Learning to Generalise in Sparse Reward Navigation Environments

Asad Jeewa[1,2(✉)], Anban W. Pillay[1,2], and Edgar Jembere[1,2]

[1] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville 4000, South Africa
asad.jeewa@gmail.com, {pillayw4,jemberee}@ukzn.ac.za
[2] Centre for Aritificial Intelligence Research, Cape Town, South Africa

Abstract. It is customary for RL agents to use the same environments for both training and testing. This causes the agents to learn specialist policies that fail to generalise even when small changes are made to the training environment. The generalisation problem is further compounded in sparse reward environments. This work evaluates the efficacy of curriculum learning for improving generalisation in sparse reward navigation environments: we present a manually designed training curriculum and use it to train agents to navigate past obstacles to distant targets, across several hand-crafted maze environments. The curriculum is evaluated against curiosity-driven exploration and a hybrid of the two algorithms, in terms of both training and testing performance. Using the curriculum resulted in better generalisation: agents were able to find targets in more testing environments, including some with completely new environment characteristics. It also resulted in decreased training times and eliminated the need for any reward shaping. Combining the two approaches did not provide any meaningful benefits and resulted in inferior policy generalisation.

Keywords: Generalisation · Curriculum learning · Sparse rewards · Navigation

1 Introduction

A fundamental challenge in reinforcement learning (RL) is that of generalisation [7]. It is customary for RL agents to use the same environments for both training and testing [8], as is the case for the Arcade Learning Environment [3], the classic RL benchmark. Agents therefore exhibit breakthrough results on very specific tasks but fail to generalise beyond the training environment [28]. Making small changes to the environment or task often leads to a dramatic decrease in performance [41,42]. This is because agents tend to memorise action sequences and therefore overfit to the training environments [7].

The generalisation problem is compounded in sparse reward environments. RL agents learn behaviour based solely on rewards received through interactions

A. Gerber (Ed.): SACAIR 2020, CCIS 1342, pp. 85–100, 2020.
https://doi.org/10.1007/978-3-030-66151-9_6

with an environment [31]. However, many environments have extrinsic rewards that are sparsely distributed, meaning that the environments does not return any positive or negative feedback to the agent on most timesteps. These environments are prevalent in the real-world [29] and training RL agents in them remains a major challenge [2]. There are various novel approaches to learning in these environments [2] but they tend to emphasise learning specialist polices that fail to generalise to unseen testing environments [7].

This research focuses on policy generalisation in sparse reward navigation environments. Policy generalisation refers to the extent to which a policy transfers to unseen environments within the same domain [8] without any additional training or fine-tuning. This is a difficult task since it is only possible for agents to learn on a small subset of possible states but it is desirable that they should be able to generalise and produce a good approximation over a larger state space [38]. In this work, agents are required to learn to navigate to distant targets across multiple environments, with different characteristics or obstacle configurations.

This work focuses on two approaches. The first technique is curriculum learning. When it is difficult for an agent to learn a task directly, a training curriculum can be designed to gradually increase an agent's knowledge over time. The curriculum imposes an order on training [14]: the agent is trained on a series of simpler tasks that progressively gets more difficult [25]. This enables it to learn "skills" that can be transferred to solve difficult tasks [25]. In this manner, the curriculum can be used to bypass the sparse rewards problem [12]. Curriculum learning has been shown to decrease training times as well as improve generalisation [4,12].

The second approach introduces intrinsic rewards to augment sparse extrinsic rewards. Intrinsic rewards are generated by the agent itself, instead of relying on feedback from the environment. Curiosity is a type of intrinsic reward that encourages an agent to find "novel" states [29] and has been used to learn policies that generalise to unseen environments.

In this research, we investigate the problem of generalisation in sparse reward navigation environments by evaluating the efficacy of curriculum learning for improving generalisation in this domain. A manually-designed curriculum for sparse reward navigation environments is presented and used to train agents in a suite of hand-crafted environments. Both training and testing performance of the curriculum is empirically compared and contrasted to two baseline algorithms: curiosity-driven exploration [29] and a hybrid approach that combines the curriculum with curiosity. The policies are evaluated in multiple testing environments that are either variations of the training environments or include entirely new characteristics.

The task, algorithms and environments are formally defined in Sect. 3. The benefits and limitations of the curriculum are discussed in Sect. 4: using the curriculum resulted in polices that generalised better than curiosity as well as decreased training times. Section 5 summarises the findings and discusses directions for future work.

2 Related Work

Generalisation remains a fundamental RL problems since agents tend to memorise trajectories from their training environments instead of learning transferable skills [7]. Classic RL benchmarks like the Arcade Learning Environment (ALE) [3] focus on creating specialist agents that perform well in a single environment. New benchmarks have been proposed to focus research on generalisation. The ProcGen Benchmark [7] uses procedural generation to generate new environments. The inherent diversity in the generated environments demands that agents learn robust polices in order to succeed. A similar framework is presented in [19] with larger scale three-dimensional environments.

Justesen et al. [20] however, highlighted limitations of procedural generation: it is difficult to automatically scale the difficulty of the task [20] and the distribution of the procedurally generated environments is often different to that of human-generated environments. Procedurally generating environments may lead to overfitting to the distribution of the generated environments [20]. A novel approach that uses reinforcement learning to learn a policy for generating environments shows promising results in [23].

Our work is inspired by Savinov et al. [32]. The authors emphasised the need for separate training and testing environments and investigated generalisation in custom maze environments with random goal placements. The aims of the study were different but the principles were incorporated into the curriculum defined in Subsect. 3.3. Similar findings were highlighted in other studies [8,42].

Curriculum learning was shown to decrease training times and improve generalisation across multiple common datasets in [4]. The main idea is to split a complex task into smaller, easier-to-solve sub-problems and controlling the curriculum to ensure that the task is never too difficult for the agent [17]. Previous work manually generated training curricula for various tasks [22,34]. A limitation of this approach is the requirement of expert domain knowledge [39]. Various studies attempted to alleviate this problem by presenting novel techniques for automatically generating a curriculum [12,24,39]. Florensa et al. [12] presented a method for automatically generating a curriculum that exhibited promising results in sparse reward navigation environments. The maze environments from the study have been incorporated into this study. The curriculum in this work is manually designed though only general concepts, such as environment size and obstacle configuration, were varied so as to ensure it did not require significant fine-tuning or expert knowledge.

Curriculum learning is an implicit form of generalisation [4]. Closely related to curriculum learning is hierarchical reinforcement learning. Tessler et al. [40] presented a framework that enabled agents to transfer "skills" learnt from easy sub-tasks to difficult tasks requiring multiple skills. Agents learnt "high-level" actions that pertain to walking and movement and used these skills to learn difficult navigation tasks faster in [13]. Our curriculum has been designed to implicitly learn in this manner since there are no obstacles in the early stages of training, thereby allowing agents to focus on locomotion.

The sparse reward problem is well-studied in reinforcement learning. Many novel approaches emphasise learning specialist polices and do not focus on generalisation [7]. Reward shaping augments the reward signal with additional rewards to enable learning in sparse reward environments. It can have a detrimental effect on training if it is used incorrectly and can change the optimal policy or the definition of the task [9,26,28]. Manually engineering reward functions for each new environment is difficult [9,15]. Alternatively, reward functions were recovered from demonstrations in [15,36,37]. Shaped rewards can result in specialist policies that generalise poorly [15]. The problem was investigated in [16] where agents learnt polices that were optimised for single training environments. A major benefit of our curriculum is that it does not require any reward shaping.

An alternative to "shaping" an extrinsic reward is to supplement it with intrinsic rewards [27] such as curiosity. Curiosity-Driven Exploration by Self-Supervised Prediction [29] formally 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. A well-known limitation of the approach is that agents often find a source of randomness in an environment that allows it to inadvertently satiate its curiosity [5]. There are various other novel approaches [6,33].

Curiosity has been chosen as a baseline as it has shown promising generalisation capabilities in previous studies [5,29]. Agents struggled to generalise to environments with different textures in [5,29]. This is not relevant to this study since agents observations are vector rather than visual representations (see Subsect. 3.1).

To our knowledge, curriculum learning has not been evaluated extensively with regards to generalisation in sparse reward navigation environments.

3 Methodology

3.1 The Task

The goal of the agent is to navigate from its starting point to a fixed distant target, with obstacles or walls placed along its route. The agent is required to learn foresight: it needs to learn to move further away from the target in the present, in order to find the target in the future. The task is a variation of the classic point-mass navigation task in various studies [10,11]. We consider an agent interacting with an environment in discrete time steps. At each time step t, the agent gets an observation o_t of the environment and then takes an action a_t from a set of actions A.

The observation set O comprises the coordinates of the agent's current position, the coordinates of the target, the distance to the goal and rays that extend in 8 directions, at 45° intervals. These short rays provide essential feedback to the agent by enabling it to detect walls and targets that are in its vicinity and therefore adapt its policy accordingly.

The rays take on additional importance when agents are placed in previously unseen environments since they enable the agents to learn robust policies: when

an agent detects an obstacle in its vicinity, it needs to learn to move away from the obstacle, in the direction of an open path. If an agent executes memorised actions, it will move directly into walls and never reach its destination.

The ray length was tuned to balance the difficulty of the task: if the rays are too long, the agent unrealistically detects objects that are far away but if it is too short, the agent is unable to detect anything except that which is immediately in front of it. This is analogous to the field of view. The observations were stacked to equip the agents with a small memory of the immediate past. The previous ten observations were stored at any given time.

The action set a_t allows the agent to move in eight directions: forwards, backwards, sideways as well as diagonally, unlike the standard Gridworld task [42].

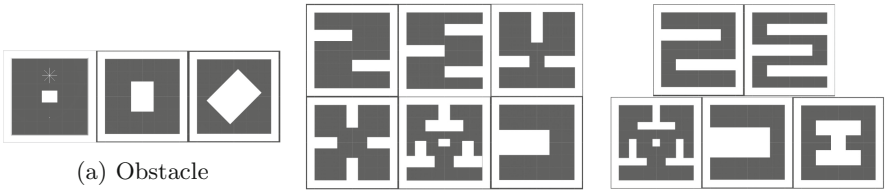
By default, before any training modifications are made, the environments are all sparse reward environments since the agent only receives a +1 reward for finding the target. The starting positions of the agent and the target are far away from each other, on different ends of the environment. The agents do not receive any intermediate rewards and incur a small penalty on every timestep, to encourage them to find the target in the shortest possible time.

3.2 Environments

There are multiple environments and each varies in terms of the configuration of walls and obstacles (see Fig. 1). This is to deter agents from learning an optimal policy in one single environment, rather learning the "skill" of finding a target in an arbitrary navigation environment. The predefined environments were carefully designed to represent high-level features or environment characteristics that include dead-ends and multiple paths to the target. We theorise that introducing agents to numerous environment features in training enables them to learn a flexible policy that enables them to find targets when similar features are found in new environments. The environments were divided into a set of training and testing environments. The generalisability of the agents was evaluated in the testing environments.

The training environments were further divided into three categories: *Obstacle* environments (see Fig. 1a) contain only a single obstacle that varies in terms of size and orientation. The sizes range from a scale of 0 to 3 and the orientation is defined as any angle from 0°, in 45° increments. The size of the agent and ray length are also depicted in Fig. 1a to illustrate the scale of the task.

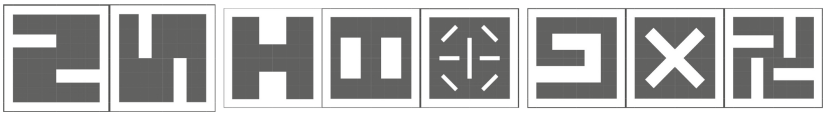
Maze environments have multiple obstacles and were subdivided based on difficulty. There are *Standard* mazes in Fig. 1b and *Difficult* mazes in Fig. 1c. *Difficult* mazes have multiple obstacles that span more than half the width of the entire environment. They also include more complex versions of some of the *Standard* mazes, by manipulating the size of each obstacle in an environment. The "u-maze" from [11] was also incorporated into this group. The difficult mazes were deliberately designed to test the boundaries of the algorithms and to identify limitations.



(a) Obstacle (b) Standard Mazes (c) Difficult Mazes

Fig. 1. Training environments

The testing environments were divided into two categories: *Orientation* and *New*. *Orientation* testing environments were created by rotating the training mazes by 90° and without changing the overall structure of the obstacles. *New* testing environments have different obstacle configurations to the training environments. New features or environment characteristics, such as bottlenecks or repeated obstacles, were incorporated into this group. This allowed us to analyse whether the agents were able to learn advanced skills and further assess the extent of the generalisation. Both these categories were further subdivided into *Standard* and *Difficult* subcategories, as per the definition used for the training environments. An illustration of the *Orientation* environments are shown in Fig. 2a. Both the *Standard New* and *Difficult New* groups, depicted in Fig. 2b and c respectively, contain three mazes each. The "spiral-maze", a commonly used maze seen in [11], was incorporated into the difficult category.



(a) Orientation (b) Standard New (c) Difficult New

Fig. 2. Testing environments

3.3 Algorithms

Curriculum Learning. A curriculum was manually designed to enable agents to learn the task of finding distant targets across multiple sparse reward navigation environments (see Algorithm 1). This is difficult since agents cannot optimise a policy for any specific environments and when the environments are large, with multiple obstacles (the most difficult version of the task), the reward feedback is sparse.

The curriculum has been designed to act as a means of bypassing the sparse rewards problem. It also improves generalisation by exposing agents to a diverse set of environments during training.

Algorithm 1. Manually-Designed Curriculum

Input: Obstacle Environments O, Obstacle Max Scale $S_{obstacle}$, Maze Environments M, Environment Max Scale $S_{environment}$, Reward Threshold $R_{threshold}$, Number Consecutive Episodes $n_{consecutive}$

```
for i ← 1 to S_environment do
    Reset episode count
    r_average = 0
    repeat (for each episode)
        r_average ← average episodic reward from previous n_consecutive episodes
        Sample an obstacle environment from O
        Sample scale from {0, 1, 2, ..., S_obstacle}
        Sample angle from {0°, 45°, 90°, 135°, ..., 315°}
        Sample agent and target starting positions
    until r_average < R_threshold
    Reset episode count
    r_average = 0
    repeat (for each episode)
        r_average ← average episodic reward from previous n_consecutive episodes
        Sample a maze environment from M
        Sample agent and target starting positions
    until r_average < R_threshold
end for
```

Environment parameters are varied over time to control the difficulty of the task to ensure that the current task is never too difficult for the agent. The first parameter is the environment size: decreasing the size, while keeping the agent size and speed the same, decreases the sparsity of rewards since the goal and target are closer to each other in smaller environments. The second parameter is the obstacle configuration, which is varied through changing the number and size of obstacles: either single obstacles or multiple obstacles in a maze-like structure.

In the early stages of training, the environments are small and contain a single obstacle or none at all. This was achieved by assigning O, in Algorithm 1, to the obstacle environments in Fig. 1a. Agents are able to learn how to control themselves by navigating around the environment to nearby targets. When the average reward (over the past 5000 consecutive episodes) reaches a predefined threshold, the difficulty is increased. The first adjustment is to increase the size and number of obstacles, through randomly sampling maze environments from Fig. 1b and in Fig. 1c. When the agent reaches the same predefined reward threshold, the environment size is increased. This two-fold difficulty adjustment keeps occurring until the agent progresses to large maze environments with multiple obstacles. This ensures that the curriculum only progresses when the agent has succeeded in its current task.

Randomly sampling environments is an important aspect of the curriculum. It is also essential that the set of training environments is diverse and incorporates a wide array of obstacle configurations [7]. This deters agents from memorising the dynamics of any particular training environment, instead learning how

to navigate past arbitrary obstacles to find distant targets. This is analogous to supervised learning i.e. training on a diverse training set al.lows for a more generalised model that does not overfit to training data. Specifically, overfitting means memorising a policy that is optimised for the training environments, resulting in poor performance in the testing environments. Similarly, policy memorisation refers to a policy that optimises the dynamics of a particular environment by memorising actions that lead to success, resulting in poor performance even when subtle changes are made to the environment [41].

The maximum environment size ($S_{environment}$ in Algorithm 1) was carefully tuned to ensure that the task is a sparse rewards problem. This was verified by running an agent trained with policy gradient on the sparse reward function (+1 for finding the target), with no exploration strategy, and observing that it was not possible for it to find the target, after a large number of training steps [16].

Inspired by various other work [21,32,42], the last aspect of the curriculum attempts to bypass the sparse rewards problem by "densifying" the training environment. The starting locations of both the agent and target are randomised at the start of every episode. This means that the target is often close to the agent, resulting in frequent feedback that enables meaningful learning. This also encourages the agent to explore different parts of the environments.

Baseline Algorithms. We compare the performance of the curriculum to curiosity-driven exploration defined by Pathak et al. in [29]. This equips the agent with an intrinsic reward that allows it to explore the training environments by seeking "novel" states, thereby gaining an understand of the dynamics of the various environments. A reward is generated through a prediction error: agents are trained to predict the next state as well as actions taken in between states. In this way, the reward only captures surprising states that have come about directly as a result of the agents actions. Curiosity has shown promising generalisation capabilities in previous studies [5,29].

In the curiosity setting, the curriculum defined in Subsect. 3.3 is omitted i.e. the size of the environment is fixed at the largest configuration and a training maze is randomly sampled from Fig. 1b and c on each episode. Training also occurs under the dense reward setting with random target and agent starting locations.

The final approach combines the curiosity reward with the hand-crafted curriculum which we term "Hybrid": Agents are trained using the curriculum from Subsect. 3.3 and the reward function is augmented with a curiosity signal. Both algorithms have been shown to improve generalisation individually [5,6,12] and it was therefore necessary to investigate if there were any merits to combining them.

3.4 Experimental Design

We evaluated the performance of the curriculum by comparing it to curiosity-driven exploration [29] and a hybrid "curiosity-curriculum" approach. All policies

are represented by neural networks with Proximal Policy Optimisation (PPO) [35] being used as an optimiser. PPO is robust [35] and requires lesser hyperparameter tuning when compared to similar methods [5]. However, an arbitrary policy gradient method could have been: the focus of this work is rather on the comparison of different training methods so consistency in the optimisation method is more important.

We defined baseline hyperparameters for each algorithm by training agents in the easiest version of the tasks. The hyperparameters were then carefully tuned and optimised by observing the training process and then tweaking the relevant parameters as required. The networks have two hidden layers, each with 256 units. The swish activation function [30] was used. The learning rate and entropy coefficient were fixed at 3.0×10^{-4} and 0.01 respectively, along with a batch size of 256 and a buffer size of 5120. All agents were trained for 20 million training steps. This was carefully tuned to ensure that the agents had sufficient time to learn. However, it was observed that agents tended to overfit to the training environments when the steps were too high [42]. After tuning the hyperparameters independently for each algorithm, the agents were trained on a cluster of machines on the Centre for High Performance Computing [1], using the Unity ML-Agents platform [18]. The polices for each algorithm were then evaluated against each other. The codebase and further details are available at https://github.com/AsadJeewa/Learning-to-Generalise-in-Sparse-Reward-Navigation-Environments.

4 Results and Discussion

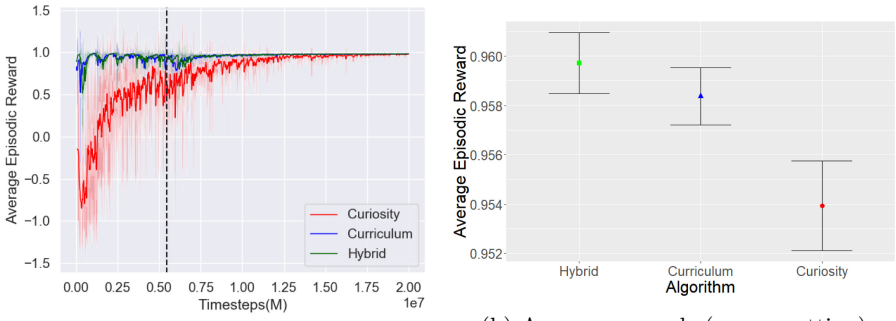
Analysis is performed in three stages: the first stage compares the training performance of each algorithm. Since the agents were trained under a dense rewards setting, with randomised agent and target starting positions for each episode, it is necessary to evaluate the algorithms under a sparse reward setting. This was achieved by positioning the agent and target at distant locations in every training environment, fixed at points that make the task as difficult as possible.

We perform a critical evaluation of the generalisability of each algorithm in the unseen testing environments. The last stage performs trajectory analysis to understand the strengths and limitations of each algorithm. It provides insight into the intricacies of how agents move within different environments.

4.1 Training Performance

The training curves are depicted in Fig. 3a i.e. the average episodic reward of the agents over time, with a smoothing factor of 0.2. For each algorithm, we performed five independent runs and computed the mean learning curve and standard deviation. Twenty independent instances of the environment were used for more efficient data collection during training.

The dashed line depicts the point at which both the curriculum and hybrid agents progressed to the final lesson, which corresponds to the training environments of the curiosity-driven agent.



(a) Learning curves during training

(b) Average rewards (sparse setting)

Fig. 3. Training performance for all algorithms

Figure 3 highlights the benefits of using the curriculum. The learning curve never drops significantly since the agents' task is never too difficult. The curriculum advances quickly in the early stages of training when the task is easier. The sudden drops in reward are indicative of points at which the task is made more difficult but the fact that the curve peaks very quickly thereafter, indicates that knowledge is being transferred between tasks. In all runs, it was noted that the curriculum agent converged significantly faster than the curiosity agent.

A major benefit of the curriculum is that there is no reward shaping necessary. This is due to the manner in which the curriculum was designed that ensures that the agents always receive sufficient reward feedback during training. We performed an empirical investigation into various different shaped rewards and found no performance improvements. Rather, the motivations of the agents became polluted [9,26]. For example, when an agent was rewarded for moving closer to the target, it lacked the foresight to move past obstacles. Shaping rewards also resulted in more specialist policies that work well in some environments, but poorly in others. Reward shaping also requires additional information which may not be available in the real-world.

The curiosity curve shows rewards slowly increasing as training progresses. The hybrid training curve is very similar to the curriculum agent. When the curiosity strength was varied, the curves still followed a similar pattern. This indicates that the curiosity rewards had little effect on the training process when coupled with the curriculum.

Figure 3b illustrates that, for all algorithms, the agents were able to efficiently find the target in all training environments, under the sparse reward setting. All algorithms have an average reward that approaches a maximum possible reward of +1. These results act as a validation of each algorithm since it indicates that all agents have obtained sufficient knowledge of the task and are able to find targets across a diverse set of mazes. This allowed us to perform a fair comparison of the generalisation capabilities of each algorithm in the testing environments. Error bars are depicted with a confidence interval of 95%.

4.2 Generalisability

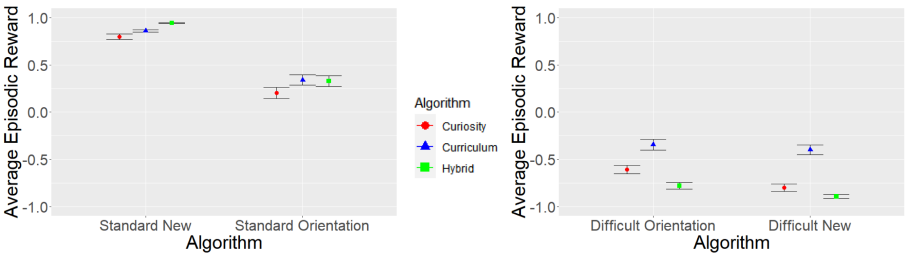
The best performing training run from Subsect. 4.1 was selected for each algorithm. The average reward was then analysed for each of the different groups of testing environments. Each algorithm was run for 1000 episodes, with a random testing environment being sampled at the start of the episode, from the corresponding testing group. This is necessary due to the stochastic nature of the polices: the agents sometimes succeed and fail in the same testing environment. This results in vastly different episodic rewards and a large number of episodes is therefore necessary to stabilise the average rewards.

When analysing the results, there are certain important considerations that need to be made. The performance of each algorithm is often different i.e. agents succeed and fail in different testing environments. There are instances when one algorithm enabled agents to navigate to the target in a short time, but another resulted in agents only finding the target after a large number of episode steps or never at all. We wish to investigate this phenomenon further in future work.

The task is not trivial since it as analogous to placing a human or vehicle in a new environment and only equipping them with information about its current location, destination and the ability to "see" what's around it. It does not have any knowledge of the dynamics of the environment that it is placed in. This means that some "exploration" is necessary and it is expected that agents will move into obstacles as they try to advance towards the goal. It is not possible to solve the generalisation problem completely: it was not expected that the agents would obtain expert performance in the testing environments. The goal is rather to transfer some knowledge that can be reused in the environments.

The policies are used "as-is" and there is no fine-tuning for any of the testing environments, as is the case in other studies [29]. It is definitely possible to improve the results in each testing environment by fine-tuning the policy though that is not the aim of this study. This work rather investigated the extent to which the learned policy generalised.

Lastly, the sample size in the testing environment groups is fairly small. There are only three environments in some groups. In future work, we wish to investigate whether the results hold when increasing the size of the groups.



(a) Standard Mazes (b) Difficult Mazes

Fig. 4. Generalisability in the maze testing environments

The average episodic rewards are in the range $[-1, 1)$. A successful run is one in which agents are able to navigate to the target. The faster an agent finds the target, the higher the reward it receives. An average reward approaching one therefore indicates that the agents successfully found the targets on all runs. A score below one indicates that on most runs, the agents were unable to find the target, across all environments, with zero representing an inflection point.

The results highlight an expected gap between training and testing performance. However, they also indicate that some generalisation has taken place.

Standard Mazes. The experiments that we conducted indicate good performance for all algorithms in the standard mazes. Figure 4a illustrates that the algorithms preformed similarly in the *Standard New* environments. Notably, all the agents were able to consistently navigate to the target in all three environments. This is promising since the obstacle configurations are completely different to the training environments. This indicates that the policy is robust and generalises well (in these environments). On average, the hybrid agents found the targets marginally faster.

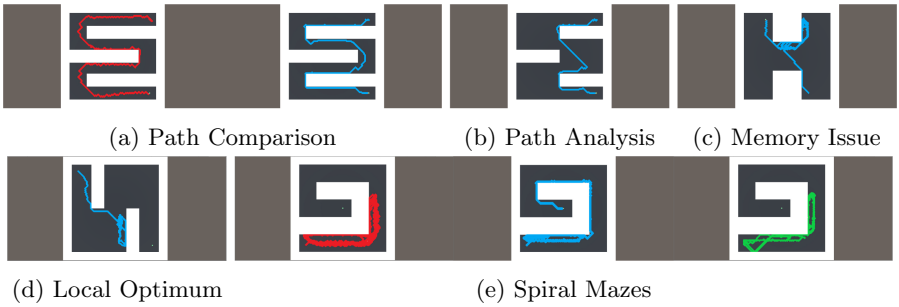
The *Standard Variation* results depict that all algorithms are able to succeed on most runs. The curriculum agent was marginally the most successful across the six environments but the performance is once again similar for all algorithms. It is encouraging that the agents succeed in some environments however, we theorise that the results can be improved by fine-tuning the set of training and testing environments, or procedurally generating training mazes to improve the robustness of the policies.

Difficult Mazes. While the results of the algorithms in the standard mazes showed similar performance, the agents trained using the curriculum performed best in the difficult mazes.

The *Difficult Orientation* results in Fig. 4b indicate that the agents weindicate that the agents were re unable to find the target on most runs. However, some transfer has taken place. The curriculum obtained the highest average reward: the result is statistically significant under a 95% confidence interval. Interestingly, both the curriculum and hybrid agents succeeded in two of the five environments but the hybrid agent took significantly longer to find the targets. The hybrid agent is the worst performing algorithm; this indicates that generalisability decreases significantly as the difficulty of the environments are increased. The performance of the curiosity-driven agent showed limited transfer to the testing environments with agents only succeeding in one environment.

Difficult New experiments show the least transfer, as expected. The curriculum agent is once again the most successful. The nature of the environments mean that agents are able to find the targets on some runs, though not consistently. The most promising result was that the curriculum agent was the only algorithm that succeeded in the "spiral-maze" [11] depicted in Fig. 2c.

4.3 Trajectory Analysis



(a) Path Comparison (b) Path Analysis (c) Memory Issue

(d) Local Optimum (e) Spiral Mazes

Fig. 5. Walkthrough trajectories

We performed trajectory analysis by analysing the movement patterns of the trained agents across the different environments, as per [32]. It was often observed that the curriculum agents tended to move in a more directed manner than the curiosity-driven agents. The curriculum agents also tended to "stick" to the walls for a longer time, using them to guide it to the target. An example of this is depicted in Fig. 5a. The trail of a curiosity-driven agent is shown in red, on the left, and that of the curriculum agent is in blue. There is further proof of this in Fig. 5b. This figure also highlights common behaviour of the curriculum agents: they initially attempted to move directly towards the goal, along the shortest possible path, but when the agents detected an obstacle, they adapted to move around it. The highlights the robustness of the policy.

A limitation of all the algorithms is that it was sometimes observed that the agents repeatedly move along a similar path and only make slight advancements towards the target, over a long period of time. However, the agents often still find their way to the target, as shown in Fig. 5c. In an attempt to alleviate this problem, we would like to look into different methods for increasing the "memory" of the agents. The number of stacked observations could be increased, so that agents can "remember" more of their previous failures, or recurrent architectures could be used.

Figure 5d shows an example of an environment in which the curriculum agent failed to find the target. The agent was progressing towards the target but then got stuck in a local optimum and kept repeating the same actions, until the maximum episode steps was reached. It is possible that the agents would eventually have found its way to the target. This result points to some memorisation in the policy. We theorise that improving the memory of the agent would also alleviate this problem.

The most promising result is shown in Fig. 5e. The spiral maze is difficult because the agent needs to learn a very specific trajectory in order to find the target. The curriculum agent was the only agent that succeeded in this environment. This further highlights the robustness of the curriculum: it was able

to continuously adapt its actions as it observed the environment. The curiosity agents, depicted on the left in Fig. 5e and the hybrid agents (on the far right) both got stuck in local optima and failed to reach the target.

5 Conclusions and Future Work

We have designed a training curriculum that improves generalisation in sparse reward navigation environments. It was evaluated against a curiosity-based agent [29] and a hybrid of the two algorithms, in a suite of manually-designed navigation environments.

The curriculum agents showed the most promising generalisation results. Agents were able to find targets in more testing environments, including some with completely new environment characteristics. There are certain environments when curiosity performed better than the curriculum agent but the performance of the agents were more erratic i.e. they sometimes performed excellently and sometimes very poorly within the same environments. Curriculum learning proved to be more a robust approach. It also resulted in decreased training times and eliminated the need for any reward shaping.

Combining curiosity with the curriculum provided no meaningful benefits. The training performance was very similar to the curriculum agent and it exhibited inferior policy generalisation in the difficult maze testing environments.

There are limitations to the curriculum, as indicated by the generalisation gap between the training and testing environments. Agents sometimes get stuck in local optimums and also repeated the same movement patterns in an episode. There is some memorisation occurring since the agents perform excellently in the training environments and struggle in some testing environments. However, the results are promising, since it shows clear evidence of knowledge transfer to unseen environments.

In future work, we propose further increasing the diversity in the training environments and fine-tuning the curriculum to further improve the results. We also wish to investigate the effects of increasing the memory of agents to deter them from repeating trajectories in testing environment. Another interesting direction is to perform further large scale analysis of the algorithms by increasing the number of testing environments, either manually or by procedurally generating them [7].

References

1. Centre for high performance computing. https://www.chpc.ac.za/
2. Andrychowicz, M., et al.: Hindsight experience replay. In: Advances in Neural Information Processing Systems, pp. 5048–5058 (2017)
3. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: an evaluation platform for general agents. J. Artif. Intell. Res. **47**, 253–279 (2013)
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 41–48. Association for Computing Machinery, Montreal (2009). https://doi.org/10.1145/1553374.1553380

5. 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
6. Burda, Y., Edwards, H., Storkey, A., Klimov, O.: Exploration by random network distillation. arXiv preprint arXiv:1810.12894 (2018)
7. Cobbe, K., Hesse, C., Hilton, J., Schulman, J.: Leveraging procedural generation to benchmark reinforcement learning. arXiv preprint arXiv:1912.01588, p. 27 (2019)
8. Cobbe, K., Klimov, O., Hesse, C., Kim, T., Schulman, J.: Quantifying generalization in reinforcement learning. arXiv preprint arXiv:1812.02341, p. 8 (2018)
9. Devlin, S.M., Kudenko, D.: Dynamic potential-based reward shaping (2012). http://eprints.whiterose.ac.uk/75121/
10. Duan, Y., Chen, X., Houthooft, R., Schulman, J., Abbeel, P.: Benchmarking deep reinforcement learning for continuous control. In: International Conference on Machine Learning, pp. 1329–1338 (2016). http://proceedings.mlr.press/v48/duan16.html, ISSN: 1938–7228 Section: Machine Learning
11. Florensa, C., Held, D., Geng, X., Abbeel, P.: Automatic goal generation for reinforcement learning agents. In: International Conference on Machine Learning, pp. 1515–1528 (2018). http://proceedings.mlr.press/v80/florensa18a.html. ISSN: 1938–7228 Section: Machine Learning
12. Florensa, C., Held, D., Wulfmeier, M., Zhang, M., Abbeel, P.: Reverse curriculum generation for reinforcement learning. arXiv:1707.05300 [cs] (2018)
13. Frans, K., Ho, J., Chen, X., Abbeel, P., Schulman, J.: Meta learning shared hierarchies. arXiv:1710.09767 [cs] (2017)
14. Hacohen, G., Weinshall, D.: On the power of curriculum learning in training deep networks. arXiv:1904.03626 [cs, stat] (2019)
15. Hussein, A., Elyan, E., Gaber, M.M., Jayne, C.: Deep reward shaping from demonstrations. In: Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), pp. 510–517. IEEE (2017)
16. Jeewa, A., Pillay, A., Jembere, E.: Directed curiosity-driven exploration in hard exploration, sparse reward environments. In: Davel, M.H., Barnard, E. (eds.) Proceedings of the South African Forum for Artificial Intelligence Research, Cape Town, South Africa, 4–6 December 2019, CEUR Workshop Proceedings, vol. 2540, pp. 12–24. CEUR-WS.org (2019). http://ceur-ws.org/Vol-2540/FAIR2019_paper_42.pdf
17. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.G.: Self-paced curriculum learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (2015). https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9750
18. Juliani, A., et al.: Unity: a general platform for intelligent agents. arXiv:1809.02627 [cs, stat] (2018)
19. Juliani, A., et al.: Obstacle tower: a generalization challenge in vision, control, and planning. arXiv:1902.01378 [cs] (2019)
20. Justesen, N., Torrado, R.R., Bontrager, P., Khalifa, A., Togelius, J., Risi, S.: Illuminating generalization in deep reinforcement learning through procedural level generation. arXiv:1806.10729 [cs, stat] (2018)
21. Kang, B., Jie, Z., Feng, J.: Policy optimization with demonstrations. In: International Conference on Machine Learning, pp. 2469–2478 (2018). http://proceedings.mlr.press/v80/kang18a.html
22. Karpathy, A., van de Panne, M.: Curriculum learning for motor skills. In: Kosseim, L., Inkpen, D. (eds.) AI 2012. LNCS (LNAI), vol. 7310, pp. 325–330. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30353-1_31

23. Khalifa, A., Bontrager, P., Earle, S., Togelius, J.: PCGRL: Procedural content generation via reinforcement learning. arXiv:2001.09212 [cs, stat] (2020)
24. Matiisen, T., Oliver, A., Cohen, T., Schulman, J.: Teacher-student curriculum learning. In: IEEE Transactions on Neural Networks and Learning Systems, pp. 1–9 (2019). https://doi.org/10.1109/TNNLS.2019.2934906
25. Narvekar, S., Stone, P.: Learning curriculum policies for reinforcement learning. arXiv:1812.00285 [cs, stat] (2018)
26. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. ICML **99**, 278–287 (1999)
27. Oudeyer, P.Y., Kaplan, F.: What is intrinsic motivation? A typology of computational approaches. Front. Neurorobotics **1**, 6 (2009)
28. Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., Song, D.: Assessing generalization in deep reinforcement learning. arXiv:1810.12282 [cs, stat] (2019)
29. Pathak, D., Agrawal, P., Efros, A.A., Darrell, T.: Curiosity-driven exploration by self-supervised prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2017), pp. 488–489. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPRW.2017.70, http://ieeexplore.ieee.org/document/8014804/
30. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv:1710.05941 [cs] (2017)
31. Ravishankar, N.R., Vijayakumar, M.V.: Reinforcement learning algorithms: survey and classification. Indian J. Sci. Technol. **10**(1), 1–8 (2017). https://doi.org/10.17485/ijst/2017/v10i1/109385, http://www.indjst.org/index.php/indjst/article/view/109385
32. Savinov, N., Dosovitskiy, A., Koltun, V.: Semi-parametric topological memory for navigation. arXiv:1803.00653 [cs] (2018)
33. Savinov, N., et al.: Episodic curiosity through reachability. arXiv:1810.02274 [cs, stat] (2019)
34. Schmidhuber, J.: POWERPLAY: training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. arXiv:1112.5309 [cs] (2012)
35. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 [cs] (2017)
36. Suay, H.B., Brys, T.: Learning from demonstration for shaping through inverse reinforcement learning, p. 9 (2016)
37. 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)
38. Sutton, R.S., Barto, A.G.: Reinforcement Learning, Second Edition: An Introduction. MIT Press, Cambridge (2018). google-Books-ID: uWV0DwAAQBAJ
39. Svetlik, M., Leonetti, M., Sinapov, J., Shah, R., Walker, N., Stone, P.: Automatic curriculum graph generation for reinforcement learning agents. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (2017). https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14961
40. Tessler, C., Givony, S., Zahavy, T., Mankowitz, D.J., Mannor, S.: A deep hierarchical approach to lifelong learning in minecraft. arXiv:1604.07255 [cs] (2016)
41. Ye, C., Khalifa, A., Bontrager, P., Togelius, J.: Rotation, translation, and cropping for zero-shot generalization. arXiv:2001.09908 [cs, stat] (2020)
42. Zhang, C., Vinyals, O., Munos, R., Bengio, S.: A study on overfitting in deep reinforcement learning. arXiv:1804.06893 [cs, stat] (2018)