have learned many different curricula from children to adults, such as mathematics, physics, English, biology, computer and so on. Agents should also define different curricula according to what they have learned. In addition, by analogy with the human learning process, these curricula should be equally or nearly equally difficult for current agents. Compared with human curricula, it is very simple for agents to define curricula of the same difficulty at different stages.

Above discusses how agents define elective curricula when there is no reward for the reward sparsity problem. While there are some rounds of experiments in which the agent gets rewards, the definition of elective curricula should not be determined solely by the probability distribution. Agents should be more focused on learning the state rewarded, and define the states rewarded as new elective curricula. See Alg.2 for a more formal description of the algorithm.

Algorithm 2: Elective Curricula Defintion

```
Input: Existing RL model $\mathcal{M}$, Old Replay buffer $R_{OldState}$
Output: Elective Curricula goals $S_{curricula}$
Initialize New Replay buffer $R_{NewState}$
while Turn< $N$ do
    while step< $M$ do
        Execute an action $s_{t,i}, a_{t,i}, r_{t,i}, s_{t,i+1} \sim \pi_{\mathcal{M}}(a_{t,i}|s_{t,i})$
        if $s_{t,i+1}$ not in $R_{OldState}$ then
            $R_{NewState} \leftarrow s_{t,i+1}$
        if $r_{t,i}! = 0$ then
            $S_{curricula} \leftarrow s_{t,i}$
if $S_{curricula}!$ then
    Sort frequency of different states in $R_{NewState}$
    Choose $K$ different states in $R_{NewState}$ with same probability as $S_{curricula}$
```

At the same time, we define several different evaluation functions to evaluate the scores obtained for agents. Once the requirements are met, we can assume agents have learned this set of curricula and help agents continue to learn new curricula. This is the origin of our elective curricula algorithm. With Elective Curricula Theory, the reward sparsity problem can be turned into several basic learning problems, which can be solved through reinforcement learning methods.

3.5 Curriculum Reward Function

For the reward sparsity problem, in most situations the agent cannot get rewards, what the agent should do is get more different states. For an experiment with m steps, we define a new loss function R_o named one-turn entropy reward, a new loss function R_n named n turns entropy reward and a new loss function R_c named curricula reward. And the whole rewards for n-turns m-steps $R_{o,n,c}$ is:

$$R_o = -\sum_{i=0}^{m-1} p_{s_i} \log p_{s_i} \tag{5}$$

$$R_n = -\sum_{t=1}^{n}\sum_{i=0}^{m-1} p_{s_{t,i}} \log p_{s_{t,i}} \tag{6}$$

$$R_c = \sum_{t=1}^{n}\sum_{i=0}^{m-1} \text{Judge}(s_{t,i}, S_{curricula}) \tag{7}$$

$$R_{o,n,c} = \lambda_1 * R_o + \lambda_2 * R_n + \lambda_3 * R_c \tag{8}$$

$$\lambda_1, \lambda_2, \lambda_3 > 0 \tag{9}$$

p_{s_i} refers to the probability of state s_i in m steps experience replay buffer. $p_{s_{t,i}}$ refers to the probability of state of the tth turn ith step $s_{t,i}$ in n turns m steps experience replay buffer. $S_{curricula}$ refers to the states which agents define as new curricula. If $s_{t,i}$ in $S_{curricula}$, $\text{Judge}(s_{t,i}, S_{curricula}) = 1$; else $\text{Judge}(s_{t,i}, S_{curricula}) = 0$.

R_o aims to help agents to experience more different states to maximize the whole entropy in m steps. R_n aims to help agents to experience more different states to maximize the whole information entropy in n turns ,m steps. R_c is a good indication of whether the agent has learned the required curricula. $\lambda_1, \lambda_2, \lambda_3$ refer to different weights for R_o, R_n, R_c. We use $R_{o,n,c}$ to test whether the agent has mastered the elective curricula they need in n turns, m steps experiment.

3.6 Curriculum Reinforcement Learning Framework

It is well accepted that human learns much better when the training examples are not randomly presented but organized in a meaningful order, gradually illustrating more concepts from easy to hard. When incorporating this principle into machine learning, the curriculum learning strategy Bengio *et al.* [2009b] demonstrates promising results in supervised learning of deep neural networks. By incorporating this curriculum into the reinforcement learning process of environment exploration, we obtain a new learning diagram which can be integrated with any off-policy RL algorithms. We refer to it as *Curriculum Reinforcement Learning*.

According to the definition introduced in Section 3.3 and 3.4, agents define some elective curricula that need to be learned in the next step through existing models and randomized trials. Then agents need better master the elective curricula through reinforcement learning algorithms. Also self imitation learning Oh *et al.* [2018] can be applied to imitate past experience which contain curricula rewards for agents.

Algorithm 3: Curriculum Reinforcement Learning Framework

```
Given: An off-policy RL algorithm $\mathcal{A}$
Initialize Replay buffer $R$
Initialize Elective Curricula goals $S_{curricula}$
Input: RL Model $\mathcal{M}$
Output: RL Model $\mathcal{M}$
while do
    #Generate Curricula
    Form Elective Curricula goals $S_{curricula}$ using the behavioral policy from $\mathcal{A}$ (Alg. 2)
    #Curriculum learning
    Using off-policy RL algorithm $\mathcal{A}$ and self-imitation learning algorithm update model $\mathcal{M}$
    #Evaluate Curricula
    while $R_{o-t,m-s,c} > curriculascore$ do
        Update Replay buffer $R$ using the behavioral policy from $\mathcal{A}$
        Break
    Clear Elective Curricula goals $S_{curricula}$
```

Here we give our Curriculum Reinforcement Learning Framework(CRLF). The learning flow is iteratively three stages. The first stage is to explore the environment by existing models and random walks and generate multi-goal elective curricula. The second stage is to do curriculum learning with an off-policy RL algorithm and self imitation learning algorithm. The third stage is to evaluate Curricula. The curricula is finished if Curricula Reward for n turns, m steps R_c reaches the threshold. The complete algorithm is shown in Alg.3.

Through Curriculum Reinforcement Learning Framework, a sparse reward problem can be turned into many reward contained problems. Curriculum Reinforcement Learning Framework imitates the way of human elective curricula in school. By exploring and imitating the past good experience, agents accelerate the convergence of reinforcement learning algorithm. The main idea on which the algorithm relies is that past experience should not be treated equally. First, agents should first learn the easy-to-reach states and gradually learn backwards. Like humans, learning how to run is unrealistic when we don't know how to walk. Second, using self-imitation learning can speed up agents'training for those 'good states'. This is also a reflect that past experience has not been treated equally.

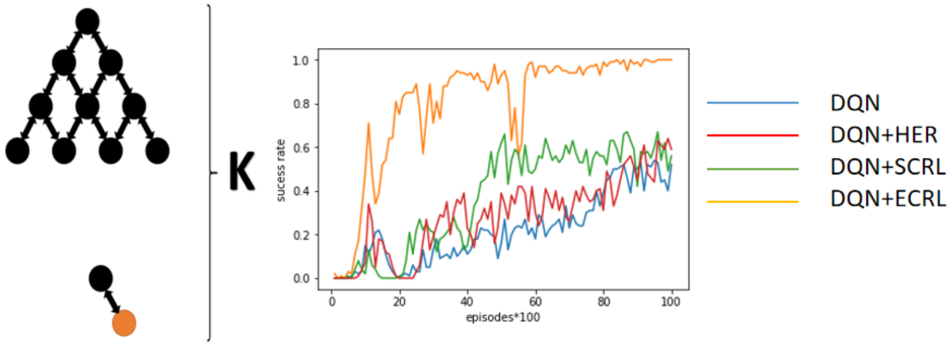

Figure 2: A goal-seeking game and Performance of different methods

4 Experiments

4.1 A goal-seeking game

The experiments are designed to answer the following questions:(1)What is learned by CRLF? (2)Is CRLF better than traditional off-policy RL algorithms? (3)Is ECRL better than SCRL in hard-exploration tasks? We used the following four methods:

- DQN:Deep Q Neural Network.
- DQN+HER:Deep Q Neural Network with Hindsight Experience Replay.
- DQN+SCRL:Deep Q Neural Network with Single Curriculum.
- DQN+ECRL:Deep Q Neural Network with Elective Curricula.

Consider a goal-seeking game in a tree like Fig.2 left. There are $i+1$ different states in the i level. Connectivity between the various levels is shown in Fig.2 left. We define the states in i level from left to right as $((i,0),(i,1),(i,2)......(i,i))$. There are four actions $(0,1,2,3)$ which represents the actions of the up-left, the up-right, the down-left, and down-right. If there is not state in environment with state s action a, the state s will keep the last state. In our experiment, we set the level K as 20 and a seek goal in level 20. Agents need to get the goal within $K-1$ steps. We used the four different methods above and the performance are shown as Fig.2 right. It is clear that DQN+ECRL performs much better than other three methods in all episodes. More importantly, we can also notice that CRLF learns faster because of self-imitation learning and helps reinforcement learning methods explore unknown environment.

4.2 Hard Exploration Atari Games

In order to verify the effectiveness of our method, we apply CRL framework to for some hard exploration games like *Montezuma's Revenge*, *Gravitar*, *Freeway* and so on. Fig.3 shows the overall preview of the first room of Montezuma's Revenge. At the first room, the agent needs to descend the ladder first, then jump over the obstacles, and finally get the key. Of course, in each level, the agent will encounter a new environment, the agent needs to go out of all the rooms in order to get out of the first level. Agents need to avoid many different obstacles. The context information received by the proxy is a bird's-eye view of the environment, including the elective curricula defined by agents (marked as diaphanous proxy). The original action consists of 17 movements, such as up, down, left, right , jump and so on. Because it is too complex to define the state space with *obs* in Atari games, we treat the position of the proxy as a state in the experiment and we use *obs* subtraction here to find the position of the controlled proxy.

We investigate how useful our CRL framework is for several hard exploration Atari games. We compared our approach against RND Burda *et al.* [2018], PPO Schulman *et al.* [2017], SIL Oh *et al.* [2018], A3C Mnih *et al.* [2016] and SOTA. And the result is shown in Table 1.

Experiments show that A2C with CRL framework achieves best results on *Montezuma's Revenge* and *Gravitar*. For hard exploration tasks, curriculum reinforcement learning framework can help agents

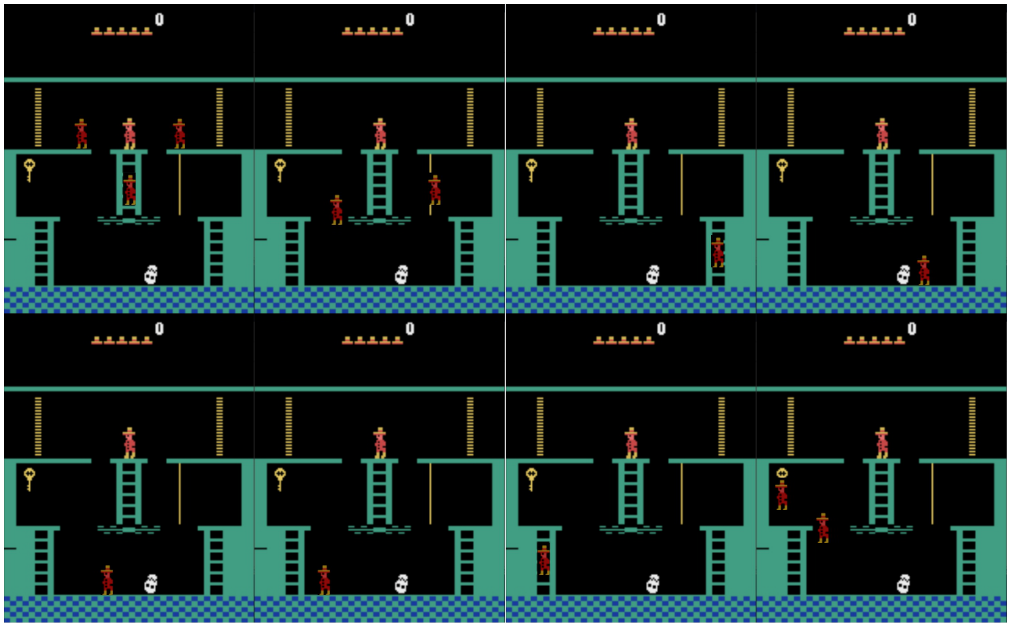

Figure 3: Elective Curricula in Montezuma's Revenge

to form different curricula and explore the environment continuously. In addition, compared with SIL, the experimental results also prove the effectiveness of elective curricula in hard exploration tasks.

Table 1: Score comparison on hard exploration Atari games

	Montezuma's Revenge	Gravitar	Private Eye	Freeway	Venture
A2C+CRL	**24328**	**4164**	10409	**34**	0
RND	8152	3906	8666	N/A	**1859**
PPO	2497	3426	105	32	0
A2C+SIL	2500	2722	8684	**34**	0
A3C	100	239	99	30	0
SOTA	3700	2209	**15806**	**34**	1813

5 Conclusion

In this paper, we proposed curriculum reinforcement learning, which aims to drive the agent to learn stagelized and hierarchical sub-goals to achieve the final goal. The curriculum reinforcement learning framework helps the agent to set up curricula through the experience of random exploration and accelerates the curricula learning through self-imitation learning. Experiments show that curriculum reinforcement learning framework is very suitable for hard exploration tasks.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *Advances in Neural Information Processing Systems*, pages 2935–2945, 2018.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*, 2017.

Sébastien Forestier and Pierre-Yves Oudeyer. Curiosity-driven development of tool use precursors: a computational model. In *38th annual conference of the cognitive science society (cogsci 2016)*, pages 1859–1864, 2016.

Ronan Fruit and Alessandro Lazaric. Exploration-Exploitation in MDPs with Options. In *International Conference on Artificial Intelligence and Statistics*, pages 576–584, 2017.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR. org, 2017.

Audrunas Gruslys, Mohammad Gheshlaghi Azar, Marc G Bellemare, and Remi Munos. The reactor: A sample-efficient actor-critic architecture. *arXiv preprint arXiv:1704.04651*, 2017.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.

Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé, III. Hierarchical imitation and reinforcement learning. In *International Conference on Machine Learning*, pages 2917–2926, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International Conference on Machine Learning (ICML)*, 2018.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

Tobias Pohlen, Bilal Piot, Todd Hester, Mohammad Gheshlaghi Azar, Dan Horgan, David Budden, Gabriel Barth-Maron, Hado van Hasselt, John Quan, Mel Večerík, et al. Observe and look further: Achieving consistent performance on atari. *arXiv preprint arXiv:1805.11593*, 2018.

Nikolay Savinov, Anton Raichuk, Raphaël Marinier, Damien Vincent, Marc Pollefeys, Timothy Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *arXiv preprint arXiv:1810.02274*, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Jürgen Schmidhuber. Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4:313, 2013.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2753–2762, 2017.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.