Inferring Program Structure with Parameterized Circuit Ansätze

Mathias Weiden, Ed Younis, Justin Kalloor, John Kubiatowicz, Costin Iancu

BQSKit

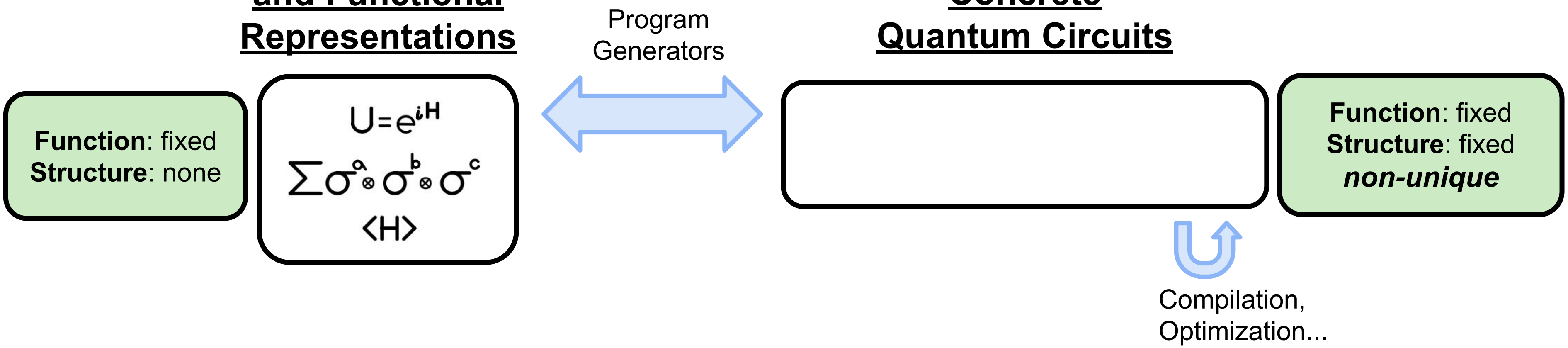
⟨AIDE|QC⟩

U.S. DEPARTMENT OF
ENERGY
Office of Science

BERKELEY LAB
Bringing Science Solutions to the World

Mathematical and Functional Representations

Concrete Quantum Circuits



Function: fixed
Structure: none

$U=e^{iH}$

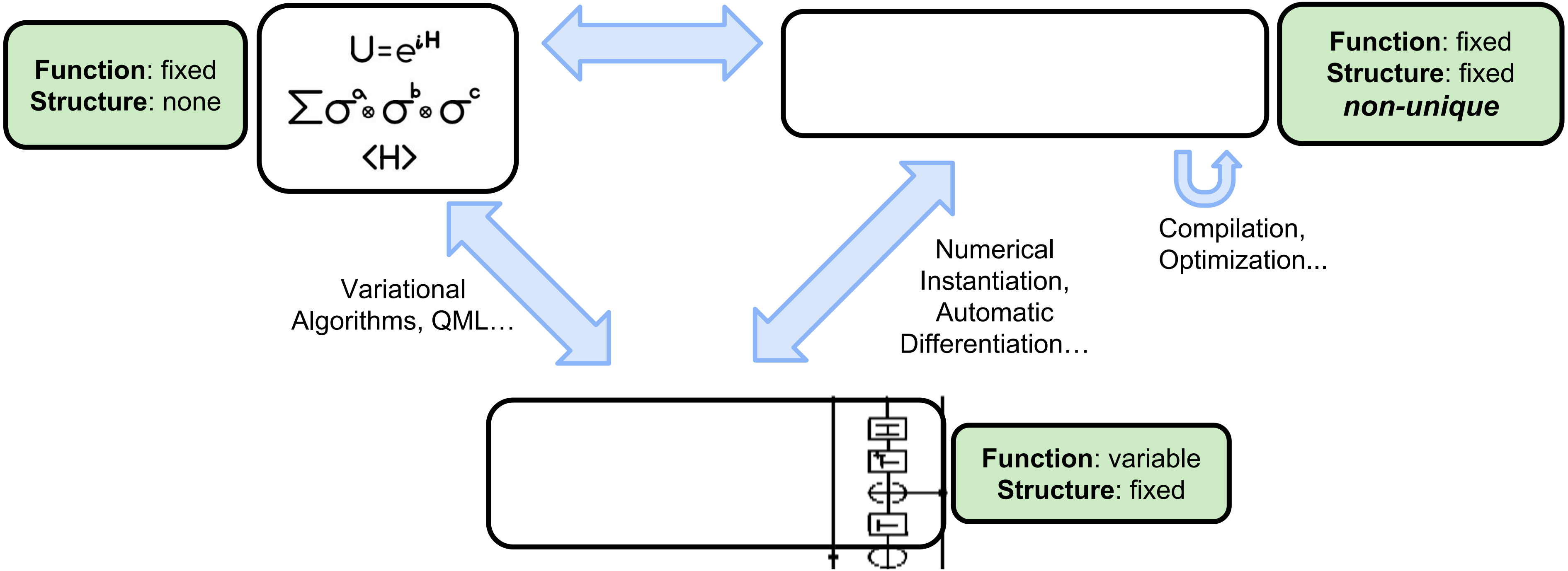
$\sum \sigma^a \otimes \sigma^b \otimes \sigma^c$

$\langle H \rangle$

Program Generators

Function: fixed
Structure: fixed
non-unique

Compilation, Optimization...



Mathematical and Functional Representations

Function: fixed
Structure: none

$U=e^{iH}$

$\sum \sigma^a \otimes \sigma^b \otimes \sigma^c$

$\langle H \rangle$

Program Generators

Concrete Quantum Circuits

Function: fixed
Structure: fixed
non-unique

Compilation, Optimization...

Numerical Instantiation, Automatic Differentiation…

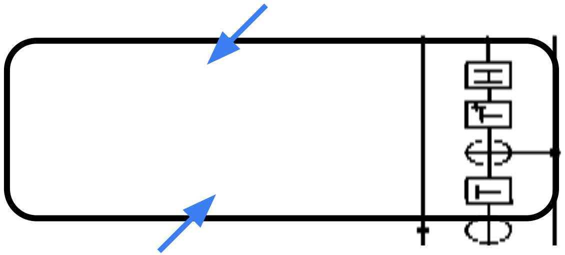
Variational Algorithms, QML…

Parameterized Circuit Ansätze

Function: variable
Structure: fixed

Parameterized Circuit Ansätze

- Fixed structure and variable function
- Maps parameters to quantum programs
- Consider parameterized single-qubit and fixed multi-qubit gates
- Ansätze are extremely expressive

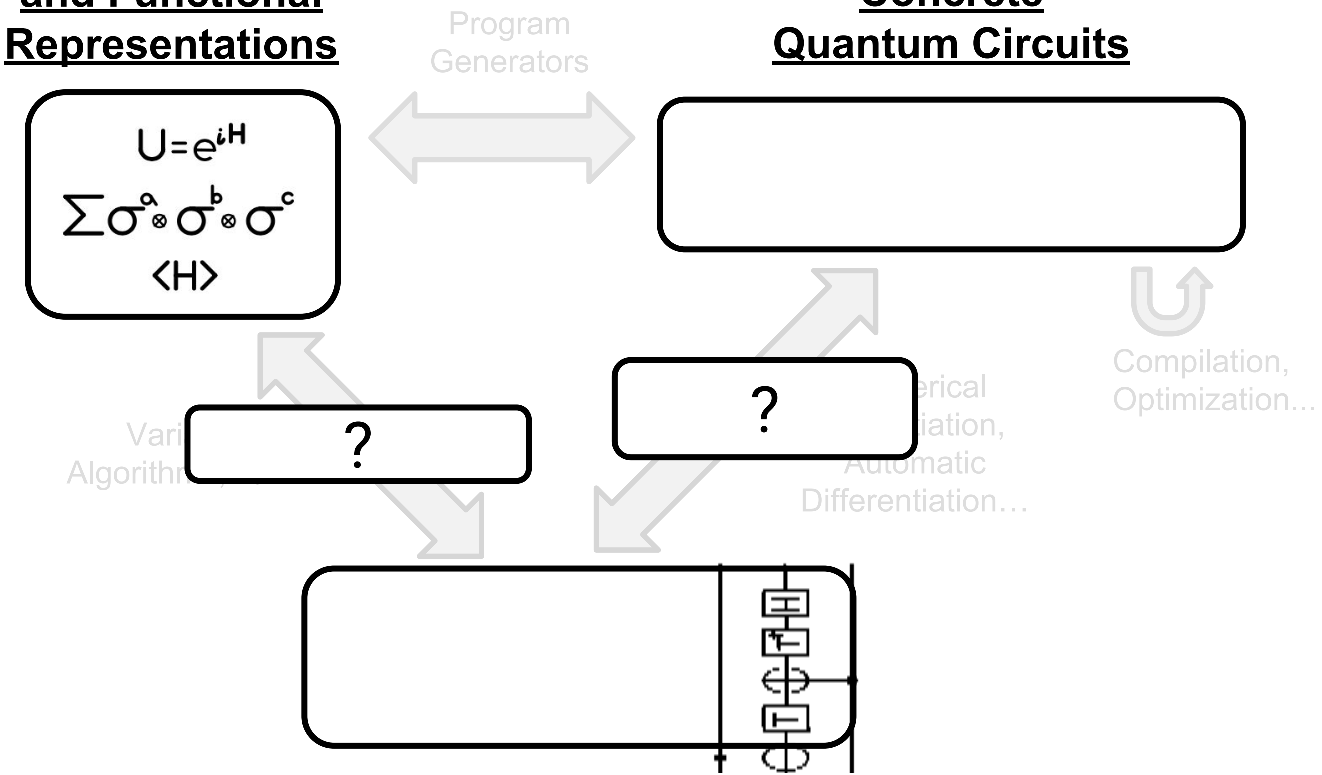


Example circuit ansatz

1. *Can we find minimal ansätze for programs?*
2. *Can the ansatz that implements a program be predicted?*
3. *Are ansätze portable across gate sets?*

Mathematical and Functional Representations

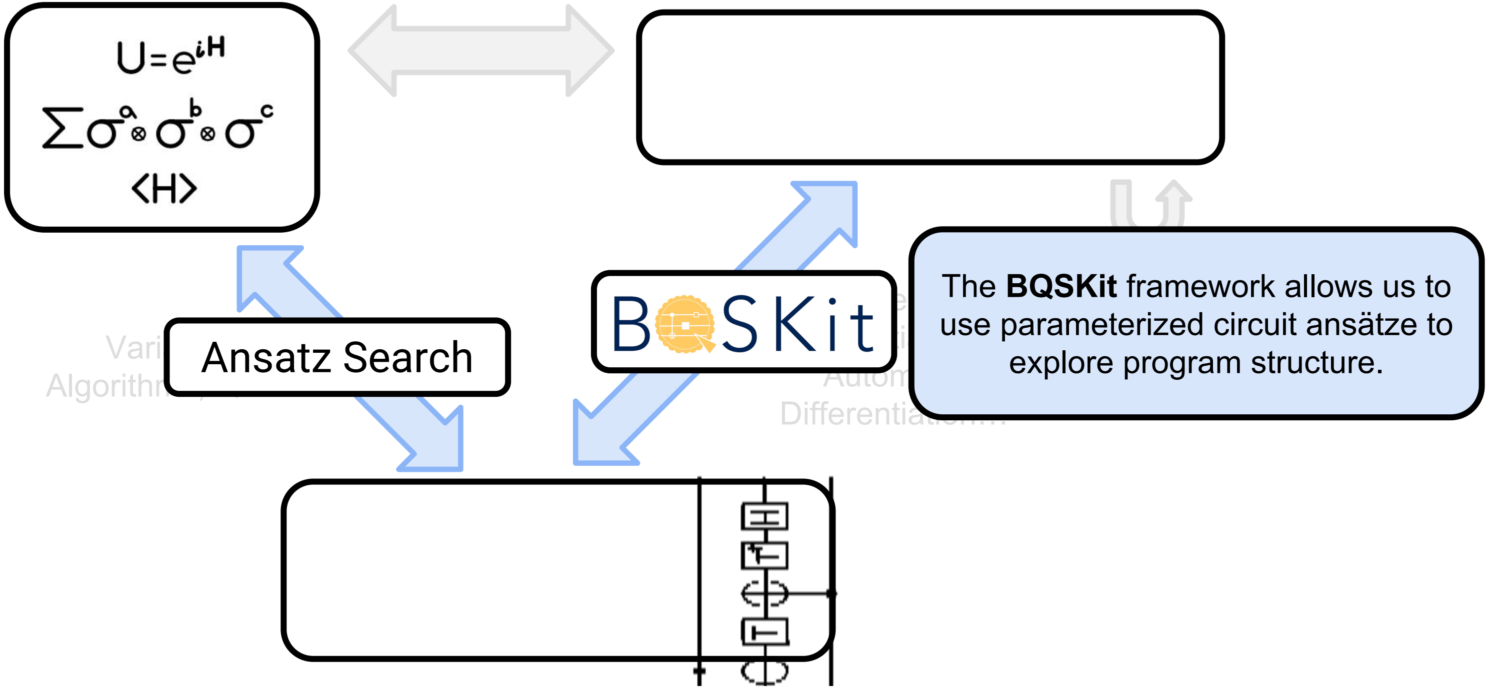
Concrete Quantum Circuits



Parameterized Circuit Ansätze

Mathematical and Functional Representations

Concrete Quantum Circuits



The **BQSKit** framework allows us to use parameterized circuit ansätze to explore program structure.

Parameterized Circuit Ansätze

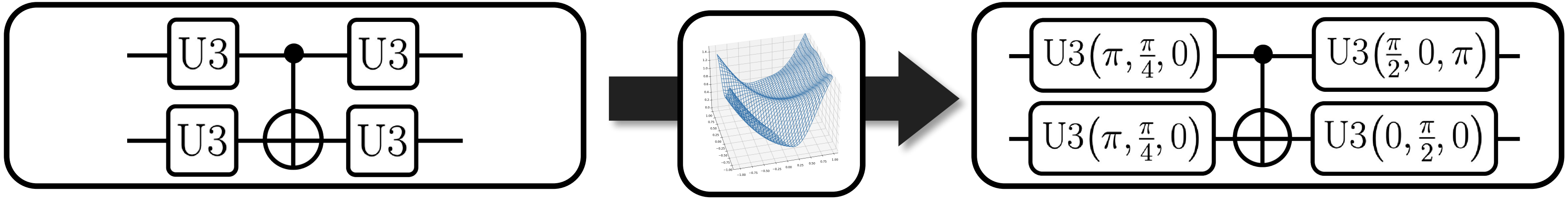
Numerical Circuit Instantiation

Instantiation: Given a parameterized quantum circuit $C : \mathbb{R}^k \mapsto U(N)$ and a target unitary $V \in U(N)$, solve

$$\arg\min_{\theta}(I - \underset{\text{(error)}}{V^\dagger C(\theta)})$$

Parameterized Circuit

Instantiated Circuit


U3
U3
U3
U3
$\mathrm{U3}(\pi, \frac{\pi}{4}, 0)$
$\mathrm{U3}(\frac{\pi}{2}, 0, \pi)$
$\mathrm{U3}(\pi, \frac{\pi}{4}, 0)$
$\mathrm{U3}(0, \frac{\pi}{2}, 0)$

Single qubit gates have no fixed parameter settings

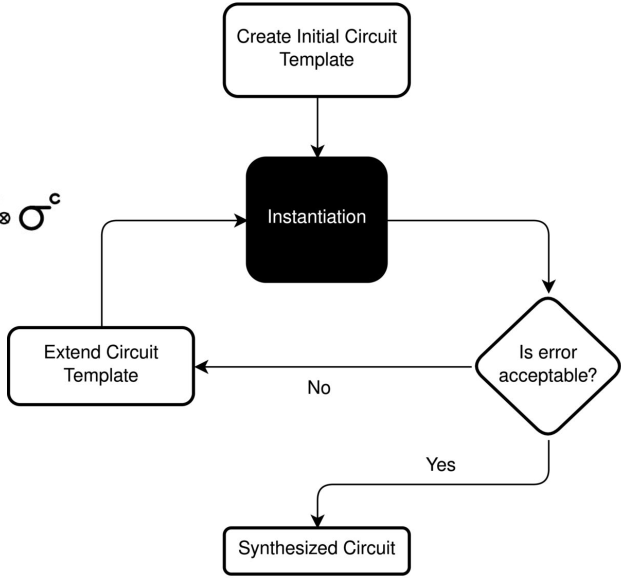
Numerical Optimization

Single qubit gate parameters are real numbers

Kukliansky, ***QFactor: A Domain-Specific Optimizer for Quantum Circuit Instantiation***

Numerical Instantiation Flows with BQSKit

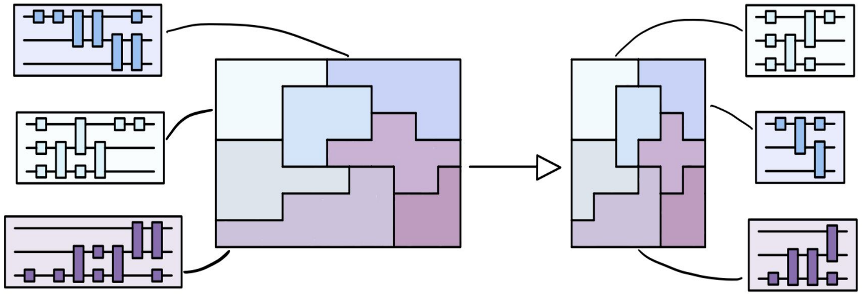
- BQSKit supports specifying program functionality with
  - **Unitary matrices** $U=e^{iH}$
  - Hamiltonians (sums of Hermitian operators) $\sum \sigma^a \otimes \sigma^b \otimes \sigma^c$
  - Loss functions to minimize $\langle H \rangle$
  - Parameters to a parameterized quantum circuit ansatz $\vec{\theta} = [\theta_1 \dots \theta_k]$
- BQSKit's general instantiation workflow enables the optimization, synthesis, and transpilation of quantum circuits



Younis, ***Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation***

Instantiation and Scaling

- Instantiation/solving for parameters has width/depth limits no matter the encoding used
    - Recent work (QFactor) shows how to scale instantiation to 12 qubits!
- Wide unitaries/circuits must be broken into manageable **panels**
    - Each panel is associated with a **partitioned unitary**
- This method can handle circuits with 1000s qubits



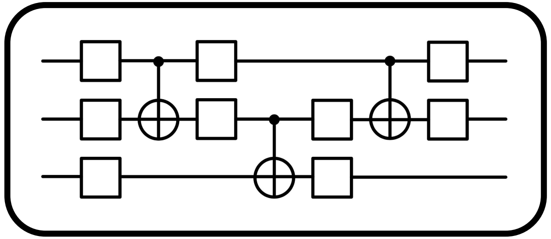
Wu, ***Reoptimization of Quantum Circuits via Hierarchical Synthesis***

1. Can we find minimal ansätze for programs?

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Program specified as Unitary

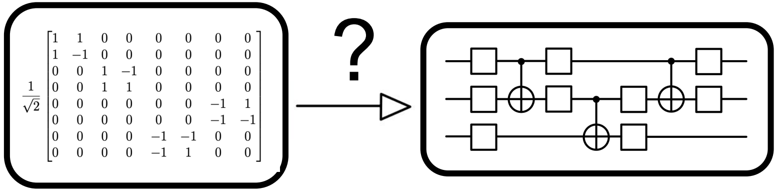
$\xrightarrow{?}$



Lowest Gate Count Circuit Ansatz

Approach

1. Collect a large dataset of unitaries
   - Consider unitaries from partitioned circuits
   - Same algorithms of various widths
   - Different algorithms
2. Enumerate circuit ansätze up to a certain depth
3. Try instantiating each ansatz given each unitary

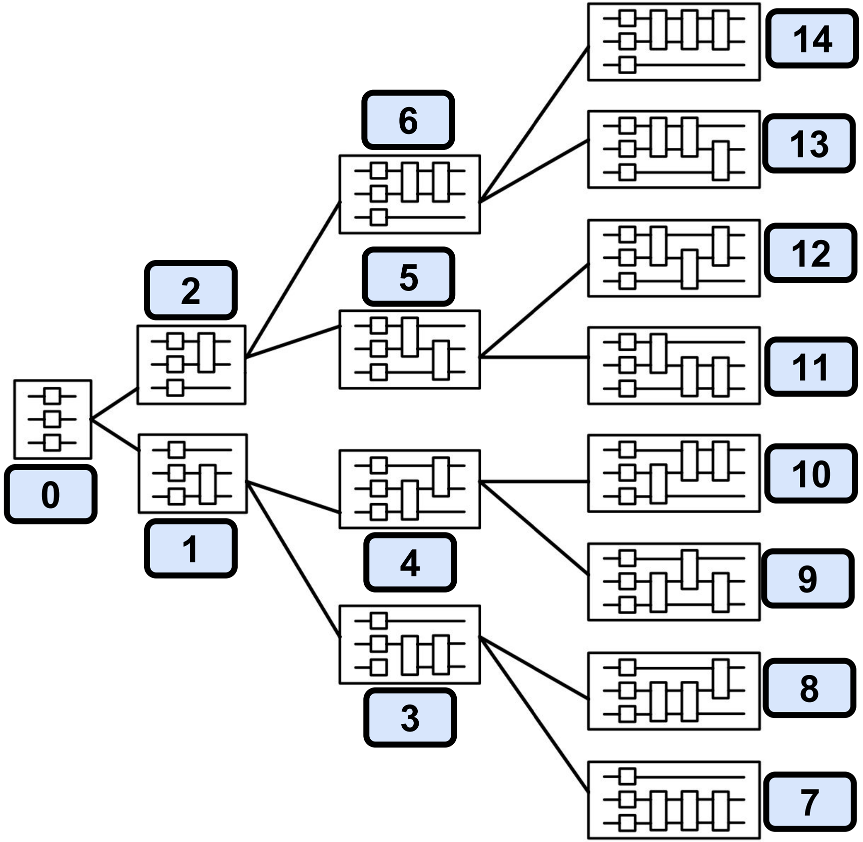


$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

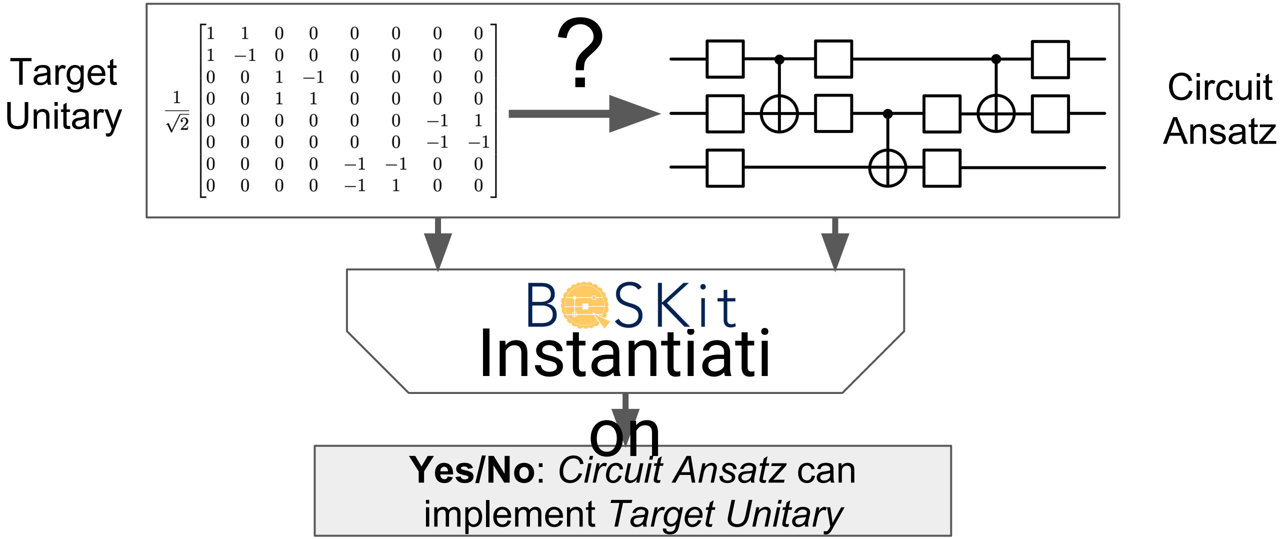
1. Can we find minimal ansätze for programs?

Enumerating Ansätze

- Consider a tree of possible circuit ansätze
- Assign an integer to each circuit ansatz
- Stop enumeration at a desired depth to keep finite output space
- Works because subcircuits from circuits are relatively simple



Instantiating each Ansatz

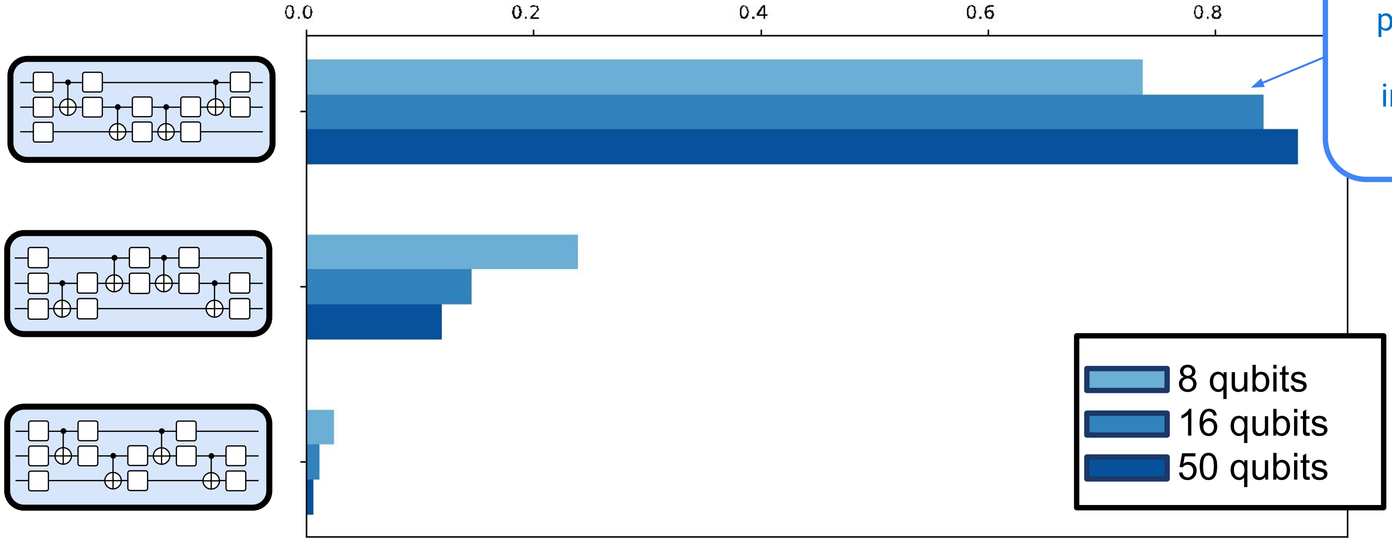


$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Yes/No: *Circuit Ansatz* can implement *Target Unitary*

Patterns Occur when Algorithms Scale

Histogram of Circuit Ansätze in TFIMs with 8, 16, and 50 Qubits



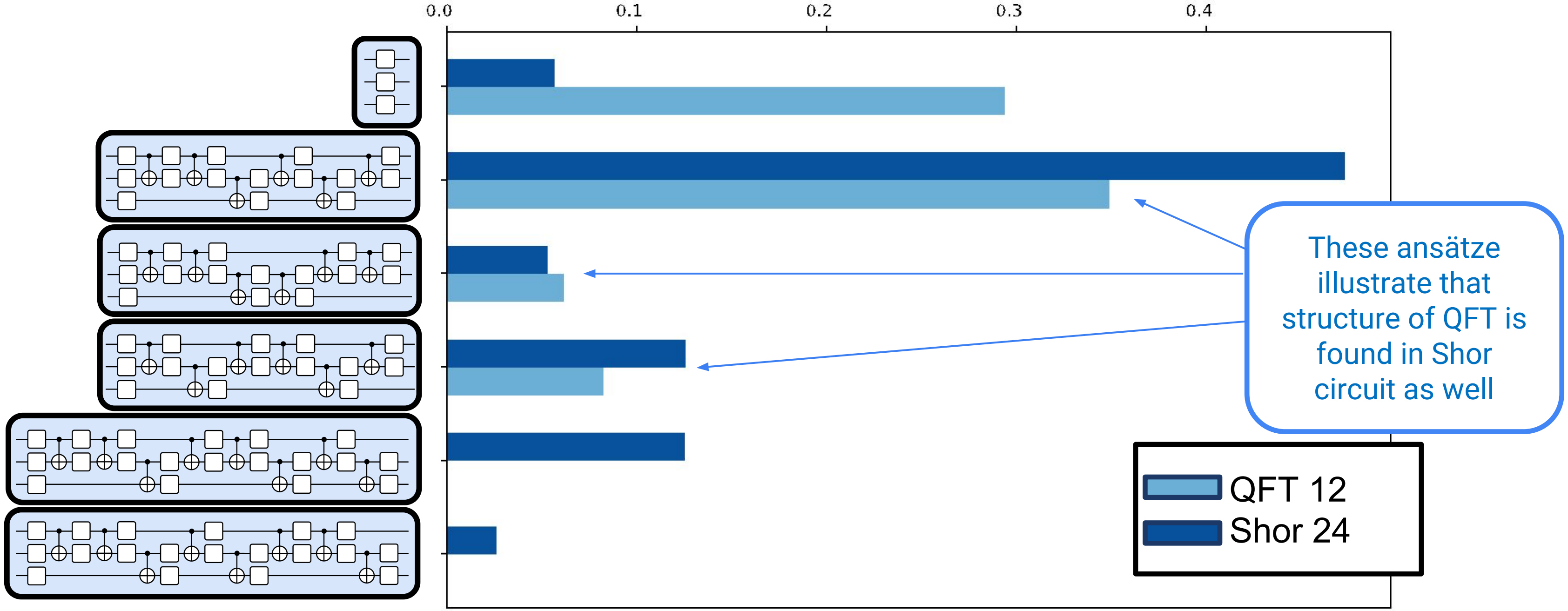
TFIM 16 has 99 partitions. 82/99 partitions are implemented by this ansatz

Three ansätze account for 99% of ansätze in TFIM circuits of various widths

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Patterns Occur When Algorithms are Composed

Histogram of Circuit Ansätze in QFT 12 and Shor 24



These ansätze illustrate that structure of QFT is found in Shor circuit as well

QFT 12 *is contained in* Shor 24: QFT and IQFT appear 16 times

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

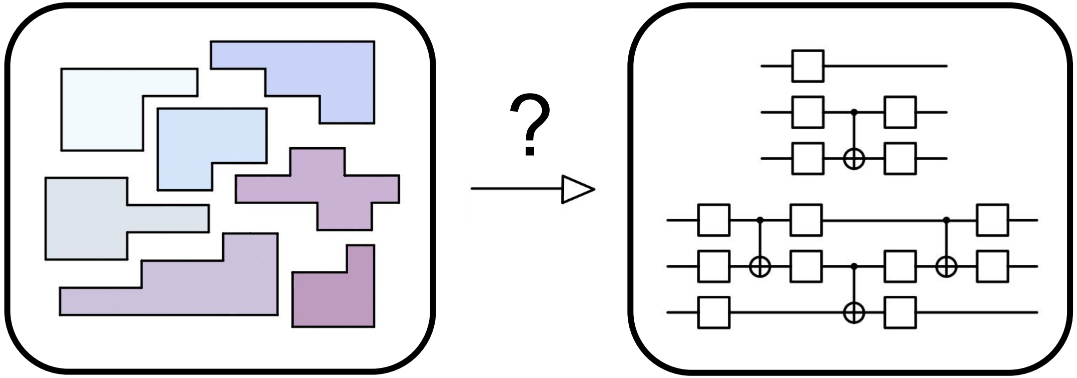
Patterns Appear in Circuit Ansätze

1. Can we find minimal ansätze for programs?

***Yes**, and **patterns** in structure can be observed for unitaries taken from **partitioned quantum circuits**.*

- Partitioned unitaries:
  - **Same circuit** family but **different widths** share ansätze
  - Ansätze from **different circuits** capture **containment** relationships
- No does not work for random unitaries

2. Can the ansatz that implements a program be predicted?



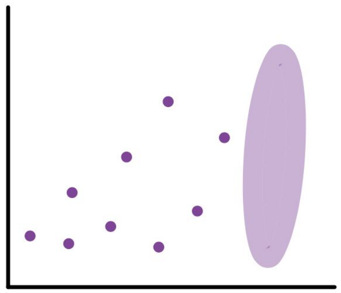
A Set of Unitaries from Partitioned Circuit Panels

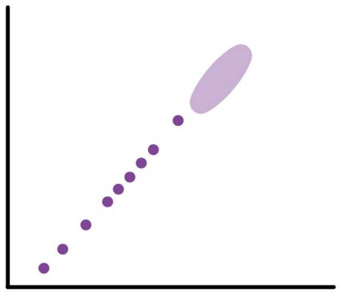
A Set of Circuit Ansätze

When Can You Expect to Make Predictions?

Example:

For these two datasets, where would you expect the next data point to fall?



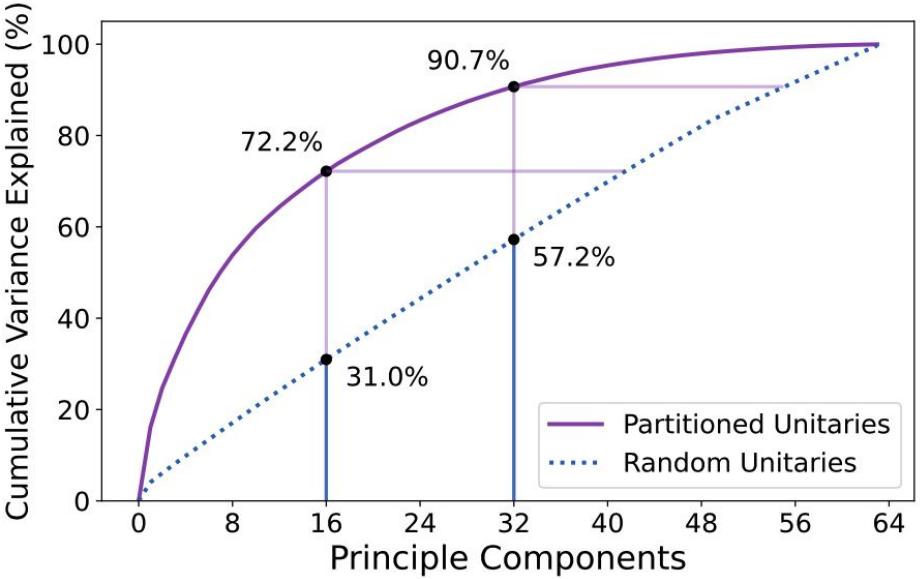


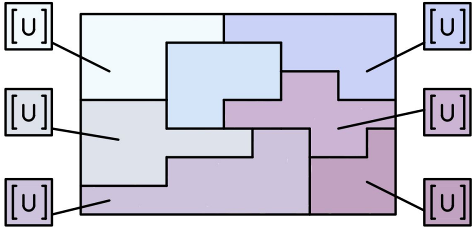
Rules of Thumb:

1. Low dimensional patterns make data predictable
2. Low dimensional data are compressible

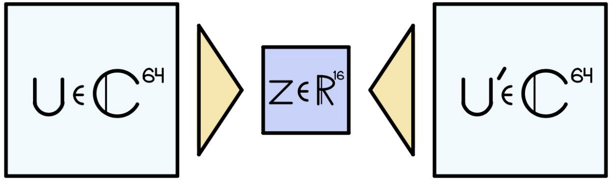
Patterns in Unitaries

- **Principal Component Analysis** reveals low dimensionality of unitaries of interest
- Implies learning patterns in unitaries is possible





Partitioned unitaries are taken from circuit panels

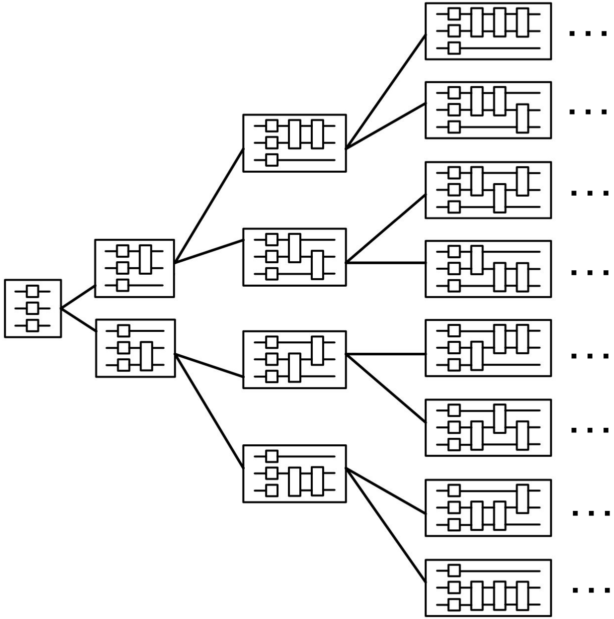


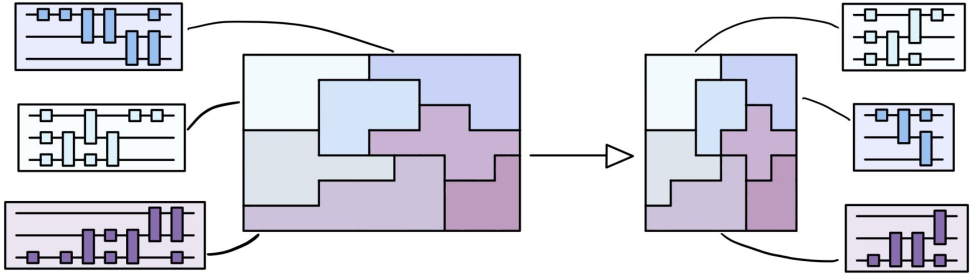
PCA quantifies how much we can compress these unitaries

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Using Patterns: Scaling Synthesis

- **Synthesis = circuit ansatz search + instantiation**
- Navigating the search tree must be done strategically
- There are many possible circuits to try





Wide unitaries/circuits must be broken into manageable pieces.

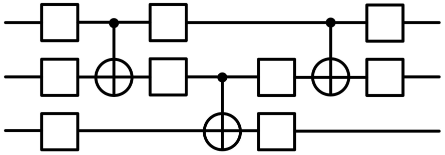
Davis, ***Towards Optimal Topology Aware Quantum Circuit Synthesis***

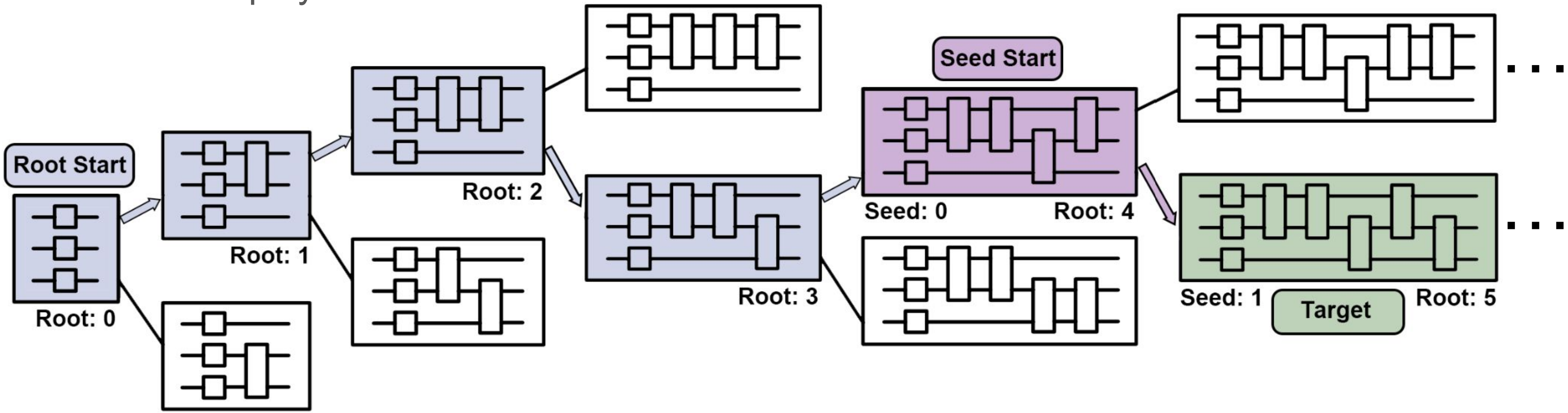
Seeded Synthesis

- Start synthesis closer to the solution circuit
- Alleviates the expensive tree search component of full bottom-up synthesis

Find a mapping from unitaries to circuits

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix} \longrightarrow$$

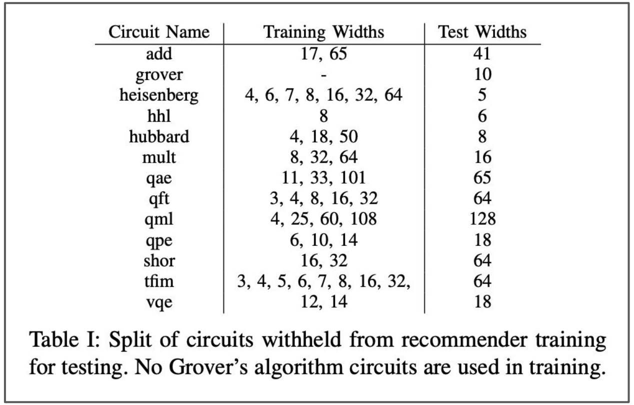




Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

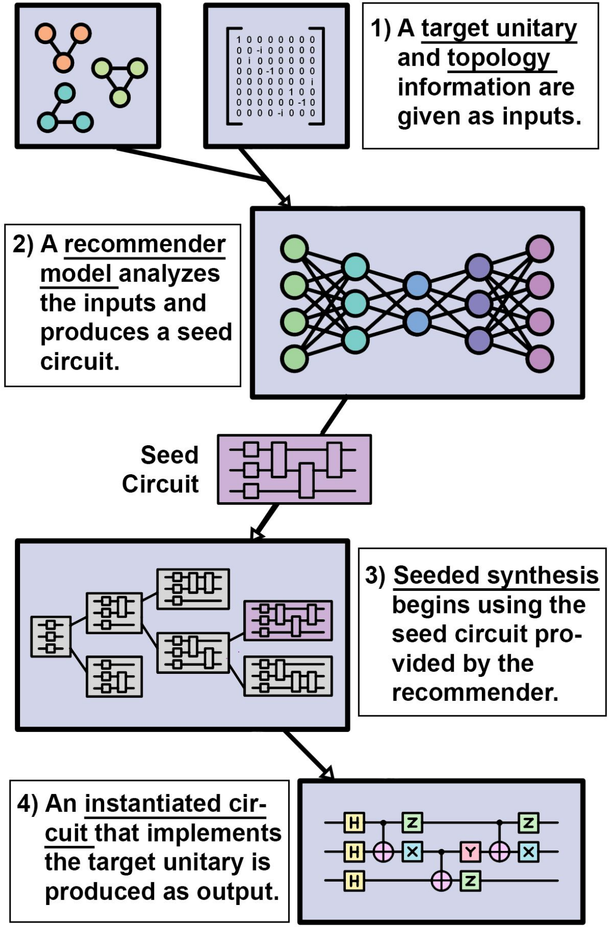
Accelerating Synthesis

- QSeed: A seeded synthesis algorithm that uses machine learning to predict good seed circuits
- **Paper:** ***Improving Quantum Circuit Synthesis with Machine Learning***



Circuit Name	Training Widths	Test Widths
add	17, 65	41
grover	-	10
heisenberg	4, 6, 7, 8, 16, 32, 64	5
hhl	8	6
hubbard	4, 18, 50	8
mult	8, 32, 64	16
qae	11, 33, 101	65
qft	3, 4, 8, 16, 32	64
qml	4, 25, 60, 108	128
qpe	6, 10, 14	18
shor	16, 32	64
tfim	3, 4, 5, 6, 7, 8, 16, 32,	64
vqe	12, 14	18

Table I: Split of circuits withheld from recommender training for testing. No Grover's algorithm circuits are used in training.



1) A target unitary and topology information are given as inputs.

2) A recommender model analyzes the inputs and produces a seed circuit.

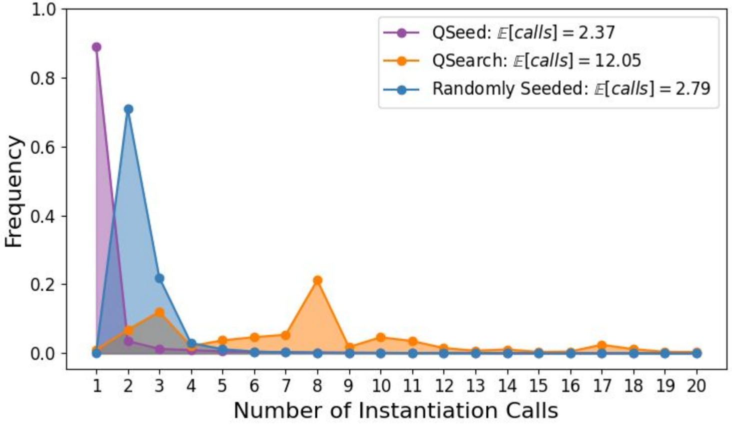
Seed Circuit

3) Seeded synthesis begins using the seed circuit provided by the recommender.

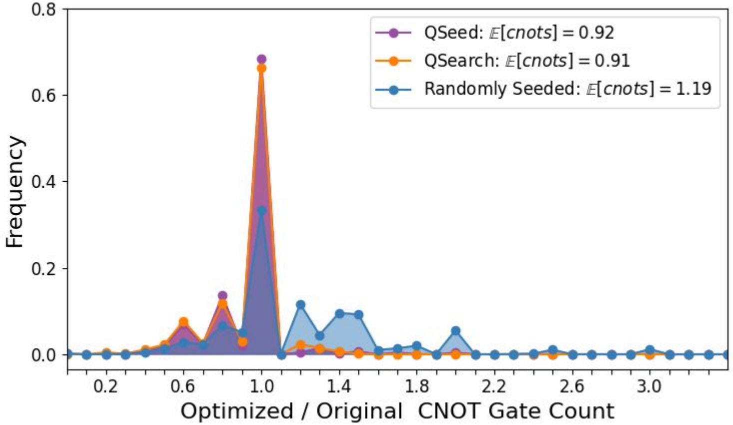
4) An instantiated circuit that implements the target unitary is produced as output.

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Accelerating Synthesis



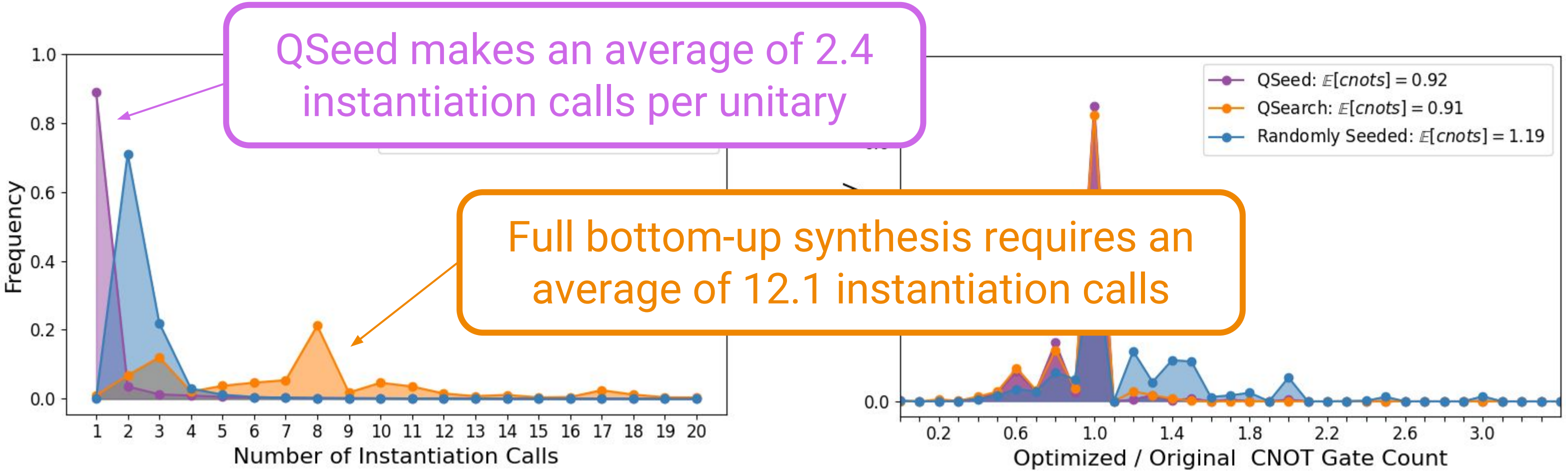
Speedup: most synthesis runs require only one instantiation call



Solution quality: gate counts very closely match optimal implementation

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Accelerating Synthesis



QSeed makes an average of 2.4 instantiation calls per unitary

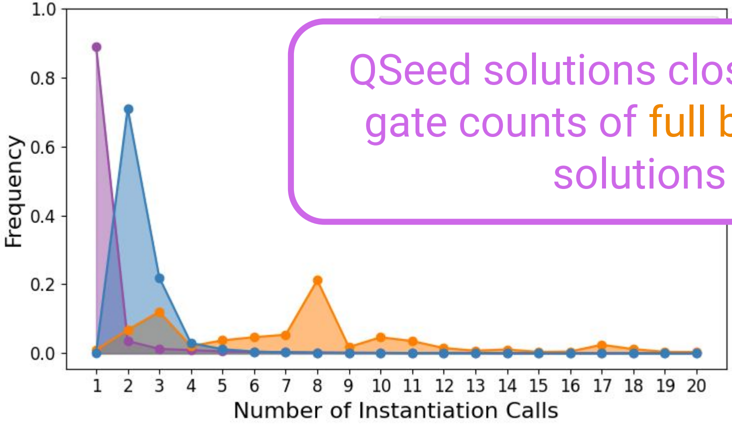
Full bottom-up synthesis requires an average of 12.1 instantiation calls

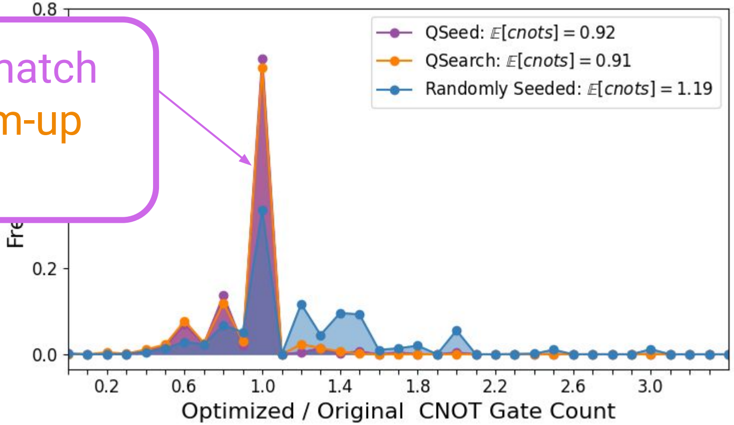
Speedup: most synthesis runs require only one instantiation call

Solution quality: gate counts very closely match optimal implementation

Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Accelerating Synthesis





QSeed solutions closely match gate counts of full bottom-up solutions

Speedup: most synthesis runs require only one instantiation call

Solution quality: gate counts very closely match optimal implementation

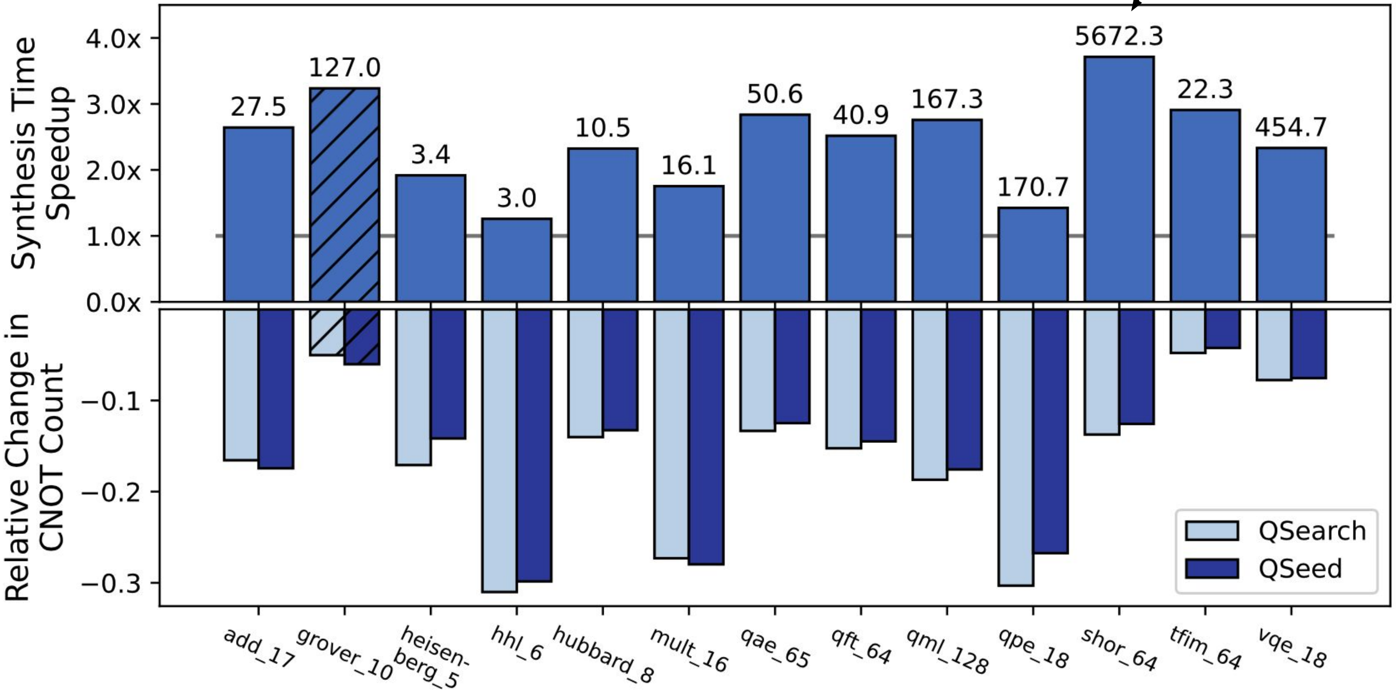
Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Accelerating Synthesis

Shor 64 is synthesized **>3x faster**! Uses **10.3% fewer CNOTs** than Qiskit!

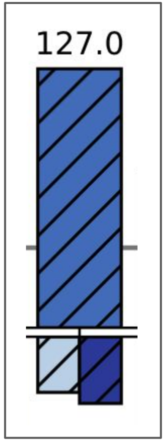
Average speedup of 2.4x

Average CNOT count difference of 1.0%

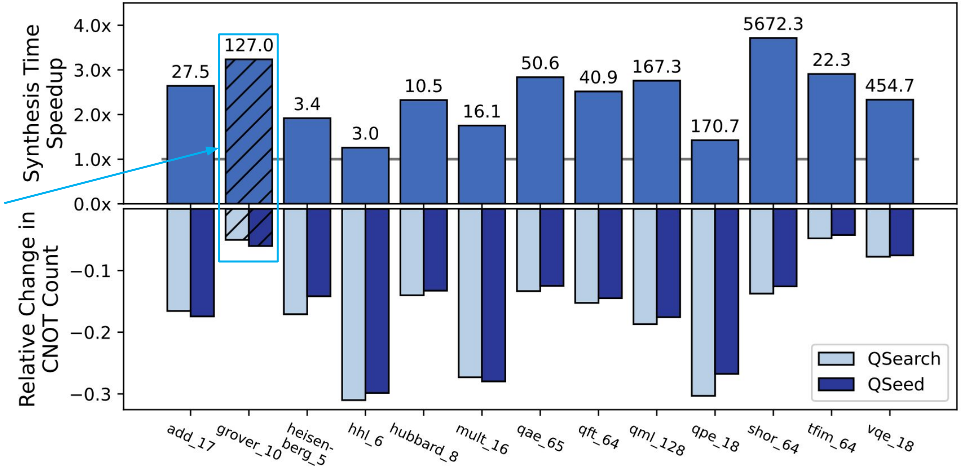


Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

Generalizing to Unseen Circuits

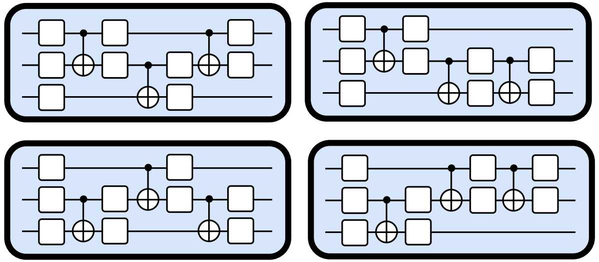


Grover 10



Time savings and comparable solution quality maintained for unseen circuits

- Simple structures dominate the Grover 10 circuit
- Examples of these simple structures are scattered throughout other circuits

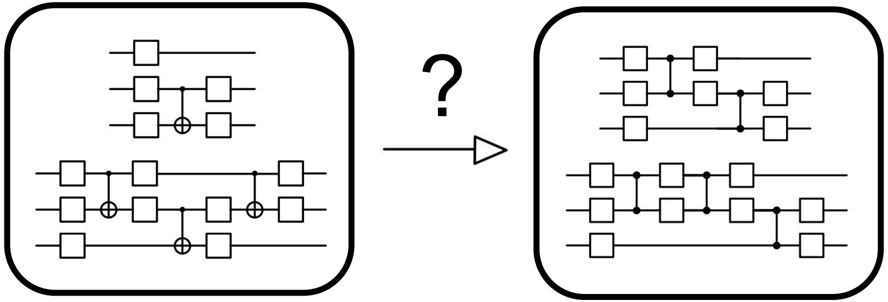


Weiden, ***Improving Quantum Circuit Synthesis with Machine Learning***

2. Can the ansatz that implements a program be predicted?

Yes. *We can predict ansätze that implement* ***unitaries*** *taken from* ***partitioned quantum circuits****. These results can* ***generalize*** *to unseen partitioned unitaries.*

3. Are ansätze portable across gate sets?

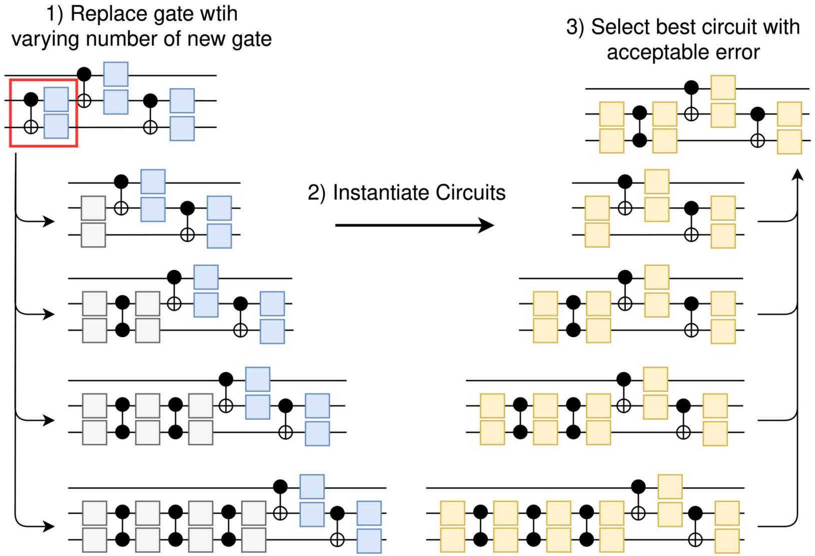


Ansätze Expressed with CX Gates

Ansätze Expressed with CZ Gates

BQSKit Retargeting and Transpilation

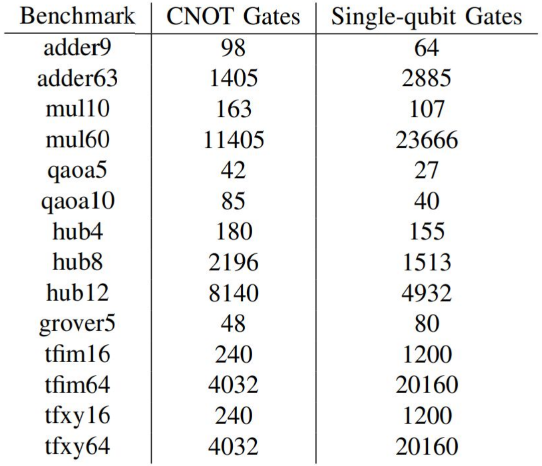
- More generic than 1-1 rules (KAK, etc.)
- Works for gate sets with multiple entangling gates
- For a given ansatz **transpilation** must choose between ***O(g^{x+1})* ansätze in the new gate set**
  g: 2-qubit gates in ansatz
  x: 2-qubit gates to implement arbitrary unitary
- **Are ansätze/structures preserved when gate set changes?**


1) Replace gate wtih varying number of new gate
2) Instantiate Circuits
3) Select best circuit with acceptable error

Younis, ***Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation***

BQSKit Transpiled Circuits are State of the Art

- **Effective due to global transformations**
- Produces circuits with an **average of 12% fewer two-qubit gates**
- Up to **51% reduction** of two-qubit gates
- Existing commercial compilers have many conversion limitations
- **Portable and flexible (N-qubit gates, qudits)**

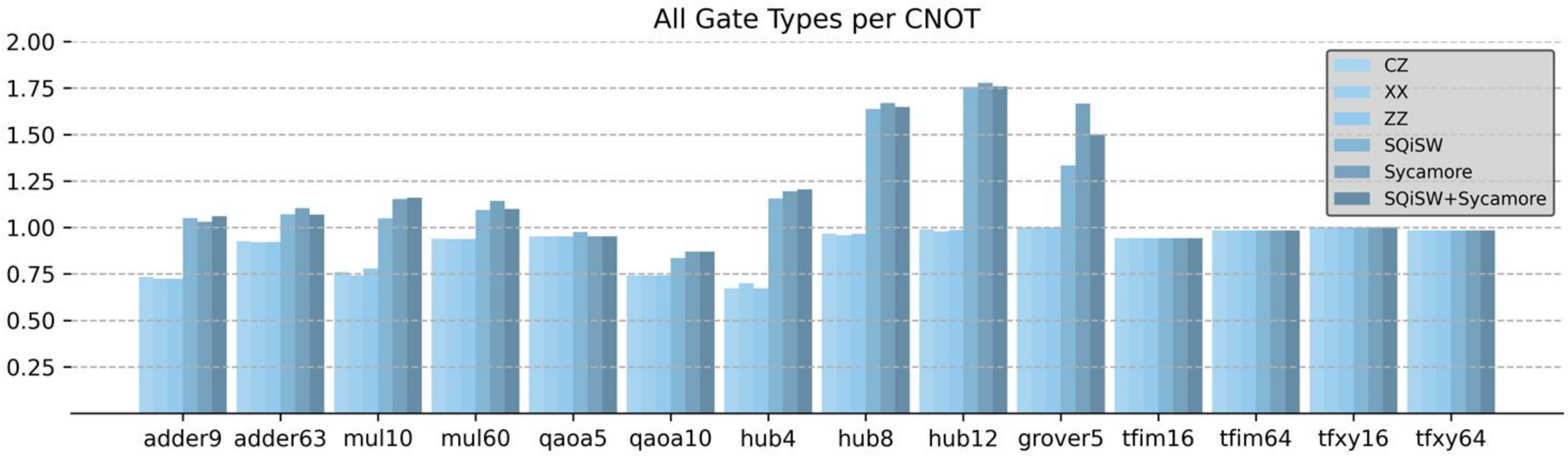


Benchmark	CNOT Gates	Single-qubit Gates
adder9	98	64
adder63	1405	2885
mul10	163	107
mul60	11405	23666
qaoa5	42	27
qaoa10	85	40
hub4	180	155
hub8	2196	1513
hub12	8140	4932
grover5	48	80
tfim16	240	1200
tfim64	4032	20160
tfxy16	240	1200
tfxy64	4032	20160

- BQSKit framework and the default instantiator configured with four multi-starts and 10^{-10} success threshold
- Retargetting converts **CNOT**s to **CZ**, **XX**, **ZZ**, $\sqrt{\textbf{iSwap}}$, and **SYC**
- Compared against Cirq, Qiskit, and Tket

Younis, ***Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation***

Algorithms Have Gate Set Preferences



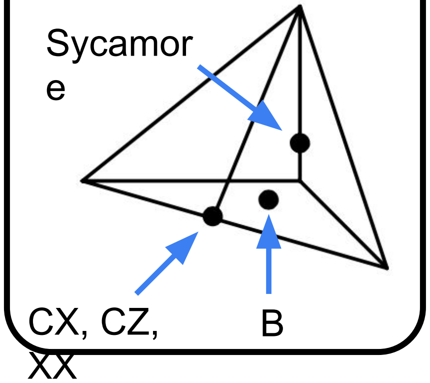
Can statements can we make about transpiled circuit ansätze?

Younis, ***Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation***

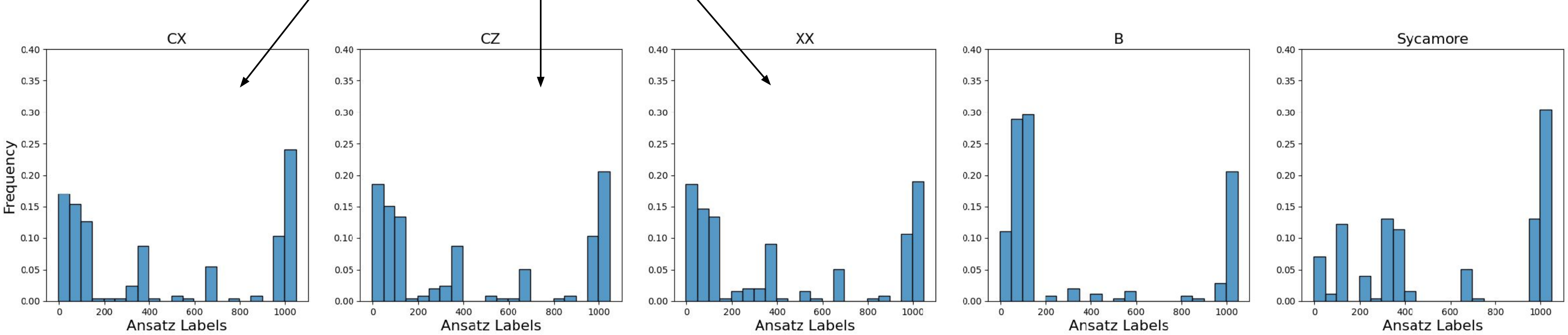
Retargeting Ansätze

- Similarities between ansatz distributions appear to be linked to gate location in the **Weyl Chamber**

Various Gates in the Weyl Chamber



When Weyl Chamber location is the same, ansätze in different gate sets are similar

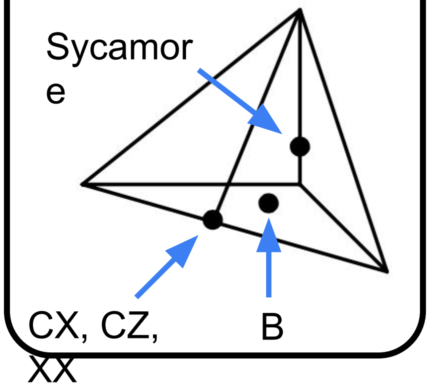


Histograms of Ansätze for Mult 60 Unitaries

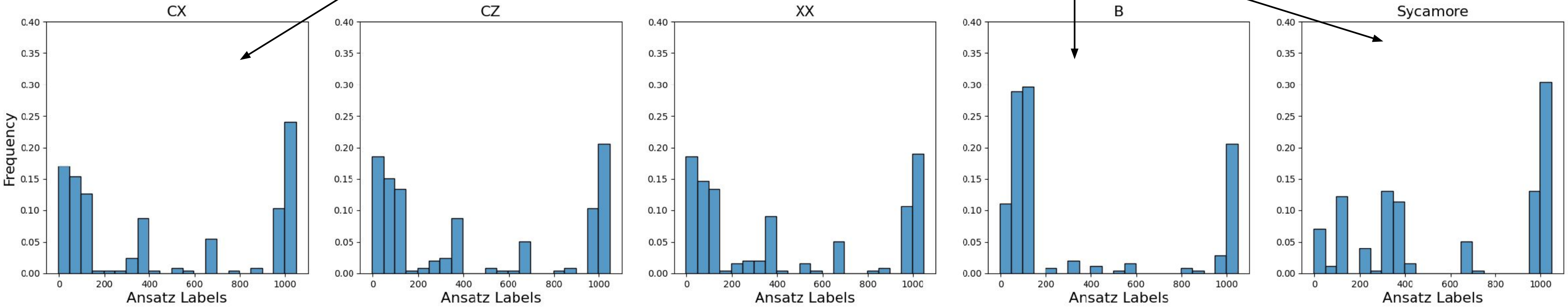
Retargeting Ansätze

- Similarities between ansatz distributions appear to be linked to gate location in the **Weyl Chamber**

Various Gates in the Weyl Chamber



Otherwise map is less straightforward, but empirically 1-to-few (far smaller than exponential upper bound)



Histograms of Ansätze for Mult 60 Unitaries

3. Are ansätze portable across gate sets?

Yes. *When* ***non-local*** *parameters of gates are the same, there is a* ***1-to-1*** *mapping between ansätze. Otherwise there is a* ***1-to-few*** *mapping.*

Conclusions

1. ***Can we find minimal ansätze for programs?***

Yes, and **patterns** in structure can be observed for unitaries taken from **partitioned quantum circuits**.

2. Can the ansatz that implements a program be predicted?

Yes. We can predict ansätze that implement **unitaries** taken from **partitioned quantum circuits**. These results can **generalize** to unseen partitioned unitaries.

3. Are ansätze portable across gate sets?

Yes. When **non-local** parameters of gates are the same, there is a **1-to-1** mapping between ansätze. Otherwise there is a **1-to-few** mapping.

Thank You!

1. Mathias Weiden, Ed Younis, Justin Kalloor, John Kubiatowicz, Costin Iancu, “***Improving Quantum Circuit Synthesis with Machine Learning***”, Accepted, 2023 (QCE)
2. Alon Kukliansky, Ed Younis, Lukasz Cincio, Costin Iancu, “***QFactor: A Domain-Specific Optimizer for Quantum Circuit Instantiation*** “, Accepted, 2023 (QCE)
3. E. Younis, C. Iancu, “***Quantum Circuit Optimization and Transpilation via Parameterized Circuit Instantiation***,” in 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE, 2022, pp. 465-475.
4. E. Younis, K. Sen, K. Yelick, C. Iancu, “ ***QFast: Conflating Search and Numerical Optimization for Scalable Quantum Circuit Synthesis***,” in 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE, 2021, pp. 232-243. *(Best Paper Award)*
5. M. G. Davis, E. Smith, A. Tudor, K. Sen, I. Siddiqi, C. Iancu, ***“Towards Optimal Topology Aware Quantum Circuit Synthesis,”*** in 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE, 2020, pp. 223-234. *(Best Paper Award)*
6. X.C. Wu, M. G. Davis, F. Chong, C. Iancu, ***“Reoptimization of Quantum Circuits via Hierarchical Synthesis,”*** in 2021 IEEE International Conference on Rebooting Computing (ICRC), IEEE, 2021, pp. 35-46.

bqskit.lbl.gov