Towards A Distributed QHPC Workflow for Gadget-Aware Circuit Cutting & Pulse Synthesis

NC STATE UNIVERSITY
Dept. of Computer Science

Ria Patel[1], Silvina Caíno-Lores[2], Frank Mueller[1]

[1]Department of Computer Science, North Carolina State University, Raleigh, NC 27606

[2]University of Rennes, Inria, CNRS, IRISA, Rennes, France

Inria

1. INTRODUCTION

PROBLEM MOTIVATION

- Deep circuit algorithms suffer severe decoherence & error accumulation.
- **Compound pulse gadget synthesis** compresses repeating discrete gate structures into optimized continuous pulses.
  - Reduces error accumulation & pulse schedule time.
- Optimal control algorithms compute dense matrices to synthesize these continuous pulses
  - Exhausts classical memory & compute resources

GAPS IN CURRENT CIRCUIT-CUTTING WORKFLOWS

- Scaling pulse gadget synthesis requires distributed workflows.
  - Circuit cutting frameworks (Qdislib, CutQC) can do this but operate at the gate or wire level.
- Partitioning circuits destroys the macro-gate's structural dependencies required for compound pulse gadget synthesis.

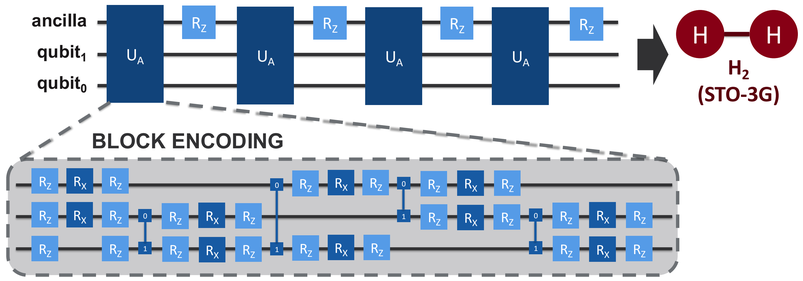
KODOKI FRAMEWORK

- **Decouples** offline compound pulse gadget synthesis from online distributed subcircuit emulation to bypass centralized memory bottlenecks.
- **Gadget-aware cutting** natively enforces macroscopic boundaries during circuit cut to preserve structural integrity.

2. QUANTUM ENVIRONMENT

QSVT CIRCUIT ON MOLECULE

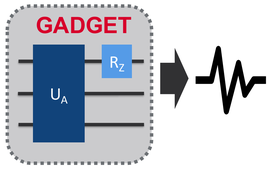
- **Quantum Singular Value Transformation** applies polynomial transforms to singular values of block-encoded (BE) matrix to compute target matrix function.
  - BE embeds non-unitary matrix into unitary matrix → explodes classical memory due to large circuit depth.
  - Circuit alternates between signal process rotations (R_Z) & BE target (U_A) to build transform.
- **QSVT for Hamiltonian simulation** maps time evolution operator (e^{-iHt}) polynomial approximate onto quantum hardware for molecular system simulation.



H_2 MOLECULAR HAMILTONIAN

$$H_{H_2} = \underbrace{-1.0524(I \otimes I)}_{\text{Scalar energy shift}} + \underbrace{0.3979(I \otimes Z) - 0.3979(Z \otimes I) - 0.0113(Z \otimes Z)}_{\text{Potential energies of electrons in orbitals}} + \underbrace{0.1809(X \otimes X)}_{\text{Electron hopping}}$$

GADGET



TIME-DEPENDENT SYSTEM HAMILTONIAN

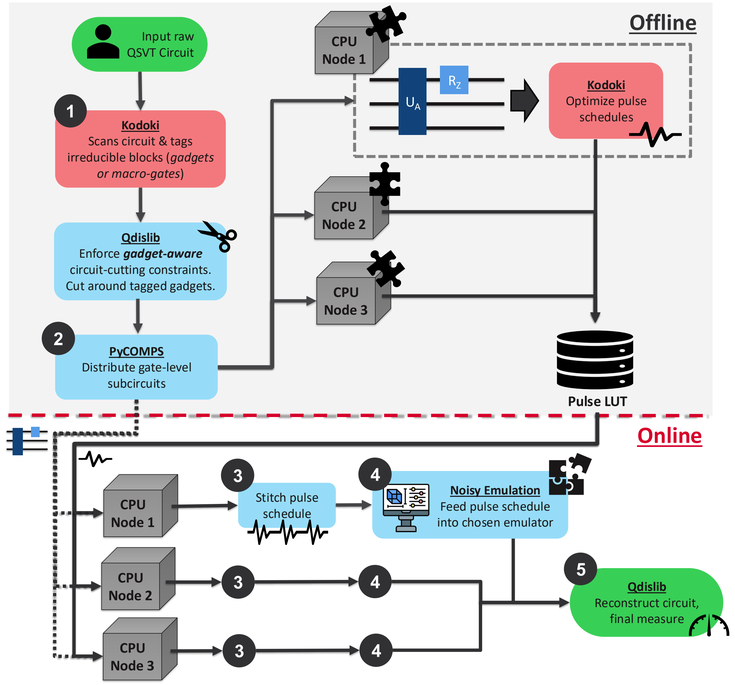
$$H_I = H_c + H_d \quad \text{(Interact Ham.)}$$

$$H_c^{(j)} = \Omega_j(t)\sigma_x^j \sum_{k=1}^{N_{\text{modes}}} \eta_k b_{j,k}(a_k + a_k^\dagger) \quad \text{(Ctrl Ham. / Drive Ham.; Rabi Freq.; Lamb-Dicke)}$$

$$H_d = \sum_{k=1}^{N_{\text{modes}}} \delta_k a_k^\dagger a_k, \qquad \delta_k = \mu - \omega_k \quad \text{(Laser Detuning)}$$

3. METHODOLOGY

DISTRIBUTED SCATTER-GATHER MODEL



Offline Synthesis (Pre-Compute):

1. **Pre-Process & Cut (Scatter):**
   - Kodoki tags gadgets based on hardware limits
   - Qdislib executes gadget-aware cuts to keep boundaries intact
2. **Distributed Synthesis (Map):**
   - PyCOMPS sends subcircuits to worker nodes for pulse synthesis
     - Stores outputs in look-up table (LUT) stored on master node

Online Emulation (Execution):

3. **Pulse Schedule Assembly (Fetch):**
   - Workers query LUT (on master node)
   - Fetch & stitch pulse gadgets together to create pulse schedule
4. **Noisy Emulation (Execute):**
   - Workers run separate noisy emulator simulations on each subcircuit
   - Prevents centralized memory bottlenecks
5. **Reconstruction (Gather):**
   - Qdislib on master node recombines subcircuit expectation values
   - Measures final state

4. HPC TESTBED & SOFTWARE STACK

NC STATE ARC CLUSTER

HARDWARE SPECS

AMD EPYC Siena	
Nodes	13
Cores	16
Threads	32
DDR5	192GB

JOB CONFIG

Nodes	2
CPUs/Task	16
Task/Node	1

SOFTWARE STACK

Python	v3.12
COMPSs	v3.3.3
QuTiP	v5.3.0
Qiskit	v2.4.2
Singularity	v3.7.1
Docker	v29.4.0
Qdislib	v1.0.1

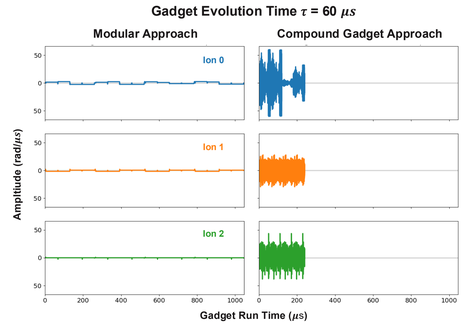


5. EVALUATION

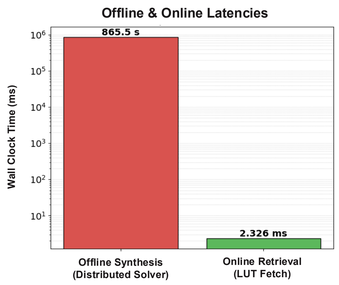
RESEARCH QUESTIONS

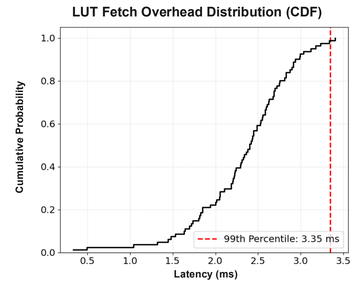
1. Can compound pulse synthesis reduce pulse schedule windows of gadgets?
2. Does moving pulse synthesis offline bypass classical compute wall? What is online latency of LUT queries?
3. What physical hardware constraints does offline optimal control successfully converge to valid pulse schedules?

A1: 4.38x temporal compression (Step 3)



A2: Reduced bottleneck to 2.326 ms & squeezed LUT fetch window (Steps 2-4)

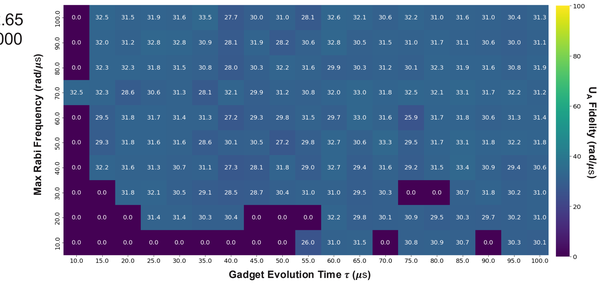




A3: Good pulses generated for $\tau \geq 60\ \mu s$ & max Rabi ≥ 40 rad/μs (Step 2)

Laser Rabi Freq (rad/μs): 2.65
Time Steps: 2000

U_A pulse synthesis:
- **Fails**: $\tau < 20\ \mu s$ → 0% fidelity
- **Converges**: $\tau \geq 60\ \mu s$ & max Rabi ≥ 40 rad/μs → $\geq 26\%$ fidelity



6. CONCLUSION & FUTURE WORK

We introduced Kodoki, a framework that:
- synthesizes compound pulse gadgets for deep circuits
- decouples offline pulse gadget synthesis & online LUT retrieval
- compresses pulse schedules for simulated trapped-ion system

FUTURE WORK

- Develop custom boundary constraint hooks within Qdislib to perform gadget-aware cutting
- Extend Kodoki to other deep circuit algorithms like variational ansatzes (VQAs) to evaluate scaling behavior across diverse compound gadgets.
- Perform multi-node strong & weak scaling. PYCOMPSs profiling to quantify task orchestration overhead, worker communication costs, & study overhead of offline synthesis & online emulation.

References:

[1] Rosa M. Badia, Javier Conejero, Jorge Ejarque, Daniele Lezzi, Francesc Lordan, and Raül Sirvent. 2026. Programming task-based workflows with COMPSs. (2026). DOI:https://doi.org/10.14279/DEPOSITONCE-25823

[2] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. 2005. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. Journal of Magnetic Resonance 172, 2 (February 2005), 296–305. DOI:https://doi.org/10.1016/j.jmr.2004.11.004

[3] Neill Lambert, Eric Giguère, Paul Menczel, Boxi Li, Patrick Hopf, Gerardo Suárez, Marc Gali, Jake Lishman, Rushiraj Gadhvi, Rochisha Agarwal, Asier Galicia, Nathan Shammah, Paul Nation, J. R. Johansson, Shahnawaz Ahmed, Simon Cross, Alexander Pitchford, and Franco Nori. 2026. QuTiP 5: The Quantum Toolbox in Python. Physics Reports 1153, (2026), 1–62. DOI:https://doi.org/10.1016/j.physrep.2025.10.001

[4] Ria Patel, Masoud Hakimi Heris, Yuan Liu, and Frank Mueller. 2026. Synthesizing Compound Pulse Gadgets for Hamiltonian Simulation on Trapped-Ion Platforms. Retrieved from https://arxiv.org/abs/2607.00826

[5] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. 2021. CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 473–486. DOI:https://doi.org/10.1145/3445814.3446758

[6] Mar Tejedor, Berta Cervera, Javier Conejero, Alba Cervera-Lierta, and Rosa M. Badia. 2025. Distributed Quantum Circuit Cutting for Hybrid Quantum-Classical High-Performance Computing. In https://arxiv.org/abs/2505.01184, 1–12.

This work was supported by the U.S. National Science Foundation (grants OSI-2531350, OMA-2120757, PHY-2325080) and the DOE Office of Science ASCR (Contract DE-SC0025384 for F. Mueller). Additional support was provided to S. Caíno-Lores by the French National Research Agency (JCJC ANR-25-CE46-6886-01), the France 2030 HOI initiative (ANR-22-PNCQ-0002), and the Metropolitan Area of Rennes (AIS #202500254O).