Qdislib: Implementation and Evaluation of Circuit Cutting and Sampling Techniques for Quantum Circuits

Mar Tejedor
Barcelona Supercomputing Center (BSC)
Spain

Rosa M. Badia
Barcelona Supercomputing Center (BSC)
Spain

Abstract

Near-term quantum processors remain constrained by limited qubit counts, restricting the execution of large quantum circuits. Circuit cutting addresses this limitation by partitioning a circuit into smaller fragments that can be executed independently and classically reconstructed. However, multiple cutting strategies exist, each presenting different trade-offs in computational overhead, hardware requirements, reconstruction accuracy, and runtime performance.

This work presents a systematic evaluation of the four circuit-cutting techniques implemented in Qdislib: gate cutting based on quasiprobability decomposition (QPD), ancilla-assisted Hadamard gate cutting, standard wire cutting, and an optimized wire-cutting formulation. Experimental evaluation demonstrates that optimized wire cutting reduces computational overhead from 8^k to 3^k+4^k fragment evaluations while preserving exact reconstruction, whereas Gate Hadamard matches the accuracy of standard gate cutting while eliminating mid-circuit measurements. Runtime projections further illustrate the scalability enabled by distributed execution on HPC systems.

1 Introduction

Circuit cutting enables quantum circuits that exceed the qubit capacity of available hardware by partitioning them into smaller fragments that are executed independently and later classically reconstructed. Although several cutting techniques have been proposed, each introduces different trade-offs between computational overhead, sampling cost, hardware requirements, and reconstruction accuracy. Selecting the most appropriate method therefore depends on both the target application and the available computing resources.

Qdislib is an open-source framework that unifies these techniques within a single software library while enabling scalable execution on heterogeneous quantum–classical platforms. It provides automatic cut discovery, cost estimation, semantic fragment caching, and distributed execution through PyCOMPSs. A modular backend interface supports Qiskit, Qibo, CUDA-Q, PennyLane, CPU and GPU simulators, and real quantum hardware, allowing applications to execute without backend-specific modifications.

2 Circuit Cutting Methods

The poster compares the four circuit-cutting strategies currently implemented in Qdislib.

- **Gate Cutting (QPD)** decomposes a cut two-qubit gate into six quasiprobability terms whose weighted combination reconstructs the original operation. This approach requires mid-circuit measurements but introduces no additional ancilla qubits.
- **Gate Cutting (Hadamard)** replaces mid-circuit measurements with ancilla-assisted Hadamard tests. It preserves the same 6^k scaling while producing equivalent reconstructed expectation values.
- **Standard Wire Cutting** reconstructs cut quantum wires using measurement-and-preparation operations over the Pauli bases, requiring 8^k fragment configurations.
- **Optimized Wire Cutting** exploits linear dependencies within the reconstruction equations to reduce the number of fragment evaluations from 8^k to $3^k + 4^k$ without affecting the final reconstructed result.

The independent execution of subcircuits naturally exposes parallelism. Qdislib leverages the PyCOMPSs runtime to automatically generate task graphs in which fragment evaluations execute concurrently across HPC resources, while synchronization is only required during the reconstruction stage.

3 Experimental Evaluation

Experimental evaluation was performed using the Qiskit Aer simulator executing on CPU resources. The study compares theoretical complexity, reconstruction accuracy, execution time, and distributed scalability across all four methods.

Theoretical analysis shows that standard wire cutting exhibits the highest computational cost, requiring 8^k fragment evaluations. Optimized wire cutting substantially reduces this cost to $3^k + 4^k$, while both gate-cutting approaches require 6^k configurations. Although Gate Hadamard introduces ancilla qubits, it removes mid-circuit measurements while preserving identical asymptotic complexity.

Accuracy experiments demonstrate that wire-cutting methods consistently achieve the lowest reconstruction error as the number of measurement shots increases, owing to their lower sampling variance. Standard Gate QPD and Gate Hadamard produce nearly identical expectation values, confirming that the Hadamard formulation preserves correctness while reducing execution overhead by replacing hardware-supported mid-circuit measurements with ancilla-assisted circuits.

The poster also compares expectation-value reconstruction with full probability-distribution reconstruction. While both modes execute the same fragment configurations, reconstructing the complete output distribution requires additional memory and classical post-processing to recover all computational basis states. Experimental results illustrate how increasing the number of measurement shots progressively reduces sampling noise and total variation distance.

Finally, runtime projections demonstrate that the algorithmic improvements directly translate into practical scalability. Optimized wire cutting consistently provides the lowest projected execution time for increasing numbers of cuts, while distributed execution

across 64 MareNostrum 5 nodes reduces projected runtimes from years to only a few hours for representative workloads.

4 Conclusion

This work presents a systematic evaluation of the four circuit-cutting techniques implemented in Qdislib under a unified software framework. The results demonstrate that no single strategy is universally optimal. Gate Hadamard preserves the accuracy of standard gate cutting while eliminating mid-circuit measurements, at the cost of introducing two ancilla qubits per cut, which can become a limiting factor as the number of cuts increases. In contrast, optimized wire cutting significantly reduces computational overhead by lowering fragment scaling from 8^k to $3^k + 4^k$.

Together, these results provide practical geuidelines for selecting circuit-cutting strategies across simulators, HPC systems, and emerging quantum hardware. Combined with Qdislib's distributed execution capabilities, they establish a scalable foundation for hybrid quantum–classical computing and large-scale circuit-cutting workloads.

References

[1] T. Peng, A. W. Harrow, M. Ozols, and X. Wu. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125(15):150504, 2020.
[2] K. Mitarai and K. Fujii. Constructing a virtual two-qubit gate by sampling single-qubit operations. *New Journal of Physics*, 23(2):023021, 2021.
[3] K. Mitarai and K. Fujii. Overhead for simulating a non-local channel with local channels by quasiprobability sampling. *Quantum*, 5:388, 2021.
[4] A. W. Harrow and A. Lowe. Optimal quantum circuit cuts with application to clustered Hamiltonian simulation. *PRX Quantum*, 6(1):010316, 2025.
[5] M. Tejedor, B. Casas, J. Conejero, A. Cervera-Lierta, and R. M. Badia. Orchestrating Quantum-HPC Workflows with Distributed Quantum Circuit Cutting. In *Proceedings of the SC Workshops*, pages 1898–1906, 2025.