Variational Quantum Circuit Design for Quantum Reinforcement Learning on Continuous Environments

Georg Kruse[1], Theodora-Augustina Drăgan[3], Robert Wille[2] and Jeanette Miriam Lorenz[4]
[1]*Fraunhofer Institute for Integrated Systems and Device Technology, Erlangen, Germany*
[2]*Technical University of Munich, Department of Informatics, Munich, Germany*
[3]*Fraunhofer Institute for Cognitive Systems IKS, Munich, Germany*
[4]*Ludwig Maximilian University, Faculty of Physics, Munich, Germany*

Keywords: Quantum Reinforcement Learning, Variational Quantum Circuit Design, Continuous Actions Space.

Abstract: Quantum Reinforcement Learning (QRL) emerged as a branch of reinforcement learning (RL) that uses quantum submodules in the architecture of the algorithm. One branch of QRL focuses on the replacement of neural networks (NN) by variational quantum circuits (VQC) as function approximators. Initial works have shown promising results on classical environments with discrete action spaces, but many of the proposed architectural design choices of the VQC lack a detailed investigation. Hence, in this work we investigate the impact of VQC design choices such as angle embedding, encoding block architecture and postprocessesing on the training capabilities of QRL agents. We show that VQC design greatly influences training performance and heuristically derive enhancements for the analyzed components. Additionally, we show how to design a QRL agent in order to solve classical environments with continuous action spaces and benchmark our agents against classical feed-forward NNs.

1 INTRODUCTION

Quantum computing (QC) is a research field that is drawing a lot of attention due to the expected computational advantages. There are many possible application fields, such as quantum chemistry, cryptography, search algorithms and others (Dalzell et al., 2023). Moreover, quantum hardware is becoming increasingly accessible, with noisy intermediate scale quantum (NISQ) devices already being available. This creates the possibility of designing, implementing and benchmarking QC algorithms that are NISQ-friendly and comparing them against classical methods in order to assess potential quantum advantage at the current state of technology.

Quantum machine learning is one of the most promising candidates to show quantum advantage on NISQ hardware. Variational quantum algorithms (VQA) for supervised learning (Pérez-Salinas et al., 2020), for unsupervised learning (Benedetti et al., 2019; Du et al., 2020), and for reinforcement learning (Jerbi et al., 2021; Skolik et al., 2022) have been proposed and have already been implemented on NISQ machines. In supervised learning, neural networks (NN) were replaced with variational quantum circuits (VQC). While initial studies suggest that VQCs inhibit preferable properties such as better trainability (McClean et al., 2018), other analyses of important properties such as learning capability and generalization errors (Abbas et al., 2021; Caro et al., 2022; Banchi et al., 2021) remain inconclusive with regard to the advantages of quantum computation. Whether VQCs show reliable advantage over NNs therefore remains an open question (Qian et al., 2022).

The literature in the subdomain of quantum reinforcement learning (QRL) is yet sparse. Multiple approaches have been proposed and can be divided into several categories, ranging from quantum-inspired methods that mainly use classical computation, to purely quantum approaches that require fault-tolerant devices that are not yet available (Meyer et al., 2022). A main branch of research are hybrid quantum-classical algorithms that contain VQCs as function approximators whose trainable parameters are updated using classical methods, such as gradient descent. This branch of QRL, also referred to as VQC-based QRL, is of special interest since the

Kruse, G., Drăgan, T., Wille, R. and Lorenz, J.
Variational Quantum Circuit Design for Quantum Reinforcement Learning on Continuous Environments.
DOI: 10.5220/0012353100003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 393-400
ISBN: 978-989-758-680-4; ISSN: 2184-433X

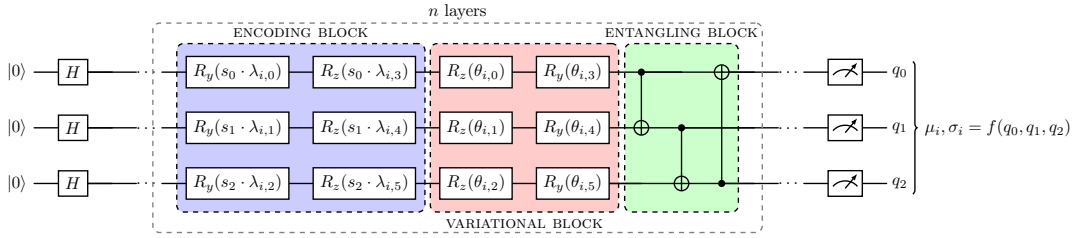


Figure 1: An exemplary VQC with three qubits consists of multiple layers *n*. Each layer has three blocks: an encoding block (with input state *s* and scaling parameters λ), a variational block (with variational parameters θ) and an entangling block (a daisy chain of CNOT entangling gates), followed by measurement and postprocessing steps.

possible beneficial properties of VQCs such as better trainability and generalization (Abbas et al., 2021; Banchi et al., 2021) can be transferred to RL algorithms. In this branch of research, quantum advantage has already been shown on an artificial benchmark (Jerbi et al., 2021). Recent works have mainly followed the architecture and hyperparameter choices of previous publications (Jerbi et al., 2021; Skolik et al., 2022), albeit these choices have been insufficiently investigated, making additional empirical studies necessary.

While the majority of QRL literature focuses on algorithms for environments with discrete action spaces, Wu et al. proposed a QRL solution for *quantum* continuous action space (CAS) environments (Wu et al., 2020). While Acuto et al. model QRL agents on *classical* CAS environments (Acuto et al., 2022), they still use additional NNs as postprocessing layers. This approach makes it difficult to distinguish between the contribution of quantum and classical part of the algorithm. Another open question for VQC-based QRL is therefore the adaptation to CAS environments without the use of additional classical NNs.

Based on the identified gaps in literature on the construction of QRL algorithms and the design of VQCs, our contributions are as follows: First, we show how to design VQC-based QRL for classical CAS environments without the use of additional NN as pre- or postprocessing layers. Second, we investigate VQC design choices by analysing the influence of angle embedding, encoding block design and readout strategies on the performance of the agent, benchmarked against two classical CAS OpenAI Gym environments, Pendulum-v1 and LunarLander-v2.

2 RELATED WORKS

In classical computing, one uses bits and strings of bits to encode information in one of two possible states 0 or 1, whereas in quantum computing the basic unit of information is the quantum bit – or, for short, the qubit. A qubit, opposed to a bit, can be in an infinite amount of states and is usually represented as a superposition of two basis states. Qubits are manipulated by quantum gates, which are operations that act on one or multiple qubits and transform their state, changing their probabilities. A series of multiple gates is called a quantum circuit, and if some parameters of these gates are trainable, it becomes a VQC. In this work, we focus on the subfield of QRL where the data is classical and the algorithm uses a hybrid quantum-classical approach, which contains VQCs as function approximators. The general architecture of a VQC used in this work is depicted in Fig. 1. It consists of three qubits, represented by three horizontal lines, which are initialized in the basis state $|000\rangle$. On these qubits act a sequence of quantum gates, indicated by the boxes on these lines, which change the state of the qubits. The gates are separated into three different blocks: A data encoding block, which transforms qubits depending on the classical input, a variational block with trainable variational gates, and an entangling block, where two qubit gates are used to entangle the qubits. Together the three blocks form a layer, which can be repeated several times. The repetition of a data encoding block in a VQC is known as data reuploading. At the end of the VQC, the qubits are measured and, if necessary, a classical post-processing step is applied to adapt the output of the measurement to the task at hand.

In the branch of QRL this work focuses on, classical RL algorithms are modified by replacing parts of the computational process with VQCs. Among these RL algorithms one can find deep Q-learning (Skolik et al., 2022), policy gradient (Jerbi et al., 2021), as well as Actor-Critic (AC) methods such as Proximal Policy Optimization (PPO) (Drăgan et al., 2022). In many works before and after the VQC a single NN linear layer is used for data pre- and post-processing (Acuto et al., 2022; Park et al., 2023).

Some works also focus on quantum environments: the authors of (Wu et al., 2020) propose a quantum Deep Deterministic Policy Gradient (DDPG) algo-

rithm and apply it to a CAS task, namely the quantum state generation. The solution is benchmarked on one-qubit and two-qubit cases. While the algorithm is successful, it is not presented how to adapt it to a classical environment, i.e., how to embed the data and interpret the measurements, which are the biggest challenges of CAS environments.

3 QUANTUM ACTOR-CRITIC

As has been shown in previous works, Q-learning (Skolik et al., 2022), Policy Gradient (Jerbi et al., 2021) algorithms, as well as actor-critic algorithms such as PPO (Drăgan et al., 2022) can be adapted to VQC-based QRL. Building on the works of Drăgan et al., in this section we show how to advance this approach to CAS environments without the need of additional classical pre- or postprossesing layers. This is especially important, since the class of problems state-of-the-art QRL focuses on is still quite simple. Differentiating between the contribution of quantum and classical part of the algorithm can therefore pose a difficult question. This is why we aim to reduce the complexity of the classical pre- and postprocessing to simple input- and output scalings, rather than entire NN layers as has been previously proposed in QRL solutions for CAS environments (Acuto et al., 2022; Park et al., 2023).

The PPO algorithm consists of an actor and a critic, which are each represented by one function approximator (classically a NN). The actor estimates the policy function $\pi_\Theta(s_t)$, while the critic estimates the value function $V_\Phi(s_t)$, both at a given state s_t at time step t.(in the following we drop the index t for simplicity).

3.1 Quantum Actor for Continuous Actions

In order to draw continuous actions from the policy function $\pi_\Theta(s)$, the output of the actor needs to be reparameterized. To calculate the value of a given continuous action a_i, the function approximator of the actor needs to compute two variables for each action, namely the mean μ_i and the variance σ_i of a normal distribution $\mathcal{N}$ from which the action a_i is then drawn $a_i \sim \mathcal{N}(\mu_i, \sigma_i)$.

We now consider the computation of the policy π_Θ of the actor with a VQC as function approximator instead of a NN. The actor VQC $U_\Theta(s)$ is parameterized by input scaling parameters λ_a, variational parameters θ and output scaling parameters w_{μ_i} and w_{σ_i}, where $\Theta = (\lambda_a, \theta, w_{\mu_i}, w_{\sigma_i})$. To compute action a_i as a factorized Gaussian, the mean μ_i and standard deviation σ_i are calculated based on the observables O_{μ_i} and O_{σ_i} as follows:

$$\mu_i = \langle 0^{\otimes n} | U_\Theta(s)^\dagger O_{\mu_i} U_\Theta(s) | 0^{\otimes n} \rangle \cdot w_{\mu_i} \quad (1)$$

and

$$\sigma_i = \exp\big(\langle 0^{\otimes n} | U_\Theta(s)^\dagger O_{\sigma_i} U_\Theta(s) | 0^{\otimes n} \rangle \cdot w_{\sigma_i} \big). \quad (2)$$

Since O_{μ_i} and O_{σ_i} are arbitrary Pauli operators, the output values for mean and variance can not scale beyond the interval of [-1, 1]. Therefore the classical scaling parameters w_{μ_i} and w_{σ_i} are crucial in order to apply VQC-based RL to classical CAS environments.

3.2 Quantum Critic for Value Estimation

To retrieve the information for the value estimate of the critic, we follow the approach of (Skolik et al., 2022). Let $U_\Phi(s)$ be the critic VQC parameterized by $\Phi = (\lambda_c, \phi, w_{O_{v_i}})$, where analogously to the actor VQC, λ_c are the parameters used for input scaling, ϕ are the variational parameters, and $w_{O_{v_i}}$ refers to the output scaling parameters. Then the value of a given state s is computed using Eq. 3

$$V_\Phi(s) = \sum^{n} \langle 0^{\otimes n} | U_\Phi(s)^\dagger O_{v_i} U_\Phi(s) | 0^{\otimes n} \rangle \cdot w_{O_{v_i}} \quad (3)$$

We obtain the value of $V_\Phi(s)$ by either a single or a sequence of observables O_{v_i} acting on n qubits. We introduce another scaling parameter $w_{O_{v_i}}$, since the value estimate of the critic also needs to scale beyond the interval of $[-1, 1]$ for most RL tasks. In the following we discuss the choice of the number of VQC layers n and demonstrate how its value, as well as the choice of the observables, can greatly influence QRL performance.

4 VARIATIONAL QUANTUM CIRCUIT DESIGN

Due to the small number of empirical studies in the field of QRL, the degrees of freedom in VQC design choices are enormous. In this work we therefore need to restrict our investigations: The basis of our analysis will be the widely-used hardware efficient Ansatz proposed by Jerbi et al. enhanced using data reuploading as proposed by Skolik et al. (Jerbi et al., 2021; Skolik et al., 2022). Our only modification to this Ansatz will be the replacement of CZ entangling gates with CNOT entangling gates. This is due

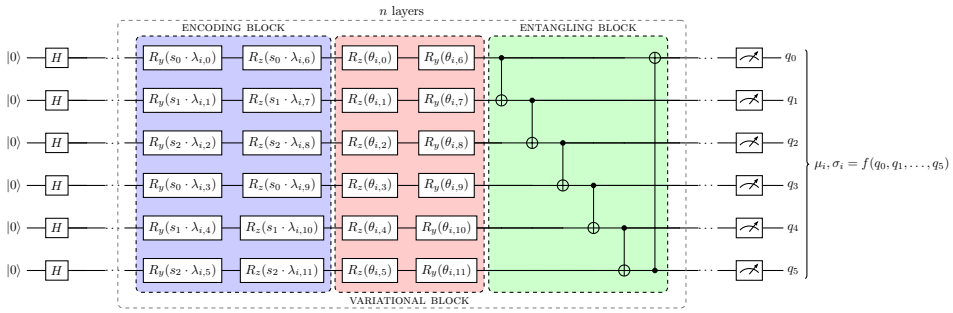


Figure 2: An example of a *stacked* VQC with six qubits applied for an environment with observation space of size three. The encoding block is repeated (stacked) vertically two times such that each state feature s_i is encoded twice on two distinct qubits in each layer n. The other blocks follow the design of Fig. 1.

to the fact that chain CZ entanglement may lead to large amounts of parameters which do not influence the output of the VQC, as the number of qubits increases.

The basic architecture of the used VQC is depicted in Fig. 1. Each layer of the VQC consists of three blocks: A data encoding block, a variational block and an entangling block. After n such layers are concatenated, measurements are conducted, followed by an additional postprocessing step. In this work we investigate three design choices for this VQC. First, we evaluate the influence of different preprocessing steps on the classical state s used for angle embedding. Second, we propose a new encoding block architecture and benchmark it against the basic encoding block. Third, we analyse the influence of different observables and postprocessing steps on the training performance.

4.1 Angle Embedding

Data encoding greatly influences the behaviour of VQCs (Schuld et al., 2021). One of the ways the classical environment state s can be encoded into a quantum state suitable for a VQC is angle embedding. This is done using one or more rotation gates (ref. Fig. 1). Since these gates have a periodicity of 2π while the observation space of a classical environment can be outside this interval, various works have proposed to encode each feature s_i of the classical environment state s as $\arctan(s_i \cdot \lambda_i)$ (Skolik et al., 2022), where λ_i denotes a classical trainable scaling parameter. This encoding has the caveat that for classical observation spaces with large absolute feature values, trigonometric transformations such as *arctan* and *sigmoid* will make the features almost indistinguishable for the QRL agent. To overcome this caveat, we propose to previously normalize the features s_i to an interval of $[-\frac{\pi}{2}, \frac{\pi}{2}]$. The normalized features $\hat{s}_i$ are encoded into the rotation gates as $\hat{s}_i \cdot \lambda_i$, either with or without an additional nonlinear transformation (ref. Fig. 3).

4.2 Encoding Block

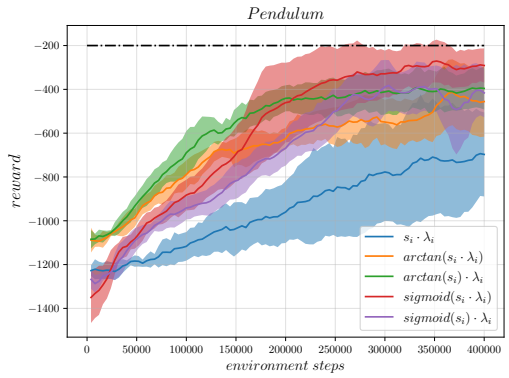
Previous works, which do not use additional NNs for pre- or postprocessing, generally design the encoding block of the VQC such that each feature of the observation space is encoded into one qubit using angle embedding. This strategy limits the size of the VQC to the observation space size of the task at hand, limiting the potential of VQCs. This problem is accompanied by the fact that an increase of the number of layers has previously been shown to improve training performance only until a certain threshold (Skolik et al., 2022).

To overcome this issue we propose a novel data encoding approach in order to increase the number of exploitable qubits: Instead of encoding each feature of the state s using angle embedding only once, we stack s such that each feature is encoded several times. An illustration of this *stacked* VQC is shown in Fig. 2. This architecture enables VQC-based QRL agents to scale beyond the previous VQC sizes, potentially enhancing their training capabilities due to a higher amount of trainable parameters without the need of additional layers.

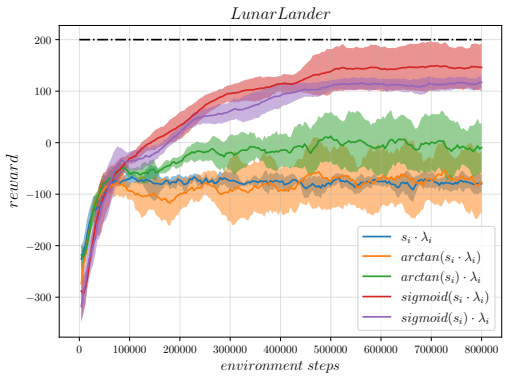
4.3 Observables and Postprocessing

The choice of observables and postprocessing steps, jointly referred to in this work as readout configuration, has previously been shown to be crucial for the performance of the agent on discrete learning tasks (Meyer et al., 2023). Therefore, the choice of the observables O_{μ_i} and O_{σ_i} for the actor is investigated in this work.

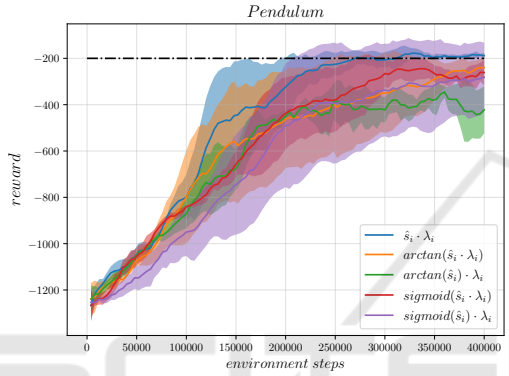
For the actor VQC, we compare single qubit observables for the mean $O_{\mu_i} = Z_i$ and variance $O_{\sigma_i} = Z_{i+1}$, where Z_i are Pauli-Z operators on the respective



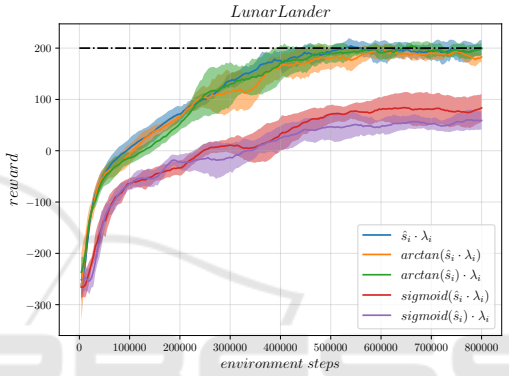
(a) 3 qubits, 3 layers, M_1 readout, no normalization.



(b) 6 qubits, 6 layers, M_1 readout, no normalization.



(c) 3 qubits, 3 layers, M_1 readout, with normalization.



(d) 6 qubits, 6 layers, M_1 readout, with normalization.

Figure 3: Evaluation of encoding strategies: Each angle embedding consists of rotation gates R_y and R_z (ref. Fig. 1) and the same readout M_1 (compared in Fig. 5) is used for all runs. The target reward of the environments are indicated by the dotted black line. In Fig. 3a and 3b the training curves for Pendulum-v1 and LunarLander-v2 are depicted without a previous normalization of the state s. In Fig. 3c and 3d the state s is previously normalized to an interval of $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Each solid line represents the mean of five seeds, the shaded area indicates the standard deviation.

qubit. We compare these single-qubit observables to multi-qubit observables, as well as to a combination of the two approaches (ref. Tab. 1). Since the expectation value of the unscaled observables O_{μ_i} and O_{σ_i} lie in $[-1,1]$, while the continuous action space of a given environment can potentially lie in $(-\infty,\infty)$, we use one trainable parameter w_i for each observable as postprocessing step. Previous work has already analyzed the impact of non-trainable scaling parameters for Q-learning (Skolik et al., 2022), so we will not investigate this design choice here.

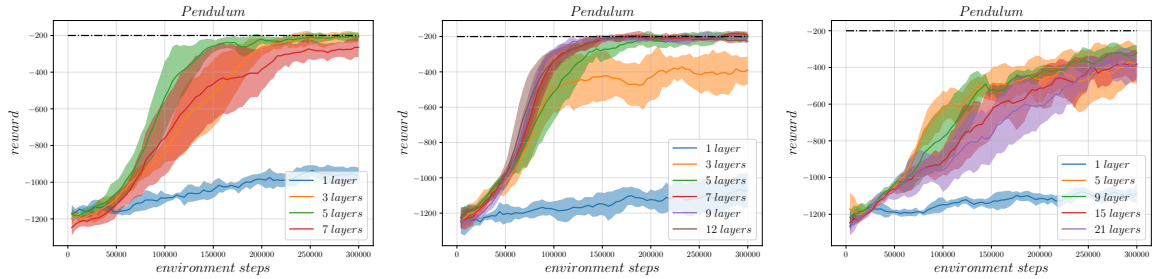
For the critic, the observable O_{v_i} is either a single Pauli-Z operator on the first qubit, the sum of single Pauli-Z operators on all qubits or a multi qubit measurement on all qubits. As postprocessing step we introduce for each respective expectation value a trainable scaling parameter w_i. In the following we investigate the impact of different readout configurations M - with varying observables and postprocessing steps - on the training performance of the QRL agent.

Table 1: Table of all readout configurations M_1 to M_7 in Fig 5, where i is the index of the action a_i and j is the index of the qubit.

	O_{μ_i}	O_{σ_i}	O_v
M_1	$Z_i w_{\mu_i}$	$Z_{i+1} w_{\sigma_i}$	$\sum(Z_j \cdot w_{v_j})$
M_2	$Z_i w_{\mu_i}$	$Z_{i+1} w_{\sigma_i}$	$Z_0 \cdot w_{v_0}$
M_3	$Z_i w_{\mu_i}$	$Z_{i+1} w_{\sigma_i}$	$\prod(Z_j) \cdot w_{v_0}$
M_4	$Z_i Z_{i+1} w_{\mu_i}$	$Z_{i+2} w_{\sigma_i}$	$\sum(Z_j \cdot w_{v_j})$
M_5	$Z_i Z_{i+1} w_{\mu_i}$	$Z_{i+2} w_{\sigma_i}$	$Z_0 \cdot w_{v_0}$
M_6	$Z_i Z_{i+1} w_{\mu_i}$	$Z_{i+2} w_{\sigma_i}$	$\prod(Z_j) \cdot w_{v_0}$
M_7	$Z_i Z_{i+1} Z_{i+2} w_{\mu_i}$	$Z_{i+3} Z_{i+4} Z_{i+5} w_{\sigma_i}$	$\sum(Z_j \cdot w_{v_j})$
M_8	$Z_i Z_{i+1} Z_{i+2} w_{\mu_i}$	$Z_{i+3} Z_{i+4} Z_{i+5} w_{\sigma_i}$	$Z_0 \cdot w_{v_0}$
M_9	$Z_i Z_{i+1} Z_{i+2} w_{\mu_i}$	$Z_{i+3} Z_{i+4} Z_{i+5} w_{\sigma_i}$	$\prod(Z_j) \cdot w_{v_0}$

5 NUMERICAL RESULTS

In this section we analyze the influence of the VQC design choices on two CAS environments with different observation space sizes, action space sizes and



(a) 3 qubits, $\hat{s}_i \cdot \lambda_i$ encoding, M_1 readout. (b) 6 qubits, $\hat{s}_i \cdot \lambda_i$ encoding, M_1 readout. (c) 9 qubits, $\hat{s}_i \cdot \lambda_i$ encoding, M_1 readout.

Figure 4: Comparison of different VQC encoding block sizes: In all runs the same angle embedding ($\hat{s}_i \cdot \lambda_i$) and readout ($M_1$, ref. Tab. 1) is used. The state features s_i are encoded one, two or three times in Fig. 4a, 4b and 4c respectively as depicted in Fig. 2. Each solid line represents the mean of five seeds, the shaded area indicates the standard deviation.

difficulties: The Pendulum-v1 environment with observation space of size three and one continuous action and the LunarLander-v2 environment with observation space of size eight and two continuous actions. On Pendulum-v1, we benchmark VQCs with 3, 6 and 9 qubits and evaluate different design choices. On LunarLander-v2, we select the 6 most informative features of the 8 features of the observation space. This is because for the used VQC architecture, the variance of the expectation values of the observables starts to vanish quickly, hindering training already at eight qubits.

5.1 Angle Embedding

In Fig. 3 various angle embedding strategies are evaluated on the two environments. In Figs. 3a and 3b, the state features s_i are encoded into two rotation gates R_y and R_z using *arctan* or *sigmoid* functions and scaling parameters λ_i. None of these encodings enable the VQC-based QRL agents to solve the environments. On both benchmarks the best performing agents utilize a $sigmoid(s_i \cdot \lambda_i)$ encoding, but nevertheless fail to reach the target rewards.

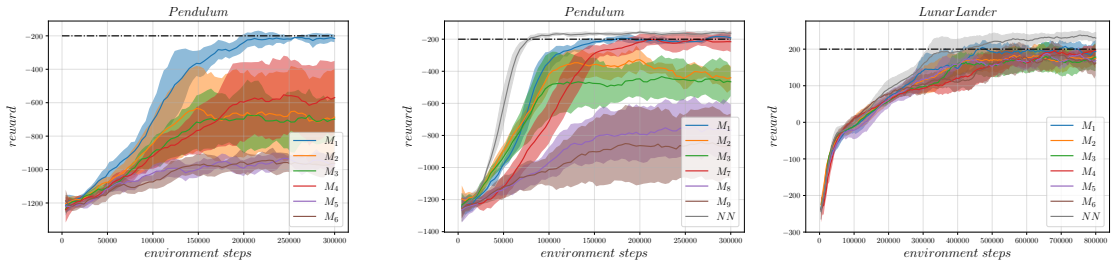
In classical RL, input states are generally normalized to the interval of $[-1, 1]$ in order to enhance training performance. Following this idea of previous normalization of states, in Fig. 3c and 3d s_i is previously normalized to the interval of $[-\pi/2, \pi/2]$ to $\hat{s}_i$. In Fig. 3c it can be seen that the encoding without any nonlinear function scaling outperforms all other encodings, while in Fig. 3d the encodings using *arctan* functions perform similarly well.

While the use of nonlinear functions such as *arctan* and *sigmoid* is widely spread across literature, we show that they do not enhance training performance, but instead can even lead to poorer results. Instead, simple normalization as utilized in classical RL, combined with a trainable parameter for each input feature shows the best performance across runs. Hence, in the further comparison, the encoding $\hat{s}_i \cdot \lambda_i$ is used and referred to as basic encoding.

5.2 Encoding Block

Previous works, which do not use additional NNs for pre- or postprocessing, have always used a VQC of the same size as the observation space of the classical environment. In order to evaluate the usage of different sized encoding blocks, in Figs. 4 we compare the training performance on Pendulum-v1 with a 3, 6 and 9 qubit VQC, where each feature of the state s is encoded one, two and three times respectively (ref. Fig. 2). All VQCs fail to train with a single layer, even though the amount of trainable parameters in the variational block increases from 6 to 12 and 18. As data reuploading is introduced by using more than one layer, training performance is improved across all architectures. But simply introducing more parameters by adding more layers has previously been shown to only improve performance until a certain threshold (Skolik et al., 2022). This can be seen in Fig. 4a, where performance reaches a peak at five layers. Skolik et al. suggest that this behaviour occurs because at a certain number of layers, training begins to be hindered due to overparameterization of the VQC. Interestingly, the amount of layers required for successful training increases with the amount of qubits: For larger VQCs with six qubits best performance is observed for nine layers (Fig. 4b), while the nine qubit VQC fails to solve the environment in the given time frame (Fig. 4c). Therefore, overparameterization can not be interpreted as an absolute number of trainable parameters, but rather depends on the number of qubits used: Our results indicate that greater qubit numbers also require greater numbers of trainable parameters.

It has been previously shown that for small VQCs



(a) 3 qubits, 5 layers, $\hat{s}_i \cdot \lambda_i$ encoding. (b) 6 qubits, 7 layers, $\hat{s}_i \cdot \lambda_i$ encoding. (c) 6 qubits, 7 layers, $\hat{s}_i \cdot \lambda_i$ encoding.

Figure 5: Comparison of readout configurations (ref. Tab. 1) and benchmarking against best performing NNs: In Fig. 5b and 5c the black lines indicate the best performing classical NN with two hidden layers with 64 neurons each and ReLU activation functions based on the hyperparameter search. Each solid line represents the mean of five seeds, the shaded area indicates the standard deviation.

the amount of trainable parameters required for successful training is lower than for classical NNs (Drăgan et al., 2022). Our findings suggest that this phenomenon is restricted to small VQCs and does not apply to larger VQCs. Moreover, the vanishing gradients start to hinder training already at nine qubits (ref. Fig. 4c).

5.3 Observables and Postprocessing

In Fig. 5 the influence of different choices of observables and postprocessings for actor and critic is shown. The different readout configurations are listed in Tab. 1. On the Pendulum-v1 environment the choice of observable is crucial for the success of training (Fig. 5a). The M_1 readout configuration is the only configuration which leads to successful training with a three qubit VQC. Also in Fig. 5b, the M_1 readout performs best for the *stacked* VQC, followed by M_7, the only other configuration leading to successful training. Finally, in Fig. 5c different readout configurations for LunarLander-v2 are shown. Here no clear trend can be observed.

Our results show that observables and postprocessing steps can be crucial for training performance in some cases, while in others barely influence the performance of the agents. Only the M_1 readout configuration has no negative influence across all experiments.

5.4 Benchmark Against Classical Agents

Finally we perform an extended hyperparameter search for the classical RL agents and benchmark the best performing classical RL agents against the QRL agents in Figs. 5b and 5c. We evaluated 117 different classical agents for Pendulum-v1 and 36 for LunarLander-v2. On both benchmarks the best performing NNs have two hidden layers with 64 neurons each and ReLU activation functions, resulting in 4416 and 4736 trainable parameters on the two benchmarks, while the QRL agents have 176 and 178 trainable parameters. On one hand, the performance gap could be explained by the difference of the number of trainable parameters. On the other hand, it remains to be shown if the beneficial properties such as better trainability (Abbas et al., 2021) and generalization (Banchi et al., 2021) also hold for larger VQCs with comparable amounts of trainable parameters.

6 CONCLUSIONS

In this work we showed how to construct a quantum reinforcement learning agent for classical environments with continuous action spaces based on a hybrid quantum-classical algorithm that employs variational quantum circuits as function approximators. Our approach does not require any additional classical neural network layers as pre- or postprocessing steps. Instead, only trainable scaling parameters are required in order to adapt the output of the variational quantum circuit to the size of the continuous action space.

Additionally, we investigated several variational quantum circuit design choices with respect to their influence on training performance. While nonlinear functions such as *arctan* have been widely used throughout quantum reinforcement learning literature for angle embedding, we show in our experiments that such functions actually hinder training performance. Instead, normalization - in combination with trainable scaling parameters - yields the best training results.

The number of qubits of previous designs of variational quantum circuits was limited to the size of the

observation space due to angle embedding. We proposed a new encoding block architecture - *stacked* VQC - which allows the utilization of additional qubits, resulting in improved training performance. It has been previously shown that an increase of the number of layers improves training performance only until a threshold (Skolik et al., 2022). We reveal a similar trend: an increase of the number of qubits substantially improves training performance, but also only until a certain limit. Our work indicates that current VQC architectures therefore are limited both in the number of layers, as well as in the amount of qubits, and thus dictate both the depth and the width of the circuit, respectively. While we investigated and enhanced current variational quantum circuit design choices, future work should aim to further improve upon these results as well as explore novel circuit architectures in order to bridge the performance gap between QRL and RL.

ACKNOWLEDGEMENTS

The research is part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

REFERENCES

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., and Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409.

Acuto, A., Barilla, P., Bozzolo, L., Conterno, M., Pavese, M., and Policicchio, A. (2022). Variational quantum soft actor-critic for robotic arm control. *arXiv preprint arXiv:2212.11681*.

Banchi, L., Pereira, J., and Pirandola, S. (2021). Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2(4):040321.

Benedetti, M., Garcia-Pintos, D., Perdomo, O., Leyton-Ortega, V., Nam, Y., and Perdomo-Ortiz, A. (2019). A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1).

Caro, M. C., Huang, H.-Y., Cerezo, M., Sharma, K., Sornborger, A., Cincio, L., and Coles, P. J. (2022). Generalization in quantum machine learning from few training data. *Nature Communications*, 13(1).

Dalzell, A. M., McArdle, S., Berta, M., Bienias, P., Chen, C.-F., Gilyén, A., Hann, C. T., Kastoryano, M. J., Khabiboulline, E. T., Kubica, A., Salton, G., Wang, S., and Brandão, F. G. S. L. (2023). Quantum algorithms: A survey of applications and end-to-end complexities.

Drăgan, T.-A., Monnet, M., Mendl, C. B., and Lorenz, J. M. (2022). Quantum reinforcement learning for solving a stochastic frozen lake environment and the impact of quantum architecture choices. *arXiv preprint arXiv:2212.07932*.

Du, Y., Hsieh, M.-H., Liu, T., and Tao, D. (2020). Expressive power of parametrized quantum circuits. *Phys. Rev. Res.*, 2:033125.

Jerbi, S., Gyurik, C., Marshall, S., Briegel, H., and Dunjko, V. (2021). Parametrized quantum policies for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28362–28375.

McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1).

Meyer, N., Scherer, D., Plinge, A., Mutschler, C., and Hartmann, M. (2023). Quantum policy gradient algorithm with optimized action decoding. In *International Conference on Machine Learning*, pages 24592–24613. PMLR.

Meyer, N., Ufrecht, C., Periyasamy, M., Scherer, D. D., Plinge, A., and Mutschler, C. (2022). A survey on quantum reinforcement learning.

Park, S., Kim, J. P., Park, C., Jung, S., and Kim, J. (2023). Quantum multi-agent reinforcement learning for autonomous mobility cooperation. *IEEE Communications Magazine*.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., and Latorre, J. I. (2020). Data re-uploading for a universal quantum classifier. *Quantum*, 4:226.

Qian, Y., Wang, X., Du, Y., Wu, X., and Tao, D. (2022). The dilemma of quantum neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Schuld, M., Sweke, R., and Meyer, J. J. (2021). Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430.

Skolik, A., Jerbi, S., and Dunjko, V. (2022). Quantum agents in the gym: a variational quantum algorithm for deep q-learning. *Quantum*, 6:720.

Wu, S., Jin, S., Wen, D., and Wang, X. (2020). Quantum reinforcement learning in continuous action space. *arXiv preprint arXiv:2012.10711*.