# Reinforcement Learning based Quantum Circuit Optimization via ZX-Calculus

**Jordi Riu**[1], Jan Nogué[1], Gerard Vilaplana[1], Artur Garcia-Saez[1,2], Marta P. Estarellas[1]
[1]Qilimanjaro Quantum Tech., Carrer dels Comtes de Bell-Lloc, 161, 08014 Barcelona, Spain
[2]Barcelona Supercomputing Center, Plaça Eusebi Güell, 1-3, 08034 Barcelona, Spain

jordi.riu@qilimanjaro.tech

Current quantum devices face significant challenges, such as the presence of noise and decoherence in the physical systems that implement quantum circuits [1]. These challenges limit the scalability and the reliability of quantum computation, and pose a major obstacle for achieving quantum advantage over classical computation. Therefore, it is essential to minimize the resources required to implement quantum algorithms, while preserving the functionality and the fidelity of the computation. In the gate-based computational paradigm, one of the common approaches for optimizing quantum circuits is to apply algebraic identities to perform gate permutations and gate cancellations in the original circuit [2, 3]. However, the number of available identities that can be applied to a circuit grows very quickly, as these identities need to be described for multiple combinations of gates.

ZX-Calculus [4] is a graphical language that facilitates reasoning about quantum processes. Most notably, it has emerged as a versatile tool for manipulating quantum circuits. A ZX-diagram depicts a more general representation of a quantum circuit and it can be modified using a much more reduced set of transformations called ZX-rules, some of which can not be represented using the gate-based formalism. The basic elements of a ZX-diagram are spiders (nodes) and wires (edges). Spiders can be of two types: Z and X, and they can be interpreted as tensors composed of Pauli-Z and Pauli-X eigenstates, respectively [FIG.1].

*Local complementation* and *pivoting* [5,6] are the two essential rules that are used to *simplify* ZX-diagrams in graph-like form. Both rules are inspired in their counterparts from graph theory and change the diagram by rewiring its connections and modifying the phases of the spiders involved in the transformation [FIG.2]

Even though quantum circuit optimization via ZX-calculus has shown very promising results, current state of the art approaches suffer from the fact that, after the simplification process is finished, one needs to translate the diagram back into an equivalent quantum circuit. This process can be very inefficient or even unfeasible in some cases [7] and, paradoxically, it can output circuits that are more computationally-expensive than the initial ones [8].

To correct this, we use Reinforcement Learning [9] to train an agent to perform the right set of transformations to any ZX-diagram, i.e, those that yield an improved quantum circuits at the end of the extraction process. In more detail, we use an actor-critic method, the Proximal Policy Optimization algorithm (PPO) [10], and employ Graph Neural Networks (GNNs) [11] to interpolate both the policy and value functions [FIG.3].

Graph Neural Networks (GNNs) are a type of artificial neural networks that are particularly well suited to work with data that can be represented as graphs, as it is the case for ZX-diagrams. These type of networks typically involve message passing layers that propagate the information of each node of the network to its nearest neighbours, i.e., the nodes that are connected to it.

We train our agent to target single and two-qubit gate count reduction on random circuits of Clifford+T gates, which are known to allow for universal quantum computation. After training, we benchmark our approach against the state-of-the-art ZX-calculus [12, 13] based algorithms as well as gate-based optimizers for such tasks.

The agent is shown to be able to improve on the results obtained by all the competitors and, not only that, it can generalize the strategies learned from 5-qubit circuits to 60-qubit circuits of up to 1200 gates [FIG 4,5,6] whilst remaining competitive in terms of computational performance.

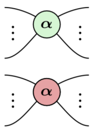
Our work lays a new foundation for quantum circuit optimization via ZX-Calculus, proving to improve the stat-of-the art for continuous use but also demonstrating capabilities to optimise particular circuits via strategies such as curriculum learning. A preprint version of this work can be found at [14].

## References

1. J. Preskill, Quantum computing in the NISQ era and beyond, Quantum 2, 79 (2018)
2. S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A 70, 052328 (2004)
3. V. Kliuchnikov and D. Maslov, Optimisation of Clifford circuits. Phys. Rev. A 88 052307 (2013)
4. B. Coecke and A.Kissinger, Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning (Cambridge University Press).
5. A. Kotzig, Eulerian lines in finite 4-valent graphs and their transformations, In Colloquium on Graph Theory Tihany 1966, pages 219 (Academic press, 1968)
6. A. Bouchet, Graphic representations of isotropic systems, J. Comb. Theory Ser. A 45, 58-76 (1987)
7. N. de Beaudrap, A. Kissinger, and J. van de Wetering, Circuit extraction for zx-diagrams can be P-hard (Schloss Dagstuhl - Leibniz-Zentrum for Information, 2022)
8. R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, Graph-theoretic simplification of quantum circuits with the ZX-Calculus, Quantum 4, 279 (2020)
9. R. S. Sutton and A. G. Barton, Reinforcement Learning: An Introduction, 2nd ed (the MIT press, 2018)
10. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal Policy Optimization algorithms (2017) [cs.LG]

11. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P.S. Yu, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems 32, 4 (2021)
12. A. Kissinger and J. van de Wetering, PyZX: Large automated diagrammatic reasoning, Electronic Proceedings in Theoretical Computer Science, 318, 229 (2020)
13. C. Holker, Causal flow preserving optimisation of quantum circuits in the ZX-calculus, (2024), arXiv:2312.02793 [quant-ph].
14. J.Riu, J. Nogué, G. Vilaplana, A. Garcia-Saez and M. P. Estarellas, Reinforcement Learning Based Quantum Circuit Optimization via ZX-Calculus, (2023), arXiv:2312.11597 [quant-ph]


# Figures



$$:= \underbrace{|0\cdots0\rangle}_{m}\underbrace{\langle0\cdots0|}_{n} + e^{i\alpha}\underbrace{|1\cdots1\rangle}_{m}\underbrace{\langle1\cdots1|}_{n}$$

$$:= \underbrace{|+\cdots+\rangle}_{m}\underbrace{\langle+\cdots+|}_{n} + e^{i\alpha}\underbrace{|-\cdots-\rangle}_{m}\underbrace{\langle-\cdots-|}_{n}$$

**Figure 1.** Green and red spiders representation in the Hillbert space.

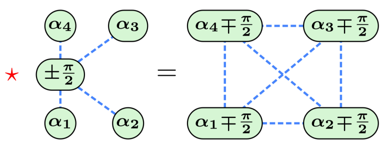



**Figure 2.** Graphical representation of the effect of the Local complementation rule when applied to a spider in the diagram (marked with a red asterisk).

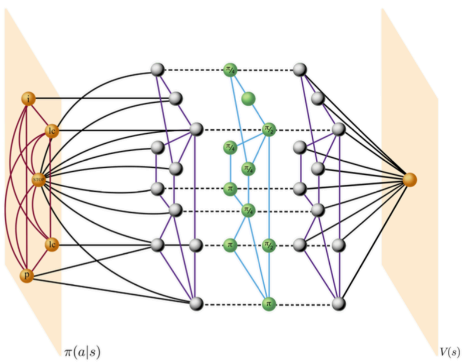



**Figure 3.** Schematic Overview of the Actor and Policy Net-works: The policy network, denoted by π(a|s), is visualized with each green spider from the diagram correspond- ing to a silver node, while the blue wires are depicted as purple connections. Action nodes are highlighted in orange, with interconnections among themselves and with the pertinent silver nodes, which provide a complete description of the actions. The critic network shares a similar architecture.

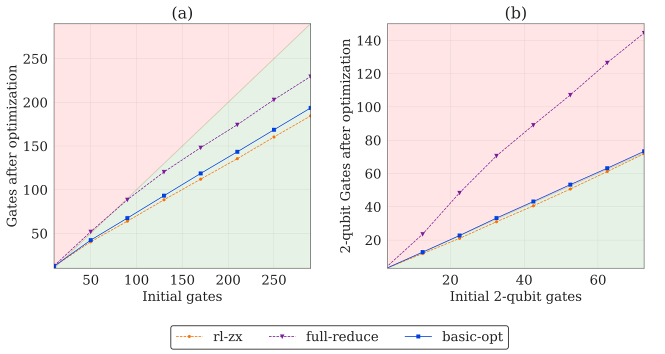



**Figure 4.** Comparative analysis of the RL-ZX agent against the algorithms of the library PyZX for random circuits of 10 qubits. Compared methods include: The agent trained on random circuits of 5 qubits and 60 gates (orange), the full_reduce algorithm (purple) and the basic_optimization algorithm (blue). Regions shaded in red indicate instances of unsuccessful compressions, while those shaded in green denote successful compressions. (a) Total Gate Count. (b) 2-Qubit Gate Count.

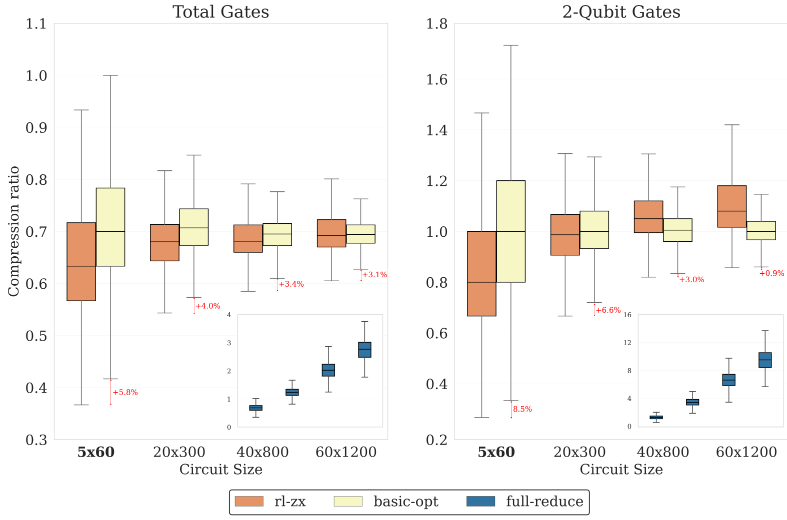



**Figure 5.** Scalability analysis of the RL-ZX agent against the algorithms of the library PyZX for random circuits of up to 60 qubits. Compared methods include: The agent trained on random circuits of 5 qubits and 60 gates (orange), a gate-based optimizer (yellow) and the full_reduce algorithm (blue). (a) Total Gate Count. (b) 2-Qubit Gate Count

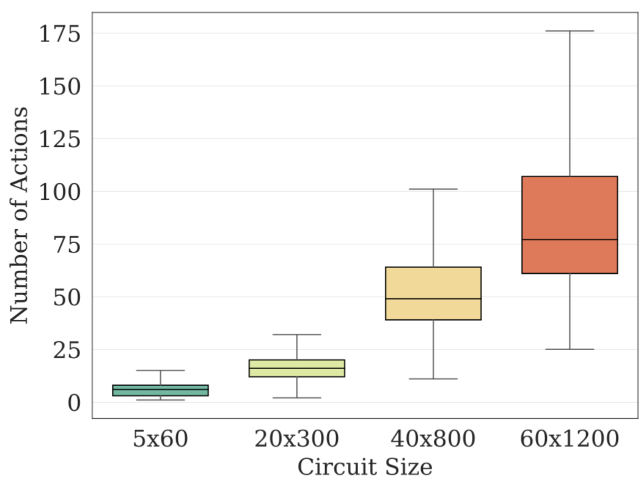



**Figure 6.** Number of actions performed by the agent trained with random circuits of 5 qubits and 70 gates for different circuit sizes.