The Role of Diverse Replay for Generalisation in Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In reinforcement learning (RL), key components of many algorithms are the exploration strategy and replay buffer. These strategies regulate what environment data is collected and trained on and have been extensively studied in the RL literature. In this paper, we investigate the impact of these components in the context of generalisation in multi-task RL. We investigate the hypothesis that collecting and training on more diverse data from the training environment will improve zero-shot generalisation to new environments/tasks. We motivate mathematically and show empirically that generalisation to states that are "reachable" during training is improved by increasing the diversity of transitions in the replay buffer. Furthermore, we show empirically that this same strategy also shows improvement for generalisation to similar but "unreachable" states and could be due to improved generalisation of latent representations.

1 Introduction

An important aspect of reinforcement learning research is investigating methods that can generalise from the training conditions to various new and unseen deployment scenarios. In this paper, we refer to generalisation as the goal of transferring a policy trained in a set of training environments/tasks to a set of testing environments/tasks. Popular approaches in the current literature try to improve generalisation performance by increasing the similarity or explicitly tackling the differences between the training and testing tasks (Kirk et al., 2023). What these methods have in common is that they focus on the differences between the distributions of tasks, rather than the specific data (states and actions) encountered in those tasks.

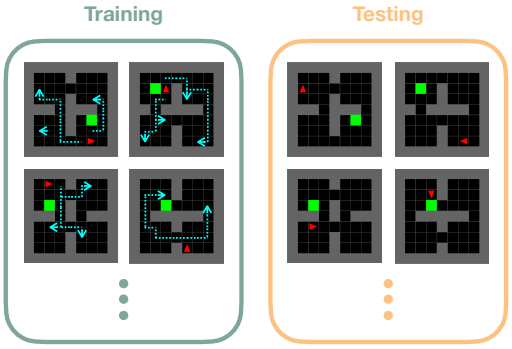


Figure 1: Example of a generalisation problem to various 4-room configurations/tasks. With sufficient exploration (cyan) in the training configurations, the agent can encounter states that will likely help it generalise to the testing configurations/tasks.

Key components of many reinforcement learning algorithms are the strategies to collect, store and sample data on which to train. For this reason, extensive literature exists on the problems of exploration (collect) (Amin et al., 2021), replay buffers (store) (Isele & Cosgun, 2018; Bruin et al., 2016) and replay sampling strategies (sample) (Schaul et al., 2016; Zhang & Sutton, 2017; Andrychowicz et al., 2017). It is common in generalisa-

Submitted to 16th European Workshop on Reinforcement Learning (EWRL 2023). Do not distribute.

tion approaches to treat the exploration and replay storing and sampling strategies in the same way one would in singleton RL: only explore and replay new transitions to the extent necessary to learn the current task.

In this paper, we investigate the role of the exploration strategy and the replay buffer on generalisation performance beyond the singleton perspective. We theorise that a more diverse strategy of collecting and sampling training data will improve performance in certain generalisation scenarios (see Figure 1). Our key contributions are the following:

- We define a measure of *reachability* that will allow us to prove that more diverse exploration/replay strategies lead to improved generalisation performance under certain restrictive conditions.
- We empirically show that generalisation to *reachable* starting states benefits from more diverse exploration/replay strategies.
- We show empirically that generalisation to similar but *unreachable* starting states also benefits from more diverse exploration/replay strategies.
- We analyse the latent representations induced by more diverse exploration/replay strategies and find they are likely responsible for the improved generalisation performance.

2 Background

The goal of reinforcement learning is to optimise decision-making in a Markov decision process (MDP). A Markov decision process is often denoted as a 6-tuple $\mathcal{M} = (S, A, T, R, p_0, \gamma)$ where S is a set of states called the state space, A a set of actions called the action space, $T : S \times A \to \mathcal{P}(S)$ a transition probability function (where $\mathcal{P}()$ denotes the probability function), $R : S \times A \to \mathbb{R}$ a reward function, $p_0 : \mathcal{P}(S)$ a distribution of initial states and $\gamma \in [0, 1)$ a discount factor. In reinforcement learning, the goal is to find the policy function $\pi : S \to \mathcal{P}(A)$ that maximises in expectation the discounted sum of rewards: $\pi^* := \arg\max_\pi \mathbb{E}_\pi\big[\sum_{t=0}^{\infty} \gamma^t r_t\big]$ where π^* is the optimal policy and $\mathbb{E}_\pi$ denotes the expectation over the Markov chain $(s_0, a_0, r_0, s_1, a_1, r_1, ...)$ induced by the policy π, initial state distribution p_0, transition function T and reward function R (Akshay et al., 2013). The steady-state distribution $\rho^\pi : \mathcal{P}(S)$ of the Markov chain induced by policy π in MDP $\mathcal{M}$ defines the proportion of time spent in each state as the number of transitions within $\mathcal{M}$ goes to infinity. This distribution can be proven to be unique under certain conditions of the policy π and MDP $\mathcal{M}$ (Konstantopoulos, 2009).

In Q-learning the goal is to learn the optimal Q-function $Q^*(s, a) := \mathbb{E}_{\pi^*}\big[\sum_{t=0}^{\infty} \gamma^t r_t\big|_{a_0=a}^{s_0=s}\big]$ which represents the expected future discounted return given a current state s and action a. Deep Q learning (DQN) is a popular deep reinforcement learning algorithm that trains a neural network q_ϕ *online* to satisfy the optimal Bellman equation $Q^*(s, a) = r(s, a) + \gamma \max_{a'} Q^*(s', a')$. It achieves this by minimising the mean squared error between the left-hand side and right-hand side of the optimal Bellman equation $\min_\phi \mathbb{E}\big[(r + \gamma \max_{a'} q_\phi(s', a') - q_\phi(s, a))^2\big| < s, a, r, s' >\sim \mathcal{D}\big]$ over a dataset of transitions $\mathcal{D}$ called the replay buffer. Typically, the replay buffer has fixed size and stores transitions collected during training in a first-in-first-out (FIFO) principle. Transitions are commonly collected with an ϵ-greedy exploration policy: $\pi_\epsilon(a|s) = (1 - \epsilon)\pi(a|s) + \epsilon\frac{1}{|A|}$, which selects random actions with probability ϵ or a 'greedy' action according to $\pi(a|s)$ with probability $1 - \epsilon$. The inherent generalising capabilities of the neural network in DQN allow the agent to learn even in continuous state spaces (where it is impossible to collect experience for every possible state).

2.1 Contextual Markov Decision Process

A contextual MDP (CMDP) (Kirk et al., 2023) is a specific type of MDP where the state space S is factored into an underlying state space S' and a *context space* C: $S = S' \times C$. The context $c \in C$ is drawn at the start of an episode and fixed thereafter. The context can be thought of as defining the environment/task an agent has to solve in a multi-task setup. A set of contexts then defines a set of environments/tasks the agent has to solve. In this paper, we will assume the context c is fully observable, which means the state $s_0 \in S$ at the start of an episode fully determines the task/environment the agent has to solve that episode. An example can be found in Figure 1 where the

agent's starting location and orientation, the goal location and the location of the doorways (referred to as the 4-room *topology*) fully determines the agent's context/task, it has to solve in that episode.

2.2 Zero-shot Policy Transfer

It is possible to define a generalisation setting by defining an MDP $\mathcal{M}|_C$ with a context space C and context sets $C_{train}, C_{test} \subseteq C$ sampled during training and testing. For example, in a typical generalisation setting the agent is tested in the MDP $\mathcal{M}|_{C_{test}}$ on contexts C_{test} that are unseen (held-out) during training ($C_{train} \cup C_{test} = \emptyset$).

In particular, we consider in this paper the zero-shot policy transfer (ZSPT) problem class as defined in Kirk et al. (2023). A ZSPT problem is defined through a choice of training and testing context sets C_{train}, C_{test} with the objective of maximising return in the testing MDP $\mathcal{M}|_{C_{test}}$, whilst only being allowed to train on $\mathcal{M}|_{C_{train}}$. The agent is expected to perform *zero-shot* generalisation in the testing contexts, so without a fine-tuning or adaptation period. Because we assume the context c is fully observable, and therefore determined by the starting state $s_0 \in S$, we will refer to the context and starting state interchangeably ($\mathcal{M}|_{C_{train}}$ becomes $\mathcal{M}|_{S_0^{train}}$ where S_0^{train} is the set of starting states $s_0 \in S_0^{train}$ that correspond with C_{train}).

3 Related Work

Zero-shot generalisation in contextual MDPs is a well-studied area of research with a diverse range of different approaches. One branch of research is focused on increasing the similarity between the training and testing context sets. Some examples of this are the use of (visual) data augmentations (Raileanu et al., 2021; Lee et al., 2020; Zhou et al., 2021) or (dynamical) domain randomisation (Tobin et al., 2017; Sadeghi & Levine, 2017; Peng et al., 2018) to increase the complexity of the training contexts, with the aim to subsume the testing contexts in this increased training complexity. Another branch of research is focused on explicitly handling the (expected) differences between training and testing. This includes approaches that use inductive biases (Raileanu et al., 2021; Higgins et al., 2017; Zambaldi et al., 2018, 2019; Kansky et al., 2017; Wang et al., 2021; Tang et al., 2020; Tang & Ha, 2021) or non-RL specific regularisation (Cobbe et al., 2019; Ada et al., 2019; Tishby & Zaslavsky, 2015; Igl et al., 2019; Lu et al., 2020; Eysenbach et al., 2021) to overcome the challenges induced by the variations between training and testing.

All the above-mentioned works use techniques to improve generalisation that are not necessarily specific to RL. For example, most of the work applies ideas from supervised learning focused on the differences between the training and testing datasets, to the differences between the training and testing *context sets*, rather than the data collected from those contexts. Others (Igl et al., 2021; Lyle et al., 2022) do look into RL-specific problems with generalisation but are mostly focused on the effects non-stationary data or temporal difference update rules have on the generalisation capacity of neural networks. The work most similar to ours is a workshop paper (Jiang et al., 2022) that proposes a new exploration algorithm motivated by a similar argument as used in our paper. Our work differs in that we don't propose a new exploration method, but instead investigate how and why exploration and replay buffer strategies affect generalisation performance in certain types of contextual MDPs. As such, in Section 5.5 we observe primarily a representation learning effect which seems in contrast to some of the results in (Jiang et al., 2022) that suggest RL-specific effects beyond representation learning.

4 Method

In the multi-task RL literature, exploration and experience replay is often treated as one would do in singleton RL: only explore and replay new transitions enough to learn the current task. In this section, we will define a generalisation setting to *reachable states* and use it to prove that the singleton perspective on exploration and experience replay can be sub-optimal for generalisation in multi-task RL.

We define a reachable state s_r as a state for which there exists a policy whose probability of encountering s_r during training is non-zero. This gives us the following definition of a set of reachable states S_r in an MDP $\mathcal{M}|_{S_0^{train}}$.[1]

Definition 1. *The set of reachable states $S_r(\mathcal{M}|_{S_0^{train}})$ consists of all states s_r for which there exists a sequence of actions that give a non-zero probability of ending up in s_r when performed in the MDP $\mathcal{M}|_{S_0^{train}}$.*

Corollary 1.1. *Any state s' that is reachable from a state $s \in S_r(\mathcal{M}|_{S_0^{train}})$ in the reachable set, has to be itself in the reachable set: $s' \in S_r(\mathcal{M}|_{S_0^{train}})$.*

Corollary 1.1 entails that you cannot leave the reachable set $S_r(\mathcal{M}|_{S_0^{train}})$ through interaction with the environment.

Using this definition of reachable states we can define a particular instance of the ZSPT problem. We define *ZSPT to reachable states* as a ZSPT problem where the initial states during testing S_0^{test} are part of the set of reachable states during training $S_0^{test} \subset S_r(\mathcal{M}|_{S_0^{train}})$. This particular instance of the ZSPT problem has an interesting property.

Corollary 1.2. *An optimal policy π that achieves maximal return from any state in the reachable state space $S_r(\mathcal{M}|_{S_0^{train}})$, will have optimal performance in the ZSPT to reachable states problem setting.*

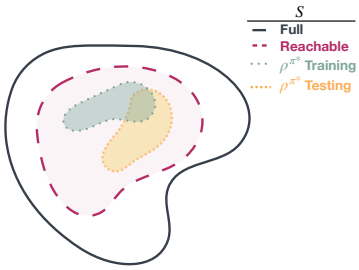


Figure 2: Illustrative example of the state space covered by the set of reachable states $S_r(\mathcal{M}|_{S_0^{train}})$ and the steady state distributions ρ^{π^*} of the optimal policy during training and testing in a ZSPT to reachable states problem. It illustrates how a broader optimal coverage of the reachable state space during training can improve test time performance.

Recall that performance in the ZSPT problem is defined as the performance in the testing MDP $\mathcal{M}|_{S_0^{test}}$, which in the case of ZSPT to reachable states, has a state space that consists only of reachable states (due to Corollary 1.1) It follows naturally that an policy optimal on the entire reachable state space $S_r(\mathcal{M}|_{S_0^{train}})$ also has to be optimal in $\mathcal{M}|_{S_0^{test}}$.

So, in ZSPT to reachable states it is better to explore and replay transitions from all over the reachable state space: even though some transitions might be irrelevant during training (not part of the steady-state distribution of the optimal policy), they could be crucial during testing (see Figure 2). This is in contrast with singleton RL, where it can be (correctly) assumed that states/transitions not encountered by the optimal policy during training, will not be encountered by the optimal policy during testing.

The above holds for an idealised scenario where it is possible to learn an exact optimal policy for every reachable state. However, with function approximation in large/continuous state-action spaces, the exact optimal solution might not always be achievable. Nevertheless, a more diverse exploration and replay of the reachable state space will likely improve generalisation performance.

This insight does not directly carry over to the general ZSPT problem: if testing is no longer constrained to reachable states $S_r(\mathcal{M}|_{S_0^{train}})$, then training on a larger part of the reachable state space might not improve test performance. However, with the inherent generalisation capabilities of neural networks, we still expect a more diverse replay buffer to benefit generalisation to unreachable (but similar) states through learning better generalising latent representations.

5 Experiments

In this section, we empirically evaluate the role of a diverse replay buffer on generalisation to new starting states. We first validate that increased diversity of the replay buffer (through exploration and replay buffer parameters) in the function approximation setting will lead to a policy that is optimal on a larger part of the reachable state space. We then evaluate the effect a more diverse replay buffer has

[1] Similar to a definition of reachability for Markov chains in Velasquez et al. (2023)

on generalisation performance to both reachable and unreachable starting states. Lastly, we analyse the latent representations of the agents trained with different levels of replay buffer diversity.

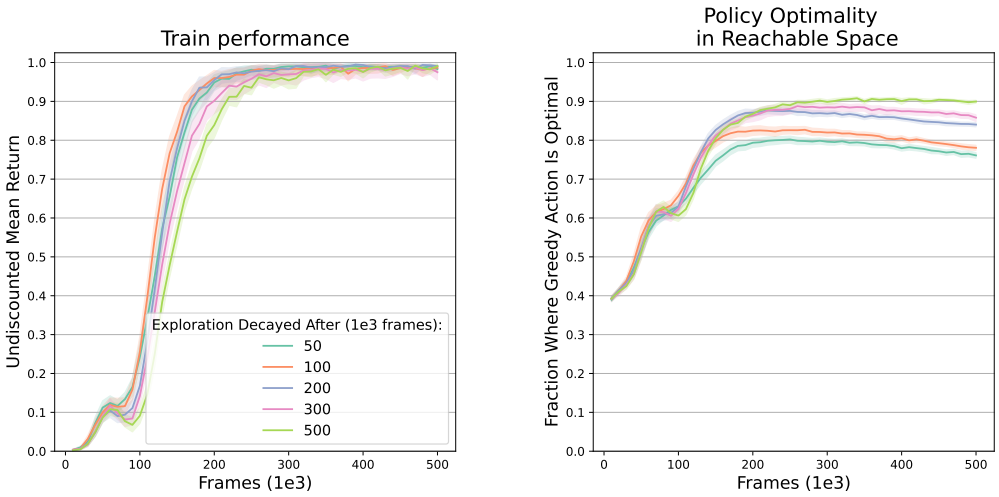
5.1 4-Room Grid World

Our investigation is performed in a small 4-room grid world (Figure 1) adapted from the `FourRooms` environment from the MiniGrid benchmark (Chevalier-Boisvert et al., 2018). The state space (including the task/context space) is fully observable and has three actions: move forward, turn left, and turn right. Between training and testing the only differences are the starting states s_0 (or equivalently, tasks), which vary along the following dimensions: the 4-room topology, the starting location of the agent, the direction the agent is facing and the location of the goal. During training, the agent will encounter a fixed number of different start states and will be tested on a number of unseen starting states (generated from the same distribution). See Appendix A for more details on the environment.

In this environment, reachability is regulated through variations in the goal location and topology. If two states share their topology and goal location, then they are both reachable from one another. Conversely, if two states differ in either the topology or goal location, they are unreachable.

5.2 Policy Optimality in Reachable Space

In this experiment, training is done on a fixed set of 40 starting states which differ in their agent location, agent direction, goal location and topology. The agent is trained using DQN with linearly decaying ϵ-greedy exploration and a FIFO replay buffer. In order to increase the diversity of transitions in our replay buffer, we vary the amount of exploration performed. Note that increasing the amount of exploration does not necessarily lead to a consistently more diverse replay buffer, as the more diverse exploratory data might drop out of the first-in first-out buffer. To avoid this, we use a replay buffer size equal to the total length of training (500.000 environment steps). Therefore, the replay buffer never drops any experiences and varying the amount of exploration induces different ratios of diversity in the buffer. See Appendix B for more training details.



(a) Training performance

(b) Policy Optimality

Figure 3: Left: the mean undiscounted greedy return during training for different levels of exploration. Right: fraction of reachable space where the greedy policy is optimal. Shown are the mean and a 95% confidence interval over 100 seeds.

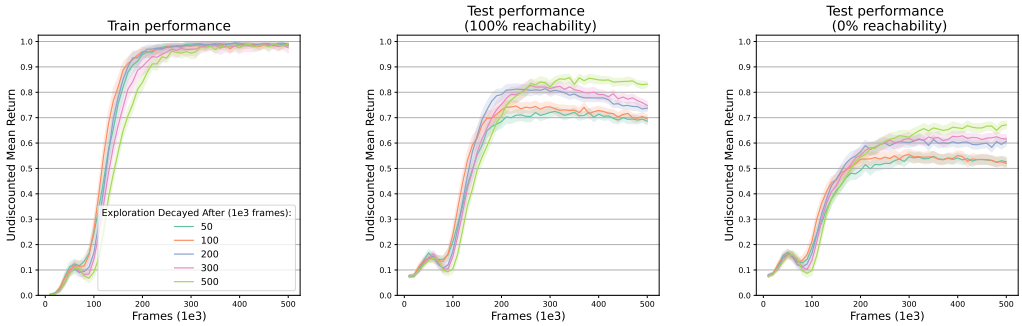
We compare five different linearly decaying $\epsilon-$greedy strategies (decayed after 50.000, 100.000, 200.000, 300.000 and 500.000 environment steps) and how they affect training performance and the fraction of reachable space for which our greedy policy is optimal. The results can be found in Figure 3. Comparing Figures 3a with 3b shows a clear trend that more exploration, and therefore

more diverse data in the replay buffer, leads to a higher fraction of reachable state space where the policy is optimal whilst training performance is mostly unaffected.

5.3 Generalisation To Unseen Starting States

For evaluating the generalisation capabilities, the agent is trained on the same 40 starting states as the previous experiment and tested on 40 new and unseen starting states. Two testing sets are constructed to evaluate the agent's performance on both reachable and unreachable starting states.

The 100% reachable test set is constructed by taking every training start state and changing only the agent location and agent direction (keeping the topology and goal location the same). The 0% reachable (unreachable) test set is constructed by taking every start state in the 100% reachable test set and changing the topology to one not encountered during training (keeping the agent location, direction and goal location the same as the 100% reachable set).



(a) Training performance (b) Testing performance (c) Testing performance

Figure 4: The mean undiscounted greedy return for different levels of exploration. Left: during training. Middle: when testing on a 100% reachable test set. Right: when testing on a 0% reachable test set. Shown are the mean and a 95% confidence interval over 100 seeds.

Figure 4 shows the performance during training, testing on 100% reachable start states and testing on 0% reachable start states. Figure 4a shows the same training performance as can be found in Figure 3a. Comparing Figure 4a with 4b, there is a clear trend that a more diverse replay buffer leads to better generalisation to reachable starting states (higher test performance whilst training performance is largely unaffected). Comparing this with Figure 3b shows a correlation between a higher fraction of reachable space where the policy is optimal and higher generalisation performance to reachable states. Figure 4c shows that this trend even holds for generalisation to unreachable starting states.

5.4 Uniform Replay over State-action Space

In order to further solidify the role of a diverse replay buffer, the experiment is repeated but with a S, A-*uniform* replay buffer. An S, A-uniform replay buffer samples its mini-batches uniformly over the state-action space $(S \times A)_{\mathcal{D}} \subseteq S \times A$ supported in the buffer $(S \times A)_{\mathcal{D}} = \{(s, a) \mid (s, a) \in \mathcal{D}\}$, rather than uniformly over the experiences $(s, a, r, s') \in \mathcal{D}$ in the buffer (which we call buffer-uniform here). The resulting agent is trained (approximately) uniform over the reachable state-action space (assuming sufficient exploration that ensures every (s, a) is in the buffer at least once). We observed the S, A-uniform buffer to train slightly slower and therefore compensated by increasing the total amount of training steps (1 million steps).

Figure 5 shows the performance of the S, A-uniform buffer during training and testing and its policy optimality in reachable space. For comparison, the best performing buffer-uniform replay buffer (ϵ-greedy decayed after 500.000 steps) is shown for 1 million training steps. The exploration for the S, A-uniform buffer is also decayed after 500.000 steps to ensure sufficient coverage of reachable space and a fair comparison between the two methods. The figure shows the testing performance of the buffer-uniform replay buffer starts declining again after 500.000 steps. This is in line with our previous results as the fraction of diverse data sampled from the replay buffer starts declining after exploration has decayed. In contrast, the S, A-uniform replay buffer does not suffer from this decline

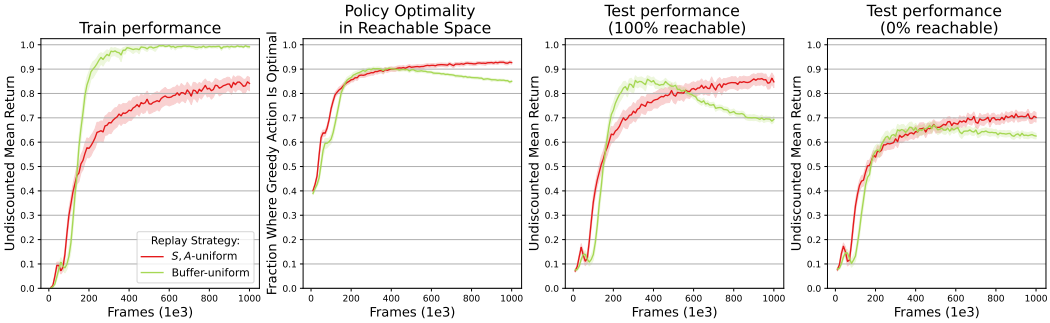


Figure 5: Performance of a uniform over state-action space (S, A-uniform) replay sampling strategy compared to the regular (buffer-uniform) replay sampling strategy. The buffer-uniform replay sampling strategy has ϵ-greedy decayed after 500.000 steps. Shown are the mean and 95% confidence interval over 100 seeds.

in diversity and instead keeps improving as training continues, outperforming the buffer-uniform buffer in everything other than training performance.

5.5 Analysing Latent Representation

The following experiment examines what part of the network is responsible for the difference in (generalisation) performance of the agents trained in Section 5.3. We analyse the learned latent representations by fine-tuning them on the (narrow) steady-state distribution ρ^{π^*} of the optimal policy during training. By evaluating on the entire reachable space and the test sets after fine-tuning we can evaluate whether the latent representations have learned to generalise or not.

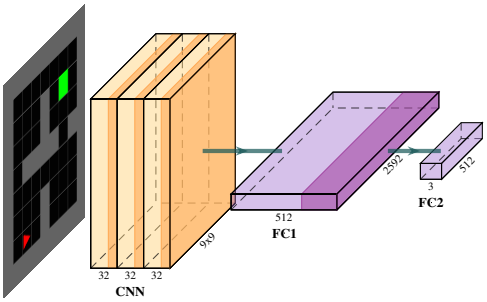


Figure 6: The architecture used in the experiments in Section 5.5. The dark-shaded regions indicate ReLU activation functions.

The experiment is structured as follows. In order to isolate the influence of the latent representation, we simplify the network by replacing the fully connected (FC) layers after layer FC1 with a linear probe (Alain & Bengio, 2016) (see Figure 6). Parts of the network in Figure 6 are frozen after the training done in Section 5.3. The downstream layers are reset and fine-tuned on narrow on-policy data collected by the converged (approximately optimal) agents. In order to avoid the challenges associated with offline RL (Levine et al., 2020), we allow for a small amount of additional online data collection (up to 4.000 environment samples), which does not contain any explicit exploration. See Appendix C for more details.

Figure 7 shows the performance when freezing the CNN and FC1 layers and fine-tuning FC2. This is compared to fine-tuning FC1 and FC2 with only the CNN frozen. The frozen parameters are taken from agents trained with less diverse and more diverse buffers (ϵ-greedy decayed after 50.000 and 500.000 steps respectively). The results show that the frozen CNN plus FC1, in contrast to only the frozen CNN, can quickly recover the training performance of the original agent. Moreover, it also recovers the generalisation performance, which depends on the diversity of the replay buffer of the original agent.

This suggests that the FC1 layer after the CNN has learned a latent representation that generalises to a large part of the reachable and unreachable space, even if the following layer (FC2) is fine-tuned on very narrow transition data. Increasing the diversity of the training data on which the original agents were trained, seems to induce a better generalising latent representation, both in terms of reachable and unreachable generalisation. Furthermore, since fine-tuning recovers the original performance, it

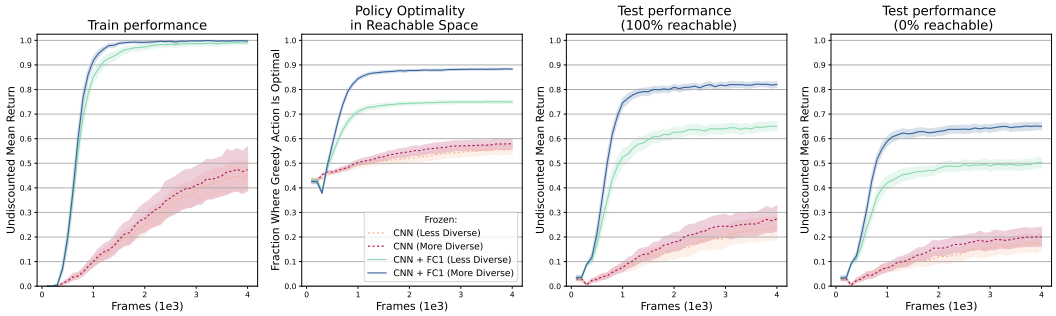


Figure 7: Fine-tuning performance with a frozen CNN or a frozen CNN and FC1. The less and more diverse frozen parameters are taken from the agents trained with ϵ-greedy decayed after 50.000 and 500.000 steps respectively. Shown are the mean and 95% confidence interval over 100 seeds.

seems the difference in generalisation performance of the original agents in Section 5.3 is due to the different latent representations learned by those agents. This seems in contrast to some of the results in Jiang et al. (2022), which suggest a component of the generalisation performance induced by more diverse training data is due to RL-specific effects beyond representation learning.

6 Conclusion & Future Work

In this paper, we reasoned that being optimal in a larger part of the reachable state space will improve zero-shot policy transfer to reachable states. We empirically showed this to hold in a 4-room grid world environment. We found a more diverse replay buffer to produce a policy that is optimal in a larger part of reachable space and has higher generalisation performance to reachable states. Additionally, a more diverse replay buffer was empirically shown to lead to improved generalisation performance to similar (in-distribution) but unreachable states. Finally, we found that the latent representation induced by the fully connected layer following the CNN portion of the network generalises to reachable and unreachable states when fine-tuned on very narrow data.

We believe this work demonstrates the non-trivial role of exploration and the replay buffer for generalisation in reinforcement learning. For now, this has been demonstrated in a relatively simple grid world environment. In future work, we would like to investigate whether the generalisation benefits will sustain when scaling up to more complex benchmarks. Furthermore, we have an intuition on why a more diverse exploration and replay strategy would result in better generalisation performance to unreachable states. We believe it could be promising to formalise and test this intuition in follow-up experiments.

References

Suzan Ece Ada, Emre Ugur, and H. Levent Akin. Generalization in Transfer Learning. *CoRR*, abs/1909.01331, 2019. URL `http://arxiv.org/abs/1909.01331`. arXiv: 1909.01331. 3

S. Akshay, Nathalie Bertrand, Serge Haddad, and Loïc Hélouët. The Steady-State Control Problem for Markov Decision Processes. In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio (eds.), *Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*, volume 8054 of *Lecture Notes in Computer Science*, pp. 290–304. Springer, 2013. doi: 10.1007/978-3-642-40196-1_26. URL `https://doi.org/10.1007/978-3-642-40196-1_26`. 2

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *CoRR*, abs/1610.01644, 2016. URL `http://arxiv.org/abs/1610.01644`. arXiv: 1610.01644. 7, 13

Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van Hoof, and Doina Precup. A Survey of Exploration Methods in Reinforcement Learning. *CoRR*, abs/2109.00157, 2021. URL `https://arxiv.org/abs/2109.00157`. arXiv: 2109.00157. 1

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5048–5058, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html. 1

Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuska. Improved deep reinforcement learning for robotics through distribution-based experience retention. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, South Korea, October 9-14, 2016*, pp. 3947–3952. IEEE, 2016. doi: 10.1109/IROS.2016.7759581. 1

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic Gridworld Environment for Gymnasium, 2018. URL https://github.com/Farama-Foundation/Minigrid. 5, 12

Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying Generalization in Reinforcement Learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 2019. URL http://proceedings.mlr.press/v97/cobbe19a.html. 3

Ben Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Robust Predictable Control. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 27813–27825, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e9f85782949743dcc42079e629332b5f-Abstract.html. 3

Irina Higgins, Arka Pal, Andrei A. Rusu, Loïc Matthey, Christopher P. Burgess, Alexander Pritzel, Matthew M. Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1480–1490. PMLR, 2017. URL http://proceedings.mlr.press/v70/higgins17a.html. 3

Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. Generalization in Reinforcement Learning with Selective Noise Injection and Information Bottleneck. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13956–13968, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e2ccf95a7f2e1878fcafc8376649b6e8-Abstract.html. 3

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient Non-stationarity and Generalisation in Deep Reinforcement Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Qun8fv4qSby. 3

David Isele and Akansel Cosgun. Selective Experience Replay for Lifelong Learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3302–3309. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16054. 1

Yiding Jiang, J. Zico Kolter, and Roberta Raileanu. Uncertainty-Driven Exploration for Generalization in Reinforcement Learning. December 2022. URL https://openreview.net/forum?id=GZDsKahGY-2. 3, 8

Ken Kansky, Tom Silver, David A. Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, D. Scott Phoenix, and Dileep George. Schema Networks: Zero-shot Transfer with a Generative Causal Model of Intuitive Physics. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1809–1818. PMLR, 2017. URL http://proceedings.mlr.press/v70/kansky17a.html. 3

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A Survey of Zero-shot Generalisation in Deep Reinforcement Learning. *J. Artif. Intell. Res.*, 76:201–264, 2023. doi: 10.1613/jair.1.14174. 1, 2, 3

Takis Konstantopoulos. Markov chains and random walks. *Lecture notes*, 2009. 2

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJgcvJBFvB. 3

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643. arXiv: 2005.01643. 7

Xingyu Lu, Kimin Lee, Pieter Abbeel, and Stas Tiomkin. Dynamics Generalization via Information Bottleneck in Deep Reinforcement Learning. *CoRR*, abs/2008.00614, 2020. URL https://arxiv.org/abs/2008.00614. arXiv: 2008.00614. 3

Clare Lyle, Mark Rowland, Will Dabney, Marta Kwiatkowska, and Yarin Gal. Learning Dynamics and Generalization in Deep Reinforcement Learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14560–14581. PMLR, 2022. URL https://proceedings.mlr.press/v162/lyle22a.html. 3

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pp. 1–8. IEEE, 2018. doi: 10.1109/ICRA.2018.8460528. 3

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html. 12

Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic Data Augmentation for Generalization in Reinforcement Learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5402–5415, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2b38c2df6a49b97f706ec9148ce48d86-Abstract.html. 3

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight Without a Single Real Image. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma (eds.), *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi: 10.15607/RSS.2017.XIII.034. URL http://www.roboticsproceedings.org/rss13/p34.html. 3

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05952. 1

Yujin Tang and David Ha. The Sensory Neuron as a Transformer: Permutation-Invariant Neural Networks for Reinforcement Learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 22574–22587, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/be3e9d3f7d70537357c67bb3f4086846-Abstract.html. 3

Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. In Carlos Artemio Coello Coello (ed.), *GECCO '20: Genetic and Evolutionary Computation Conference, Cancún Mexico, July 8-12, 2020*, pp. 414–424. ACM, 2020. doi: 10.1145/3377930.3389847. 3

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pp. 1–5. IEEE, 2015. doi: 10.1109/ITW.2015.7133169. 3

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pp. 23–30. IEEE, 2017. doi: 10.1109/IROS.2017.8202133. 3

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, December 2015. URL http://arxiv.org/abs/1509.06461. arXiv:1509.06461 [cs]. 12

Alvaro Velasquez, Ismail Alkhouri, K. Subramani, Piotr Wojciechowski, and George Atia. Optimal Deterministic Controller Synthesis from Steady-State Distributions. *Journal of Automated Reasoning*, 67(1):7, January 2023. ISSN 1573-0670. doi: 10.1007/s10817-022-09657-9. URL https://doi.org/10.1007/s10817-022-09657-9. 4

Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised Visual Attention and Invariance for Reinforcement Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 6677–6687. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00661. URL https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Unsupervised_Visual_Attention_and_Invariance_for_Reinforcement_Learning_CVPR_2021_paper.html. 3

Vinícius Flores Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David P. Reichert, Timothy P. Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew M. Botvinick, Oriol Vinyals, and Peter W. Battaglia. Relational Deep Reinforcement Learning. *CoRR*, abs/1806.01830, 2018. URL http://arxiv.org/abs/1806.01830. arXiv: 1806.01830. 3

Vinícius Flores Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David P. Reichert, Timothy P. Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew M. Botvinick, Oriol Vinyals, and Peter W. Battaglia. Deep reinforcement learning with relational inductive biases. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkxaFoC9KQ. 3

Shangtong Zhang and Richard S. Sutton. A Deeper Look at Experience Replay. *CoRR*, abs/1712.01275, 2017. URL http://arxiv.org/abs/1712.01275. arXiv: 1712.01275. 1

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain Generalization with MixStyle. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=6xHJ37MVxxp. 3

Hyper-parameter	Value
Total timesteps	500 000
Vectorised environments	10
DQN	
Buffer size	500 000
Batch size	256
Discount factor γ	0.99
Max. gradient norm	1
Gradient steps	1
Train frequency (steps)	10
Target update interval (steps)	10
Target soft update coefficient τ	0.01
Exploration initial ϵ	1
Exploration final ϵ	0.01
Adam	
Learning rate	1×10^{-4}
Weight decay	1×10^{-5}
CNN	
Kernel size	3
Stride	1
Padding	1
Padding mode	Circular
Channels	32

Table 1: Hyper-parameters used for the experiments

A 4-Room Grid World

The 4-room grid world used in our experiments is adapted from the `FourRooms` environment from the MiniGrid benchmark (Chevalier-Boisvert et al., 2018) and differs in certain ways from the default MiniGrid configuration. For one, the action space is reduced from the default seven actions (turn left, turn right, move forward, pick up an object, drop an object, toggle/activate an object, end episode) to just the first three actions (turn left, turn right, move forward). Also, the reward function is changed slightly to reward 1 for successfully reaching the goal and 0 otherwise (as opposed to the $1 - 0.9 * (\frac{\text{step count}}{\text{max steps}})$ given upon success by the default MiniGrid environment). Additionally, the size of the environment is reduced from the default 19 (8×8 rooms) to 9 (3×3 rooms).

Furthermore, the observation space is made fully observable and customised. Our agent receives a $4 \times 9 \times 9$ tensor that is centred around the agent's current location. The four binary-encoded channels contain the following information:

- **Channel 0:** The location of the agent (always in the centre).
- **Channel 1:** The hypothetical location where the agent would move to given the current direction it's facing (and ignoring any collisions with walls).
- **Channel 2:** The location of the walls.
- **Channel 3:** The location of the goal.

The implementation of the 4-room grid is also customised to allow for more control over the factors of variation (topology, agent location, agent direction, goal location) during the generation of a start state/task. This acts functionally the same as the `ReseedWrapper` from MiniGrid except that it allows for more control and therefore easier design and construction of the training and testing sets.

B Experimental Details

We use a DQN implementation forked from the Stable-Baselines3 (Raffin et al., 2021) repository and adapted to support double Q learning (van Hasselt et al., 2015). The network architecture consists of

three convolutional layers followed by four fully connected layers with ReLU activation functions (except for the last layer). The hidden dimensions of the linear layers are $[512, 128, 64]$. A full list of parameters can be found in Table 1. The experiments are performed on a single computer with a RTX 3070 GPU, 32 GB of memory and 12 cores (8 performance and 4 efficient). A single run of 500.000 steps had a runtime of roughly 8 minutes.

C Analysing Latent Representation

In order to more accurately analyse the performance due to some latent part of the network, we follow a process akin to linear probing (Alain & Bengio, 2016). For the agent with frozen CNN and FC1, the original remaining network (see above) is shortened by appending only a single linear layer (without activation function) to the frozen layers. We tried the same with the frozen CNN but found the single linear layer to not be sufficient to learn anything. Instead, we keep the fully connected layer FC1 followed by a single linear layer (but reset and fine-tune both). This way the network architectures are also identical between the two agents with differently frozen layers.

To obtain the steady-state distribution ρ^{π^*} of the optimal policy during training we simply perform greedy rollouts by the converged policies of our agents trained with ϵ-greedy decayed after 500.000 steps. At the start of fine-tuning, the agent's buffer is initialised with the data collected from these rollouts. Fine-tuning is performed with the regular DQN algorithm but with certain hyper-parameters adjusted to speed up training and prohibit exploration. The adjusted hyper-parameters can be found in Table 2.

Hyper-parameter	**Value**
Total timesteps	4 000
Vectorised environments	1
DQN	
Buffer size	4 000
Gradient steps	10
Train frequency (steps)	1
Target update interval (steps)	1
Target soft update coefficient τ	1
Exploration initial ϵ	0
Exploration final ϵ	0

Table 2: Hyper-parameters used for the experiments in Section 5.5

C.1 Additional Experiments

Since, the fine-tuning in Section 5.5 recovered the original performance of the agents, it seems the network used in the experiments in sections 5.3 and 5.4 was over-parametrised. For completeness, we reran the main experiment in Section 5.3 but with the smaller network as shown in Figure 6. The results can be found in Figure 8.

The results show that the smaller network achieves a higher policy optimality in reachable space and a higher generalisation performance to reachable and unreachable states than the original network. Moreover, the difference between the more and less diverse replay buffers is slightly less than before. This could be due to a smaller network overfitting less to the training data, resulting in a smaller difference between better and worse generalising strategies. However, we argue that our main results with the larger network imitate a more realistic scenario in which the network architecture isn't finely tuned to the complexity of the problem. In this case, the network has a higher chance of overfitting to the training data and the positive effects of a diverse buffer could be larger.

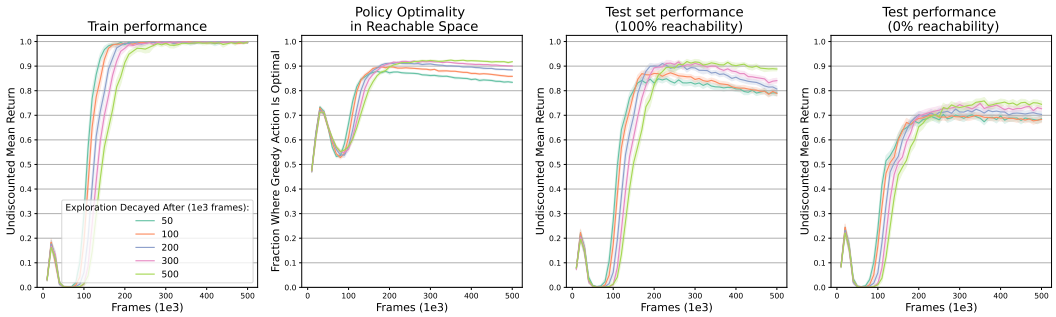


Figure 8: Performance if using a smaller network as shown in Figure 6. Shown are the mean and 95% confidence interval over 100 seeds.