

# Multi-Objective Evolutionary Architecture Search for Parameterized Quantum Circuits

Li Ding <sup>1,\*</sup>  and Lee Spector <sup>1,2,\*</sup> 

<sup>1</sup> Manning College of Information & Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01002, USA

<sup>2</sup> Department of Computer Science, Amherst College, Amherst, MA 01002, USA

\* Correspondence: liding@umass.edu (L.D.); lspector@amherst.edu (L.S.)

**Abstract:** Recent work on hybrid quantum-classical machine learning systems has demonstrated success in utilizing parameterized quantum circuits (PQCs) to solve the challenging reinforcement learning (RL) tasks, with provable learning advantages over classical systems, e.g., deep neural networks. While existing work demonstrates and exploits the strength of PQC-based models, the design choices of PQC architectures and the interactions between different quantum circuits on learning tasks are generally underexplored. In this work, we introduce a Multi-objective Evolutionary Architecture Search framework for parameterized quantum circuits (MEAS-PQC), which uses a multi-objective genetic algorithm with quantum-specific configurations to perform efficient searching of optimal PQC architectures. Experimental results show that our method can find architectures that have superior learning performance on three benchmark RL tasks, and are also optimized for additional objectives including reductions in quantum noise and model size. Further analysis of patterns and probability distributions of quantum operations helps identify performance-critical design choices of hybrid quantum-classical learning systems.

**Keywords:** quantum computing; quantum machine learning; reinforcement learning; evolutionary algorithms; multi-objective optimization



**Citation:** Ding, L.; Spector, L. Multi-Objective Evolutionary Architecture Search for Parameterized Quantum Circuits. *Entropy* **2023**, *25*, 93. <https://doi.org/10.3390/e25010093>

Academic Editor: Andreas Wichert

Received: 16 November 2022

Revised: 27 December 2022

Accepted: 28 December 2022

Published: 3 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Noisy Intermediate-Scale Quantum (NISQ) technology [1] reveals the potential of quantum systems to have significant and reliable advantage in solving computing tasks over classical systems. One of the most encouraging areas of quantum computing research is hybrid quantum-classical learning systems, which utilizes parameterized quantum operations together with classical optimization methods to solve the designed learning tasks. In particular, a standard way to model hybrid systems is to use parameterized quantum circuits (PQCs) [2]. Recent work has demonstrated that PQC-based systems are able to handle a variety of supervised and unsupervised tasks such as classification [3–6] and generative modeling [7–10]. Moreover, some recent work [11,12] further demonstrates that PQCs can be used to construct parameterized quantum policies to solve more complex reinforcement learning problems, and shows provable learning advantages in some specific RL environments over classical methods such as deep neural networks (DNNs).

An essential element of PQC-based systems is the design and structural arrangement of quantum and classical operations. Prior work [11–13] has shown significant impact of utilizing some key components, as well as analyzed different design choices through empirical studies. However, the development of best-performing architectures of hybrid quantum learning systems still relies on human ingenuity. In classical learning systems, architecture search methods that aim to automate the process of discovering and evaluating the architecture of complex systems have been extensively explored in the past decade, e.g., neural architecture search (NAS) [14]. In the context of quantum computing, while recent work [15] made early attempts to apply evolutionary NAS methods [16] in searching

for PQC architectures, their method does not fully consider the interactions between quantum gates, which results in much complexity and computational redundancy. Overall, automated search and optimization of architectures of hybrid quantum learning systems have not been sufficiently explored.

In this work, we aim to explore using multi-objective genetic algorithms [17] to automatically design the architecture of hybrid quantum-classical learning systems that can solve complex RL problems. We start by adopting the ideas of successful approaches in NAS and QAS using genetic algorithms [15,16], which have more flexible architecture search spaces and require less prior knowledge than other gradient-based approaches such as RL. We propose MEAS-PQC, a Multi-objective Evolutionary Architecture Search framework for PQC-based quantum learning models, which uses a multi-objective genetic algorithm with quantum-specific configurations to perform efficient searching of optimal PQC architectures. More specifically, our method improves prior work [15] in three ways: (1) we adopt the Uniform Mutation by Addition and Deletion (UMAD [18]) mutation operator to enable searching for architectures with variable lengths; (2) our method specifically considers equivalent quantum operations in the architecture and eliminates unnecessary redundancy; (3) we further extend quantum architecture search to a more practical setting by forming a multi-objective optimization problem, which considers specific factors in real-world quantum computing, e.g., interactions between quantum gates and quantum noise, in addition to learning performance.

To validate the proposed method, we implement our method on three benchmark RL environments from OpenAI Gym [19], which has been extensively used in RL research. The experiments consider both single- and multi-objective settings. For single-objective, our method outperforms prior work by a significant margin on learning performance, and we also perform qualitative analysis on the results to further interpret and discover the essential design choices of PQC-based systems. For multi-objective, we observe that our method is able to search for architectures that have good learning performance as well as meet computational constraints such as quantum noise and model size.

To summarize, the contributions of this work are two folds: (1) we propose MEAS-PQC, which improves prior work [15] by adding performance-critical components including UMAD mutation and simplification of equivalent quantum gates; (2) we form quantum architecture search as a multi-objective optimization problem, and shows that our method can search for PQC architectures that not only have good learning performance, but are also more efficient and robust to potential quantum noise.

## 2. Preliminaries and Related Work

In this section, we introduce some basic concepts of quantum computation that are closely related to this work, and give a detailed description of parameterized quantum circuits and other related work.

### 2.1. Quantum Computation Preliminaries

A complex Hilbert space of  $2^n$  dimensions usually serves a general representation of a multi-qubit quantum system with  $n$  qubits. The quantum state of the system is mathematically written as a vector  $|\psi\rangle$ , which has unit norm  $\langle\psi|\psi\rangle = 1$ .  $\langle\psi|$  is the conjugate transpose and  $\langle\psi|\psi\rangle$  represents the inner-product under the bra-ket notation. We represent the computational basis states of a multi-qubit system as tensor-products of single-qubit basis states. For example, a two-qubit state  $|01\rangle = |0\rangle \otimes |1\rangle$ , where  $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  and  $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  are two single-qubit basis states.

A quantum gate is mathematically represented as a unitary operator  $U$  acting on qubits. In quantum machine learning, a few common quantum gates are extensively used,

e.g., the single-qubit rotation operators  $R_x, R_y, R_z$ . Given a rotation angle  $\theta$ , the matrix representations of rotation operators are

$$R_x(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}. \tag{1}$$

An *entangled state* (or *entanglement*) is a quantum state of a composite system that can not be written as a tensor-product of the states of its components. An entanglement can be created by applying appropriate 2-qubit gates. In this work, we mainly use controlled-Pauli-Z gates to generate entanglement. The Controlled-Z gate (CZ) is a symmetric gate,

$$CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z \tag{2}$$

A projective measurement of quantum states is described by an observable,  $M$ , which is a Hermitian operator on the state space of the quantum system being observed. The observable has a spectral decomposition

$$M = \sum_m m P_m, \tag{3}$$

where  $P_m$  is the projector onto the eigenspace of  $M$  with eigenvalue  $m$ . Upon measuring the state  $|\psi\rangle$ , the probability of getting result  $m$  is given by

$$p(m) = \langle \psi | P_m | \psi \rangle, \tag{4}$$

and the expectation value of the measurement is

$$E(M) = \sum_m m \cdot p(m) = \langle \psi | M | \psi \rangle. \tag{5}$$

For a more detailed introduction to quantum computation and its basic concepts, we refer the readers to Nielsen and Chuang [20].

### 2.2. Parameterized Quantum Circuits

Given a fixed  $n$ -qubit system, a parameterized quantum circuit (PQC) is defined by a unitary operation  $U(s, \theta)$  that acts on the current quantum states  $s$  considering the trainable parameters  $\theta$ . In this work, we mainly consider two types of PQCs: variational PQCs (V-PQCs) [2,21] and data-encoding PQCs (D-PQCs) [13,22], which are widely used in quantum machine learning. The V-PQCs are composed of single-qubit rotations  $R_x, R_y, R_z$  with the rotation angles as trainable parameters. Similar to classical learning models, the PQC parameters can also be optimized with gradient-based algorithms. The task of learning is mathematically expressed as the minimization of a defined objective function  $L(\theta)$  with respect to the parameter  $\theta$  of a V-PQC. The gradient descent algorithm updates  $\theta$  towards the direction of descending the loss

$$\theta \leftarrow \theta - \eta \nabla_{\theta} L, \tag{6}$$

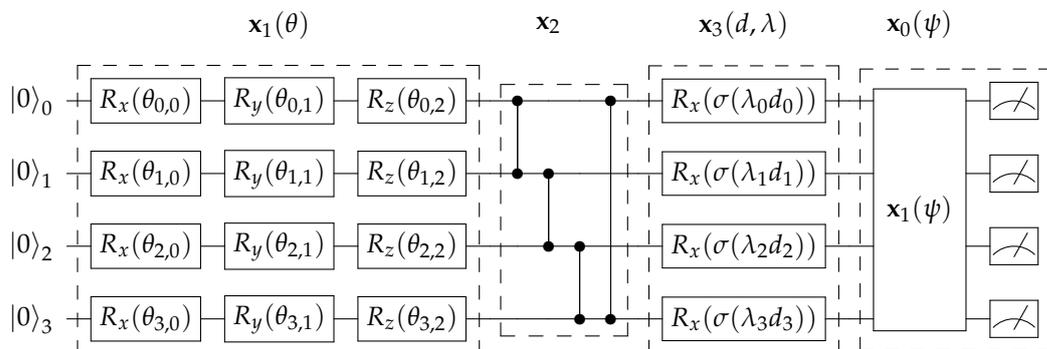
where  $\eta$  is the learning rate and  $\nabla_{\theta} L$  is the gradient vector. The gradient can be estimated through a finite difference of partial derivatives

$$\frac{\partial L}{\partial \theta_i} \approx \frac{L(\theta + \Delta e_i) - L(\theta - \Delta e_i)}{2\Delta} \tag{7}$$

where  $\Delta$  is a small constant and  $e_i$  is the Cartesian unit vector in the  $i$  direction.

The D-PQCs have a similar structure with rotations, but the angles are the input data  $d$  scaled by a trainable parameter  $\lambda$ . The input data are usually normalized to have similar scales. For example, in our experiments, the input data are state variables in the

RL environments, and each of them is normalized according to its min-max values. The structures of both PQCs are depicted in Figure 1, which we describe in detail later in Section 3.2.



**Figure 1.** Illustration of a simple 4-qubit architecture in the MEAS-PQC search space. The architecture (with genome encoding 1 – 2 – 3 – 0) is composed of 4 operations: (1) Variational PQC ( $x_1$ ) performs rotations on each qubits according to parameters  $\theta$ ; (2) Entanglement ( $x_2$ ) performs circular entanglement to all the qubits; (3) Data-encoding PQC ( $x_3$ ) performs rotations on each qubit according to the input data  $d$ , scaling parameter  $\lambda$ , and activation function  $\sigma$ ; (4) Measurement ( $x_0$ ) adds another Variational PQC ( $x_1$ ) and performs measurement to obtain the observable values.

Recent work [11] proposes to use an alternating-layered architecture [13,22] to implement parameterized quantum policies for RL, which basically applies an alternation of V-PQC (followed by an entanglement) and D-PQC till the target depth, while this architecture is simple and effective, it is obvious to see that this general design can be easily modified and probably improved by changing the placement of its components. In this work, we aim to optimize the design of such PQC-based systems with architecture search methods.

### 2.3. Quantum Architecture Search

Early research [23] has shown the usage of genetic programming to solve specific quantum computing problems from an evolutionary perspective. Common architecture search approaches, such as greedy algorithms [24,25], evolutionary algorithms [26–29], reinforcement learning [30,31], and gradient-based learning [32] have also been attempted to solve tasks such as quantum control, variational quantum eigensolver, quantum error mitigation, and entanglement purification. However, most of these approaches target optimizing either specific pure quantum circuits or single-qubit quantum operations, instead of more complex multi-qubit hybrid systems.

More recently, a few approaches have been proposed to optimize the architectures involving parameterized quantum circuits. Grimsley et al. [24] proposed a method that iteratively adds parameterized gates and re-optimizes the circuit using gradient descent. Ostaszewski et al. [33] proposed an energy-based searching method for optimizing both the structure and parameters for single-qubit gates and demonstrated its performance on a variational quantum eigensolver. Ding and Spector [15] proposed EQAS-PQC, which applies existing evolutionary NAS method to perform search on PQC architectures, while EQAS-PQC shows promising and encouraging results, their method suffers considerable computational redundancy in the searched architectures. In this work, we take steps further and propose a more general multi-objective architecture search framework that is specialized for hybrid quantum-classical systems.

## 3. Method

We propose MEAS-PQC, an evolutionary framework that uses multi-objective genetic algorithms to perform quantum architecture search for PQC-based hybrid quantum-classical systems. In this section, we describe the major components of MEAS-PQC in-

cluding the genetic algorithm, encoding scheme, search process, and the multi-objective optimization framework in detail.

### 3.1. Genetic Algorithm for Quantum Architecture Search

Genetic algorithms (GAs) refer to a class of population-based computational paradigms, which simulate the natural evolution process to evolve programs by using artificial genetic operations (e.g., crossover and mutation) to optimize fitness or objective functions. It has been successfully adopted in recent work for optimizing complex systems such as neural networks [16,34,35]. In the perspective of GAs, we view the architectures of quantum circuits as *phenotypes*, and define representations as *genotypes* on which the genetic operations can be easily applied. Similar to many other genetic algorithms, MEAS-PQC iteratively generates a population of candidates (architectures) through genetic operations on the given parents, and selects parents for the next generation based on fitness evaluation.

In this work, we adopt the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [36] to optimize the search process. NSGA-II is an evolutionary algorithm that has been successfully employed in various single- and multi-objective optimization problems including NAS [16]. The algorithm has the following procedures:

1. Perform a non-dominated sorting in the population of quantum architectures and classify them according to an ascending level of non-domination based on objectives.
2. Use crowding distance, which is related to the density of solutions with similar objective metrics, to perform Crowding-sort that makes the population less dense.
3. Generate offspring using crowded tournament selection, then apply genetic operators such as mutation and crossover.

In the context of this work, for single-objective optimization, MEAS-PQC generates a population of PQC architectures in the search space, and iteratively selects the ones with good learning performance. For multi-objective, it uses an elitist principle to select the non-dominated candidates using the above procedure. The benefit of using NSGA-II is that it automatically selects a good set of quantum architectures while balancing different objectives. For multi-objective optimization, one naive way is to sum all the objectives numerically, which requires to carefully weigh different objectives to avoid dominance of objectives that have large value scale. Instead, our method does not need to perform such tuning, which is more efficient and robust.

### 3.2. Encoding Scheme and Search Space

An encoding scheme is the interface for abstracting the architectures to genomes, where the genes are representations for different quantum operations. We follow the general encoding scheme as proposed in Ding and Spector [15], which consists of some basic functional quantum circuits on a single qubit or multiple qubits that has been widely adopted in prior work [11]. An illustration of a simple PQC architecture in the search space is depicted in Figure 1. More specifically, we define four basic operation encodings  $\mathbf{x} = \{x_0, x_1, x_2, x_3\}$ , and the corresponding genes are represented as integers  $\{0, 1, 2, 3\}$ . Given a fixed  $n$ -qubit state, we define the following operations:

- $x_1$ : Variational PQC—A circuit with single-qubit rotations  $R_x, R_y, R_z$  (Equation (1)) performed on each qubit, with the rotation angles as trainable parameters. For generality, we consider qubit rotation in 3-dimensional space (i.e., applying  $R_x, R_y, R_z$ ) for all the qubits. This operator is used to change the qubit states based on the trainable parameters.
- $x_2$ : Entanglement—A circuit that performs circular entanglement to all the qubits by applying one or multiple controlled-Z gates (Equation (2)). In this work, we only consider circular entanglement, which has been widely used in prior studies.
- $x_3$ : Data-encoding PQC—A circuit with single-qubit rotations  $R_x(\theta)$  (Equation (1)) performed on each qubit, with the rotation angle  $\theta$  being the input data  $d$  scaled by trainable parameters  $\lambda$ ,

$$\theta = \sigma(\lambda d) \quad (8)$$

where  $\sigma(\cdot)$  is the activation function. Similar to  $x_1$ , this operator is also used to change the qubit states, but based on scaled/transformed input values, which is how the input data takes effect.

While prior work Ding and Spector [15] uses linear activation, we propose to use non-linear activations such as TANH to allow effective stacks of consecutive  $x_3$  circuits, since linear activation leads to multiple consecutive rotations being equivalent to a single rotation (Equation (11)).

- $x_0$ : Measurement—A Variational PQC ( $x_1$ ) with trainable parameters followed by measurement to obtain the qubit observables. The outcome is a binary value for each qubit with different probabilities. The outputs are computed by a linear weighting of the observables by another set of trainable parameters for each output, with optional activation functions, e.g., SOFTMAX for action probabilities. The architecture encoding/decoding is terminated when approaching  $x_0$ .

The search space of MEAS-PQC depends on the maximal length of the genomes. Since the encoding will terminate when approaching to  $x_0$ , all the genomes will have only one  $x_0$  at the end. So the search space is the sum of possible operations (except  $x_0$ ) for all the possible lengths less than the maximum length. Given a maximum length of the genomes  $n$ , the search space of our method is

$$\Omega_{x,n} = \sum_{i=1}^n (|\mathbf{x}| - 1)^{i-1} - K \quad (9)$$

where  $K$  is the number of duplicate genomes that can be decoded to equivalent architectures, which is described in detail in the following section.

### 3.3. Evolutionary Quantum Architecture Search

MEAS-PQC is proposed to generate diverse sequential combinations of quantum operators and iteratively search for good candidates with respect to the objectives. The search process is an evolutionary algorithm, and we elaborate on the following key components:

*Mutation.* To efficiently scan the search space for better genomes, we use Uniform Mutation by Addition and Deletion (UMAD [18]). UMAD is a mutation method that performs addition and deletion of genes separately. It has shown superior performance in many evolutionary computing applications such as program synthesis. For each mutation operation, UMAD deletes/adds genes to the genome controlled by addition/deletion rate  $r$ . Given genome of length  $l$ , there are  $l$  positions for deletion and  $l + 1$  positions for addition (including start and end). We perform size-neutral UMAD with

$$r_{del} = 1/l, r_{add} = 1/(l + 1) \quad (10)$$

so that the expected number of added/deleted genes is 1 after each mutation.

Comparing to traditional mutation operators, such as polynomial mutation that has been used in prior QAS work [15], UMAD has two major advantages. First, the traditional mutation operator replaces a randomly chosen gene with a new, randomly generated gene, and thus the length of genome does not change. In our case, we expect the algorithm to search for architectures with different lengths based on the parent, which can be achieved by UMAD. Secondly, since UMAD dissociates the additions from the deletions, it can thus provide more paths through the search space from a parent to a descendant, especially when specific genes are essential to the performance [18].

*Equivalence of Quantum Operators.* It is worth noting that some architectures in the search space are computationally equivalent, as some quantum operations can be linearly

combined. More specifically, given two rotation operators on the same axis  $R_x(\theta_1), R_x(\theta_2)$  (x-axis for instance), from Equation (1), we can derive that

$$R_x(\theta_1) \cdot R_x(\theta_2) = \begin{bmatrix} \cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} - \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2} & -i \cos \frac{\theta_1}{2} \sin \frac{\theta_2}{2} - i \cos \frac{\theta_2}{2} \sin \frac{\theta_1}{2} \\ -i \cos \frac{\theta_1}{2} \sin \frac{\theta_2}{2} - i \cos \frac{\theta_2}{2} \sin \frac{\theta_1}{2} & \cos \frac{\theta_1}{2} \cos \frac{\theta_2}{2} - \sin \frac{\theta_1}{2} \sin \frac{\theta_2}{2} \end{bmatrix}. \quad (11)$$

Let  $\theta_3 = \theta_1 + \theta_2$ , we have  $R_x(\theta_3) = R_x(\theta_1) \cdot R_x(\theta_2)$ . In other words, two consecutive rotations can be equivalently replaced by one rotation. As a result, if the genome has two or more consecutive  $x_1$ , we can reduce the genome by keeping only one  $x_1$ .

Similarly, given Equation (2), we have

$$CZ \cdot CZ = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad (12)$$

which means two consecutive  $x_2$  becomes actually identity, which can be safely removed from the genome. We perform duplicate elimination at each generation based on these equivalence of quantum operators to avoid the redundancy in the population.

### 3.4. Multi-objective Optimization for Quantum-Classical Systems

In this work, we consider three objectives for PQC-based learning systems: learning performance, quantum noise, and model size. First, the main objective is the learning performance. For each generation, we decode the population to different architectures, and use the architectures to construct PQC policies for the RL agents. The learning performance is computed as the average episode reward in the target RL environment to represent the area under the learning curve.

We also consider two other objectives that are essential to real-world applications of quantum systems: quantum noise and model size. Quantum noise usually refers to unexpected (and typically unwanted) random variation due to the discrete nature of photons [37]. The effect of quantum noise that occurs throughout a computation process can be quite complex, as there are many potential causes, e.g., thermal fluctuations, mechanical vibrations. While some recent work [38] reported the system-wide Pauli and measurement errors on specific quantum hardware regarding different numbers of qubits, the errors of different quantum gates and operations have not been found in past empirical studies. As a result, given no knowledge of the actual quantum hardware, we propose a rough measure of the theoretical quantum noise in PQC based systems.

Given a system with  $n$  qubits in the MEAS-PQC search space with operations  $x_i, i \in 0, 1, 2, 3$ , we define the system-level noise

$$\Delta = n \cdot (3 \sum_{i=0,1} \alpha_i \cdot \#x_i + 2\alpha_2 \cdot \#x_2 + \alpha_3 \cdot \#x_3). \quad (13)$$

In other words, we assume that the total quantum noise is proportion to the total number of quantum gates operating on any qubit. For this work, we take an artificial setting of  $\alpha_i = 1, i \in 1, 2, 3$ , meaning different gates are considered to have the same level of quantum noise. Notably, this setting can be further refined for specific quantum hardware.

For any parameterized learning models, model size, or number of parameters, is always an important factor that is closely related to computation cost, memory, storage, and other computation-related requirements. For our method, we also calculate the total number of parameters as

$$N = n \cdot (3 \cdot \sum_{i=0,1} \#x_i + \#x_3). \quad (14)$$

Note that since the search space increases exponentially to the maximum length of the architecture (Equation (9)), it is important to add penalty to the model size when searching for PQC architectures.

## 4. Experiments

In this section, we describe the implementation details and experimental results of MEAS-PQC on the benchmark RL environments regarding both single- and multi-objective optimization. We also compare our method to prior work to demonstrate the advantage of it against commonly-used alternating-layer PQC architectures (SOFTMAX-PQC by Jerbi et al. [11]) and classic genetic algorithms (EQAS-PQC by Ding and Spector [15]).

### 4.1. RL Environments

In this work, we consider three classical RL benchmark environments from the OpenAI Gym [19]: CartPole, MountainCar, and Acrobot, which have been widely used in RL research, including prior work on quantum RL [11,12,15]. The CartPole task is to prevent the pole from falling over by controlling the cart. For MountainCar, the goal is to drive up the mountain by driving back and forth to build up momentum. Acrobot refers to a swing-up task, in which the system must use the elbow (or waist) torque to move the system into a vertical configuration then balance. Detailed description can be found in Brockman et al. [19].

The specifications for RL environments are presented in Table 1, where the reward is the step reward and  $\gamma$  is the discount factor for future rewards. We follow prior work to use a single qubit to represent each state variable, while it is hard to interpret the meaning of qubit states, the D-PQC rotates the qubit based on the value of the state variable. In the scope of this work, we do not further optimize the number of qubits as a hyperparameter, but it can certainly be done with the proposed method, by including the size of qubits as a variable in the search space.

**Table 1.** RL environment specifications and hyperparameters. (\*: The reward function of MountainCar-v0 has been modified from the standard version in OpenAI Gym, following the practices in Jerbi et al. [11], Duan et al. [39]).

Environment	# States/Qubits	# Actions	Reward	$\gamma$	Horizon	Episodes
CartPole-v1	4	2	+1	1.0	500	600
MountainCar-v0	2	3	$-1 + height^*$	1.0	200	1200
Acrobot-v1	6	3	-1	1.0	500	1200
Environment	Learning Rates ( $\alpha_\theta, \alpha_w, \alpha_\lambda$ )		Observables	Value-Function Baselines		
CartPole-v1	0.01, 0.1, 0.1		$[Z_0 Z_1 Z_2 Z_3]$	None		
MountainCar-v0	0.01, 0.1, 0.01		$[Z_0, Z_0 Z_1, Z_1]$	Linear baseline [39]		
Acrobot-v1	0.01, 0.1, 0.01		$[Z_0, \dots, Z_5]$	Linear baseline [39]		

### 4.2. Implementation Details

For all the experiments, we perform quantum computation via noiseless simulation. The quantum circuits are implemented using Cirq [40] and TensorFlow Quantum [41]. The main search process is implemented using the *pymoo* [42] framework. Other specifications and hyper-parameters for architecture search and RL training are described as follows.

**Architecture Search.** MEAS-PQC uses NSGA-II with a population size of 30 and runs for 50 generations. The maximum length of architectures is set to 20, i.e., the largest architecture will have 20 consecutive operations varying from  $x_1$  to  $x_3$ , followed by measurement  $x_0$ . For each generation, we evaluate the learning performance of the population by running 5 trials in the RL environments with a reduced number of episodes to a factor of 0.5, which improves the efficiency of evolution. We also compute the estimation of quantum noise and computational cost as described in Section 3.4. For the final results, we evaluate the architectures produced by our method as well as prior work for 20 trials in order to reduce the variance caused by inevitable failure cases due to bad initialization of parameters and environment states.

**RL training during search.** We set the hyperparameters such as learning rates and observables following the general practice in Jerbi et al. [11], Ding and Spector [15], which

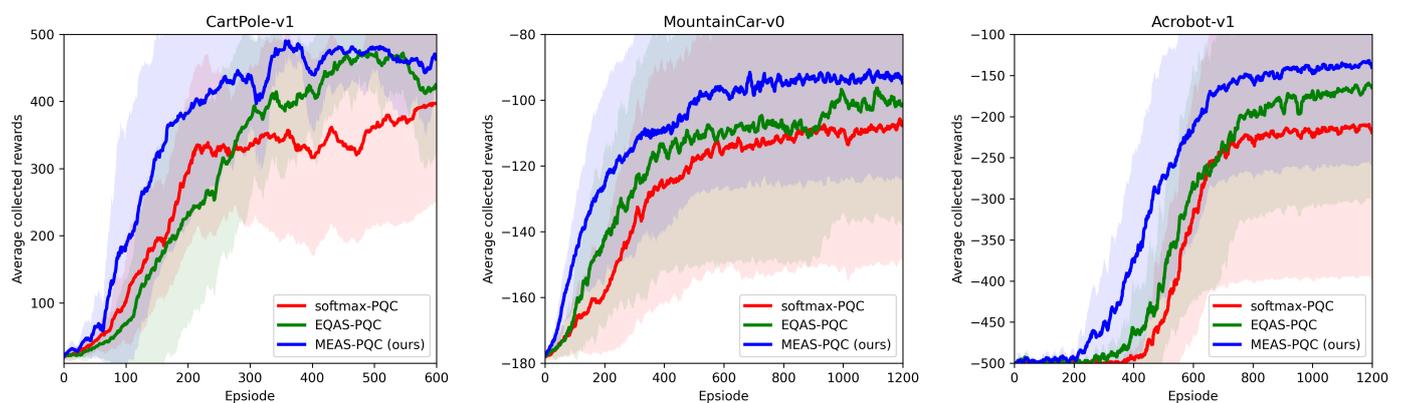
are also summarized in Table 1. All the agents are trained using REINFORCE [43], which is a basic Monte Carlo policy gradient algorithm. We apply the value-function baseline [44] in MountainCar to stabilize the Monte Carlo process, which has been commonly used in recent RL methods [39].

*Time Complexity Analysis.* Since the search runs for 50 generations with a population size of 30, and we reduced the number of episodes to a factor of 0.5, the theoretical time complexity upper bound of the search process is  $750 \times$  the runtime of training one single PQC model. In practice, we find our method takes only around  $200 \times$  the cost of training an alternating-layer PQC, due to the following factors: (1) a reduced number of episodes does not affect the evaluation for selection in GAs but can significantly reduce the training time, because later episodes are usually close to the optimal solution, meaning the RL agents will likely keep running for a longer period; (2) some of the architectures in the search space are either too small or not effective, taking much less time to train and evaluate.

We also use parallelization in the search process, which further reduces the clock time to around  $40 \times$  training a single alternating-layer PQC, i.e., around 2 days on a 64-core machine. More specifically, for each generation, all the candidate architectures are trained and evaluated simultaneously. All the experiments were run on a computing cluster with up to 64 cores per job. The total estimated compute time of all the experiments in this work is 30,000 CPU hours.

#### 4.3. Single-Objective Results

First, we evaluate the learning performance of the proposed MEAS-PQC by training with a single objective of average rewards. The purpose of this experiment is to show the effectiveness of variable-length genome representation in MEAS-PQC by using UMAD. We compute and visualize the average learning performance over 20 trials of the best-performing architectures searched by MEAS-PQC and compare to recent work (EQAS-PQC [15] and SOFTMAX-PQC [11]). To ensure a fair comparison, for SOFTMAX-PQC, we use the depth of 5, resulting in an architecture with length 16, which is greater than the resulting architectures searched by MEAS-PQC for all the environments. The learning performance is represented as the area under the learning curve, which is identical to the average collected reward. As shown in Figure 2, we can see that our method is able to find PQC architectures that outperform both EQAS-PQC and the standard alternating-layer PQC by a significant margin. While the final rewards may converge to optimum for all the methods (especially for easy RL environments like CartPole), the PQC architectures searched by our method is able to learn much quicker in early episodes.



**Figure 2.** Learning performance on benchmark RL environments. We plot the learning curves (smoothed by a temporal window of 10 episodes) averaged over 20 trials of the resulting MEAS-PQC architecture compared against EQAS-PQC [15] and SOFTMAX-PQC [11] on three benchmark RL environments. The shaded areas represent the standard deviation of the average collected reward.

#### 4.4. Multi-Objective Results

In Section 3.4, we propose three objectives: learning performance, quantum noise, and model size. Given the objective functions, we test the proposed method under different multi-objective settings. More specifically, 1-obj is optimizing using only the learning performance (average reward); 2-obj is optimizing both learning performance and quantum noise; 3-obj is optimizing all three objectives. The results are shown in Table 2.

**Table 2.** Multi-objective Evaluation. For each method, we evaluate the performance based on the three objectives: learning performance, quantum noise, and computation cost, as described in Section 3.4. The results show that, with multi-objective optimization, MEAS can produce architectures with less quantum noise and computation cost, while still maintaining superior learning performance.

Environment	Objectives	SOFTMAX-PQC [11]	EQAS-PQC [15]	MEAS-PQC (Ours)		
				1-obj	2-obj	3-obj
CartPole	Avg. Reward	271.0 ± 78.8	317.6 ± 67.7	349.8 ± 71.3	<b>351.6 ± 64.8</b>	342.3 ± 76.2
	Quantum noise	30	20	25	17	<b>15</b>
	Model size	20	<b>12</b>	19	13	13
MountainCar	Avg. Reward	−126.6 ± 33.4	−119.5 ± 32.8	<b>−108.2 ± 25.8</b>	−110.7 ± 21.4	−111.3 ± 25.2
	Quantum noise	30	21	20	<b>14</b>	17
	Model size	20	13	14	<b>8</b>	11
Acrobot	Avg. Reward	−353.0 ± 93.7	−328.5 ± 74.6	<b>−280.7 ± 79.3</b>	−283.4 ± 78.6	−289.3 ± 75.5
	Quantum noise	30	17	22	19	<b>15</b>
	Model size	20	13	16	<b>11</b>	<b>11</b>

We can observe that first, the proposed MEAS-PQC (1-obj) outperforms other methods by a significant margin with architectures that have less or comparable quantum noise and model size. For different multi-objective settings, we can observe that both MEAS-PQC (2-obj) and MEAS-PQC (3-obj) have reduced quantum noise and model size comparing to MEAS-PQC (1-obj), while maintaining similar performance. By using multi-objective optimization, the proposed method is able to search for architectures not only have good learning performance, but also optimized for real-world computation needs such as quantum-noise tolerance and model size.

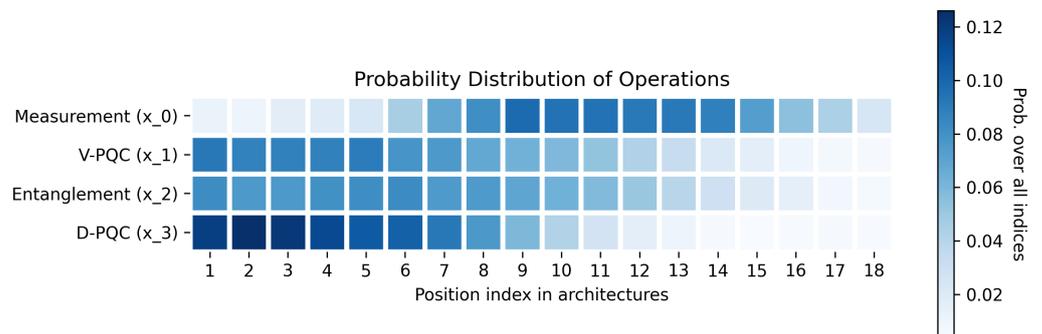
#### 4.5. Pattern Analysis in Quantum Architectures

To further interpret the results and discover the performance-critical design choices of PQCs as opposed to the commonly-used alternating-layer architecture, we perform qualitative analysis on the top-performing architectures searched by MEAS-PQC. We perform pattern mining on the architectures and calculate the frequency of patterns with different lengths of 2, 3, 4, and 5. The results are shown in Table 3, and we can observe that consecutive  $x_3$  and alternating  $x_1, x_2$  are the most common patterns, which indicates that there may exist essential substructures of quantum circuits that lead to good learning performance.

In addition, we also calculate and visualize the probability distribution of each quantum operation over the positions in the architecture, as shown in Figure 3. We can observe that the V-PQC has a similar frequency as entanglement, which aligns with the alternating-layer design. However, the frequency of D-PQC has an obvious decreasing trend, indicating that it is better to have more Data-encoding PQCs at the beginning of the architecture. This finding further indicates that more classical computation is likely to be needed in the shallow part of the hybrid quantum learning models.

**Table 3.** Pattern frequency in top-performing PQC architectures. We select 30 top-performing architectures searched by MEAS-PQC (10 for each RL environment), and calculate the frequency of patterns with different lengths. We can observe that consecutive  $x_3$  and alternating  $x_1, x_2$  are the most common patterns.

len-2		len-3		len-4		len-5	
Pattern	Freq.	Pattern	Freq.	Pattern	Freq.	Pattern	Freq.
(3, 3)	0.282	(3, 3, 3)	0.213	(3, 3, 3, 3)	0.122	(1, 2, 1, 2, 1)	0.087
(2, 1)	0.245	(1, 2, 1)	0.184	(2, 1, 2, 1)	0.122	(1, 3, 3, 3, 3)	0.075
(1, 2)	0.205	(2, 1, 2)	0.142	(1, 2, 1, 2)	0.102	(2, 1, 2, 1, 2)	0.069
(1, 3)	0.099	(2, 1, 3)	0.067	(1, 3, 3, 3)	0.068	(3, 3, 3, 3, 2)	0.064
(3, 2)	0.066	(3, 2, 1)	0.059	(3, 3, 3, 1)	0.063	(3, 3, 3, 3, 1)	0.064



**Figure 3.** Probability distribution of quantum operations over positions in top-performing PQC architectures. We select 30 top-performing architectures searched by MEAS-PQC (10 for each RL environment), and visualize the probability distributions of each operation over its positions.

## 5. Discussion of Limitations and Broader Impacts

There are several limitations to our work. First, due to framework constraints, all the experiments are conducted using a simulation backend, and the quantum noise and computation cost are evaluated based on the proposed theoretical metrics. Second, for generality, this work considers genome abstractions for more structural quantum operations, e.g., qubit rotation in 3-dimensional space and circular entanglement. It is easy to extend our work to have a much larger search space by using single-dimensional rotations and partial entanglement, and we look forward to exploring these extensions in future work. The primary goal of this work is to show that the proposed method can automate and improve the macro architecture design of PQC-based hybrid quantum-classical learning systems, with a fair comparison to prior work.

In general, we expect our work to have positive societal impacts. While prototyped in a simulated environment, the proposed method could be beneficial to near-term quantum computing applications, especially quantum RL. Since modern RL tasks usually require huge amounts of computation, our method may provide significant computational advantages regarding both time and energy consumption. Our work further considers constraints such as quantum noise and model size, which could make the proposed method more pragmatic for real-world applications.

On the other hand, like other recent work in RL and machine learning research in general, our work could have negative societal impacts. Concerns such as whether RL applications will have positive impact on the society are naturally inherited by our work. More specifically for quantum RL, while we made an effort to explain the results, the overall interpretability of quantum systems is still not as good as that of many classical systems, which may lead to negative consequences. Moreover, the requirement for quantum computing hardware to obtain computational benefits may increase inequality because of unequal access to quantum computing resources.

## 6. Conclusions

In this work, we propose MEAS-PQC, a multi-objective evolutionary quantum architecture search framework for hybrid quantum learning systems. MEAS-PQC uses a population-based genetic algorithm with UMAD and quantum-specific configurations to evolve PQC architectures by exploring the search space of quantum operations. Experimental results show that our method can significantly improve the learning performance of PQC-based systems in solving benchmark reinforcement learning problems. The results on multi-objective optimization further show that our method can search for architectures that have good learning performance and are also optimized for reduction in quantum noise and model size. We also perform analysis to extract and interpret the performance-critical architecture design choices. In future work, we expect to apply our method on real quantum hardware and use hardware-specific metrics for objectives to show the performance in a more pragmatic setting.

**Author Contributions:** Conceptualization, L.D. and L.S.; Methodology, L.D.; Software, L.D.; Validation, L.D.; Formal analysis, L.D.; Resources, L.S.; Writing—original draft, L.D.; Writing—review & editing, L.S.; Visualization, L.D.; Supervision, L.S.; Funding acquisition, L.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Science Foundation under Grant No.1617087. Any opinions, findings, and conclusions expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work was performed in part using high performance computing equipment obtained under a grant from the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are openly available via OpenAI Gym [19]: <https://www.gymnasium.dev/>.

**Acknowledgments:** The authors would like to thank Sofiene Jerbi for providing implementation details for reproducing baseline results in the prior work [11], as well as Donald Towsley, Edward Pantridge, Thomas Helmuth, Ryan Boldi, and Anil Saini for their comments and suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
2. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [CrossRef]
3. Schuld, M.; Bocharov, A.; Svore, K.M.; Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **2020**, *101*, 032308. [CrossRef]
4. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [CrossRef] [PubMed]
5. Schuld, M.; Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef] [PubMed]
6. Li, G.; Song, Z.; Wang, X. VSQL: Variational shadow quantum learning for classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 8357–8365.
7. Liu, J.G.; Wang, L. Differentiable learning of quantum circuit born machines. *Phys. Rev. A* **2018**, *98*, 062324. [CrossRef]
8. Zhu, D.; Linke, N.M.; Benedetti, M.; Landsman, K.A.; Nguyen, N.H.; Alderete, C.H.; Perdomo-Ortiz, A.; Korda, N.; Garfoot, A.; Brecque, C.; et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **2019**, *5*, eaaw9918. [CrossRef]
9. LaRose, R.; Coyle, B. Robust data encodings for quantum classifiers. *Phys. Rev. A* **2020**, *102*, 032420. [CrossRef]
10. Huang, H.L.; Du, Y.; Gong, M.; Zhao, Y.; Wu, Y.; Wang, C.; Li, S.; Liang, F.; Lin, J.; Xu, Y.; et al. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.* **2021**, *16*, 024051. [CrossRef]
11. Jerbi, S.; Gyurik, C.; Marshall, S.; Briegel, H.; Dunjko, V. Parametrized Quantum Policies for Reinforcement Learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 28362–28375.
12. Skolik, A.; Jerbi, S.; Dunjko, V. Quantum agents in the gym: A variational quantum algorithm for deep q-learning. *arXiv* **2021**, arXiv:2103.15084.
13. Pérez-Salinas, A.; Cervera-Liarta, A.; Gil-Fuster, E.; Latorre, J.I. Data re-uploading for a universal quantum classifier. *Quantum* **2020**, *4*, 226. [CrossRef]
14. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *J. Mach. Learn. Res.* **2019**, *20*, 1997–2017.

15. Ding, L.; Spector, L. Evolutionary quantum architecture search for parametrized quantum circuits. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Boston, MA, USA, 9–13 July 2022; pp. 2190–2195.
16. Lu, Z.; Whalen, I.; Boddeti, V.; Dhebar, Y.; Deb, K.; Goodman, E.; Banzhaf, W. Nsga-net: Neural architecture search using multi-objective genetic algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 419–427.
17. Deb, K. Multi-objective optimisation using evolutionary algorithms: An introduction. In *Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing*; Springer: London, UK, 2011; pp. 3–34.
18. Helmuth, T.; McPhee, N.F.; Spector, L. Program synthesis using uniform mutation by addition and deletion. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 1127–1134.
19. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.
20. Nielsen, M.A.; Chuang, I. Quantum computation and quantum information. *Am. J. Phys.* **2002**, *70*, 558. [[CrossRef](#)]
21. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242–246. [[CrossRef](#)]
22. Schuld, M.; Sweke, R.; Meyer, J.J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **2021**, *103*, 032430. [[CrossRef](#)]
23. Spector, L. *Automatic Quantum Computer Programming: A Genetic Programming Approach*; Springer New York, NY, USA, 2004; Volume 7.
24. Grimley, H.R.; Economou, S.E.; Barnes, E.; Mayhall, N.J. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat. Commun.* **2019**, *10*, 1–9. [[CrossRef](#)]
25. Tang, H.L.; Shkolnikov, V.; Barron, G.S.; Grimley, H.R.; Mayhall, N.J.; Barnes, E.; Economou, S.E. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum* **2021**, *2*, 020310. [[CrossRef](#)]
26. Las Heras, U.; Alvarez-Rodriguez, U.; Solano, E.; Sanz, M. Genetic algorithms for digital quantum simulations. *Phys. Rev. Lett.* **2016**, *116*, 230504. [[CrossRef](#)]
27. Du, Y.; Huang, T.; You, S.; Hsieh, M.H.; Tao, D. Quantum circuit architecture search: Error mitigation and trainability enhancement for variational quantum solvers. *arXiv* **2020**, arXiv:2010.10217.
28. Lu, Z.; Shen, P.X.; Deng, D.L. Markovian quantum neuroevolution for machine learning. *Phys. Rev. Appl.* **2021**, *16*, 044039. [[CrossRef](#)]
29. Krastanov, S.; Albert, V.V.; Jiang, L. Optimized entanglement purification. *Quantum* **2019**, *3*, 123. [[CrossRef](#)]
30. Niu, M.Y.; Boixo, S.; Smelyanskiy, V.N.; Neven, H. Universal quantum control through deep reinforcement learning. *NPJ Quantum Inf.* **2019**, *5*, 1–8. [[CrossRef](#)]
31. Kuo, E.J.; Fang, Y.L.L.; Chen, S.Y.C. Quantum architecture search via deep reinforcement learning. *arXiv* **2021**, arXiv:2104.07715.
32. Zhang, S.X.; Hsieh, C.Y.; Zhang, S.; Yao, H. Differentiable quantum architecture search. *arXiv* **2020**, arXiv:2010.08561.
33. Ostaszewski, M.; Grant, E.; Benedetti, M. Structure optimization for parameterized quantum circuits. *Quantum* **2021**, *5*, 391. [[CrossRef](#)]
34. Liu, Y.; Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G.; Tan, K.C. A survey on evolutionary neural architecture search. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [[CrossRef](#)]
35. Ding, L.; Spector, L. Optimizing Neural Networks with Gradient Lexicase Selection. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
36. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving from Nature*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 849–858.
37. Clerk, A.A.; Devoret, M.H.; Girvin, S.M.; Marquardt, F.; Schoelkopf, R.J. Introduction to quantum noise, measurement, and amplification. *Rev. Mod. Phys.* **2010**, *82*, 1155. [[CrossRef](#)]
38. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [[CrossRef](#)]
39. Duan, Y.; Chen, X.; Houthoofd, R.; Schulman, J.; Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1329–1338.
40. Hancock, A.; Garcia, A.; Shedenhelm, J.; Cowen, J.; Carey, C. Cirq: A Python Framework for Creating, Editing, and Invoking Quantum Circuits. 2019. Available online: <https://github.com/quantumlib/Cirq> (accessed on 15 November 2022).
41. Broughton, M.; Verdon, G.; McCourt, T.; Martinez, A.J.; Yoo, J.H.; Isakov, S.V.; Massey, P.; Halavati, R.; Niu, M.Y.; Zlokapa, A.; et al. Tensorflow quantum: A software framework for quantum machine learning. *arXiv* **2020**, arXiv:2003.02989.
42. Blank, J.; Deb, K. Pymoo: Multi-objective optimization in python. *IEEE Access* **2020**, *8*, 89497–89509. [[CrossRef](#)]
43. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [[CrossRef](#)]
44. Greensmith, E.; Bartlett, P.L.; Baxter, J. Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning. *J. Mach. Learn. Res.* **2004**, *5*, 1471–1530.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.