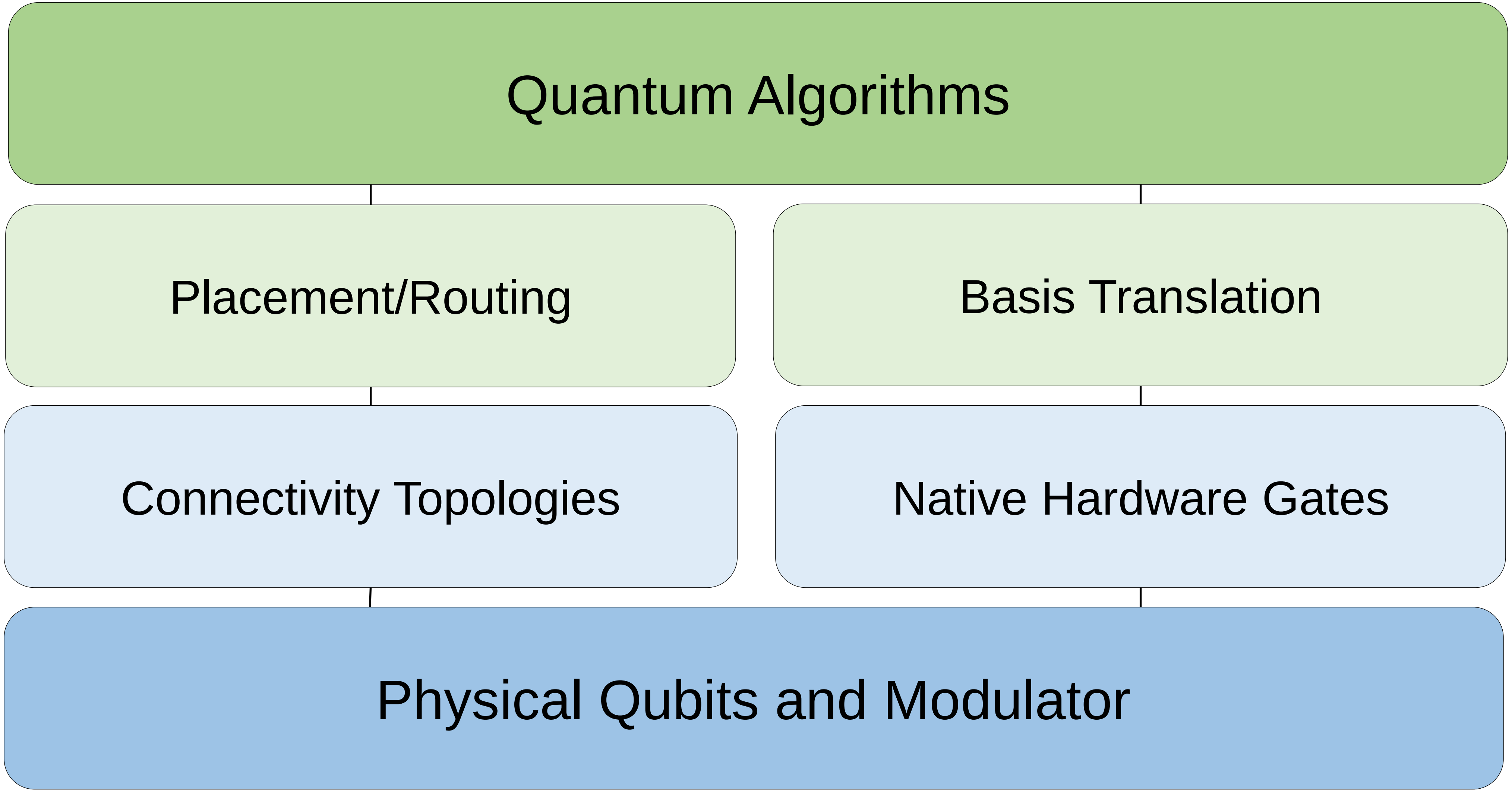
Quantum Circuit Decomposition and Routing Collaborative Design

Evan McKinney[†],
M. Hatridge[§], A.K. Jones[†]

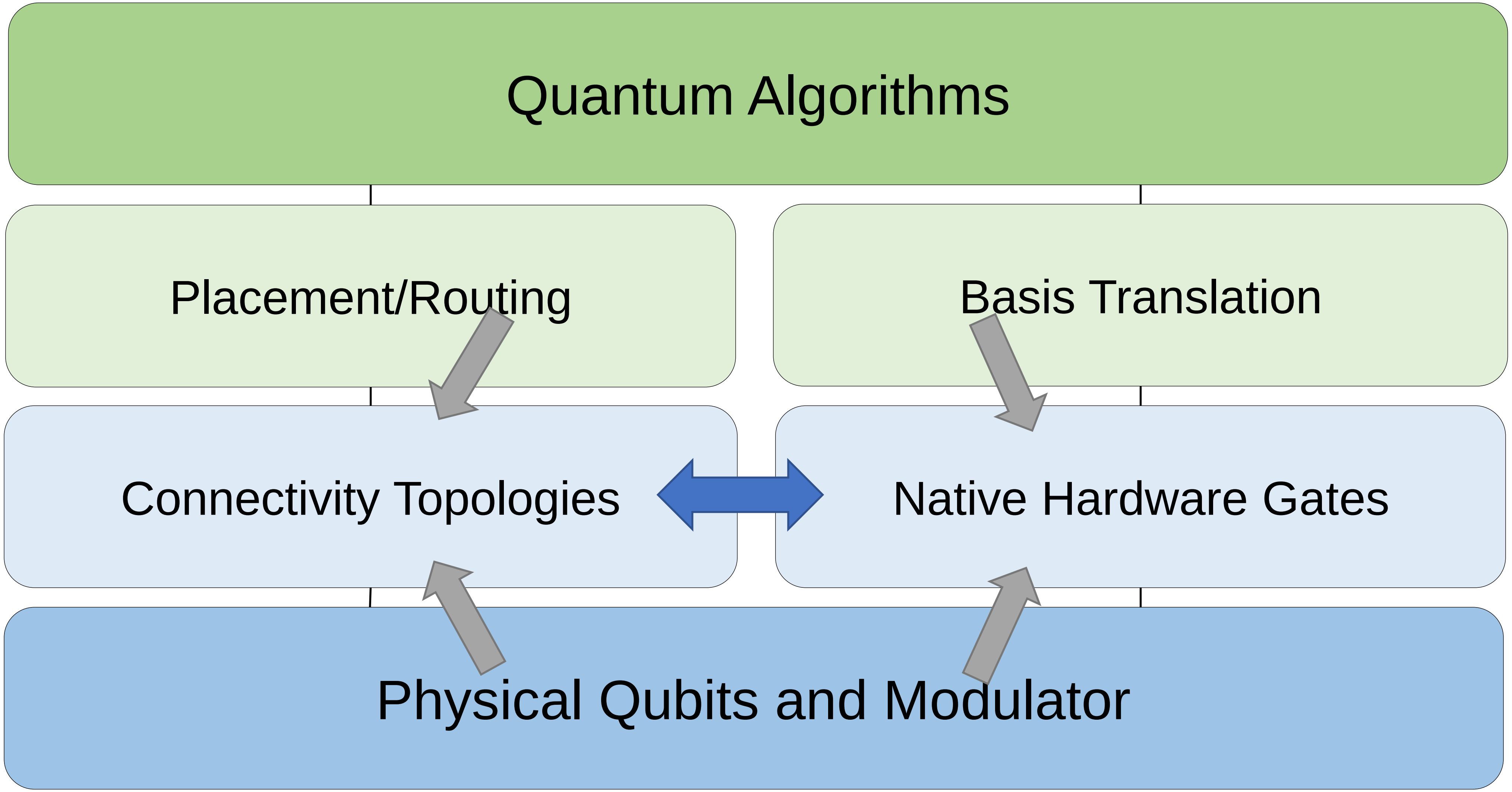
[†]Department of Electrical and Computer Engineering, University of Pittsburgh
[§]Department of Physics and Astronomy, University of Pittsburgh

Quantum Computer Systems (QuCS) 2023

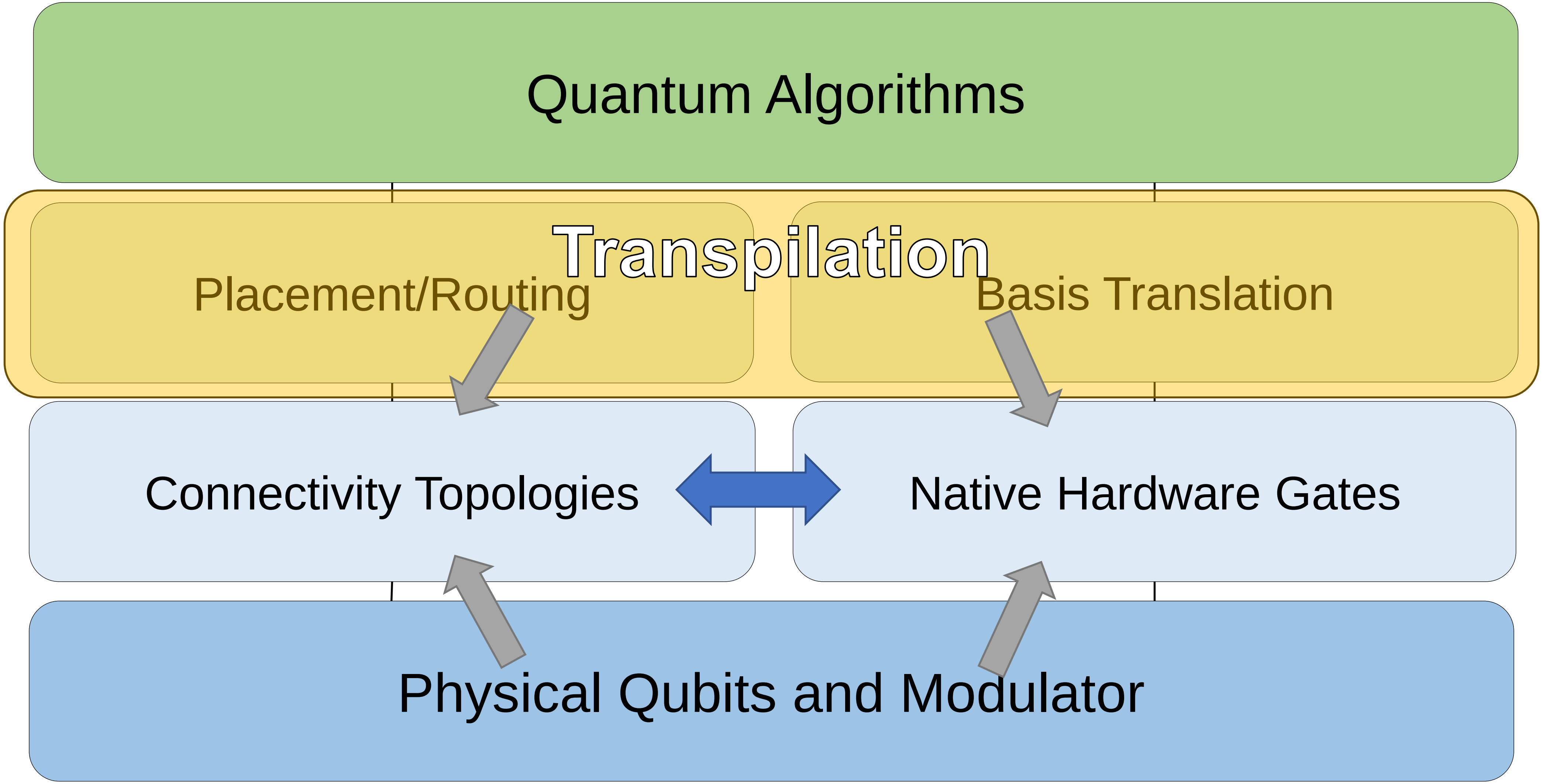
Quantum computer co-design



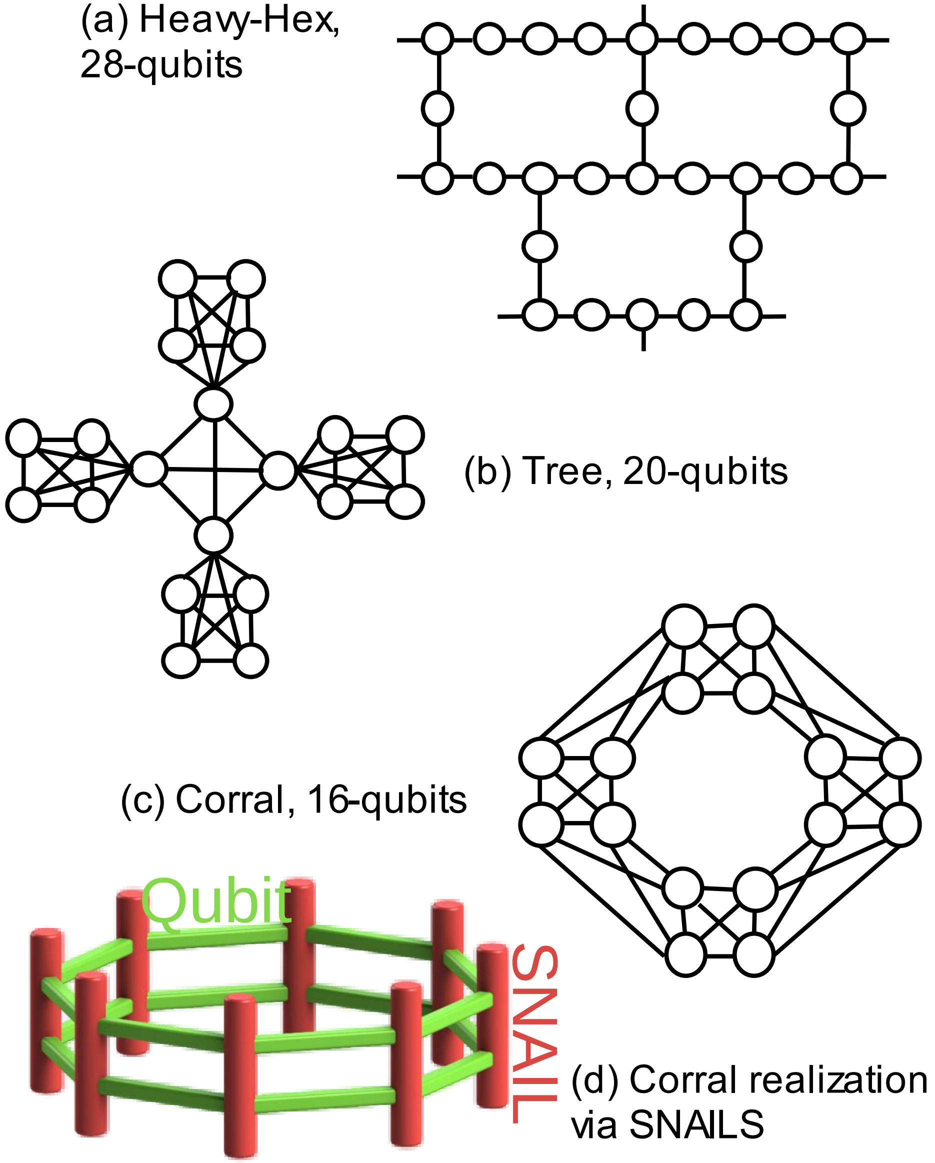
Quantum computer co-design



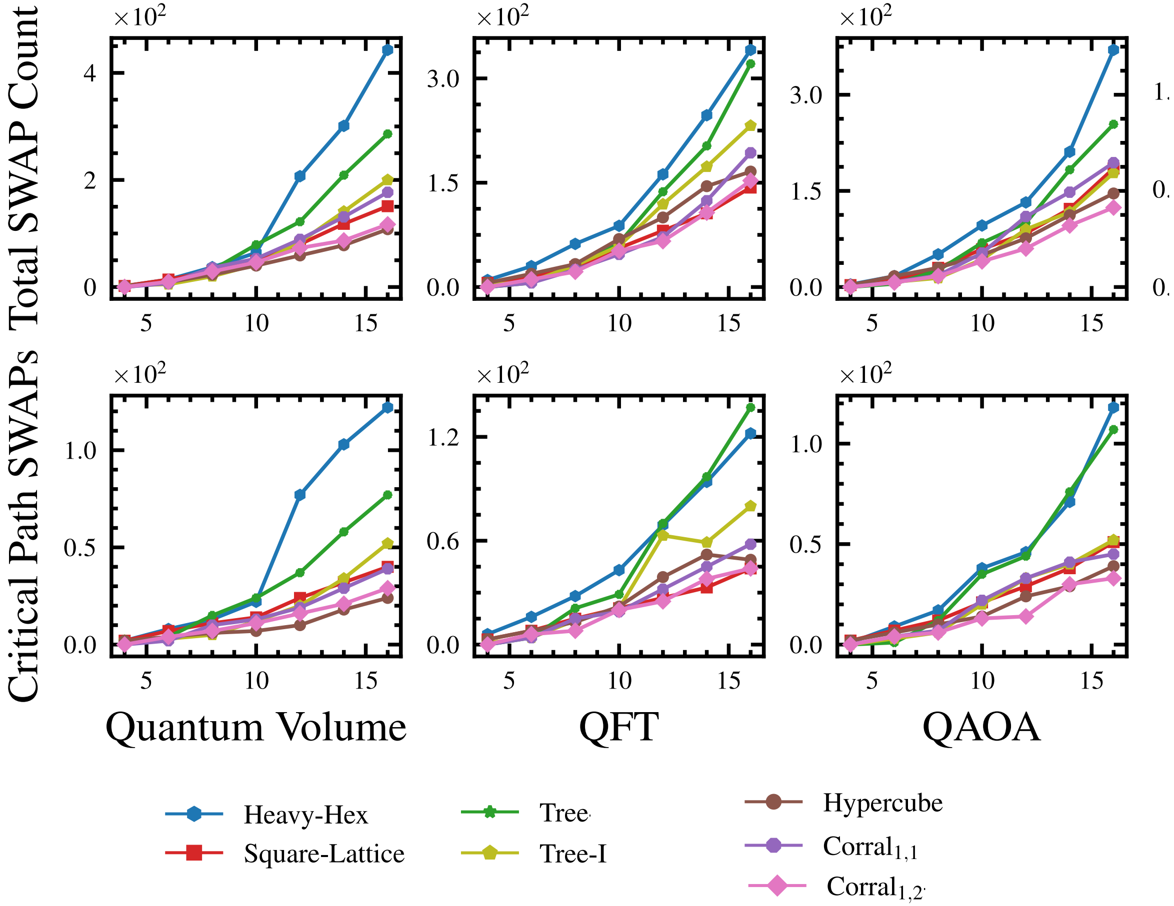
Quantum computer co-design



Topology co-design

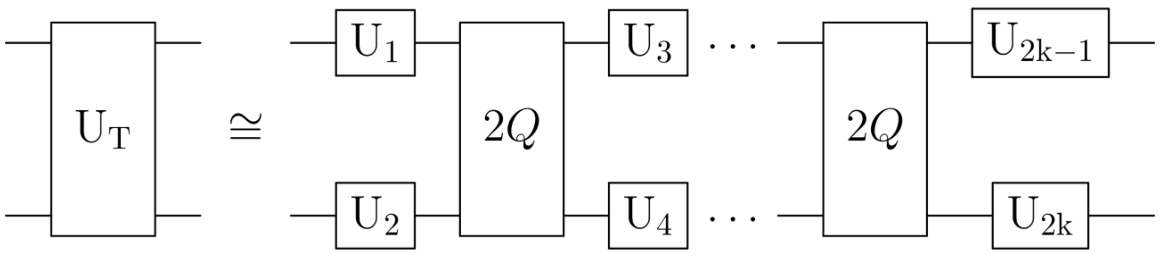


- Transpile circuits to Hatlab connectivity
- Co-design study topology networks



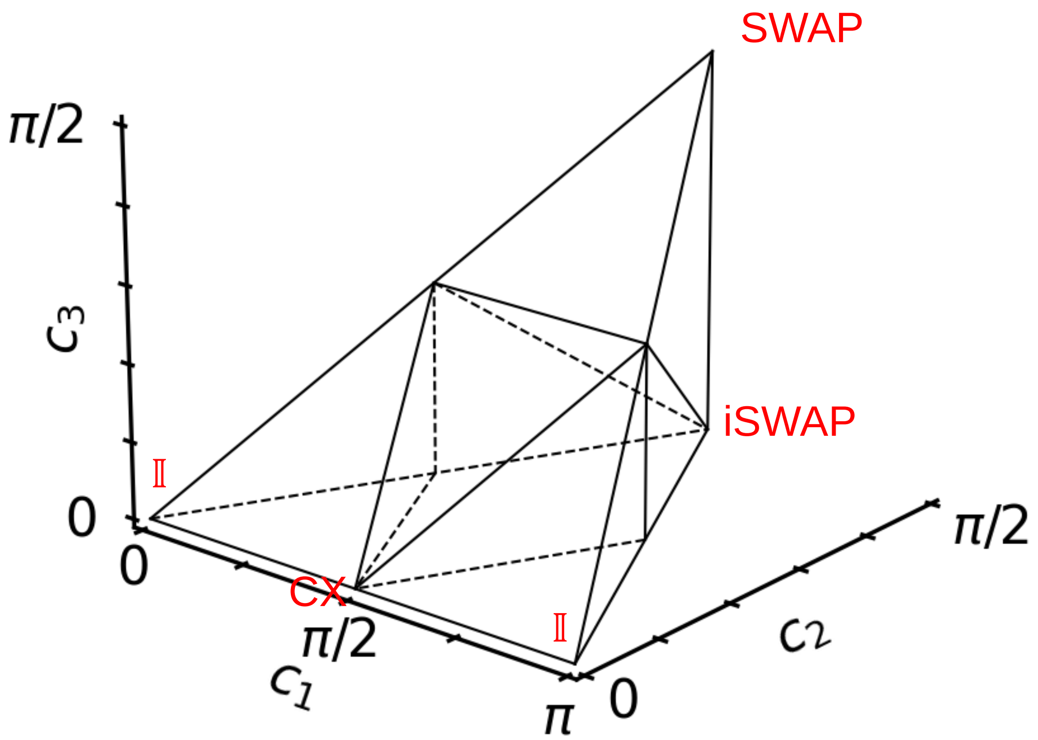
Two-qubit basis gates

- Decompose all algorithm gates into new basis using repeated applications



- An optimal basis gate *reduces overall duration*
  - Powerful gates need less applications
  - Fidelity limited by decoherence in time

- Weyl Chamber visualizes the set of all 2Q gates

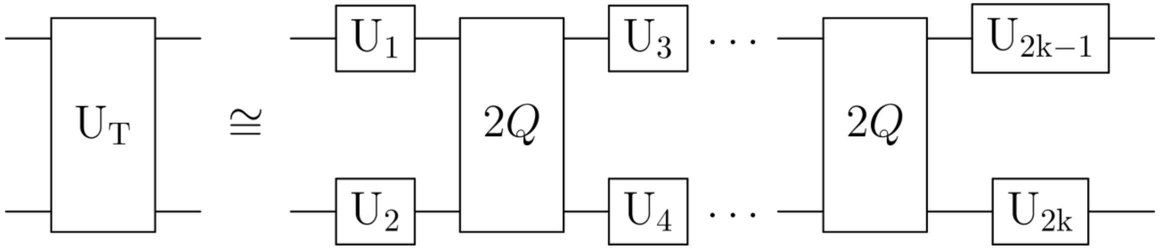


Y. Makhlin, **Quantum Info. Process. 1**, (2002)

McKinney, et al. **ISCA** (2023)

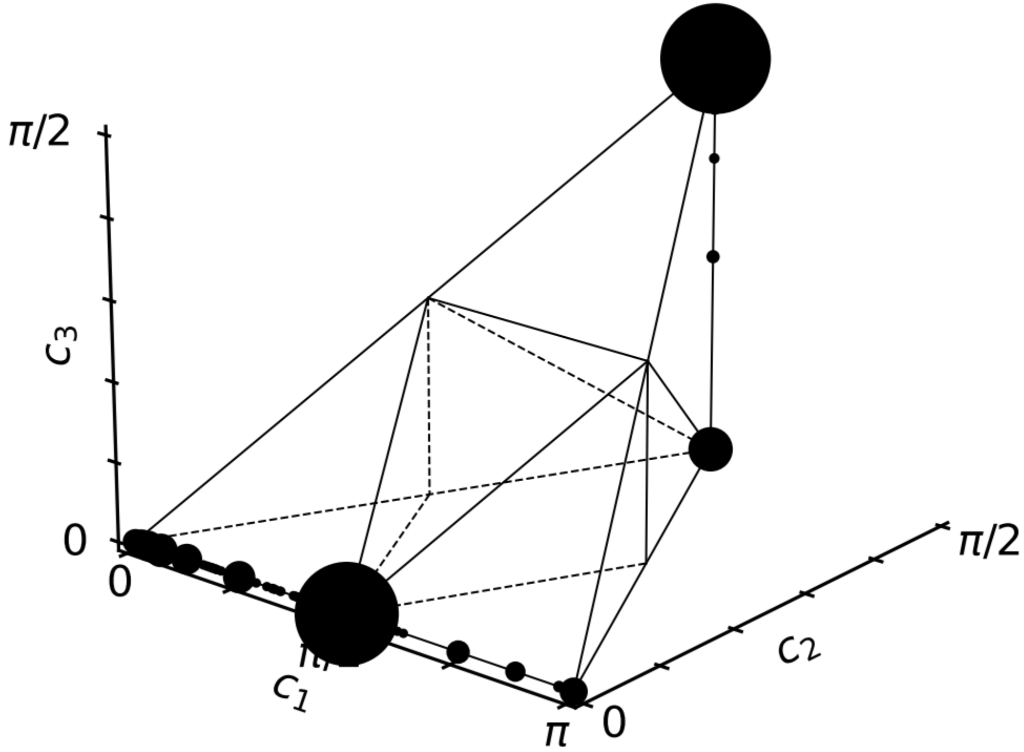
Two-qubit basis gates

- Decompose all algorithm gates into new basis using repeated applications



- An optimal basis gate *reduces overall duration*
  - Powerful gates need less applications
  - Fidelity limited by decoherence in time

- Weyl Chamber visualizes the set of all 2Q gates



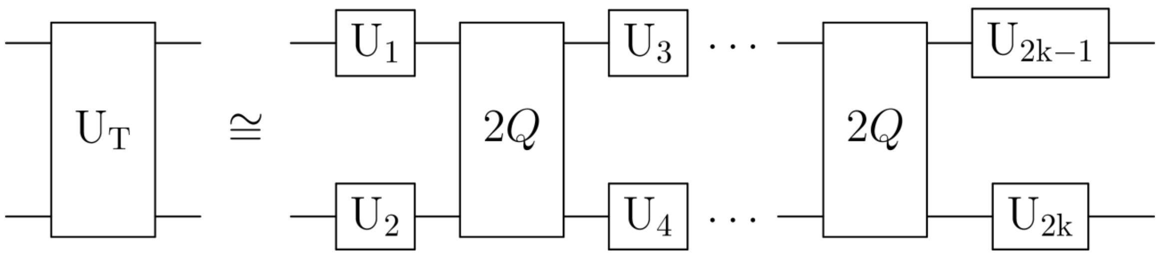
- NISQ algorithms dominated by CX and SWAP gates

Y. Makhlin, **Quantum Info. Process. 1**, (2002)

McKinney, et al. **ISCA** (2023)

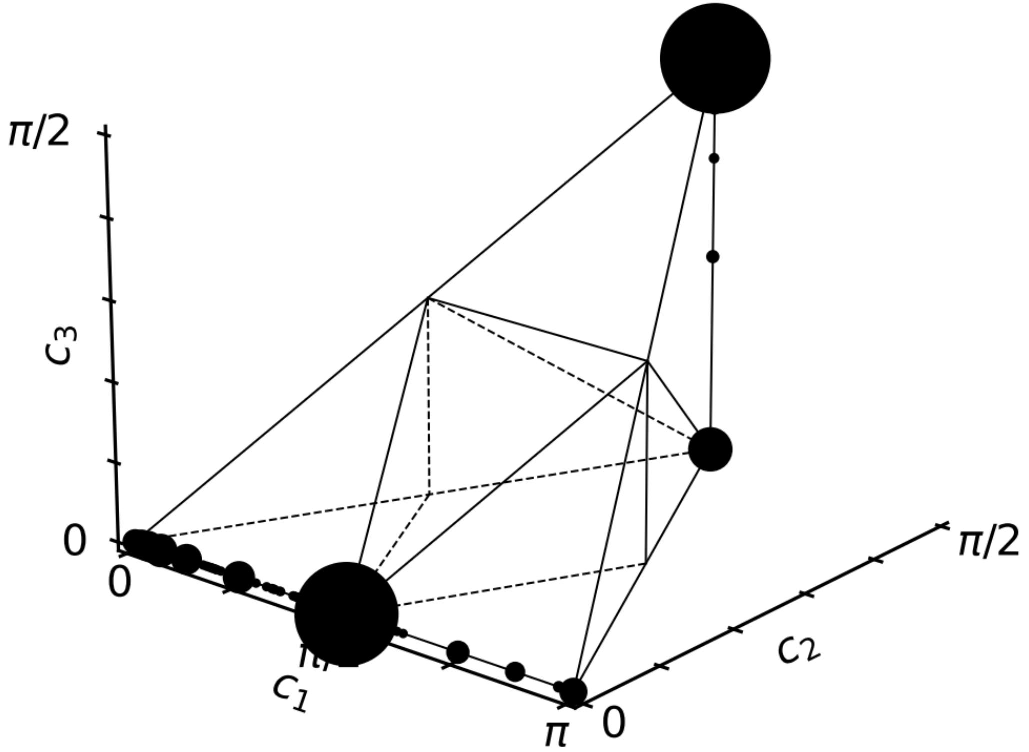
Two-qubit basis gates

- ➢ Decompose all algorithm gates into new basis using repeated applications



- ➢ An optimal basis gate *reduces overall duration*
  - ➢ Powerful gates need less applications
  - ➢ Fidelity limited by decoherence in time

- ➢ Weyl Chamber visualizes the set of all 2Q gates



- ➢ NISQ algorithms dominated by CX and SWAP gates

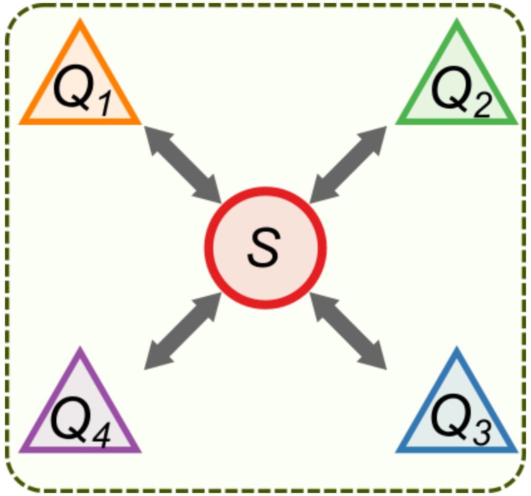
- ➢ **Goal: Use both decomposition efficiency and hardware latency = overall duration**

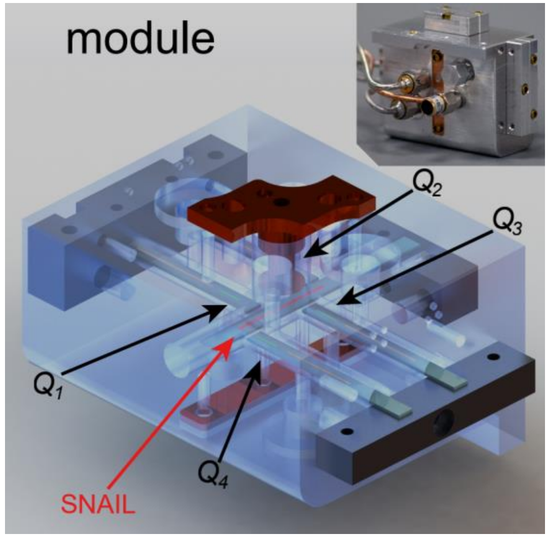
Y. Makhlin, **Quantum Info. Process. 1**, (2002)

McKinney, et al. **ISCA** (2023)

Conversion/Gain candidate basis gates

Four qubit SNAIL-based quantum module





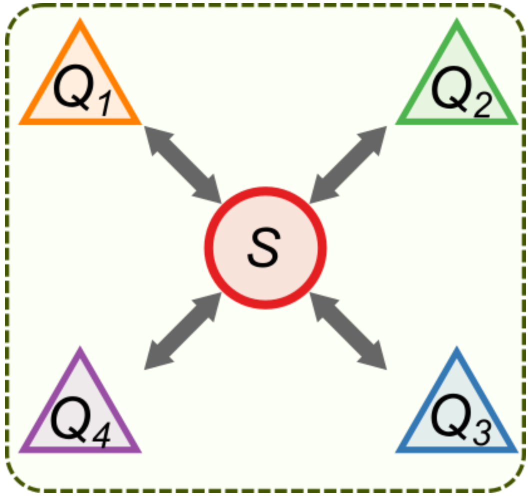
- **Engineerable interactions yields a basis gate design-space**

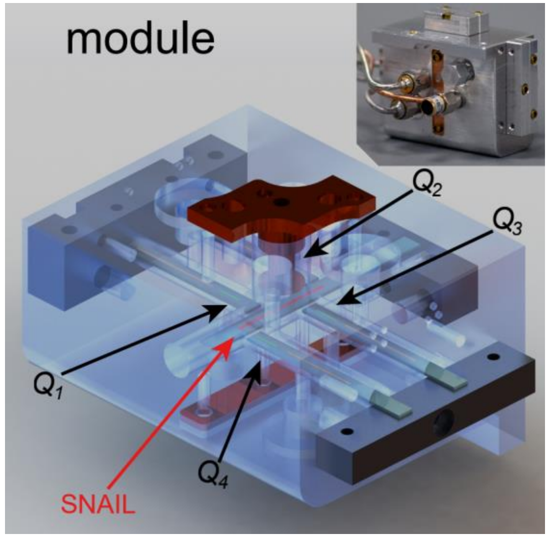
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger)$$

Xia, et al. **arXiv:2306.10162** (2023)
Zhou, et al. **npj Quantum Inf 9**, 54 (2023)

Conversion/Gain candidate basis gates

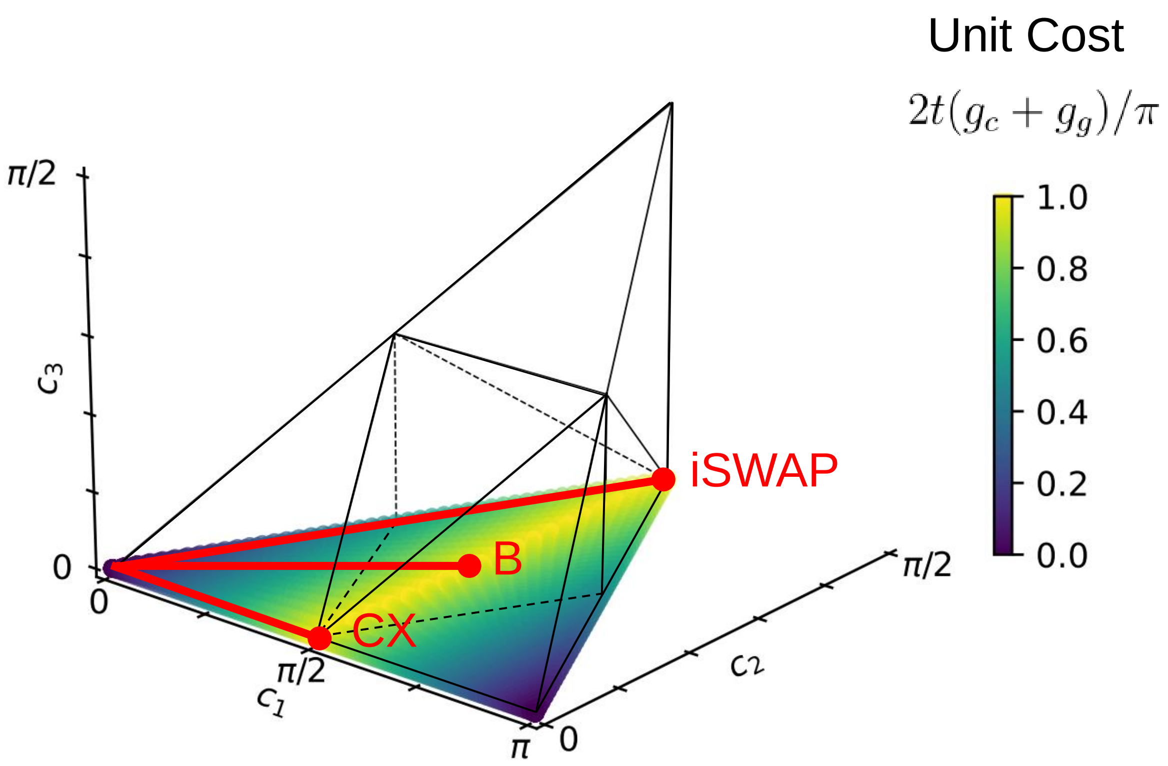
Four qubit SNAIL-based quantum module





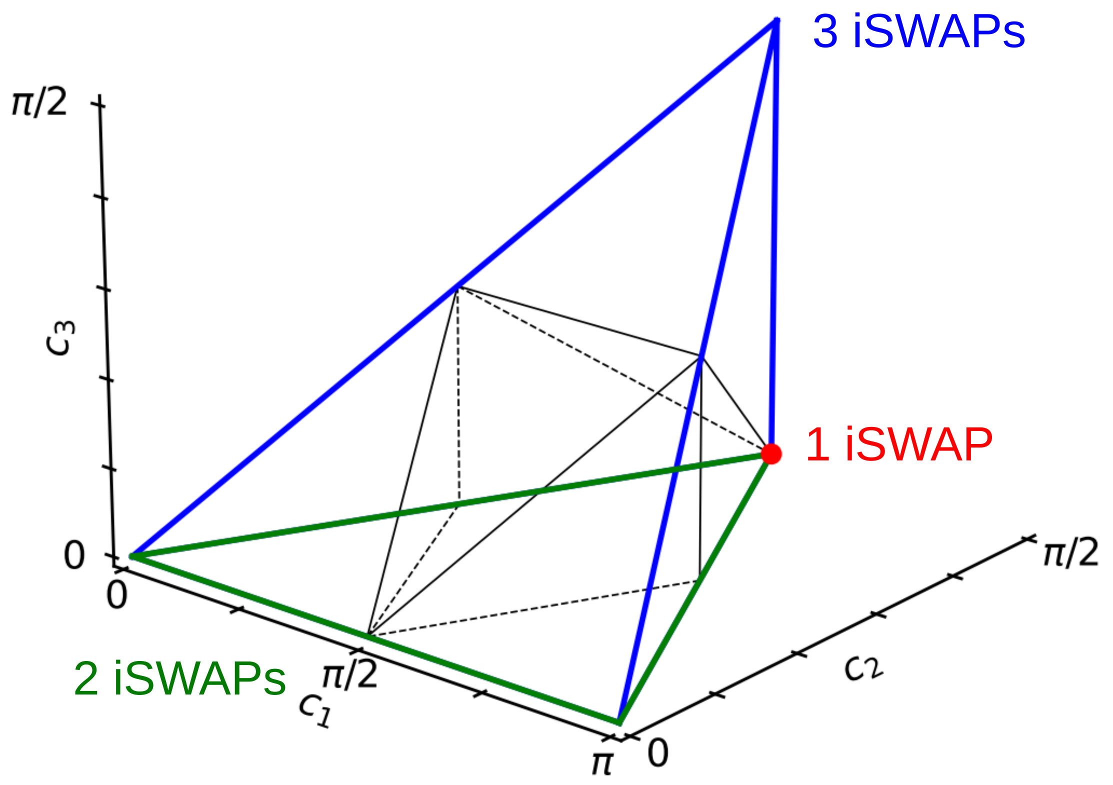
- **Engineerable interactions yields a basis gate design-space**

$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger)$$



Xia, et al. **arXiv:2306.10162** (2023)
Zhou, et al. **npj Quantum Inf 9**, 54 (2023)

Basis coverage volumes



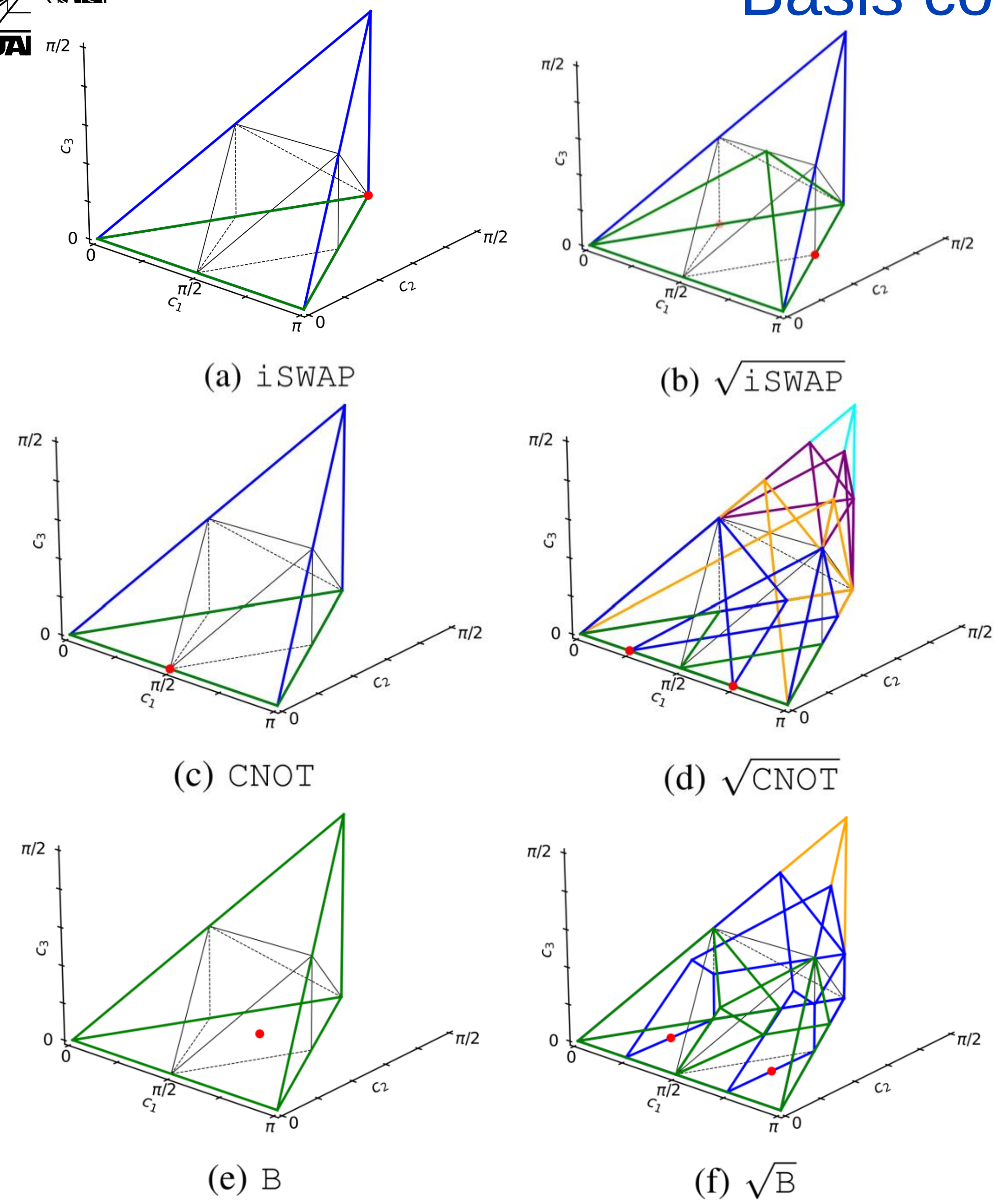
- **Monodromy polytopes** finds minimum gate applications for any 2Q target gate
- A single gate is locally equivalent to itself
- SWAP is the most expensive target

Target\Basis	iSWAP
CNOT	2.0
SWAP	3.0
Haar	3.0

Peterson, et al. ***Quantum* 4** (2020)

McKinney, et al. **ISCA** (2023)

Basis coverage volumes



- **Monodromy polytopes** finds minimum gate applications for any 2Q target gate
- A single gate is locally equivalent to itself
- SWAP is the most expensive target

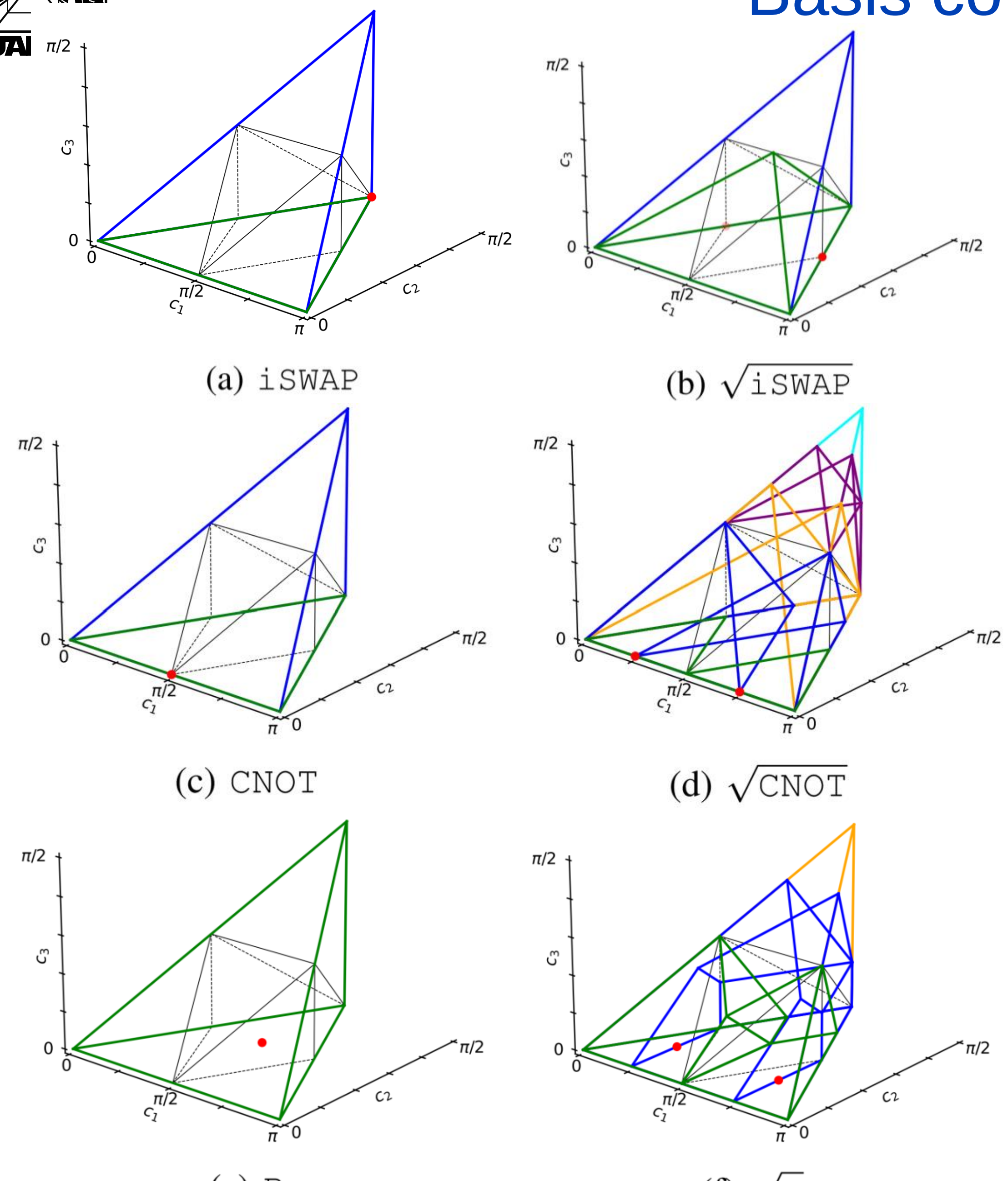
Decomposition *gate count* costs

Target\Basis	iSWAP	$\sqrt{\text{iSWAP}}$	CX	$\sqrt{\text{CX}}$	B	$\sqrt{\text{B}}$
CNOT	2.0	2.0	1.0	2.0	2.0	2.0
SWAP	3.0	3.0	3.0	6.0	2.0	4.0
Haar	3.0	2.2	3.0	3.5	2.0	3.1

Peterson, et al. ***Quantum* 4** (2020)

McKinney, et al. **ISCA** (2023)

Basis coverage volumes



(a) iSWAP (b) $\sqrt{\text{iSWAP}}$ (c) CNOT (d) $\sqrt{\text{CNOT}}$ (e) B (f) $\sqrt{\text{B}}$

- **Monodromy polytopes** finds minimum gate applications for any 2Q target gate
- A single gate is locally equivalent to itself
- SWAP is the most expensive target

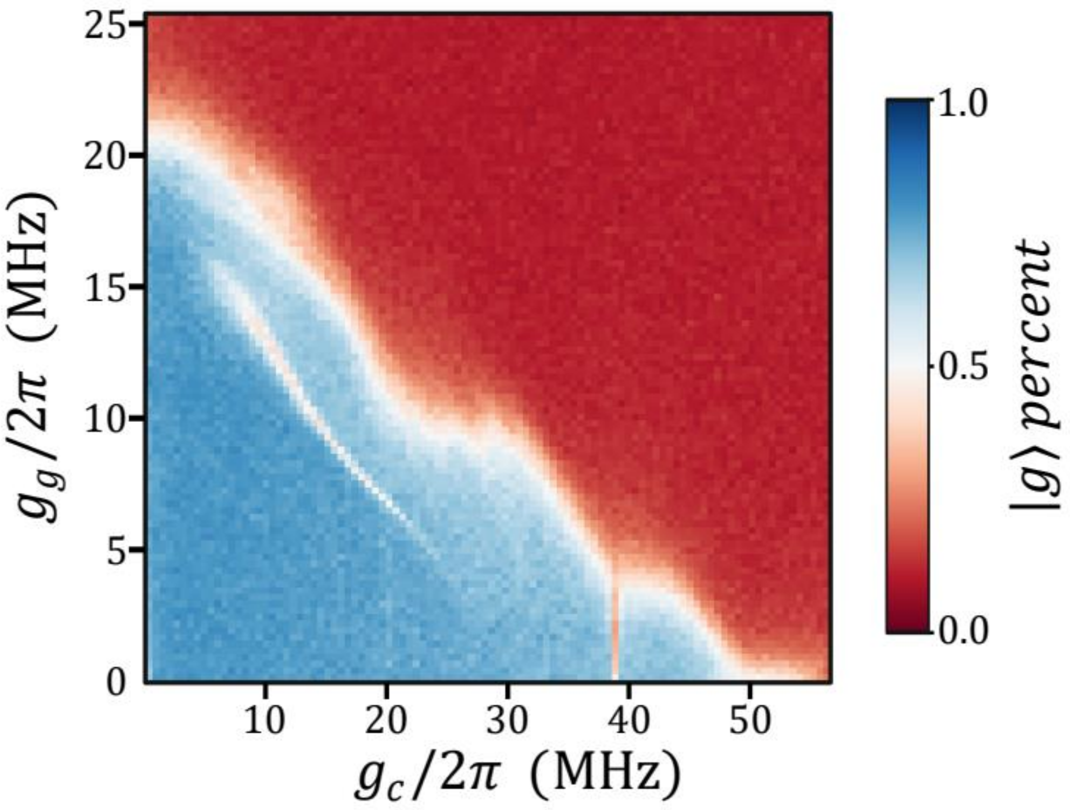
Decomposition *gate count* costs

Target\Basis	iSWAP	$\sqrt{\text{iSWAP}}$	CX	$\sqrt{\text{CX}}$	B	$\sqrt{\text{B}}$
CNOT	2.0	2.0	1.0	2.0	2.0	2.0
SWAP	3.0	3.0	3.0	6.0	2.0	4.0
Haar	3.0	2.2	3.0	3.5	2.0	3.1

Peterson, et al. ***Quantum* 4** (2020)

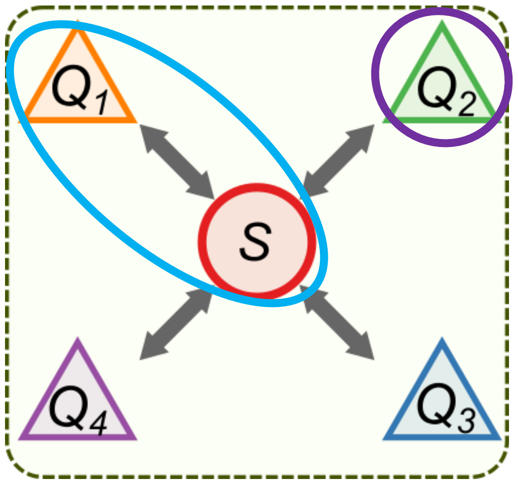
McKinney, et al. **ISCA** (2023)

Hardware speed limits



Limitation of SNAIL when driving both gain and conversion

Module

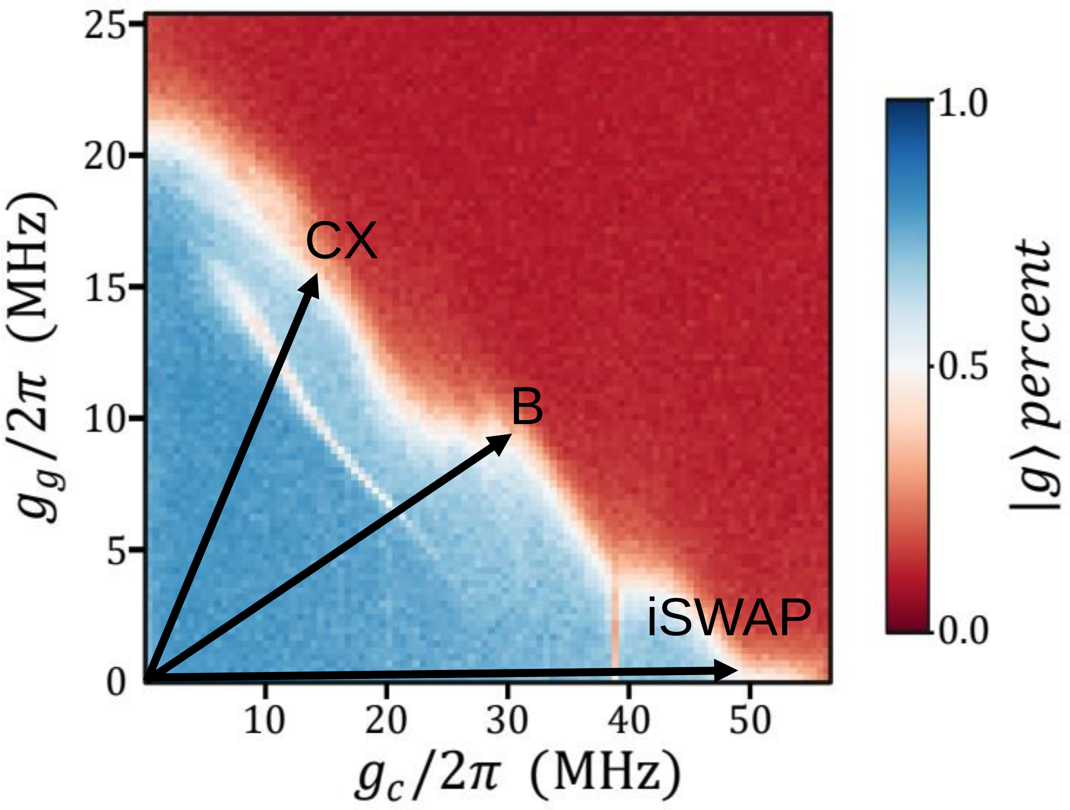


Drives applied between SNAIL and qubit

Measure second qubit to witness SNAIL breakpoint

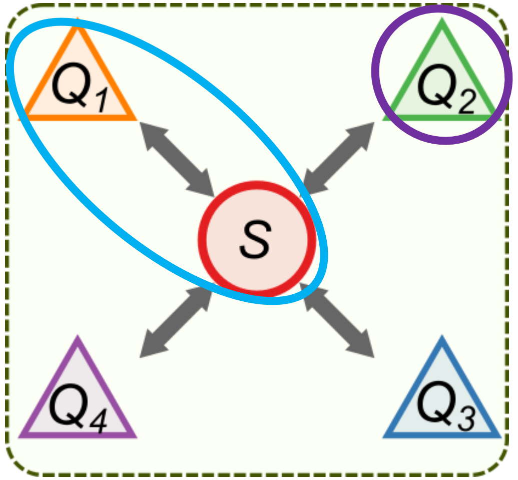
Zhou, et al. **npj Quantum Inf 9**, 54 (2023).

Hardware speed limits



Limitation of SNAIL when driving both gain and conversion

Module

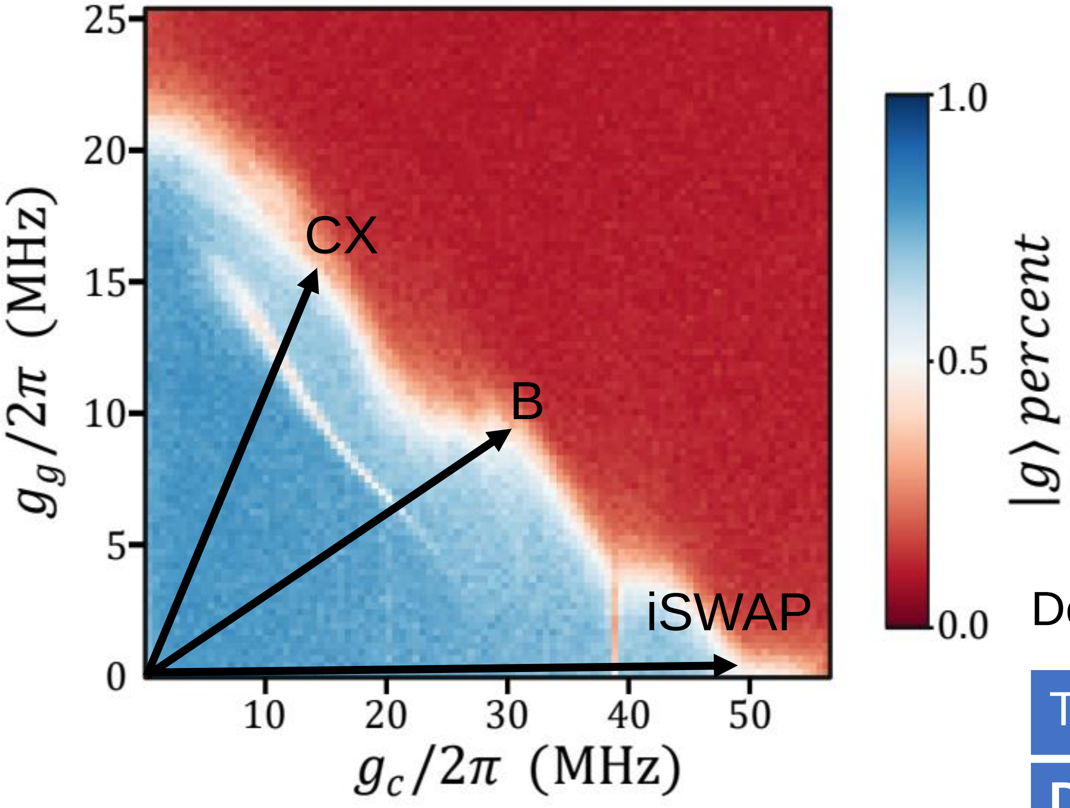


Drives applied between SNAIL and qubit

Measure second qubit to witness SNAIL breakpoint

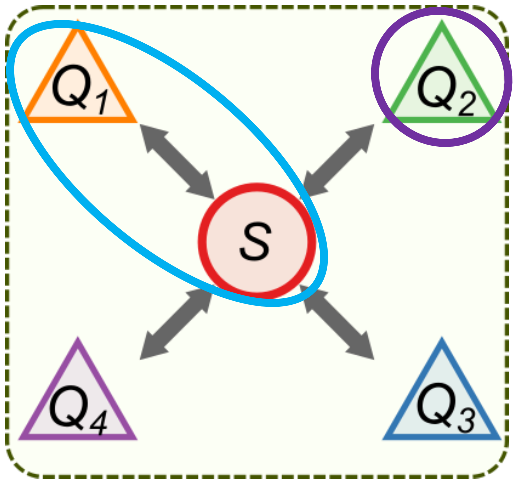
Zhou, et al. **npj Quantum Inf 9**, 54 (2023).

Hardware speed limits



Limitation of SNAIL when driving both gain and conversion

Module



Drives applied between SNAIL and qubit

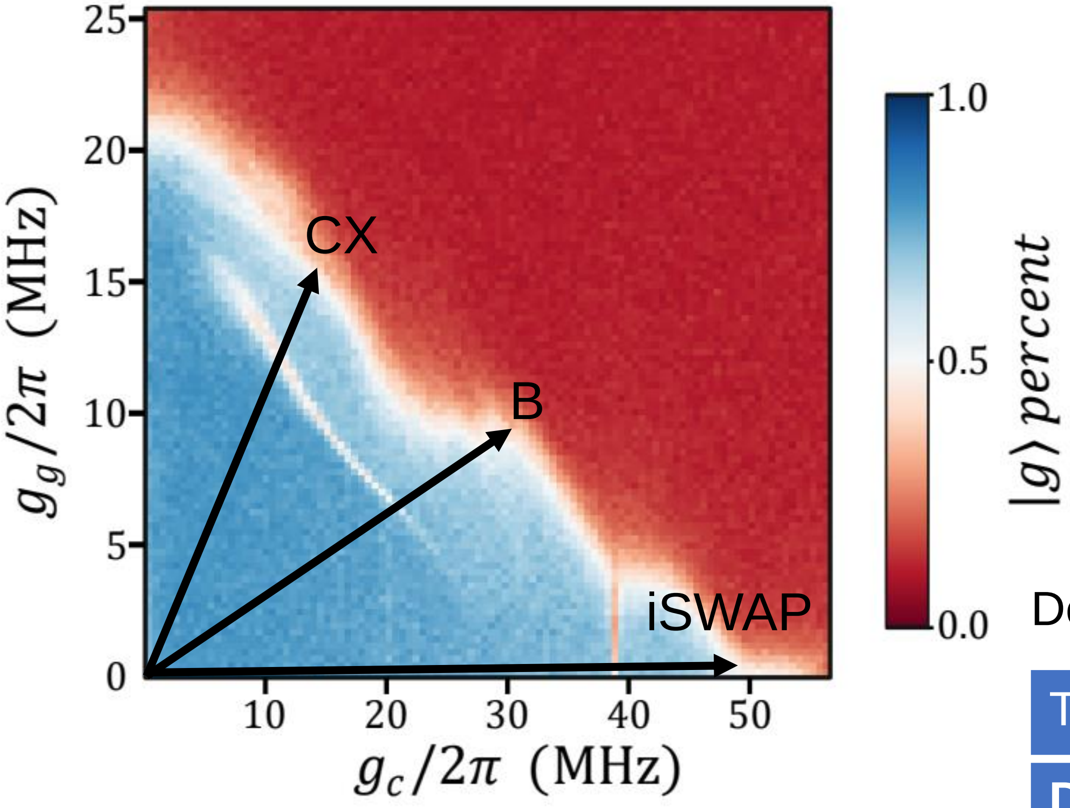
Measure second qubit to witness SNAIL breakpoint

Decomposition normalized *duration* costs

Target\Basis	iSWAP	$\sqrt{\text{iSWAP}}$	CX	$\sqrt{\text{CX}}$	B	$\sqrt{\text{B}}$
Duration	**1.0**	**0.5**	**1.8**	**0.9**	**1.4**	**0.7**
CNOT	2.0	1.0	1.8	1.8	2.8	1.4
SWAP	3.0	1.5	5.4	5.4	2.8	2.8
Haar	3.0	1.1	5.4	3.2	2.8	2.2

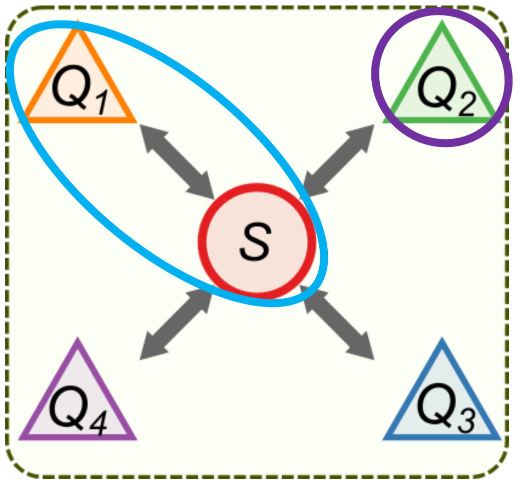
Zhou, et al. **npj Quantum Inf 9**, 54 (2023).

Hardware speed limits



Limitation of SNAIL when driving both gain and conversion

Module



Drives applied between SNAIL and qubit

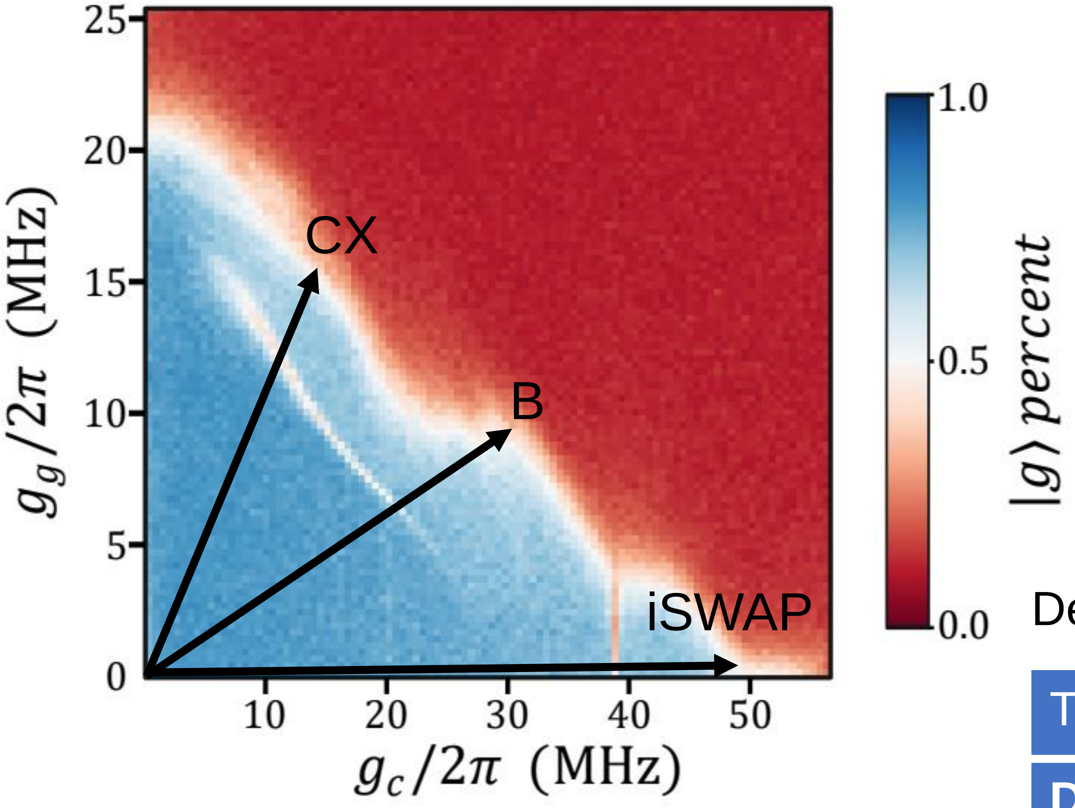
Measure second qubit to witness SNAIL breakpoint

Decomposition normalized *duration* costs

Target\Basis	iSWAP	$\sqrt{\text{iSWAP}}$	CX	$\sqrt{\text{CX}}$	B	$\sqrt{\text{B}}$
Duration	**1.0**	**0.5**	**1.8**	**0.9**	**1.4**	**0.7**
CNOT	2.0	1.0	1.8	1.8	2.8	1.4
SWAP	3.0	1.5	5.4	5.4	2.8	2.8
Haar	3.0	1.1	5.4	3.2	2.8	2.2

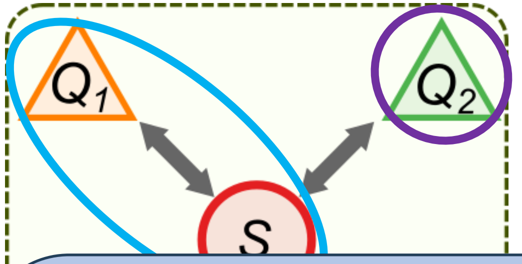
Zhou, et al. **npj Quantum Inf 9**, 54 (2023).

Hardware speed limits



Limitation of SNAIL when driving both gain and conversion

Module



Drives applied between SNAIL and qubit

ubit

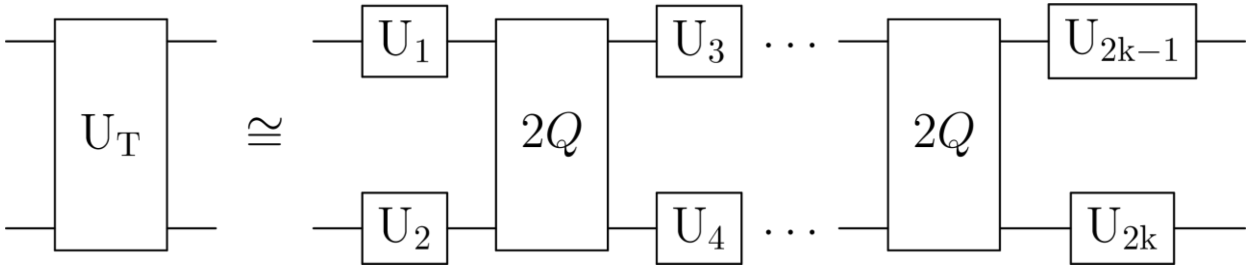
Partial pulsed gates are a good value
Why not make the pulses increasing small?

Decomposition normalized *duration* costs

Target\Basis	iSWAP	$\sqrt{\text{iSWAP}}$	CX	$\sqrt{\text{CX}}$	B	$\sqrt{\text{B}}$
Duration	**1.0**	**0.5**	**1.8**	**0.9**	**1.4**	**0.7**
CNOT	2.0	1.0	1.8	1.8	2.8	1.4
SWAP	3.0	1.5	5.4	5.4	2.8	2.8
Haar	3.0	1.1	5.4	3.2	2.8	2.2

Zhou, et al. **npj Quantum Inf 9**, 54 (2023).

Extended candidate basis gates

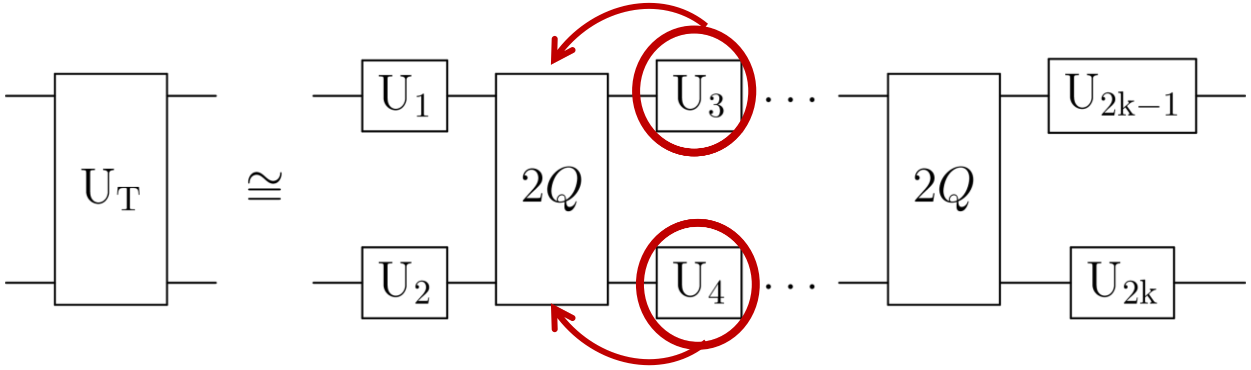


- **Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration**

$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger)$$

McKinney, et al. **ISCA** (2023)

Extended candidate basis gates

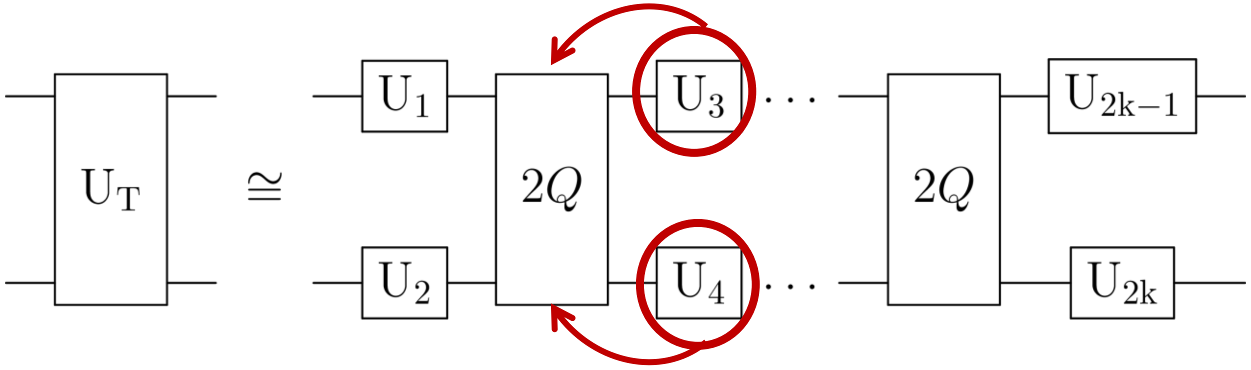


- **Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration**

$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger) \\ +\epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$

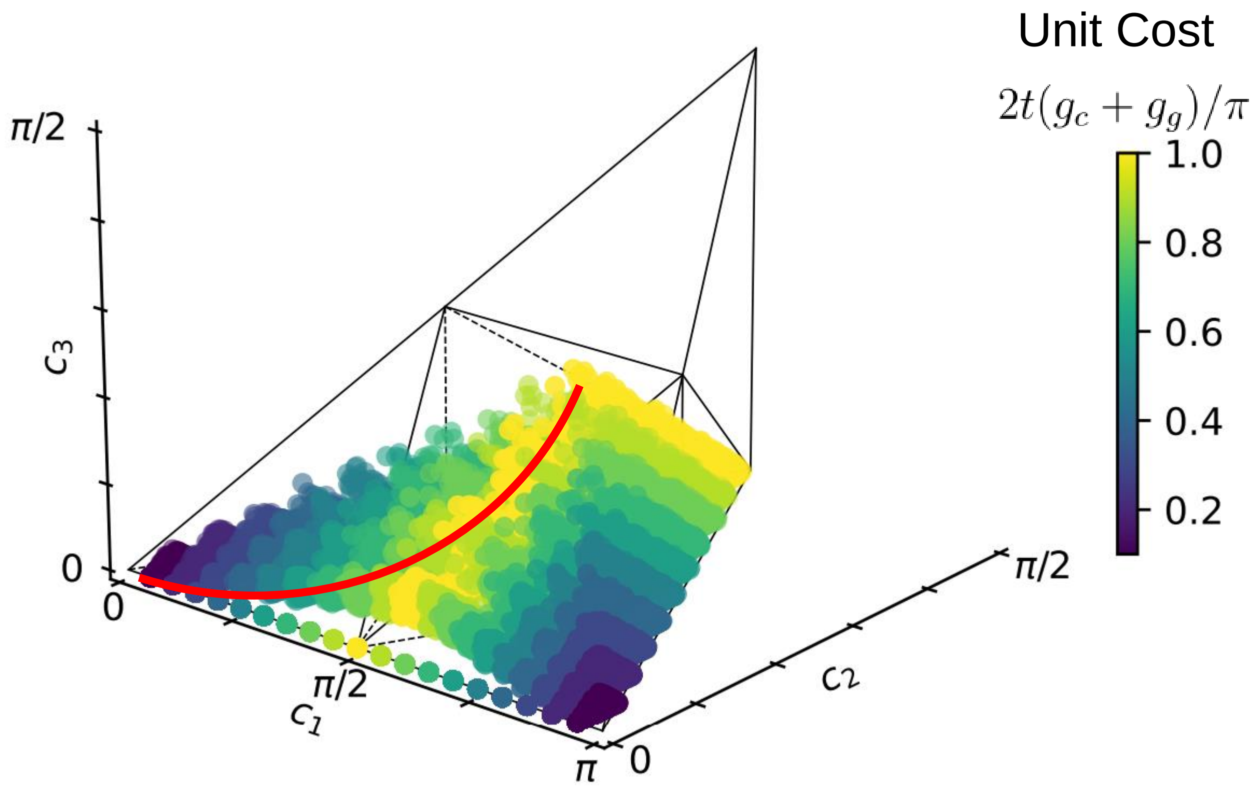
McKinney, et al. **ISCA** (2023)

Extended candidate basis gates



- **Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration**

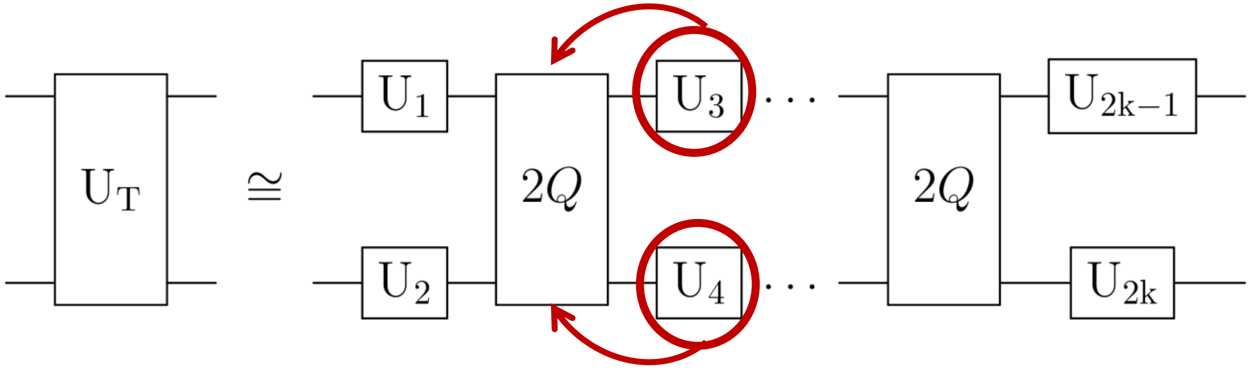
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger)$$
$$+\epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$



- Parallel-Drive "steers" to previously inaccessible regions

McKinney, et al. **ISCA** (2023)

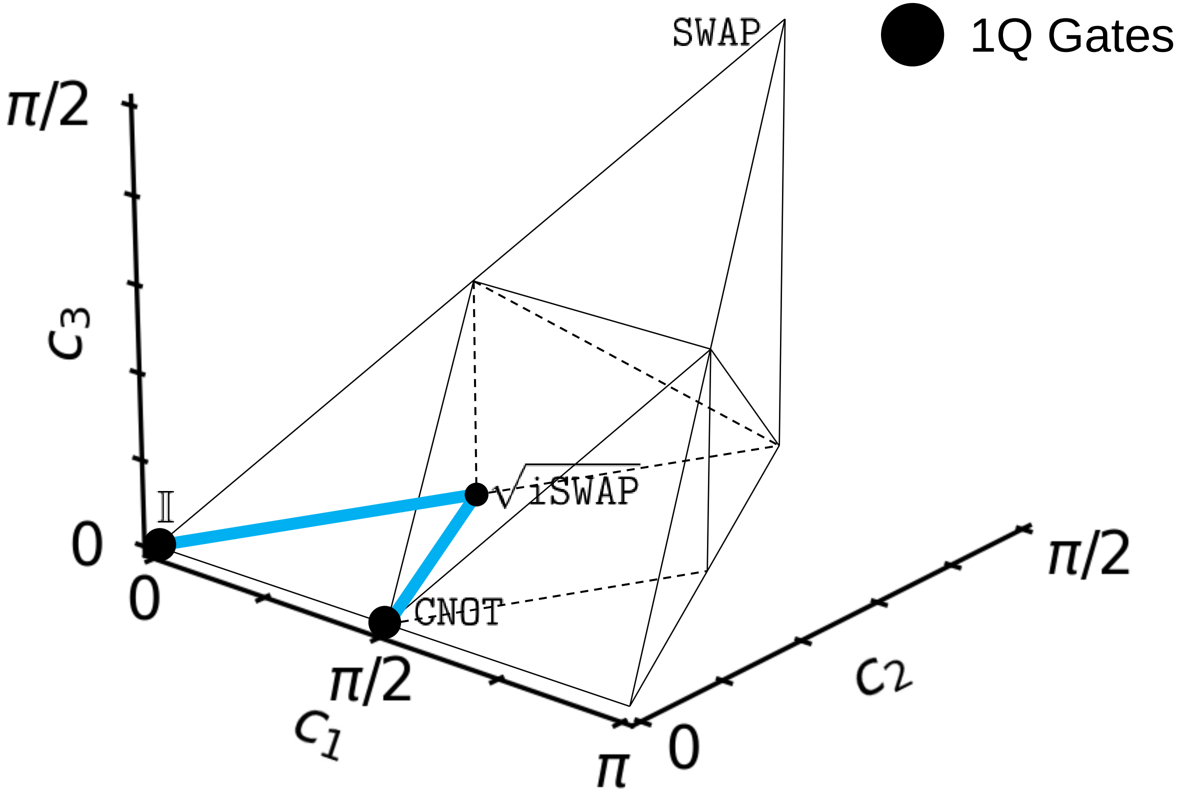
Extended candidate basis gates



- **Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration**

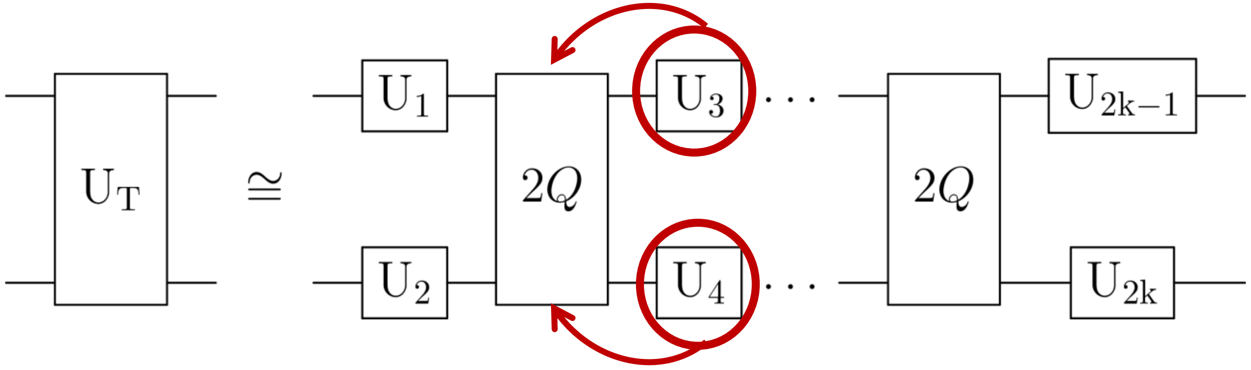
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger) + \epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$

McKinney, et al. **ISCA** (2023)



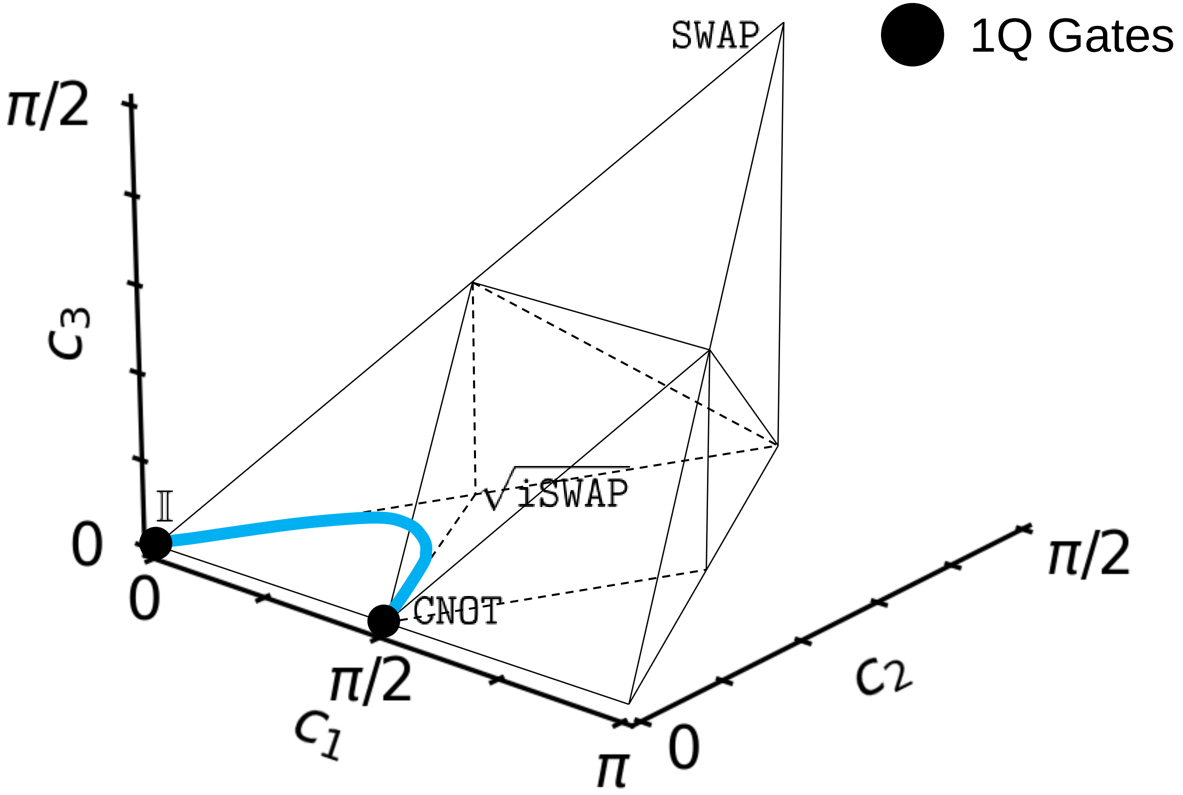
- Parallel-Drive “steers” to perform decomposition

Extended candidate basis gates



- Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration

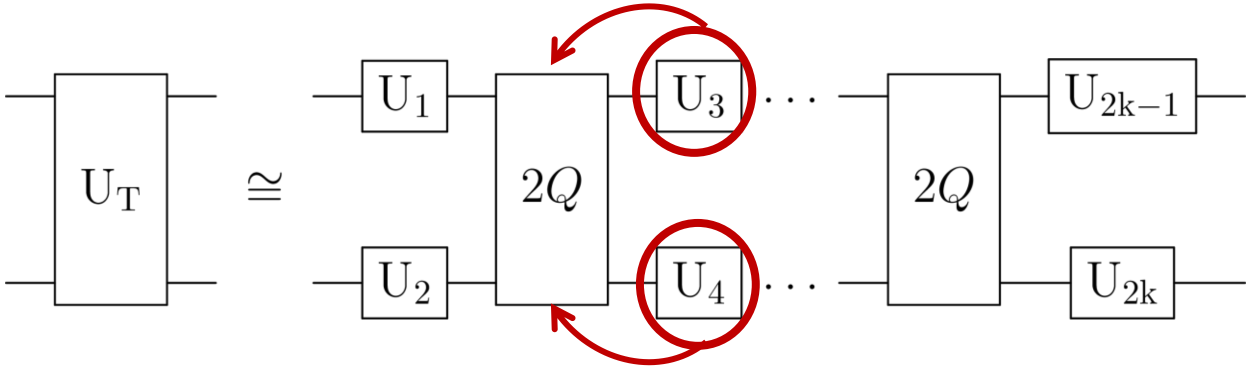
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger) + \epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$



- Parallel-Drive "steers" to perform decomposition

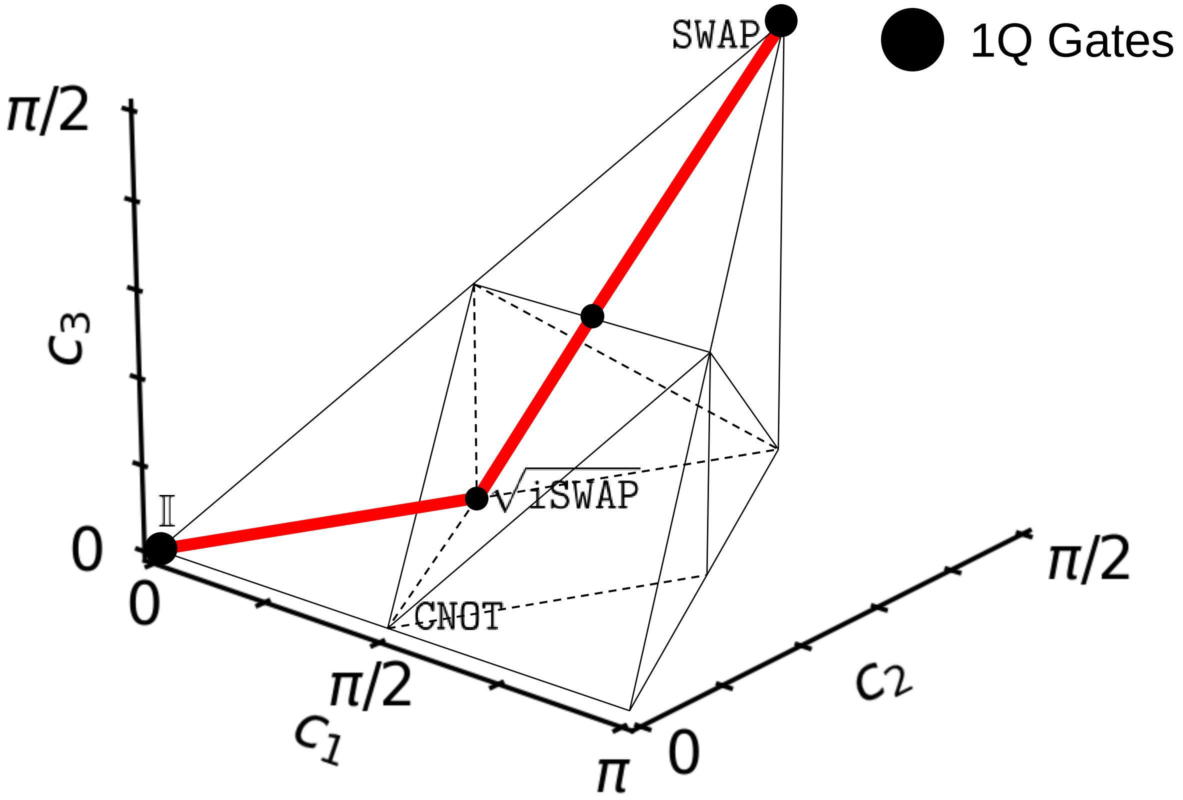
McKinney, et al. **ISCA** (2023)

Extended candidate basis gates



- **Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration**

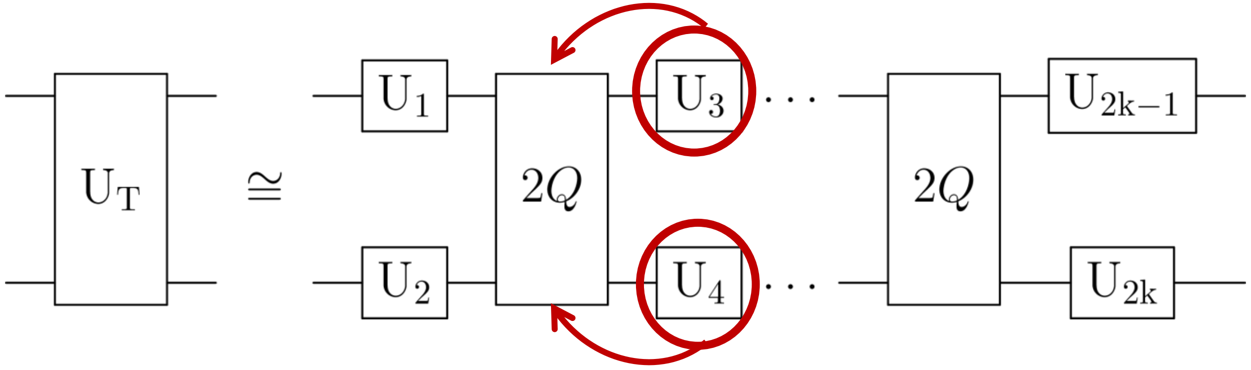
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger) + \epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$



- Parallel-Drive "steers" to perform decomposition

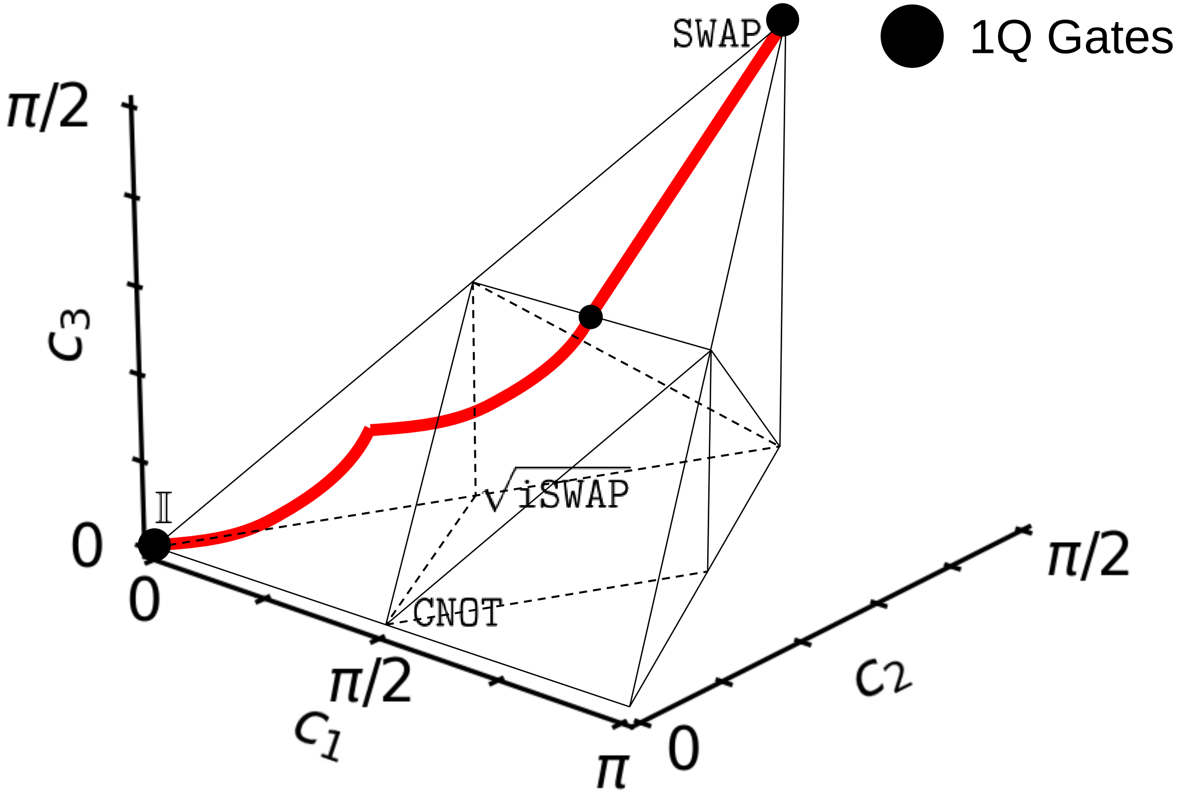
McKinney, et al. **ISCA** (2023)

Extended candidate basis gates



- Drive qubits independently from the SNAIL in discrete time steps equivalent to basis gate duration

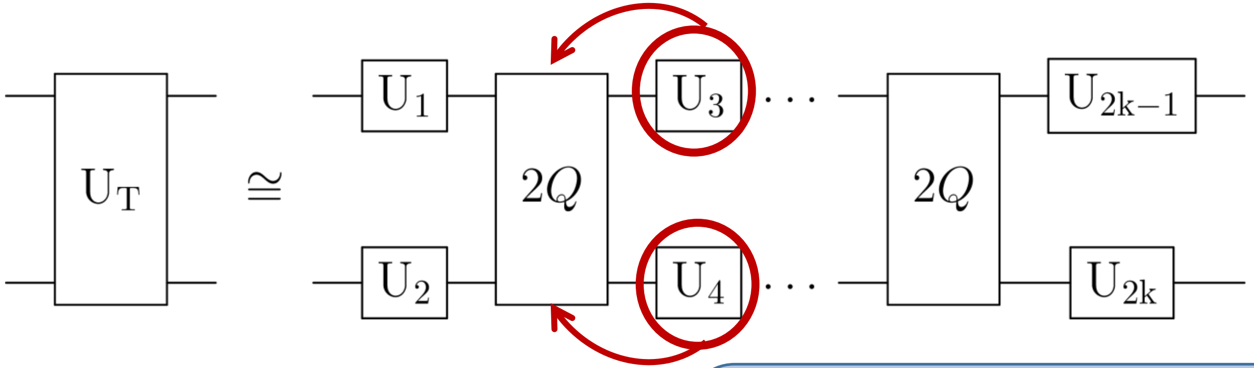
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger) + \epsilon_1(t)(a+a^\dagger) + \epsilon_2(t)(b+b^\dagger)$$



- Parallel-Drive "steers" to perform decomposition

McKinney, et al. **ISCA** (2023)

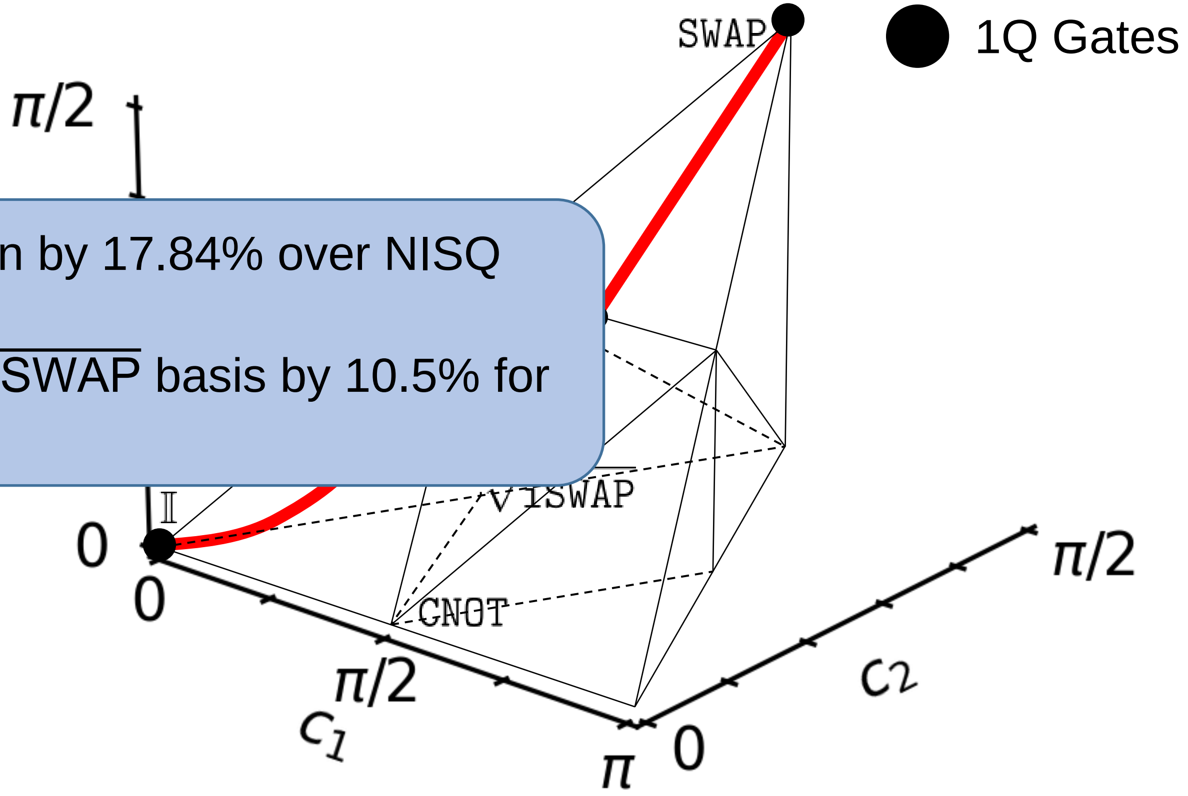
Extended candidate basis gates



1. Decrease circuit duration by 17.84% over NISQ benchmarks!
2. Improve fidelity using $\sqrt{\text{iSWAP}}$ basis by 10.5% for random gates

- ➢ **Drive qubits independently … discrete time steps equival… duration**

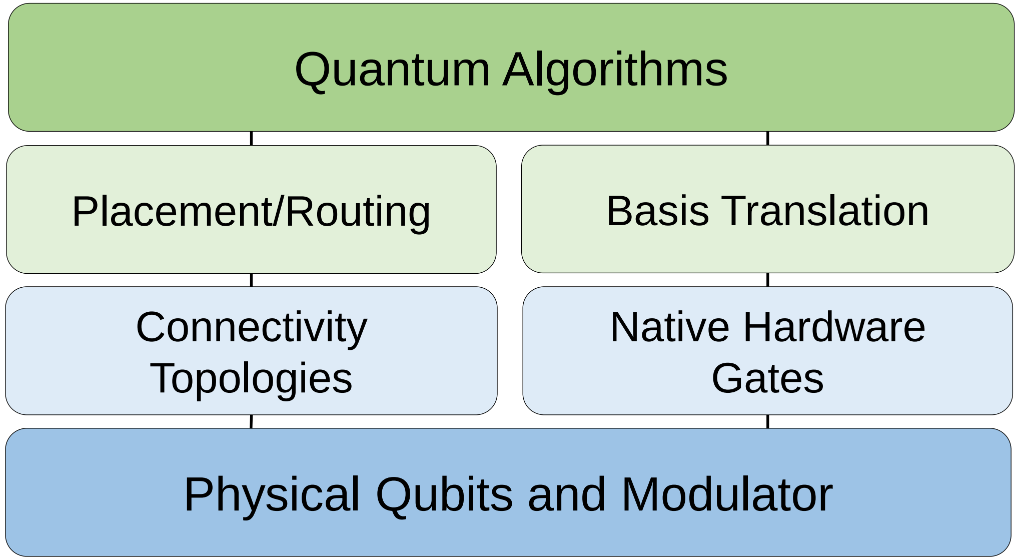
$$\hat{H} = g_c(e^{i\phi_c}a^\dagger b + e^{-i\phi_c}ab^\dagger) + g_g(e^{i\phi_g}ab + e^{-i\phi_g}a^\dagger b^\dagger)$$
$$+\epsilon_1(t)(a + a^\dagger) + \epsilon_2(t)(b + b^\dagger)$$

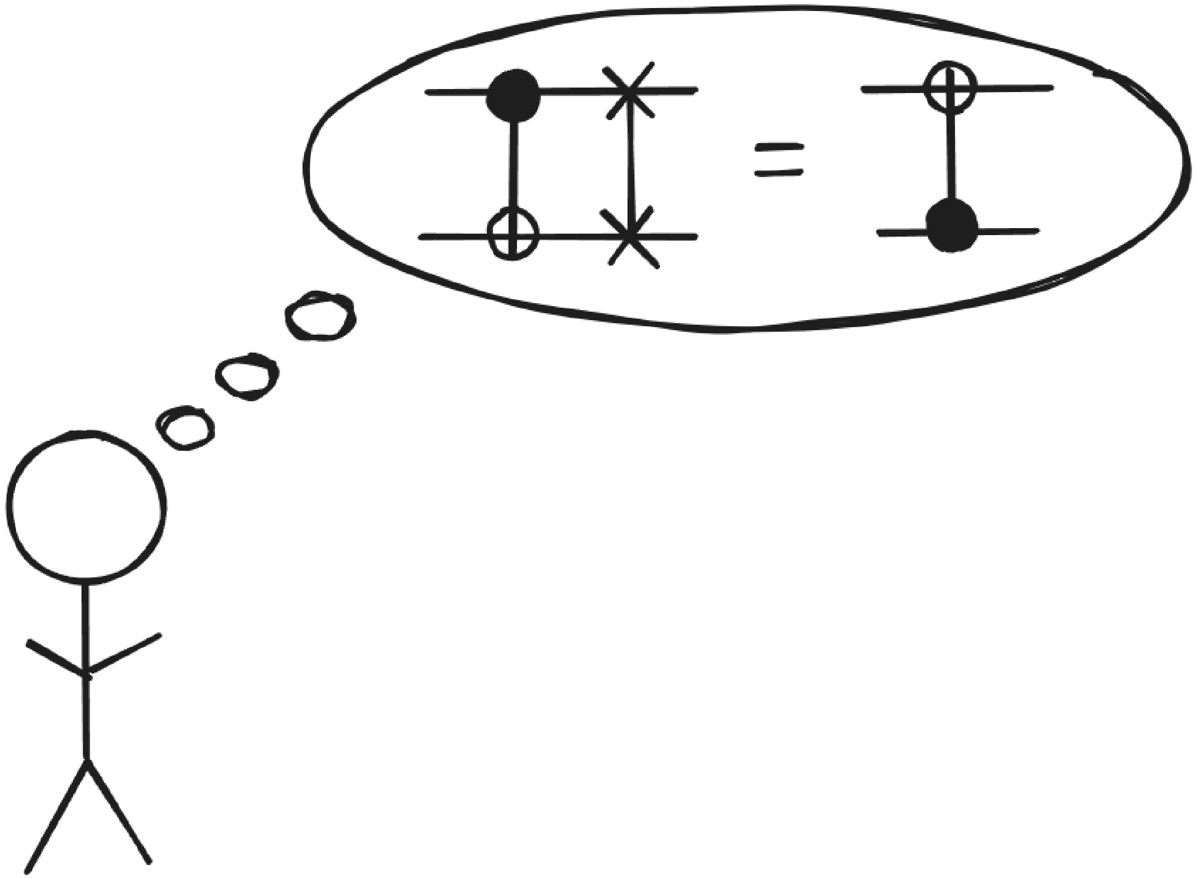


- ➢ Parallel-Drive “steers” to perform decomposition

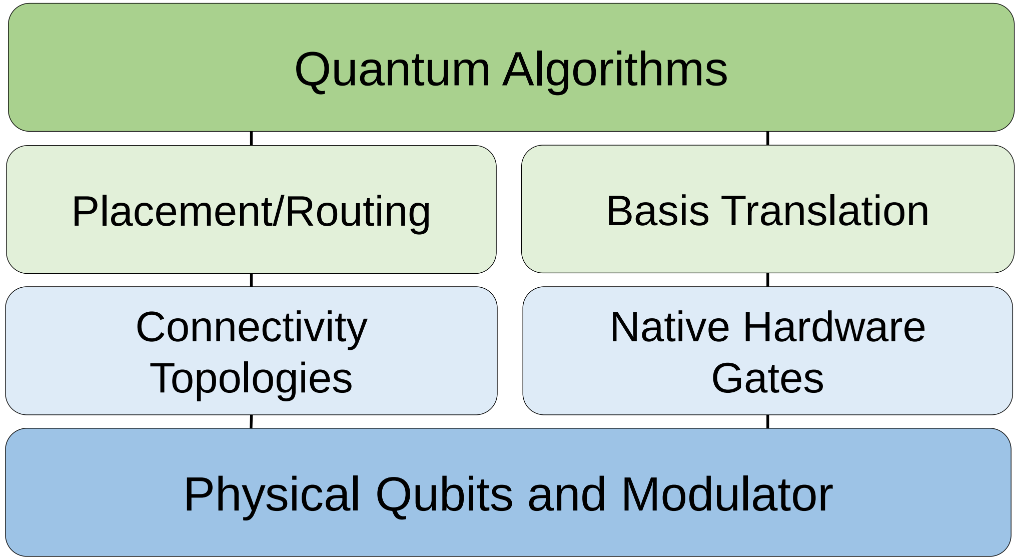
McKinney, et al. **ISCA** (2023)

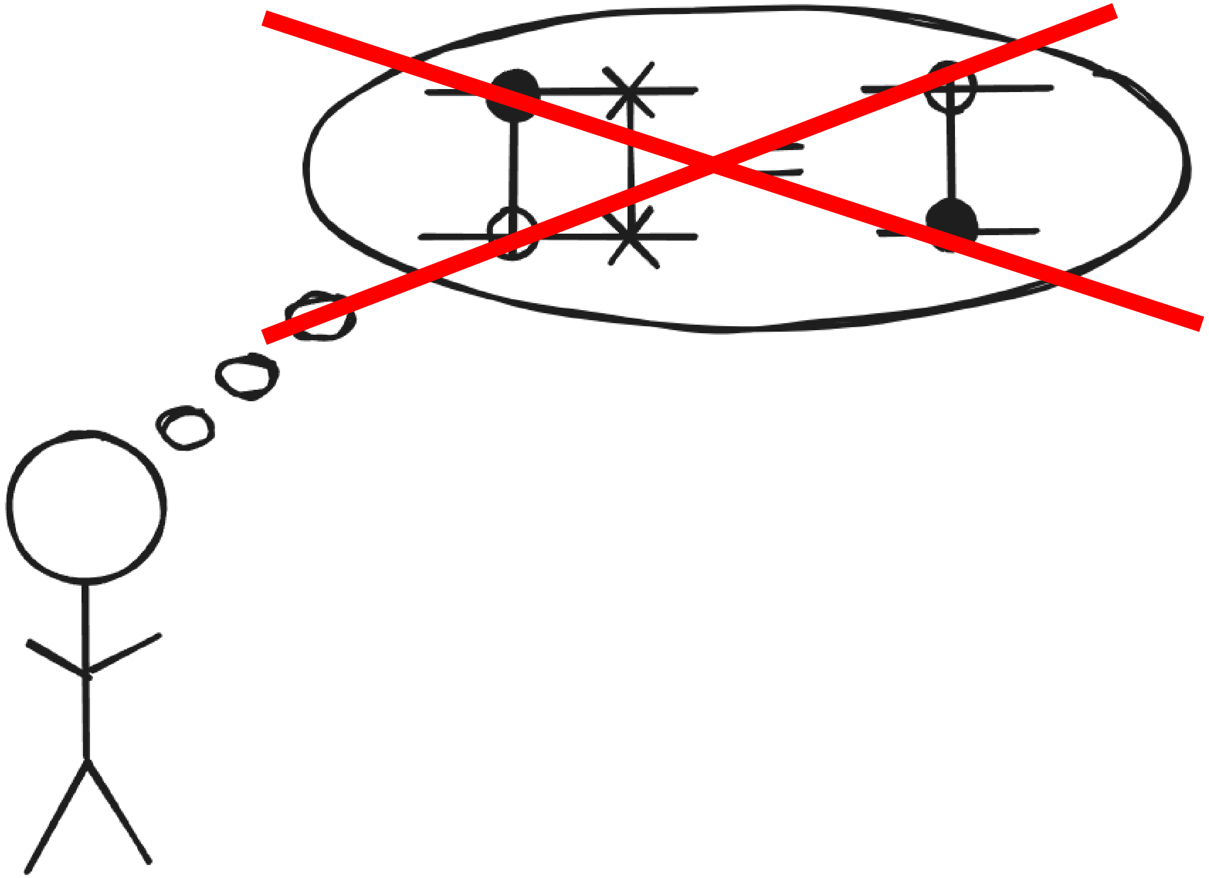
What do SWAPs do?





What do SWAPs do?





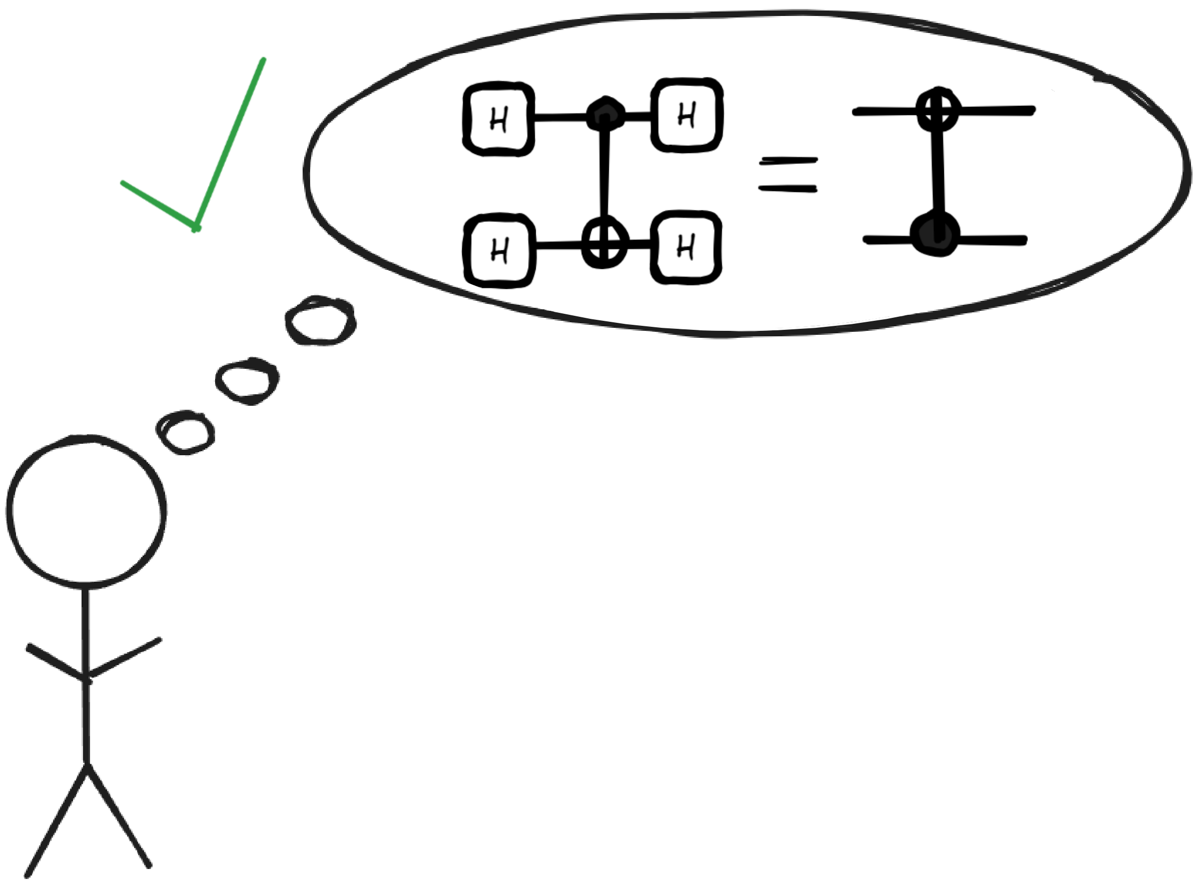
What do SWAPs do?

Quantum Algorithms

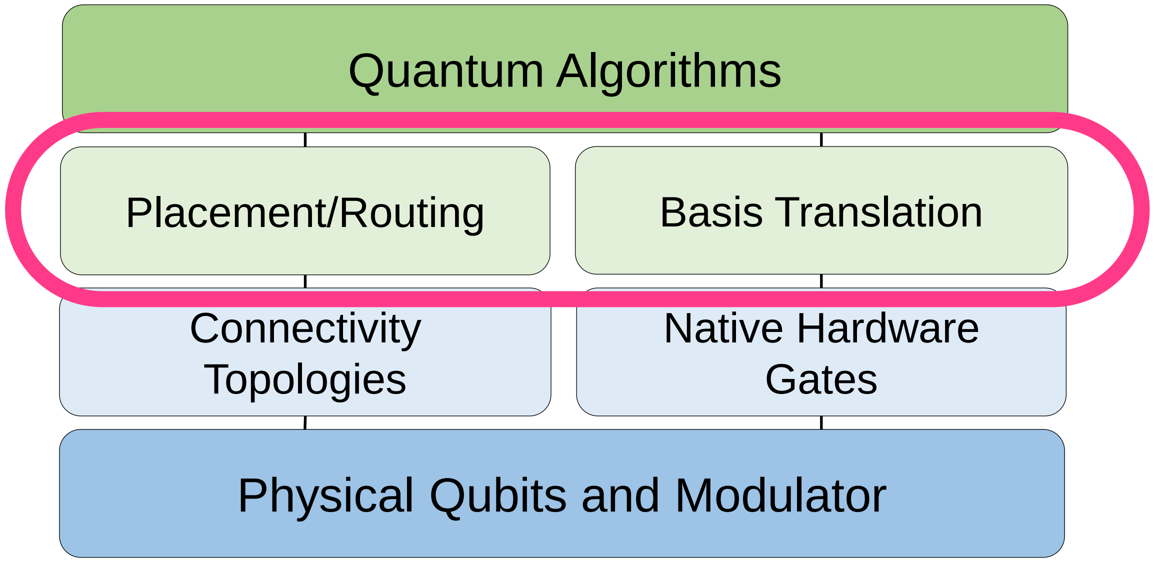
Placement/Routing | Basis Translation

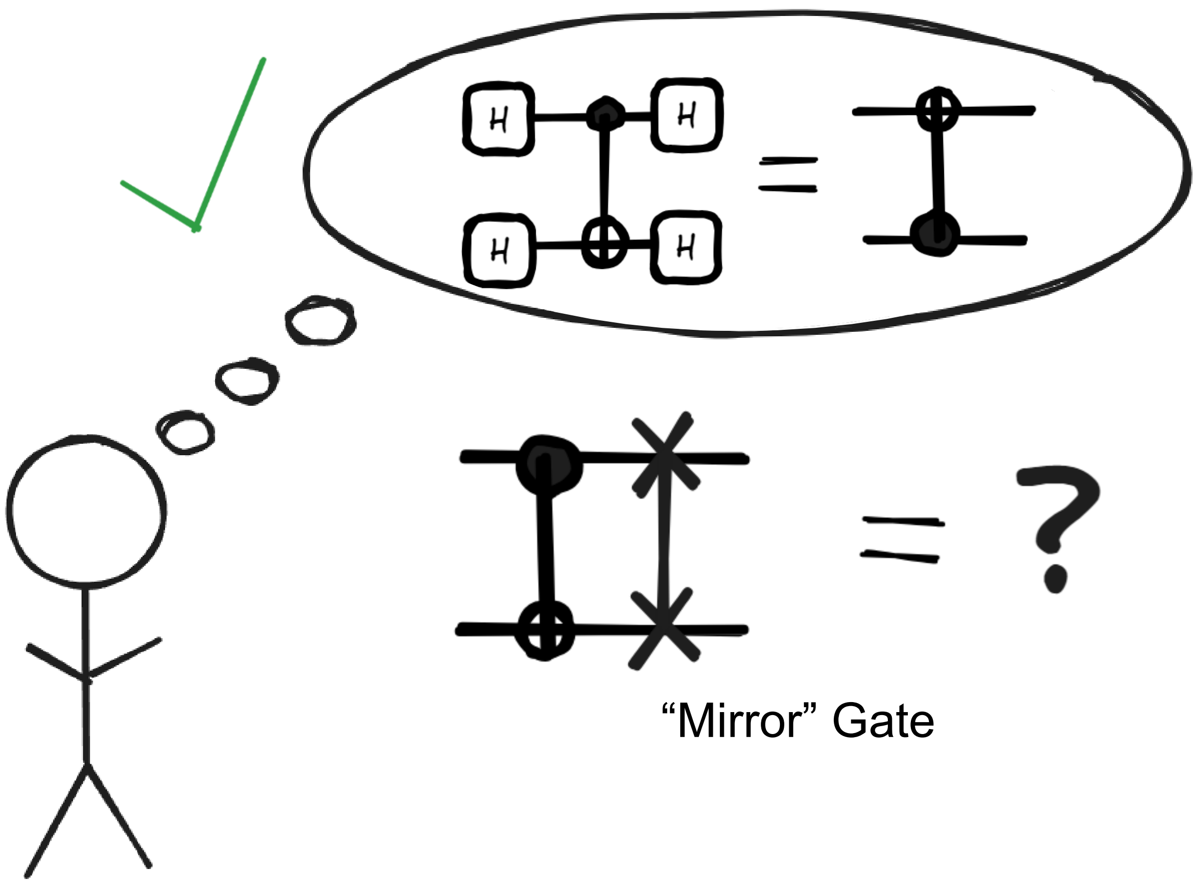
Connectivity Topologies | Native Hardware Gates

Physical Qubits and Modulator



What do SWAPs do?





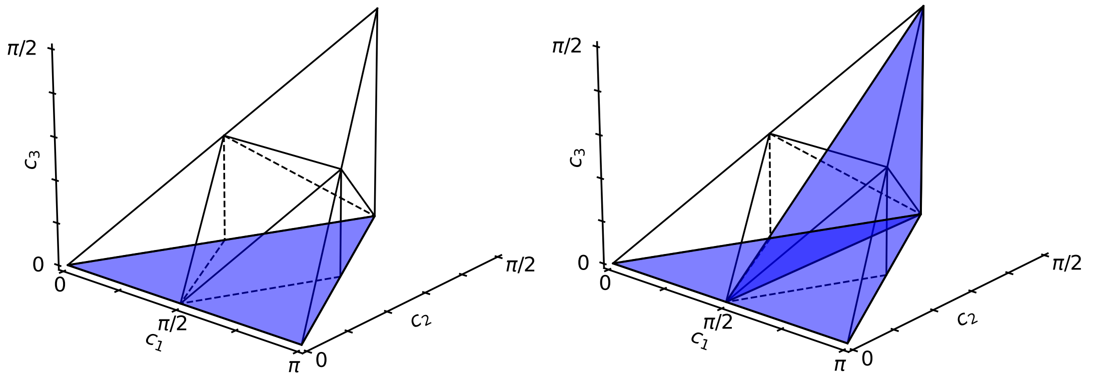
➢ **Key Idea: Routing and Basis Translation are not independent, all SWAP gates must also be decomposed.**

Mirror-inclusive coverage sets

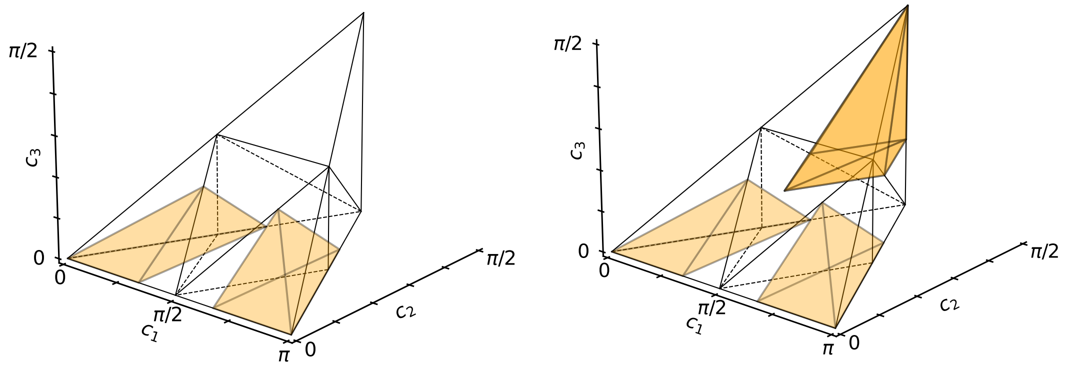
- Compute using `monodromy`:
  - Union of coverage volume and the mirror coverage volume
  - OR include a 0-cost SWAP gate in the basis set

For all, $k = 2$

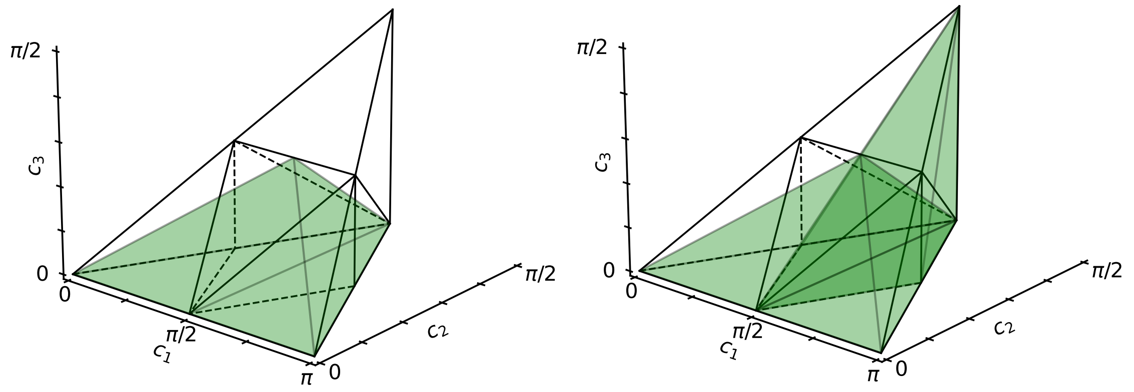
CNOT



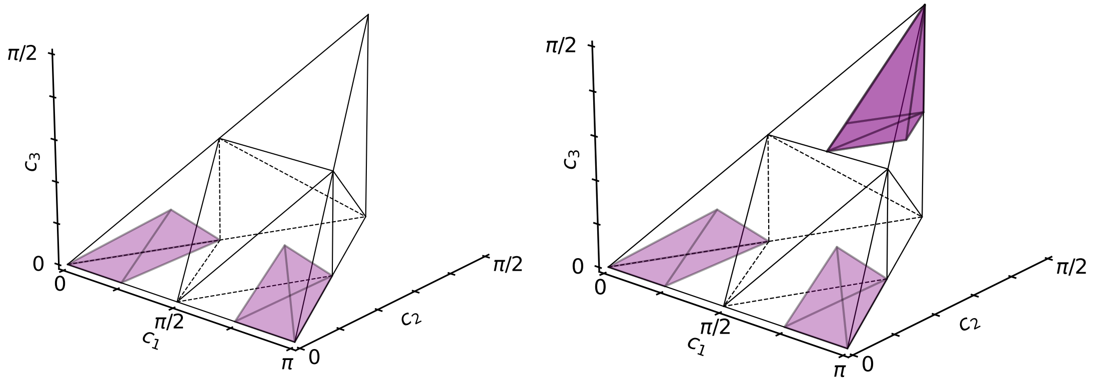
$\sqrt[3]{\text{iSWAP}}$



$\sqrt{\text{iSWAP}}$



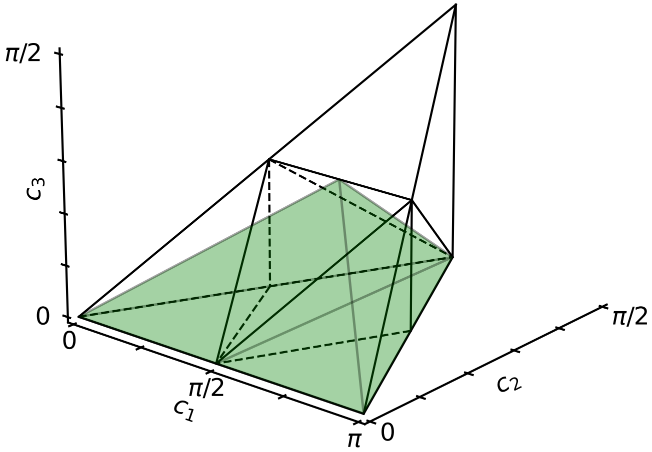
$\sqrt[4]{\text{iSWAP}}$

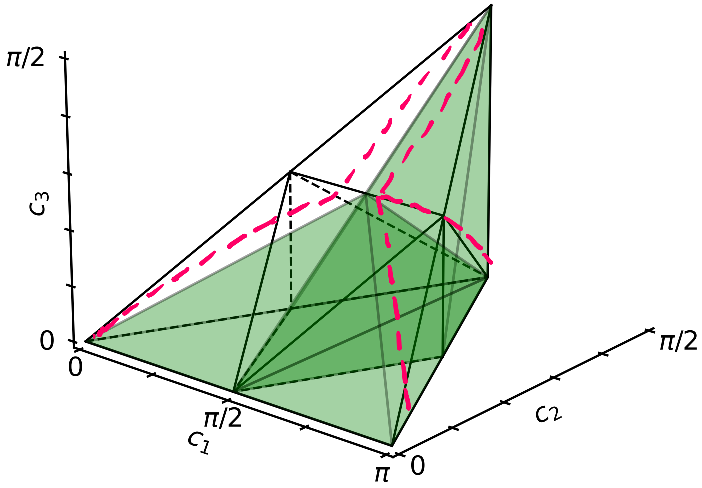


Monte Carlo Haar scores

- Intuition: Approximate decomp threshold defines an acceptable inflated polytope volume.

Javadi, Ali. **APS March Meeting** (2023)



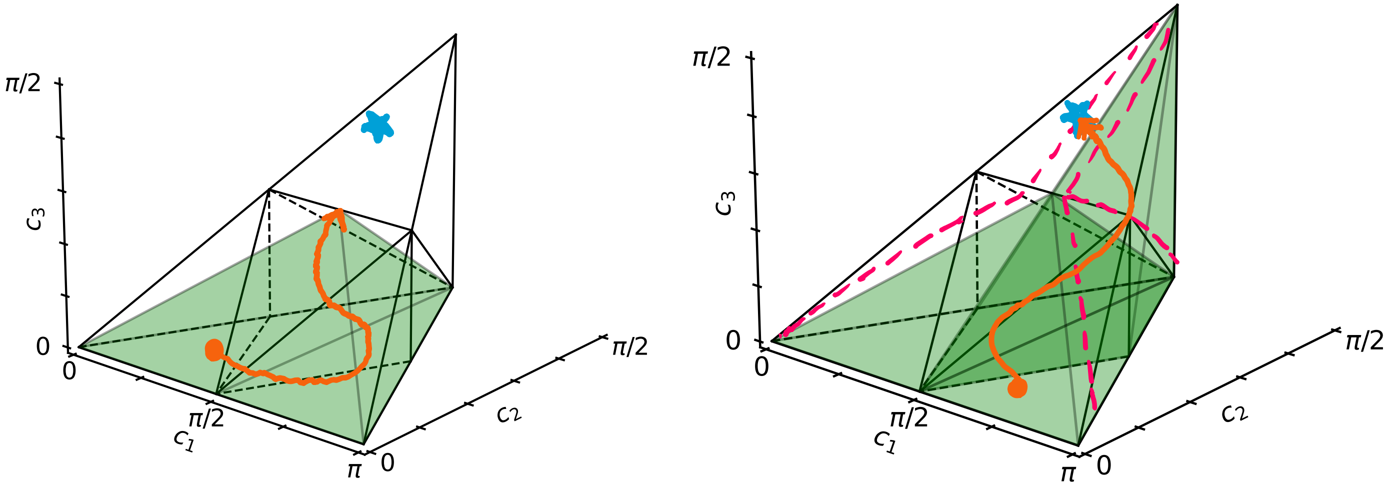


McKinney, et al. **arXiv:2308.03874** (2023)

Monte Carlo Haar scores

- Intuition: Approximate decomp threshold defines an acceptable inflated polytope volume.

Javadi, Ali. **APS March Meeting** (2023)



Exact: Fail

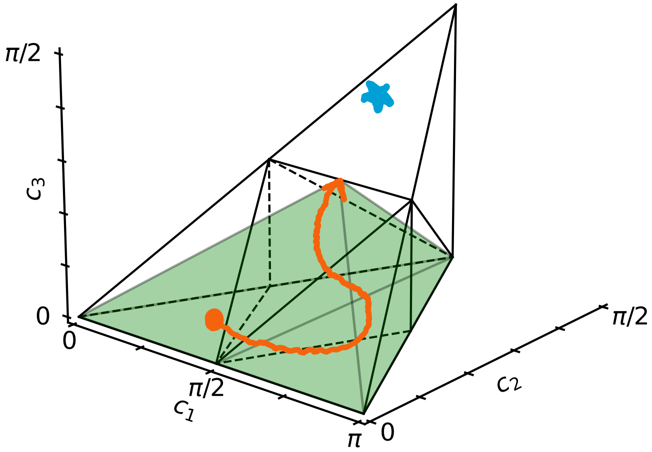
Approx. + Mirrors: Success

McKinney, et al. **arXiv:2308.03874** (2023)

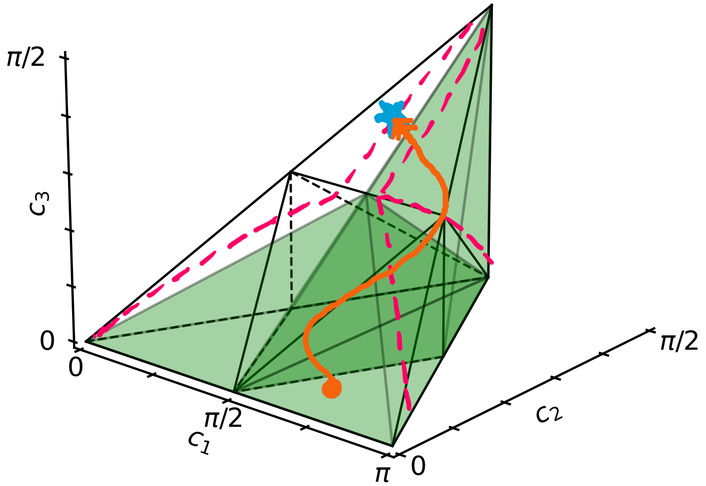
Monte Carlo Haar scores

- Intuition: Approximate decomp threshold defines an acceptable inflated polytope volume.

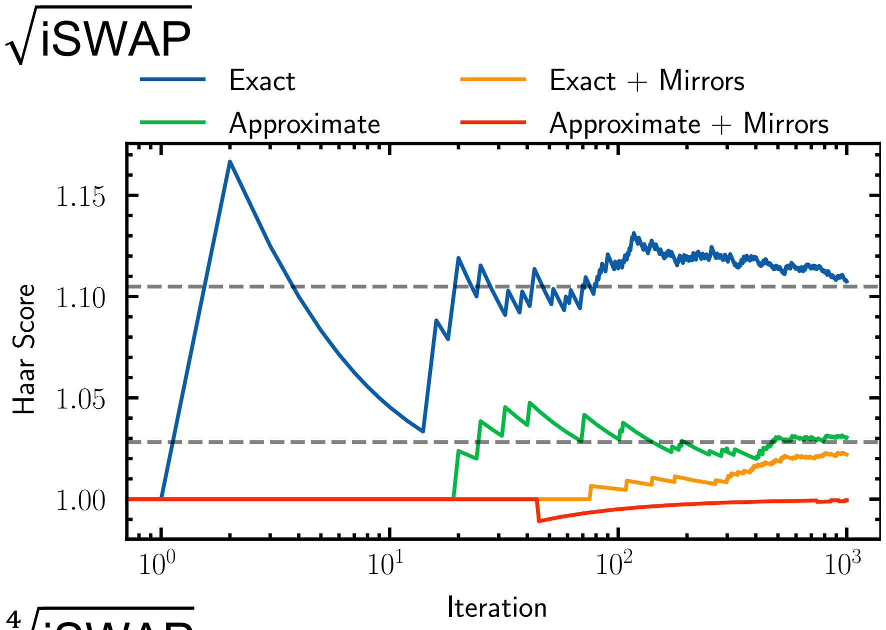
Javadi, Ali. **APS March Meeting** (2023)

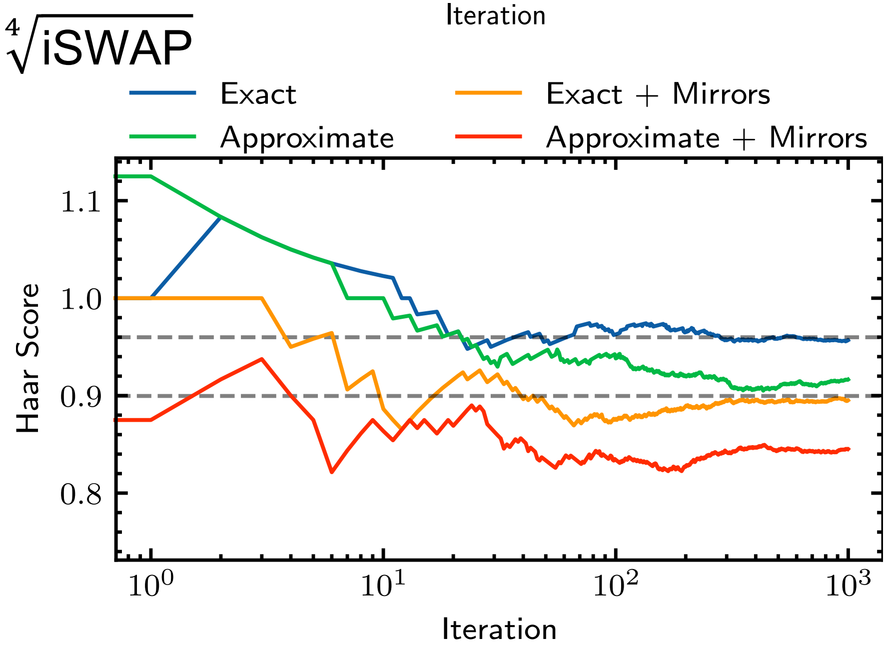


Exact: Fail



Approx. + Mirrors: Success

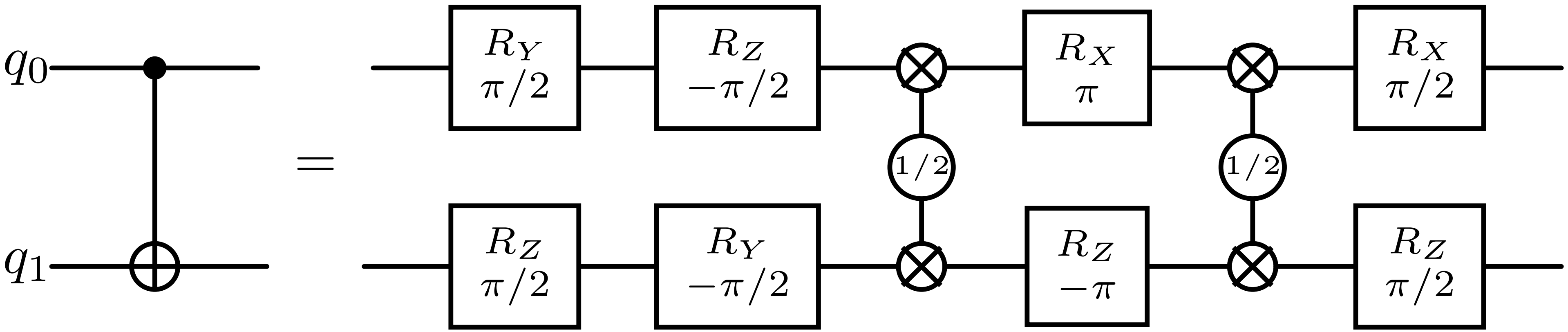


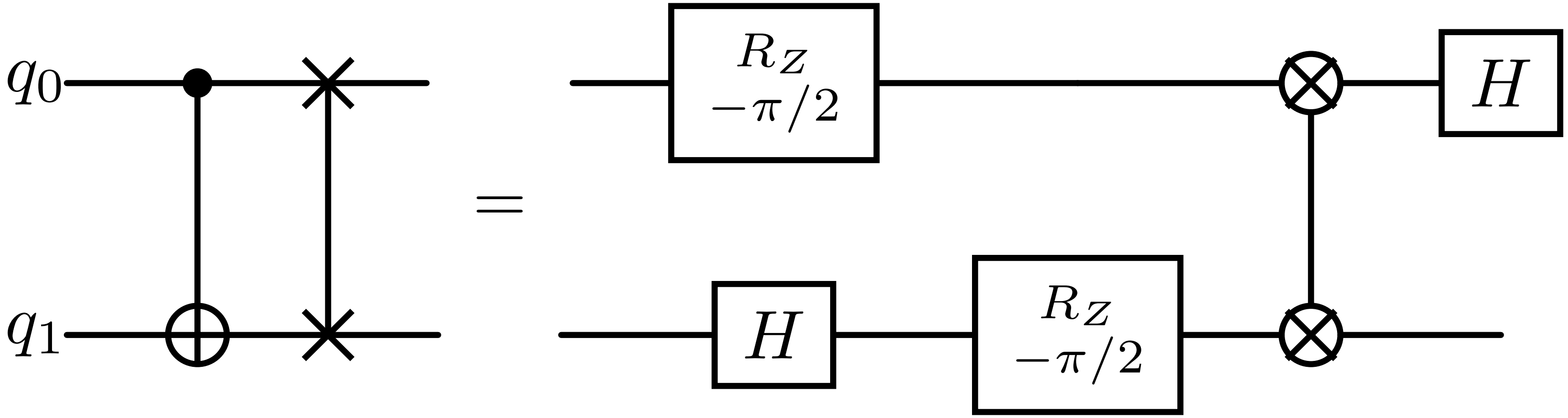


- $\sqrt{\text{iSWAP}}$ with approximate decomp + mirrors has an 8.8% relative decrease in total infidelity

McKinney, et al. **arXiv:2308.03874** (2023)

Decomposition identities

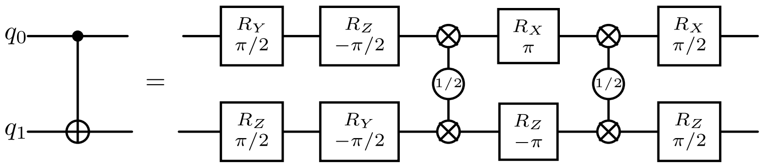


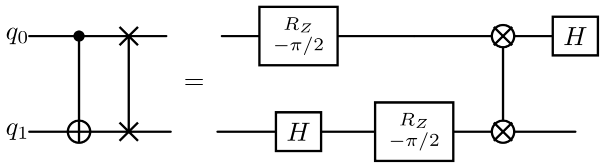


Schuch, et al. **Physical Review A 67.3** (2003)

McKinney, et al. **arXiv:2308.03874** (2023)

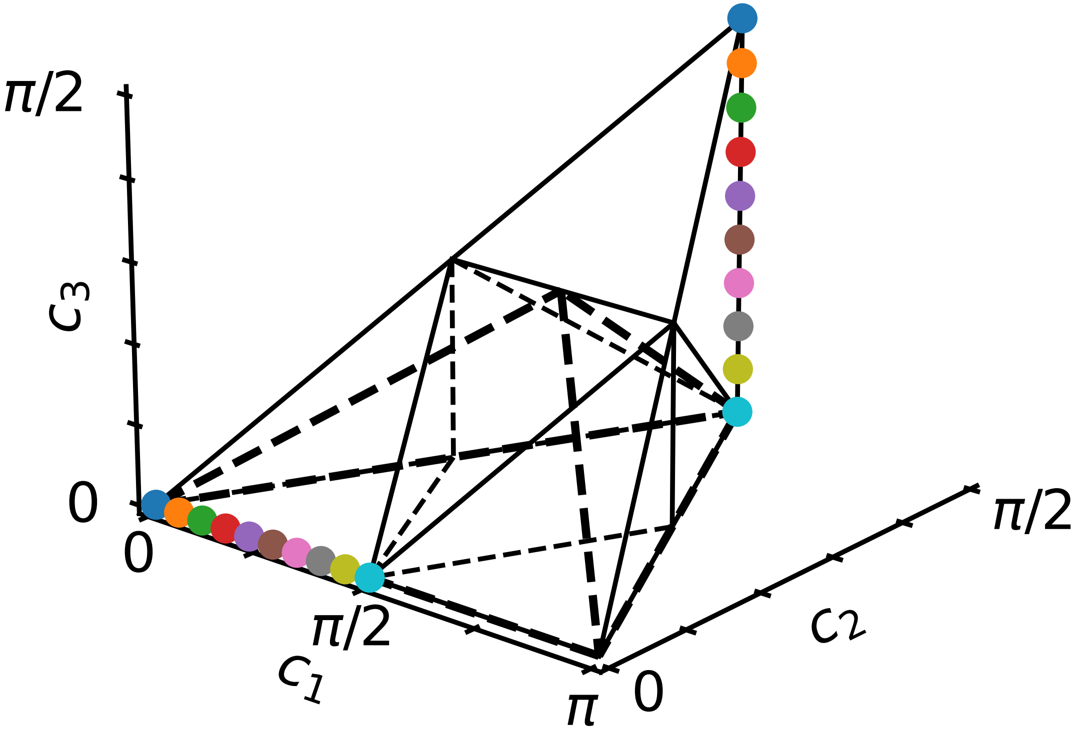
Why does this work?





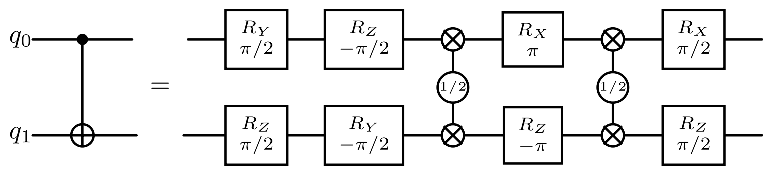
- CPHASE gates mirror to pSWAP gates

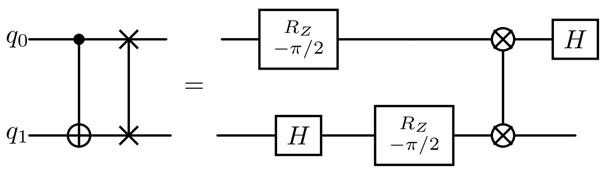
$$(a', b', c') = \begin{cases} \left(\frac{\pi}{4} + c, \frac{\pi}{4} - b, \frac{\pi}{4} - a\right) & \text{if } a \leq \frac{\pi}{4} \\ \left(\frac{\pi}{4} - c, \frac{\pi}{4} - b, a - \frac{\pi}{4}\right) & \text{else} \end{cases}$$



Peterson, et al. **Quantum 6** (2022)

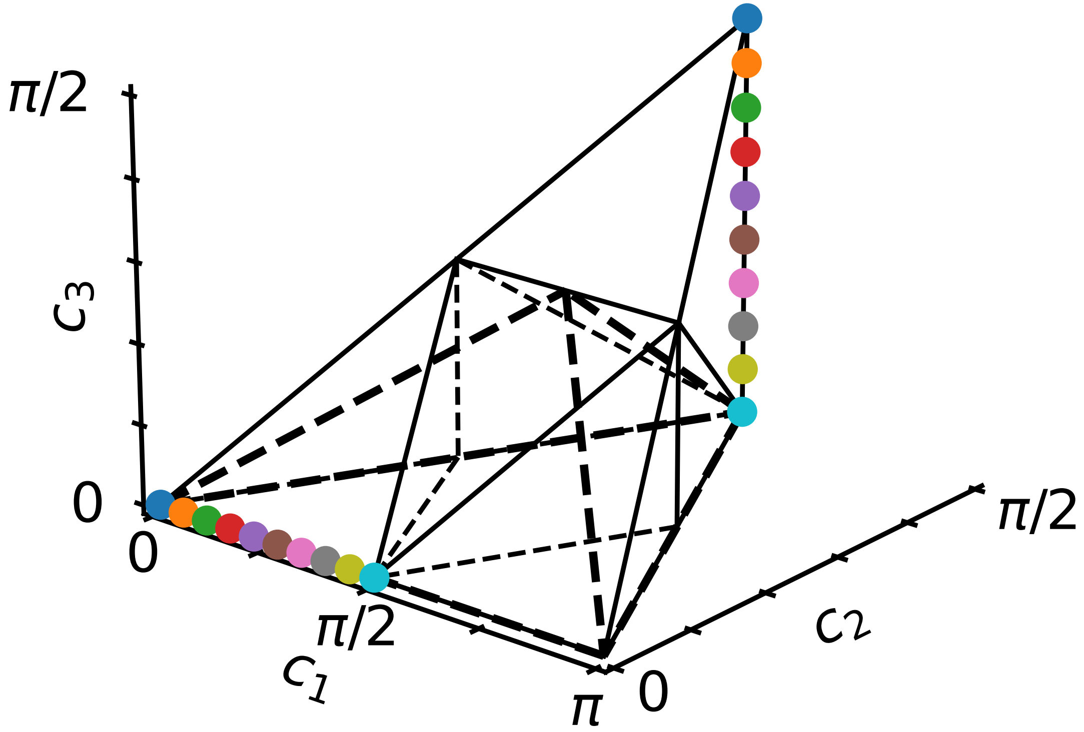
Why does this work?

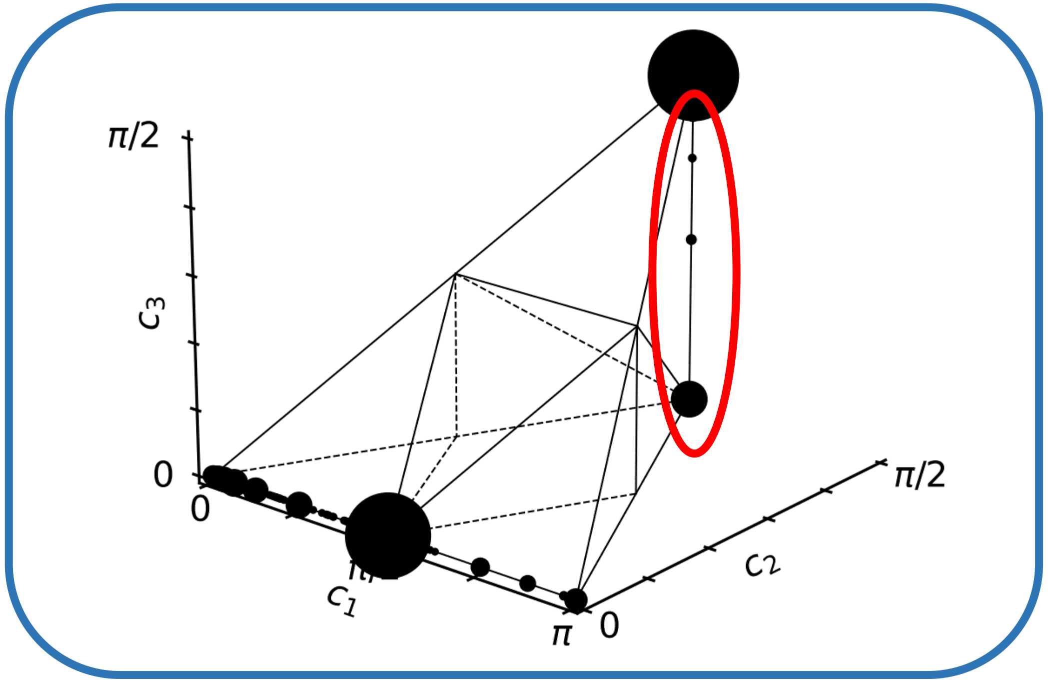




➢ CPHASE gates mirror to pSWAP gates

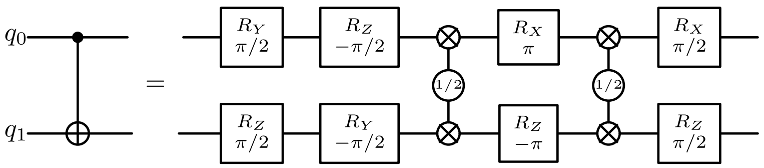
$$(a', b', c') = \begin{cases} \left(\frac{\pi}{4} + c, \frac{\pi}{4} - b, \frac{\pi}{4} - a\right) & \text{if } a \leq \frac{\pi}{4} \\ \left(\frac{\pi}{4} - c, \frac{\pi}{4} - b, a - \frac{\pi}{4}\right) & \text{else} \end{cases}$$

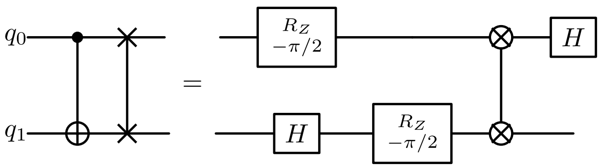




Peterson, et al. **Quantum 6** (2022)

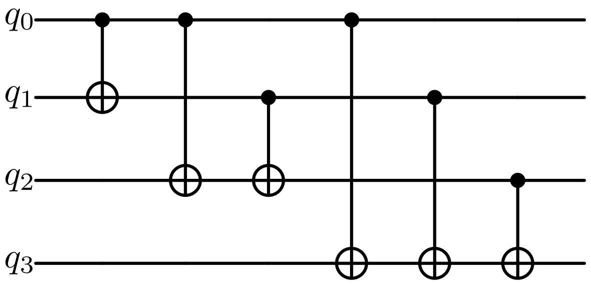
Using this identity for data movement



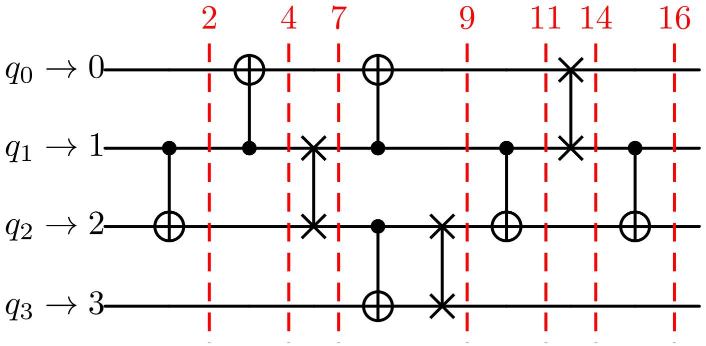


- ➢ Intuition: For every CX, decide whether output qubit ordering is (q0, q1) or (q1, q0) based on whether it makes the qubits closer to their next qubit pair

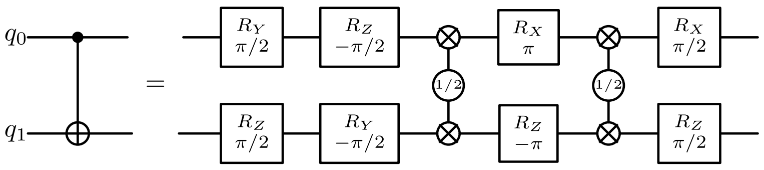
Goal: Full entanglement on a line topology

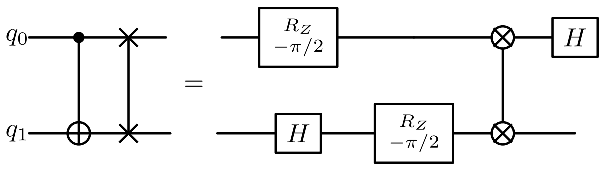


Qiskit



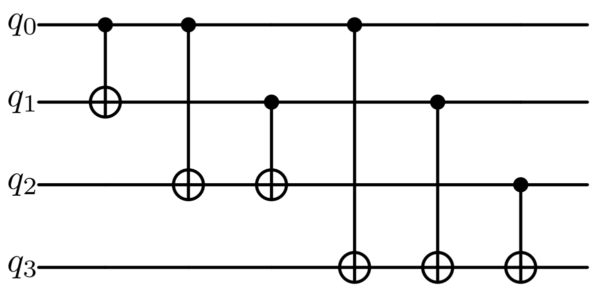
Using this identity for data movement



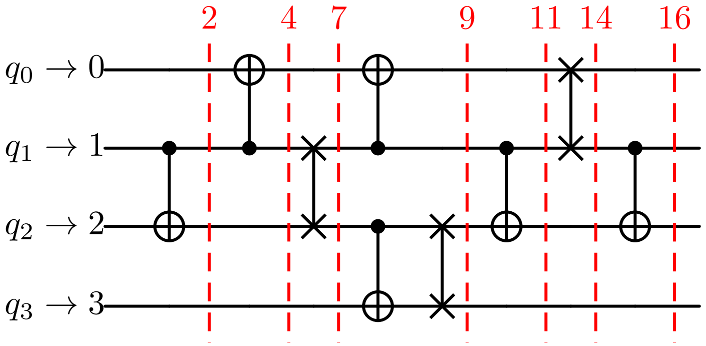


> ➢ Intuition: For every CX, decide whether output qubit ordering is (q0, q1) or (q1, q0) based on whether it makes the qubits closer to their next qubit pair

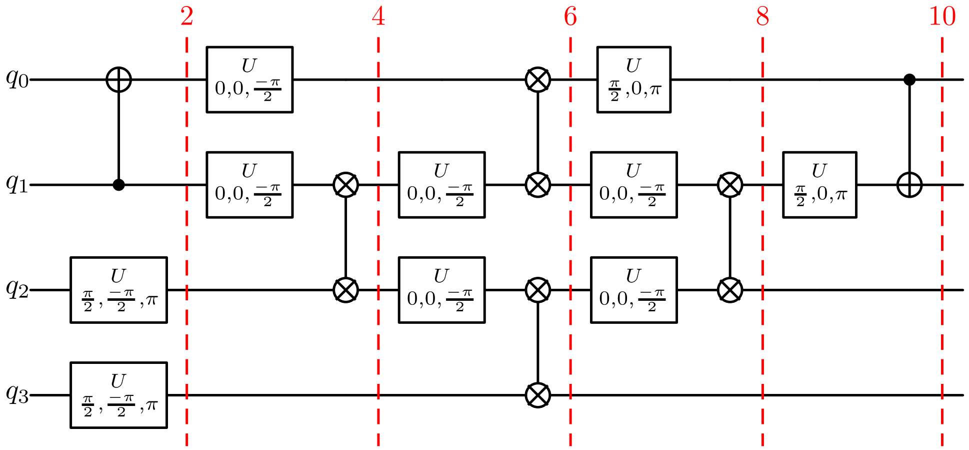
Goal: Full entanglement on a line topology



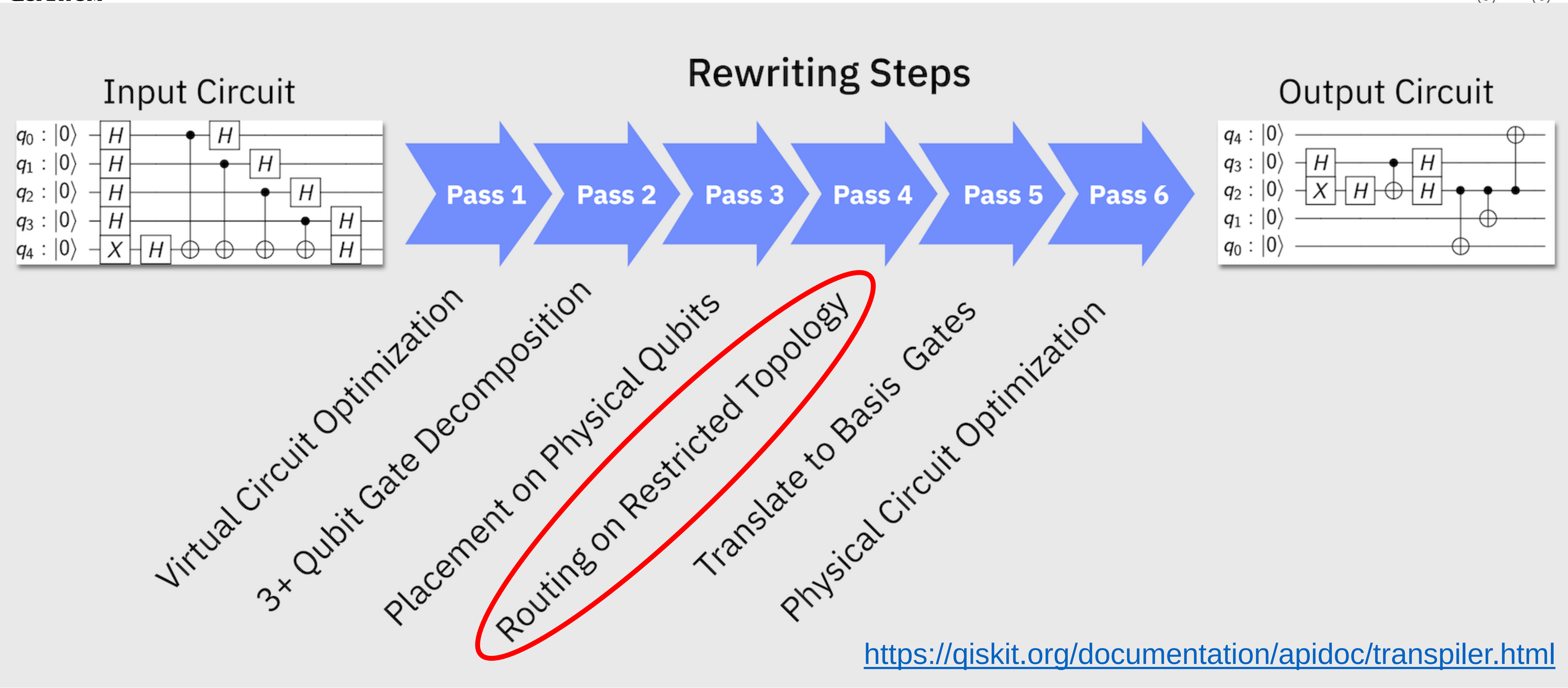
Qiskit



MIRAGE

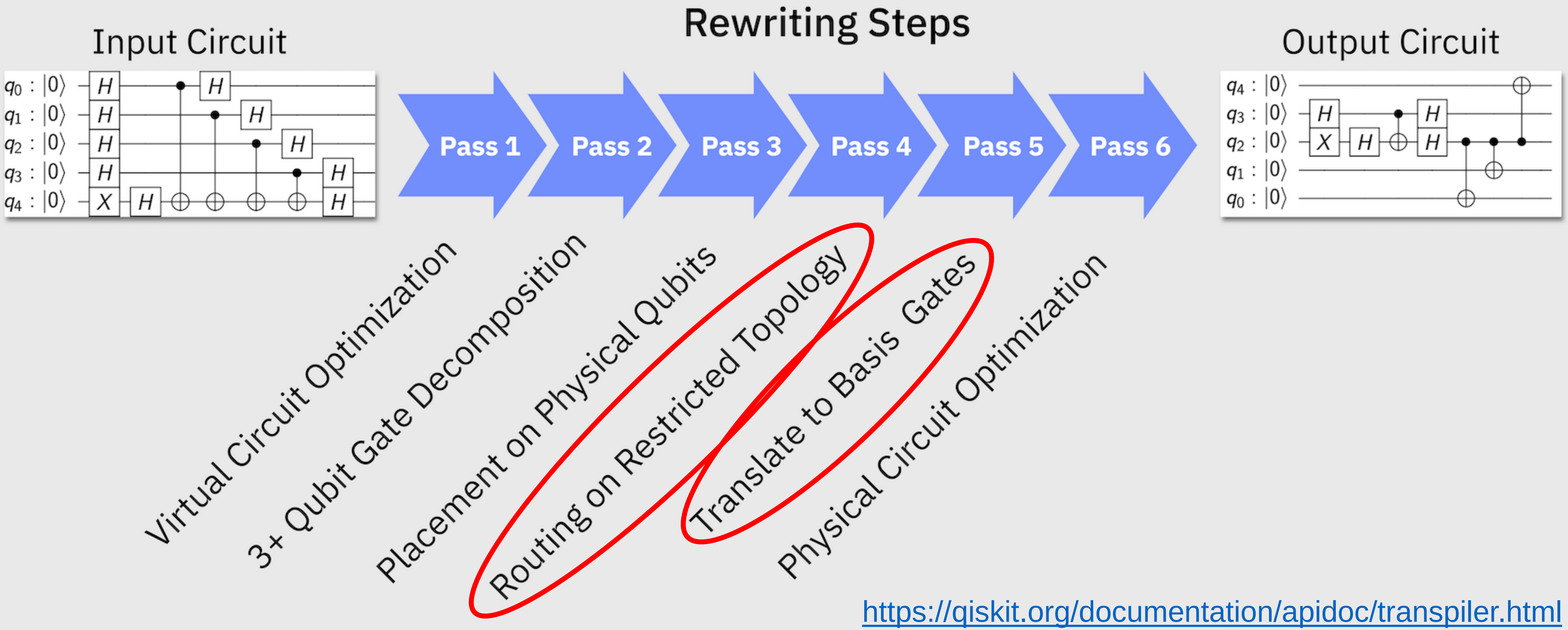


Qiskit flow



Li, et al. **ASPLOS** (2019)

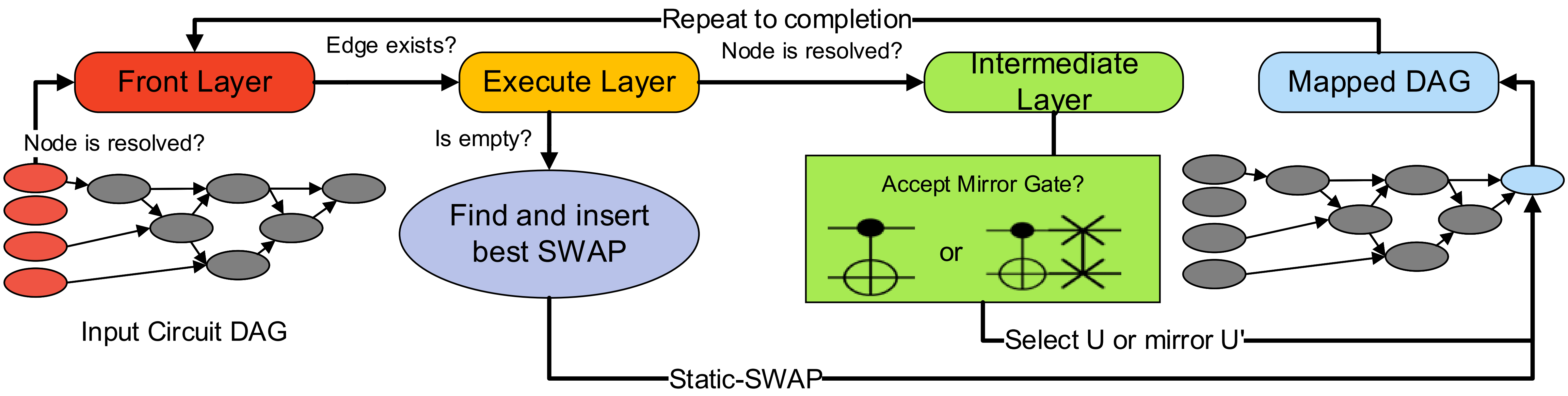
Qiskit flow



https://qiskit.org/documentation/apidoc/transpiler.html

Li, et al. **ASPLOS** (2019)

Mirage flow

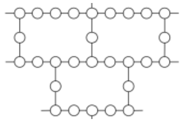


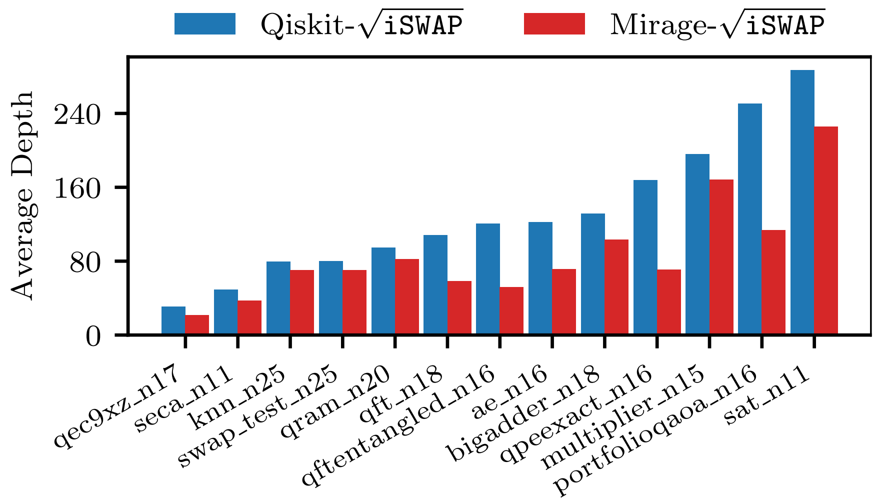
- Simple yet powerful modification to SABRE:
  - Each gate must pass through an Intermediate Layer
  - Considers if substituting the *mirror* would reduce topological distance cost

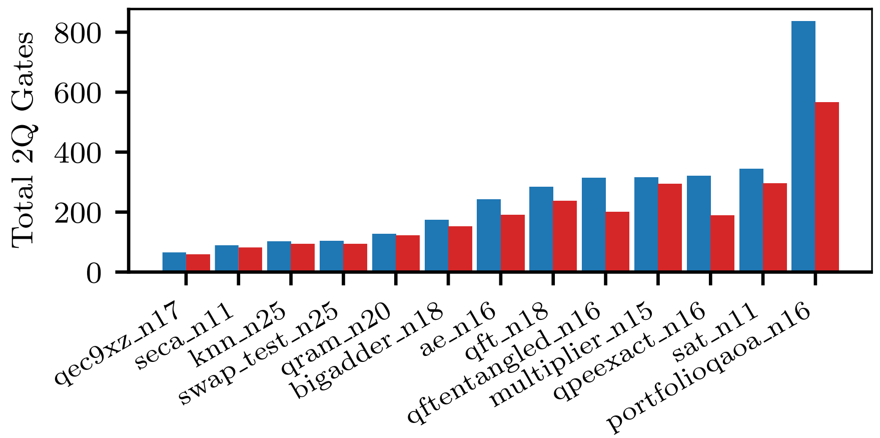
Li, et al. **ASPLOS** (2019)

Gate count results

- For the Heavy-Hex topology
  - Average depth decrease of **31.19%**
  - Average total gate decrease of **16.97%**

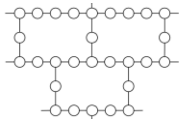


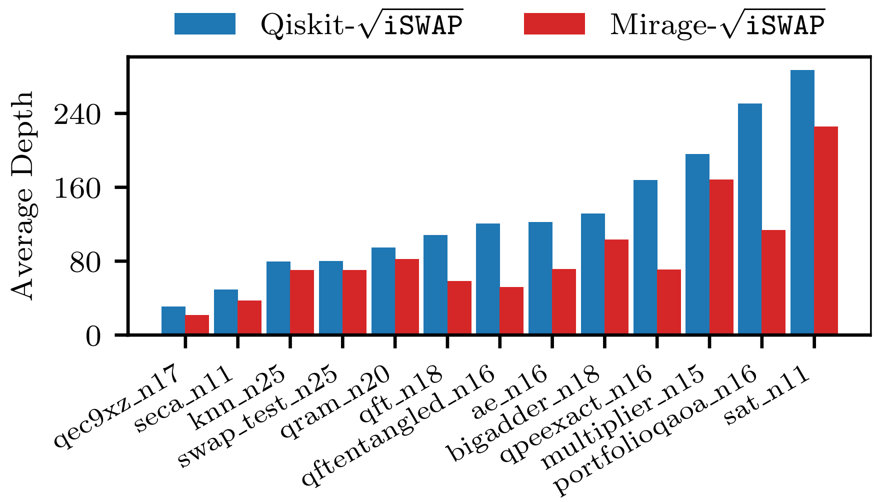


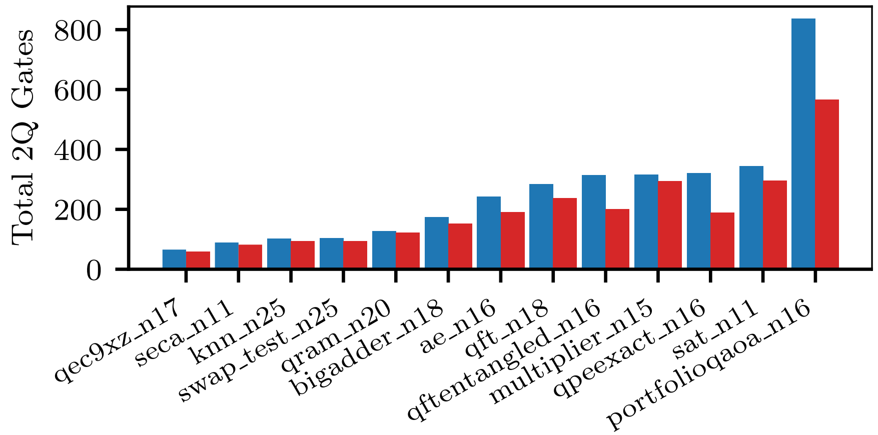


Gate count results

- For the Heavy-Hex topology
  - Average depth decrease of **31.19%**
  - Average total gate decrease of **16.97%**







- Use as a Qiskit Transpiler Plugin

Usage

```
from qiskit.transpiler import CouplingMap
coupling_map = CouplingMap.from_grid(6, 6)
```

1. Use as a Qiskit-Plugin

Integrate MIRAGE into your existing transpilation pipeline:

```
from qikist import transpile
mirage_qc = transpile(
                qc, # input circuit
                optimization_level = 3, # default: Qiskit's highest level
                coupling_map=coupling_map,
                basis_gates= ["u", "xx_plus_yy", "id"],
                routing_method="mirage",
                layout_method="sabre_layout_v2",
)
```

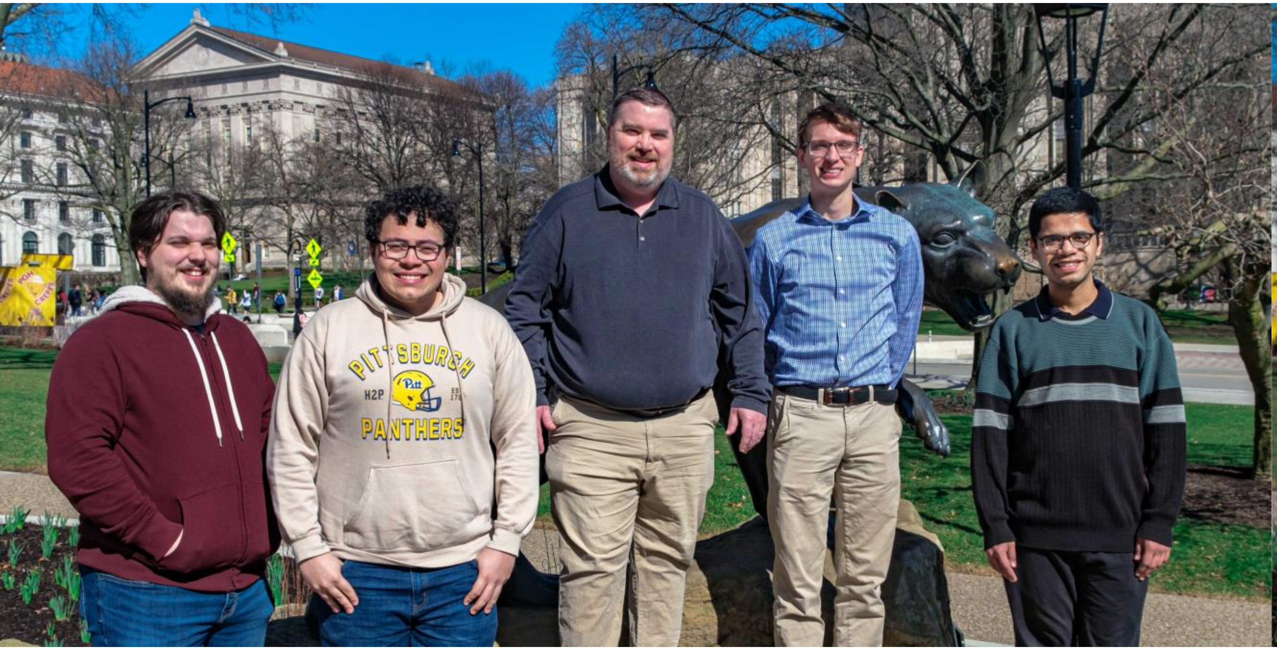
- Software optimizations:
  - Depth post-selection criteria
  - Variable mirror acceptance thresholds
  - Fast block consolidate w/ coord caching

https://github.com/Pitt-JonesLab/mirror-gates

Conclusion

- Parallel-drive basis **decreases circuit duration by 17.84%**
- Mirror gates with approximate decomposition **reduce infidelity by 9%**
- Heavy-Hex circuit benchmarks, **decrease depth by 31.19%** compared to Qiskit

McKinney, et al. **arXiv:2308.03874** (2023)



evm9.dev