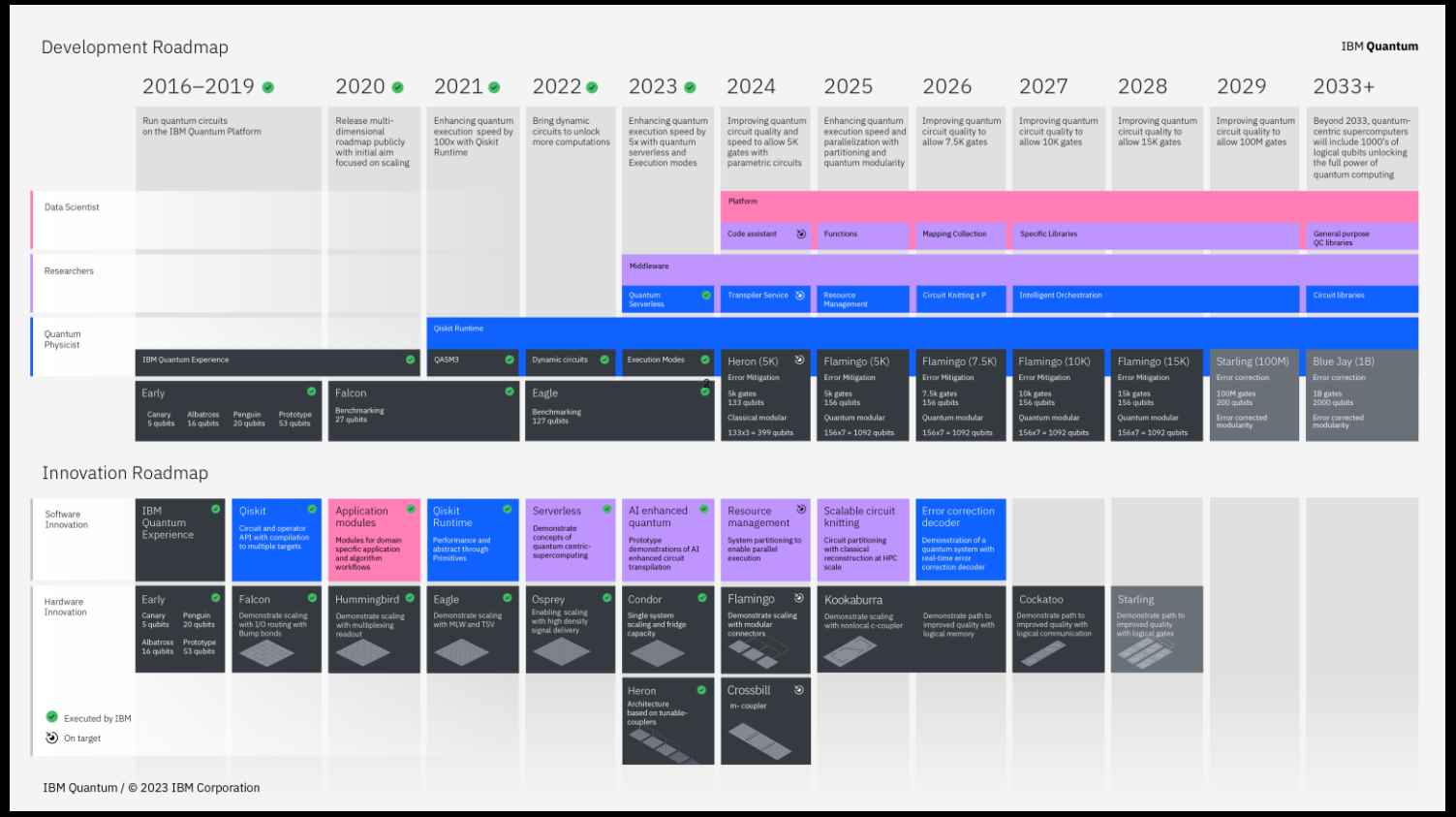
Crosstalk-aware qubit mappings for arbitrary hardware topologies

Team 8: Case Chicago Quantum

Motivation

- NISQ-era devices are highly susceptible to noise
- Circuit fidelity degradation due to accumulated errors limit scalability
- Classically-enabled error mitigation is a near-term remedy



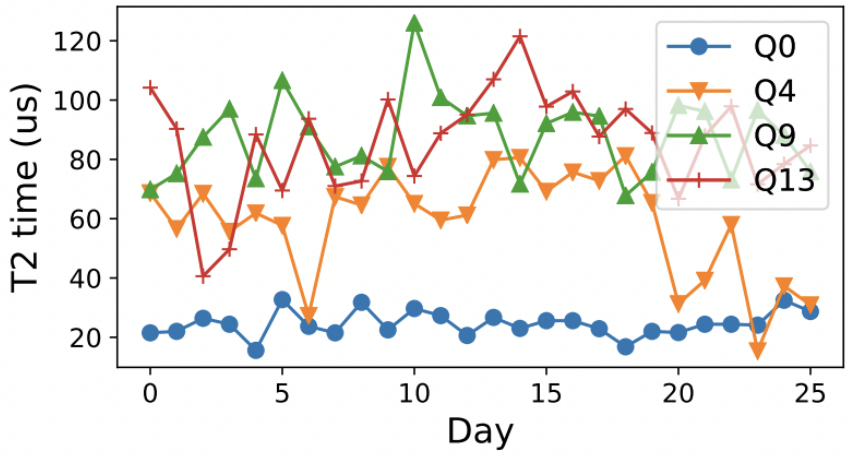
Case study: compiler mappings

“BARE” CIRCUIT MAPPINGS

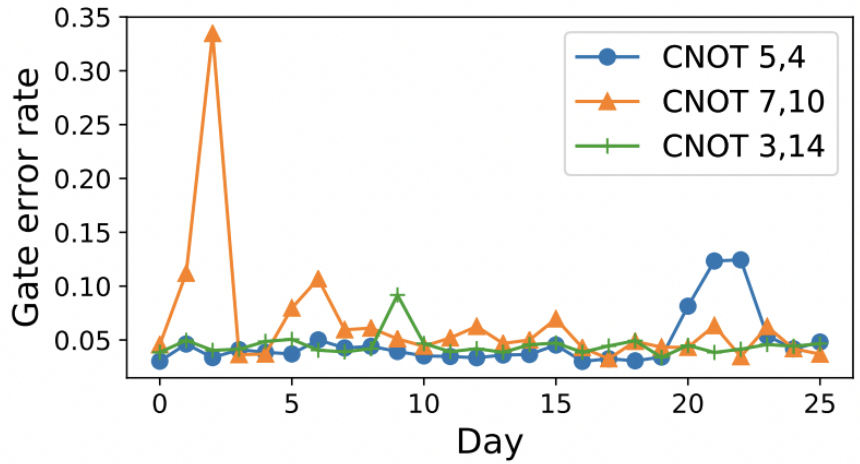
- Arbitrarily maps logical qubits from a quantum circuit to physical qubits
- Unrealistically assumes all qubits are equally reliable and incur identical noise
- Prioritizes processor connectivity over noise robustness

ENTANGLEMENT-BASED SCHEMES

- Maps logical qubits to physical qubits based on which have the corresponding highest fidelity two-qubit gates
  + Reflects that two-qubit error rates are the main source of error
- Classical precomputation clock cycles much quicker than quantum

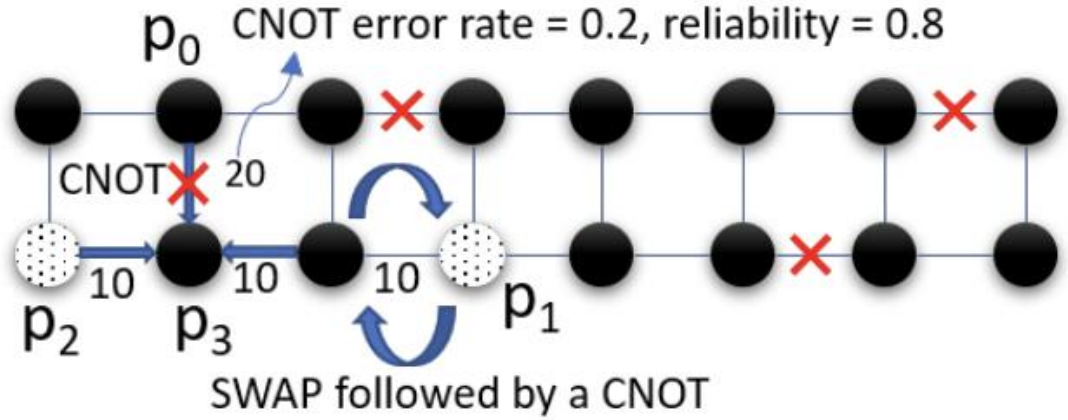


(a) Coherence time (T2)

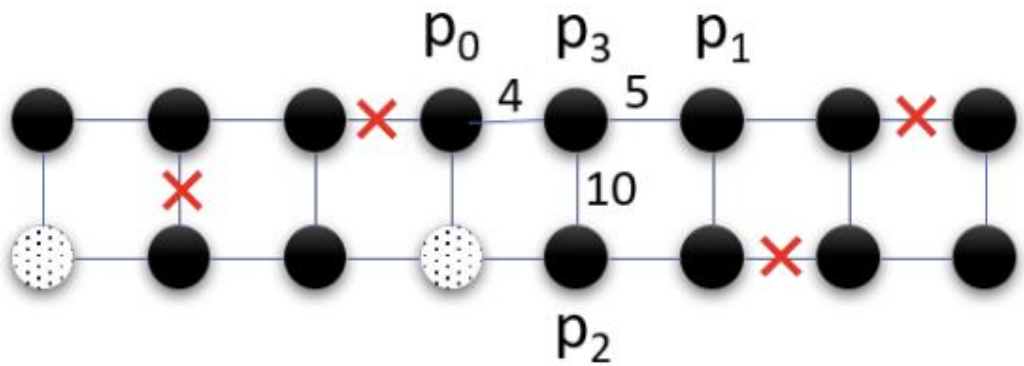


(b) CNOT gate error rate

Figure 1. Daily variations in qubit coherence time (larger is better) and gate error rates (lower is better) for selected elements in *IBMQ 16 Rueschlikon*. The qubits and gates that are most or least reliable are different across days.

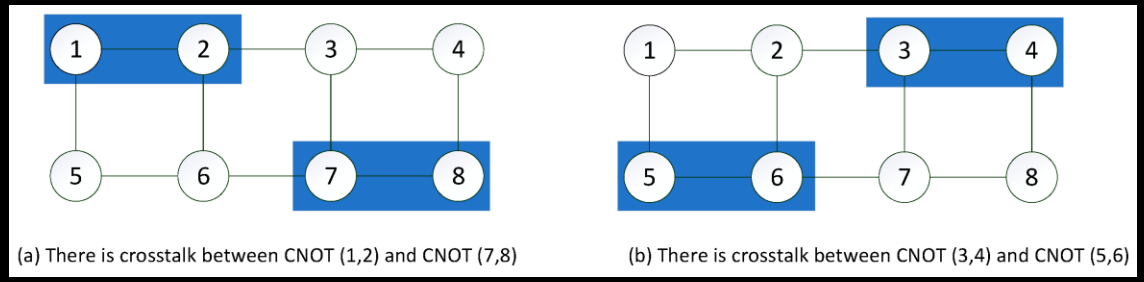


(b) Layout of qubits in *IBMQ16* and a naive mapping for BV4.



(c) Optimized mapping for BV4

But we can do better



- The result of this work suggests it is optimal to cluster qubit mappings together in high edge density areas
  + Makes circuit more susceptible to ***crosstalk*** noise from multiqubit gates
- Crosstalk is responsible for **2/3 of the total error** on the IBM Quantum Montreal processor (27 qubits) in random many-qubit circuit layers
  + Scales severely with parallelized quantum algorithms
  + Dominates noise behavior
- Can we combine compiler mapping results with crosstalk mitigation?

Our Approach

Generalizes the previous result!
Our approach is **proven** to be better than simple EBS!

We base our approach on

- the most recent
- highest performance
- holistic

mapping scheme: QuCloud+

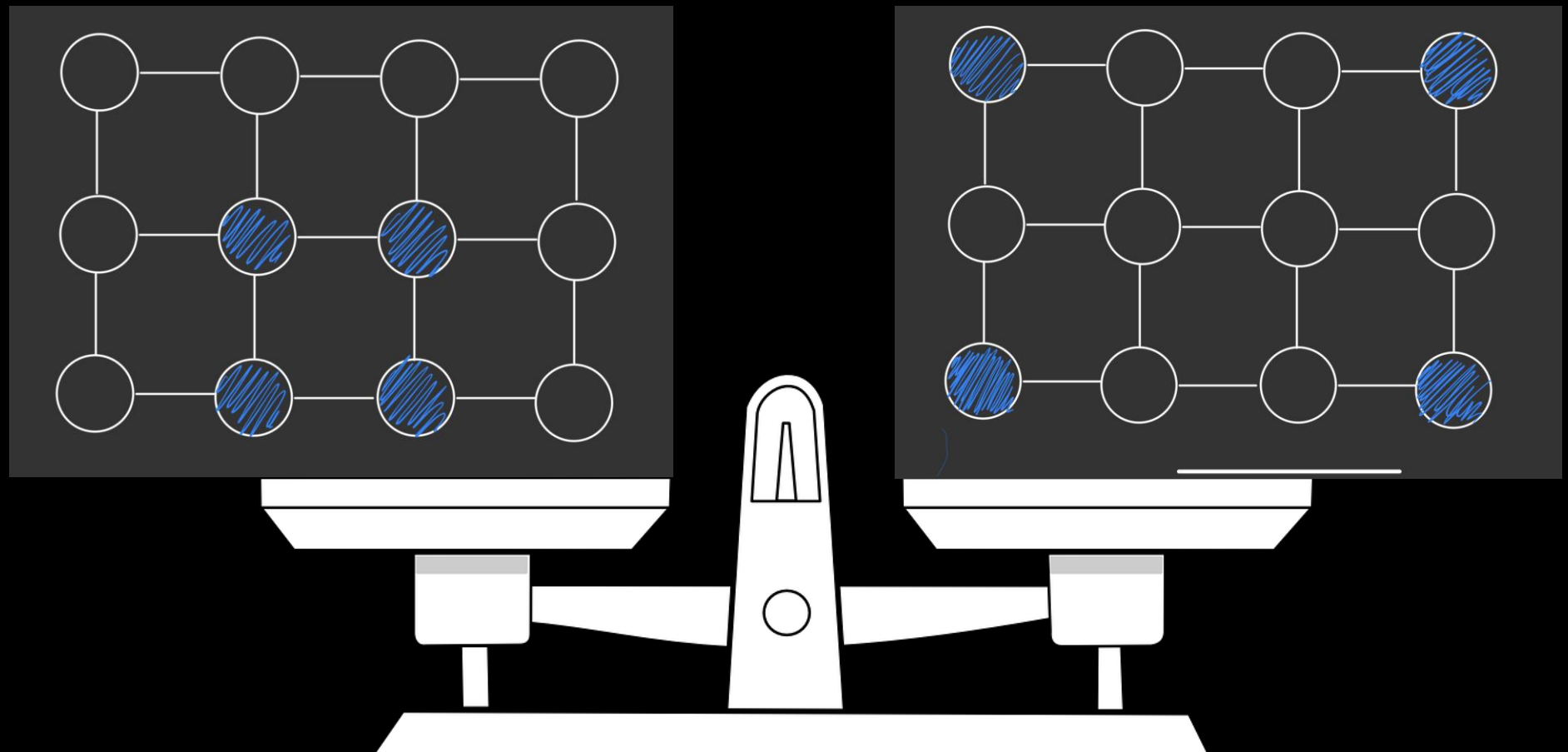
Table 1. Qualitative Comparison of Quantum Program Mapping Approaches

Studies	Approaches	Optimization Objectives	Contributions	Needs Improvements
SABRE [28]	Heuristic approach	Number of gates / circuit depth	Simplified heuristic search space reduces algorithm complexity; reserve traversal for high-quality initial mapping	Unawareness of gate errors
NASSC [30]	Heuristic approach	Number of gates / circuit depth	Leveraging post-mapping gate optimizations to reduce the number of CNOT gates after the mapping process	Unawareness of gate errors in initial mapping generation
FRP [12]	Heuristic approach	Fidelity	Multi-programming for higher throughput; fair qubit partitioning for fidelity; delayed instruction scheduling for lower decoherence errors	Unaware of topology in initial mapping
QuCloud [31]	Heuristic approach	Fidelity / number of gates	Community detection assisted qubits partitioning for fidelity; inter-program SWAPs for fewer additional gates	Unawareness of crosstalk
Palloq [40]	Heuristic approach	Fidelity	Placing idle qubits between concurrent circuits for crosstalk mitigation	Under-utilization of robust qubits due to physical buffers
SATMAP [34]	Optimal approach	Number of gates	Maximum satisfiability (MAXSAT) based solver; Reducing the size of MAXSAT constraints by exploiting the structure of quantum circuits	Unawareness of gate errors
BIP [17]	Optimal approach	Fidelity / circuit depth	Binary integer program (BIP) based solver; merging CNOTs with adjacent SWAPs for fewer additional gates	Reducing crosstalk conflicts with the optimization of errors and depth
OLSQ-GA [49]	Optimal approach	Number of gates / circuit depth	SWAP absorption for fewer additional gates, i.e., incorporating SWAPs into other gates without additional cost	Unawareness of gate errors
Work in [6]	Optimal approach	Circuit depth	Integer Linear Programming (ILP) formulation for the mapping problem	Unawareness of gate errors
Qmap [27]	Optimal + Heuristic	Number of gates / circuit latency	Considering actual gate duration to reduce circuit latency; MOVE operations for fewer additional gates	Unawareness of gate errors
MUQUT [7]	Optimal + Heuristic	Number of gates / circuit depth / fidelity	Integer Linear Programming (ILP) formulation for the mapping problem; sliding the initial mapping over the chip for higher fidelity	Incompatible with non-grid architecture

Striking the balance

Densely-packed, adjacent qubits for minimal SWAP operations

Far qubits to minimize local cross-talk



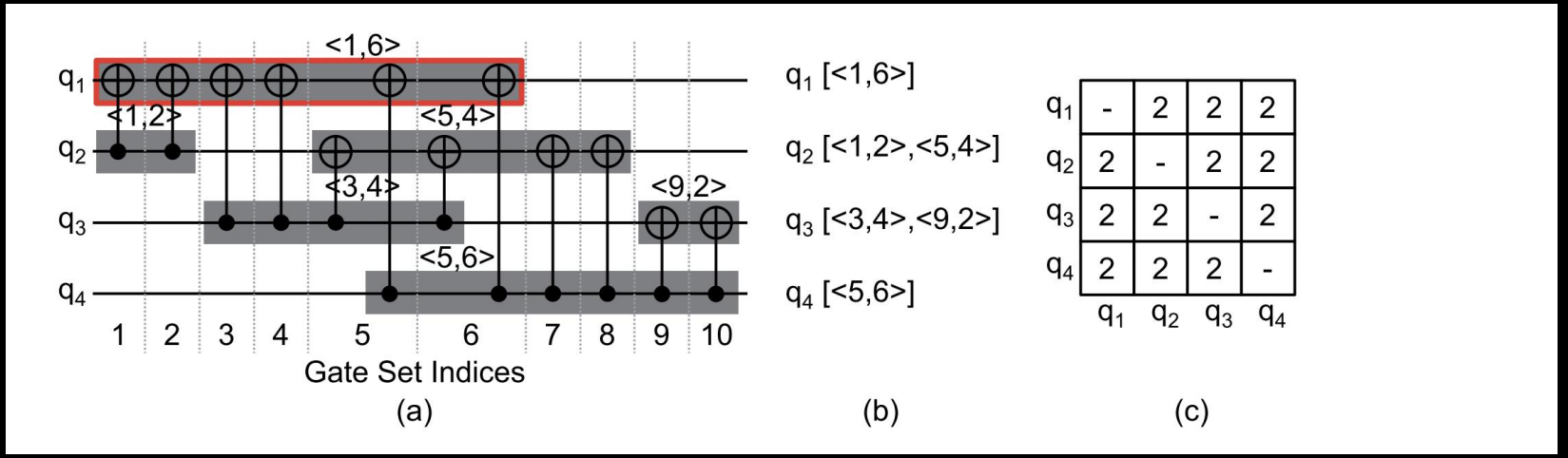
Preliminaries

- Our platform is **fully universal** and independent of both the **quantum circuit** and **physical hardware**
- This includes arbitrarily multi-qubit gates, arbitrary quantum processor topologies, and arbitrary errors (coherent, Pauli-twirled, correlated, etc.)
- Only assumption: the capability for randomized benchmarking and simultaneous randomized benchmarking on the processor
- User may input processor topology graph or utilize default 2D/3D rectangular lattice qubit arrangements

  + Nearest-neighbor connectivities

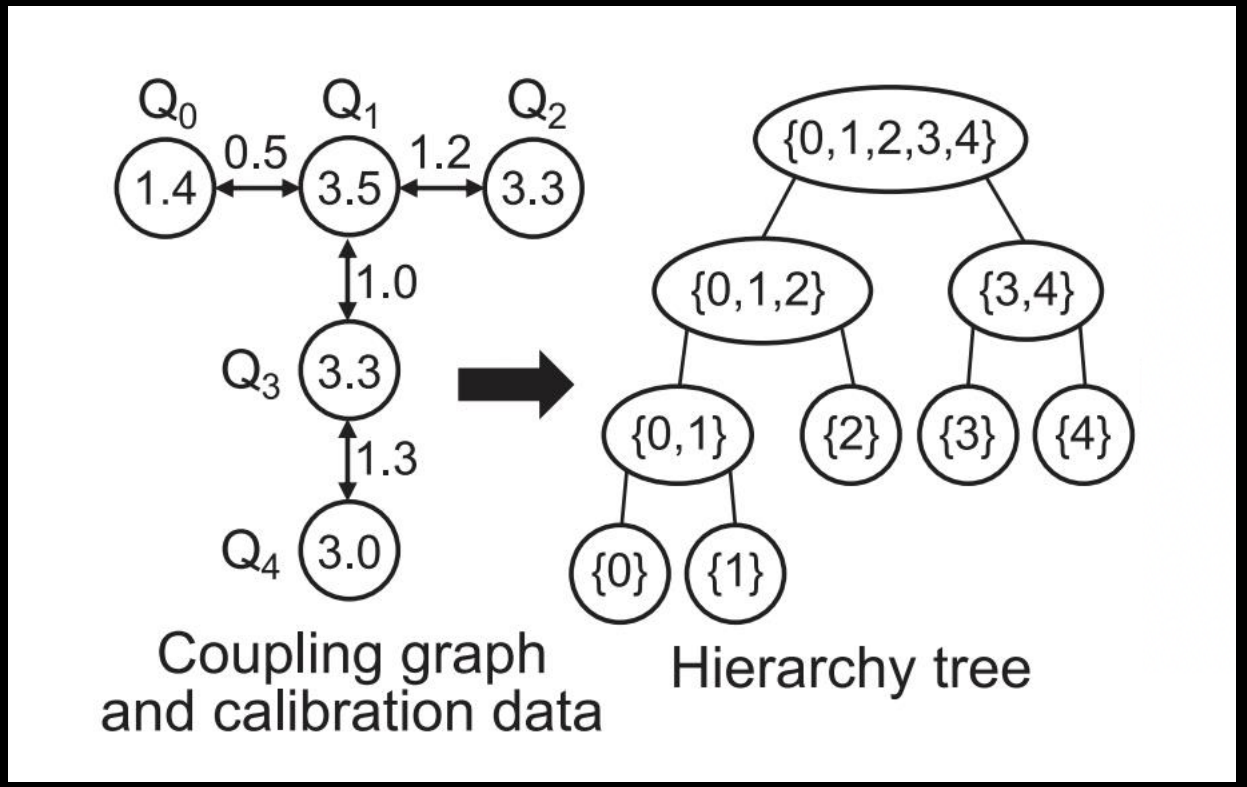
Methods: Algorithm 1

- From the quantum circuit, we can construct an involvement list and coupling strength matrix
- Involvement list: represents the time-series parallel operations of multi-qubit gates
- Coupling strength matrix: number of CNOT operations between every pair of qubits



Methods: Algorithm 2

- Construction of a “hierarchy tree” consisting of qubit “communities” depending on crosstalk hotspots
- Finds the most crosstalk-robust subgraph for mapping m qubits, with $0 \leq m \leq Processor\ size$
- Optimal subgraphs (communities) are formed through ground-up merging

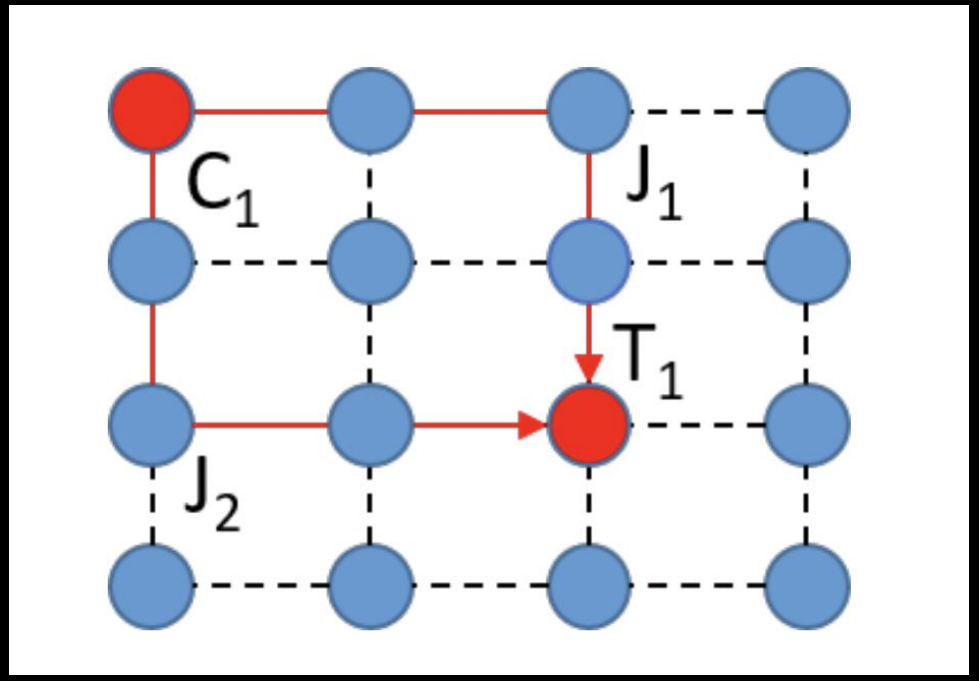


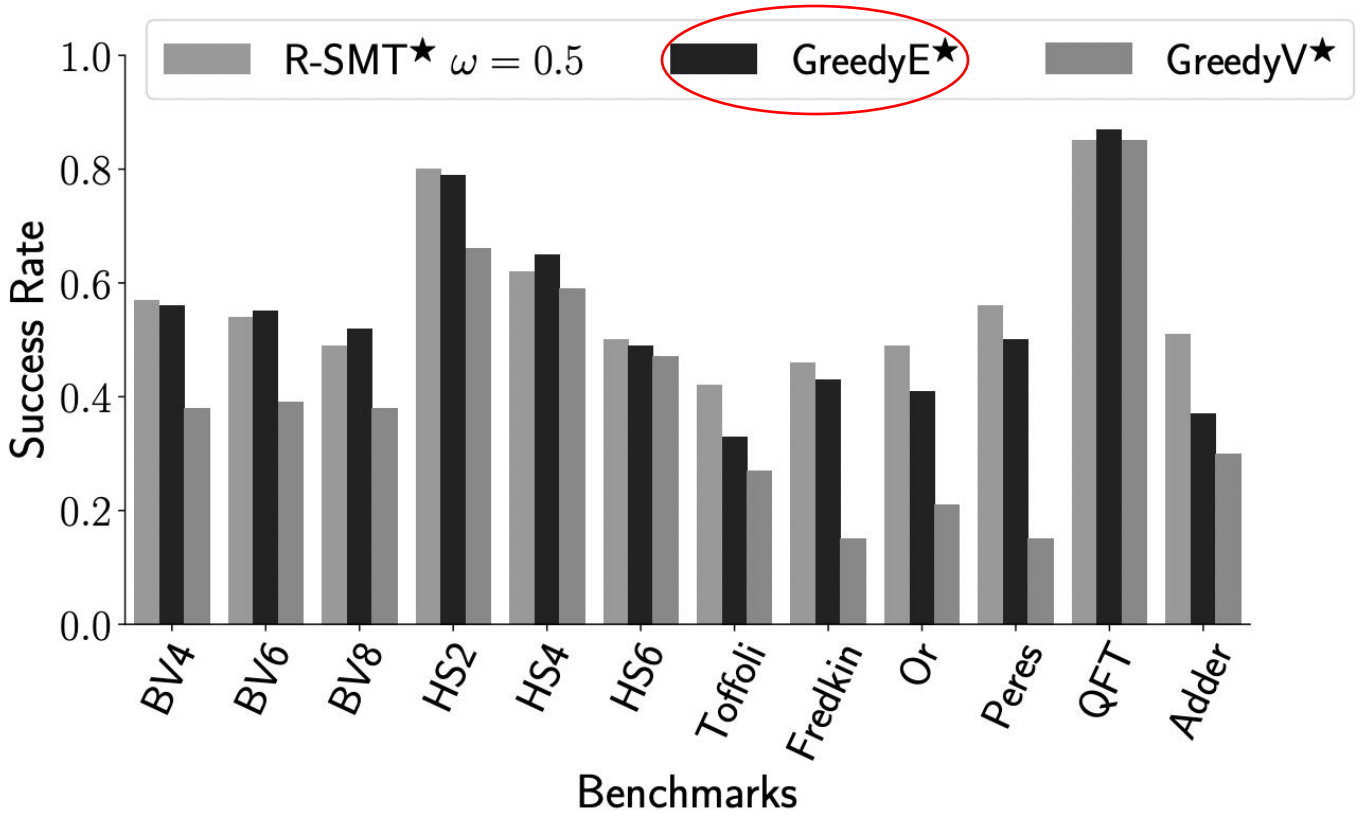
Methods: Algorithm 3

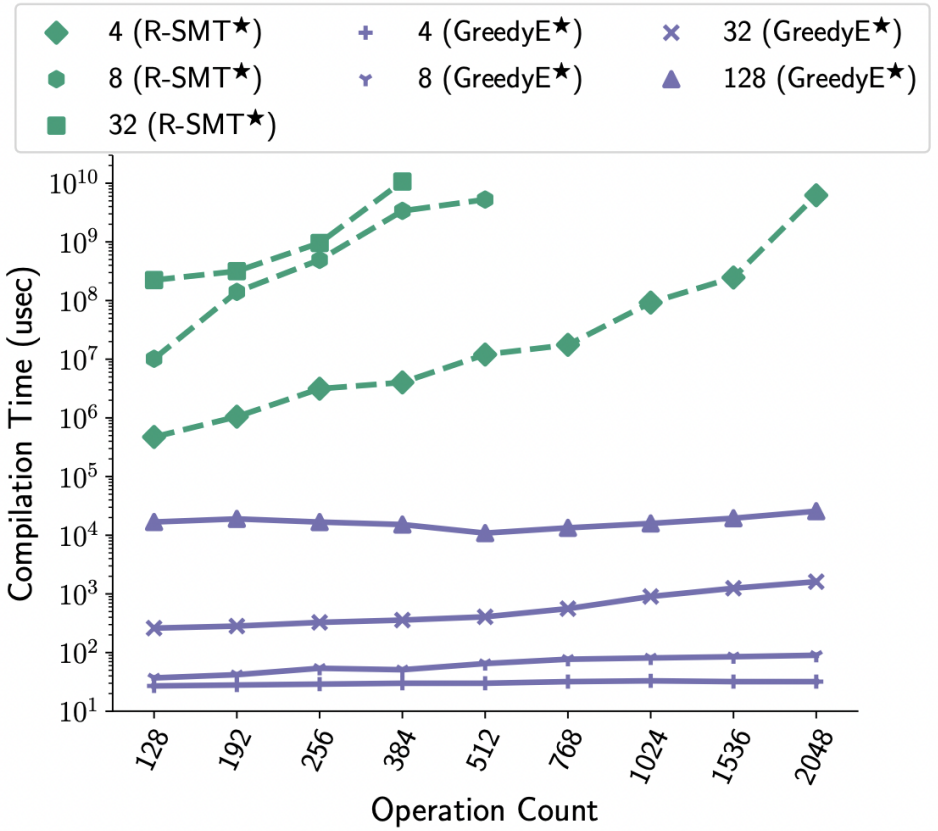
- Within optimal communities: greedy algorithm to organize qubit pairs based on increasing two-qubit gate frequency at gates with higher fidelity
- May use SWAP gates to route around lower fidelity two-qubit gates or from far away crosstalk-mitigation nodes
- Trade-off between SWAP chaining

from far away qubit vs low fidelity

CNOT mirrors crosstalk consideration

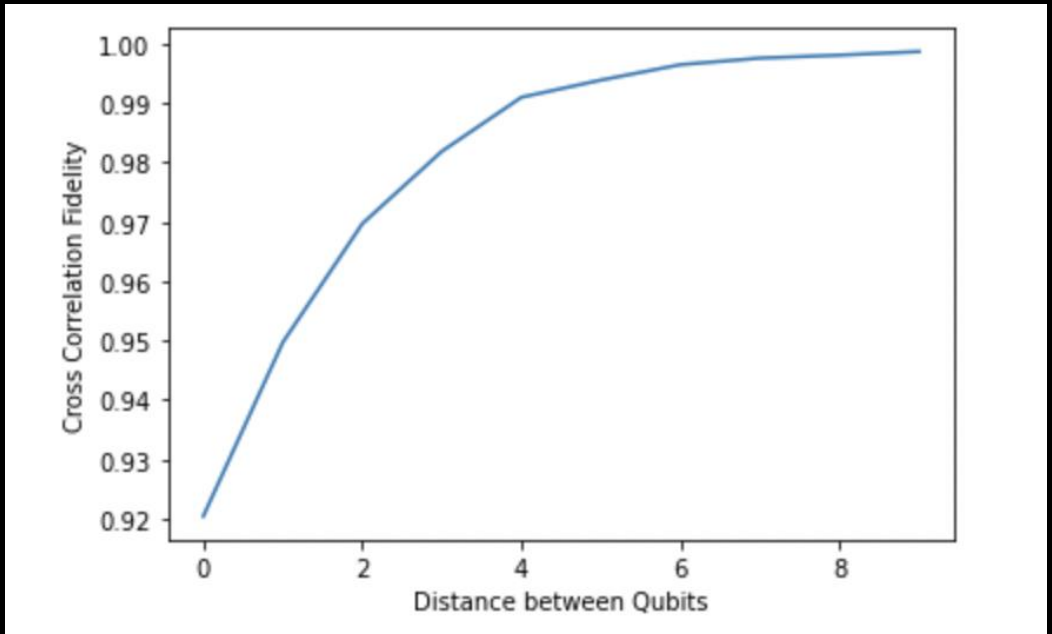



R-SMT★ $\omega = 0.5$
GreedyE★
GreedyV★
Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
BV4
BV6
BV8
HS2
HS4
HS6
Toffoli
Fredkin
Or
Peres
QFT
Adder
Benchmarks


4 (R-SMT★)
8 (R-SMT★)
32 (R-SMT★)
4 (GreedyE★)
8 (GreedyE★)
32 (GreedyE★)
128 (GreedyE★)
Compilation Time (usec)
10^{10}
10^{9}
10^{8}
10^{7}
10^{6}
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
128
192
256
384
512
768
1024
1536
2048
Operation Count

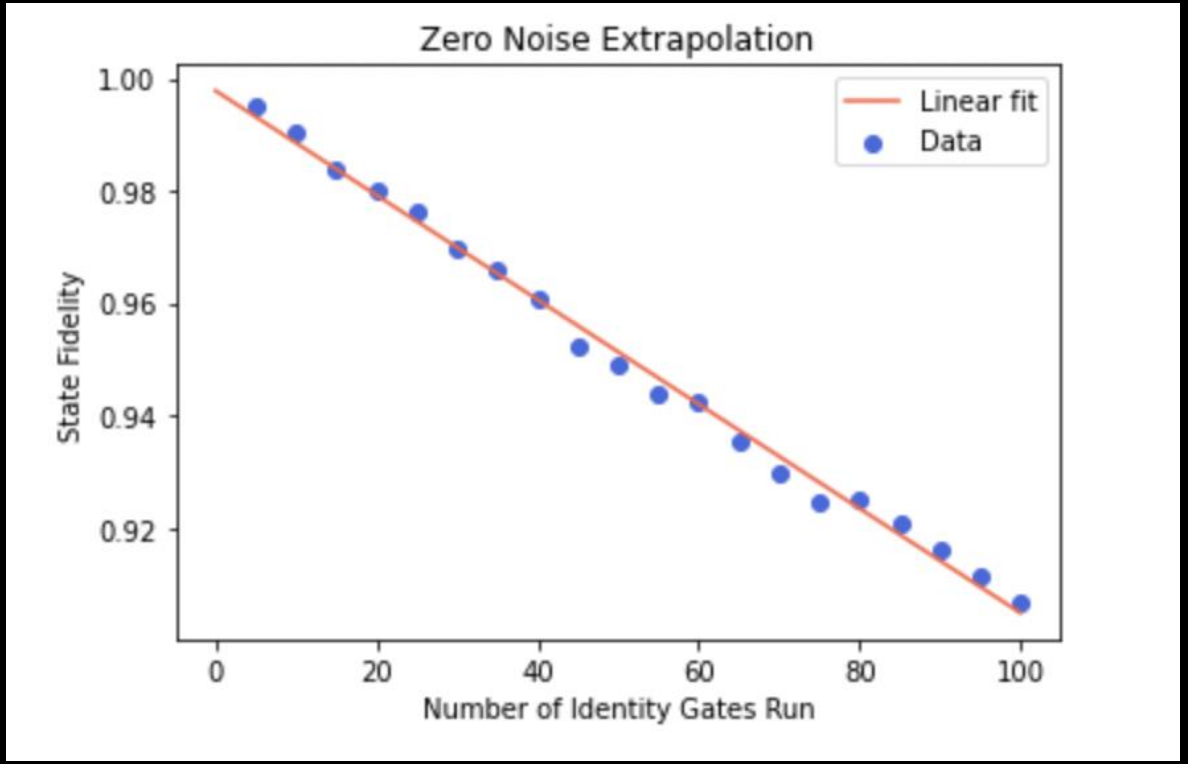
Noise Modelling

- We utilize the Kraus Operator formalism for Noise Channels
- In particular, we experimented with correlation noise with XX and YY Kraus operators corresponding to leaked Rabi Oscillations in trapped ions
- Making this dependent on distance between ions, we characterized cross correlation



Noise Modelling (cont.)

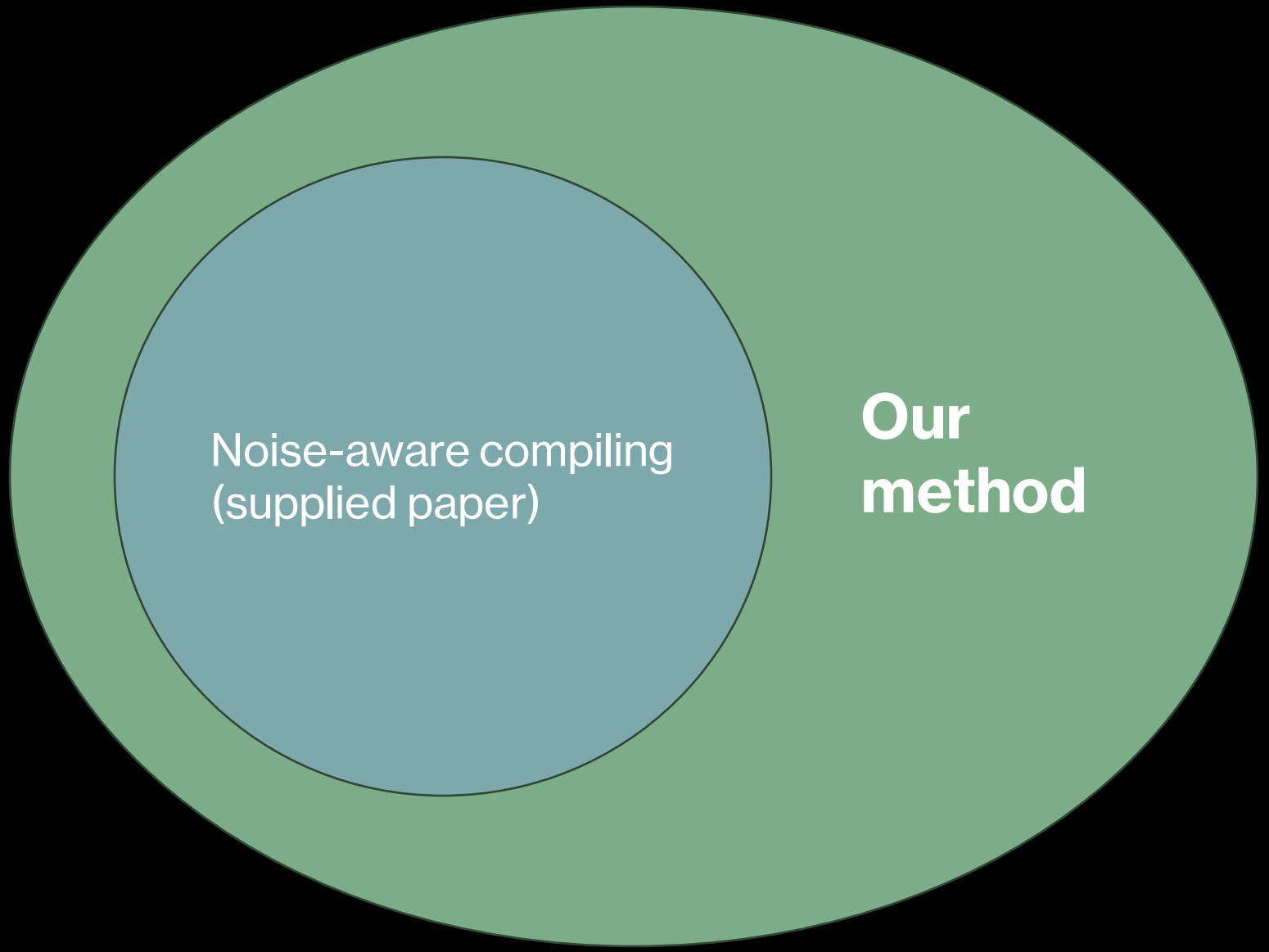
- Zero error extrapolation for single qubit fidelity characterization
- Running different of identity gates to back extrapolate zero error fidelity for a single qubit



Experimental Results

- We tested our method on a 5-qubit Bernstein-Vazirani on an 11-qubit processor with 2D planar connectivity
- Our approach resulted in a 98.9% average fidelity (vs 98.4% using simple noise-aware compilation)

  ~30% increase fidelity


Noise-aware compiling
(supplied paper)
Our
method

Scalability analysis

- Simultaneous randomized benchmarking (SRB) has negligible overhead
  + Fidelity evaluations only occur ~once every few days
  + Construction of the hierarchy tree takes $O(n^4)$ time but also needs to only be constructed once
  + Edge-based greedy optimization is at most $O(n)$, i.e. not a dominant term
  + Classical co-computation compiler overhead is insignificant
    - No additional resources to active gate implementations

Applications

- No platform natively supports yet our compilation technique
- Allows for new compilation schemes for every architecture via Braket backend
- E.g. IonQ's high level QIS suite has error mitigation, elementary noise-adaptive mapping, post-selection, native gate compilation, and gate conjugation techniques

  + Our method implements a novel, circuit-aware structure via Braket

  + Gaussian correlated noise are a particular challenge in trapped ions

Thank you to MIT-CQE & our AWS mentor!