Directed curiosity-driven exploration in hard exploration, sparse reward environments

Asad Jeewa[0000−0003−4329−8137], Anban Pillay[0000−0001−7160−6972], and Edgar Jembere[0000−0003−1776−1925]

University of KwaZulu-Natal, Westville 4000, South Africa
asad.jeewa@gmail.com
{pillayw4,jemberee}@ukzn.ac.za

Abstract. Training agents in hard exploration, sparse reward environments is a difficult task since the reward feedback is insufficient for meaningful learning. In this work, we propose a new technique, called Directed Curiosity, that is a hybrid of Curiosity-Driven Exploration and distance-based reward shaping. The technique is evaluated in a custom navigation task where an agent tries to learn the shortest path to a distant target, in environments of varying difficulty. The technique is compared to agents trained with only a shaped reward signal, a curiosity signal as well as a sparse reward signal. It is shown that directed curiosity is the most successful in hard exploration environments, with the benefits of the approach being highlighted in environments with numerous obstacles and decision points. The limitations of the shaped reward function are also discussed.

Keywords: Sparse Rewards · Hard Exploration · Curiosity · Reward Shaping · Navigation.

1 Introduction

A reinforcement learning agent learns how to behave based on rewards and punishments it receives through interactions with an environment [18]. The reward signal is the only learning signal that the agent receives [2]. Many environments have extrinsic rewards that are sparsely distributed, meaning that most timesteps do not return any positive or negative feedback. These environments, known as sparse reward environments [12,21], do not provide sufficient feedback for meaningful learning to take place [17]. The most difficult sparse reward environments are those where an agent only receives a reward for completing a task or reaching a goal, meaning that all intermediate steps do not receive rewards. These are referred to as terminal reward environments [7].

Closely related to the sparse rewards problem is the issue of exploration. Exploration algorithms aim to reduce the uncertainty of an agents understanding of its environment [4]. It is not possible for an agent to act optimally until it has sufficiently explored the environment and identified all of the opportunities for reward [24]. An agent may never obtain positive rewards without an intuitive

exploration strategy when rewards are sparse. Hard exploration environments are environments where local exploration strategies such as ϵ-greedy are insufficient [4]. In these environments, the probability of reaching a goal state through local exploration is negligible.

These types of environments are prevalent in the real-world [17] and training reinforcement learning (RL) agents in them forms one of the biggest challenges in the field [1]. This research focuses on learning in hard exploration, terminal reward environments.

A popular approach to learning in these environments is reward shaping, which guides the learning process by augmenting the reward signal with supplemental rewards for intermediate actions that lead to success [15]. This ensures that the agent receives sufficient feedback for learning.

Intrinsic rewards that replace or augment extrinsic rewards is another area of research that has exhibited promising results [4,7,17]. Instead of relying on feedback from the environment, an agent engineers its own rewards. Curiosity is a type of intrinsic reward that encourages an agent to find "novel" states [17].

In this research, we present *Directed Curiosity*: a new technique that hybridises reward shaping and Curiosity-Driven Exploration [17] to allow agents to explore intelligently. The algorithm is defined in Section 3 and the custom navigation environments used for evaluation are described in Section 4. The performance of the algorithm is evaluated by comparing it to its constituent algorithms i.e. agents trained with only the shaped reward and only the curiosity reward. Directed Curiosity is shown to be the most robust technique in Section 5. The environment characteristics that are suited to this technique are highlighted and the limitations of the shaped reward function are also discussed.

2 Related Work

Learning in hard exploration, sparse reward environments is a well-studied area in reinforcement learning. Reward shaping is a popular approach that augments the reward signal with additional rewards to enable learning in sparse reward environments. It is a means of introducing prior knowledge to reduce the number of suboptimal actions [9] and guide the learning process [14]. A concern is that when reward shaping is used incorrectly, it can have a detrimental effect and change the optimal policy or the definition of the task [9,15].

Potential-Based Reward Shaping has been proven to preserve the optimal policy of a task [9,15]. It defines ϕ, a reward function over states that introduces "artificial" shaped reward feedback [3]. The potential function F is defined as a difference between ϕ of the next state s' and the current state s with γ as a discount factor on $\phi(s')$.

The restriction on the form of the reward shaping signal limits its expressiveness [14]. Potential-Based Advice is a similar framework that introduces actions in the potential function [26]. A novel Bayesian approach that augments the reward distribution with prior beliefs is presented in [14].

It is difficult to manually engineer reward functions for each new environment [9,11]. Implicit reward shaping is an alternate approach that learns from demonstrations of target behaviour. A potential-based reward function is recovered from demonstrations using state similarity in [22] and through inverse reinforcement learning methods in [21]. The shaped reward function is learnt directly from raw pixel data in [11].

An alternative to "shaping" an extrinsic reward is to supplement it with intrinsic rewards [16] such as curiosity. Curiosity-Driven Exploration by Self-Supervised Prediction [17] is a fundamental paper that defined a framework for training curious agents. Curiosity empowers the agent by giving it the capability of exploration, enabling it to reach far away states that contain extrinsic rewards. Much research has built upon the findings of this paper. Large scale analysis of the approach is performed in [7] where agents learned to play various Atari Games using intrinsic rewards alone. A limitation of the approach is that it struggles to learn in stochastic environments [7].

Classic work in [6,13] investigated balancing exploration and exploitation in polynomial time and has inspired much research in the area of intelligent exploration. Count-based exploration methods generate an exploration-bonus from state visitation counts [24]. It has been shown to achieve good results on the notoriously difficult "Montezuma's Revenge" Atari game in [4,5]. Exploration bonuses encourage an agent to explore, even when the environment's reward is sparse [4], by optimising a reward function that is the sum of the extrinsic reward and exploration bonus.

Approximating these counts in large state spaces is a difficult task [24]. Hash functions were used in [24] to extend the method to high-dimensional, continuous state spaces. Random Network Distillation (RND) [8] is a novel technique that consists of a fixed randomly initialised target network and a prediction network. The target network outputs a random function of the environment states which the prediction network learns to predict. An intrinsic reward is defined as the loss of the prediction network. It achieved state of the art performance on "Montezuma's Revenge" [5] in 2018.

Other methods of exploration include maximising empowerment [10], wherein the long-term goal of the agent aims to maximise its control on the environment, using the prediction error in the feature space of an auto-encoder as a measure of interesting states to explore, and using demonstration data to learn an exploration policy [23].

3 Directed Curiosity

We propose a new reward function that is made up of two constituents: a distance-based shaped extrinsic reward and a curiosity-based intrinsic reward.

3.1 Distance-Based Reward Shaping

Shaping rewards is a fragile process since small changes in the reward function result in significant changes to the learned policy [25].

Various functions were engineered and compared. It is essential that the positive and negative rewards are balanced. In an episode, the agent should not receive more positive rewards for moving closer to the target, or more negative rewards for moving further away, so as not to introduce loopholes for the agent to exploit. If the weighting of positive rewards is too high, the agent learns to game the system by delaying reaching the target to gain more positive rewards in an episode. If the weighting of the negative rewards is too high, the agent does not receive sufficient positive reinforcement to find the target. This means that the shaped rewards alter the optimal policy of the original task [15].

The shaped reward should encourage the agent to keep advancing towards the target by favouring consecutive positive moves and punishing consecutive negative ones. It must not dominate the terminal reward such that the agent is no longer incentivised to find the target and its motivations become polluted. To overcome these issues, a shaped reward function based on relative distance between target and agent is used.

Algorithm 1 Distance-based shaped reward function

Input: Agent position P_{agent}, target position P_{target}, maximum distance D_{max}, previous distance D_{prev}, reward coefficient C
  Calculate distance $D_{current} \leftarrow \text{distance}(P_{agent}, P_{target})$
  Calculate reward signal: $R \leftarrow D_{current}/D_{max}$
  if $D_{current} < D_{prev}$ **then**
    return $C \cdot (1 - R)$
  else
    return $C \cdot (-R)$
  end if

There are various benefits to Algorithm 1. The agent is penalised if it stays still and the shaped reward signal can be controlled using the reward coefficient C. This ensures that the episodic shaped rewards cannot exceed terminal positive reward. There is a balance between positive and negative rewards since they are both relative to the change in distance. The agent receives the highest reward when it moves closest to the target and the highest penalty when it moves furthest away. This means that the shaped reward function is policy invariant i.e. it does not alter the goal of the agent to learn the optimal path to the target.

Since the rewards are shaped exclusively based on distance metrics that do not take into account the specific dynamics of the environment, the same function can be used across different environments, and in general, for navigation tasks. A limitation of this approach is that the target location needs to be known. We have investigated using ray casts to find the location of the target if it is unknown, however, the scope of this research is to teach an agent to navigate past obstacles and find an optimal path, given a starting point and a destination. The definition of the task changes drastically, from a navigation-based one to

a goal-finding or search task, when the location is unknown. This is a possible area for future work.

3.2 Curiosity-Driven Exploration

Pathak et al. [17] formally defined a framework for training curious agents that involves training two separate neural-networks: a forward and an inverse model that form an Intrinsic Curiosity Model (ICM). The inverse model encodes the current and next observation into a feature space ϕ and learns to predict the action $\hat{a}_t$ that was taken between the occurrence of the two encoded observations. The forward model is trained to take the current encoded observation and action and predict the next encoded observation.

$$r_i^t = \frac{\eta}{2}\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2 \tag{1}$$

In order to generate a curiosity reward signal, the inverse and forward dynamics models' loss functions are jointly optimised i.e. curiosity is defined as the difference between the predicted feature vector of the next state and the real feature vector of the next state. η is a scaling factor.

As an agents explores, it learns more about its environment and becomes less curious. A major benefit of this approach is that it is robust: by combining the two models, the reward only captures surprising states that have come about directly as a result of the agents actions.

3.3 Intelligent Exploration

We propose hybridising curiosity [17] and distance-based reward shaping. Using reward shaping alone is flawed since the agent cannot navigate past obstacles to find a target. Using curiosity alone may cause the agent to spend too much time exploring, after the target has been found, and get trapped in a suboptimal state. By combining the two approaches the agent is able to explore and learn about the dynamics of the environment, while always keeping in mind its goal of finding an optimal path to the target. The agent learns in a more directed and intuitive manner. Curiosity enables the agent to find the target, while the shaped rewards provide feedback to the agent that enables it to learn a path to the goal.

Directed Curiosity simultaneously maximises two reward signals. The reward function components are somewhat conflicting so it is essential to find a balance between them. The agent needs sufficient time to explore the environment, while also ensuring that it does not converge to a suboptimal policy too quickly. This is similar to the exploration vs exploitation Problem in RL. We balance the reward by manually tuning weights attached to both the constituent reward signals. In future work, we wish to find a means of dynamically weighting the reward signals during training. We also wish to investigate alternative means of combining them.

Algorithm 2 Directed Curiosity-Driven Exploration

Input: Initial policy π_0, extrinsic reward weighting w_e, intrinsic reward weighting w_i, max steps T, decision frequency D
for $i \leftarrow 0$ to T **do**
  Run policy π_i for D timesteps
  Calculate distance-based shaped reward r_e^t (Algorithm 1)
  Calculate intrinsic reward r_i^t (Equation 1)
  Compute total rewards $r_t = w_i \cdot r_i^t + w_e \cdot r_e^t$
  Take policy step from π_i to π_{i+1}, using PPO [20] with reward function r_t
end for

PPO [19] is a popular policy gradient method that is robust and simpler than alternative approaches. Our algorithm is trained using PPO though an arbitrary policy gradient method can be used.

4 Methodology

4.1 Learning Environment

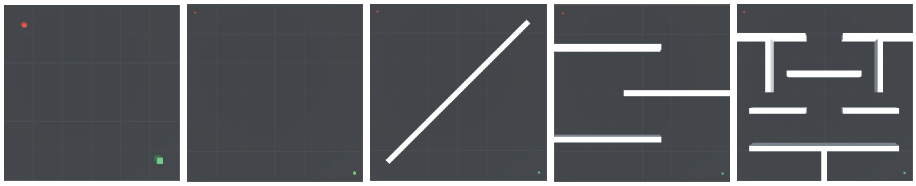
A custom testing environment was created to analyse the performance of our technique, based on the principal of pathfinding. It consists of a ball and a target. The ball is an agent that must learn to navigate to the target, in the shortest possible time (see Fig. 1). The agent is penalised every time it falls off the platform, since there are no walls along the boundaries and it receives a positive reward upon reaching the target. An episode terminates upon falling off the platform, reaching the target, or after a maximum number of steps.

The benefit of this environment is that it defines a simple base task of finding an optimal path to a target. This allows us to perform thorough analysis of the algorithm by continuously increasing the difficulty of the task. In this way, we are able to identify its limitations and strengths. The environment represents a generalisation for navigation tasks wherein an agent only receives positive feedback upon reaching its destination.

The agent is equipped with a set of discrete actions. Action 1 defines forward and backward movement while action 2 defines left and right movement. Simultaneously choosing the actions allows the agent to move diagonally. The agent's observations are vectors representing its current position and the target position. It is not given any information about the dynamics of the environment. The agent must learn an optimal policy that finds the shortest path to the target.

The baseline reward function was carefully tuned: a +100 reward is received for finding the target, -100 penalty for falling off the platform and -0.01 penalty every timestep. The reasoning behind the selected values is to remove bias from the experiments. An agent cannot fall into a local optimum by favouring a single suboptimal policy. This is because a policy that immediately falls off the platform and a policy that learns to remain on the platform for the entire episode, without

finding the goal, will both return roughly the same episodic reward. This function was used as a baseline that was tuned for each new environment.



(a) BasicNav (b) HardNav (c) ObstacleNav (d) MazeNav1 (e) MazeNav2

Fig. 1: Learning Environments. The agent is shown in the top left in red and the target is shown in the bottom right in green.

For the simplest version of the task, the agent and target are placed at fixed locations, on the opposite sides of the platform, without any obstacles between them. We term this an easy exploration task since it is possible for an agent trained with only the sparse reward function to find the target. This is achieved by tuning the floor to agent ratio and agent speed. The shaped reward coefficient C in Algorithm 1 was amplified to 0.1 due to the simplicity of the environment. This is referred to as BasicNav (see Fig. 1a).

The next environment, termed HardNav (see Fig. 1b), is significantly larger. It is a hard exploration environment [17], since an agent trained with a sparse reward function is never able to find the target. Due to the increased number of episode steps, the shaped reward coefficient C in Algorithm 1 was dampened to 0.001.

We also perform testing in environments with walls that block the direct path to the goal and make finding the target more difficult. ObstacleNav (see Fig. 1c) has a single obstacle that is deliberately placed perpendicular to the optimal path to the target, forcing the agent to have to learn to move around the obstacle. The agent is never explicitly given any information about the obstacle. This environment was designed to test the limitations of Directed Curiosity since shaping the reward to minimise the distance to goal is counter-intuitive because it leads the agent directly into the obstacle. The coefficient C in Algorithm 1 was dampened to 0.001.

The remaining set of environments contain multiple walls and obstacles in a maze-like structure. These environments were designed to investigate if the agent can learn to move further away from the target at the current timestep, in order to pass obstacles and reach the target at a later timestep i.e. it needs foresight to succeed. We term the first maze as MazeNav1 (see Fig. 1d).

The last environment is the most difficult version of the task since it has dead-ends and multiple possible paths to the goal. This allows us to investigate the robustness of Directed Curiosity. Even after finding the target, it is difficult to generalise a path from the starting point to the destination since it is easy for the

agent to get stuck in dead-ends or behind obstacles. We term this environment as MazeNav2 (see Fig. 1e). Due to the increased complexities, the terminal reward was increased to +1000 and the shaped reward coefficient C in Algorithm 1 was dampened to 0.000001.

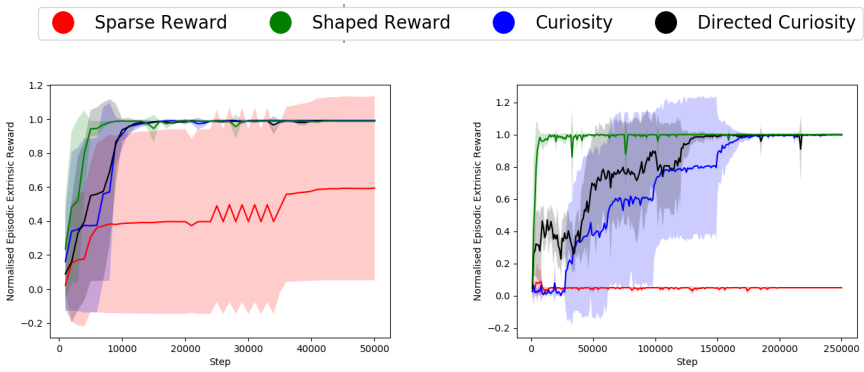
4.2 Hyperparameter Optimisation

It is important to carefully tune the hyperparameters for each environment. The success of the algorithms hinge on these values. Although literature guided this process, the hyperparameters were manually optimised, since the experiments were performed in custom environments. The base hyperparameters were found in BasicNav and then fine-tuned for all other environments, in order to cater for the increased complexities. PPO is a robust learning algorithm that did not require significant tuning [7], once the base hyperparameters were identified and this is a major reason for its selection.

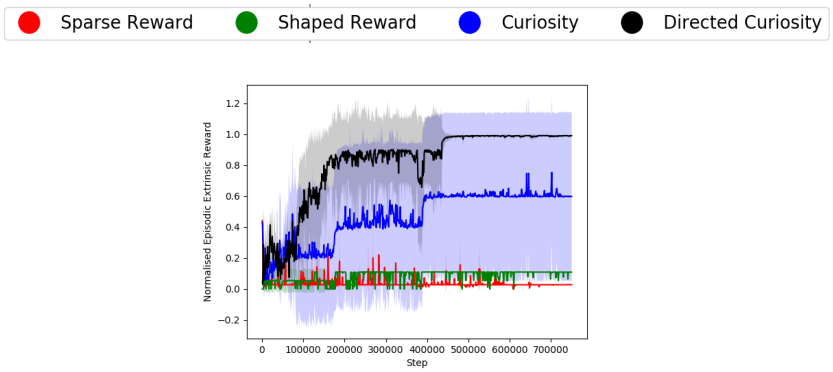
Hyperparameter tuning was essential in ensuring that the algorithms were able to perform meaningful learning. By attempting to tune the parameters to the best possible values, we were able to perform a fair comparison. The notable parameters are a batch size of 32, experience buffer size of 256 and a learning rate of $1.0e-5$. The strength of the entropy regularization β is $5.0e-3$ and the discount factor γ for both the curiosity and extrinsic reward is 0.99. The extrinsic reward weighting is 1.0 and the curiosity weighting is 0.1. The network has 2 hidden layers with 128 units. The baseline parameters were adjusted for each environment: the maximum training steps is 50000 in BasicNav, 250000 in HardNav, 750000 in ObstacleNav and 1000000 in MazeNav1 and MazeNav2.

5 Results

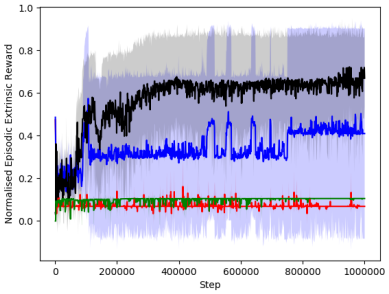
Each algorithm was run five times in every environment. 30 parallel instances of the same environment are used for data collection during training.



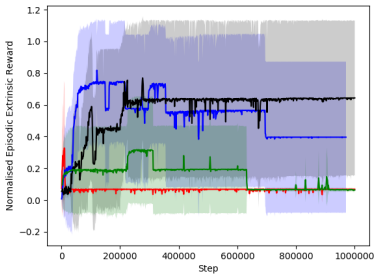
(a) BasicNav (b) HardNav



(c) ObstacleNav



(d) MazeNav1



(e) MazeNav2

Fig. 2: Learning curves for all environments. The average curve from the five runs is shown.

The sparse rewards agent does not perform consistently in BasicNav. The agent is able to find the target and learn an optimal policy on some runs only. This is the reason for the high variance in Fig. 2a. In the hard exploration environments, the agent learns to avoid falling off the platform but is unable to find the target on all runs and therefore receives no positive rewards in training. This highlights the need for an exploration strategy.

The reward shaping agent performs well in BasicNav. This is because the shaped rewards act as a definition of the task since there are no obstacles blocking the direct path to the goal. Continuously moving closer to the target on every timestep leads the agent to the goal in the shortest time. Even in HardNav, the agent is able to learn an optimal policy very quickly, for the same reasons.

The deficiencies of using the shaped reward only are exposed when obstacles are introduced (see Fig. 2c, Fig. 2d). The agent fails to find the target on all runs in ObstacleNav and MazeNav1 and gets stuck behind obstacles. This is because the shaped reward function is a greedy approach and the agent is not equipped with the foresight to learn to move around the obstacles. It cannot learn to move

further away from the target at the current time, in order to reach the target at a later stage.

In MazeNav2 (see Fig. 2e), the agent was able to find the target on two runs. Even though there are multiple obstacles, the optimal path to the goal in MazeNav2 is similar to that in HardNav. The agent "ignores" the obstacles and avoids dead-ends by acting simplistically. By the end of training, however, the agent was unable to converge to an optimal policy on any of the runs.

These results show that distance-based reward shaping provides the agent with some valuable feedback, but without an intuitive exploration strategy, the agent lacks the foresight needed to moves past obstacles that block it's path to the target.

The curiosity agent was able to consistently learn an optimal policy in the environments without obstacles. However, Fig. 2a shows that the curiosity agent takes longer to converge to an optimal policy in BasicNav. This highlights that curiosity is not necessary in environments that are not hard exploration. In HardNav (see Fig. 2b), the curiosity agent is still able to find an optimal policy on all runs, but it is significantly slower than the shaped reward function.

The necessity of the curiosity signal is highlighted when obstacles are introduced. Not only does it enable the agent to find the distant target, it also implicitly learns about the dynamics of the environment, allowing the agent to learn how to move past multiple obstacles.

In ObstacleNav (see Fig. 2c), the agent is still able to learn an optimal policy on most runs. The performance of the agent is not as successful in MazeNav1 (see Fig. 2d) and MazeNav2 (see Fig. 2e).The agent successfully learns an optimal policy on two of the runs. In these environments, it is difficult to converge to an optimal policy, once the target has been found. One reason for this is that the agent keeps exploring after initially finding the target and gets stuck behind obstacles and in dead-ends, eventually converging to an unsuccessful policy, without being able to reach the target again. The curiosity signal is insufficient to direct the agent back to the target and learn a path to the destination. This is the reason for the increase of the average reward in the early stages of training and the subsequent drop thereafter in Fig. 2e.

These results indicate that curiosity equips an agent with the ability to find a target in hard exploration environments with obstacles, but the agent requires additional feedback to consistently learn a path from the start point to the destination.

The Directed Curiosity agent is shown to be the most robust technique. Fig. 2a and Fig. 2b show that Directed Curiosity always finds an optimal policy in BasicNav and HardNav. It converges to a solution faster than the curiosity agent in HardNav, due to the additional shaped reward feedback.

The hard exploration environments highlight the benefits of the technique. It is the only technique that converges to an optimal solution on all runs in ObstacleNav (see Fig. 2c). Curiosity enables the agent to find the target and move past the obstacle, while the shaped rewards provide additional feedback

that allows the agent to learn an optimal path to the target, once it has been found.

MazeNav1 (see Fig. 2d) and MazeNav2 (see Fig. 2e) exhibit promising results since the Directed Curiosity agent learns an optimal policy on more runs than any other technique i.e. on three of the five runs. Training is more stable than the Curiosity agent. The agent always finds the target during training, however, it is unable to consistently find an optimal policy on all runs. A major reason is due to the limitations we have highlighted with the shaped reward function. In future work, we wish to investigate a more intuitive reward function that has foresight. Another reason is due to the complexities we have introduced in these environments. The reward feedback is not sufficient to guide the agent out of dead-ends back to the target. However, these results indicate that the two components of Directed Curiosity, when balanced correctly, allow the agent to learn in a more directed and intuitive manner.

For all algorithms, the variance of the results increase with the difficulty of the task since the agents do not always converge to an optimal policy i.e. when the agent does not learn a path to the target, it does not receive the terminal reward and hence its episodic rewards are significantly lower. PPO learns a stochastic policy, hence, even on the successful runs, the algorithms converge at different times. Due to the inherent randomness in the algorithm, the agent explores differently on every run and thus visits states in a different order.

6 Conclusions and Future Work

A new approach to learning in hard exploration, sparse reward environments, that maximises a reward signal made up of a hybrid of Curiosity-Driven Exploration [17] and distance-based reward-shaping, is presented. This algorithm is compared to baseline algorithms in a custom pathfinding environment and it is shown that the technique enables agents to learn in a more directed and intuitive manner.

The Directed Curiosity agent was the most robust technique. It was able to consistently learn an optimal policy in hard exploration environments with a single obstacle, and learned optimal polices more often then the other techniques, in hard exploration environments with multiple obstacles and dead-ends.

In future work, we wish to investigate alternative reward functions that are more flexible than the current greedy approach. We would like to perform further testing in existing benchmarked environments and in domains other than navigation. This requires further research into “intelligent exploration”, through hybridising different shaped reward signals and exploration strategies. Another interesting direction is to create environments with multiple targets and agents.

References

1. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O.P., Zaremba, W.: Hindsight experience replay. In: Advances in Neural Information Processing Systems. pp. 5048–5058 (2017)

2. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep Reinforcement Learning: A Brief Survey. IEEE Signal Processing Magazine **34**(6), 26–38 (Nov 2017). https://doi.org/10.1109/MSP.2017.2743240, http://ieeexplore.ieee.org/document/8103164/
3. Badnava, B., Mozayani, N.: A new Potential-Based Reward Shaping for Reinforcement Learning Agent. arXiv:1902.06239 [cs] (May 2019), http://arxiv.org/abs/1902.06239, arXiv: 1902.06239
4. Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., Munos, R.: Unifying count-based exploration and intrinsic motivation. In: Advances in Neural Information Processing Systems. pp. 1471–1479 (2016)
5. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
6. Brafman, R.I., Tennenholtz, M.: R-MAX - A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. Journal of Machine Learning Research **3**(Oct), 213–231 (2002), http://www.jmlr.org/papers/v3/brafman02a.html
7. Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., Efros, A.A.: Large-Scale Study of Curiosity-Driven Learning. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rJNwDjAqYX
8. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation. arXiv preprint arXiv:1810.12894 (2018)
9. Devlin, S.M., Kudenko, D.: Dynamic Potential-Based Reward Shaping (Jun 2012), http://eprints.whiterose.ac.uk/75121/
10. Gregor, K., Rezende, D.J., Wierstra, D.: Variational intrinsic control. arXiv preprint arXiv:1611.07507 (2016)
11. Hussein, A., Elyan, E., Gaber, M.M., Jayne, C.: Deep reward shaping from demonstrations. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 510–517. IEEE (2017)
12. Kang, B., Jie, Z., Feng, J.: Policy Optimization with Demonstrations p. 10 (2018)
13. Kearns, M., Singh, S.: Near-optimal reinforcement learning in polynomial time. Machine learning **49**(2-3), 209–232 (2002)
14. Marom, O., Rosman, B.: Belief reward shaping in reinforcement learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
15. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML. vol. 99, pp. 278–287 (1999)
16. Oudeyer, P.Y., Kaplan, F.: What is intrinsic motivation? A typology of computational approaches. Frontiers in neurorobotics **1**, 6 (2009)
17. Pathak, D., Agrawal, P., Efros, A.A., Darrell, T.: Curiosity-Driven Exploration by Self-Supervised Prediction. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 488–489. IEEE, Honolulu, HI, USA (Jul 2017). https://doi.org/10.1109/CVPRW.2017.70, http://ieeexplore.ieee.org/document/8014804/
18. Ravishankar, N.R., Vijayakumar, M.V.: Reinforcement Learning Algorithms: Survey and Classification. Indian Journal of Science and Technology **10**(1) (Jan 2017). https://doi.org/10.17485/ijst/2017/v10i1/109385, http://www.indjst.org/index.php/indjst/article/view/109385
19. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv preprint arXiv:1511.05952 (2015)
20. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs] (Jul 2017), http://arxiv.org/abs/1707.06347, arXiv: 1707.06347

21. Suay, H.B., Brys, T.: Learning from Demonstration for Shaping through Inverse Reinforcement Learning p. 9 (2016)
22. Suay, H.B., Brys, T., Taylor, M.E., Chernova, S.: Reward Shaping by Demonstration. In: Proceedings of the Multi-Disciplinary Conference on Reinforcement Learning and Decision Making (RLDM) (2015)
23. Subramanian, K., Isbell Jr, C.L., Thomaz, A.L.: Exploration from demonstration for interactive reinforcement learning. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. pp. 447–456. International Foundation for Autonomous Agents and Multiagent Systems (2016)
24. Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., Abbeel, P.: #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 2753–2762. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/6868-exploration-a-study-of-count-based-exploration-for-deep-reinforcement-learning.pdf
25. Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., Riedmiller, M.: Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. arXiv:1707.08817 [cs] (Jul 2017), http://arxiv.org/abs/1707.08817, arXiv: 1707.08817
26. Wiewiora, E., Cottrell, G.W., Elkan, C.: Principled methods for advising reinforcement learning agents. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). pp. 792–799 (2003)