ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and
Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9.
Available from http://www.i6doc.com/en/.

Potential analysis of a Quantum RL controller in the context of autonomous driving

M. Lautaro Hickmann[1], Arne Raulf[1], Frank Köster[1],
Friedhelm Schwenker[2], and Hans-Martin Rieser[1]

1- German Aerospace Center (DLR) - Institute for AI Safety and Security
Ulm and St. Augustin - Germany

2- University of Ulm - Institute of Neural Information Processing
James-Franck-Ring 89081 Ulm - Germany

Abstract. The potential of quantum enhanced Q-learning with a focus on its applicability to a lane change manoeuvre is investigated. In this context we solve multiple simple reinforcement learning environments using variational quantum circuits. The achieved results were similar to or even better than those of a simple constrained classical agent. We could observe promising behaviour on the more complex lane change manoeuvre task, which has an environment with an observation vector size twice larger than commonly used ones. For the Frozen Lake environment we found indications of possible quantum advantages in convergence rate.

1 Introduction

The research field of Quantum Reinforcement Learning (QRL) is in an early stage of development. Recently, quantum enhanced deep Q-learning algorithms have been shown to work on small environments with discrete or continuous state space using Variational Quantum Circuits (VQCs) as approximators [1].

In Refs. [2, 3] experimental speed-ups for QRL were shown by using effective quantum algorithms on environments created with both classical and quantum communication channels between the agent and the environment.

In this work, we investigated the potential of QRL for solving complex realistic tasks in classical environments, such as merging into a highway in the context of automated driving [4]. This use case is complex while still using an observation vector size that can be directly encoded into a quantum circuit. We focused on a feasibility study and a constrained comparison with simple classical models.

2 Quantum Reinforcement Learning

In Reinforcement Learning (RL) an agent has to learn an optimal interaction strategy with an environment through trial and error, getting rewarded depending on it's behaviour.

We used a value-based, off-policy deep Q-learning approach with experience replay and fixed Q-value targets in all experiments, following the training procedure suggested in Ref. [1].

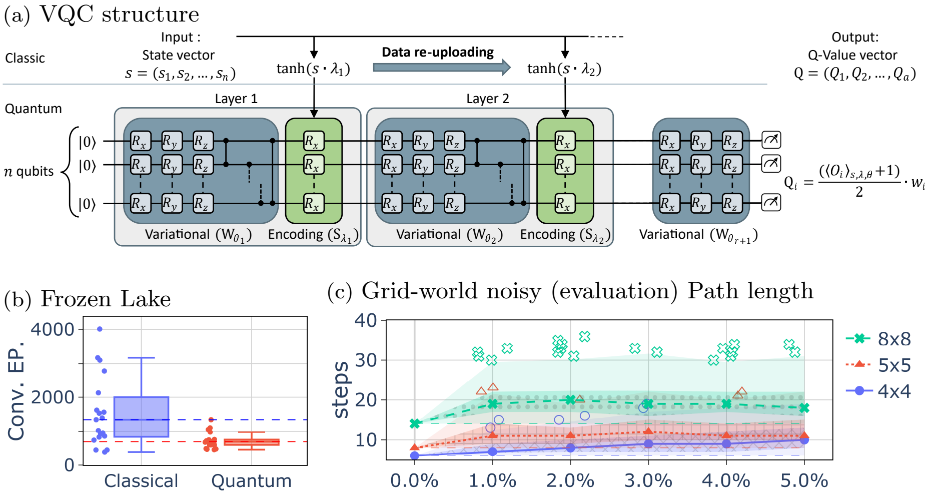


Fig. 1: **(a)** VQC structure used for Q-value predictions, adapted from [5]. **(b)** Convergence epoch in the Frozen Lake environment (dashed lines represent the median). **(c)** Quantum agent's path length for different noise probabilities on various grid worlds.

To approximate the Q-values we used simple fully connected dense feed-forward neural networks for the classical agents. For the quantum agents we used a VQC architecture based on Refs. [1, 5], shown in Figure 1a.

To improve the quantum agent's ability to adapt to environments, we used classical pre- and post-processing [1]. Starting with a classical observation vector s (of length n) a layer-wise prepossessing $\phi_\lambda(s) = (\tanh(s \cdot \lambda_1), \dots, \tanh(s \cdot \lambda_r))$ was applied, with trainable scaling parameters λ_i and r data encoding layer repetitions, i.e. number of data re-uploads. $\phi_\lambda(s)$ was then encoded into a n-qubits system initialised in the ground state $|0^{\otimes n}\rangle$ using rotational encoding and data re-uploading. The Q-value for action i in state s is then given by:

$$Q(s,i) = \frac{\left(\langle 0^{\otimes n} | U_\theta(\phi_\lambda(s))^\dagger O_i U_\theta(\phi_\lambda(s)) | 0^{\otimes n} \rangle\right) + 1}{2} \cdot w_{o_i},$$

with $U_\theta(\phi_\lambda(s))$ the unitary operator describing the state preparation carried out by the parameterised quantum circuit, O_i the observable (selected as part of a hyperparameter search) and w_{o_i} the post-processing scaling weight.

3 Experiments and Results

The lane change manoeuvre environment was proposed in Ref. [4], using the ADORe[1] framework. It presents a scenario where a vehicle has to merge into randomly simulated moving traffic from a slip road. Humans can make decisions intuitively in such cases, but a rule-based derivation is not straightforward [4].

[1] `https://github.com/eclipse/adore`

		Environments								
		Frozen Lake	Grid World of size $l \times l$					Cart Pole		Lane Change
Metrics			4	5	8	11	16	v_0	v_1	
n		4	4	5	6	7	8	4		14
Train.	Quantum	96	96	119	142	165	188	94		326
params	Classical	121	121	134	147	160	173	93(95)		*1334*
Envs.	Quantum	100%	100%	100%	100%	90%	85%	100%	65%	52%
solved	Classical	100%	100%	100%	100%	100%	85%	100%(90%)	35%(75%)	*93*

Table 1: Metrics per environment. With n the observation vector size. Metrics for the classical agent with 2 hidden layers are shown in parenthesis. *The classical results for the Lane Change environment are taken from Ref [4].*

Based on a continuous 14-dimensional observation vector, the agents have to select one of up to four traffic gaps to merge into. This gap is communicated to a classical algorithmic non-trainable trajectory planner that tries to perform the manoeuvre [6]. The agents get a sparse rewarded in case of a successful merge.

To assess the feasibility of solving the described environment using QRL, we first investigated simpler environments with similar properties. We evaluated the capabilities of the implemented VQC for solving tasks with Q-learning by using the same architecture for all experiments. Furthermore, we investigated the effects of observation vector scaling and simple noise on the agents' performance.

To enable the comparison of classical and quantum agents, we constrained the former to have a similar number of trainable parameters as the latter, and used the same inputs for both kind of agents. We used quantum agents with 5 re-upload-layers on all experiments, following Ref. [1]. For the classical agents we used a single hidden layer constrained to 13 neurons, except for the last two environments. This constrain leads to a comparable number of trainable parameters for each environment, as shown in Table 1. For each experiment we carried out an extensive hyperparameter search for both kind of agents, and retrained the best models with 20 different seeds.

All environments used had an OpenAI gym interfaces. To train, simulate and define the quantum circuits we used Cirq, TF-Quantum and TF-Agents[2].

3.1 Frozen Lake

We first tested the non-slippery Frozen Lake[3] environment, which has a discrete state space, unlike the lane change case. The agents must learn to cross a (fixed) frozen lake from start to finish without falling into any holes.

For this and the following environment, the agents can move in the 4 discrete cardinal directions. The state vector for a map of side length l ($l = 4$ for Frozen Lake) is the $\lceil \log_2(l^2) \rceil$-bit binary representation of the agent's current position enumerated from 0 to $(l^2 - 1)$ in row-major order.

[2] Gym is now called gymnasium see `https://gymnasium.farama.org/`. See `https://quantumai.google/cirq` and `https://www.tensorflow.org` for details on the libraries used. The code is available upon reasonable request.

[3] See `https://gymnasium.farama.org/environments/toy_text/frozen_lake/` for details.

Since both the environment and agents were deterministic, one evaluation episode suffices to determine whether the environment has been solved.

As shown in Table 1, both agents could solve the environment perfectly but the quantum agent converged within half the epochs needed by the classical agent (see Figure 1b). This hints at a possible training complexity advantage.

3.2 Scalability using Grid Worlds

We used grid world like environments of different sizes to test the agents' ability to handle scaling state spaces. These environments were derived from the Frozen Lake environments by eliminating the holes, thus making their complexity depend solely on the state vector size.

Table 1 shows the different map sizes used and their respective observation vector size, i.e. the number of qubits used. In this experiment we chose the hyperparameter configurations that worked best over all map sizes while making sure that all agents used the same number of data points per training epoch.

As shown in Table 1, both agents solve most of the environments perfectly with performance degrading for larger maps. Both kinds of agents behaved similarly regarding the number of epochs needed until convergence. In all cases, the agents first found suboptimal paths already solving the environment and later converged to paths of optimal length.

3.3 Cart Pole

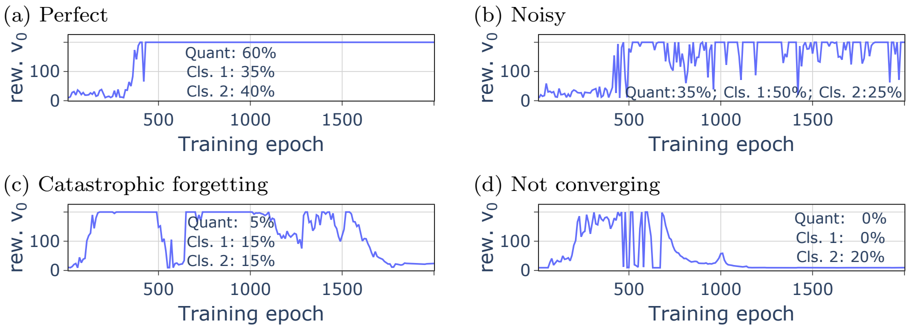
We used the Cart Pole[4] environment to test the agents' performance on continuous state spaces. The agents have to learn to balance a pole upright on a cart that can be discretely moved horizontally on a frictionless track. Differently to the lane change case the agents get an immediate reward per step. For each episode, the environment is initialised in a valid random state.

There are two versions of the Cart Pole environment, that differ in the maximal episode length. We trained the agents on the shorter v_0 version and tested their generalisation on the longer v_1 version without retraining. Each environment is solved if the agent achieves a score $\geq 95\%$ of the maximal episode length in 100 consecutive evaluation episodes.

As shown in Table 1, both the classical and quantum agent could solve the v_0 environment perfectly. But when testing their generalisation capabilities on longer episodes, the classical one did not achieve stable results. To remediate this, we extended it to have 2 hidden layers containing 9 and 4 neurons, achieving satisfactory results on v_1, although its performance on v_0 decreased (shown in parentheses in Table 1). On the other hand, the quantum agent could solve the v_0 environment perfectly and also generalised to the v_1 version with comparable results to the extended classical agent, without modifications.

Fig. 2 shows prototypical evaluation histories and the percentage of runs per model that behaved similar to each prototypical example. Notably, the quantum agent had the most perfect runs and the least catastrophic forgetting runs.

[4] See `https://gymnasium.farama.org/environments/classic_control/cart_pole/`.


(a) Perfect

(b) Noisy

(c) Catastrophic forgetting

(d) Not converging

Fig. 2: Prototypical examples of the mean reward evaluation history in the Cart Pole v_0 environment, carrying out 10 evaluation episodes every 10 training epochs. The percentages denote the subjective visual classification of the runs into the classes.

3.4 Noise

Since current quantum hardware is noisy, we tested how it affected the quantum agents' performance. We chose to use a simple circuit wide depolarising noise model[5], using a variable unified noise probability.

Due to the extreme computational cost of noisy training, we did not do a new hyperparameter search. Instead, we used the best models found in the noise-free experiments and retrained them using noisy versions of their circuits. Due to the non-determinism introduced by noisy circuits, we evaluated on 100 episodes.

The Frozen Lake environment could still be solved for unified noise probabilities $\leq 0.15\%$, for probabilities up to 0.3% some evaluation episodes could be solved and for higher ones the agent would never solve the environment.

For Grid Worlds, we analysed the first three map sizes. The quantum agents could still solve them under higher noise levels, but using increasingly suboptimal actions. This lead to significantly longer paths (as shown in Fig. 1c) but the agents still arrived at the goal since there were no holes.

The continuous state-space Cart Pole task was unsolvable using noisy circuits.

3.5 Lane Change

Lastly, we investigated the capabilities of our selected quantum architecture on the lane change environment. Due to the environment's time- and computational complexity, we only trained a quantum agent and compared it to the results reported in Ref. [4]. We used the same number of layers as in the previous environments, having four time less trainable parameters than the classical agent used in Ref. [4] (see Table 1). We carried out a minimal hyperparameter search using only half of the epochs used in Ref. [4]. The best configuration was then retrained using the full number of epochs.

[5] See `https://www.tensorflow.org/quantum/tutorials/noise` for further details.

Our results represent a first proof-of-concept for solving the environment with a quantum agent. Although we did not achieve comparable success rates to the classical agent (as reported in Table 1), the behaviour demonstrated by the quantum agent was consistent with observations on other environments when using suboptimal hyperparameters. Since we did only a minimal hyperparameter search, the used configuration should be considered suboptimal and with a more profound configuration exploration a quality comparable to the classical agent should be achievable.

4 Conclusion

We showed empirically that the used quantum architecture could solve environments with continuous or discrete state spaces, discrete action spaces, and immediate or delayed reward. The results for the Frozen Lake environment hint at a possible quantum advantage on sample complexity. Furthermore, we found that already small amounts of noise significantly deteriorate the achieved performance. Both quantum and classical agents showed similar behaviour regarding the state space scaling on grid world environments. Finally, we could solve episodes of the lane change environment, but did not yet achieve stable results. Nevertheless, this presents a promising behaviour on such a complex environment, with a three and a half times larger observation vector size compared to previous work. Overall, the quantum agent presented a promising potential for solving various QRL tasks.

It is important to note that due to the constraints imposed on the comparisons between the classical and quantum agents, our conclusions cannot be extrapolated to a more general case without further considerations.

References

[1] Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning. *Quantum*, 6:720, May 2022.

[2] V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund, S. Wölk, H. J. Briegel, and P. Walther. Experimental quantum speed-up in reinforcement learning agents. *Nature*, 591:229–233, 3 2021.

[3] A. Hamann, V. Dunjko, and S. Wölk. Quantum-accessible reinforcement learning beyond strictly epochal environments. *Quantum Machine Intelligence*, 3(2), August 2021.

[4] Matthias Nichting, Thomas Lobig, and Frank Köster. Case study on gap selection for automated vehicles based on deep q-learning. In *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, pages 252–257, 2021.

[5] Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko. Parametrized quantum policies for reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28362–28375. Curran Associates, Inc., 2021.

[6] Daniel Heß, Ray Lattarulo, Joshue Pérez, Julian Schindler, Tobias Hesse, and Frank Köster. Fast maneuver planning for cooperative automated vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1625–1632, 2018.