Challenges for Reinforcement Learning in Quantum Circuit Design

Philipp Altmann
LMU Munich
Munich, Germany
philipp.altmann@ifi.lmu.de

Jonas Stein
LMU Munich
Munich, Germany

Michael Kölle
LMU Munich
Munich, Germany

Adelina Bärligea
TU Munich
Munich, Germany

Maximilian Zorn
LMU Munich
Munich, Germany

Thomas Gabor
LMU Munich
Munich, Germany

Thomy Phan
University of Southern California
Los Angeles, USA

Sebastian Feld
Delft University of Technology
Delft, The Netherlands

Claudia Linnhoff-Popien
LMU Munich
Munich, Germany

***Abstract*—Quantum computing (QC) in the current NISQ era is still limited in size and precision. Hybrid applications mitigating those shortcomings are prevalent to gain early insight and advantages. Hybrid quantum machine learning (QML) comprises both the application of QC to improve machine learning (ML) and ML to improve QC architectures. This work considers the latter, leveraging reinforcement learning (RL) to improve quantum circuit design (QCD), which we formalize by a set of generic objectives. Furthermore, we propose `qcd-gym`, a concrete framework formalized as a Markov decision process, to enable learning policies capable of controlling a universal set of continuously parameterized quantum gates. Finally, we provide benchmark comparisons to assess the shortcomings and strengths of current state-of-the-art RL algorithms.**

***Index Terms*—Reinforcement Learning, Quantum Computing, Circuit Optimization, Architecture Search**

I. Introduction

Quantum circuits comprise a combination of unitary compositions, or quantum gates, executed on a quantum computer to alter a particular quantum state. Mostly driven by theoretical promises of achieving exponential speed-ups in computational tasks such as integer factorization [2] and solving linear equation systems [3], quantum computing (QC) has become a rapidly growing field of research. However, the current era of *noisy intermediate-scale quantum* (NISQ) hardware [4] poses considerable limitations that prevent the realization of these speed-ups. Moreover, prevalent circuit architectures are mostly handcrafted. Therefore, the central task of *quantum circuit design* (QCD) consists of finding a sequence of quantum gates to achieve a specific objective, potentially given (hardware) limitations. We mainly consider two categories, architecture search and circuit optimization, more specifically, the preparation of arbitrary states and the composition of unitary operations. Among other ML approaches, we find *reinforcement learning* (RL) to be specifically suited for these objectives, learning tasks in various discrete and continuous sequential decision-making problems with temporal dependence, without the need for granular specification. Also, sophisticated mechanisms balancing exploration and exploitation provide a helpful foundation for discovering viable architectures. Through this research, we aim to provide comprehensive insight into various objectives associated with quantum circuit design, fostering collaboration and innovation within the AI research community. However, in contrast to most related work, we approach this topic bottom-up by dividing problems of interest into their core objectives, aiming to shed light on the challenges that current RL algorithms face in this field of research and laying the groundwork for all-embracing automated circuit design.

This is an extended version of [1].

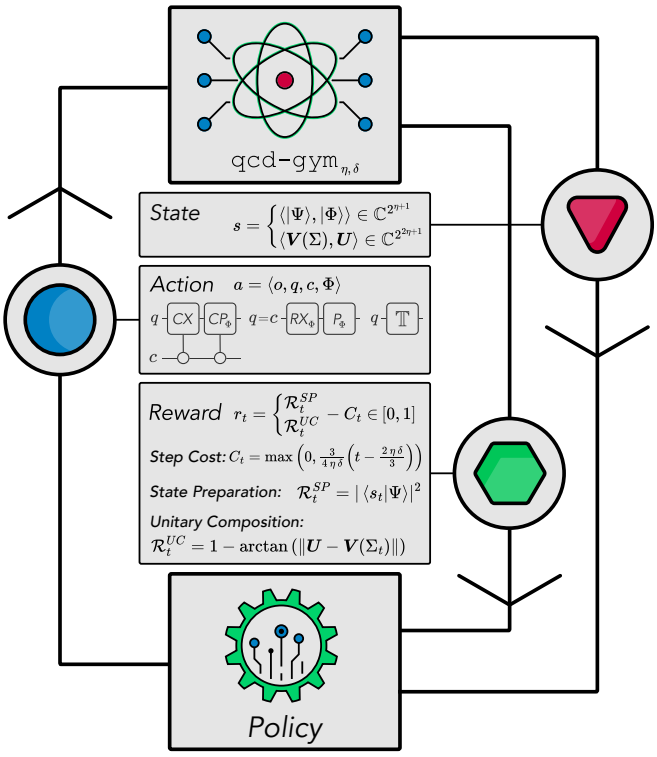


Fig. 1. `qcd-gym` for η qubits with depth δ for generating a sequence Σ of continuously parameterized operation a (blue) to optimally resemble a target state $|\Phi\rangle$ or unitary $\boldsymbol{U}$ in a single optimization loop, based on the observed state s (red) and the reward r_t (green).

979-8-3315-4137-8/24/$31.00 ©2024 IEEE
DOI 10.1109/QCE60285.2024.00187

Overall, we provide the following contributions:

- We formulate core objectives for reinforcement learning arising from quantum circuit design (QCD).
- We define `qcd-gym`, a generic RL framework for QCD to learn executing parameterized operations (cf. Fig. 1).
- We demonstrate the challenges current state-of-the-art RL approaches face in QCD.

II. Background

A. Quantum Computing

First, we provide a brief overview of the basic concepts of quantum computing. For more detailed information, see Nielsen and Chuang [5]. A quantum computer is built from qubits (wires) to execute a quantum circuit defined by a sequence of elementary operations (quantum gates) to carry and manipulate the quantum state. Unlike classical bits, which can be in the state 0 or 1, qubits can take on any superposition of these two, so their state can be represented as:

$$\alpha\left|\mathbf{0}\right\rangle + \beta\left|\mathbf{1}\right\rangle = \alpha\begin{bmatrix}1\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \mathbb{C}^2. \tag{1}$$

$\left|\mathbf{0}\right\rangle$ and $\left|\mathbf{1}\right\rangle$ are quantum computational basis states of the two-dimensional Hilbert space spanned by one qubit, and $|\alpha|^2 + |\beta|^2 = 1$. A two-qubit state can be represented by $\left|\boldsymbol{ab}\right\rangle = \left|\boldsymbol{a}\right\rangle\left|\boldsymbol{b}\right\rangle = \left|\boldsymbol{a}\right\rangle \otimes \left|\boldsymbol{b}\right\rangle \in \mathbb{C}^{2^2}$, with the 2^2 computational basis states $\left|\mathbf{00}\right\rangle$, $\left|\mathbf{01}\right\rangle$, $\left|\mathbf{10}\right\rangle$, and $\left|\mathbf{11}\right\rangle$. Following this formulation, quantum gates can be understood as unitary matrix operations $\boldsymbol{U} \in \mathbb{C}^{2^n \times 2^n}$ applied to the state vector of the n qubits, on which the gate acts. Their only condition of being unitary (i.e., $\boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{I}$) ensures that the operations inherent in them are reversible. With the identity $\boldsymbol{I}$, the most elementary single-gate operations contain:

$$\boldsymbol{I} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix};\ \boldsymbol{X} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix};\ \boldsymbol{Z} = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}. \tag{2}$$

To represent arbitrary quantum operations also requires two-qubit gates, such as the well-known CNOT gate, which is the quantum version of the classical XOR gate:

$$\boldsymbol{CX}_{a,b} = \left|0\right\rangle\left\langle 0\right| \otimes \boldsymbol{I} + \left|1\right\rangle\left\langle 1\right| \otimes \boldsymbol{X} \tag{3}$$

A CNOT gate flips the state of the target qubit $\left|\boldsymbol{b}\right\rangle$ if and only if the control qubit $\left|\boldsymbol{a}\right\rangle = \left|\mathbf{1}\right\rangle$. In fact, it can be shown that any multi-qubit logic gate (i.e., unitary matrix of arbitrary size) can be composed of CNOT and single-qubit gates, which is a concept called *universality* of gate sets [6]. One universal set of gates, close to current quantum hardware, consists of CNOT (3), X-Rotation (4), and PhaseShift (5) operations [7]:

$$\boldsymbol{RX}(\lambda) = \exp\left(-i\frac{\lambda}{2}\boldsymbol{X}\right) \tag{4}$$

$$\boldsymbol{P}(\phi) = \exp\left(i\frac{\phi}{2}\right) \cdot \exp\left(-i\frac{\phi}{2}\boldsymbol{Z}\right) \tag{5}$$

In addition, we use the *Controlled-PhaseShift* defined as:

$$\boldsymbol{CP}(\phi) = \boldsymbol{I} \otimes \left|0\right\rangle\left\langle 0\right| + \boldsymbol{P}(\phi) \otimes \left|1\right\rangle\left\langle 1\right| \tag{6}$$

Furthermore we refer to the gate set $\{\boldsymbol{CX}, \boldsymbol{H}, \boldsymbol{S}\}$ as the *Clifford group*, representing an alternative universal gate set with $\boldsymbol{S} = \boldsymbol{P}(\pi/2)$, and the Hadamard operator $\boldsymbol{H}$:

$$\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} \tag{7}$$

B. Reinforcement Learning

To reflect the temporal component of executing operations in a quantum circuit, we formulate the problem of quantum circuit design as a *Markov decision process (MDP)* $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ with discrete time steps t, a set $\mathcal{S}$ of states s_t, a set $\mathcal{A}$ of actions a_t, the transition probability $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ from s_t to s_{t+1} when executing a_t, a scalar reward $r_t = \mathcal{R}(s_t, a_t)$, and the discount factor $\gamma \in [0, 1)$ for calculating the *discounted return* [8]:

$$G_t = \sum_{k=0}^{\infty} \gamma^{t+k} r_{t+k} \tag{8}$$

The goal is to find an optimal *policy* π^* with the action-selection probability $\pi(a_t \mid s_t)$ that maximizes the expected return. Policy π can be evaluated with the *value function* $Q^\pi(s_t, a_t) = \mathbb{E}_\pi[G_t \mid s_t, a_t]$ or the *state-value function* $V^\pi(s_t) = \mathbb{E}_{\pi,\mathcal{P}}[G_t \mid s_t]$. The *optimal value function* $Q^* = Q^{\pi^*}$ of any *optimal policy* π^* satisfies $Q^* \geq Q^{\pi'}$ for all policies π' and state-action pairs $\langle s_t, a_t \rangle \in \mathcal{S} \times \mathcal{A}$.

Value-based RL approaches such as DQN approximate the optimal value function Q^* using parameterized function approximators Q_θ to derive a behavior policy π based on multiarmed bandit selection [9], [10]. Alternatively, *Policy-based RL* approximates π^* from trajectories τ of experience tuples $\langle s_t, a_t, r_t, s_{t+1} \rangle$ generated by π_θ, directly learning the policy parameters θ via gradient ascent on $\nabla_\theta \mathbb{E}_{\pi_\theta}[G(\tau)]$ [9], [11]. For simplicity, we omit the parameter indices θ of π, V, and Q for the following. *Advantageous Actor-Critic* (A2C) incorporates the value network V^π to build the advantage $A_t^\pi = Q^\pi(s_t, a_t) - V^\pi(s_t)$, enabling updates using policy roll-outs of incomplete episodes, where π represents the actor and V^π represents the critic [12]. *Proximal Policy Optimization* (PPO) offers an alternative approach, maximizing the surrogate objective

$$L = \min\left\{\omega_t A_t^\pi, \text{clip}(\omega_t, 1-\epsilon, 1+\epsilon)A_t^\pi\right\}, \tag{9}$$

with $\omega_t = \frac{\pi(a_t|s_t)}{\pi_{old}(a_t|s_t)}$, where $\pi_{old}(a_t \mid s_t)$ is the previous policy, and the clipping parameter ϵ restricts the magnitude of policy updates [13]. PPO has been recently proven to ensure stable policy improvement, thus being a theoretically sound alternative to vanilla actor-critic approaches [14]. Similarly, applying clipping to twin Q functions in combination with delayed policy updates with *Twin Delayed DDPG* (TD3) has been shown to improve *Deep Deterministic Policy Gradient* (DDPG), approximating both Q and π [15]. Building upon those improvements, *Soft Actor Critic* (SAC) bridges the gap between value-based Q-learning and policy gradient approaches and has shown state-of-the-art performance in various continuous control robotic benchmark tasks [16].

Yet, most RL algorithms face central challenges, such as exploring sparse reward landscapes and high-dimensional (continuous) action spaces. Nevertheless, besides robotic control or image-based games, RL has already been successfully applied to a wide range of current challenges in QC [17]–[19]. By considering quantum circuit design as a sequential decision-making problem solved via RL, we hope to explore scalable yet adaptable approaches for their automated construction and optimization. Note that alternative sequential decision-making approaches, like planning, might also be utilized.

III. Quantum Circuit Design Objectives for Reinforcement Learning

For a unified collection of RL challenges that arise from quantum architecture search and circuit optimization, we formalize generic *state preparation* and *unitary composition* objectives along with reward measures and specific target states in the following.

A. State Preparation (SP)

SP denotes the objective of finding a sequence of operations that will generate a desired quantum state from a given initial state $|\mathbf{0}\rangle^{\otimes\eta}$, with the number of available qubits η. Denoting the final output state as $s_t = |\mathbf{\Phi}\rangle \in \mathbb{C}^{2^\eta}$, the generated gate sequence can be seen as the unitary mapping

$$\mathcal{U} : |\mathbf{0}\rangle^{\otimes n} \mapsto |\Phi\rangle \tag{10}$$

We measure success in this objective based on the distance between $|\Phi\rangle$ and the target state $|\Psi\rangle$ using the *fidelity* F, measuring the overlap between the two states:

$$F(\Phi, \Psi) = |\langle \Phi \mid \Psi \rangle|^2 \quad \in \ [0, 1] \tag{11}$$

This definition intrinsically suggests a setting of sparse rewards for the SP objective, as we are not interested in the quantum state during an episode but rather want to maximize the final fidelity. Therefore, we define the accompanying reward as follows:

$$\mathcal{R}^{SP}(s_t, a_t) = F(s_t, \Psi) \tag{12}$$

As a proof of concept, we propose using the following states:

- The *Bell state*, to reflect basic entanglement:

$$|\Phi^+\rangle \equiv \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{13}$$

- The *Greenberger–Horne–Zeilinger state* (GHZ), to reflect entanglement of a variable number of n qubits:

$$|GHZ\rangle \equiv \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}} \tag{14}$$

- The *Haar random state*, as a benchmark for preparing an arbitrary state:

$$|\tilde{U}\rangle = \tilde{U} \, |\mathbf{0}\rangle^{\otimes\eta} \tag{15}$$

However, note that further, more intricate states are easily extendable without any adaptations to the proposed framework. Also, ignoring the global phase, this objective requires less accurate reproduction of the target, which we rectify by an alternative distance measure used for the following objective.

B. Unitary Composition (UC)

UC denotes the objective of composing an arbitrary unitary transformation with only a finite choice of operations. UC can also be seen as a more general case of SP, in which the mapping of Eq. (10) must not only apply to a specific input state but to all possible inputs. Note that while the choice of operations is finite, the gate set itself is arguably infinite due to including the parameter Φ into the action space. A necessary condition for decomposing a theoretical unitary circuit into a primitive set of operations is that this set is universal by means of Dawson and Nielsen [20]. The *Solovay-Kitaev theorem*, however, states that there exists a finite sequence of gates that approximate any target unitary transformation $\boldsymbol{U}$ to arbitrary accuracy without providing such a sequence. Assuming that any circuit or sequence of gates $\Sigma_t = \langle a_0, \ldots, a_t \rangle$, generated by policy π, can be represented as a unitary matrix $\boldsymbol{V}(\Sigma_t) \in \mathbb{C}^{2^\eta \times 2^\eta}$, we define the distance D to the target V as:

$$D(\boldsymbol{U}, \boldsymbol{V}(\Sigma_t)) = \|\boldsymbol{U} - \boldsymbol{V}(\Sigma_t)\|^2 , \tag{16}$$

with $\|\cdot\|$ denoting the Frobenius norm. As this can take on any positive real number and still reflects a loss to be minimized, we propose the following bounded *similarity* score as a reward to be maximized for unitary composition:

$$R^{UC}(a_t, s_t) = 1 - \arctan(D(\boldsymbol{U}, \boldsymbol{V}(\Sigma_t))) \tag{17}$$

As a proof of concept, we use the following unitaries:

- The *Hadamard* operator $\boldsymbol{H}$ (7), creating superposition
- A random operator $\tilde{\boldsymbol{U}}$ according to Ozols [21]
- The *Toffoli* operator defined as:

$$CCX_{a,b,c} = |0\rangle\langle 0| \otimes \boldsymbol{I} \otimes \boldsymbol{I} + |1\rangle\langle 1| \otimes \boldsymbol{CX}, \tag{18}$$

Lastly, we consider *Hamiltonian simulation* (HS), which can be formalized as follows. Given the Hamiltonian $\mathcal{H}$ of a quantum system, the solution to the Schrödinger equation yields a time evolution of the system according to $|\Psi(\tau)\rangle = \exp(-i\mathcal{H}\tau)\,|\Psi_0\rangle$. According to the Suzuki-Trotter formula, these dynamics can be approximately factorized with a rigorously upper-bounded error [22]. Thus, a sequence Σ_t can be optimally chosen to produce

$$|\Phi\rangle = \left(\prod_t \Sigma_t \right) |\Psi\rangle , \tag{19}$$

such that $|\Phi\rangle$ is as close as possible to $|\Psi\rangle$ [23]. However, it can be observed that the simulation of the time evolution of a Hamiltonian $\mathcal{H}$ is mathematically equivalent to the UC of the unitary operator $\exp(-i\mathcal{H}\tau)$. While this fact can typically not be exploited in practice due to the required exponential computational resources in classical computing, we argue that RL should perform approximately equally in the UC and HS objectives due to their mathematical equivalence.

IV. Quantum Circuit Designer

To use RL for learning to solve the previously proposed objectives, we define `qcd-gym`, a generic *quantum circuit design* (QCD) environment formalized as an MDP. Fig. 1 shows a schematic rendering of the environment. To ensure maximal compatibility with various use cases and scenarios down the line, our framework only corresponds to a quantum device (simulator or real hardware), parameterized by its number of available qubits η and its maximally feasible circuit depth δ. Besides these parameters, used to constrain generated circuits to a practicable extent, we omit additional tunable parameters like solution thresholds. To constantly monitor the current circuit, $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ is implemented by a quantum simulator, which allows for efficient state vector readout.

A. State

At each time step t, the agent observes the state s_t reflecting the current quantum circuit, represented by the full complex state vector $|\Psi\rangle$ or the unitary operator $\boldsymbol{V}(\Sigma_t)$ resulting from the current sequence of operations Σ_t, and the intended target:

$$s_t = \begin{cases} \langle |\Psi\rangle, |\Phi\rangle \rangle \in \mathbb{C}^{2^{\eta+1}} & \textit{target state } |\Phi\rangle \\ \langle \boldsymbol{V}(\Sigma_t), \boldsymbol{U} \rangle \in \mathbb{C}^{2^{2\eta+1}} & \textit{target unitary } \boldsymbol{U} \end{cases} \tag{20}$$

This state representation is chosen as it contains the maximum amount of information about the output of the circuit. Although this information is only available in quantum simulators (on real hardware, $\mathcal{O}(2^\eta)$ many measurements would be needed), it represents a starting point for RL from which future work should extract a sufficient and efficiently obtainable subset of information. Furthermore, this state representation is sufficient for defining an MDP-compliant environment, as operations in this state must be reversible. Thus, no history of states is needed, and decisions can be made based solely on the current state. Similar to continuous robotic control tasks [24], we additionally provide the intended target to be observed. Note that this addition does not require additional information, as the target representation is also necessary to calculate the reward. Furthermore, this representation can be extended to allow for mid-circuit measurements when switching to a density matrix-based representation of the state, however, at the cost of a quadratically larger state space. All available qubits are initially in state $|0\rangle$.

B. Action

Given this observation, the agent needs to choose a suitable action consisting of three fragments: The operation Γ, the target Ω, and the parameter Φ. To provide a balanced action space, we propose using an extension of the previously defined universal gate set, giving the agent a choice between controlled or uncontrolled operations along the $\mathbb{X}$ or $\mathbb{Z}$ axis. Additionally, the agent can choose to terminate the current episode, resulting in the following operation choice $\Gamma \in \{\mathbb{Z}, \mathbb{X}, \mathbb{T}\}$. Upon episode termination, all unused qubits are disregarded. Thus, the additional *terminate* action $\mathbb{T}$ enables the optimization of compact yet effective architectures within the defined limits for available qubits η and feasible depth δ. If not terminated, an episode is truncated once the maximum circuit depth δ is reached. Secondly, the agent needs to choose the target of the operation, one qubit q for applying the operation, and one control qubit c, $\Omega = \{q, c\} \in [0, \eta - 1]^2$. For the *terminate* operation, the target fragment is discarded. If the operation qubit equals the control qubit, an uncontrolled operation is performed, resulting in the following set of gates $\mathbb{G} = \{\boldsymbol{RX}, \boldsymbol{CX}, \boldsymbol{P}, \boldsymbol{CP}\}$ (cf. Eqs. (3)-(6)) to be applied:

$$g(o, q, c) : \Gamma \times \Omega \mapsto \mathbb{G} = \begin{cases} \begin{cases} \boldsymbol{RX} & q = c \\ \boldsymbol{CX} & q \neq c \end{cases} & o = \mathbb{X} \\ \begin{cases} \boldsymbol{P} & q = c \\ \boldsymbol{CP} & q \neq c \end{cases} & o = \mathbb{Z} \end{cases} \tag{21}$$

Selecting actions from a universal set of gates, based on the current state, ensures the production of only valid circuits. In contrast to most related work, we refrain from using a discrete action space with a gate set like *Clifford + T*. Instead, we aim to exploit the capabilities of RL to perform continuous control over the gate parameterization. Thereby, we enable granular state-control via the last action fragment $\Phi \in [-\pi, \pi]$, facilitating the generation of optimized circuits in a unified circuit design approach. Otherwise, a second optimization pass would be needed to improve the placed continuous gates, increasing the application's overall complexity. Also, using an external optimizer for parameterization does not suit our approach, as we aim to benchmark the suitability of RL for the design of optimized quantum circuits. Note that the parameter is omitted for the CNOT gate $\boldsymbol{CX}$. This helps reduce the action space and is motivated by the empirical lack of CRX gates in most circuit architectures. Overall, this results in the 4-dimensional action space $\langle o_t, q_t, c_t, \Phi_t \rangle = a_t \in \mathcal{A} = \{\Gamma \times \Omega \times \Theta\}$. Thus, following Eq. (21), the operation $U = g(o_t, q_t, c_t)(\Phi_t)$ is applied at time step t.

C. Reward

The reward is kept at 0 until the end of an episode is reached (i.e., t is terminal, either by truncation or termination). To incentivize the use of few operations, a step-cost $\mathcal{C}_t$ is applied when exceeding two-thirds of the available operations σ:

$$\mathcal{C}_t = \max\left(0, \frac{3}{2\sigma}\left(t - \frac{\sigma}{3}\right)\right), \tag{22}$$

with $\sigma = \eta \cdot \delta \cdot 2$. We choose this bound to prevent prevalent solutions solely optimizing for minimal qubit and operation usage, which would cause immediate termination of the episode. To further incentivize the construction of useful circuits, we use the previously defined task reward functions $\mathcal{R}^*$ according to Eq. (12) and Eq. (17) for the SP and UC objectives respectively, s.t.:

$$\mathcal{R} = \begin{cases} \mathcal{R}^*(s_t, a_t) - C_t, & \text{if } t \text{ is terminal.} \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

However, in contrast to the overall domain, those are specific to the given problem instance or objective, defined previously, and might be extended to address further objectives.

V. Related Work

A. Quantum Architecture Search

Currently, prevailing quantum circuit architectures are either manually designed – encoding problem-specific domain-knowledge in, e.g., *variational quantum eigensolvers* (VQEs) [25] and approximate methods like the *quantum approximate optimization algorithm* model (QAOA) [26] – or heuristically constructed via iterative search. In this work, we specifically consider quantum architecture search for the composition of unitary gate circuit (UC), the preparation of quantum states (SP), and the simulation of a given Hamiltonian (HS) [27]. *Quantum architecture search* (QAS) denotes the process of automatically engineering quantum circuits to find a sequence of quantum gates achieving specific tasks. Due to the likeness to the similarly connected and layered classical (neural-)network structures, QAS, in some cases, also takes inspiration from neural architecture search (NAS) [28]. Hence, deep-learning NAS concepts like augmented [29] or differentiable architecture search [30] and topological (graph-)augmentation [31] have found application in QAS in similar form [32].

Recently, evolutionary-inspired algorithms are also commonly used for their effectiveness in searching large problem spaces [33]–[36]. However, their convergence to a few best solutions out of a preset population of candidates does imply the generation of finely specialized, problem-specific circuits.

In contrast, motivated by the sequential nature of quantum circuits, we approach QAS as a dynamic decision-making problem. Therefore, we optimize a policy to select actions consisting of a quantum gate placement such that the final circuit maximizes the cumulative reward for a given objective. Utilizing RL/ML for this purpose has been shown to work with both the optimization and the generation of quantum circuits. Ostaszewski, Trenkwalder, Masarczyk, *et al.* [37], for instance, propose using RL to iteratively generate VQE Hamiltonians, or more generally, Mortazavi, Qin, and Yan [38] propose a single-step MDP for the exploration of optimization sample-choices. However, their work denotes a problem-specific top-down approach, whereas we strive for a bottom-up approach, formulating generic RL challenges from the field of QAS. Likewise, RL approaches have been proposed to prepare certain quantum states of varying difficulty [39]–[41], producing a more generic challenge. Similar to Sogabe, Kimura, Chen, *et al.* [40], we propose a reward based on the distance between the final and target states, where Kuo, Fang, and Chen [39] use a distance threshold, introducing an additional hyperparameter.

To the best of our knowledge, existing approaches essentially utilize discrete gate sets, such as the Clifford group defined above, e.g., [37], [42]. Using operations with predefined parameterizations might require a two-step procedure: generating a circuit architecture and then optimizing the circuit parameterization. Classical optimization routines are traditionally employed to perform this task, including gradient-free approaches, such as evolutionary algorithms, and gradient-based approaches using the parameter-shift rule [43]. Note that both methods require numerous circuit executions to find optimal parameters. In contrast, our approach seeks to integrate this parameter optimization task directly into the RL environment by enabling continuous actions. Thus, agents inherently seek optimal parameters in the generated circuits.

B. Quantum Control

In this paper, we propose the challenge of composing a target unitary rather than performing a backward search starting from the target state since the decomposition of prepared states or already given unitary operations constitutes a well-known challenge [44]–[46]. Composing such a valid unitary circuit for specific input states and all possible inputs requires high-precision operations. Since the tasks considered in this work (state preparation and unitary approximation) are commonly considered in quantum control, we also briefly highlight two related approaches that might constitute suitable baseline candidates but fall outside the scope of our proposed RL structure.

The GRAPE (Gradient Ascent Pulse Engineering) algorithm is used in quantum control to find optimal control pulses. The essence of GRAPE is to maximize a target function that represents the fidelity of the quantum operation being controlled relative to a desired quantum operation. It does this by iteratively adjusting the control pulses based on the gradient of the fidelity concerning the controls. This approach helps fine-tune quantum operations to achieve higher precision and efficiency in quantum computing systems.[47]

The CRAB (Chopped Random Basis Quantum Optimization) algorithm is designed for quantum optimal control. It works by expressing the control fields as expansions on a randomly chosen basis, which simplifies the problem from optimizing a functional form to finding optimal parameters on this basis. The algorithm iteratively adjusts these parameters to maximize the fidelity of the quantum operation. This method is particularly efficient because it reduces the complexity of the control problem, allowing for faster and more efficient optimization in quantum systems. You can read more about it in the full article here. [48]

C. Quantum Circuit Optimization

In addition to searching for feasible circuit architectures, RL has also been considered for *quantum circuit optimization* (QCO). The most apparent challenge in QCO is optimizing the circuit structure to reduce its depth and global gate count. The goal is to find a logically equivalent circuit using fewer gates from a given quantum circuit, thus reducing the runtime. In addition to approaches such as the use of ZX calculus [49], employing RL has also been proposed, either, with the agent operating on the entire circuit [50], or, incorporating optimization through a gate-by-gate procedure [37]. Ostaszewski, Trenkwalder, Masarczyk, *et al.* [37] even argue that RL will naturally find shorter and better-performing solutions, as it is designed to maximize the discounted sum of rewards. In practice, we propose applying a step cost for each executed operation. Therefore, the structural optimization of the circuits is inherent in the design of our approach.

To achieve practical effectiveness, quantum circuits must also be designed to accommodate the constraints of existing NISQ hardware. Beyond the circuit depth, these constraints include the limited number and connectivity of qubits, the available gate set, and the high susceptibility to errors [27]. The qubit routing problem describes the challenge of mapping logical to physical qubits [51], which might not be connected due to experimental limitations. SWAP operations are applied to connect pairs, allowing arbitrarily placing two-qubit gates, such as CNOT. However, taking the resulting circuit depth into account, this constitutes an NP-hard combinatorial optimization problem [52]. Besides popular approaches such as sorting networks [53], or proprietary compiler-specific methods, the application of RL operating on the entire circuit has been shown to outperform state-of-the-art methods [54], [55]. Regarding the design of high-fidelity error-tolerant gate sets, compensating for the processes that induce hardware errors on NISQ devices, Baum, Amico, Howell, *et al.* [56] proposed using RL to manipulate a small set of accessible experimental control signals. Considering the qubit routing problem and error mitigation, our approach does not yet consider these hardware-specific properties, but it can be easily extended to do so. Concerning reduced hardware connectivity, the action space for multi-qubit gates could be reduced to qubits that are directly connected to each other in the hardware topology. Regarding Error Mitigation, the state of the RL agent could be changed from state vectors to density matrices to model errors occurring, e.g., due to decoherence or imperfect gates.

Overall, those applications further underscore the efficacy of RL in quantum computing. Yet, most approaches consider RL for specific applications, using specialized discrete gate sets, where the parameters need to be optimized post hoc.

VI. Implementation

The QCD environment is implemented based on *gymnasium* [57] and *qiskit* [58]. All implementations are available here [1].

A. Environments

As previously elaborated, state preparation (SP) and unitary composition (UC) objectives are either closely related to the others mathematically, accounted for by the environment design, or can be easily incorporated. As defined in section III, we used states $\Psi \in \{|\Phi^+\rangle, |\tilde{U}\rangle, |GHZ\rangle\}$, with $\eta \in \{2, 2, 3\}$ and $\delta \in \{12, 12, 15\}$ respectively, for SP, and unitaries $U \in \{\boldsymbol{H}, \tilde{\boldsymbol{U}}, \boldsymbol{CCX}\}$, with $\eta \in \{1, 2, 3\}$ and $\delta \in \{9, 12, 63\}$ respectively, for UC. As a proof-of-concept for the proposed environment architecture, we only conducted experiments for this initial set of targets. However, note that the QCD allows for easy extension of both objectives by defining arbitrary target states and unitaries.

B. Action Space

Regarding the implementation of the action space, the main challenge we face is the mixed nature of quantum control: While selecting gates and qubits to be applied resembles a discrete decision problem, we argue that integrating continuous control over the parameterization of the placed gate is crucial to unveiling RL's full potential for QC. As current RL algorithms do not implement policies with mixed output spaces, we opt for a fully continuous action space and discretize the choice over the gate, qubit, and control contained. Yet, our long-term vision is to enable mixed control to suit quantum control optimally.

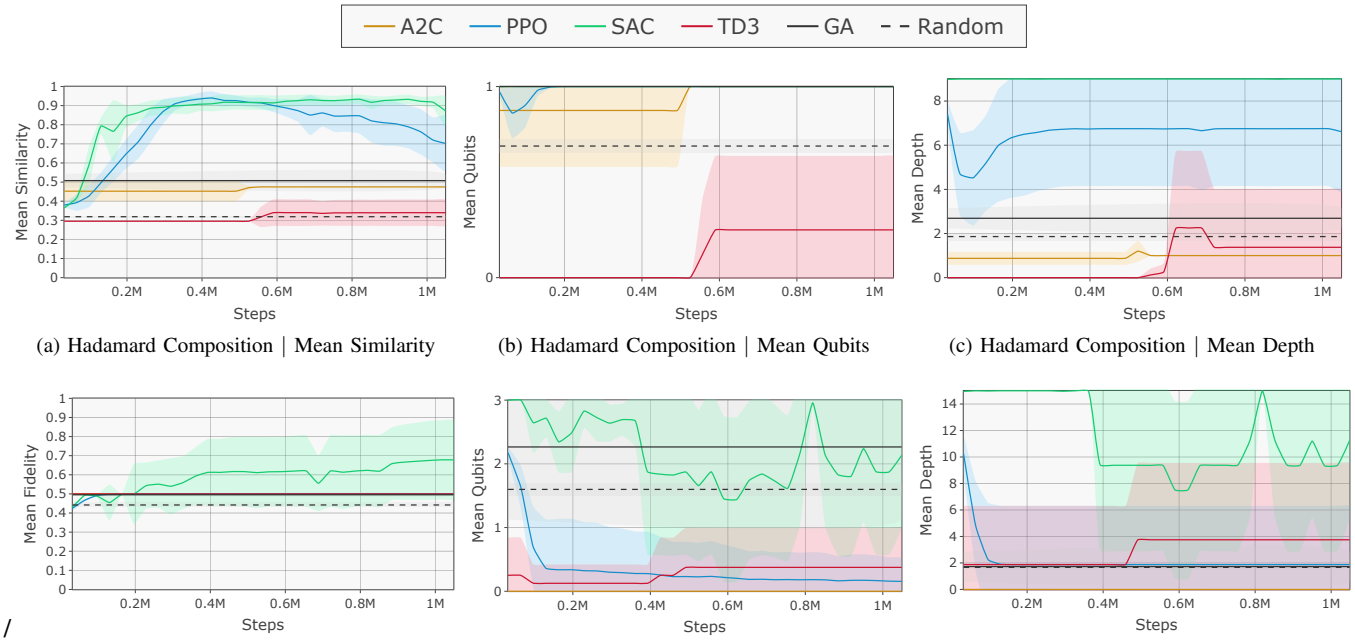
C. Baselines

To provide a diverse set of state-of-the-art model-free baselines, we decided for A2C [12], PPO [13], TD3 [15], and SAC [16]. A discretization of the proposed action space (especially the parameterization) would require $3 \cdot \eta^2 \cdot \varrho$ dimensions, with a resolution $\varrho = 4$, resulting in more than 100 dimensions of actions only for a 3-qubit circuit. We, therefore, omitted DQN baselines, which only support discrete action spaces. All baselines are implemented extending Raffin, Hill, Gleave, *et al.* [59] and use the default hyperparameters suggested in their corresponding publication. All policies are trained for 1M time steps. Additionally, we provide a *Random* baseline averaged over 100 episodes of randomly chosen actions. To reflect alternative state-of-the-art approaches for automated quantum circuit design, we furthermore provide results from a genetic algorithm (*GA*). The approach is implemented according to Sünkel, Martyniuk, Mattern, *et al.* [35], with 20 individuals optimized for 50 epochs, using the cost-adjusted objectives previously defined to evaluate their fitness. Furthermore, note that we chose the maximum depth δ for all non-random tasks such that an optimal handcrafted solution would require only a third of the available depth using all available qubits, which indirectly serves as an additional human baseline. For significance, all results are averaged over eight random seeds, reporting both the mean and the 95% confidence interval.

D. Metrics

As a performance metric, we report the pure objective rewards for the final circuits (without applying the operation cost): The *Mean Fidelity* according to Eq. (12) for state preparation tasks and the *Mean Similarity* according to Eq. (17) for unitary composition tasks. To reflect capabilities regarding circuit optimization or the generation of optimized circuits, we furthermore provide the final number of utilized qubits (*Mean Qubits*) and the final depth of the circuit (*Mean Depth*). Note that the final depth of the circuit is affected by the scheduling of the sequence of operations generated by the policy, which is currently handled by the simulation. However, scheduling tasks have also been shown to be solvable RL objectives and could be integrated into the reward function to reduce the need for external compilation. Note that, regardless of defining constraints for the number of qubits η and the depth δ beforehand, the choice on the number of operations is made by the policy. For smoothing, all metrics are averaged over a sliding window of 100 episodes (or individuals for the GA).

[1] https://github.com/philippaltmann/QCD



(a) Hadamard Composition | Mean Similarity (b) Hadamard Composition | Mean Qubits (c) Hadamard Composition | Mean Depth

(d) GHZ State Preparation | Mean Fidelity (e) GHZ State Preparation | Mean Qubits (f) GHZ State Preparation | Mean Depth

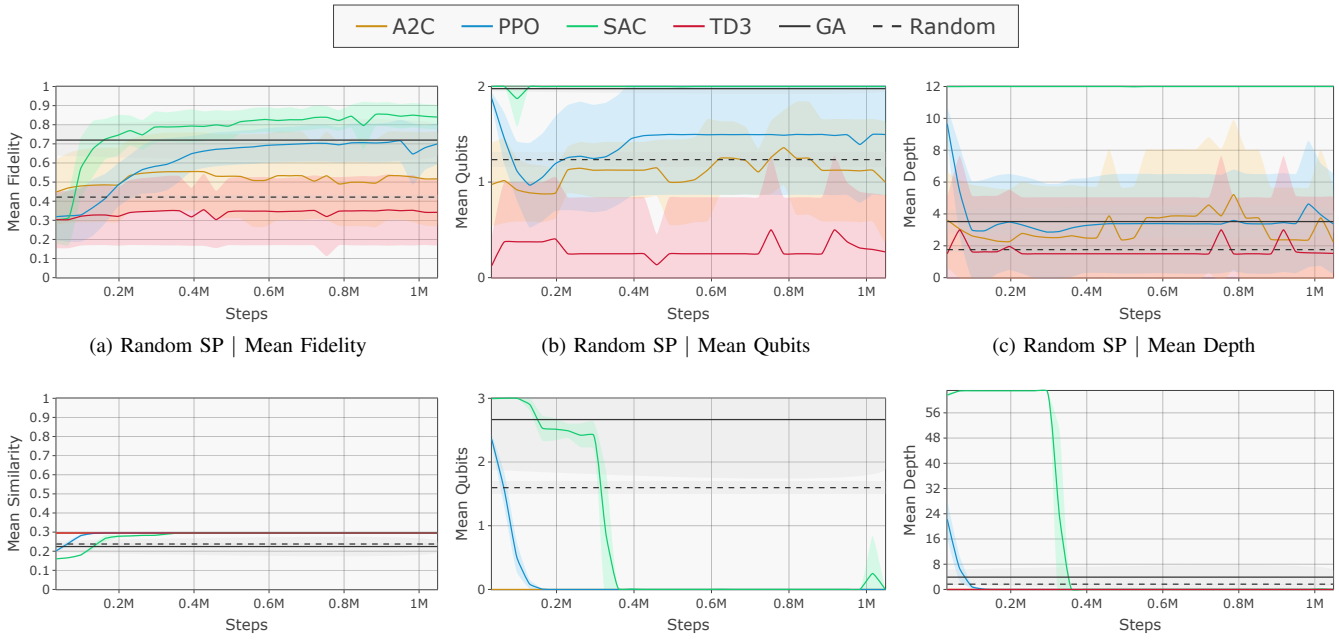
Fig. 2. **Quantum Circuit Design Evaluation:** Benchmarking A2C (orange), PPO (blue), SAC (green), and TD3 (red) for Hadamard Composition (Fig. 2a-2c) and GHZ State Preparation (Fig. 2e-2f) with regards to the Mean Metric (Fidelity and Similarity, higher is better), Mean Qubits utilized, and Mean Depth of the resulting circuit, against a GA (gray) and a Random baseline (dashed line). Shaded areas mark the 95% confidence intervals. Overall, SAC shows the highest objective performance with the highest qubit and depth utilization (which could be further improved towards the optimal 1/3 operation utilization).

VII. Evaluation

In the following, we aim to demonstrate the challenges of current RL approaches inherent in QCD. First, we look at two elementary objectives: Constructing the Hadamard operation and preparing the 3-qubit GHZ state. As introduced earlier, the Hadamard operator is an integral component for exploiting superposition in QC and is, therefore, included in most elementary gate sets. However, as we focus on the choice of parameterized gates, it is precluded from our action space. Playing a central role in most quantum algorithms nevertheless, it is crucial for RL to reconstruct it. Reconstruction is possible via $H = P(\pi/2)RX(\pi/2)P(\pi/2)$. Connecting the Hadamard state to another or multiple qubits via one or multiple CNOTs yields the Bell and GHZ states, respectively. Both states likewise play a vital role in QC algorithms, expressing maximum entanglement of the involved qubits. Given their close resemblance, please refer to the Appendix for further results for preparing the Bell state.

Fig. 2 shows the evaluation results. Regarding the Mean Similarity for the Hadamard composition, all compared algorithms exceed the random baseline. Yet, presumably due to the intricate action space, A2C and TD3 stagnate at a performance below the GA at 0.5. PPO and SAC, on the other hand, exceed this local optimum for nearly empty circuits and reach an above-90% resemblance to the target Hadamard unitary within 400k time steps. Looking at the qubit and depth utilization reveals that both approaches converge to steadily using the available qubit. While PPO converges to a mean depth slightly above six, SAC utilizes the full available depth, which indicates further potential for optimizing toward the secondary compactness objective. Overall, because we intend to build a single framework for designing optimized circuits, QCD is prone to getting stuck in local minima, which is introduced by its nature in multi-objective optimization.

Given that the subsequent task of creating entanglement builds upon the skill of putting qubits into superposition, only SAC and PPO can be expected to succeed in this task. This constitutes a further challenge that QC yields for RL: Complex circuit architectures are often built upon hierarchical building blocks. This characteristic justifies multi-level optimization approaches resembling similar hierarchies. On the other hand, the proposed low level might yield the exploration of unconventional approaches providing out-of-the-box solutions to complex multi-level challenges. Indications of said prospects can be observed for the GHZ state preparation results, where SAC reaches a mean fidelity of around 0.7, again significantly outperforming the compared approaches and baselines at a constant performance of 0.5. For the 2-qubit Bell state (cf., Fig. 4), PPO also manages to evade this local optimum, while SAC shows near-optimal performance. However, we look at a different objective of state preparation here, which might be easier, as no exact resemblance of the operations is required. Looking at the mean qubits and depth utilized for the GHZ state preparation, SAC, in contrast to the previous Hadamard composition, explores lower operation utilization while exceeding the local optimum for nearly empty circuits (which yields a fidelity of 0.5). This indicates that, while posing a significant challenge, current RL algorithms are capable of



(a) Random SP | Mean Fidelity (b) Random SP | Mean Qubits (c) Random SP | Mean Depth

(d) Toffoli Composition | Mean Similarity (e) Toffoli Composition | Mean Qubits (f) Toffoli Composition | Mean Depth

Fig. 3. **Quantum Circuit Design Evaluation:** Benchmarking A2C (orange), PPO (blue), SAC (green), and TD3 (red) for Random State Preparation (Fig. 3a-3c) and Toffoli Composition (Fig. 3d-3f) with regards to the Mean Metric (Fidelity and Similarity, higher is better), Mean Qubits utilized, and Mean Depth of the resulting circuit, against a GA (gray) and a Random baseline (dashed line). Shaded areas mark the 95% confidence intervals. While random states are prepared predominantly well, utilizing all qubits and reasonable depths, the Toffoli Composition exhibits a local similarity optimum of 0.3 for empty circuits.

exploring hierarchical structures required for building more complex circuit architectures while optimizing for a secondary compactness objective. Nevertheless, these multi-objective and sparse reward landscapes require explorative capabilities to overcome inherent local minima.

To benchmark more advanced scenarios, Fig. 3 shows the evaluation results for random state preparation and Toffoli composition. The ability to quickly put qubits into an arbitrary state is crucial to the success of various approaches, especially in QML using variational quantum circuits. Please refer to Fig. 4 for a benchmark of resembling arbitrary unitary compositions. The Toffoli unitary reflects an operation that cannot be directly implemented on most hardware and, therefore, needs to be automatically decomposed.

For random state preparation, SAC again outperforms all compared baselines with a final mean fidelity of 0.85. PPO performs comparably to the GA. Notably, all approaches except for TD3 show the capability of learning the objective at hand, even though preparing random states is not considered trivial, which validates our chosen environment design. Furthermore, the additional compactness objective is shown to be effective, considering that the mean depths predominantly cover only half of the available operations. The better-performing approaches primarily utilize both available qubits.

Yet, the exploration challenge inherent in the specific actions required to attain more complex targets again is more notable for the Toffoli composition, offering depths up to 63 across three qubits. All RL approaches converge to a locally optimal solution with a mean similarity of 0.3, while both baselines show lower performance around 0.2. Given that an optimal (handcrafted) composition requires a total of 24 concise operations across all three qubits, this unsuccessful behavior is not surprising. Looking at both the mean number of qubits and the mean circuit depth reveals primal convergence to empty circuits, indicated by a low depth and low qubit utilization. Thus, RL exploits a deficiency in the reward landscape, revealing a local optimum for empty circuits. Due to the additional step cost, overcoming this local optimum would first require a performance decrease caused by the use of additional operations. On the other hand, the genetic and random baselines produce small yet non-empty circuits that perform even worse regarding their mean similarity to the target unitary. As expected, random unitary compositing poses a similarly difficult objective, where the best-performing approaches also merely exceed the random baseline. These results emphasize the need for more directive reward signals to guide policy optimization in such complex, potentially hierarchical tasks with sparse solution landscapes.

Overall, the presented experimental results provide certainty that the proposed `qcd-gym` environment can be used to learn skills in the domain of QC using RL. Yet, current state-of-the-art approaches do not provide sufficient performance for robust and scaleable applicability. Besides the proposed objectives and their corresponding metrics, the circuit depth and the number of used qubits have proven to be viable metrics for an in-depth analysis of the learned behavior. Across all challenges, SAC has yielded superior results due to its advanced continuous control capabilities.

VIII. Conclusion

We introduced `qcd-gym`, a framework for benchmarking RL for use in low-level quantum control. Furthermore, we motivated and proposed an initial set of objectives for both quantum architecture search and quantum circuit optimization, including the composition of unitary operations and the preparation of quantum states. Benchmark results for various state-of-the-art model-free RL algorithms and comparisons to a random and an evolutionary optimization baseline showed RL's competitiveness for QCD while ensuring the soundness of the proposed framework. Those empirical results also highlighted current challenges to be overcome for RL to be widely applicable for quantum control.

Those challenges mainly include exploring multi-modal reward landscapes in combination with complex, non-uniform, high-dimensional action spaces and sparse, extinct target requirements. While RL research has proposed various approaches to tackle these challenges [60]–[63], we believe addressing those quantum-control-specific demands would advance both fields.

Considering the additional objectives, minimizing operation count and circuit depth while maximizing performance separately and applying multi-objective RL concepts might improve the ability to deal with those independent goals. Furthermore, smoother reward metrics, including intermediate rewards, are needed to ease exploring the intricate action space for quantum control. Those could include more hardware-friendly similarity metrics, also considering specific connectivities or further constraints. Also, the proposed state space is currently limited by its scaling exponentially with the number of qubits. For an agent to focus on relevant parts of the state space, an attention mechanism or partial observability might be helpful to further decrease the exploration challenge [64]. In addition, simulations are currently only possible for qubit numbers of $\eta < 50$. Thus, approximate simulations might be helpful to increase viable circuit sizes. Future work should also follow up with collecting additional relevant states, unitaries, and Hamiltonians for SP, UC, and HS. Also, as previously mentioned, the overall objective might be extended to account for error mitigation. Furthermore, the action space might be extended to allow for lower-level quantum control, enabling a deeper hardware integration.

Overall, however, we believe that `qcd-gym` provides a profound base for future research on the application of RL to QC to build upon.

Acknowledgements

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

References

[1] P. Altmann, A. Bärligea, J. Stein, *et al.*, "Quantum circuit design: A reinforcement learning challenge," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 2024, pp. 2123–2125.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[3] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, 2009.

[4] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[5] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[6] A. Barenco, C. H. Bennett, R. Cleve, *et al.*, "Elementary gates for quantum computation," *Physical review A*, vol. 52, no. 5, 1995.

[7] A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, 1997.

[8] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[9] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] V. Mnih, A. P. Badia, M. Mirza, *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, PMLR, 2016, pp. 1928–1937.

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

[14] J. Grudzien, C. A. S. De Witt, and J. Foerster, "Mirror learning: A unifying framework of policy optimisation," in *International Conference on Machine Learning*, PMLR, 2022, pp. 7825–7844.

[15] S. Dankwa and W. Zheng, "Twin-delayed ddpg: A deep reinforcement learning technique to model a continuous movement of an intelligent robot agent," in *Proceedings of the 3rd international conference on vision, image and signal processing*, 2019, pp. 1–5.

[16] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[17] M. Bukov, A. G. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, "Reinforcement learning in different phases of quantum control," *Physical Review X*, vol. 8, no. 3, 2018.

[18] L. Colomer, M. Skotiniotis, and R. Muñoz-Tapia, "Reinforcement learning for optimal error correction of toric codes," *Physics Letters A*, vol. 384, no. 17, 2020.

[19] S. van der Linde, W. de Kok, T. Bontekoe, and S. Feld, "Qgym: A gym for training and benchmarking rl-based quantum compilation," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 02, 2023, pp. 26–30. DOI: 10.1109/QCE57702.2023.10179.

[20] C. Dawson and M. Nielsen, "The solovay-kitaev algorithm," *Quantum Information and Computation*, vol. 6, no. 1, pp. 81–95, Jan. 2006. DOI: 10.26421/qic6.1-6.

[21] M. Ozols, "How to generate a random unitary matrix," *unpublished essay on http://home. lu. lv/sd20008*, 2009.

[22] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, 1996.

[23] A. Bolens and M. Heyl, "Reinforcement learning for digital quantum simulation," *Physical Review Letters*, vol. 127, no. 11, 2021.

[24] R. de Lazcano, K. Andreas, J. J. Tai, S. R. Lee, and J. Terry, *Gymnasium robotics*, version 1.2.4, 2023. [Online]. Available: http://github.com/Farama-Foundation/Gymnasium-Robotics.

[25] A. Peruzzo, J. McClean, P. Shadbolt, *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, no. 1, Jul. 2014.

[26] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[27] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, *et al.*, "Noisy intermediate-scale quantum algorithms," *Reviews of Modern Physics*, vol. 94, no. 1, 2022.

[28] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[29] P. Selig, N. Murphy, D. Redmond, and S. Caton, "Deepqprep: Neural network augmented search for quantum state preparation," *IEEE Access*, vol. 11, pp. 76 388–76 402, 2023. DOI: 10.1109/ACCESS.2023.3296802.

[30] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, "Differentiable quantum architecture search," *Quantum Science and Technology*, vol. 7, no. 4, p. 045 023, 2022.

[31] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[32] A. Giovagnoli, V. Tresp, Y. Ma, and M. Schubert, "Qneat: Natural evolution of variational quantum circuit architecture," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023, pp. 647–650.

[33] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, "A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver," *arXiv preprint arXiv:1910.09694*, 2019.

[34] D. Chivilikhin, A. Samarin, V. Ulyantsev, I. Iorsh, A. Oganov, and O. Kyriienko, "Mog-vqe: Multiobjective genetic variational quantum eigensolver," *arXiv preprint arXiv:2007.04424*, 2020.

[35] L. Sünkel, D. Martyniuk, D. Mattern, J. Jung, and A. Paschke, "Ga4qco: Genetic algorithm for quantum circuit optimization," *arXiv preprint arXiv:2302.01303*, 2023.

[36] S. Ding, Z. Jin, and Q. Yang, "Evolving quantum circuits at the gate level with a hybrid quantum-inspired evolutionary algorithm," *Soft Computing*, vol. 12, pp. 1059–1072, 2008.

[37] M. Ostaszewski, L. M. Trenkwalder, W. Masarczyk, E. Scerri, and V. Dunjko, "Reinforcement learning for optimization of variational quantum circuit architectures," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 182–18 194, 2021.

[38] M. S. Mortazavi, T. Qin, and N. Yan, "Theta-resonance: A single-step reinforcement learning method for design space exploration," *arXiv preprint arXiv:2211.02052*, 2022.

[39] E.-J. Kuo, Y.-L. L. Fang, and S. Y.-C. Chen, "Quantum architecture search via deep reinforcement learning," *arXiv preprint arXiv:2104.07715*, 2021.

[40] T. Sogabe, T. Kimura, C.-C. Chen, *et al.*, "Model-free deep recurrent q-network reinforcement learning for quantum circuit architectures design," *Quantum Reports*, vol. 4, no. 4, pp. 380–389, 2022.

[41] T. Gabor, M. Zorn, and C. Linnhoff-Popien, "The applicability of reinforcement learning for the automatic generation of state preparation circuits," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022, pp. 2196–2204.

[42] M. Kölle, T. Schubert, P. Altmann, M. Zorn, J. Stein, and C. Linnhoff-Popien, "A reinforcement learning environment for directed quantum circuit synthesis," *arXiv preprint arXiv:2401.07054*, 2024.

[43] G. E. Crooks, "Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition," *arXiv preprint arXiv:1905.13311*, 2019.

[44] Y.-H. Zhang, P.-L. Zheng, Y. Zhang, and D.-L. Deng, "Topological quantum compiling with reinforcement learning," *Physical Review Letters*, vol. 125, no. 17, 2020.

[45] M. Pirhooshyaran and T. Terlaky, "Quantum circuit design search," *Quantum Machine Intelligence*, vol. 3, pp. 1–14, 2021.

[46] M. B. Mansky, S. L. Castillo, V. R. Puigvert, and C. Linnhoff-Popien, "Near-optimal quantum circuit construction via cartan decomposition," *Physical Review A*, vol. 108, no. 5, p. 052 607, 2023.

[47] P. de Fouquieres, S. G. Schirmer, S. J. Glaser, and I. Kuprov, "Second order gradient ascent pulse engineering," *Journal of Magnetic Resonance*, vol. 212, no. 2, pp. 412–417, 2011.

[48] T. Caneva, T. Calarco, and S. Montangero, "Chopped random-basis quantum optimization," *Physical Review A*, vol. 84, no. 2, p. 022 326, 2011.

[49] A. Kissinger and J. van de Wetering, "Reducing the number of non-clifford gates in quantum circuits," *Physical Review A*, vol. 102, no. 2, Aug. 2020.

[50] T. Fösel, M. Y. Niu, F. Marquardt, and L. Li, "Quantum circuit optimization with deep reinforcement learning," *arXiv preprint arXiv:2103.07585*, 2021.

[51] A. M. Childs, E. Schoute, and C. M. Unsal, "Circuit transformations for quantum architectures," in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2019.

[52] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, "Qubit allocation," in *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, 2018, pp. 113–125.

[53] D. S. Steiger, T. Häner, and M. Troyer, "Projectq: An open source software framework for quantum computing," *Quantum*, vol. 2, p. 49, 2018.

[54] S. Herbert and A. Sengupta, *Using reinforcement learning to find efficient qubit routing policies for deployment in near-term quantum computers*, 2018.

[55] M. G. Pozzi, S. J. Herbert, A. Sengupta, and R. D. Mullins, "Using reinforcement learning to perform qubit routing in quantum compilers," *ACM Transactions on Quantum Computing*, vol. 3, no. 2, pp. 1–25, 2022.

[56] Y. Baum, M. Amico, S. Howell, *et al.*, "Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer," *PRX Quantum*, vol. 2, no. 4, 2021.

[57] M. Towers, J. K. Terry, A. Kwiatkowski, *et al.*, "Gymnasium," 2023. [Online]. Available: https://github.com/Farama-Foundation/Gymnasium.

[58] Qiskit contributors, *Qiskit: An open-source framework for quantum computing*, 2023. DOI: 10.5281/zenodo.2573505.

[59] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 12 348–12 355, 2021.

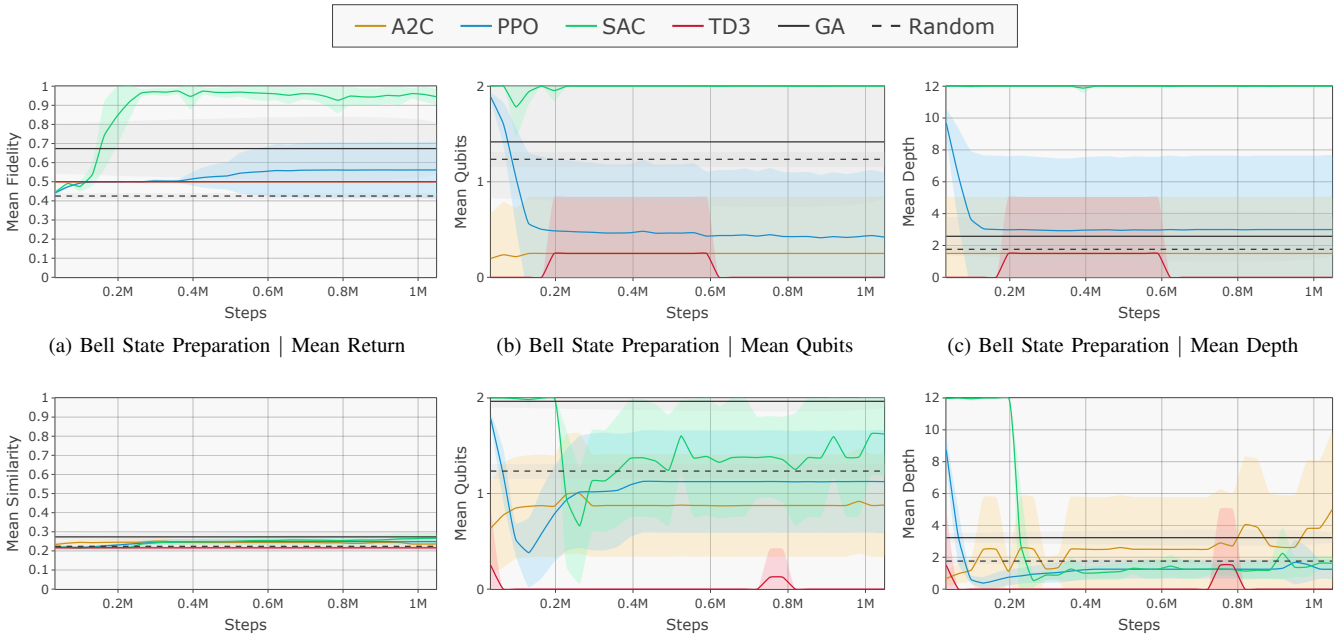
[60] C. F. Hayes, R. Rădulescu, E. Bargiacchi, *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, p. 26, 2022.

[61] C. Kim, Y. Seo, H. Liu, *et al.*, "Guide your agent with adaptive multimodal rewards," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[62] B. Li, H. Tang, Y. ZHENG, *et al.*, "HyAR: Addressing discrete-continuous action reinforcement learning via hybrid action representation," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=64trBbOhdGU.

[63] R. Devidze, P. Kamalaruban, and A. Singla, "Exploration-guided reward shaping for reinforcement learning under sparse rewards," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5829–5842, 2022.

[64] P. Altmann, F. Ritz, L. Feuchtinger, J. Nüßlein, C. Linnhoff-Popien, and T. Phan, "Crop: Towards distributional-shift robust reinforcement learning using compact reshaped observation processing," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, E. Elkind, Ed., Main Track, International Joint Conferences on Artificial Intelligence Organization, Aug. 2023, pp. 3414–3422. DOI: 10.24963/ijcai.2023/380.



(a) Bell State Preparation | Mean Return (b) Bell State Preparation | Mean Qubits (c) Bell State Preparation | Mean Depth

(d) Random Composition | Mean Return (e) Random Composition | Mean Qubits (f) Random Composition | Mean Depth

Fig. 4. **Additional Benchmark Results:** Benchmarking A2C (orange), PPO (blue), SAC (green), and TD3 (red) for Bell State Preparation (Fig. 4a-4c) and Random Composition (Fig. 4d-4f) with regards to the Mean Metric (Fidelity and Similarity, higher is better), Mean Qubits utilized, and Mean Depth of the resulting circuit, against a GA (gray) and a Random baseline (dashed line). Shaded areas mark the 95% confidence intervals. While the Bell state is almost optimally prepared using SAC, composing random unitary operations yields serious challenges that hinder optimal convergence.