Work-in-Progress: Multi-Teacher Curriculum Design for Sparse Reward Environments

Chaitanya Kharyal*
IIIT Hyderabad & Microsoft
Hyderabad, Telangana
chaitanyajee@gmail.com

Tanmay K. Sinha*
IIIT Hyderabad
Hyderabad, Telangana
tanmay.kumar@research.iiit.ac.in

Matthew E. Taylor
University of Alberta & Alberta
Machine Intelligence Institute (Amii)
Edmonton, Canada
matthew.e.taylor@ualberta.ca

ABSTRACT

While reinforcement learning agents have had many impressive successes, such agents can often face difficulty in sparse reward environments. Agents often face this difficulty in real-world tasks — it can take a long time before an agent stumbles upon a rare positive outcome without guidance. To combat this problem, we propose an technique that we call Adversarial Multi-Teacher Curriculum Design with Traces. This technique involves multiple independent teachers engaged in a game against a goal-conditioned student. The primary algorithmic novelty, relative to existing work, is engaging multiple teachers and using a behavior cloning loss. In addition, we also introduce a new sparse reward environment for simulated driving in PyBullet. Empirical results show the potential of our algorithm in this novel domain.

KEYWORDS

Reinforcement Learning, Curriculum Learning, Adversarial Learning

1 INTRODUCTION

Many reinforcement learning [16] (RL) algorithms struggle to learn in sparse reward environments because of insufficient feedback from the environment resulting in weaker gradient updates. Many solutions like reshaping the reward function, rewarding the exploratory behaviour based on novelty / information theoretic metrics, learning an intrinsic reward function etc. were introduced but these algorithms still fail in sparse reward environments.

We approach this problem using a curriculum learning [10] approach. As prior works have shown, while it is impractical for humans to set the entire curricula, it is inefficient to use a heuristic-based teacher agent to setup curricula as they do not work on all environments or do not generalize well. Thus, taking inspiration from generative adversarial networks [8] and building on the prior work of adversarially motivated intrinsic goals [13], we train the teacher agent based on its ability to set goals in a curriculum style. We then train a goal conditioned agent to reach these goal states. We introduce the notion of allowing multiple teachers to train simultaneously while generating a curriculum for a single student, and show that this approach can outperform a curriculum generated by a single teacher. Furthermore, we analyse the effect of the behavioural cloning loss that we used on both the teacher and student agents. To evaluate the performance of our method, we introduce a novel driving game that is also resettable, `PyBullet-driving`. In cases where there is a single start state or the start state can be precisely set, such as in `PyBullet-driving`, our results show the potential of our algorithm to learn significantly faster than when learning without a curriculum.

*Both authors have contributed equally

Proc. of the Adaptive and Learning Agents Workshop (ALA 2022), Cruz, Hayes, da Silva, Santos (eds.), May 9-10, 2022, Online, https://ala2022.github.io/. 2022.

2 RELATED WORK

Many algorithms use intrinsic motivation to help reinforcement learning algorithms handle challenging sparse reward environments. There are many interpretations/formulations for intrinsic motivation. Some approaches use state visitation counts as its metric [1] , some define it as a measure of reducing uncertainty of prediction models [4], and others as having unpredictable consequences on the environment or entropy of the goal distribution [14]. Some approaches also learn agents intrinsic rewards based on the rewards from the environment [17].

[11] provided a classification and literature review of several kinds of curriculum methods in reinforcement learning. In curriculum learning [2], the student (agent) learns to perform on increasingly difficult tasks or subgoals. In this work, we want to focus on the case where the teacher agents sets these goals that are neither too difficult nor too easy to achieve. Handcrafted curricula can be time consuming and infeasible. AMIGo [5] introduced a novel way of setting goals that are adversarially motivated — the teacher agent learns to set goals by training in a fashion similar to generator in a typical GAN setting. Our approach differs by 1) requiring the teacher agent to reach the goal itself and 2) enabling the student to learn from an additional signal of behavior cloning loss (over the teacher's trajectories). The closest setting to our work is introduced in [12]. This paper's contributions include:

(1) Introducing simultaneous, multiple teacher training and show that it outperforms existing baselines
(2) Introducing a novel sparse reward driving simulator environment that we believe will be a good environment for future curriculum learning research.
(3) Analyzing the effect of behavior cloning loss on both the teacher and student agents.
(4) Providing empirical results suggesting the promise of this method.

3 ADVERSARIAL MULTI-TEACHER CURRICULUM DESIGN WITH TRACES

This section describes our novel approach and suggests why it can outperform other existing approaches in sparse reward environments. Then, we define the behavioural cloning loss used to train

the student agent. Finally, we formalise Adversarial Multi-Teacher Curriculum Design with Traces in Algorithm 1.

3.1 Method

We train two types of agents — a Teacher and a Student. In each Teacher-Student rollout, we sample a starting state $s_0 \in \mathcal{S}$ and starting from this state, the Teacher, with policy $\pi_T(a \mid s)$, interacts with the environment for a number of time steps to reach the state $g_t \in \mathcal{S}$. Again, starting from the same state s_0, the Student, with goal conditioned policy $\pi_S(a \mid s, g_t)$, interacts with the environment and tries to achieve goal state g_t. The Teacher should aim to generate increasingly difficult goals that form an ideal curriculum for the Student to learn to reach a wide variety of difficult goal states.

There are two reasons why this approach may be helpful to student learning. First, we know by construction that g_t is reachable from starting state s_0. Moreover, a Teacher provides a valid trajectory from s_0 to g_t that the student can use to enhance the Student's learning through behavioural cloning. Second, with a proper reward function as a function of the Student's ability to reach the set goals and validity of the goals, the training of the Teacher will ensure that the goals it sets are incrementally difficult for Student to achieve.

While traditional curriculum learning methods typically have either a fixed curriculum or a single teacher agent, we consider the case with multiple teachers. In each rollout, one of the Teacher agents sets the goal and the Student tries to reach that goal. We repeat this until each Teacher agent has set m goals. Once the rollout data is collected, we update the model parameters for each of the agents.

3.2 Reward Structure

We assign sparse rewards to both the Teacher and Student agents, based on whether the Student is able to reach the goal set by a Teacher. If Student reaches the goal set by a Teacher, that Teacher gets a single reward of -5 and Student gets a single reward of +5. On the other hand, when Student does not reach the goal, the Teacher gets a reward of +5 and the Student gets a reward of 0.

3.3 Behavioural Cloning

For assisting a Student's learning, we use a Behavioural Cloning loss ($\mathcal{L}_{BC}$) for the Student along with a standard TD loss (e.g., the TD3 actor loss).

$$\mathcal{L}_{BC} = \mathbb{E}_{(s_t, g_t) \in \{\texttt{Student's Mini-Batch}\}} \left[\| \pi_B(a \mid s_t, g_t) - \pi_A(a \mid s_t) \|^2 \right]$$

3.4 Algorithm

We denote the n Teacher agents as A_1, A_2,... A_n. We denote the Student agent as B. Consequently we represent the parameters of actor and critic networks of Teacher agents with $\theta_{A_1}, \cdots, \theta_{A_n}$ and that of the Student agent with θ_B. In every "episode," m times for every Teacher agent, we do a rollout of the Teacher agent followed by a rollout of the Student agent that aims to reach the goal set by the corresponding Teacher agent. After all the rollouts, we update the parameters of each Teacher agent (using the loss functions used in typical reinforcement learning algorithm such as TD3) followed by an update to the Student agent using a typical RL loss and a behavior cloning loss functions. We repeat this loop for a fixed number of episodes or until the Student agent learns to reach all the goals set by teachers.

Data: N, m ; `//Number of Teacher agents, multiplier`
Data: $\theta_{A_1}, \cdots, \theta_{A_n}, \theta_B$; `//Parameters for the agents`
for `episode` $= 1, 2, \cdots$ **do**
  for `trial` $= 1, 2, \cdots, N \cdot m$; `//Rollouts`
  do
    `Teacher`$[i//m]$ `sets goal;`
    `Student tries to achieve goal;`
  end
  for $i = 1, 2, \cdots, N \cdot m$ **do**
    `Update` $\theta_{A_{(i//m)}}$; `//RL Loss`
    `Update` θ_B ; `//RL and BC Loss`
  end
end

Algorithm 1: Multi Teacher Asymmetric Self-play

4 EXPERIMENTAL DOMAIN

Due to the unavailability of popular sparse reward environments for curriculum learning [1], we have created a driving environment using PyBullet [6]. The code for the driving environment is available at https://github.com/kharyal/pybullet-driving-env. Some experiments were also run using Cogment [15] and the code is made available at https://github.com/kharyal/cogment-verse/tree/main

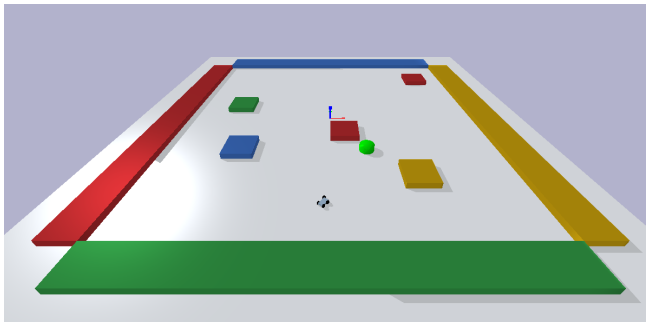

Figure 1: The PyBullet driving environment allows a Teacher to set the goal state (•) by interacting with the environment and the Student tries to reach the goal set by a Teacher.

Observation Space. The environment returns an observation vector and an occupancy map. The observaction vector is composed of $[x, y, \psi, \theta, \phi, v_x, v_y, g_x, g_y]$, where x and y are the x and y coordinates of the agent, ψ, θ, and $\phi \in [-1, 1]$ are the normalized Euler angles, and v_x, v_y are agent the velocities in the x and y directions, and g_x and g_y define the x and y coordinates of the goal.

[1] In particular, a resettable environment (or one with a single, fixed start state) is required as the Student needs to start from the same initial location as Teacher [13]. For example, most of the popular OpenAI gym [3] environments (like LunarLander-v2, BipedalWalker-v2, HandManipulateBlock-v0 etc.) do not support this kind of resetability.

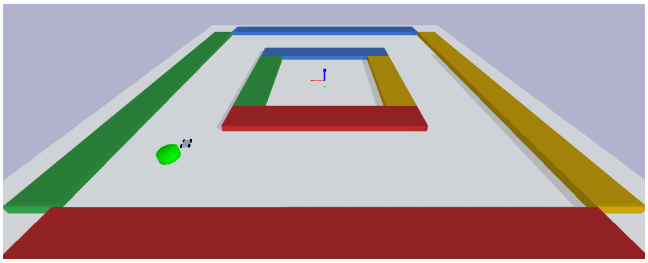

Figure 2: Racetrack task

The occupancy map is a 75×75, three-channeled binary map: the first channel denotes the position of car, the second channel shows obstacles on the map, and the third shows the goal location.

Note that we provide both the occupancy map and the observation vector as the vector does not have information about the obstacles' position, and the image does not contain information about agent orientation and velocity.

Action Space. The action is a continuous 2D vector $[t, s]$, where $t \in [0, 1]$ is the throttle and $s \in [-0.6, 0.6]$ is the steering angle.

Transition Dynamics. All obstacles are placed randomly for one Teacher-Student roll-out according to a configurable probability distribution. Given the action, PyBullet solves for the joint and motor control of the car's steering using realistic physics (including friction and inertia), to produce the subsequent state.

5 EXPERIMENTS

This section empirically evaluates Adversarial Multi-Teacher Curriculum Design with Traces in our driving domain. First, we consider how a Teacher can improve a Student's learning (relative to no curriculum). Second, we analyse how the number of Teacher agents effects the Student's performance on unseen goals in a sparse-reward environment. Third, we test how important the behavioural cloning loss is to Student's learning.

We implement all the Teacher and Student agents as independent policies with similar network architectures with memory. Unlike Teacher, Student has an extra condition on the goal in its policy. We use TD3 [7] to train both Teacher and Student agents. The hyperparameters used for the training are given in Table (2). When the teacher runs, we allow it to run for 250 environmental steps — the goal the teacher sets is the teacher agent's location on step 250. The student has a maximum number of 375 environmental steps — if the student reaches the goal, the episode stops.

Experiment specifics are summarized in Table 1. *Multiplier* denotes the number of goals each teacher sets in one episode and is introduced to ensure that each student gets to solve for an equal number of goals. For example, each teacher in a 4-teacher run sets one goal in one episode, while each teacher in a 2-teacher run sets two goals, thus making the total number of goals set in each run constant. For the no-curriculum baseline, since there are no teachers, the multiplier denotes the number of random goals that the student agent solves for in one episode.

We measure the generalization ability of trained agents by testing them on randomly generated, unseen goals. Apart from this, we consider a racetrack-like configuration with unseen obstacles (Figure 2).

Table 1: Run specific parameters

Run	*# Teacher*	*Multiplier*
1-Teacher	1	4
2-Teacher	2	2
4-Teacher	4	1
no-curriculum baseline	-	4
no-BCL baseline	1	4

Table 2: Hyperparameters used for TD3

Hyperparameter	*Value*
Discount Factor (γ)	0.99
Policy noise (σ_{TD3})	0.1
Policy noise clip (c)	0.2
Policy frequency	2
Optimizer	Adam[9]
Learning rate	3×10^{-4}
τ	0.005
Batch Size	512
Replay Buffer size	5000

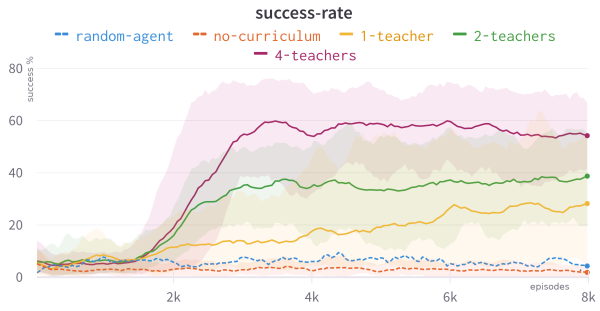

Figure 3: This graph of the success rate (%) on random goals versus episodes (averaged over 7 runs) shows that generalizability increases with the increase in number of Teacher agents. Moreover, we can note that the no-curriculum baseline fails to learn anything useful due to the sparse reward setting.

5.1 Impact of Adversarial Teacher Curriculum Design with Traces

Figure 3 shows Student's success rate on unseen goals, when trained with different number of Teacher agents. Consider the difference between the blue learning curve (with a single teacher) and the purple learning curve (where there is no curriculum). This result shows how a teacher can help improve the learning rate of the student. Due to the sparse reward nature of the environment, the

agent trained without curriculum fails to learn, as it doesn't get enough feedback from the environment. Note that the total number of environment interactions in an episode for the no-curriculum agent is less than the agents in our method since there are no teacher agent. To compensate for this, we run the experiment for a longer time and show that the no-curriculum agent does not learn much even after the 4-teacher agent has reached its peak.

5.2 Impact of Multiple Teachers

Figure 3 also compares the impact of different numbers of Teacher agents. As the number of Teachers increases, the adversarial training is more stable, which leads to better generalization - as seen in Figures (4) and (5), 4 Teachers provide better adversarial challenge. This prevents the Student from overfitting on the narrow set of goals a single Teacher generates.

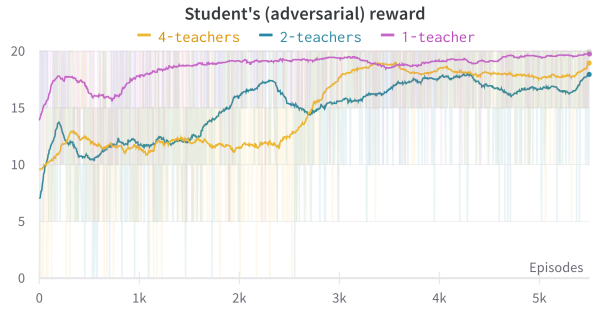

Figure 4: Student's (adversarial) reward averaged over 100 episodes vs episodes. It can be noted that 2 and 4 Teacher provide much more adversarial challenge to Student as compared to single Teacher. For lesser number of Teacher agents, Student can easily overfit the goals set by them until they learn to set new goals. This results in reduced generalizability on unseen goals despite Student's reward being extremely high throughout the training.

5.3 Impact of Behavioural Cloning

Figure 6 shows the impact of behavior cloning. Surprisingly, removing the behavior cloning loss returns the student performance to that similar to the performance when having no curriculum. Our hypothesis for this behavior is the challenging nature of the sparse reward environment accompanied by longer episodes making it practically impossible for the Student agent (a normal goal conditioned RL agent) to learn just based on the typical RL gradient updates (like policy gradient). This inability of the Student agent to learn quickly further leads to lack of feedback to the Teacher agent(s) making them only as good as a random Teacher agent that generates arbitrary goals. However this is only a preliminary result and we consider thorough ablation studies to be part of the future work.

6 CONCLUSION AND FUTURE WORK

We have proposed a novel Adversarial Multi-Teacher Curriculum Design with Traces which uses multiple learning agents to produce

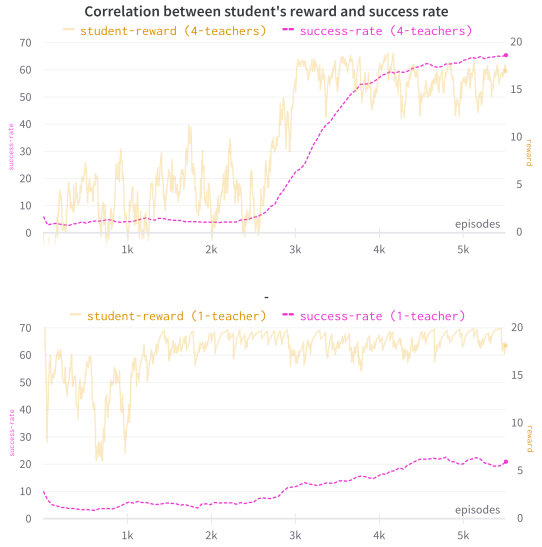

Figure 5: Success rate (left axis) and Student's reward (right axis) on random goals vs episodes. For more Teacher agents (top), better Student's reward hints towards better generalization, which is not the case with lesser number of Teacher agents (bottom). This suggests a better adversarial learning with increasing number of Teacher agents.

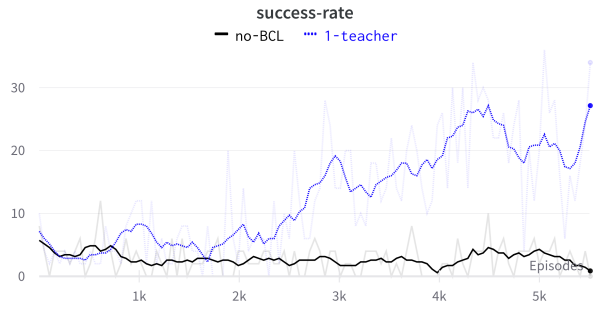

Figure 6: Average success rate vs episodes for 1 Teacher and no-BCL baseline. This graph demonstrates that without the Behavioural Cloning Loss, Student fails to learn much.

a diverse curriculum. We have also shown that this approach works well on sparse reward environments. We have also created a new environment which can act as a test-bench for future work in the field of curriculum based and adversarial RL.

We have multiple goals for future work. First, we will test this approach in multiple domains (with resettable simulators). Second, we will investigate whether the Teacher and Student agents can begin episodes from slightly different start states (e.g., a robotic arm can be reset to a fixed start state, but it will never be perfectly the same). Third, we will analyze Teachers' learned polices to understand what successful goal setting looks like in different domains. Fourth, we will further analyze why Adversarial Multi-Teacher

Curriculum Design with Traces performs so much better than the case when there is no behavior cloning loss.

ACKNOWLEDGMENTS

We would like to acknowledge and extend ours warmest thanks to Sai Krishna Gottipati for the insightful discussions and help throughout the process of working on this paper. This work has taken place in part in the Intelligent Robot Learning Lab at the University of Alberta, which is supported in part by research grants from the AI4Society; the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; and NSERC.

REFERENCES

[1] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. *CoRR* abs/1606.01868 (2016). arXiv:1606.01868 http://arxiv.org/abs/1606.01868

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/1553374.1553380

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:1606.01540 [cs.LG]

[4] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. 2018. Exploration by Random Network Distillation. *CoRR* abs/1810.12894 (2018). arXiv:1810.12894 http://arxiv.org/abs/1810.12894

[5] Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B. Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. 2021. Learning with AMIGo: Adversarially Motivated Intrinsic Goals. arXiv:2006.12122 [cs.LG]

[6] Erwin Coumans and Yunfei Bai. 2016–2021. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org.

[7] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. arXiv:1802.09477 [cs.AI]

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[9] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[10] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research* 21, 181 (2020), 1–50. http://jmlr.org/papers/v21/20-212.html

[11] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *CoRR* abs/2003.04960 (2020). arXiv:2003.04960 https://arxiv.org/abs/2003.04960

[12] OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique Ponde de Oliveira Pinto, Alex Paino, Hyeonwoo Noh, Lilian Weng, Qiming Yuan, Casey Chu, and Wojciech Zaremba. 2021. Asymmetric self-play for automatic goal discovery in robotic manipulation. *CoRR* abs/2101.04882 (2021). arXiv:2101.04882 https://arxiv.org/abs/2101.04882

[13] OpenAI OpenAI, Matthias Plappert, Raul Sampedro, Tao Xu, Ilge Akkaya, Vineet Kosaraju, Peter Welinder, Ruben D'Sa, Arthur Petron, Henrique P. d. O. Pinto, Alex Paino, Hyeonwoo Noh, Lilian Weng, Qiming Yuan, Casey Chu, and Wojciech Zaremba. 2021. Asymmetric self-play for automatic goal discovery in robotic manipulation. arXiv:2101.04882 [cs.LG]

[14] Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. 2019. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. *CoRR* abs/1903.03698 (2019). arXiv:1903.03698 http://arxiv.org/abs/1903.03698

[15] A. I. Redefined, Sai Krishna Gottipati, Sagar Kurandwad, Clod'eric Mars, Gregory Szriftgiser, and François Chabot. 2021. Cogment: Open Source Framework For Distributed Multi-actor Training, Deployment & Operations. *CoRR* abs/2106.11345 (2021). arXiv:2106.11345 https://arxiv.org/abs/2106.11345

[16] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html

[17] Zeyu Zheng, Junhyuk Oh, and Satinder Singh. 2018. On Learning Intrinsic Rewards for Policy Gradient Methods. *CoRR* abs/1804.06459 (2018). arXiv:1804.06459 http://arxiv.org/abs/1804.06459