Scientific Discovery and the Cost of Measurement – Balancing Information and Cost in Reinforcement Learning

Colin Bellinger,[1] **Andriy Drozdyuk,** [2] **Mark Crowley** [3] **Isaac Tamblyn** [3, 4, 5]

[1] National Research Council of Canada
[2] Carleton University
[3] University of Waterloo
[4] University of Ottawa
[5] Vector Institute for Artificial Intelligence

colin.bellinger@nrc-cnrc.gc.ca, andriy@drozdyuk.com, mark.crowley@uwaterloo.ca, isaac.tamblyn@uottawa.ca

Abstract

The use of reinforcement learning (RL) in scientific applications, such as materials design and automated chemistry, is increasing. A major challenge, however, lies in fact that measuring the state of the system is often costly and time consuming in scientific applications, whereas policy learning with RL requires a measurement after each time step. In this work, we make the measurement costs explicit in the form of a costed reward and propose a framework that enables off-the-shelf deep RL algorithms to learn a policy for both selecting actions and determining whether or not to measure the current state of the system at each time step. In this way, the agents learn to balance the need for information with the cost of information. Our results show that when trained under this regime, the Dueling DQN and PPO agents can learn optimal action policies whilst making up to 50% fewer state measurements, and recurrent neural networks can produce a greater than 50% reduction in measurements. We postulate the these reduction can help to lower the barrier to applying RL to real-world scientific applications.

Introduction

Deep reinforcement learning (DRL) has recently demonstrated its great potential in challenging simulated sequential decision making environment (Mnih et al. 2015). As a result, there is a growing interest in applying DRL to complex real-world problems. We are particularly interested in the application of DRL to improve and expedite design problems, such as those in materials science and pharmacology. The sequential decision making problems in these areas, however, often have explicit costs associated with measuring the current state of the environment. On the hand, reinforcement learning (RL) generally requires long training times with significant exploration. This places a high state measurement burden on the use of RL in applications with explicit measurement costs. We define this scenario as a Markov Decision Process (MDP) with explicit state measurement costs.

The existing RL solutions are not designed for MDPs with explicit state measurement costs. In the canonical MDP framework, observations of the state of the environment are produced automatically at each time sets and have no explicit associated costs. Generally, agents are either agnostic to the state observations provided by the environment in the sense that they learn from what they receive, or they attempt to improve the quality of observations through deep feature representations (Mnih et al. 2015), maintaining a belief state for partially observable MDPs (POMDP) (Kaelbling, Littman, and Cassandra 1998), or taking actions to change the state of the environment in order to gain a better understanding of it (Bossens, Townsend, and Sobey 2019). There is no class of algorithms, however, that aims to balance learning a policy that maximises the discounted sum of rewards with minimising the incurred measurement costs.

In materials design(Steiner et al. 2019; Burger et al. 2020), determining the microscopic state of the environment after applying a process (such as heat, stirring, etc), requires the use of costly measurement and characterisation equipment. For RL to be useful in this application, an agent must be more thoughtful about which states, and at which time, are most useful to measure. The same is true for problems of combinatorial optimisation via quantum-annealing(Mills, Ronagh, and Tamblyn 2020); repeated measurements of the system have a high associated cost (they are destructive) and therefore significantly increase the time-to-solution (and hence cost).

The optimal solution should both achieve the design goal and limit the total costs associated with carrying out the design. We denote the reward minus the cost as the costed reward and the sum of the long-term discounted rewards minus costs as the costed return (Bellinger et al. 2021). The fundamental question in this current work is, can DRL be used to find a non-trivial policy which balances the need for information with the cost of information acquisition?

In our previous work, we developed a tubular RL algorithm to maximize the costed return (Bellinger et al. 2021). Here, we explore deep learning strategies to learn policies for sequential decision making problems with explicit costs. We do this with a particular focus on setting up an approach that enables the use of off-the-shelf DRL algorithms. The fundamental property of costed reward problems is that the DRL agent can freely explore the relationship between actions and rewards with the objective of learning a policy that achieves a high costed return, but is charged each time it measures the next state.

In this work, we present a strategy to modify the stan-

dard reinforcement learning framework to enable off-the-shelf DRL algorithms to achieve the goal of maximising the costed return. Our solution includes the DRL agents that learn both an action policy and a state measurement policy. To enable this, we expand the state and action spaces, and add an explicit cost and one-state memory to the environment. Our results show that the modified framework enables agents trained with PPO (Schulman et al. 2017) and Dueling DQN (Wang et al. 2016) learn interesting state measurement policies that depend on the unknown underlying dynamics of the environment to minimise costs whilst learning an action policy that efficiently achieves the goal. Moreover, our results also suggest that adding recurrent memory to the model architecture leads to further improvements in the costed reward. We demonstrate this behaviour on Open AI gym environments (Brockman et al. 2016) using the costed reward wrapper class developed for this research[1].

Related Work

This work has some relationship to previous work on *active reinforcement learning* (Akrour, Schoenauer, and Sebag 2012; Krueger et al. 2016; Schulze and Evans 2018) that modified the action space. In particular, we utilised the previously proposed strategy of action pairs that encode a directive about how to move and whether to measure the state of the environment or utilise an oracle. The previous work, however, involved using a human experts to address the challenge of defining a complete reward signal. Alternatively, our work is focused on reducing the costs associate with measure the state of the environment.

As with our work, measurement costs sometimes occur POMDPs (Kaelbling, Littman, and Cassandra 1998). Moreover, the use of the last measured state is can be seen as partial observably. Classic POMDPs techniques, however, require learning a model of the underlying dynamics or having *a-priori* knowledge of them. Critically, while uncertainty in POMDPs is a consequence of the external environment, here the uncertainty in a function of agent's choice to forgo measuring the state of the environment to lower its observation costs. A POMDP agent can only indirectly affect uncertainty by choosing actions that change the environment thereby lowering its uncertainty. Alternatively, under our framework, the agent can always opt to measure the state of the environment to removing uncertain at a cost. As a result, our proposal resides in the grey area between fully observable Markov Decision Processes (MDPs) and Partially Observable Markov Decision Processes (POMDPs). The agent can choose to operate exclusively in a fully observable world by paying for a state measurement, or learn a that periodically measure at cost. Therefore, can be seen as a sub-category of POMDP that can be efficiently solved using off-the-shelf DRL. The effectiveness of a tabular RL setup version using classic POMDP methods was demonstrated in our previous work (Bellinger et al. 2021)

From an alternate perspective, this class of problem can been seen as a novel subclass of *mixed observable MDP (MOMDP)* (Ong et al. 2010), because the agent learns from a blend of fully observable measurements of the next state and noisy estimates of that state. In standard POMDPs and MOMDPs the agent operates with a degree of uncertainty due to the incomplete observable information received at each time step but without any notion of measurement cost itself. As far as we are aware, our proposed solution is the first to enable the agent explicitly decide to utilise either fully or partially observable measurements at each time set.

[1]Costed reward wrapper class code: http://clean.energyscience.ca/codes

Problem Formulation

This work focus on the classic RL setup which includes the environment as a tuple: (S, A, P, S', R, γ). These are the standard components of an MDP, where S is the state-space, A is the action-space, $P(s'|s, a)$ is the state transition probabilities, $R(s, a)$ is the reward function, and $\gamma \in [0, 1]$ is a discount factor. P and R are not known by the agent.

In this work, we focus on episodic environments with continuous states, $S \in \mathbf{R}^n$, discrete action sets $A = \{1, ..., |A|\}$, and stationary state-transition dynamics. The standard RL objective is to learn a policy $\pi : s \rightarrow a$ that maximises the **return**, which is defined as the discounted sum of rewards: $v(s) = E\Big[\sum_{t=0}^{\infty} \gamma^t \big(R(s_t, a_t)\big) \mid s = s_0\Big]$.

As introduced above, our objective is to facilitate off-the-shelf DRL algorithms maximise the costed return. As a result, we incorporate the explicit measurement costs into the return as:

$$v(s) = E\Big[\sum_{t=0}^{\infty} \gamma^t \big(R(s_t, a_t) - C(m_t)\big) \mid s = s_0\Big], \quad (1)$$

where $C(m_t)$ is measurement costs at time t.

For DRL to learn a non-trivial policy that maximises the costed return, it needs the ability to explicitly decide which actions to take and whether or not to measure the next state at each time step. Formally, the agent learns action and measurement policies $\pi : s \rightarrow a, m$. To enable DRL to , we propose a five simple modifications to the classic RL framework.

The first modification is to expand the action space to action pairs (atomic-action, measurement directive). The atomic-action is the standard action (*e.g.* add heat, move up, *etc.*), and the measurement directive is a Boolean flag that indicates if the current state of the environment should be measured and returned after the application of the atomic action.

When the agent selects an action pair that gives the directive to measure the environment ($m = 1$), the measurement is returned to the agent for use in the selection of the next action pair. In addition, we augment the environment to give it a memory of the last measured state. When the agent selects an action pair that gives the directive not to measure the environment ($m = 0$), the environment returns the last measured state to the agent. In order to learn a useful policy, however, the agent needs to know when it is seeing the current measurement of the system and when it is seeing a stale one. We provide for this by expanding the state space to include a Boolean flag at the end of the state representation.

When the environment returns the last measured state to the agent, the flag is set to zero, otherwise, it is set to one. As an alternative, this augmentation can be omitted if the agent is give memory of its last action pair. However, augmenting the state space rather than the agent enables the use of off-the-self DRL. The results are expected to be the same either way.

Most importantly, in order to encourage the agent to learn a policy that only measures the state of the environment when it is necessary, we add an explicit user-defined measurement cost to the environment. The environment subtracts this cost from the reward if and only if the agent uses the measure directive. Therefore, an agent which measures at every time step will also know the current state of the system, but the reward at each time step will be reduced. On the other hand, a agent that never measures will save the measurement costs, but also will never have a clear picture of where it is in the state space. A non-trivial policy will balanced the cost of information with the need for information in order to efficiently (in terms of time and cost) achieve goal.

The our wrapper class for the Open AI gym (Brockman et al. 2016) can be accessed here: http://clean.energyscience.ca/codes.

Experimental Setup

We evaluate the performance of PPO (Schulman et al. 2017) and Dueling DQN (Wang et al. 2016) on the costed version of the Open AI gym environments Cartpole, Acrobot and Lunar lander (Brockman et al. 2016). The DRL algorithms are implemented using Pytorch and the Tianshou deep learning packages (Weng et al. 2021), and the results were record with Weights and Biases (Biewald 2020). The experiments were executed on Ubuntu 18.04 desktop running a GeForce RTX 2080 Ti GPU.

In our experiments, we first compare the performance of the agents on the standard Open AI gym environments (we refer to this as the vanilla environments, vanilla=1) to the costed reward version of the environments (vanilla=0). These results serve to reveal the impact (if any) of costed reward wrapper class on policy learning in the target environments. The main focus of the experiments are: *a*) our analysis of the agents ability to learn non-trivial policies in the costed reward environments with increases measurements costs, and *b*) our analysis of the impact of cost on the agents learned measurement policy. All of the results reported below are averaged over 10 independent trials. The code is available upon request and will be made public after publication.

Results

Vanilla Environment Versus Costed Environment

This section evaluates if the costed reward wrapper has any strong negative impact on policy learning in selected Open AI Gym environment. This analysis is conducted comparing the agents performance on the vanilla environments to the performance on the costed reward version of the environment with the measurement cost is set to 0. The corresponding results for Dueling DQN agents training on the vanilla Cartpole and Acrobot environments are presented in Figure 1. The results plot the median test rewards on the vanilla environment versus the costed reward version of environment with cost = 0. The results show the agents learning under the costed reward regime converge to the same median reward as on the vanilla environments. This result also holds for Lunar Lander, but is omitted due to space considerations.

We could expect some lag in the learning of agents on the costed reward versions of the environments because these have larger state and actions spaces. Indeed, we see this lag with agent learning on the costed reward version of Cartpole, where it takes slightly longer to start learn, but not in the other environments. This is of interest because Cartpole is easiest of the three environment. More research is required to understand this phenomenon.

Learning with a cost

In this section, we evaluate the agents ability to learn in the costed reward regime as the measurement costs are increased. The median costed reward results for Dueling DQN on Cartpole and Acrobot are depicted in Figure 2. In particular, each Figure plots the performance curves for cost 0, 0.1, 0.2 and 0.3.

As verified in the previous section, the cost=0 curves show the optimal performance for the agents on each of the environments. Therefore, these serve as an ambitious, though achievable, objective for the agents learning with costs greater than 0. Achieving this goal would require the agent to never measure the state of the environment. This is clearly not possible, but it aids our analysis. On the other hand, the lower benchmark is an agent the measures at every time step. This result is the agent's reward at each time step having the cost subtracted. For Cartpole, this bounds the reward of an agent that learns an optimal action policy between 200 and 200-(200*cost). Specifically, 180, 160 and 140 for costs of 0.1, 0.2, and 0.3 respectively. For Acrobot, this amounts to approximately -71, -78, and -84.

The results for Cartpole and Acrobot show that whilst the agents policies converge, the levels are well below the ambitious upper targets, and they are significantly above the baseline that solves the problem by measuring at every time step. On Cartpole, for example, the medians of the best reward per trial are 190.03, 180.07, and 170.10. Therefore, the agents are able to learn non-trivial measurement policies that enable them to solve the problem and reduce their measurement costs. This performance trend holds for Lunar lander as well.

Learned Measurement policies

In this section, we take a deeper dive into the measurement policies learned by the agents on the costed reward environments. Here, we aim to understand what the measurement policy behaviour looks like, and if / how it depends on cost and the dynamics of the environment.

To explore this question, we load the best policy learned by the respective agents and test them over 10 independent episodes. First, we compare the measurement policy learned

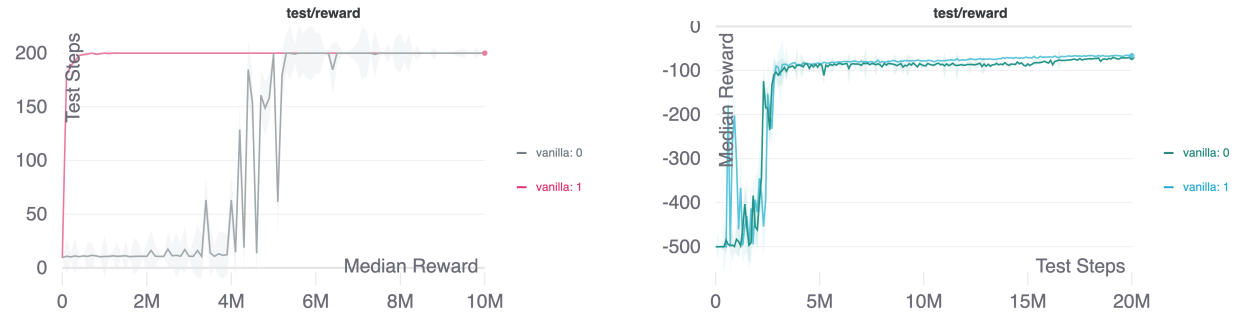

Figure 1: Comparison of Cartpole (left) and Acrobot (right) environments in the vanilla mode and in the costed reward mode with cost equal 0. The results illustrate that expanding the state and action-space have minimal impact on policy learning.

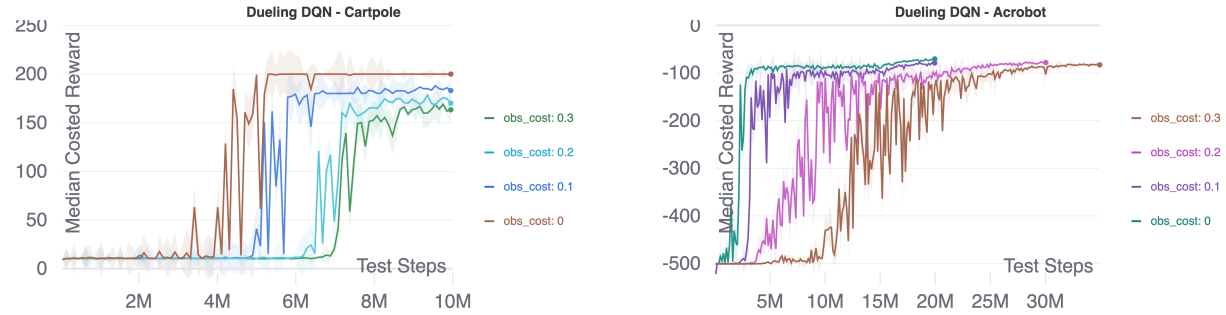

Figure 2: Comparison of Cartpole (left) and Acrobot (right) environments in the vanilla (m=1) with increasing measurement costs. The results illustrate that increasing the cost reduces the costed rewards, but the agent is able to learn a policy that balances the frequency and cost of measurements with the need to obtain information to achieve the goal.

on Cartpole with costs of 0 and 0.3. The plots in Figure 3 depict the measurement directives issued by the learned policy during the evaluation. The x-axis specifies the time steps of the episode and the y-axis specifies the independent episode trial. At each time step, the the colour of the rectangle indicates if the agent's learned policy requested a measurement or not. The first plot corresponds to a cost of 0 and illustrates that on the Cartpole environment with the cost set to 0, the agent learns to measure at each time step. This is clearly the best policy when measurements are free.

The second plot in Figure 3 illustrates the measurement policy when the cost is set 0.3. Here we see the agent learns a policy that alternates between measuring and not measuring at consecutive time steps. In both the case of cost 0 and 0.3 the agents learn optimal action policies that enable the pole to stay upright until the maximum of 200 steps. Importantly, when the agent is change a fee 0.3 for measuring the state, it is able to learn a more cost efficient policy.

The results for costs 0.1 and 0.2, which are withheld for space, show that the Dueling DQN and PPO agents learns the same measurement policy. This indicates that a standard DRL agent operating in the proposed costed setup can take at most one step without measuring. This a consequence of the proposed costed reward setup. More memory in the environment or an alternative setup are required to step more steps without measuring. We consider the use of recurrent networks in the next section.

The results in Figure 4 demonstrate the measurement policies learning by Dueling DQN on the Acrobot and Lunar Lander environments with a measurement cost of 0.3. These results show that the agents do not learn the clean pattern produced by the agents Cartpole. Rather, the measurement policies are quite unique and are dependent on the dynamics of the environment. In the case of Lunar Lander (right plot), early in the episode, the policy takes measurement at each step (shown by the prevalence orange rectangles in the early steps of the episode). We postulate that this is to account for the changes in environment that occur at the start of each episode. Once the agent has established an understanding of the environment in the current episode, the policy shifts to alternating between measuring and not measuring (alternating blue and orange rectangles in the middle of the episode.) Finally, when the lander is close to and lined up over the landing zone, the agent no longer needs to measure at all (all blue rectangles towards the end of the episode).

For Acrobot, shown in the left plot of Figure 4, the agent learns to use the alternating measure-do-not-measure pattern for the first two-thirds of the episodes. Acrobot, like Cartpole, starts each episode in the same position with the same dynamics. Therefore, the agent can forego some measurements. Alternative, there are many trajectories to flip the Acrobot over the goal line and as the agent approaches achieving goal, the action choice becomes more sensitive, thus the agent learns that needs to measure frequently as it gets closer to achieving the goal.

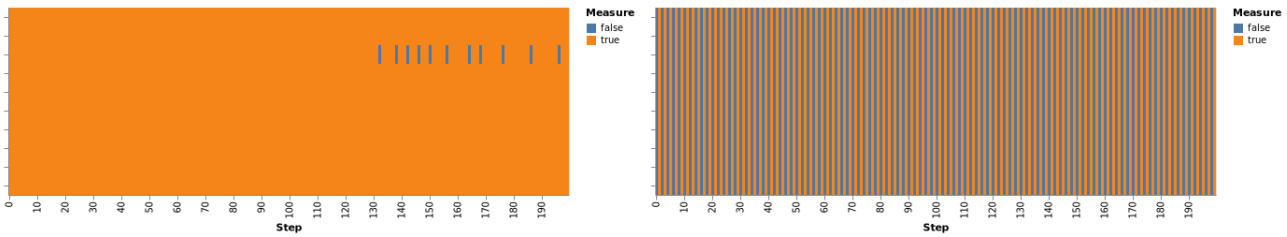

Figure 3: Comparison of the measurement behaviour of Dueling DQN on Cartpole environments with cost of 0 and 0.3. The show that when the cost is greater than zero, the agent learns to measure only when necessary to achieve the goal.

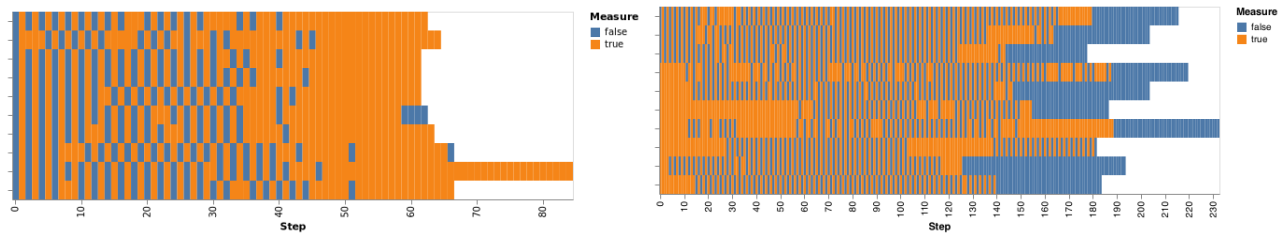

Figure 4: Comparison of the measurement behaviour of Dueling DQN on Acrobot environments (left) and lunar lander environment (right) with cost of 0.3. This demonstrates that when the cost is greater than zero, the agent learns to measure only when necessary to achieve the goal. Interestingly, the measurement pattern demonstrates that the measurement policy is dependent upon the dynamics of the system.

Recurrent Memory

The final set of results are presented in Figure 6 and Figure 5. These results compare the use of Dueling DQN to DQN with a recurrent network (DRQN). The objective is to explore the idea of adding recurrent memory to the agent. The preliminary results on Cartpole show that Deuling DQN with recurrent memory can can learn to make fewer measurements of the environment, whilst still achieving the goal. As a result, it acquires a higher costed reward.

The results in Figure 5 compare the median of the best costed reward achieved by the Dueling DQN and DRQN on Cartpole. The results are shown for costs of 0, 0.1, 0.2 and 0.3. For cost=0, both agents learn policies that achieve the optimal reward. As the cost is increased from 0, the plot shows that DRQN has a growing advantage in this environment in terms of the costed reward. It is also revealed that the Dueling DQN policy as very little standard deviation between test trails (the variance bars are too small so see on this plot.) Conversely, the DRQN performance of the DRQN policy across trials has a noticeable standard deviation. Nonetheless, even with the larger standard deviation, DRQN has a clear advantage.

Figure 6 presents the measurement policy for Dueling DQN and DRQN on Cartpole with the measurement cost of 0.3. In the case of Dueling DQN (left plot), we see the familiar alternating measurement strategy. Alternatively, in the DRQN plot on the right, we see that the plot is dominated by blue rectangles with orange ones interspersed. This illustrates that DRQN learns an action policy that solves the environment whilst learning a measurement policy that takes consecutive steps without measuring, thereby achieving the higher costed rewards seen in Figure 6. This demonstrates that recurrent DRL can exploit its hidden state representation to tract the true state of the environment within the costed reward regime. Importantly, it can utilise this to reduce the minimum measurement frequency required by the agent and achieve a higher costed reward.

Discussion

In laboratory science and design where measuring the state of the system is costly and time consuming, it is standard that the research workflow smoothly shifts between take real measurements of the state of the system and inferring sets state based on textbook knowledge or past experience. On the other hand, the standard RL framework requires a measurement at teach time step. Although RL is expected to be beneficial in many scientific applications, it need for a frequent and predetermined state measurement pattern can place a significant barrier on the application of RL to problems with high measurement costs. The above results demonstrate that the proposed costed reward regime enables off-the-shelf DRL algorithms to learn a non-trivial state measurement policy whilst learning an optimal action selection policy. This is one step towards enable the use of RL in many scientific applications.

In spite of the benefits demonstrated with the use of the proposed costed reward regime, we see the reliance of action pairs a limitation. The environments considered here had relatively small actions spaces. As a result, mapping the actions to action pairs, which doubles the size of the action space, had minimal negative impact on the effective learning rate. Real-world applications, however, often have a large number of discrete actions or have a continuous action space. The current solution does not have a straightforward mapping to

Figure 5: Comparison of the best costed rewards of Dueling DQN and DRQN on Cartpole environments with increasing costs. As the costs increase, DRQN exploits is recurrent memory to measure less frequently and achive a higher costed reward.

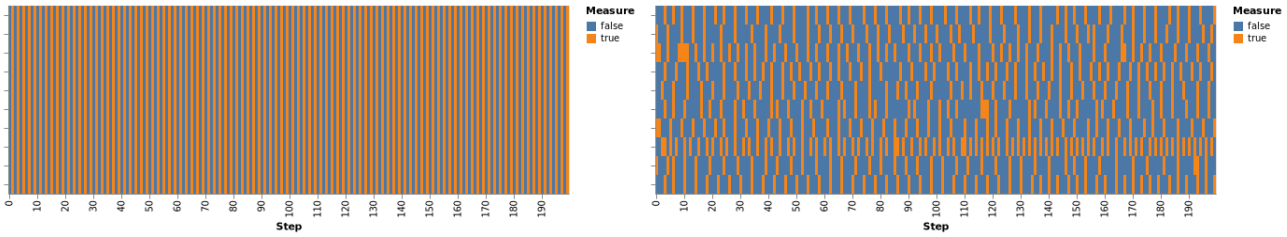

Figure 6: Comparison of the measurement behaviour of Dueling DQN and DRQN on Cartpole environments with cost of 0.3. The show that when the cost is DRQN is able to further reduce its measurement requirements.

continuous action spaces, a side from discretizing the action space, which is not always ideal.

The results demonstrate DRL algorithms learning within the costed reward regime can successfully learn a measurement policy that reduce the need for measurement. The best achievable reduction is to cut the measurement requirement in half. The reduction, however, is limited by the complexity of the environment. In the case of Cartpole, which as simple dynamics and a single start state, the maximum possible reduction is achieved. In more complex environments, the agent achieves meaningful reductions but can not fully reduce the measurements to half. Our results for recurrent DRL demonstrate that it can help further reduce in the measurement frequency. Our on-going work is explore the full potential the recurrent DRL improve the measurement requirements in more complex applications and areas of the state space, such as at the end of Acrobot episodes and the beginning of Lunar Lander episodes.

Conclusion

Many science-based applications have high temporal and monetary costs associated with measuring the state of the environment. This places a significant barrier on the use of RL in scientific applications such as materials design and pharmacology. As a result, developing strategies that reduce the number of state measurements demanded by RL is an important feature to improving its applicability. To achieve this, we postulate that the RL agent, much like a laboratory researcher, should be given the control to decided it is necessary to measure the state of the system. From an RL perspective, this poses an interesting challenge that requires the agent to learn a policy that balances the need for information with the cost of obtaining it, while attempting to learn an optimal action policy.

This work proposed a set of modifications to the standard RL framework which facilitates the use of off-the-shelf DRL costed reward regime. Our results show that within this setup, Dual DQN and PPO can learn optimal action policies and a measurement policy that reduces the measurement cost. Using DRL, the agents can learn to cut the total cost in half, whereas DRQN can reduce the cost further by leveraging its internal memory to facilitate successive steps without measuring the state of the environment.

Our on-going work is focused on developing a simulated chemistry lab environment as a test-bed that is closer to the real-world applications of interest. From a technical perspective, we are exploring alternatives to the action pair setup so as to not require a doubling the action space, and conducting more research into the use of recurrent network architectures in a wider variety of costed reward environments and with policy gradient methods.

Acknowledgements

This research was performed under the auspices of the National Research Council of Canada's AI for Design Program.

References

Akrour, R.; Schoenauer, M.; and Sebag, M. 2012. April: Active preference learning-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 116–131. Springer.

Bellinger, C.; Coles, R.; Crowley, M.; and Tamblyn, I. 2021. Active Measure Reinforcement Learning for Observation Cost Minimization. In *The 34th Canadian Conference on Artificial Intelligence*.

Biewald, L. 2020. Experiment Tracking with Weights and Biases. Software available from wandb.com.

Bossens, D.; Townsend, N. C.; and Sobey, A. 2019. Learning to learn with active adaptive perception. *Neural Networks*, 115: 30–49.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. arXiv:1606.01540.

Burger, B.; Maffettone, P. M.; Gusev, V. V.; Aitchison, C. M.; Bai, Y.; Wang, X.; Li, X.; Alston, B. M.; Li, B.; Clowes, R.; Rankin, N.; Harris, B.; Sprick, R. S.; and Cooper, A. I. 2020. A mobile robotic chemist. *Nature*, 583(7815): 237–241.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2): 99–134.

Krueger, D.; Leike, J.; Evans, O.; and Salvatier, J. 2016. Active reinforcement learning: Observing rewards at a cost. In *Future of Interactive Learning Machines, NIPS Workshop*.

Mills, K.; Ronagh, P.; and Tamblyn, I. 2020. Finding the ground state of spin Hamiltonians with reinforcement learning. *Nature Machine Intelligence*, 2(9): 509–517.

Mnih, V.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529.

Ong, S. C.; Png, S. W.; Hsu, D.; and Lee, W. S. 2010. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research*, 29(8): 1053–1068.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schulze, S.; and Evans, O. 2018. Active reinforcement learning with monte-carlo tree search. *arXiv preprint arXiv:1803.04926*.

Steiner, S.; Wolf, J.; Glatzel, S.; Andreou, A.; Granda, J. M.; Keenan, G.; Hinkley, T.; Aragon-Camarasa, G.; Kitson, P. J.; Angelone, D.; et al. 2019. Organic synthesis in a modular robotic system driven by a chemical programming language. *Science*, 363(6423).

Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; and Freitas, N. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, 1995–2003. PMLR.

Weng, J.; Chen, H.; Yan, D.; You, K.; Duburcq, A.; Zhang, M.; Su, H.; and Zhu, J. 2021. Tianshou: a Highly Modularized Deep Reinforcement Learning Library. *arXiv preprint arXiv:2107.14171*.