Evolutionary Design of Parameterized Quantum Circuits for Classification and Reinforcement Learning Tasks

Devroop Kar[0000−0003−1404−1076], Daniel Krutz[0000−0002−0310−3086], and Travis Desell[0000−0002−4082−0439]

Rochester Institute of Technology, Rochester NY 14623, USA
{dk7405,dxkvse,tjdvse}@rit.edu

Abstract. This paper extends Evolutionary eXploration of Augmenting Quantum Circuits (EXAQC), a unified framework for the automated design and optimization of parameterized quantum circuits (PQCs) to include reinforcement learning tasks. Designing effective quantum circuits is challenging as the performance of a circuit is highly dependent on architecture, including gate composition, qubit interactions, and depth, as well as the target quantum system. Traditional approaches relying on fixed templates or heuristic designs often fail to generalize across tasks. To address this, EXAQC formulates circuit construction as a distributed evolutionary optimization problem, jointly exploring circuit structure and parameters in parallel worker processes using mutation and crossover operators. Experiments on 4 standard classification benchmarks demonstrate that EXAQC achieves competitive or superior accuracy with more efficient circuits. Additionally, evaluations on 4 different reinforcement learning environments show that evolved circuits can learn effective policies and reach near-optimal rewards with relatively low parameter counts. Overall, EXAQC provides a scalable, task-adaptive approach for quantum circuit discovery, highlighting the potential of evolutionary methods in advancing quantum machine learning.

Keywords: Neuroevolution · Genetic Algorithms · Quantum Computing · Quantum Circuits · Parallelization

1 Introduction

Quantum machine learning (QML) leverages principles such as superposition, entanglement, and interference to enhance learning and decision-making processes [8, 16, 23, 53]. It has shown promise in improving efficiency across optimization [17, 21, 44], pattern recognition [62], and data analysis [61]. Variational quantum circuits (VQCs) underpin many QML approaches [15, 21, 35, 52], but their effectiveness depends heavily on architectural choices such as gate composition, qubit connectivity, and measurement design [54, 58]. Existing methods rely largely on hand-crafted or heuristic circuit templates, which often fail to generalize across tasks. Circuit topology significantly impacts expressivity, trainability, and robustness, often more than optimization or loss design [4, 12, 18]. Poor architectures can lead to barren plateaus, weak gradients, and insufficient entanglement [3,39,46]. This motivates automated circuit discovery methods that jointly optimize structure and parameters rather than assuming fixed designs. Evolutionary computation has shown strong performance in neural architecture search [33], symbolic regression [28], and reinforcement learning [29]. Its ability to explore large, discrete design spaces makes it well-suited for quantum circuit optimization [5,9,24,35,40,49,57,66], yet existing approaches often restrict search spaces or have high computational cost, limiting real-world applicability.

Recent work in quantum circuit design tackles quantum architecture search (QAS) as an optimization problem over large combinatorial spaces, often using evolutionary methods. Prior studies have explored circuit synthesis balancing fidelity and depth [56], as well as evolutionary approaches for classification using measurement-based objectives [65]. Methods such as QNEAT [14] and multi-objective formulations [10] jointly optimize circuit structure and parameters, while supervised approaches including EQNAS [30], adaptive evolutionary search [31], and continuous optimization methods [37] demonstrate effectiveness on standard datasets. Other works focus on representation design [13, 34] and hybrid quantum-classical models [32,48]. In reinforcement learning, prior efforts mainly optimize parameters of fixed circuit ansätze [1, 7, 19, 26]. Overall, these methods show strong results but typically rely on fixed circuit templates or task-specific objectives. This motivates the need for a unified framework that can adapt circuit structure, training objectives, and execution backends. Additional details on related works are provided in the Appendix (Section 6) in the supplementary material[1].

Building on Kar *et al.* [20], this work provides an extensive analysis of EXAQC, our evolutionary framework for task-driven parameterized quantum circuit (PQC) discovery, providing additional results on classification tasks, as well as extending it to reinforcement learning tasks. We include new evaluations on Gymnasium reinforcement learning environments [60]. Results investigate the utility of the various mutation and crossover operations, showing that these operations and in particular a new n-ary crossover method are effective. Search progress in terms of both loss/fitness and circuit size is analyzed, showing that EXAQC is able to quickly achieve high performing PQCs with more compact PQCs as compared other state-of-the-art works.

2 Methodology

This paper provides a full presentation of Evolutionary Exploration of Augmenting Quantum Circuits (EXAQC)[2] which was first briefly introduced with preliminary results in [20]. EXAQC takes inspiration from the EXAMM neuroevolu-

[1] The Appendix and supplementary material are available at https://doi.org/10.5281/zenodo.19645683

[2] EXAQC source code is open source and publicly available at: https://github.com/travisdesell/exaqc

tion algorithm [45] and its genetic programming variant EXA-GP [41, 42], with new mutation operators designed to handle the restrictions of quantum computing gates (i.e., fixed numbers of input and output qubits per gate). EXAQC adapts EXAMM and EXA-GP's binary crossover, extending it into a novel *n-ary* crossover operator. It also provides a new exponential crossover operator. It leverages EXAMM/EXA-GP's Lamarckian weight inheritance [36] to reduce the amount of backpropagation and reinforcement learning episodes required for the evolved circuits. It utilizes a distributed, asynchronous master-worker framework with a steady state population for easy scalability and automated load balancing (described in detail in Appendix 6.2). EXAQC provides a wide range of potential gates that span nearly the entire range of available gates in Qiskit or Pennylane, which are provided in Appendixes 6.4 and 6.5 for Qiskit and Pennylane, respectively.

2.1 Genome Representation

Genomes are represented by a list of input qubit identifiers, a list of output qubit identifiers (which may or may not overlap with the input qubit identifiers) and a list of gates. Each gate has a depth (a float between 0.0 and 1.0), list of input and output qubit identifiers, dictionary of trainable parameters and an innovation number (similar to NEAT [55], EXAMM and EXA-GP) which is is utilized the identify the same gate across multiple genomes for purposes of crossover. Unlike other strategies which utilize a fixed genome size, the list of gates is unbounded allowing for circuits to grow as large as needed. Genomes can be exported to and imported from a given JSON format for easy portability.

2.2 Mutation and Crossover Operators

Due to space constraints, we refer the reader to Kar et al. [20] or Appendix 6.3 in the supplementary material for a detailed description and visual representations of each mutation and crossover operator. In short, *add gate* selects a random depth in the circuit and adds a random gate whose input qubits are effected by the circuits input qubits and output qubits have a computational path to the circuits readout qubits. *Reorder gate* selects a gate at random in the circuit, disables it and creates a copy of it at a new random depth; *enable and disable gate* randomly selects a gate and enables or disables it; and *swap gate* selects a gate at random, disables it and creates a new gate at a random depth between the nearest gate before and after it in the network, and then randomly changes one of its input or output qubits to a different qubit. *Exponential crossover* selects a pivot point at a random depth in both parent networks, and then utilizes all gates up to the first parents pivot point in the child network, as well as all gates after the second parents pivot point. *N-ary crossover* ($N = 5$ for this work) creates a child with all gates that appear in the primary (most fit) parent and any other parent. Gates which only appear in the most fit parent are added at the *best_keep_rate* (0.75 for this work), and if a gate exists in any other gate, it is added to the child genome at the *other_keep_rate* (0.25 for this work).

For parameterized gates in multiple parents, a randomized simplex approach is used [22, 59], where a randomized line search is performed between the weight in the best parent and the average of the weights in the other parents.

Note that it is possible for the *disable gate*, *reorder gate* and *swap qubits* mutations to result in circuits where the inputs do not affect the outputs. After a genome is created by mutation (or crossover), a forward traversal is performed over its gates in sorted order to make sure that at least one input flows to at least one output. If this is not the case, the genome is discarded as invalid and EXAQC will attempt to generate another one via mutation or crossover.

2.3 Loss Functions for Evolutionary Quantum Circuit Optimization

Evaluating candidate quantum circuits during evolutionary search requires objective functions that are informative, differentiable under simulation, and aligned with the intended task. EXAQC implements a number of loss functions for classification (fidelity loss, cross entropy, focal loss, balanced focal loss) and reinforcement learning (REINFORCE [63], Proximal Policy Optimization [51] and Actor Critic [27]). For this work we utilize cross entropy and balanced focal loss for classification tasks, and REINFORCE for reinforcement learning tasks.

Balanced Focal Loss on Quantum Readout Probabilities The circuit forward pass produces either a quantum state vector ψ or measurement probabilities, which are normalized and, if needed, converted via $|\psi|^2$ or marginalized over selected readout wires to obtain a valid probability distribution. This processed output is then used in the loss computation, either directly comparing states (e.g., fidelity loss) or comparing probabilities against target labels (e.g., cross-entropy).To address class imbalance and improve focus on hard-to-classify samples, balanced focal loss modulates standard cross-entropy by down-weighting well-classified examples. Let $p \in \mathbb{R}^K$ denote the marginal distribution mapped to K classes, y the one-hot encoded label, α_k the class-balancing weight for class k, and $\gamma \geq 0$ the focusing exponent. The balanced focal loss is:

$$\mathcal{L}_{\text{BFL}} = -\sum_{k=1}^{K} \alpha_k \, y_k \, (1 - p_k)^{\gamma} \log(p_k). \tag{1}$$

Policy Gradient (REINFORCE) For RL, the EXAQC framework adopts a hybrid quantum–classical architecture in which the quantum circuit acts as a feature extractor and directly parameterizes the policy. Specifically, environment observations are encoded into a quantum state (via angle or basis encoding), and the circuit outputs a vector of expectation values (or probabilities). This output forms a fixed-length feature representation that is interpreted as policy parameters without requiring an additional learnable linear layer in the REINFORCE setting. Formally, let the circuit output be denoted as $z(s; \theta) \in \mathbb{R}^K$, obtained from measuring expectation values of the parameterized quantum circuit. The features are directly mapped to logits via

$$\ell(s;\theta) = \log\left(p(s;\theta)\right), \tag{2}$$

where $p(s;\theta)$ denotes normalized output probabilities. The stochastic policy is then defined as

$$\pi_\theta(a \mid s) = \frac{\exp(\ell_a(s;\theta))}{\sum_{a'} \exp(\ell_{a'}(s;\theta))}. \tag{3}$$

A quantum circuit parameterized by θ is thus interpreted as a policy $\pi_\theta(a \mid s)$, mapping environment states s to action distributions a. The objective is to maximize the expected cumulative return:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right], \tag{4}$$

where r_t denotes the reward at time t and $\gamma \in [0,1]$ is the discount factor.

For stochastic policies, we employ the REINFORCE objective:

$$\mathcal{L}_{\mathrm{PG}} = -\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \log \pi_\theta(a_t \mid s_t)\, G_t\right], \tag{5}$$

where $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ is the discounted return. In this formulation, $\log \pi_\theta(a_t \mid s_t)$ is computed directly from the circuit outputs $z(s_t;\theta)$, meaning gradients propagate from the policy loss through the log-probabilities into the quantum circuit parameters. This enables the circuit to directly learn representations that shape the action distribution to maximize long-term rewards.

3 Benchmark Tasks

We evaluate EXAQC using both classical supervised datasets and reinforcement learning Gym environments as an extension of our previous work [20]. Classification benchmarks were *Iris* (4 real valued features, 3 classes), *Wine* (13 real valued features, 3 classes), *Seeds* (7 geometric features, 3 classes) and *Breast Cancer* (30 features, 2 classes) from the UCI machine learning repository. RL tasks included *CartPole-v1* (a class control task with discrete actions), *FrozenLake-v1* (grid based navigation with discrete states and actions), *MountainCarContinuous-v0* (continuous control and continuous actions), and *Walker2d-v5* (high dimensional control with continuous states and actions).

These benchmarks were selected as they are well studied for comparison to SOTA and that they capture fundamentally different learning paradigms. Supervised datasets enable controlled analysis of circuit expressivity, parameter efficiency, and generalization under well-defined input-output mappings; while RL environments introduce sequential decision-making, delayed rewards, and exploration challenges, which are critical for assessing the adaptability and policy-learning capabilities of evolved quantum circuits. Together, these benchmarks provide a holistic evaluation framework, highlighting the capabilities of EXAQC to generalize across static prediction tasks and dynamic control settings.

4 Results and Discussion

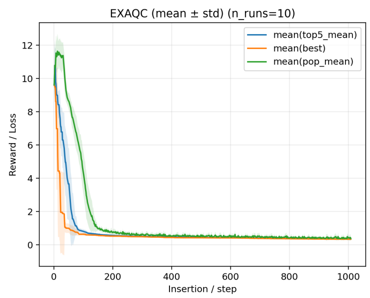
The EXAQC experiments for classification utilized a single steady state population with a maximum size of 30 genomes. Pennylane was used as the target quantum framework. Individuals were only generated by mutating the base empty genome (no gates), 1-3 times (selected uniformly at random) until the population was full. The potential of multiple mutations provides a more diverse initial population and the ability to break out of local minima. After the population was initalized, 1-3 mutations were performed on the child (again selected uniformly at random) 70% of the time. Crossover was performed otherwise, with *binary crossover* at 10%, *n-ary crossover* (with $n = 5$) at 10%, and *exponential crossover* at 10%. When mutation was applied, *add gate* was selected at 55%, *reorder gate* at 10%, *qubit swap* at 10%, *enable gate* at 5%, and *disable gate* at 10% and *clone* at 10%. Classification experiments were run until a total of 1000 genomes were evaluated. For training parameterized gates, each generated circuit was trained for 10 epochs using the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0000 due to PQC weights being angles[3]. All runs were performed utilizing 12 MPI processes (1 main process and 11 workers). The RL experiments were run for 1000 genomes in case of *CartPole*, *FrozenLake* and *MountainCar-continuous*. 2000 genomes were used for *Walker2d* to identify convergence and highest reward. Each circuit was trained for 1000 episodes and evaluated on 100 episodes. Search hyperparameters were the same as classification, except using a maximum population size of 50 genomes. Instructions for reproducing all experiments in this work are provided in the README of the open source EXAQC repository[4].

Dataset	**Method**	**Best Accuracy**	**Avg Accuracy**	**Std. Dev. Accuracy**	**Best N Parameters**	**Best N Gates**
Iris	Nguyen *et al.* [43]	-	95.33	± 0.0125	23	8
	Benitez *et al.* [2]	-	91.33	± 0.0393	-	23
	Maldonado *et al.* [38]	-	94.0	± 1.56	-	-
	Woun *et al.* [64]	100.0	-	± 1.6	-	-
	EXAQC (Ours)	100.0	96.33	± 3.79	5	7
Seeds	EXAQC (Ours)	100.0	95.71	± 2.56	9	9
Wine	Nguyen *et al.* [43]	-	98.34	± 0.0021	> 600	$10(2 \times 5)$
	Woun *et al.* [64]	96.0	-	± 1.3	-	-
	EXAQC (Ours)	97.22	91.94	± 3.15	16	25
Breast Cancer	Nguyen *et al.* [43]	-	98.34	± 0.0021	> 2500	$14(2 \times 7)$
	Woun *et al.* [64]	93.1	-	± 1.3	-	-
	EXAQC (Ours)	95.61	94.26	± 0.55	20	27

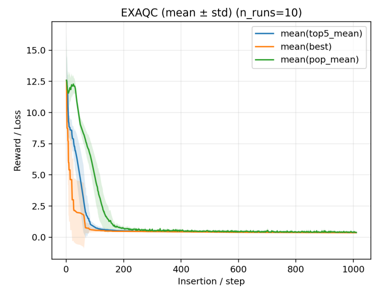
Table 1: Comparison of EXAQC to fixed and evolved state-of-the-art methods.

[3] Utilizing weight decay value would bias weights to 0 which aids standard neural networks, however this bias is detrimental to PQC angular weights.

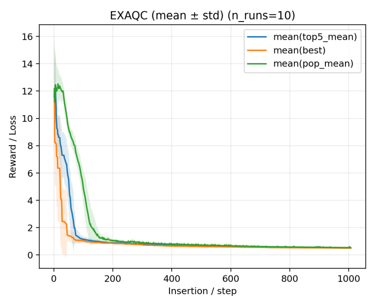
[4] https://github.com/travisdesell/exaqc



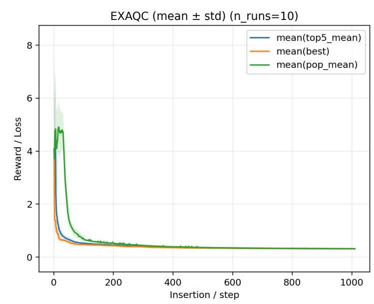
(a) Iris



(b) Seeds



(c) Wine

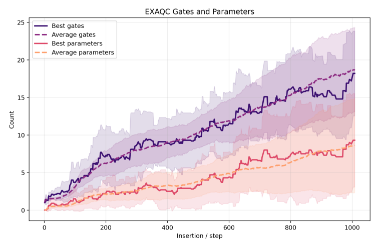


(d) Breast Cancer

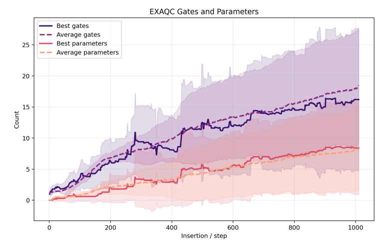
Fig. 1: Combined loss plots across repeated experiments. Plots show the best genome loss, loss of the top-5 genomes and the population average loss as a whole.

4.1 Classification Benchmarks

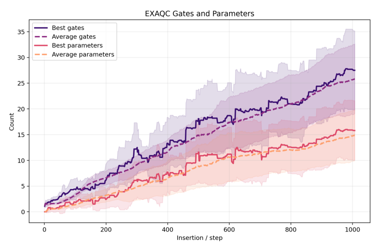
The evolved quantum circuits were trained on the classification datasets using balanced cross-entropy loss due to imabalanced datasets, utilizing the readout probabilities for output quantum states against each of the class labels. We used N_q qubits as input, same as the number of input features and $\lceil \log_2 n \rceil$ as the number of outputs where n is the number of classes. Results across the benchmark datasets are provided in Table 1 and compared to other state of the art methods in terms of best and average performance across experiments, as well as PQC sizes (when available). These works were selected based on their quantum inspired approaches involving traditional models like SVM as well as equivalent evolutionary approaches as well. To our knowledge these works highlighted the best results in this specific domain. Even with a very modest budget of 1000 evaluated genomes, we find that EXAQC can quickly find well performing quantum circuits for classification tasks. Appendix 6.6 provides visualizations of the best found circuits for each classification benchmark.



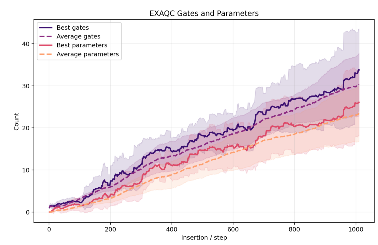
(a) Iris



(b) Seeds



(c) Wine



(d) Breast Cancer

Fig. 2: Gate and parameter complexity plots across different classification datasets. Plots out the gates and parameters for the best identified genome and the average metrics for the population.

From Table 1, EXAQC achieves consistently strong performance across all benchmarks, attaining perfect or near-perfect best accuracy on Iris and Seeds, and competitive results on Wine and Breast Cancer. These results are comparable to or exceed prior approaches such as QSVM [64] and QES-TPE [43], demonstrating the effectiveness of evolutionary search in discovering highly expressive quantum circuit architectures. Notably, EXAQC achieves this performance with significantly fewer parameters. Compared to other evolutionary approaches such as EQAS [66], EXAQC achieves a favorable balance between circuit complexity and performance, evolving highly efficient circuits that generalize well across tasks. This highlights the ability of EXAQC to identify compact yet expressive circuits, aligning with prior findings on the importance of circuit structure in quantum model expressibility [50, 54].

Figures 1 and 2 demonstrates the collective loss and circuit size plots across 10 independent runs for each of the classification benchmark problems, respectively. Plots show that EXAQC swiftly finds well performing circuits, and then refines them for higher performance, which requires steady growth as both gate and parameter counts grow steadily over the course of evolution. This indicates that improved predictive performance is generally associated with progressively richer circuit structures. The best genomes consistently track above the population averages, showing that the most successful solutions tend to exploit greater structural and parametric capacity. The growth is also task-dependent: simpler

Dataset	Method	Best Reward	Avg Reward	Std. Dev. Reward	Best N Parameters	Best N Gates
CartPole	Ding *et al.* [10]	-	349.8	±71.3	19	< 25
	Chen *et al.* [6]	500	-	-	26	-
	Kolle *et al.* [25]	498.0	-	±4.47	24	12
	Giovagnoli *et al.* [14]	500	-	-	-	-
	EXAQC (Ours)	500	484.58	±15.13	1	2
FrozenLake	Kolle *et al.* [26]	1.0	-	-	5313	-
	Giovagnoli *et al.* [14]	1.0	-	-	-	-
	Druagan *et al.* [11]	0.85	-	±0.16	81	-
	EXAQC (Ours)	1.0	0.7	±0.12	3	8
Mountaincar-C.	EXAQC (Ours)	99.351	94.517	±13.51	2	9
Walker-2d	EXAQC (Ours)	1010.20	871.57	±191.403	3	9

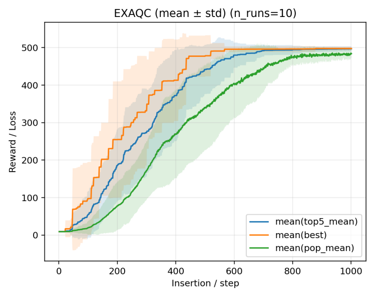
Table 2: Comparison with other state-of-the-art methods involving PQCs as well as evolutionary approaches for RL benchmarks. Please note that Giovagnoli *et al.* [14] does not explicitly report the reward and standard deviation metrics and Ding *et al.* [10] offers a proportional metric for number of gates.

datasets such as Iris and Seeds require more moderate increases in complexity, whereas higher-dimensional problems such as Wine and especially Breast Cancer drive substantially larger gate and parameter counts. This suggests that EXAQC adapts circuit size to dataset difficulty, allocating deeper and more parameterized quantum models when the classification boundary is more complex.

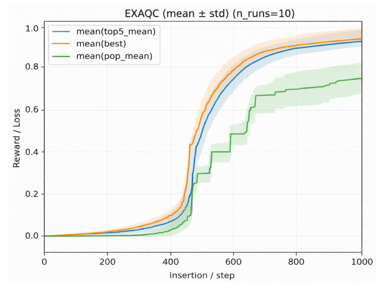
While EXAQC demonstrates strong best-case performance, the average accuracy and variance indicate some sensitivity to stochastic evolutionary dynamics, particularly on simpler datasets such as Iris and Wine. However, on more complex datasets such as Breast Cancer, the variance is substantially lower, suggesting more stable convergence when richer representations are required. Overall, these results suggest that EXAQC provides an effective and scalable framework for quantum circuit design, capable of producing high-performing solutions with reduced parameter overhead.

4.2 Reinforcement Learning Benchmarks

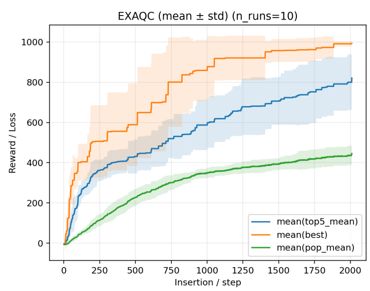
The evolved quantum circuits were trained on reinforcement learning environments using the REINFORCE policy gradient method, where the expectation values of the output qubits were used to parameterize action distributions. For each environment, the number of input qubits corresponded to the dimensionality of the state space, while the number of output qubits was determined by the action space. Performance was evaluated using the average episodic return. Results across the benchmark environments are summarized in Table 2, and Figures 3 and 4 show reward and circuit size progress over the course of the experiments. Appendix 6.7 provides visualizations of the best found circuits for each RL benchmark.



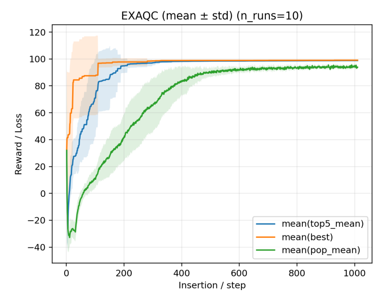
(a) CartPole



(b) FrozenLake



(c) Walker2d

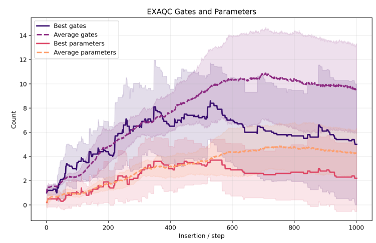


(d) Mountaincar-Continuous

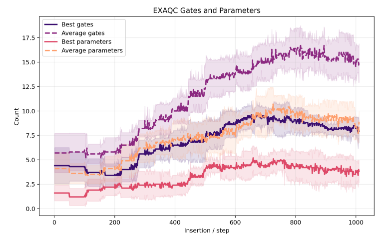
Fig. 3: Combined reward plots across repeated experiments. Plots show the best genome loss, loss of the top-5 genomes and the population average loss as a whole.

Even with a modest budget of evaluated genomes, we find that EXAQC is able to rapidly discover high-performing policies, particularly in the CartPole environment. The best genome achieves the maximum reward early in the search (within approximately 200–300 genome insertions), indicating that effective circuit structures can be identified quickly. FrozenLake shows interesting behavior in that it takes awhile for the evolved PQCs to achieve enough complexity, after which the performance rapidly increases. The top-k mean fitness exhibits a step-wise improvement pattern, reflecting the discrete nature of evolutionary updates as improved genomes enter the elite set. Meanwhile, the population mean increases more gradually, suggesting that beneficial circuit structures are progressively propagated throughout the population.

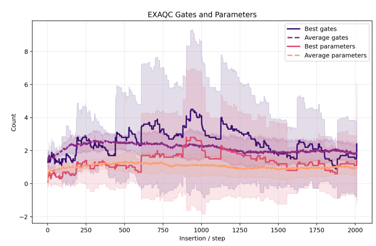
Across all environments the number of gates and parameters initially increases as evolution explores more expressive circuit structures, reflecting a phase of architectural expansion. Over time, the best-performing genomes tend to stabilize or even slightly reduce in complexity, indicating a pruning effect where redundant gates and parameters are removed in favor of more efficient representations. Notably, more complex environments (e.g., FrozenLake) sustain higher gate and parameter counts, while simpler or continuous control tasks (e.g., Cart-



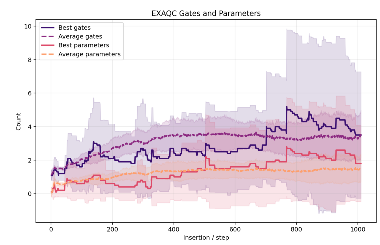
(a) CartPole



(b) FrozenLake



(c) Walker2d



(d) Mountaincar-Continuous

Fig. 4: Gate and parameter complexity plots across different RL environments for evolving genome populations. Plots out the gates and parameters for the best identified genome and the average metrics for the population.

Pole, MountainCar) converge to relatively compact circuits, suggesting that the evolutionary process adapts circuit complexity to task difficulty.

For the quantum circuit capable of solving CartPole (Figure 13a in Appendix 6.7) we can see that despite its small size, the circuit demonstrates that only a handful of parametrized rotations and limited entanglement are sufficient to capture the low-dimensional dynamics of CartPole. This compact structure highlights that expressive policies can emerge from minimal quantum resources, with the circuit efficiently encoding state information into a feature representation that is rich enough to solve the task. In contrast, for sparse and stochastic environments such as FrozenLake, the learning dynamics are significantly more challenging. Due to the sparse reward structure, most genomes receive near-zero reward during the early stages of evolution, resulting in a prolonged exploration phase. Once a successful trajectory is discovered, the best fitness exhibits sharp improvements; however, the population mean increases more slowly due to the difficulty of consistently reproducing successful policies under stochastic transitions. This leads to a larger and more persistent gap between elite and average performance compared to CartPole.

In continuous control environments such as MountainCarContinuous, EXAQC exhibits a more gradual but consistent learning trajectory compared to dense reward settings. The best fitness improves rapidly once effective momentum building strategies are discovered, reaching near optimal performance early in the evolutionary process. However, the population mean increases more steadily over time, reflecting the difficulty of propagating successful continuous control policies across the population. The gap between elite and average performance narrows slowly, indicating stable but progressive convergence as the evolutionary process refines control strategies under continuous action constraints. For more complex high-dimensional control tasks such as Walker2d, the learning dynamics exhibit slower but sustained improvement over the course of evolution. The best fitness steadily increases as increasingly coordinated locomotion strategies are discovered, while the population mean improves at a more gradual pace due to the difficulty of consistently evolving stable multi-joint control policies. The persistent gap between elite and average performance reflects the complexity of the task, where small structural differences in evolved circuits can lead to significant variations in long-horizon reward accumulation. Nevertheless, EX-AQC demonstrates the ability to progressively refine policies and achieve strong performance even in challenging continuous control settings.

		Cancer	**Iris**	**Seeds**	**Wine**
exponential crossover	global best	0.022	0.023	***0.026***	**0.027**
	inserted	0.461	*0.405*	**0.410**	0.446
	discarded	0.517	0.572	*0.564*	0.527
binary crossover	global best	*0.025*	0.025	0.016	0.024
	inserted	*0.465*	**0.417**	0.319	**0.473**
	discarded	0.510	0.558	0.665	0.503
reorder gate	global best	0.020	*0.026*	0.016	0.021
	inserted	0.411	0.365	0.348	0.398
	discarded	0.568	0.610	0.636	0.581
add gate	global best	**0.032**	0.022	*0.022*	*0.025*
	inserted	0.423	0.341	0.331	0.394
	discarded	0.545	0.637	0.647	0.581
disable gate	global best	0.019	0.020	*0.022*	0.013
	inserted	0.351	0.345	0.296	0.324
	discarded	0.630	0.634	0.683	0.663
clone	global best	**0.032**	*0.026*	**0.023**	0.021
	inserted	***0.532***	***0.424***	***0.419***	***0.536***
	discarded	***0.436***	***0.550***	***0.558***	***0.442***
qubit swap	global best	0.016	0.018	0.015	0.017
	inserted	0.352	0.305	0.276	0.330
	discarded	0.632	0.677	0.709	0.653
enable gate	global best	0.022	**0.029**	0.017	**0.027**
	inserted	**0.466**	0.398	*0.351*	0.429
	discarded	0.513	0.573	0.631	0.545
n-ary crossover	global best	***0.051***	***0.038***	***0.026***	***0.042***
	inserted	0.464	0.392	0.297	*0.459*
	discarded	**0.485**	**0.570**	0.677	**0.499**

Table 3: Performance rates (%) for each mutation and crossover operation for classification tasks. Best performance (high for inserts, low for discards) is bold and italics, second best is bold, and third best is italics.

		CartPole	FrozenLake	Mountaincar-C	Walker-2d
exponential crossover	global best	0.586	0.175	0.00	0.218
	inserted	34.700	24.650	23.214	19.214
	discarded	64.714	75.175	76.786	80.568
binary crossover	global best	**0.915**	0.248	0.00	0.355
	inserted	35.126	***32.298***	16.304	18.107
	discarded	*63.959*	***67.453***	83.696	81.538
reorder gate	global best	0.422	0.215	0.719	0.307
	inserted	*35.584*	27.957	19.424	18.985
	discarded	63.994	71.828	79.856	80.707
add gate	global best	***0.973***	*0.759*	0.687	***1.352***
	inserted	35.014	28.325	23.482	***23.771***
	discarded	64.013	70.916	**75.830**	***74.877***
disable gate	global best	0.538	0.519	0.00	0.141
	inserted	29.247	20.561	20.988	*20.845*
	discarded	70.215	78.920	79.012	*79.014*
clone	global best	0.510	***1.226***	**1.587**	**1.184**
	inserted	***39.371***	**30.298**	31.746	**22.587**
	discarded	***60.119***	**68.476**	***66.667***	**76.230**
qubit swap	global best	*0.878*	0.155	0.00	*1.025*
	inserted	26.262	22.257	20.455	16.956
	discarded	72.860	77.589	79.545	82.019
enable gate	global best	0.256	0.00	***1.724***	0.491
	inserted	**36.923**	19.330	12.069	16.953
	discarded	**62.821**	80.670	86.207	82.555
n-ary crossover	global best	0.586	**1.107**	*1.00*	0.326
	inserted	34.700	*30.233*	23.00	16.213
	discarded	64.714	*68.660*	*76.00*	83.460

Table 4: Performance rates (%) for each mutation and crossover operation for RL tasks. Best performance (high for inserts, low for discards) is bold and italics, second best is bold, and third best is italics.

4.3 Operator Evaluation

Tables 3 and 4 presents statistics for what happened to each genome across each of the 10 repeated experiments on the classification and RL benchmarks, respectively. If a genome was generated with a particular mutation or crossover operation, the percentage of time that genome was inserted into the population as a global best, inserted as a non-global best, or discarded was tracked. Unsurprisingly, across both tasks the clone mutation shows the highest insertion and lowest discard rates, as it refines existing good solutions with more backpropagation. Following this, for classification, across all datasets, n-ary crossover consistently found the most global best genomes, and additionally in three of four cases had the second lowest number of discards, showing very strong performance. Similarly, binary crossover, exponential crossover also showed good performance, highlighting the utility of the crossover methods utilized in this work. Mutations also mostly performed well with the exception of qubit swaps having the highest discard rates, however it still was capable of finding global best genomes. The higher discard rates from RL tasks (especially those who found small but strong performing PQCs) reflect the search spending more time on refinement than growth, as classification tasks tended to require larger PQCs.

These observations highlight the importance of exploration in evolutionary reinforcement learning. Initially, mutation operators such as *add gate* play a critical role in constructing viable circuit structures that connect inputs to outputs and enable meaningful policy representations. As evolution progresses, refinement operators such as *reorder gate*, *disable gate*, and *parameter tuning* become increasingly important for improving performance of already well-structured circuits. Balancing exploration and exploitation through adaptive mutation strategies could further enhance performance, particularly in sparse reward environments.

5 Conclusion

This work builds upon our prior research by extending the Evolutionary eXploration of Augmenting Quantum Circuits (EXAQC) framework to a broader and more versatile setting for quantum circuit design and training. The proposed methodology remains framework-agnostic, supporting almost all available gates in both Qiskit and Pennylane, while incorporating parallelization, advanced crossover strategies, and Lamarckian weight inheritance inspired by neuroevolution and genetic programming. Empirical results demonstrate that the evolved circuits not only achieve strong performance on classical classification benchmarks but can also approximate target quantum circuits with high fidelity. Notably, the extended framework further showcases promising results on reinforcement learning environments, highlighting its capability to generalize beyond supervised tasks. An analysis of EXAQC's mutation and crossover operations are provided, showing that all are effective in the evolutionary process, with the interesting finding that our novel n-ary crossover is one of the most effective operations.

Results reinforce the potential of EXAQC as a general-purpose quantum circuit optimization framework and motivate several directions for future work. The flexibility to incorporate arbitrary objective functions enables its application to diverse problem domains, including reinforcement learning, time series forecasting, and more complex classification settings such as computer vision. Future extensions will focus on enriching the framework with more comprehensive image benchmarks and combinatorial optimization problems, as well as improving the evolutionary search process through multi-population (island-based) models and speciation strategies. Incorporating multi-objective optimization, especially in regards to designing circuits which minimize error rates on specific quantum architectures, will also be a key direction, allowing EXAQC to better balance and exploit the variety of loss functions it can support and enabling it to design PQCs refined for different target quantum systems.

Acknowledgments. This work has been supported by the U.S. National Science Foundation under Grant Number #2225354. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We also thank RIT's Research Computing team for their assistance and support [47].

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Altmann, P., Stein, J., Kölle, M., Bärligea, A., Zorn, M., Gabor, T., Phan, T., Feld, S., Linnhoff-Popien, C.: Challenges for reinforcement learning in quantum circuit design. In: 2024 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 1, pp. 1600–1610. IEEE (2024)
2. Benítez-Buenache, A., Portell-Montserrat, Q.: Bayesian parameterized quantum circuit optimization (bpqco): A task and hardware-dependent approach. Quantum Machine Intelligence **7**(2), 85 (2025)
3. Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature communications **12**(1), 1791 (2021)
4. Chen, C.C., Watabe, M., Shiba, K., Sogabe, M., Sakamoto, K., Sogabe, T.: On the expressibility and overfitting of quantum circuit learning. ACM Transactions on Quantum Computing **2**(2), 1–24 (2021)
5. Chen, S.Y.C.: Evolutionary optimization for designing variational quantum circuits with high model capacity. In: 2025 IEEE Symposium for Multidisciplinary Computational Intelligence Incubators (MCII Companion). pp. 1–5. IEEE (2025)
6. Chen, S.Y.C., Huang, C.M., Hsing, C.W., Goan, H.S., Kao, Y.J.: Variational quantum reinforcement learning via evolutionary optimization. Machine Learning: Science and Technology **3**(1), 015025 (2022)
7. Chen, S.Y.C., Yang, C.H.H., Qi, J., Chen, P.Y., Ma, X., Goan, H.S.: Variational quantum circuits for deep reinforcement learning. IEEE access **8**, 141007–141024 (2020)
8. Chiribella, G., D'Ariano, G.M., Perinotti, P.: Quantum circuit architecture. Physical review letters **101**(6), 060401 (2008)
9. Ding, L., Spector, L.: Evolutionary quantum architecture search for parametrized quantum circuits. In: Proceedings of the genetic and evolutionary computation conference companion. pp. 2190–2195 (2022)
10. Ding, L., Spector, L.: Multi-objective evolutionary architecture search for parameterized quantum circuits. Entropy **25**(1), 93 (2023)
11. Drăgan, T.A., Monnet, M., Mendl, C.B., Lorenz, J.M.: Quantum reinforcement learning for solving a stochastic frozen lake environment and the impact of quantum architecture choices. arXiv preprint arXiv:2212.07932 (2022)
12. Du, Y., Huang, T., You, S., Hsieh, M.H., Tao, D.: Quantum circuit architecture search for variational quantum algorithms. npj Quantum Information **8**(1), 62 (2022)
13. Ewen, T., Turkalj, I., Holzer, P., Wolf, M.O.: Application of zx-calculus to quantum architecture search. Quantum Machine Intelligence **7**(1), 1–12 (2025)
14. Giovagnoli, A., Tresp, V., Ma, Y., Schubert, M.: Qneat: Natural evolution of variational quantum circuit architecture. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation. pp. 647–650 (2023)
15. Grimsley, H.R., Economou, S.E., Barnes, E., Mayhall, N.J.: An adaptive variational algorithm for exact molecular simulations on a quantum computer. Nature communications **10**(1), 3007 (2019)
16. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
17. Gustafson, E.J., Tiihonen, J., Chamaki, D., Sorourifar, F., Mullinax, J.W., Li, A.C., Maciejewski, F.B., Sawaya, N.P., Krogel, J.T., Bernal Neira, D.E., et al.:

Surrogate optimization of variational quantum circuits. Proceedings of the National Academy of Sciences **122**(36), e2408530122 (2025)
18. Hubregtsen, T., Pichlmeier, J., Stecher, P., Bertels, K.: Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. Quantum Machine Intelligence **3**(1), 9 (2021)
19. Jerbi, S., Gyurik, C., Marshall, S., Briegel, H., Dunjko, V.: Parametrized quantum policies for reinforcement learning. Advances in neural information processing systems **34**, 28362–28375 (2021)
20. Kar, D., Krutz, D., Desell, T.: Investigating quantum circuit designs using neuro-evolution. arXiv preprint arXiv:2602.03840 (2026)
21. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y., Balaprakash, P.: Learning to optimize variational quantum circuits to solve combinatorial problems. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 2367–2375 (2020)
22. Kini, A.D., Yadav, S.S., Thakur, A.S., Awari, A.B., Lyu, Z., Desell, T.: Co-evolving recurrent neural networks and their hyperparameters with simplex hyperparameter optimization. In: Proceedings of the Companion Conference on Genetic and Evolutionary Computation. pp. 1639–1647 (2023)
23. Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and quantum computation. No. 47, American Mathematical Soc. (2002)
24. Kölle, M., Bintener, T., Zorn, M., Stenzel, G., Sünkel, L., Gabor, T., Linnhoff-Popien, C.: Evaluating mutation techniques in genetic-algorithm-based quantum circuit synthesis. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 907–915 (2025)
25. Kölle, M., Seidl, D., Zorn, M., Altmann, P., Stein, J., Gabor, T.: Optimizing variational quantum circuits using metaheuristic strategies in reinforcement learning. In: 2024 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 2, pp. 323–328. IEEE (2024)
26. Kölle, M., Witter, T., Rohe, T., Stenzel, G., Altmann, P., Gabor, T.: A study on optimization techniques for variational quantum circuits in reinforcement learning. In: 2024 IEEE International Conference on Quantum Software (QSW). pp. 157–167. IEEE (2024)
27. Konda, V., Tsitsiklis, J.: Actor-critic algorithms. Advances in neural information processing systems **12** (1999)
28. Kubalik, J., Babuska, R.: Neuro-evolutionary approach to physics-aware symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1264–1272 (2025)
29. Li, P., Hao, J., Tang, H., Fu, X., Zhen, Y., Tang, K.: Bridging evolutionary algorithms and reinforcement learning: A comprehensive survey on hybrid algorithms. IEEE Transactions on evolutionary computation (2024)
30. Li, Y., Liu, R., Hao, X., Shang, R., Zhao, P., Jiao, L.: Eqnas: Evolutionary quantum neural architecture search for image classification. Neural Networks **168**, 471–483 (2023)
31. Li, Y., Zhang, J., Zhou, R., Qu, Y., Xu, R.: Aqea-qas: An adaptive quantum evolutionary algorithm for quantum architecture search. Entropy **27**(7), 733 (2025)
32. Liu, J., Lim, K.H., Wood, K.L., Huang, W., Guo, C., Huang, H.L.: Hybrid quantum-classical convolutional neural networks. Science China Physics, Mechanics & Astronomy **64**(9), 290311 (2021)
33. Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G.G., Tan, K.C.: A survey on evolutionary neural architecture search. IEEE transactions on neural networks and learning systems **34**(2), 550–570 (2021)

34. Lourens, M., Sinayskiy, I., Park, D.K., Blank, C., Petruccione, F.: Hierarchical quantum circuit representations for neural architecture search. npj Quantum Information **9**(1), 79 (2023)
35. Lukac, M., Perkowski, M., Goi, H., Pivtoraiko, M., Yu, C.H., Chung, K., Jeech, H., Kim, B.G., Kim, Y.D.: Evolutionary approach to quantum and reversible circuits synthesis. Artificial Intelligence Review **20**(3), 361–417 (2003)
36. Lyu, Z., ElSaid, A., Karns, J., Mkaouer, M., Desell, T.: An experimental study of weight initialization and lamarckian inheritance on neuroevolution. The 24th International Conference on the Applications of Evolutionary Computation (EvoStar: EvoApps) (2021)
37. Ma, Q., Hao, C., Yang, X., Qian, L., Zhang, H., Si, N., Xu, M., Qu, D.: Continuous evolution for efficient quantum architecture search. EPJ Quantum Technology **11**(1), 54 (2024)
38. Maldonado-Romo, A., Montiel-Pérez, J.Y., Onofre, V., Maldonado-Romo, J., Sossa-Azuela, J.H.: Quantum k-nearest neighbors: Utilizing qram and swap-test techniques for enhanced performance. Mathematics **12**(12), 1872 (2024)
39. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. Nature communications **9**(1), 4812 (2018)
40. Miranda, F.T., Balbi, P.P., Costa, P.: Synthesis of quantum circuits with an island genetic algorithm. arXiv preprint arXiv:2106.03115 (2021)
41. Murphy, J., Desell, T.: Minimizing the exa-gp graph-based genetic programming algorithm for interpretable time series forecasting. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1686–1690 (2024)
42. Murphy, J., Kar, D., Karns, J., Desell, T.: Exa-gp: Unifying graph-based genetic programming and neuroevolution for explainable time series forecasting. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 523–526 (2024)
43. Nguyen, N., Chen, K.C.: Quantum embedding search for quantum machine learning. IEEE Access **10**, 41444–41456 (2022)
44. Nicoli, K., Anders, C.J., Funcke, L., Hartung, T., Jansen, K., Kühn, S., Müller, K.R., Stornati, P., Kessel, P., Nakajima, S.: Physics-informed bayesian optimization of variational quantum circuits. Advances in Neural Information Processing Systems **36**, 18341–18376 (2023)
45. Ororbia, A., ElSaid, A., Desell, T.: Investigating recurrent neural network memory structures using neuro-evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 446–455. GECCO '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3321707.3321795, http://doi.acm.org/10.1145/3321707.3321795
46. Pesah, A., Cerezo, M., Wang, S., Volkoff, T., Sornborger, A.T., Coles, P.J.: Absence of barren plateaus in quantum convolutional neural networks. Physical Review X **11**(4), 041011 (2021)
47. Rochester Institute of Technology: Research Computing Services (2019). https://doi.org/10.34788/0S3G-QD15, https://doi.org/10.34788/0S3G-QD15, accessed 2026-01-23
48. Rubio, Y., Olvera, C., Montiel, O.: Towards a fully quantum learning system: Quantum architecture search with quantum evolutionary algorithms. In: Artificial Intelligence and Quantum Computing: Early Innovations. Volume 1, pp. 397–412. Springer (2025)

49. Ruican, C., Udrescu, M., Prodan, L., Vladutiu, M.: A genetic algorithm framework applied to quantum circuit synthesis. In: Nature inspired cooperative strategies for optimization (NICSO 2007), pp. 419–429. Springer (2008)
50. Schuld, M., Killoran, N.: Effect of data encoding on the expressive power of variational quantum machine learning models. Physical Review A **99**(3), 032331 (2019). https://doi.org/10.1103/PhysRevA.99.032331, https://arxiv.org/abs/1803.07128
51. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
52. Sequeira, A., Santos, L.P., Barbosa, L.S.: Policy gradients using variational quantum circuits. Quantum Machine Intelligence **5**(1), 18 (2023)
53. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
54. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies **2**(12), 1900070 (2019)
55. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2), 99–127 (2002)
56. Sünkel, L., Altmann, P., Kölle, M., Stenzel, G., Gabor, T., Linnhoff-Popien, C.: Quantum circuit construction and optimization through hybrid evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 934–942 (2025)
57. Sünkel, L., Martyniuk, D., Mattern, D., Jung, J., Paschke, A.: Ga4qco: genetic algorithm for quantum circuit optimization. arXiv preprint arXiv:2302.01303 (2023)
58. Sünkel, L., Stein, J., Nüßlein, J., Rohe, T., Linnhoff-Popien, C.: Evaluating variational quantum circuit architectures for distributed quantum computing. In: 2025 IEEE International Conference on Quantum Artificial Intelligence (QAI). pp. 343–350. IEEE (2025)
59. Thakur, A.S., Awari, A.B., Lyu, Z., Desell, T.: Efficient neuroevolution using island repopulation and simplex hyperparameter optimization. In: 2023 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1837–1842. IEEE (2023)
60. Towers, M., Kwiatkowski, A., Terry, J., Balis, J.U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al.: Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032 (2024)
61. Tudisco, A., Volpe, D., Ranieri, G., Curato, G., Ricossa, D., Graziano, M., Corbelletto, D.: Evaluating the computational advantages of the variational quantum circuit model in financial fraud detection. IEEE Access **12**, 102918–102940 (2024)
62. Wang, Y., Wang, Y., Chen, C., Jiang, R., Huang, W.: Development of variational quantum deep neural networks for image recognition. Neurocomputing **501**, 566–582 (2022)
63. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3), 229–256 (1992)
64. Woun, D.J., Date, P.: Adiabatic quantum support vector machines. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 2, pp. 296–297. IEEE (2023)
65. Zhang, A., Zhao, S.: Evolutionary-based quantum architecture search. arXiv preprint arXiv:2212.00421 (2022)
66. Zhang, A., Zhao, S.: Evolutionary-based searching method for quantum circuit architecture. Quantum Information Processing **22**(7), 283 (2023)

6 Appendix

6.1 Related Works

Recent advances in automated quantum circuit design are driven by two key challenges: circuit structure strongly influences trainability, expressivity, and hardware compatibility, and the search space over gate types, connectivity, and depth is combinatorial. Consequently, quantum circuit synthesis and quantum neural architecture search (QNAS) are increasingly formulated as optimization problems, often using evolutionary or neuroevolutionary algorithms that jointly explore discrete structures and continuous parameters.

One major direction focuses on constrained circuit synthesis. Sünkel et al. [56] propose a hybrid multi-objective evolutionary approach optimizing fidelity and circuit depth, achieving strong depth reductions with minimal fidelity loss. Unlike such work, which targets synthetic objectives, the present approach emphasizes dataset-driven tasks and backend-agnostic training across frameworks such as PennyLane and Qiskit. Similarly, Zhang and Zhao [65] apply evolutionary search to classification using measurement-based metrics, whereas this work additionally supports fidelity-based supervision when target mappings are available.

Neuroevolution-based methods such as QNEAT [14] co-evolve circuit topology and parameters, demonstrating effectiveness in RL settings. In contrast, the proposed framework employs a unified genome representation that supports both task-based losses and circuit-to-circuit or state fidelity objectives. Ding and Spector [10] further emphasize multi-objective optimization, balancing accuracy and circuit complexity, aligning with our use of structural metrics to discourage inefficient architectures.

In supervised learning, EQNAS [30], Li et al. [31], and Ma et al. [37] demonstrate evolutionary QNAS for classification tasks such as MNIST and CIFAR-10. However, these methods typically rely on measurement-space objectives and predefined templates. The present work extends this paradigm by enabling fidelity-based supervision when a target circuit is available, while also supporting standard dataset-driven losses through readout distributions.

Representation design is another key aspect. Lourens et al. [34] propose hierarchical encodings to improve scalability and genetic operations. Similarly, the genome representation here emphasizes modularity through gate specifications, register-aware wiring, and flexible input-output mappings, while supporting backend-independent execution and multiple training objectives. Ewen et al. [13] explore genetic programming with ZX-calculus transformations, highlighting semantics-aware mutation strategies compatible with this framework.

Hybrid quantum-classical architectures have also been explored. Liu et al. [32] and Rubio et al. [48] investigate evolutionary search in hybrid models. The present work instead focuses on a general-purpose circuit genome capable of supporting both fidelity-based and dataset-driven learning, while incorporating richer evaluation signals beyond accuracy.

Prior work in quantum reinforcement learning largely focuses on optimizing parameters of fixed circuit architectures. Chen *et.al.* [7] replace classical Q-networks with VQCs, while Jerbi et.al. [19] extend this to policy-gradient methods. Other studies highlight optimization strategies and training stability [26], or formulate circuit construction as an RL problem [1]. In contrast, the proposed work shifts focus from parameter tuning within fixed ansätze to evolving circuit structure itself, enabling discovery of transferable architectures across both supervised and RL tasks.

6.2 EXAQC's Distributed Asynchronous Framework

The evolutionary search is parallelized using a master-worker paradigm implemented with MPI via `mpi4py` (Refer Figure 5). The master process (rank 0) is responsible for generating candidate genomes and maintaining the population, while worker processes asynchronously request genomes, evaluate them using the objective function, and return the evaluated genomes back to the master for insertion into the population. This design enables efficient task distribution, as workers operate independently without synchronization barriers, allowing heterogeneous evaluation times across genomes. The asynchronous communication pattern ensures high resource utilization and scalability, significantly reducing overall search time for computationally expensive quantum circuit evaluations. The primary advantages of this parallelism include near-linear speedup with additional workers, improved exploration of the search space through increased evaluation throughput, and the ability to scale to large populations and complex objective functions without introducing significant coordination overhead. In its current initial implementation, EXAQC utilizes a single steady state population, however the codebase has been designed to allow for plugging in varying population strategies so it can be easily extended to utilize island or multiobjective strategies in future work.

6.3 Mutation and Crossover Operators

Add Gate (Figure 6a) places a gate at a random depth $d \sim U(0,1)$. To prevent adding gates which will not be used in useful computation, the gates in the network are traversed in order from 0 to d, to determine which qubits are affected by the inputs - these make the list of potential input qubits for the new gate. Similarly, the gates are traversed in reverse order from 1.0 to d, to determine which qubits are connected to the outputs by other gates – these make the list of potential output qubits for the new gate. If the list of input and output qubits are disjoint, only gates which have control (input) and target (output) qubits are allowed. The input qubits for the new gate are selected from the possible input qubits, and the output qubits for the new gate are selected from the possible output qubits. If the gate is parameterized, its parameters are initialized randomly $\sim U(-\pi, \pi)$. The parent circuit genome will be copied as the new child (inheriting parental circuit parameters), and the gate will be added to the child genome at depth d with a newly assigned innovation number.

Enable/Disable Gate (Figure 6b) first copies the parent as a child genome, then randomly selects from the child either an enabled gate to be disabled (for *disable gate*), or a disabled gate to be enabled (for *enable gate*). If *enable gate* is selected and there are no disabled gates, or *disable gate* is selected and there are no enabled gates, this mutation will return False so another mutation or crossover attempt can be made to generate a modified child.

Reorder Gate (Figure 6c) first copies the parent as a child genome, then randomly selects a gate within the child genome. It disables that gate, creates a copy of it (with a new innovation number), and assigns it a new depth $d \sim U(0, 1)$, a new innovation number, and adds it to the child genome.

Swap Qubits (Figure 6d) first copies the parent as a child genome, then randomly selects a gate within the child genome. It disables that gate, creates a copy of it (with a new innovation number), and then selects a qubit parameter at random with the gate and assigns it to a different qubit. If the qubit was an input qubit, the new qubit is randomly selected from all possible input qubits at that depth, if it was an output qubit, the new qubit is randomly selected from all possible output qubits at that depth. Otherwise (if it was both an input and output) it is randomly selected from all possible input and output qubits. Additionally, to prevent the new gate from having the same depth as the gate it was copied from, it is assigned a new depth $d \sim U(d_{prev}, d_{next})$, where d_{prev} is the depth of

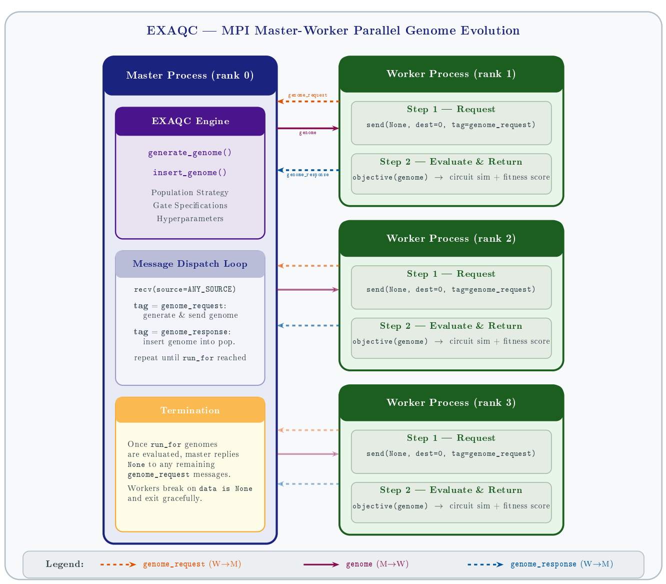


Fig. 5: Master-Worker Architecture in Evolving Genome Circuits and Tracking the Best Genomes *(sim is short for simulation)*

the previous gate in the (sorted) gate list, or 0 if there was not a gate before it, and d_{max} is the depth of the next gate in the gate list, or 1.0 if there was not a gate after it. This new gate is assigned a new innovation number and added to the child genome.

N-Ary Crossover (Figure 7) expands binary crossover to allow any number of parents. Parents are divided into the best fit parent, and all other parents. If a gate exists in the best fit parent and any other parent, it is added to the child genome. If a gate is only in the best fit parent, it is added at the *best_keep_rate* (0.75 for this work). If a gate exists in any other gate, but not the best fit parent, it is added to the child genome at the *other_keep_rate* (0.25 for this work). For parameterized gates in multiple parents, a randomized simplex approach is used [22, 59], where the randomized line search is performed between weight in the best parent, p_{best} and the average of the weights in the other parents p_{avg},

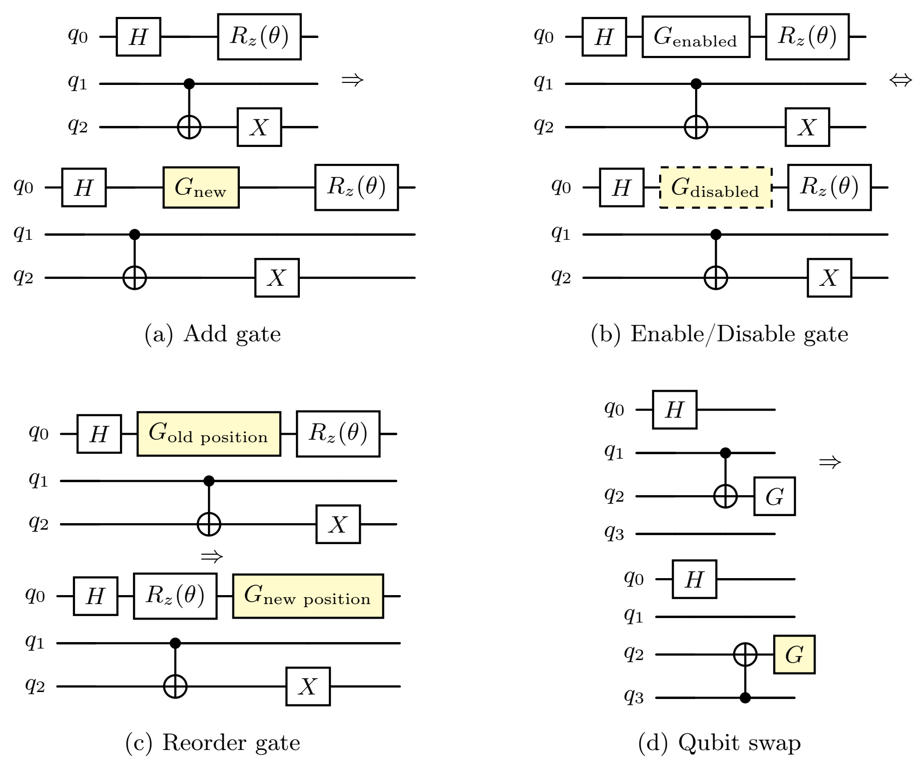
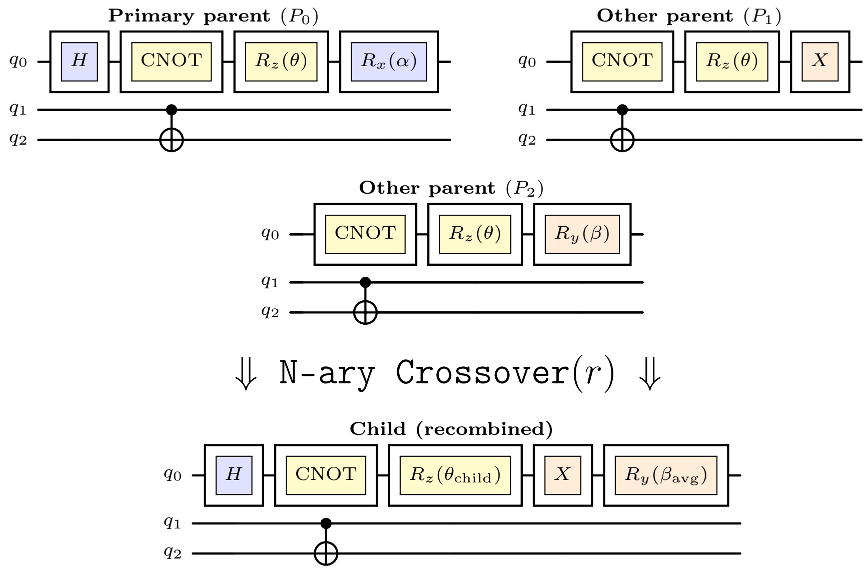


Fig. 6: Structural mutation operators used in hybrid evolutionary quantum circuit optimization. Highlighted gates indicate newly added or modified structures. Figures first presented in Kar et al. [20].



Selection rules: If innovation i appears in the primary parent and in at least one other parent (yellow), it always transfers to the child. If i appears only in the primary parent (blue), it transfers with probability $p_{\mathbf{p}}$. If i appears only in the other parents (orange), it transfers with probability p_o.
Parameter recombination (shared, parameterized gates): $r \leftarrow U(0,1)(c_2 - c_1) + c_1$, $\overline{\theta}_{\text{others}} = \frac{1}{|\mathcal{O}_i|}\sum_{k\in\mathcal{O}_i}\theta_k$, $\theta_{\text{child}} = \theta_{\text{o}} + r(\overline{\theta}_{\mathbf{p}} - \theta_{\text{o}})$.
Other-only parameter averaging (when multiple others): For a selected other-only innovation with parameters, set $\beta_{\text{avg}} = \frac{1}{m}\sum_{j=1}^{m}\beta_j$.

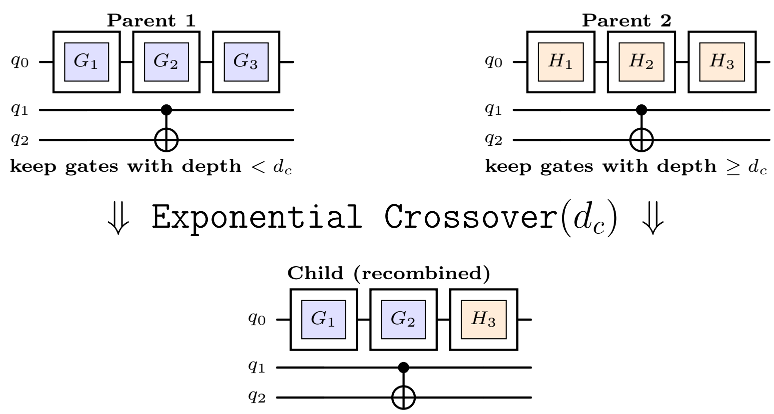
Fig. 7: N-ary crossover operator (from Kar et al. [20]).

where $l_1 = -1.0$ and $l_2 = 0.5$ are used as hyperparameters:

$$\begin{aligned} r &= (rand(0,1) * l_1) - l_2 \\ p_{new} &= p_{other} + r * (p_{best} - p_{other}) \end{aligned} \tag{6}$$

Gates which only come from one parent utilize the same weights. This follows EXAMM/EXA-GP's Lamarckian weight inheritance strategy [36]. *Binary crossover* is n-ary crossover where $n = 2$.

Exponential Crossover Exponential crossover (Figure 8) was also implemented as a way to recombine parents in a way that could potentially capture different useful gates from each parent. This creates a new child genome, and computes a random crossover depth $d_{crossover} \sim U(0,1)$. All gates from the first parent with depth $d < d_{crossover}$ are copied into the child genome, and all gates from the second parent with depth $d \geq d_{crossover}$ are copied from the second parent into the child genome.



Rule: Sample $d_c \sim U(0,1)$. Copy all gates from Parent 1 with depth $< d_c$, and all gates from Parent 2 with depth $\geq d_c$, into the child.

Fig. 8: Exponential crossover operator (from Kar et al. [20]).

6.4 Available Qiskit Gates

Gate	Method	Qubits	Parameters
Toffoli	ccx	control_qubit1, control_qubit2, target_qubit	
Symmetric	ccz	control_qubit1, control_qubit2, target_qubit	
Controlled Hadamard	ch	control_qubit, target_qubit	
Controlled Phase	cp	control_qubit, target_qubit	theta
Controlled RX	crx	control_qubit, target_qubit	theta
Controlled RY	cry	control_qubit, target_qubit	theta
Controlled RZ	crz	control_qubit, target_qubit	theta
Controlled S	cs	control_qubit, target_qubit	
Controlled $S^\dagger$	csdg	control_qubit, target_qubit	
Controlled SWAP (Fredkin)	cswap	control_qubit, target_qubit1, target_qubit2	
Controlled sqrt X	csx	control_qubit, target_qubit	
Controlled U	cu	control_qubit, target_qubit	theta, phi, lam, gamma
Controlled X	cx	control_qubit, target_qubit	
Controlled Y	cy	control_qubit, target_qubit	
Controlled Z	cz	control_qubit, target_qubit	
Double CNOT	dcx	qubit1, qubit2	
Echoed Cross-Resonance	ecr	qubit1, qubit2	
Hadamard	h	qubit	
Identity	id	qubit	
iSWAP	iswap	qubit1, qubit2	
Phase	p	qubit	theta
R	r	qubit	theta, phi
Simplified 3-Controlled Toffoli	rcccx	control_qubit1, control_qubit2, control_qubit3, target_qubit	
Simplified Toffoli (Margolus)	rccx	control_qubit1, control_qubit2, target_qubit	
RV	rv	qubit	vx, vy, vz
RX	rx	qubit	theta
RXX	rxx	qubit1, qubit2	theta
RY	ry	qubit	theta
RYY	ryy	qubit1, qubit2	theta
RZ	rz	qubit	phi
RZX	rzx	qubit1, qubit2	theta
RZZ	rzz	qubit1, qubit2	theta
S	s	qubit	
S-adjoint	sdg	qubit	
SWAP	swap	qubit1, qubit2	
sqrt X	sx	qubit	
inverse sqrt X	sxdg	qubit	
T	t	qubit	
T-adjoint	tdg	qubit	
U	u	qubit	theta, phi, lam
X	x	qubit	
Y	y	qubit	
z	z	qubit	

Table 5: Available Qiskit gates, their qubits and parameters (if parameterized).

6.5 Available Pennylane Gates

Gate	Method	Qubits	Parameters
Toffoli	ccx	control_qubit1, control_qubit2, target_qubit	
CCZ	ccz	control_qubit1, control_qubit2, target_qubit	
Controlled Hadamard	ch	control_qubit, target_qubit	
Controlled Phase	cp	control_qubit, target_qubit	phi
Controlled RX	crx	control_qubit, target_qubit	phi
Controlled RY	cry	control_qubit, target_qubit	phi
Controlled RZ	crz	control_qubit, target_qubit	phi
Controlled SWAP (Fredkin)	cswap	control_qubit, target_qubit1, target_qubit2	
Controlled X	cx	control_qubit, target_qubit	
Controlled Y	cy	control_qubit, target_qubit	
Controlled Z	cz	control_qubit, target_qubit	
Hadamard	h	qubit	
Identity	id	qubit	
iSWAP	iswap	qubit1, qubit2	
Phase	p	qubit	phi
RX	rx	qubit	phi
RY	ry	qubit	phi
RZ	rz	qubit	phi
RZZ	rzz	qubit1, qubit2	theta
S	s	qubit	
SWAP	swap	qubit1, qubit2	
T	t	qubit	
U	u	qubit	theta, phi, delta
X	x	qubit	
Y	y	qubit	
Z	z	qubit	

Table 6: Available Pennylane gates, their qubits and parameters (if parameterized).

6.6 Best Found Classification Circuits

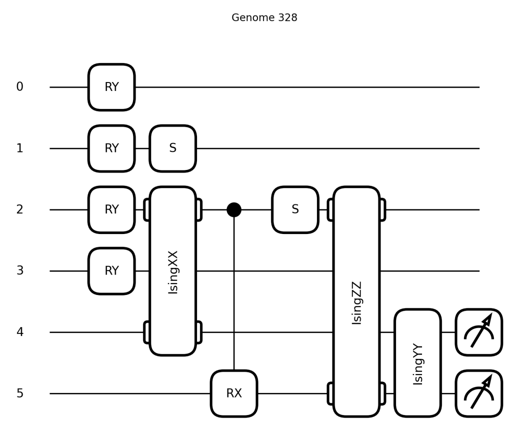


Fig. 9: Best found circuit for the iris benchmark.

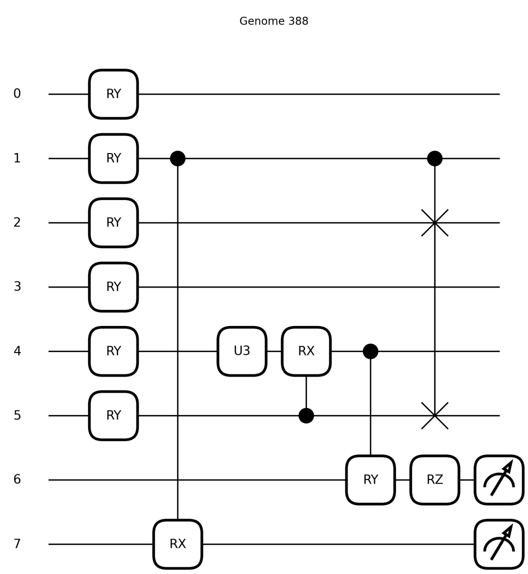


Fig. 10: Best found circuit for the seeds benchmark.

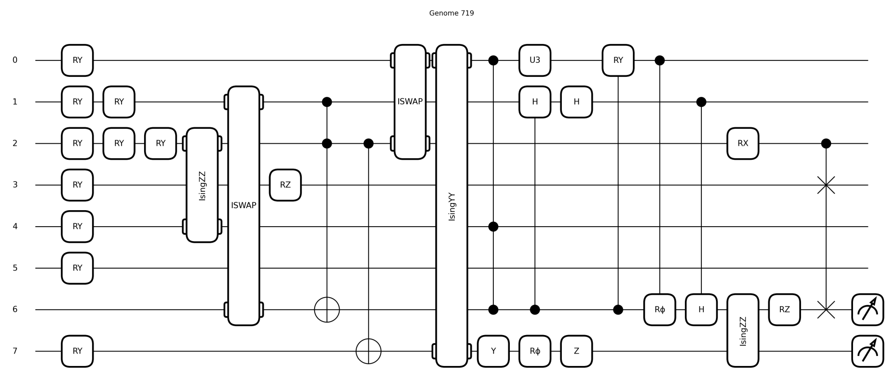


Fig. 11: Best found circuit for the wine benchmark.

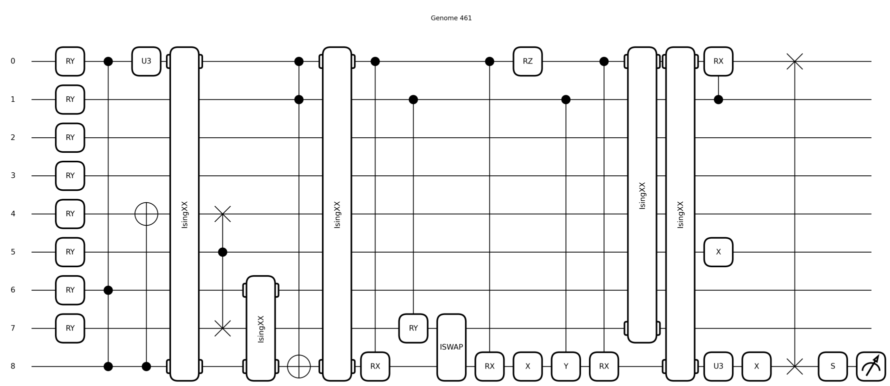
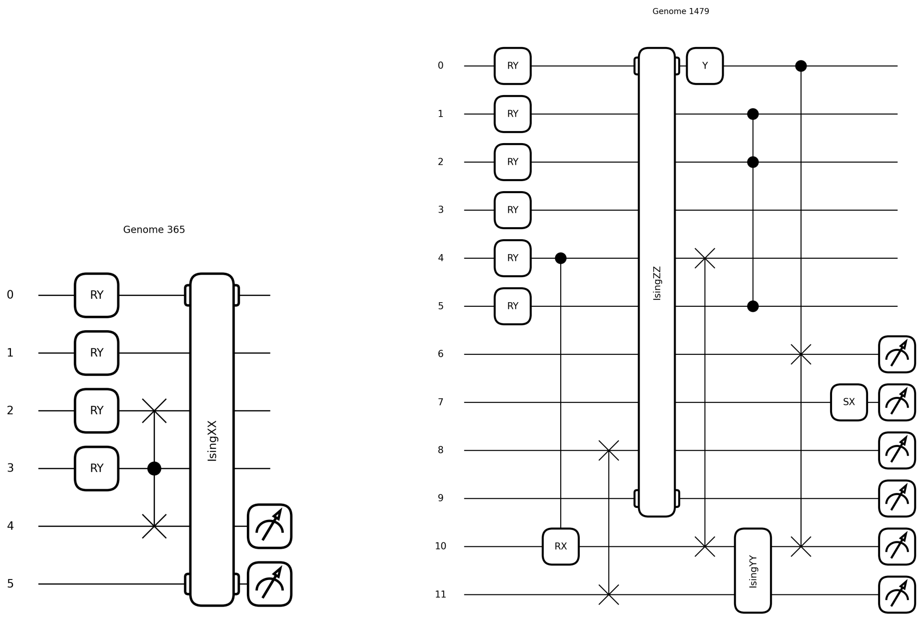


Fig. 12: Best found circuit for the breast cancer benchmark.

6.7 Best Found Reinforcement Learning Circuits



(a) Best found circuit for CartPole.

(b) Best found circuit for Walker-2d.

Fig. 13: Best evolved circuits for RL environments.

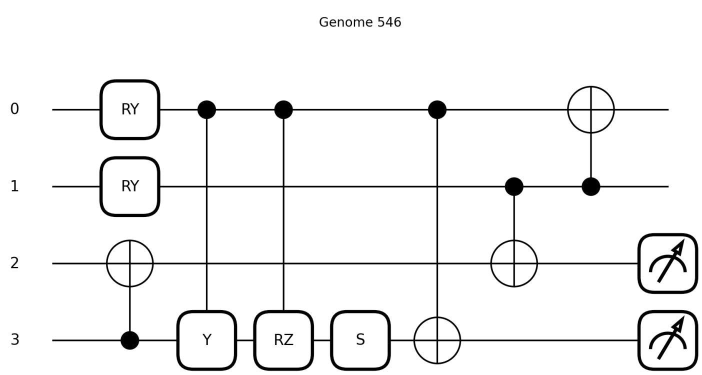


Fig. 14: Best found circuit for Mountaincar-Continuous.