Hybrid Reward-Driven Reinforcement Learning for Efficient Quantum Circuit Synthesis

Sara Giordano[1][0000-0002-0071-0278], Kornikar Sen[1][0000-0002-7007-0843], and Miguel A. Martin-Delgado[1,2][0000-0003-2746-5062]

[1] Universidad Complutense de Madrid, Madrid, 28040, Madrid, Spain
[2] CCS-Center for Computational Simulation, Universidad Politecnica de Madrid (UPM), Boadilla del Monte, 28660, Madrid, Spain

Abstract. A reinforcement learning framework based on tabular Q-learning is introduced for synthesizing quantum circuits that prepare target quantum states from a fixed initial state. The approach operates on a discretized quantum state space and employs a hybrid reward mechanism: a static, precomputed reward guides the agent toward the target, while customizable dynamic penalties discourage inefficient structures such as gate congestion and redundant state revisits. Sparse matrix representations keep the method scalable to high-dimensional environments. Benchmarked on graph-state preparation for up to seven qubits, the algorithm consistently finds minimal-depth or near-optimal circuits with optimized gate counts, and the framework extends to a universal gate set while preserving these properties.

Keywords: quantum circuits · reinforcement learning · optimization

1 Introduction and Motivation

Designing optimized quantum circuits is a central challenge in the NISQ era and for future fault-tolerant quantum computing. Reinforcement Learning (RL) has recently emerged as a powerful approach, with the ability to discover optimized circuits [1, 6]. The work in [2] addresses a state-preparation problem: given a fixed initial state $|\psi_0\rangle$, synthesize a quantum circuit that produces a specified target state $|\psi_{\text{target}}\rangle$. It combines tabular Q-learning [9] over a discretized quantum state space with a hybrid reward mechanism that simultaneously guides the agent toward the target and penalizes inefficient circuit structures, through reward-shaping principles and multi-objective RL design. Sparse matrix representations keep the method tractable up to seven qubits despite the exponentially large state-action space.

2 SWEET States: State Space Discretization

Because the full n-qubit Hilbert space $\mathcal{H}_n$ is infinite and continuous, the paper restricts the learning environment to a finite, structured subset called States

With Equal-amplitude and Encoded-phase Terms (SWEET):

$$|\psi_n\rangle = \frac{1}{\sqrt{N'}} \sum_{j=1}^{N} \alpha_j |x_j\rangle, \quad \alpha_j \in \mathcal{P}_M := \{e^{i2m\pi/M}\}_{m\in\mathbb{N}}, \tag{1}$$

where $|x_j\rangle$ are computational basis states and $M = 2^p$ controls the phase resolution. The p phase qubits are introduced only for simulation purposes; no extra physical qubits are required in the circuit. This representation is particularly well suited to graph states [7, 8], GHZ states and other uniform-amplitude entangled states. The total number of SWEET states over an n-qubit, p-phase-qubit system is $N_S = 2^{2^{n+p}} - 1$, which is doubly exponential in the number of qubits (e.g., $N_S = \mathcal{O}(10^{77})$ for $n = 7$, $p = 1$).

3 The Q-Learning Framework

States, Actions, and Sparse Storage. The paper considers the universal gate set $\mathcal{U} = \{H, \text{CNOT}, T, T^\dagger\}$. By the Gottesman–Knill theorem [3], circuits built solely from Clifford gates (H, CNOT, CZ, Pauli gates) can be efficiently simulated classically; the non-Clifford T gate is therefore necessary to achieve genuine quantum advantage. Circuit quality is assessed gate count and circuit depth, this latter being the number of time steps required. Agent states correspond to SWEET quantum states (indexed by $s \in \{1, \ldots, N_S\}$) and agent actions correspond to valid applications of gates from the chosen gate set over the n qubits. The total action count is $N_A = \sum_k g_k^n \cdot \frac{n!}{(n-k)!}$, where g_k^n is the number of distinct gate types acting on exactly k qubits. Both the reward matrix R and the Q-matrix Q (of nominal dimension $N_S \times N_A$) are stored as SQL sparse triplets $\{\texttt{state}, \texttt{action}, \texttt{value}\}$, retaining only non-zero entries and significantly reducing memory overhead.

Hybrid Reward Design. The reward function decomposes into a static and a dynamic part:

$$R(s, a) = R_{\text{sta}}(s, a) + R_{\text{dyn}}(s, a). \tag{2}$$

Static reward (offline). A layered reward is precomputed by propagating backwards from the target state using the reversibility of quantum gates:

$$R_{\text{sta}}(s, a) = \begin{cases} R_{\max} & \text{if } s \xrightarrow{a} s_{\text{target}}, \\ R_{\max}/2^k & \text{if } s \text{ is } k \text{ steps from target}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

With $R_{\max} = 10^4$, each additional stratum halves the reward, creating a gradient that helps converging.

Dynamic penalties (online). Computed during training, these penalize: (i) revisiting a state already seen in the current episode, (ii) taking an action that leaves the state unchanged, (iii) congestion, i.e. a penalty that discourages qubit reuse in successive gates, promoting parallelism and reducing depth. Individual

penalty magnitudes are set to $\sim 0.01\%$–0.1% of $R_{\max}$ so they guide without overwhelming the primary reward signal. The congestion level $\mathrm{C}(a_t)$ for action a_t acting on qubit set $\mathcal{Q}_{a_t}$ is $\mathrm{C}(a_t) = \max_{q_i \in \mathcal{Q}_{a_t}} c_i$, where c_i counts how many prior actions in the episode involved qubit q_i. A penalty is applied whenever $\mathrm{C}(a_t) > \mathrm{CT}_t$ with threshold $\mathrm{CT}_t = t/2$.

Training and Testing. Training uses ϵ-greedy Q-learning with the standard temporal-difference update [9]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Big[R(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \Big]. \tag{4}$$

Episodes of fixed length k start from the same initial state $|\psi_0\rangle$; training continues in batches until the testing phase succeeds. In the testing phase, actions are selected greedily ($\arg\max_a Q(s_t, a)$) until $|\psi_{\text{target}}\rangle$ is reached or a length cap l_c is hit. Unlike deep RL approaches, the method is strictly tabular and off-policy, enabling interpretable convergence diagnostics. Convergence is verified by monitoring the average Bellman error, which falls below 1% of $R_{\max}$ after successful training.

4 Results

Graph-State Benchmarking. Graph states [7, 4] are defined as $|G_n\rangle = \prod_{\{v_i, v_j\} \in \mathcal{E}} \mathrm{CZ}(v_i, v_j) \, |+\rangle^{\otimes n}$, for a graph $G = (\mathcal{V}, \mathcal{E})$ and constitute an ideal benchmark because their minimal circuit depth is known analytically via Vizing's theorem [5]: for bipartite graphs, $\delta = \Delta(G)$, the maximum vertex degree. Two target graph states were considered: a 4-qubit square graph state $|G_4\rangle$, requiring 4 CZ gates and optimal depth 2, and a 7-qubit bipartite graph state $|G_7\rangle$, requiring 10 CZ gates and optimal depth 4. The RL agent matched the theoretical optimum in both gate count and depth, with training requiring 10^4 episodes for $|G_4\rangle$ and 7×10^4 for $|G_7\rangle$, at episode length $k = 50$. For $n = 7$, $p = 1$, the total sparse Q- and R-matrix storage fits within 170 MB, and preliminary results for $n = 8$ confirm continued tractability.

Universal Gate Set: Three-Qubit State Preparation. The framework was also applied to states beyond graph states, with the adequate parameters in order to correctly represent the phases, up to three qubits states. Also in this case the congestion penalty helped to reduce the circuit depth at the cost of rising the gate count. The set of gates used is $\mathcal{U} = \{H, \mathrm{CNOT}, T, T^\dagger\}$. A subtlety arises in this second part because some gates combination can map SWEET states to non-SWEET states, i.e. states with unequal amplitudes that fall outside the discrete set $\mathcal{S}_n$. Since the algorithm can only register and label SWEET states, it treats each SWEET state as the representative of a broader class of states sharing the same computational-basis support and phase pattern. Any non-SWEET state encountered during training is mapped onto the nearest representative SWEET state. Nevertheless, in the studied case, the fidelity between the targeted state and the one obtained is ≈ 0.971, confirming that the output closely approximates the intended target.

5 Conclusion

The work in [2] demonstrates that a tabular Q-learning agent equipped with a hybrid reward can efficiently synthesize near-optimal quantum circuits in exponentially large and sparse environments through a tractable discretization of the state space (SWEET states) and sparse Q-table storage. Dynamic penalties and static reward steer the agent toward specific circuit properties. On graph-states, the algorithm discovers optimal depth circuits for up to seven qubits and it can be adapted to a universal gate set too. Future directions include extending the framework to deep learning and unitary synthesis and incorporating hardware-specific constraints.

Acknowledgments. This work was supported by Spanish MICIN grant PID2021-122547NB-I00, CAM Programa TEC-2024/COM-84 QUITEMAD-CM, and the MADQuantum-CM project funded by Comunidad de Madrid (Programa de Acciones Complementarias), the Spanish Ministry for Digital Transformation and Civil Service (Quantum Spain/QUANTUM ENIA), and the European Union's NextGeneration EU plan (PRTR-C17.I1 / Digital Spain 2026). K.S. and S.G. acknowledge support from MADQuantum-CM. M.A.M.-D. acknowledges partial support from the U.S. Army Research Office under Grant No. W911NF-14-1-0103.

Disclosure of Interests. The authors declare no competing interests.

References

1. Giordano, S., Martin-Delgado, M.: Reinforcement-learning generation of four-qubit entangled states. Phys. Rev. Res. **4**, 043056 (2022)
2. Giordano, S., Sen, K., Martin-Delgado, M.A.: Hybrid reward-driven reinforcement learning for efficient quantum circuit synthesis. Quantum Machine Intelligence **8**(1) (2026)
3. Gottesman, D.: The heisenberg representation of quantum computers. pp. 32–43. International Press, Cambridge, MA (1999)
4. Hein, M., Dür, W.e.a.: Entanglement in graph states and its applications. In: Casati, G., Montangero, S. (eds.) Proceedings of the International School of Physics "Enrico Fermi": Quantum Computers, Algorithms and Chaos. IOS Press, Varenna, Italy (2006)
5. Misra, J., Gries, D.: A constructive proof of vizing's theorem. Information Processing Letters **41**(3), 131–133 (1992)
6. Moflic, I., Paler, A.: Towards faster reinforcement learning of quantum circuit optimisation: Exponential reward functions. Association for Computing Machinery, New York, NY, USA (2024)
7. Raussendorf, R., Browne, D.E.e.a.: Measurement-based quantum computation on cluster states. Phys. Rev. A **68**, 022312 (Aug 2003)
8. Schlingemann, D., Werner, R.: Quantum error-correcting codes associated with graphs. Phys. Rev. A **65**, 012308 (2001)
9. Watkins, C., Dayan, P.: Q-learning. Machine Learning **8**(3), 279–292 (1992)