

AlphaClifford: Efficient Clifford Synthesis and Transpilation with Model-based RL

Daniele Lizzio Bosco, Jacopo Cossio, Carla Piazza, Giuseppe Serra
 University of Udine, Italy; University Federico II of Naples, Italy
 lizziobosco.daniele@spes.uniud.it, cossio.jacopo@spes.uniud.it

Abstract

Clifford circuits play a foundational role in quantum computing, particularly due to their importance in quantum error correction and fault-tolerant logical synthesis. While these circuits can be efficiently simulated and represented as symplectic matrices, standard synthesis methods, such as the Aaronson-Gottesman algorithm, often yield sub-optimal circuits with excessively high gate counts. In this work, we introduce AlphaClifford, a model-based Reinforcement Learning framework based on Monte Carlo Tree Search, designed to efficiently synthesize Clifford circuits from the fundamental gate set composed of H , S , and CNOT. By modeling the state space through the algebraic properties of the symplectic group, AlphaClifford effectively explores this combinatorial space to minimize overall circuit cost. For unconstrained Clifford optimization, our approach achieves a consistent reduction in both total and two-qubit (CNOT) gate counts compared to state-of-the-art synthesis heuristics. Furthermore, we demonstrate the broad applicability of our framework on two additional tasks: hardware-constrained Clifford transpilation, where we outperform existing RL-based compilers, and as a post-synthesis optimization component within a full Clifford+T logical synthesis pipeline.

1 Introduction

The field of quantum computing is rapidly approaching the era of “quantum utility” [1], but the execution of quantum algorithms remains fundamentally constrained by circuit depth and the total number of physical gates. To mitigate these hardware limitations, quantum circuit optimization plays a critical role. As equivalent unitary transformations can be realized by different sequences of quantum gates, the objective of quantum circuit synthesis is to find a sequence of gates that implements a target circuit while minimizing a specific cost metric, such as overall circuit depth or the number of two-qubit entangling gates. A closely related challenge is quantum circuit transpilation, which maps a logical circuit onto a specific physical architecture and respects strict hardware constraints, such as the limited connectivity map of the physical qubits [2].

Recently, Reinforcement Learning (RL) has emerged as a highly effective paradigm for tackling these computationally hard optimization problems [3, 4, 5]. In this work, we propose a RL-based technique to address both the synthesis and transpilation tasks specifically for the set of Clifford circuits. Generated strictly by the Hadamard (H), Phase (S), and Controlled-NOT (CNOT) gates, Clifford circuits are one of the most studied gate sets, and play a central role in the logical synthesis pipeline, particularly within the context of Clifford+T decomposition [6].

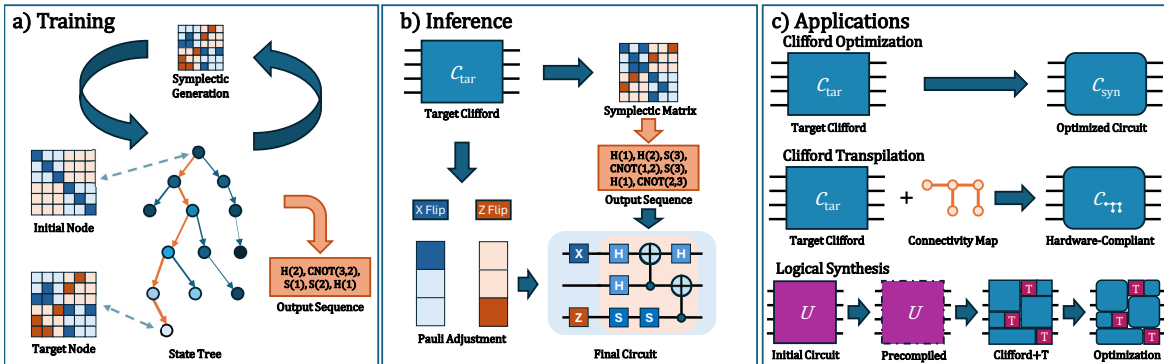


Figure 1: AlphaClifford overview: symplectic matrices are modeled as nodes in a search tree, MCTS learns policy/value guidance over Clifford actions, and the synthesized circuit is corrected by a residual Pauli layer.

To demonstrate the broad applicability of our proposed method, we evaluate it on three different tasks: unconstrained Clifford optimization, Clifford transpilation to a target hardware connectivity map, and as a refinement step in a full logical synthesis pipeline. We show that our model achieves comparable or higher performance with respect to a variety of different techniques, including up to 56% total gate reduction compared to the Aaronson-Gottesman algorithm [7] for the general Clifford optimization task, and up to 20% compared to the Clifford transpiler RL-QISKIT [4] on the task of transpilation.

2 Symplectic Representation

The n -qubit Pauli group $\mathcal{P}_n$ consists of all n -fold tensor products of the Pauli matrices $\{I, X, Y, Z\}$ together with the multiplicative phase factors $\{\pm 1, \pm i\}$. The Clifford group $\mathcal{C}_n$ is defined as the normalizer of the Pauli group within the unitary group $U(2^n)$. The group $\mathcal{C}_n$ is finitely generated by the Hadamard (H), Phase (S), and Controlled-NOT (CNOT) gates. By the Gottesman-Knill theorem [8], Clifford circuits can be efficiently simulated on classical hardware via the standard tableau representation introduced by Aaronson and Gottesman [7].

While the phase vector $\mathbf{r}$ is required to track the exact quantum state, the fundamental entangling structure and basis transformations induced by a Clifford circuit are entirely captured by the $2n \times 2n$ binary matrix $M = [\mathcal{X} \ \mathcal{Z}]$. To preserve the commutation relations of the Pauli group, the matrix M must satisfy the symplectic condition over $\mathbb{F}_2$:

$$M\Omega M^T = \Omega \pmod{2}, \quad \Omega = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}. \quad (1)$$

Consequently, M is an element of the symplectic group $Sp(2n, \mathbb{F}_2)$. Crucially, there exists a surjective group homomorphism $\phi : \mathcal{C}_n \rightarrow Sp(2n, \mathbb{F}_2)$ that maps a Clifford unitary to its corresponding symplectic matrix. The kernel of this homomorphism is precisely the Pauli group $\mathcal{P}_n$. If an RL agent restricts its objective to finding a sequence of $\{H, S, \text{CNOT}\}$ gates that yields a symplectic matrix M_{syn} matching a target symplectic matrix M_{target} , the synthesized circuit C_{syn} and the target circuit C_{target} are guaranteed to belong to the same coset of the Pauli group:

$$C_{\text{target}} \equiv P \cdot C_{\text{syn}}, \quad P \in \mathcal{P}_n. \quad (2)$$

By tracking the evolution of the phase vector $\mathbf{r}$ alongside the synthesis process, the residual Pauli correction P can be computed efficiently in $\mathcal{O}(n^2)$ time.

3 AlphaClifford

Following the model-based principle, and inspired by recent applications such as AlphaTensor [3] and AlphaCNOT [5], we propose **AlphaClifford**, a novel model-based approach for exploring the combinatorial space of symplectic matrices associated with Clifford circuits. Our method, illustrated in Fig. 1, is organized into three steps. First, we model the space of symplectic matrices as a search tree, where each node represents a matrix and directed edges correspond to Clifford gates. Second, we train a model to learn this space by predicting gate sequences that transform the initial symplectic matrix into the target one. Third, the trained model synthesizes Clifford circuits by generating a new sequence of gates with the same symplectic matrix as the target circuit.

The search space of our problem, represented by the set of symplectic matrices, is explored through a tree-based approach. For a given problem dimension n , a target Clifford circuit C represented by its symplectic matrix $M_C \in Sp(2n, \mathbb{F}_2)$, and possibly a connectivity map $\mathcal{M}$, our aim is to determine a sequence of Clifford gates equivalent to C . In the unconstrained setting, each node has $\Theta(n^2)$ children due to the possible CNOT operations between pairs of qubits, along with single-qubit gates. In the case of limited connectivity constraints, only a subset of two-qubit operations is allowed.

We adopt a Monte Carlo Tree Search [9] strategy to efficiently explore this space. The procedure follows the standard phases of selection, expansion, simulation, and backpropagation, using the Upper Confidence Bound for Trees criterion [10] to balance exploration and exploitation. We use a non-informed reward: each step receives a penalty of $-1/\sqrt{n}$ and a terminal reward of $+n$ is assigned when reaching the identity matrix.

Training follows a linear curriculum strategy [11] and stops after the maximum difficulty of $k_{\max} = 4n^2$ is reached.

4 Experimental Results

Both policy and value networks share the same architecture, i.e., a 12-layer fully connected network with 256 hidden units per layer. We structured the net using 4 residual blocks with skip connections every three layers. We trained a different model for each number of qubits n , and for each selected connectivity map for the transpilation task.

As a first task, we evaluate the ability of our model to efficiently synthesize sufficiently complex Clifford circuits. This problem can be reformulated as a Clifford optimization task: we start from a random Clifford circuit generated by `qiskit` [12], and we synthesize an equivalent circuit with gates from $\{H, S, \text{CNOT}\}$. For $n \in \{3, 4, 5, 6, 7\}$, we evaluate our model on 100 Clifford circuits involving up to $4n^2$ gates. As comparison, we consider the well-known Aaronson-Gottesman algorithm [7], as implemented in `qiskit`, and heuristic models such as A^* and Volanto [13, 14]. Results are reported in Table 1.

Table 1: Clifford optimization: average gate counts over 100 random Clifford instances.

Metric	Approach	3	4	5	6	7
Total gates	Aaronson-Gottesman	19.99	25.89	38.05	51.37	65.61
	A^* (with transv.)	12.97	19.56	26.54	35.04	43.15
	Volanto (with transv.)	12.95	19.93	28.65	38.12	48.14
	AlphaClifford₁₀	8.78	13.39	19.49	27.11	36.75
2-qubit gates	Aaronson-Gottesman	3.65	10.41	16.17	22.43	28.59
	A^* (with transv.)	6.34	10.63	15.72	21.39	27.32
	Volanto (with transv.)	6.38	11.16	17.36	24.38	32.62
	AlphaClifford₁₀	4.06	6.96	10.98	15.82	23.06

Table 2: Transpilation: total gate counts on connectivity maps from [4].

Method	3L	4L	5L	4T	5T	6T
RL-Qiskit ₁₀	12.19	22.90	33.31	21.58	30.92	46.11
AlphaClifford₁₀	9.81	17.08	27.58	15.76	25.04	38.84

Table 3: Logical synthesis pipeline: Clifford gate-count reductions, with and without Q-PreSyn [15].

Setting	Method	5	10	15	20	25
No pre-syn.	Qiskit red. (%)	0.36	0.13	0.09	0.14	0.14
No pre-syn.	AlphaClifford red. (%)	2.44	2.39	2.53	2.50	2.43
With Q-PreSyn	Qiskit red. (%)	0.35	0.17	0.15	0.12	0.20
With Q-PreSyn	AlphaClifford red. (%)	2.75	2.46	2.57	2.50	2.49

We observe that our model achieves the lowest amount of total gates, and two-qubit gates, in most cases. We stress that the advantage compared to the methods in [13] is obtained with a less expressive gate set. For transpilation, we select 6 connectivity maps from 3 to 6 qubits, following the procedure in [4], and compare against the RL-based Clifford transpiler by `qiskit`, available at [16]. Results in Table 2 shows a total gate count reduction depending on the complexity of the topology. Finally, we evaluate AlphaClifford as a component of a logical synthesis pipeline, both with and without the pre-synthesis strategy Q-PreSyn [15]: starting from random circuits, we decompose them into a Clifford+T representation and optimize adjacent Clifford blocks of at least 2 qubits. We compare our results with the standard `qiskit` transpilation, as other methods were not able to reduce the total gate count. In Table 3, we highlight the increased reduction of Clifford gates in general circuits, with respect to the standard `qiskit` transpilation.

5 Conclusion

In this work, we introduced AlphaClifford, a model-based reinforcement learning framework for Clifford circuit synthesis and transpilation. By formulating compilation as Monte Carlo Tree Search over symplectic matrices, AlphaClifford consistently discovers optimized gate sequences and achieves superior or comparable results to state-of-the-art methods across multiple settings. These results highlight the potential of model-based RL to transcend the limitations of traditional heuristic-driven compilers.

References

- [1] Youngseok Kim et al. “Evidence for the utility of quantum computing before fault tolerance”. In: *Nature* 618 (2023), pp. 500–505.
- [2] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.
- [3] Francisco J. R. Ruiz et al. “Quantum Circuit Optimization with AlphaTensor”. In: *Nature Machine Intelligence* 7 (2025), pp. 374–385.
- [4] David Kremer et al. “Practical and efficient quantum circuit synthesis and transpiling with Reinforcement Learning”. In: *arXiv preprint arXiv:2405.13196* (2024).
- [5] Jacopo Cossio et al. *AlphaCNOT: Learning CNOT Minimization with Model-Based Planning*. 2026. arXiv: 2604.13812 [cs.AI].
- [6] David Kremer, Ali Javadi-Abhari, and Priyanka Mukhopadhyay. *Optimizing the non-Clifford-count in unitary synthesis using Reinforcement Learning*. 2025. arXiv: 2509.21709 [quant-ph].
- [7] Scott Aaronson and Daniel Gottesman. “Improved simulation of stabilizer circuits”. In: *Physical Review A* 70 (2004), p. 052328.
- [8] Daniel Gottesman. “Stabilizer codes and quantum error correction”. In: *arXiv preprint quant-ph/9705052* (1997).
- [9] Remi Coulom. “Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search”. In: *Computers and Games: 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*. 2007.
- [10] Levente Kocsis and Csaba Szepesvari. “Bandit Based Monte-Carlo Planning”. In: *Machine Learning: ECML 2006, 17th European Conference on Machine Learning*. 2006.
- [11] Yoshua Bengio et al. “Curriculum learning”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009.
- [12] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. arXiv: 2405.08810 [quant-ph].
- [13] Mark Webster, Stergios Koutsoumpas, and Dan E. Browne. *Heuristic and Optimal Synthesis of CNOT and Clifford Circuits*. 2025. arXiv: 2503.14660 [quant-ph].
- [14] Kalle Volanto. “Minimizing the Number of Two-Qubit Gates in Clifford Circuits”. Master’s thesis. Aalto University, 2023.
- [15] Daniele Lizzio Bosco et al. *Quantum Circuit Pre-Synthesis: Learning Local Edits to Reduce T-count*. 2026. arXiv: 2601.19738 [quant-ph].
- [16] Qiskit. *AI-transpiler cliffords*. Hugging Face model repository. Accessed: Apr. 21, 2026. 2024.