Full-stack Optimization of Quantum Chemistry Simulation

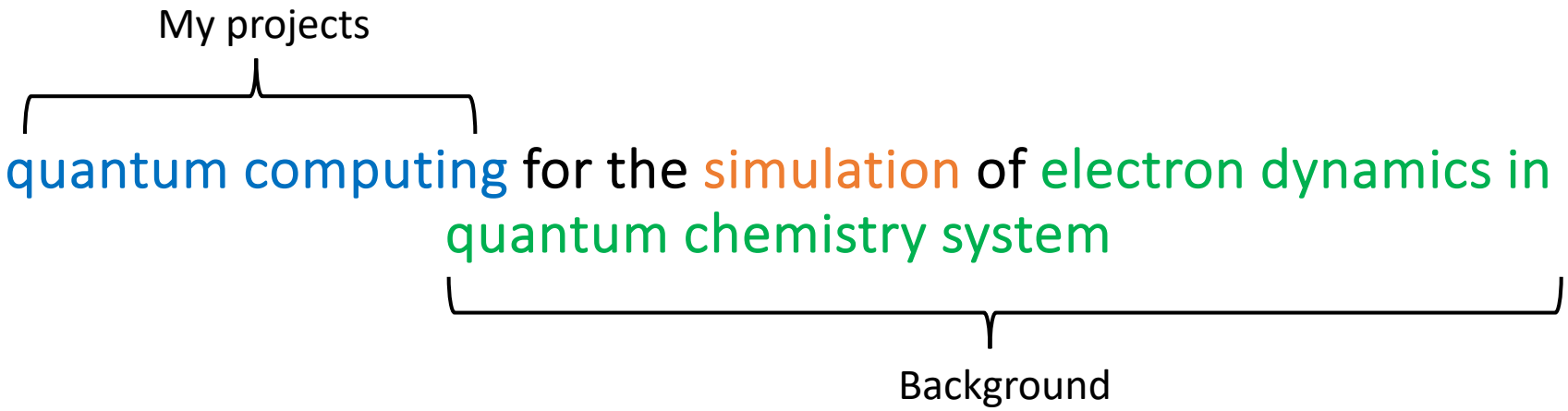
Qualification exam for Zirui Li

May 8th, 2025

Committee: Yipeng Huang, Eddy Z. Zhang, Mario Szegedy, Jingjin Yu

RUTGERS

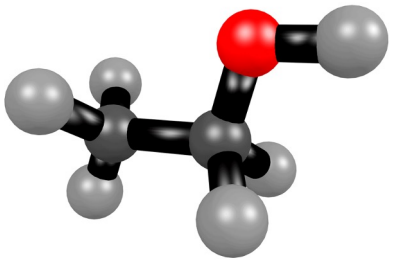
Full-stack Optimization of Quantum Chemistry Simulation



Quantum Chemistry Simulation

- Variational quantum eigensolver

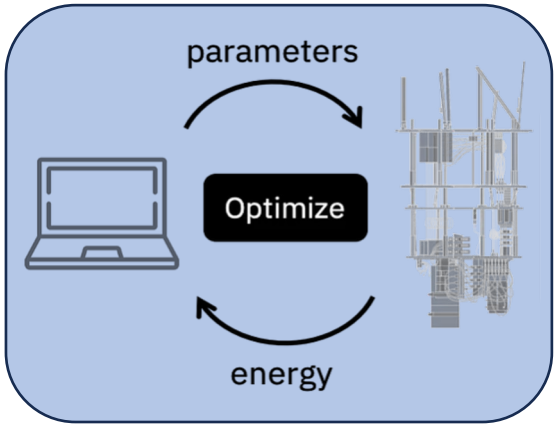
$$\hat{H}|\Psi\rangle = E_0|\Psi\rangle$$



Molecule

$\hat{H}$

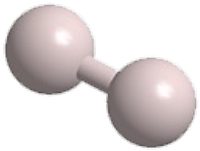
Hamiltonian matrix

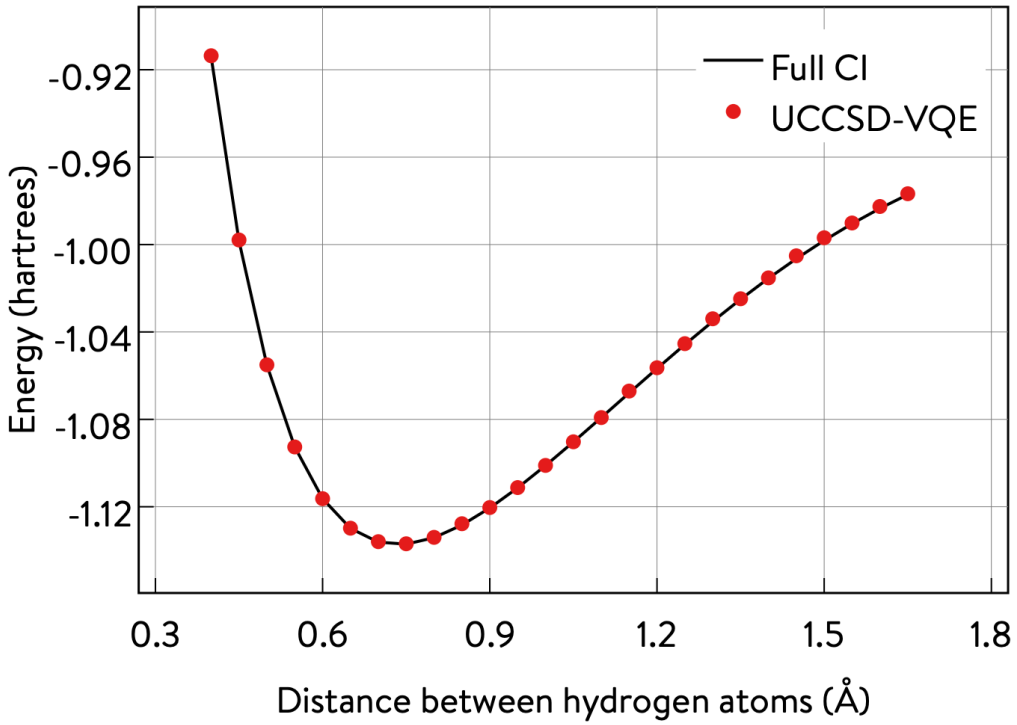


Variational quantum eigensolver

Quantum Chemistry Simulation

- Predicting molecular properties
  - Geometry, bond length, angles…

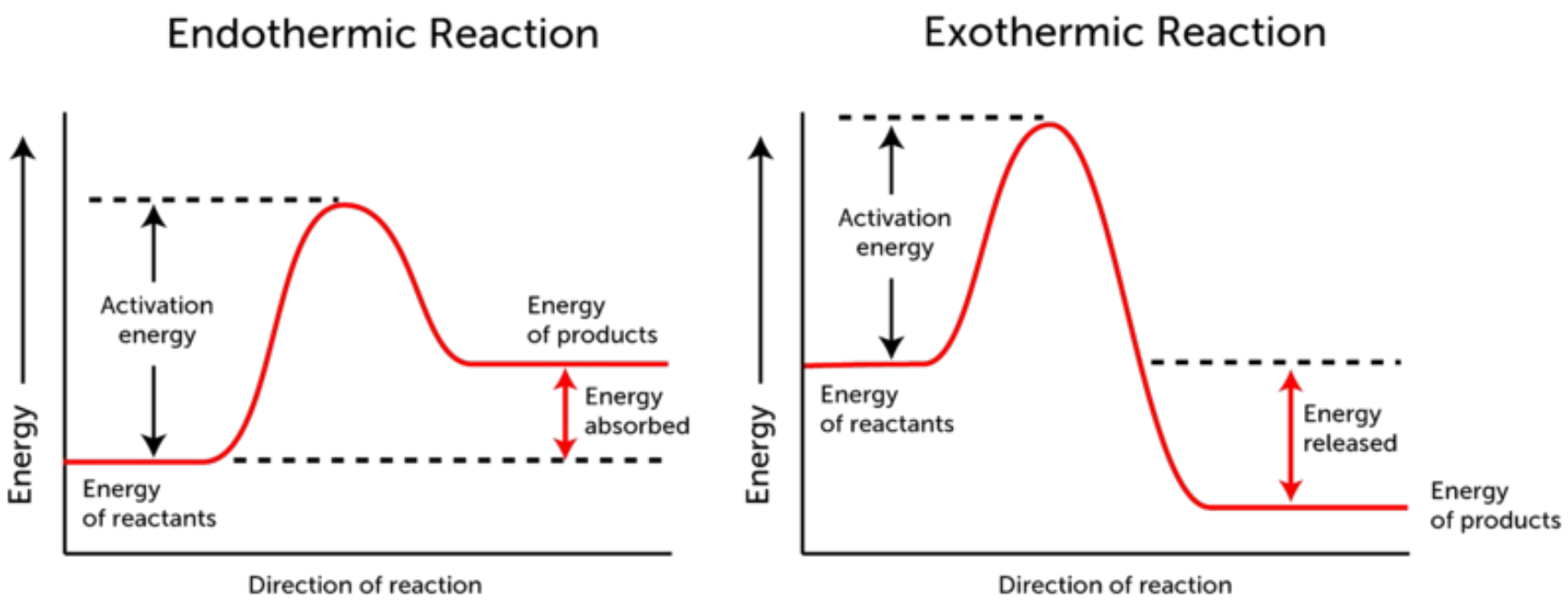




Credit:http://openqemist.1qbit.com/docs/vqe_microsoft_qsharp.html

Quantum Chemistry Simulation

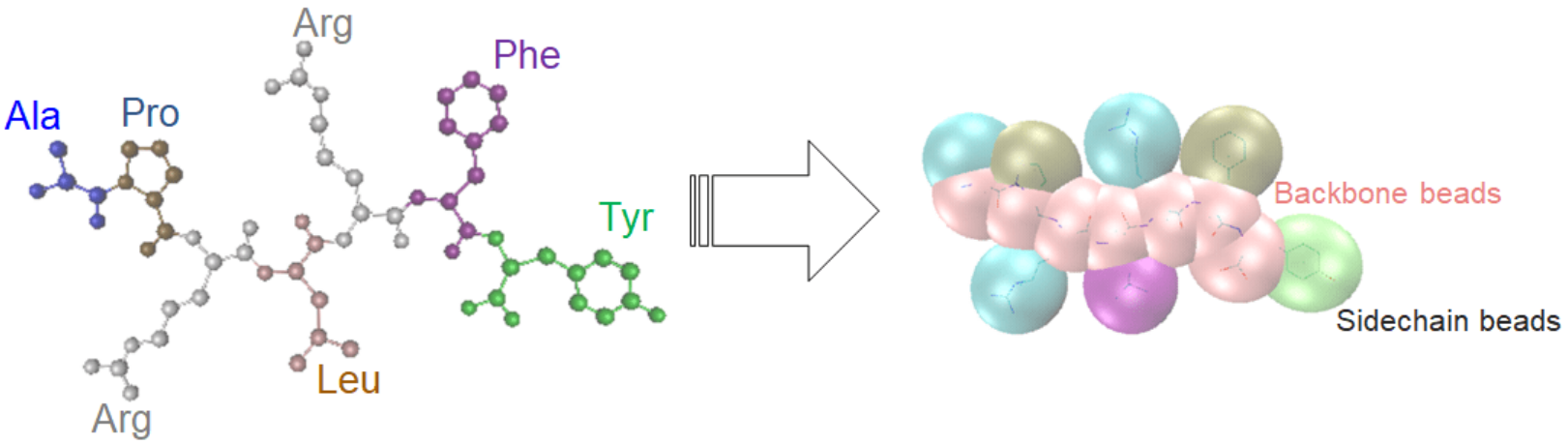
- Calculate reaction energies
  - Photoelectrochemistry, catalysts.



Credit:https://blog.rheosense.com/activation-energy-arrhenius-equation

Quantum Chemistry Simulation

- Predict protein folding
  - Solve the Hamiltonian for amino acids.



Credit: A. Robert *et al. Resource-efficient quantum algorithm for protein folding*
https://www.mathworks.com/help/matlab/math/ground-state-protein-folding-using-variational-quantum-eigensolver-vqe.html#mw_rtc_ProteinFoldingVQEExample_M_D161DD34

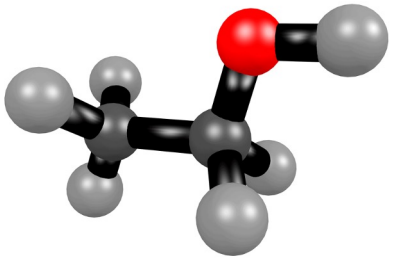
Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

Quantum Chemistry



$\hat{H}$

Molecule

Hamiltonian matrix

Quantum Chemistry

- Solve Schrodinger equation:

$$\hat{H}|\Psi\rangle = E|\Psi\rangle$$

- $\hat{H}$ is the Hamiltonian for electron dynamics (after Born-Oppenheimer Approximation):

$$\hat{H} = \underbrace{-\sum_i \frac{1}{2}\nabla_i^2}_{\text{Kinetic energy}} \underbrace{-\sum_i \sum_A \frac{Z_A}{r_{iA}}}_{\text{Electron-nucleus attraction}} + \underbrace{\sum_{i<j} \frac{1}{r_{ij}}}_{\text{Electron-electron repulsion}}$$

Quantum Chemistry

- Hartree-Fock method
- Second quantization

	State $\|\Psi\rangle$	Hamiltonian $\hat{H}$
Before	Cartesian Space R^3	$\hat{H} = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_i \sum_A \frac{Z_A}{r_{iA}} + \sum_{i<j} \frac{1}{r_{ij}}$
After	Fock Space	$\hat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$

Second quantization

- Fock space
  - Let N be the total number of orbitals.
  - M be the number of electrons.
  - Fock space basis represent occupation of orbitals.
  - Example:
    - $N = 4, \mathrm{M} = 2, |0101\rangle_F$ means orbital 0 and 2 are occupied.
    - Any 2-electron state is a linear combination of $\binom{4}{2}$ basis.
    - $\left|\Psi_{e_1 e_2}\right\rangle = c_0|0011\rangle_F + c_1\,|0101\rangle_F + c_2|0110\rangle_F + c_3|1001\rangle_F + c_4|1010\rangle_F + c_5|1100\rangle_F$

Second quantization

- Hamiltonian

$$\widehat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$$

- Creation operator: $a^\dagger |0\rangle_F = |1\rangle_F,\ a^\dagger |1\rangle_F = 0$
- Annihilation operator: $a|0\rangle_F = 0,\ a|1\rangle_F = |0\rangle_F$
- h_{pq} and V_{pqrs}: precomputed coefficients.

Second quantization

- State $|\Psi\rangle$: a 2^N dimensional vector
- Hamiltonian $\widehat{H}$: a $2^N \times 2^N$ Hermitian matrix
- $\widehat{H}|\Psi\rangle = E|\Psi\rangle$:
  - a PDE problem to a matrix eigenvalue problem

	State $\|\Psi\rangle$	Hamiltonian $\widehat{H}$
Before	Wavefunction on Cartesian Space R^3	$\widehat{H} = -\sum_i \frac{1}{2}\nabla_i^2 - \sum_i \sum_A \frac{Z_A}{r_{iA}} + \sum_{i<j} \frac{1}{r_{ij}}$
After	State vector on Fock Space	$\widehat{H} = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$

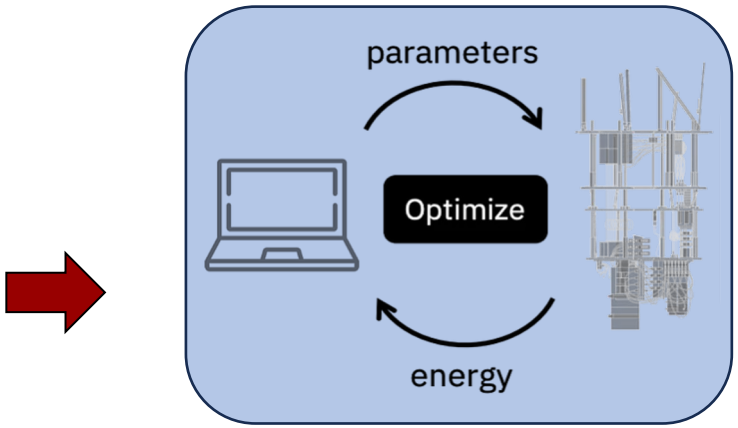
Variational Principal

- $\langle \Psi | \hat{H} | \Psi \rangle \geq E_0$, E_0 is the true ground state energy.
- Solve $\hat{H}|\Psi\rangle = E_0|\Psi\rangle \Leftrightarrow \underset{|\Psi\rangle}{\operatorname{argmin}}\left(\langle \Psi | \hat{H} | \Psi \rangle\right)$

Variational quantum eigensolver

- Use ansatz (parameterized quantum circuit) to probe the ground state $\left|\Psi\left(\vec{\theta}\right)\right\rangle = U\left(\vec{\theta}\right)|\Psi_{init}\rangle$.
- Objective function: $C\left(\vec{\theta}\right) = \left\langle\Psi\left(\vec{\theta}\right)\middle|\hat{H}\middle|\Psi\left(\vec{\theta}\right)\right\rangle$.
- Classical optimizer to minimize $\mathrm{C}\left(\vec{\theta}\right)$.

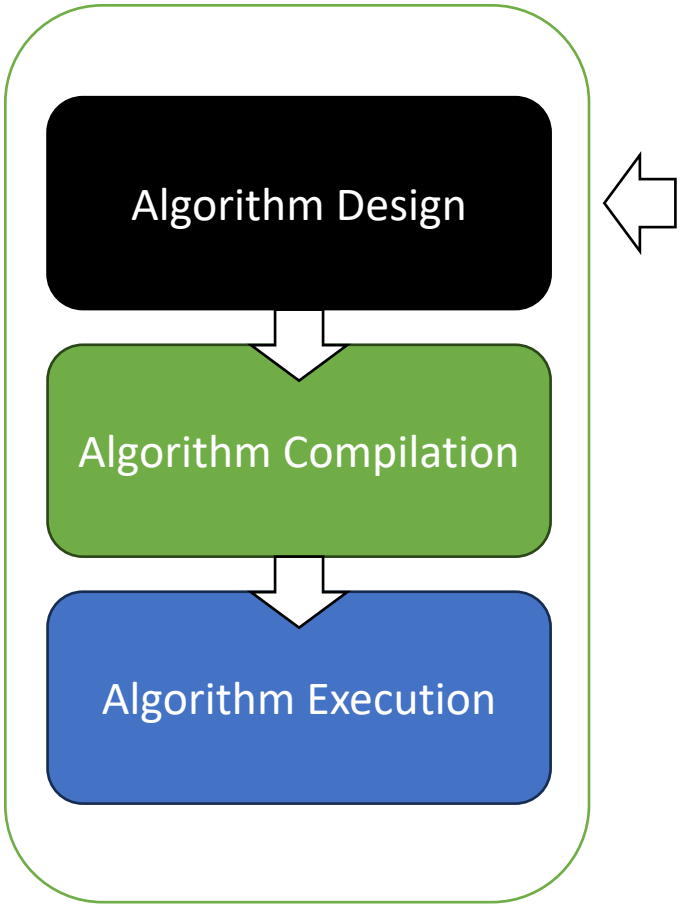
$\hat{H}$



Hamiltonian matrix

Variational quantum eigensolver

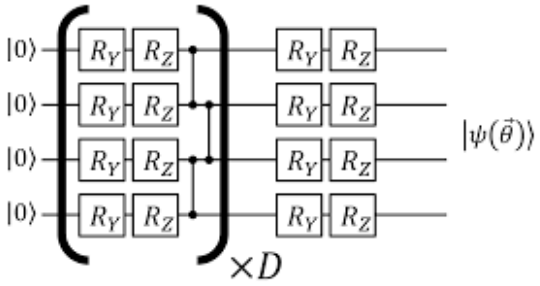
Quantum chemistry simulation stack



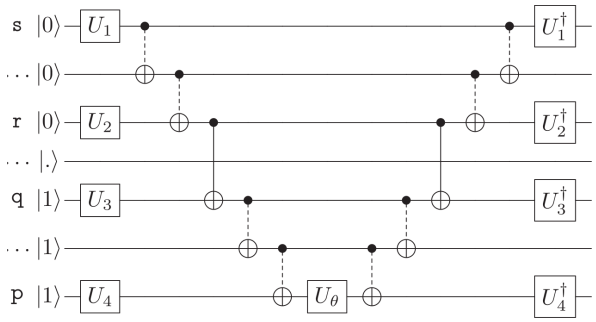
Design $U(\vec{\theta})$, the logical parameterized circuit.

Challenge: How to make the ansatz reach the ground state?

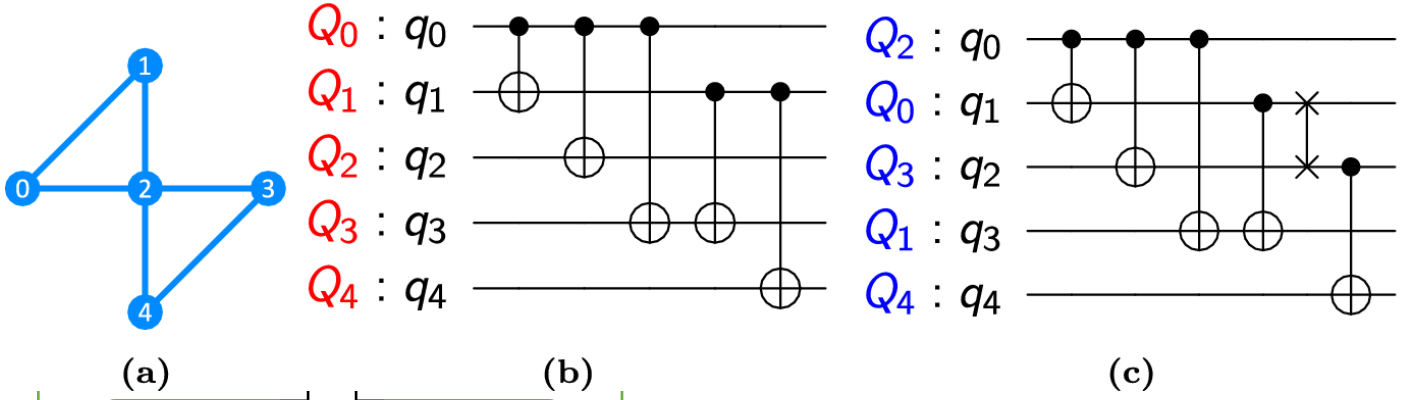
1.HWEA

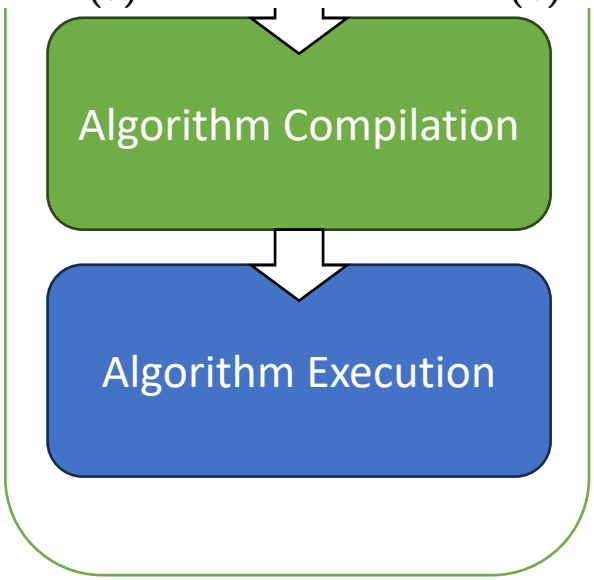


2. UCCSD



Quantum chemistry simulation stack

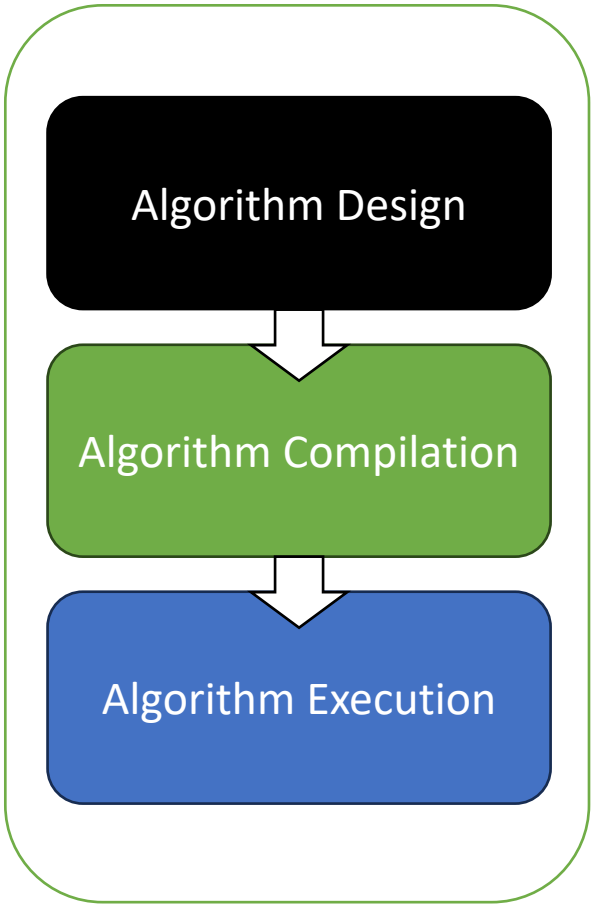




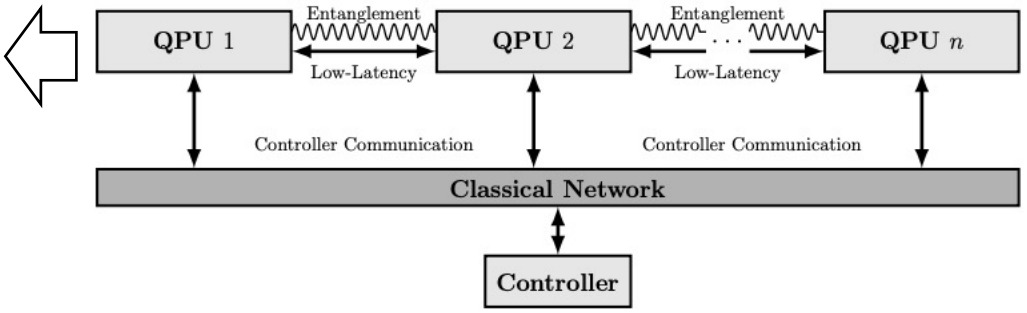
Map the logical circuit to physical circuit.
Challenge: How to efficiently map and swap qubits?

Credit: D. Ferrari *et al. Deterministic algorithms for compiling quantum circuits with recurrent patterns*
https://link.springer.com/article/10.1007/s11128-021-03150-9

Quantum chemistry simulation stack

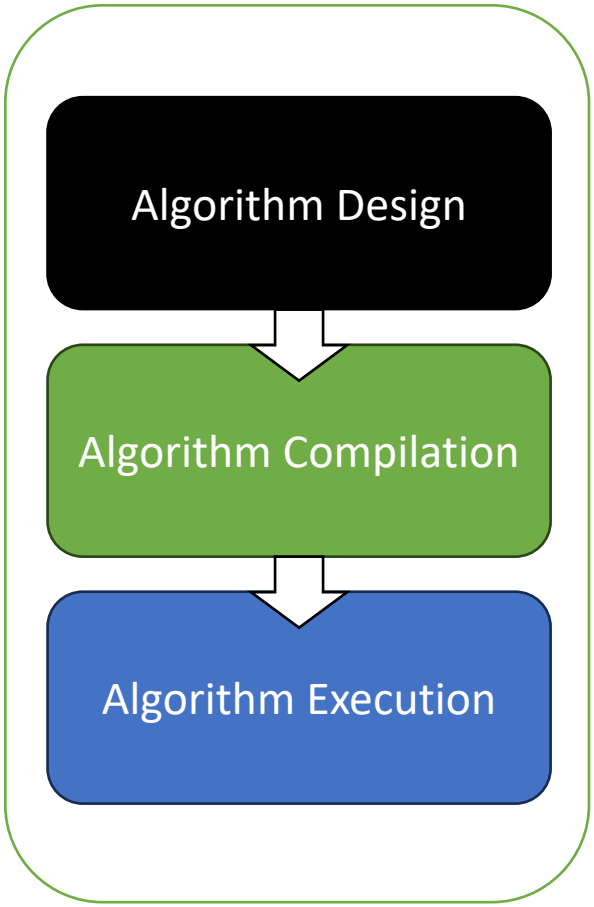


Execute physical circuit.
Challenge: Noise, sampling overhead, multi-chips, classical-quantum hybrid execution…



Credit: https://medium.com/@stephen.diadamo/distributed-quantum-computing-1c5d38a34c50

Quantum chemistry simulation stack



An ongoing project on Hamming weight preserving ansatz.

Y. Jin *et al.*, "Tetris: A Compilation Framework for VQA Applications in Quantum Computing," *ISCA 2024*

Z. Li *et al.*, "A Case for Quantum Circuit Cutting for NISQ Applications: Impact of topology, determinism, and sparsity" preprint@arxiv2412.17929

Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
  - Introduce UCCSD ansatz
  - Tetris compiler for UCCSD ansatz
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

UCCSD Ansatz

$$E_0 = \min_{\vec{\theta}} \left\langle \Psi\left(\vec{\theta}\right) \middle| \hat{H} \middle| \Psi\left(\vec{\theta}\right) \right\rangle$$

- Unitary Coupled Cluster with Singles and Doubles
  - $\left|\Psi\left(\vec{\theta}\right)\right\rangle = e^{T\left(\vec{\theta}\right) - T^{\dagger}\left(\vec{\theta}\right)} \left|\Psi_{HF}\right\rangle$
  - $T\left(\vec{\theta}\right) = \sum_{p,q} \theta_{pq} a_p^{\dagger} a_q + \sum_{p<q,r<s} \theta_{pqrs} a_p^{\dagger} a_q^{\dagger} a_s a_r$

- First-Order Trotter Approximation

$$e^{A+B} = \lim_{n \to \infty} \left(e^{A/n} e^{B/n} \right)^n$$

Jordan-Wigner Encoder

- $a_p^\dagger, a_p$: creation/annihilation operator ⇒ qubit operator
- $|\Psi\rangle$: Fock space ⇒ qubit space
- $\widehat{H}$: operator on Fock space ⇒ operator on qubit space

- Anticommutation requirements: $\{A, B\} = AB + BA$
  - $\left\{a_p^\dagger, a_q^\dagger\right\} = 0, \left\{a_p, a_q\right\} = 0, \forall p, q$
  - $\left\{a_p^\dagger, a_q\right\} = \begin{cases} 0 \; if \; p \neq q \\ 1 \; if \; p = q \end{cases}$

Jordan-Wigner Encoder

- Pauli matrices:

$$X = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}, Y = \begin{bmatrix} & -i \\ i & \end{bmatrix}, Z = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}$$

- $a_p \coloneqq Z_0 \dots Z_{p-1} \frac{X_p + iY_p}{2}$
- $a_p^\dagger \coloneqq Z_0 \dots Z_{p-1} \frac{X_p - iY_p}{2}$

- Example: Pauli string $YYXY \coloneqq Y \otimes Y \otimes X \otimes Y$

UCCSD

- $e^{\theta_{pq}\left(a_p^\dagger a_q - a_q^\dagger a_p\right)} = \exp\left(\frac{i}{2}\theta_{pq} \bigotimes_{j=p+1}^{q-1} Z_j * \left(Y_p X_q - Y_p X_q\right)\right)$

  $= \exp\left(\frac{i}{2}\theta_{pq}\left(Y_p Z_{p+1} \dots Z_{q-1} X_q - Y_p Z_{p+1} \dots Z_{q-1} X_q\right)\right)$

- $e^{\theta_{pqrs}\left(a_p^\dagger a_q^\dagger a_s a_r - a_r^\dagger a_s^\dagger a_q a_p\right)} =$

  $\exp\left(\frac{i}{8}\theta_{pqrs} \bigotimes_{j\in(p,q)\cup(r,s)} Z_j * \left(X_p X_q Y_r X_s + Y_p X_q Y_r Y_s + X_p Y_q Y_r Y_s + X_p X_q X_r Y_s - X_p Y_q X_r X_s - Y_p Y_q X_r Y_s - Y_p Y_q Y_r X_s - Y_p X_q X_r X_s\right)\right)$, let $p < q < \mathrm{r} < \mathrm{s}$.

UCCSD

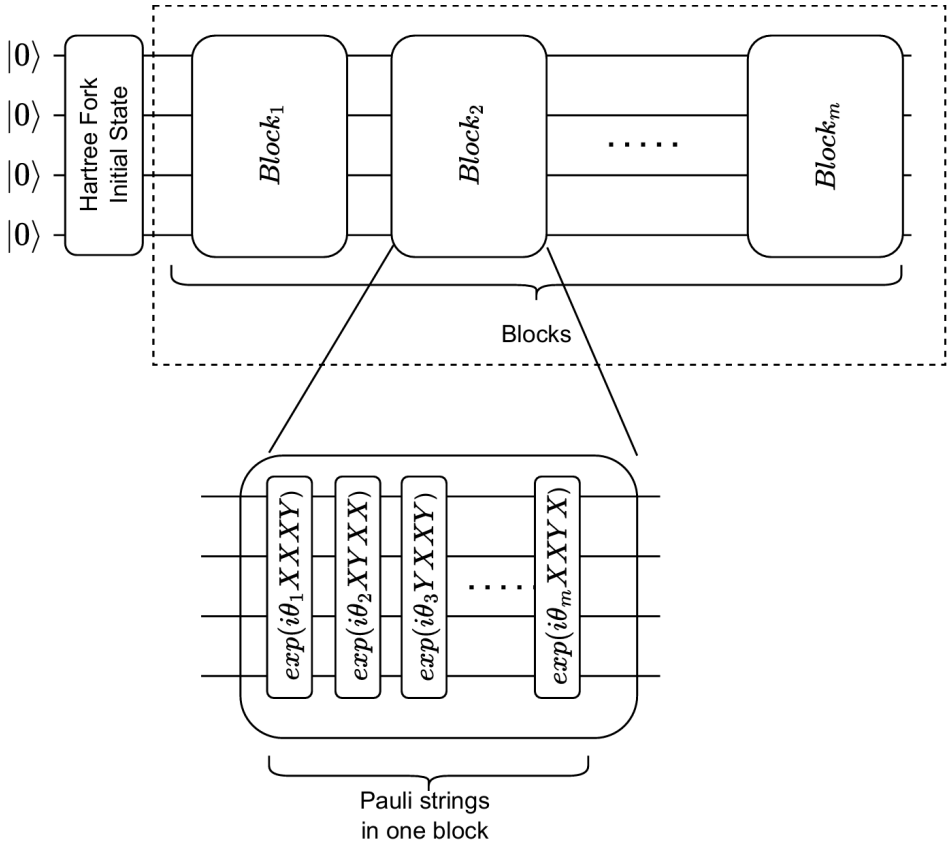
- Pauli evolution gates sequence for some p,q,r,s

$$e^{\frac{i}{8}\theta_{pqrs}*IXZZZZXIIYZZXI}$$
$$e^{\frac{i}{8}\theta_{pqrs}*IYZZZZXIIYZZYI}$$
$$e^{\frac{i}{8}\theta_{pqrs}*IXZZZZYIIYZZYI}$$
$$e^{\frac{i}{8}\theta_{pqrs}*IXZZZZXIIXZZYI}$$
$$e^{-\frac{i}{8}\theta_{pqrs}*IXZZZZYIIXZZXI}$$
$$e^{-\frac{i}{8}\theta_{pqrs}*IYZZZZYIIXZZYI}$$
$$e^{-\frac{i}{8}\theta_{pqrs}*IYZZZZYIIYZZXI}$$
$$e^{-\frac{i}{8}\theta_{pqrs}*IYZZZZXIIXZZXI}$$

  - feasible on quantum computer

UCCSD

- The ansatz becomes a sequence of blocks of Pauli evolution gates.



UCCSD

$$\left|\Psi\left(\vec{\theta}\right)\right\rangle = \prod_{j} e^{i\theta_j \mathrm{PauliString}_j} \left|\Psi_{HF}\right\rangle$$

$$\hat{H} = \sum_{k} w_k \,\mathrm{PauliString}k$$

$$C\left(\vec{\theta}\right) = \sum_{k} w_k \left\langle\Psi\left(\vec{\theta}\right)\middle|\mathrm{PauliString}k\middle|\Psi\left(\vec{\theta}\right)\right\rangle$$

Takeaway:

- Jordan-Wigner transformation and two steps of Trotter approximation:
  - the ansatz becomes a list of blocks of Pauli evolution gates.
  - the Hamiltonian becomes a list of weighted Pauli strings.
- Both lists scale as $O(N^4)$, N is the number of orbitals.

Outline

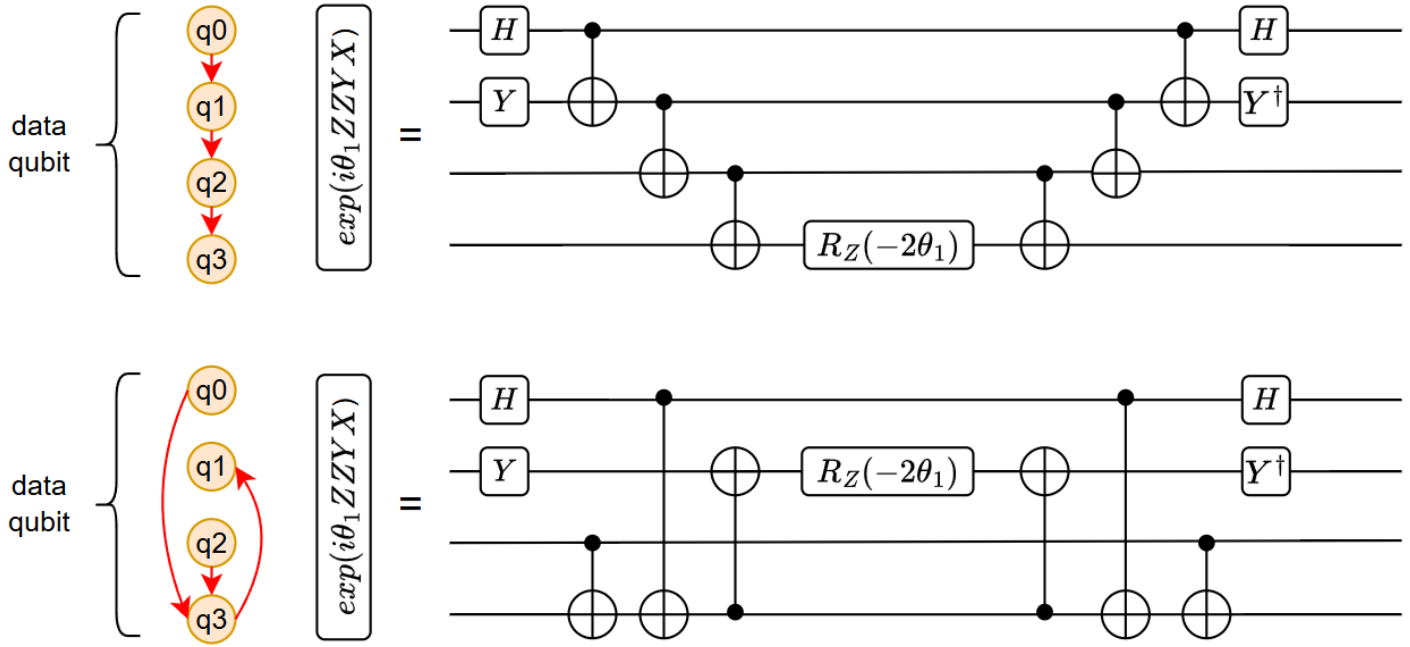
- Background of quantum chemistry
- Quantum chemistry algorithm compilation
  - Introduce UCCSD ansatz
  - Tetris compiler for UCCSD ansatz
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

Tetris: A Compilation Framework for VQA Applications in Quantum Computing

Yuwei Jin*, Zirui Li*, Fei Hua, Tianyi Hao,
Huiyang Zhou, Yipeng Huang, Eddy Z. Zhang

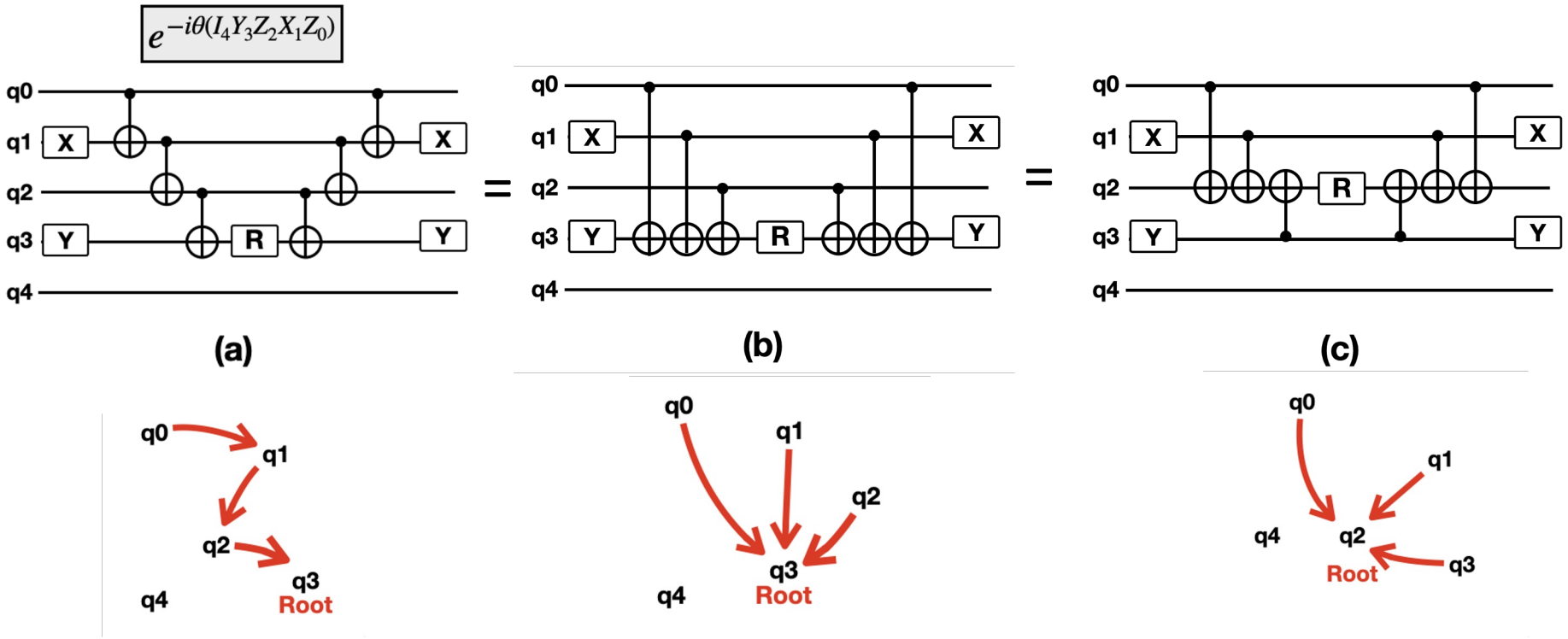
How to compile a single Pauli string?

- Example: $\exp(i\theta_1 ZZYX)$.



Note: Y gate here is $R_X(\frac{\pi}{2})$. It has different meaning from Pauli letter Y in the pauli string.

Root and leaves

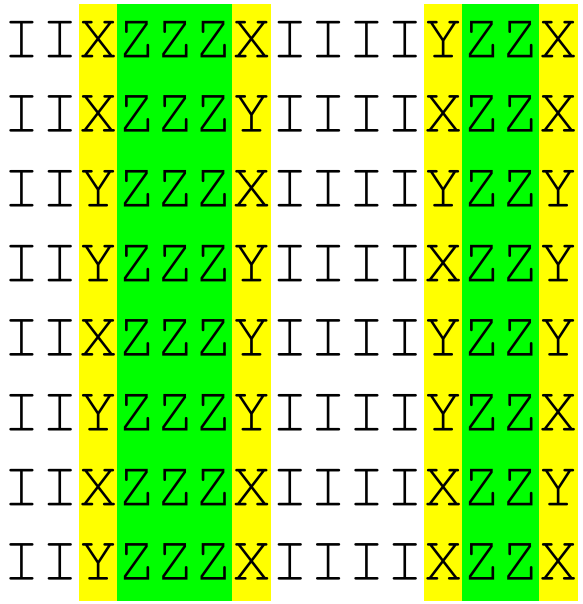


Cancelation opportunity

- Inside each block, the Pauli strings usually share the same I and non-I locations.

IIXZZZXIIIIYZZX
IIXZZZYIIIIXZZX
IIYZZZXIIIIYZZY
IIYZZZYIIIIXZZY
IIXZZZYIIIIYZZY
IIYZZZYIIIIYZZX
IIXZZZXIIIIXZZY
IIYZZZXIIIIXZZX

Cancelation opportunity



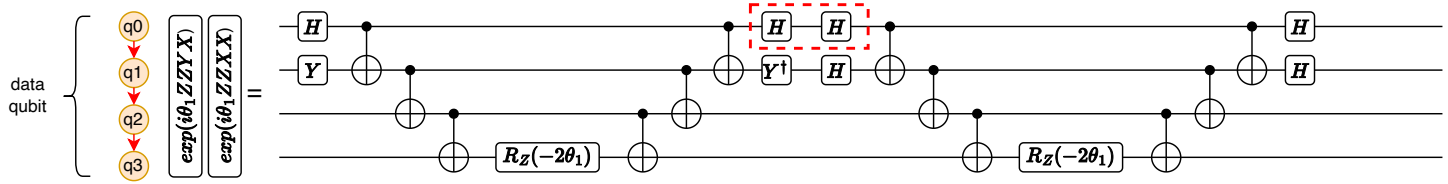
- The green part is the qubits that share the same letters for every Pauli string.
- The green part is cancelable.

Cancelation opportunity

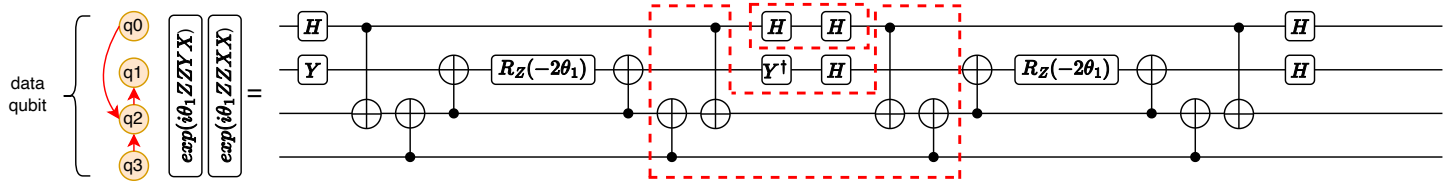
ZZYX

ZZXX

- If we put green part away from the leaf, no cancellation opportunity.

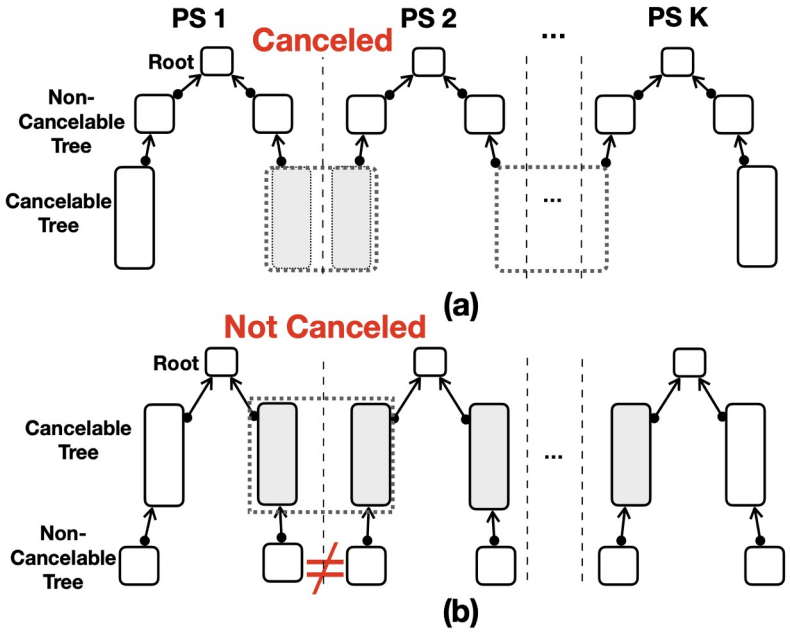


- If we put green part close to the leaf, cancellation will happen.



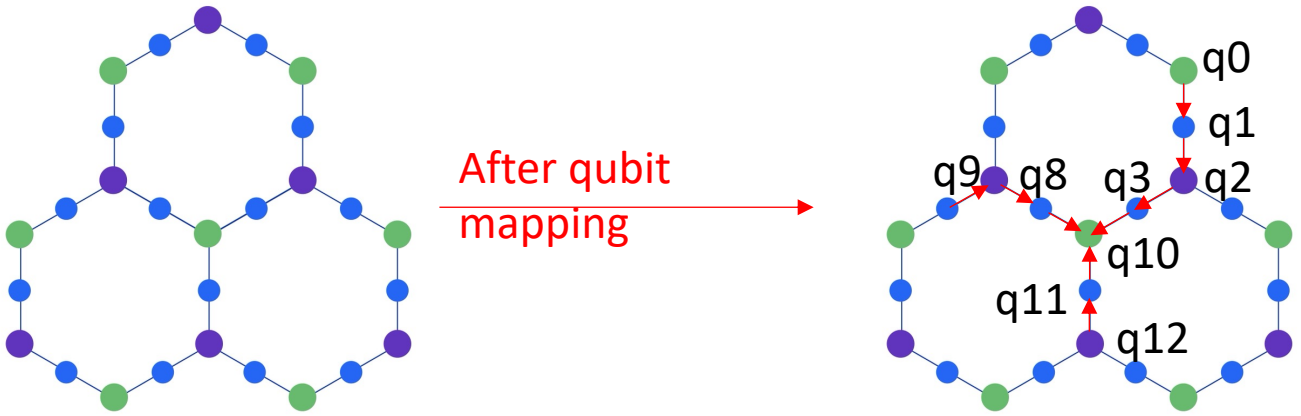
Cancelation opportunity

- Takeaway:
  - Qubits that have identical letters are cancelable
  - Put the cancelable qubits at the leaves part of the tree can cancel more gates.

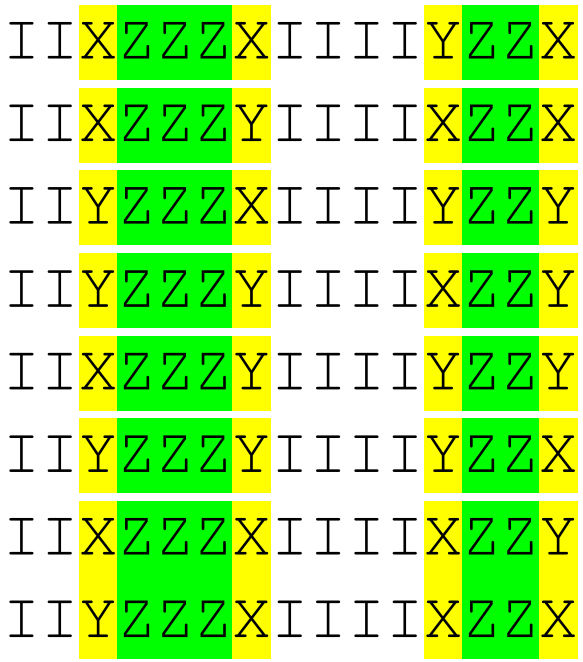


Connectivity Constrains

- Superconducting quantum computers have connectivity constrains.
- Example: Map qubits in Pauli `IIXZZZXIIIIYZZX` evolution gate to the heavyhex topology hardware.
  - Logical qubit 0, 1, 2, 3, 8, 9, 10, 11, 12 are non-I.



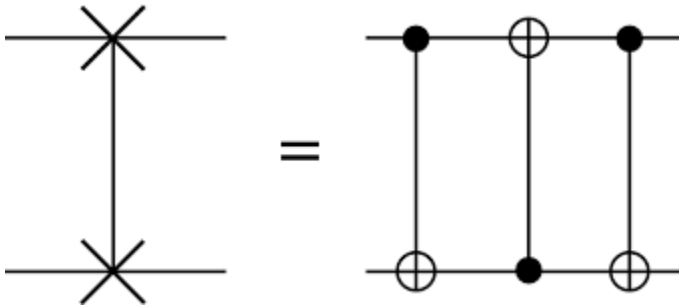
What's a good qubit mapping?



- Our proposal: Map as many green part to the leaves of the tree.

Qubit Routing

- We know how to find a good qubit mapping for each block.
- Q: How to switch qubit mappings between blocks?
  - A: Add swap gates. (Each swap gate costs 3 CNOT gates.)



An important metric

- 2-qubit gate count
  - (2-qubit gates are hard and noisy).

    Total 2-qubit gate count

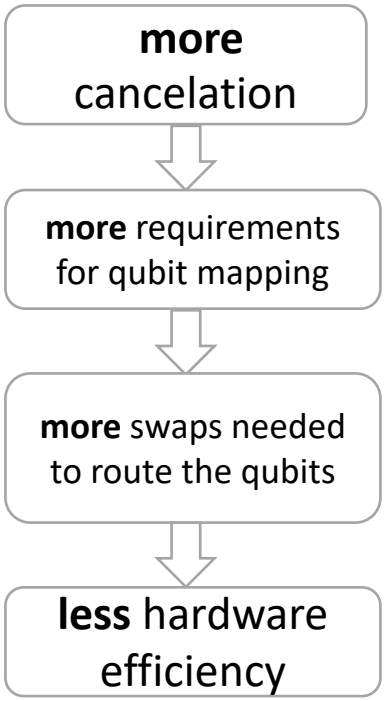
    =2-qubit gate count in the logical circuit

    - 2-qubit gates canceled

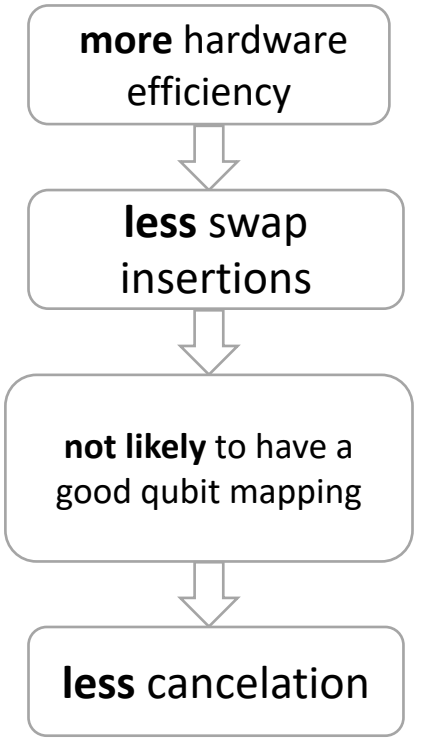
    + 2-qubit gates added from swaps
- Goal: maximize cancelation, minimize swaps

Tradeoff between cancelation and swap insertions

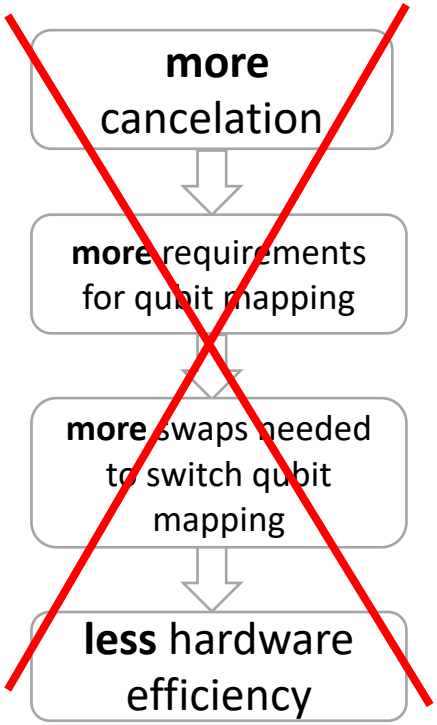
PCOAST
(QCE'23 Paykin *et al.*)



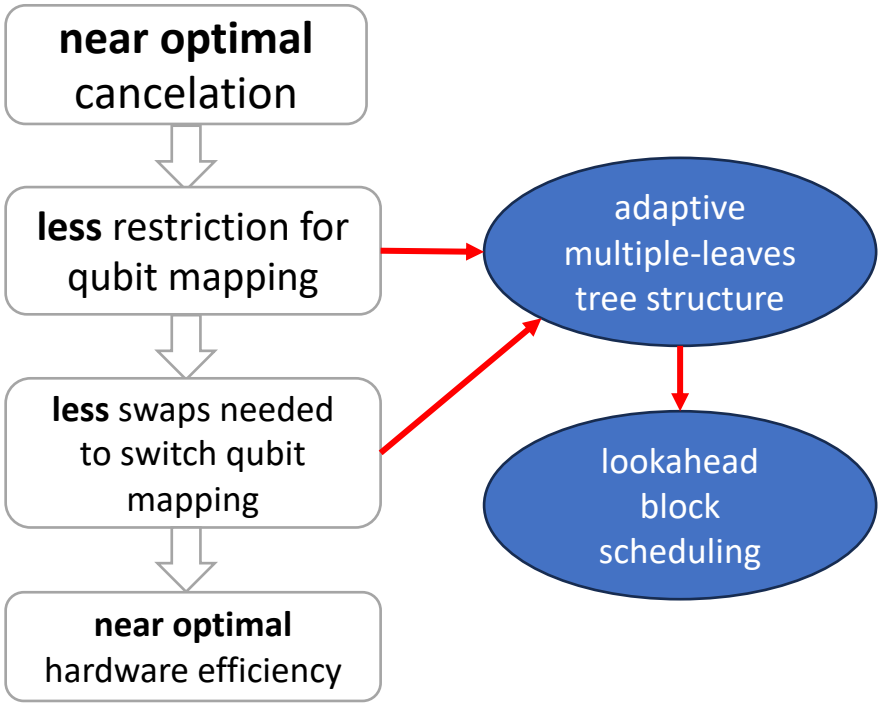
Paulihedral
(ASPLOS' 22 Li *et al.*)



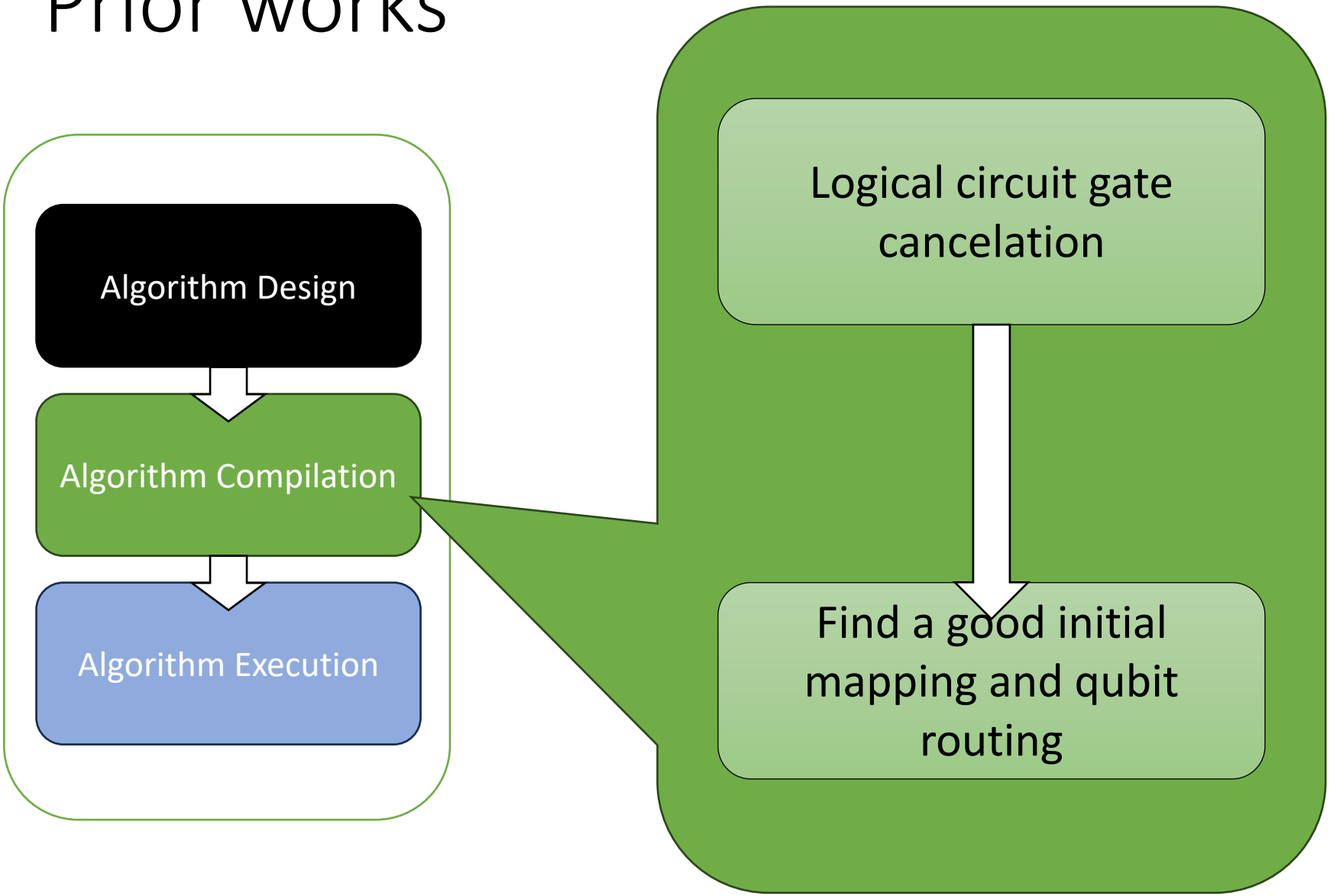
Tradeoff between cancelation and swap insertions



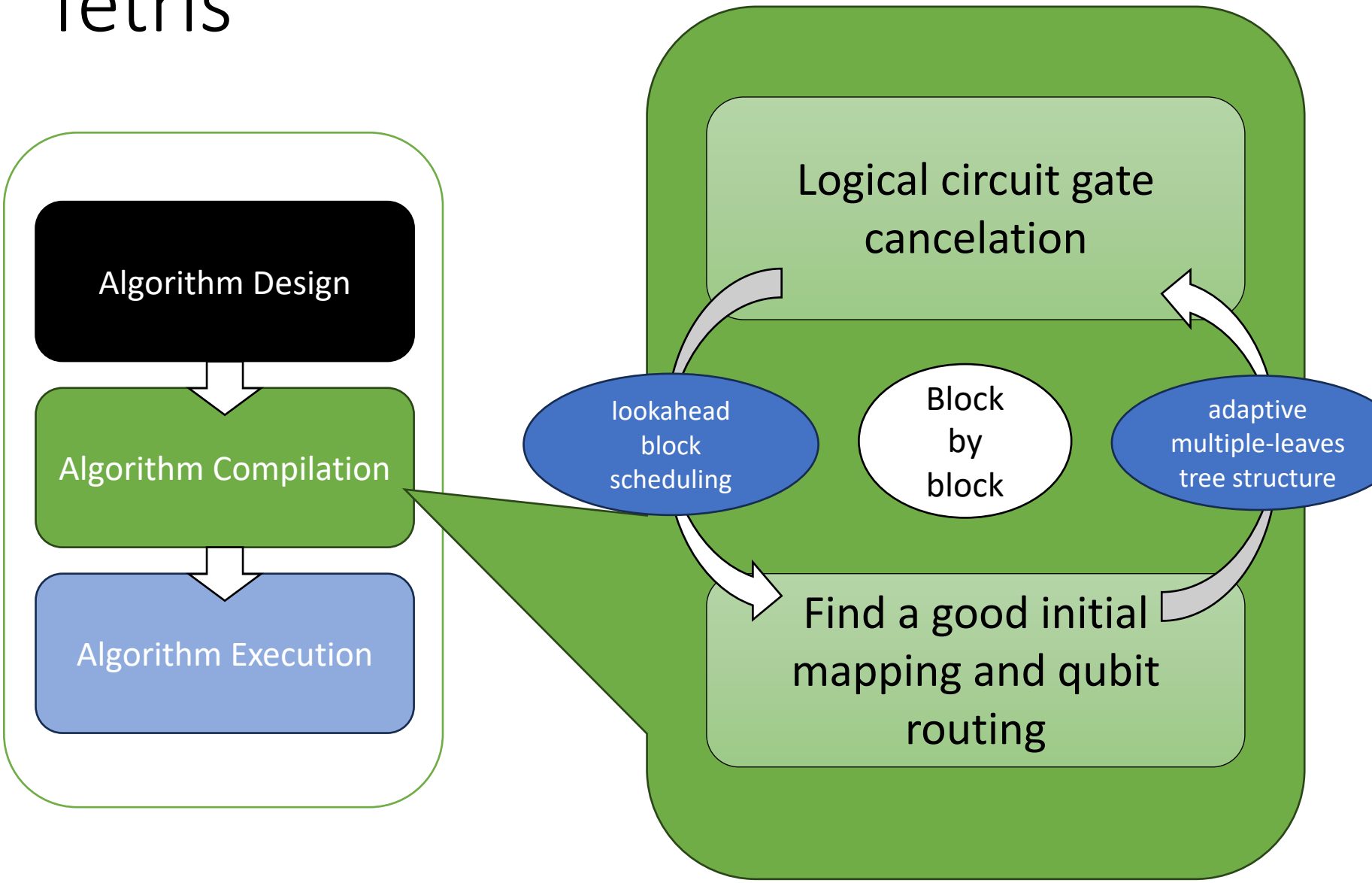
Tetris (@ISCA'24)



Prior works

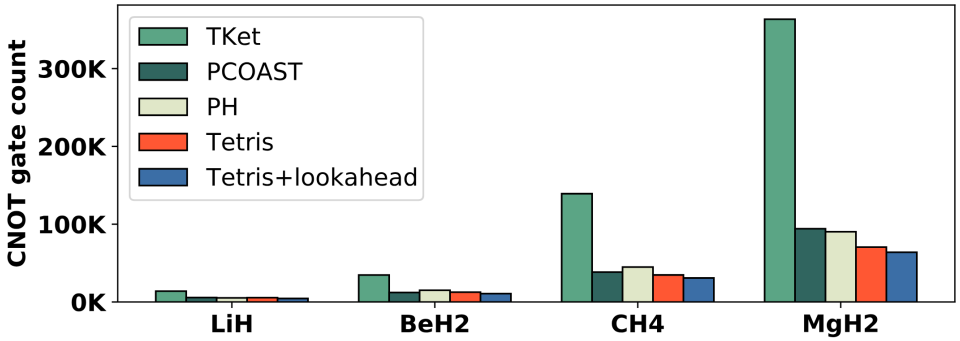

Algorithm Design
Algorithm Compilation
Algorithm Execution
Logical circuit gate cancelation
Find a good initial mapping and qubit routing

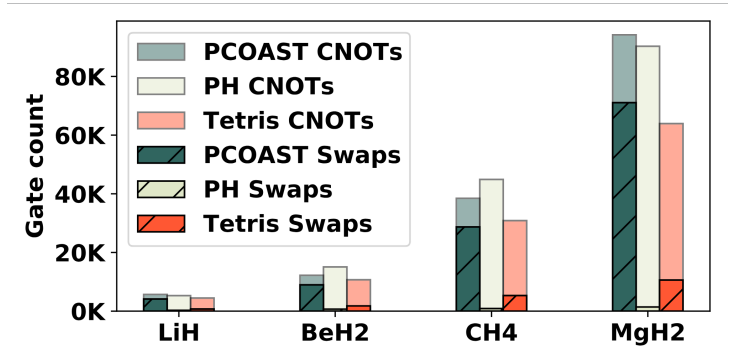
Tetris

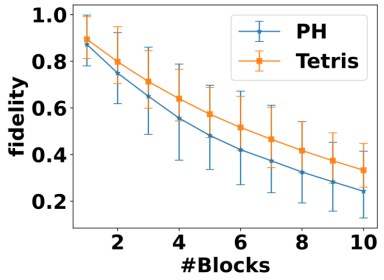

Algorithm Design
Algorithm Compilation
Algorithm Execution
Logical circuit gate cancelation
lookahead block scheduling
Block by block
adaptive multiple-leaves tree structure
Find a good initial mapping and qubit routing

Evaluation

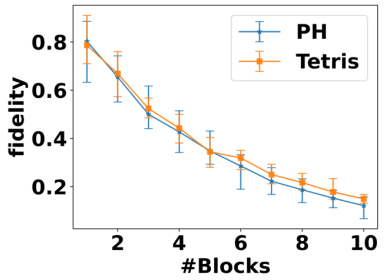
- Reduce the CNOT gate count by ~ 20%.







Molecule: LiH (12 qubits)



Molecule: CO2 (30 qubits)

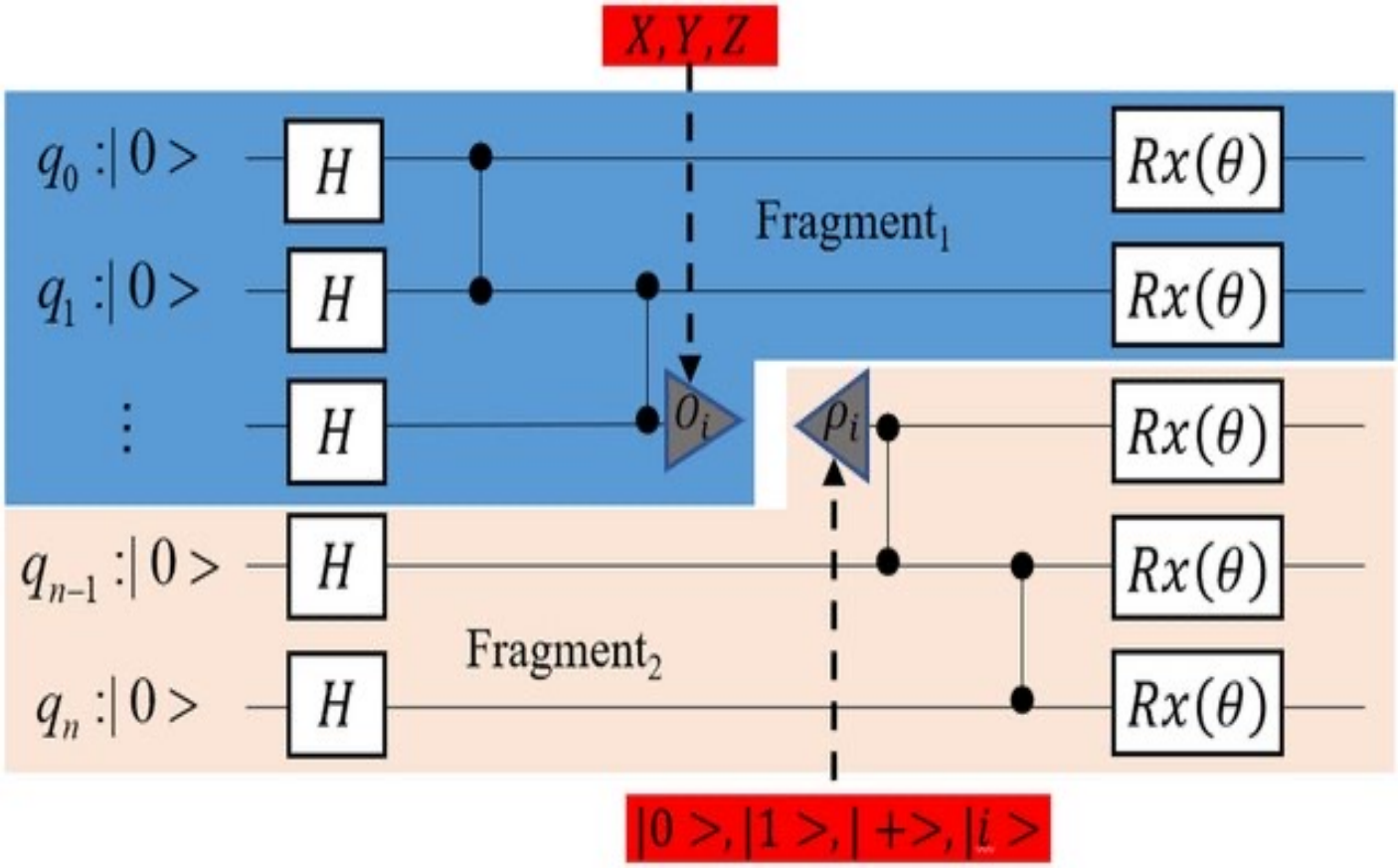
Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

A Case for Quantum Circuit Cutting for NISQ Applications: Impact of topology, determinism, and sparsity

Zirui Li, Minghao Guo, Mayank Barad, Wei Tang, Eddy Z. Zhang, Yipeng Huang

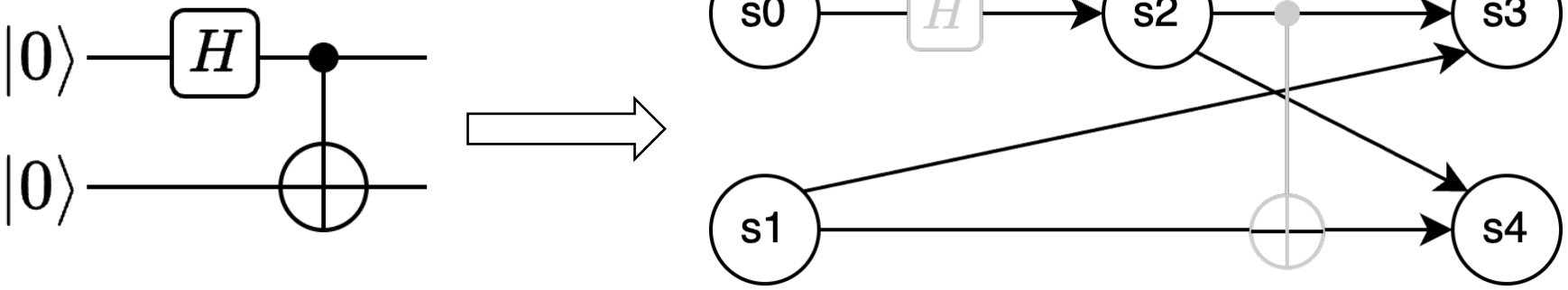
Quantum circuit cutting



Credit: Lian, Hang & Xu, Jinchen & Zhu, Yu & Fan, Zhiqiang & Liu, Yi & Shan, Zheng. (2023). Fast reconstruction algorithm based on HMC sampling. Scientific Reports. 13. 10.1038/s41598-023-45133-z.

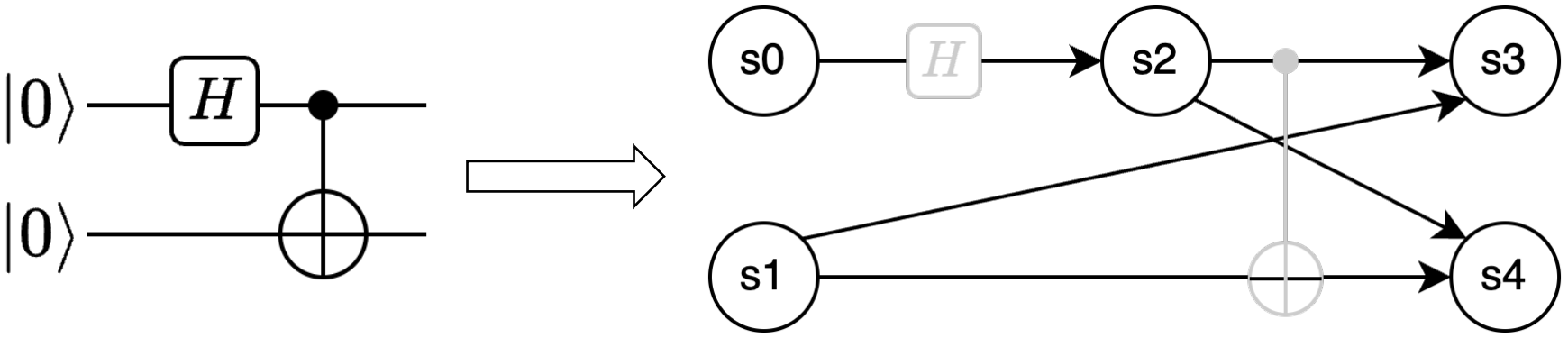
Quantum circuit to Bayesian network

- Record the density matrix in Pauli string format.
- Identity matrix and Pauli matrices:
  $I = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, X = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}, Y = \begin{bmatrix} & -i \\ i & \end{bmatrix}, Z = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}$
- I, X, Y, Z form an orthogonal basis for all Hermitian matrices in $\mathbb{C}^{2\times 2}$.
- 4^n Pauli strings form the basis for n-qubit density matrix state in $\mathbb{C}^{2^n \times 2^n}$



Quantum circuit to Bayesian network

- The Bell state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = CNOT(I \otimes H)|00\rangle$ in density matrix representation calculate by hand :
  - The initial state $\rho_0 = |00\rangle\langle 00| = \frac{1}{4}II + \frac{1}{4}IZ + \frac{1}{4}ZI + \frac{1}{4}ZZ.$
  - After Hadamard $\rho_1 = H\rho_0 H^\dagger = \frac{1}{4}II + \frac{1}{4}IX + \frac{1}{4}ZI + \frac{1}{4}ZX.$
  - After CNOT $\rho_2 = CNOT\rho_1 CNOT^\dagger = \frac{1}{4}II + \frac{1}{4}XX - \frac{1}{4}YY + \frac{1}{4}ZZ.$

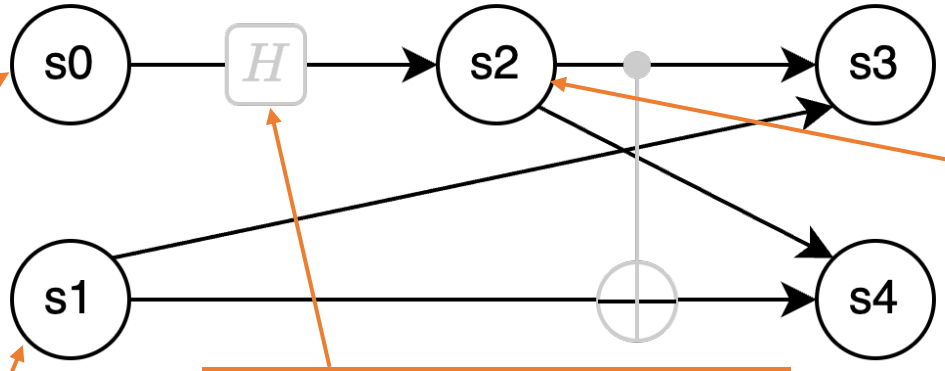


s0	w
I	0.5
X	0
Y	0
Z	0.5

Tensor 1

s1	w
I	0.5
X	0
Y	0
Z	0.5

Tensor 2



s0	s2	w
I	I	1
X	Z	1
Z	X	1
Y	Y	-1
otherwise		0

Tensor 3

s2	w
I	0.5
X	0.5
Y	0
Z	0

Tensor 4
Contract tensor 1 and tensor 3, get tensor 4.

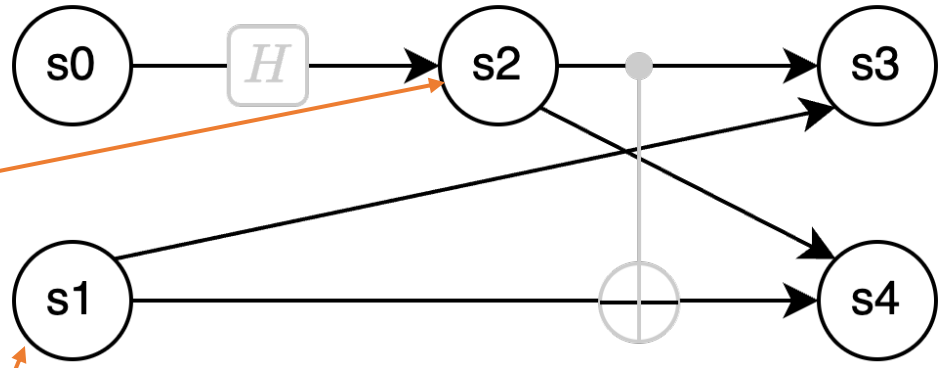
This tensor's size is 16 but only 4 entries have non-zero weights.

s2	w
I	0.5
X	0.5
Y	0
Z	0

Tensor 4

s1	w
I	0.5
X	0
Y	0
Z	0.5

Tensor 2

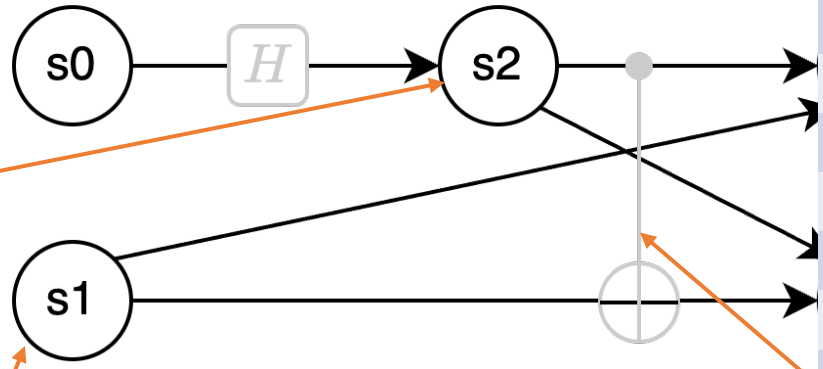


s2	w
I	0.5
X	0.5
Y	0
Z	0

Tensor 4

s1	w
I	0.5
X	0
Y	0
Z	0.5

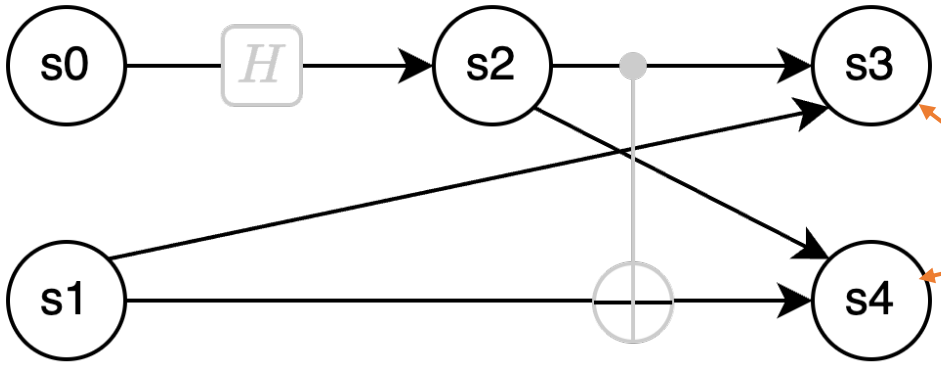
Tensor 2



Next page: contract tensor 4, 2 and 5, get tensor 6.

s2	s1	s3	s4	w
I	I	I	I	1
I	X	I	X	1
I	Y	Z	Y	1
I	Z	Z	Z	1
X	I	X	X	1
X	X	X	I	1
X	Y	Y	Z	1
X	Z	Y	Y	-1
Y	I	Y	X	1
Y	X	Y	I	1
Y	Y	X	Z	-1
Y	Z	X	Y	1
Z	I	Z	I	1
Z	X	Z	X	1
Z	Y	I	Y	1
Z	Z	I	Z	1
otherwise				0

Tensor 5
This tensor's size is 256 but only 16 entries have non-zero weights.



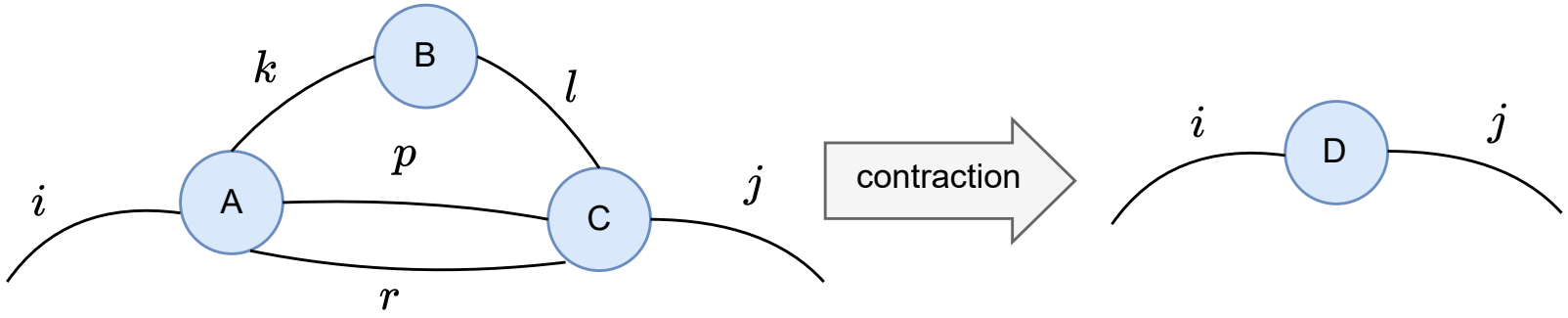
s3	s4	w
I	I	0.25
X	X	0.25
Y	Y	-0.25
Z	Z	0.25
otherwise		0

Tensor 6

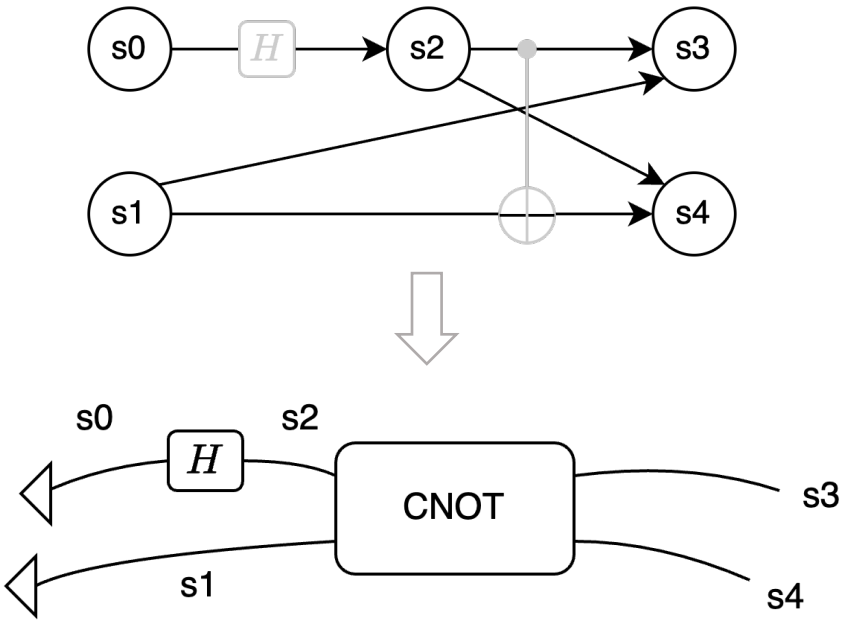
Tensor 6 matches the hand-calculated $\frac{1}{4}II + \frac{1}{4}XX - \frac{1}{4}YY + \frac{1}{4}ZZ.$

Tensor contraction

- $D_{i,j} = \sum_{k,l,p,r} A_{i,k,p,r} B_{k,l} C_{l,p,r,j}$



Bayesian network to tensor network



After contraction:

s3	s4	w
I	I	0.25
X	X	0.25
Y	Y	-0.25
Z	Z	0.25
otherwise		0

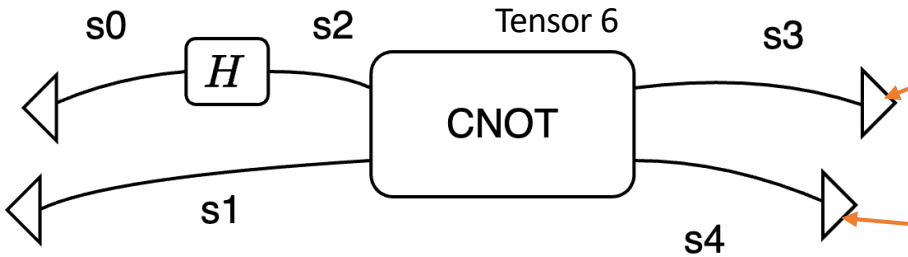
Tensor 6

Expectation value on an observable

- If we want to know the expectation value on observable YY.
- $\hat{O} = YY$, the bell state is $\rho = \frac{1}{4}II + \frac{1}{4}XX - \frac{1}{4}YY + \frac{1}{4}ZZ$.
- The expectation value is $trace(\rho\hat{O}) = -\frac{1}{4}trace(II) = -1$.

s3	s4	w
I	I	0.25
X	X	0.25
Y	Y	-0.25
Z	Z	0.25
otherwise		0

Tensor 6



s3	w
I	0
X	0
Y	1
Z	0

Tensor 7

s4	w
I	0
X	0
Y	1
Z	0

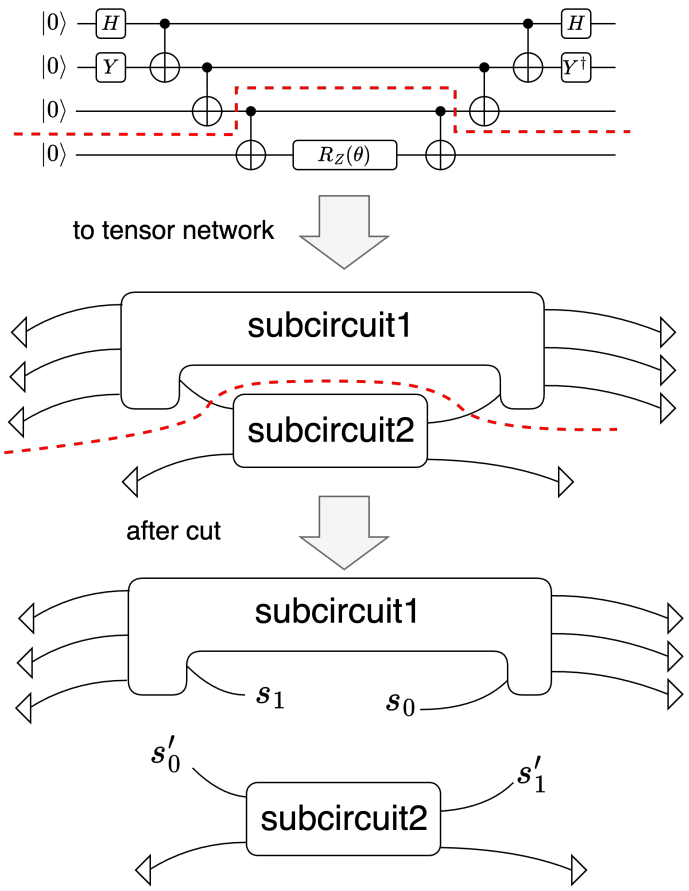
Tensor 8

Cut the tensor network

- Subcircuit 1 has 2 open edges;
- Subcircuit 2 has 2 open edges;
- Run 3*4 different settings of each subcircuit to fill in the two tensors.

s0	s1	w
I	I	?
I	X	?
I	Y	?
I	Z	?
…	…	…

s'0	s'1	w
I	I	?
I	X	?
I	Y	?
I	Z	?
…	…	…



Determinism

- Clifford gates' tensor is deterministic.
  - Clifford gates stabilize Pauli strings.
    - Clifford gate will only do a permutation of all Pauli strings.
  - Only 4^n out of $4^n * 4^n$ tensor entries are non-zero.
- For a non-Clifford gate, like T gate, the tensor is not deterministic.

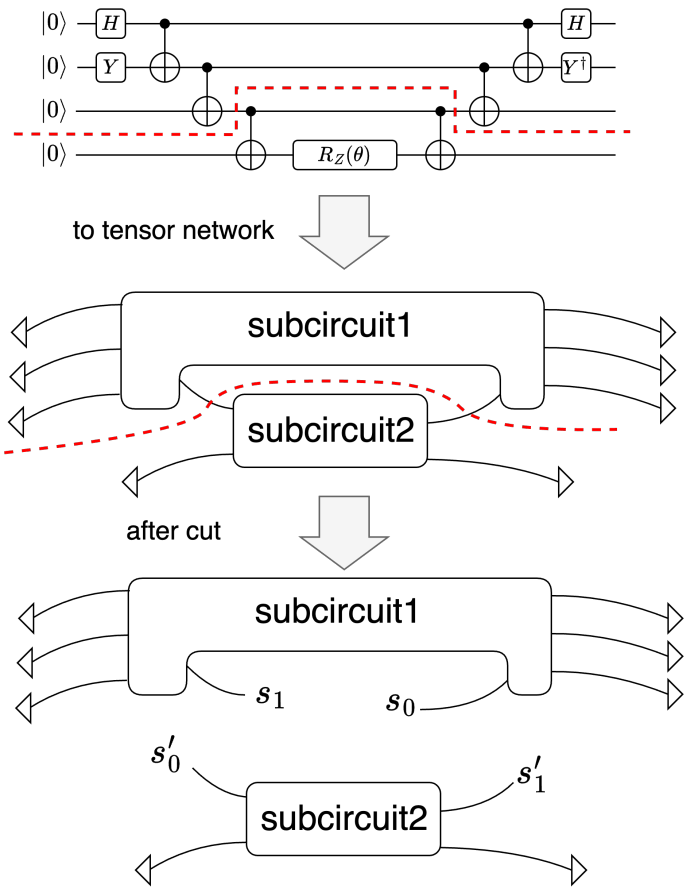
	I	X	Y	Z
I	1	0	0	0
X	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0
Y	0	$\frac{-1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0
Z	0	0	0	1

Knowledge compilation

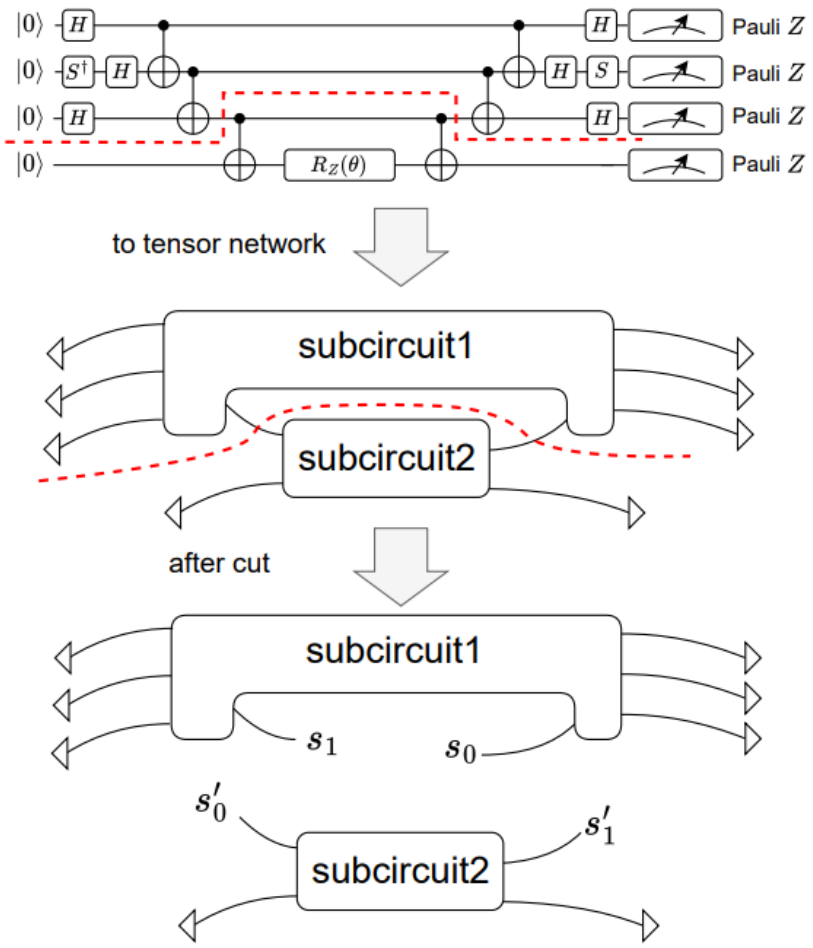
- We know which entries have zero value after knowledge compilation.
- Subcircuit 1 has 4 non-zero values.
- Subcircuit 2 has 6 non-zero values.

s0	s1	w
I	I	?
I	X	?
I	Y	?
I	Z	?
…	…	…

s'0	s'1	w
I	I	?
I	X	?
I	Y	?
I	Z	?
…	…	…



Knowledge compilation



subcircuit1: $\text{factor}_{s_1,s_0} = ZZ$

subcircuit2: when $\theta = 0$,

$\text{factor}_{s'_0 s'_1} = II + XX + YY + ZZ$, the expval is 1

when $\theta = 0.1$,

$\text{factor}_{s'_0 s'_1} = II + XX + 0.995YY + 0.995ZZ + 0.0998YZ - 0.0998ZY$,

the expval is 0.995

when $\theta = 0.3$,

$\text{factor}_{s'_0 s'_1} = II + XX + 0.955YY + 0.955ZZ + 0.296YZ - 0.296ZY$,

the expval is 0.955

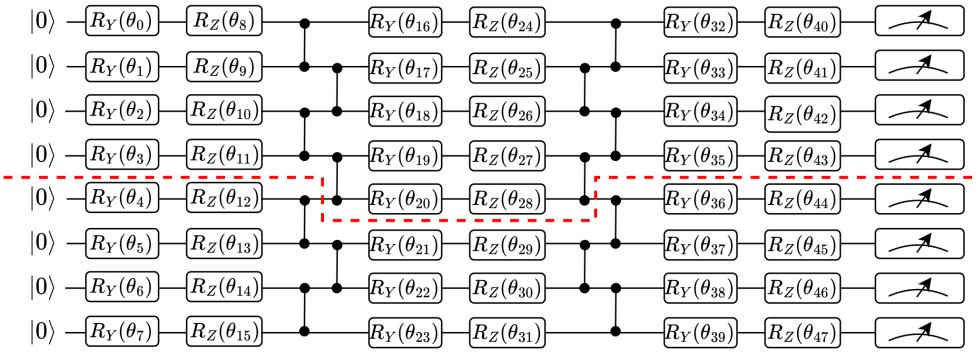
when $\theta = 0.5$,

$\text{factor}_{s'_0 s'_1} = II + XX + 0.878YY + 0.878ZZ + 0.479YZ - 0.479ZY$,

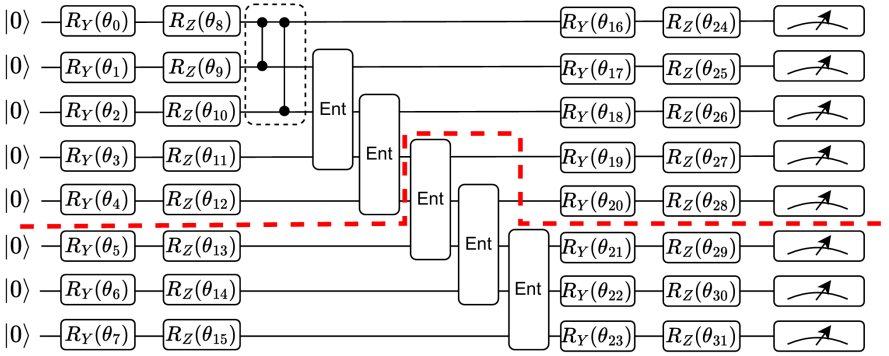
the expval is 0.878

Tensor sparsity

Workloads	#Qubits	#Layers	Subcircuit #qubits constraints	Tensor 1 sparsity	Tensor 2 sparsity	Subcircuit 1 req. experiments	Subcircuit 2 req. experiments
Pairwise HWEA	8	1	5	0.0%	0.0%	4/4	3/3
Pairwise HWEA	8	2	5	0.0%	0.0%	12/12	12/12
Pairwise HWEA	8	4	5	0.0%	0.0%	144/144	144/144
3-local HWEA	8	1	5	75.0%	75.0%	81/144	100/144
QNN	4	2	3	42.1%	84.6%	93/108	72/192

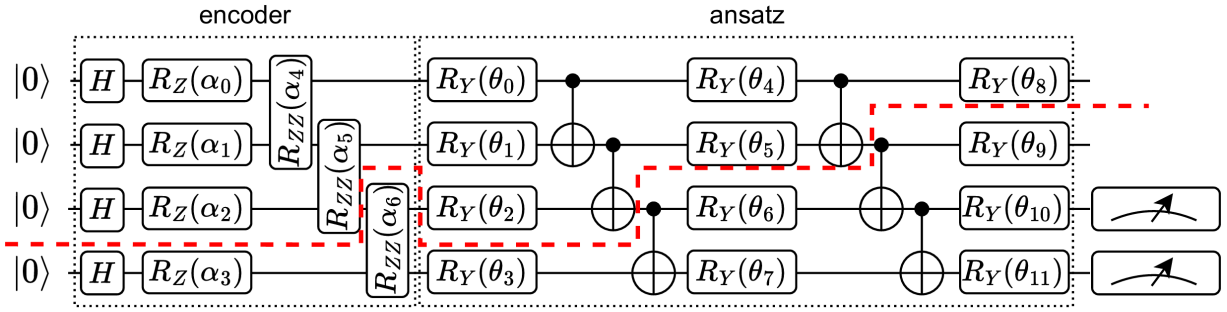


Pairwise HWEA



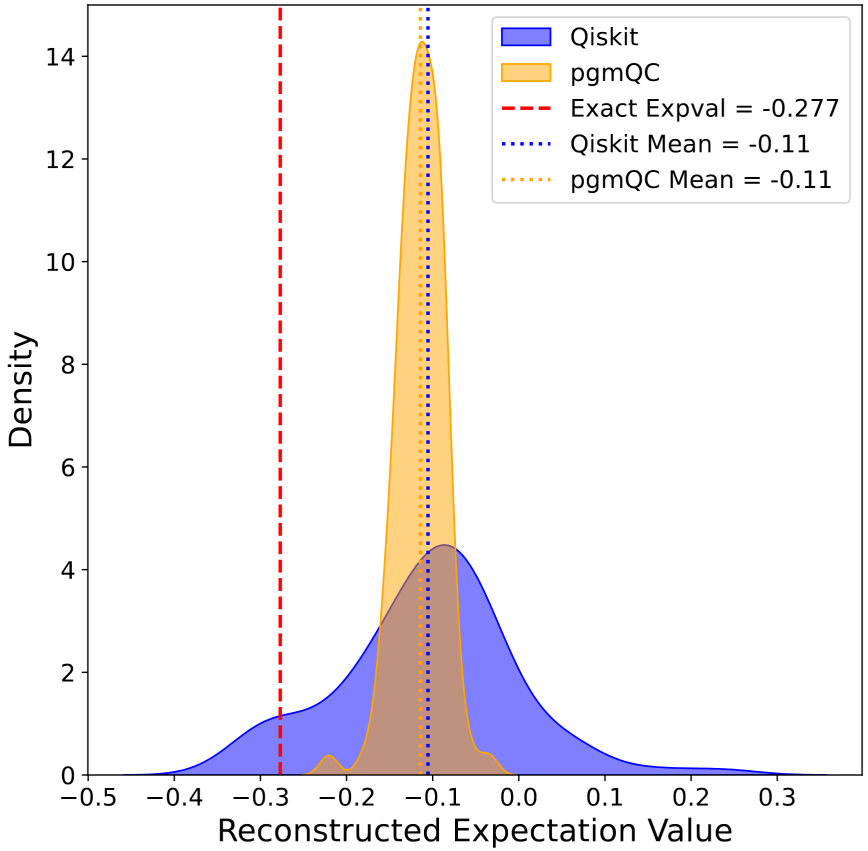
3-local HWEA

QNN



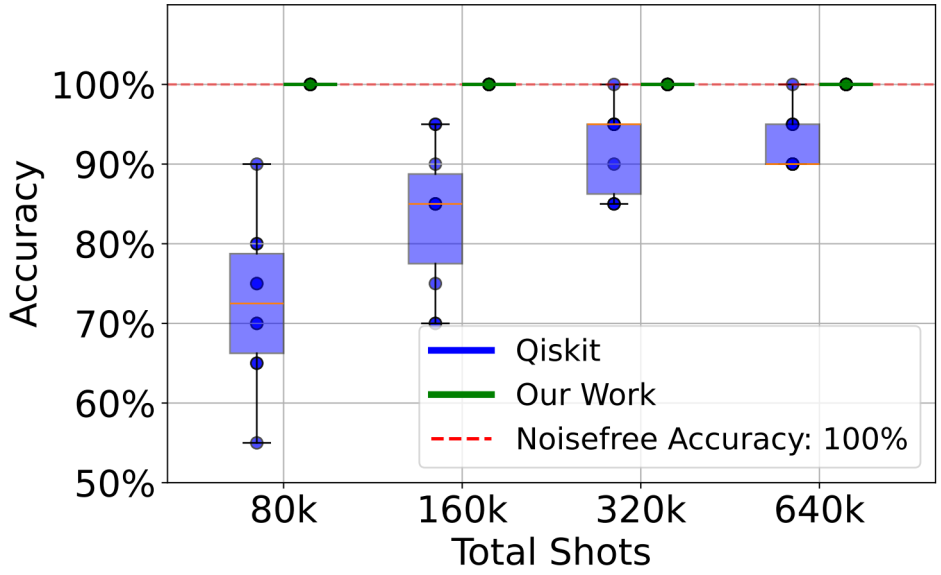
Precision and accuracy

- Higher precision.
- Same accuracy.
- #trials: 100
- #total shots: 100K

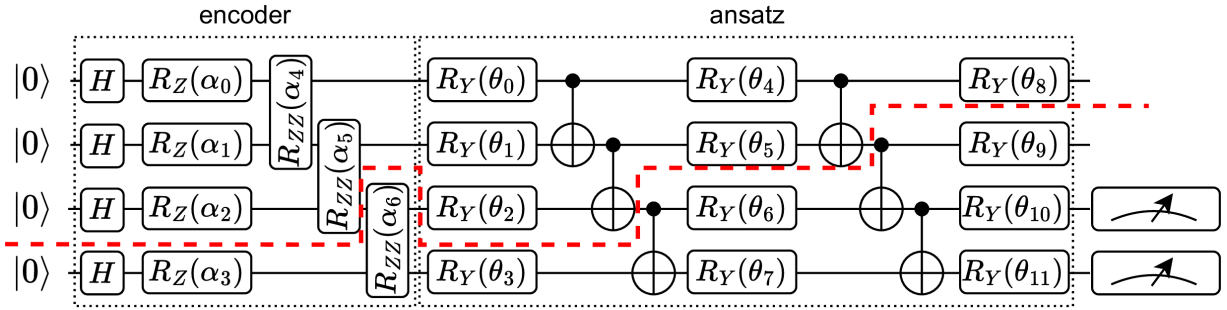


End-to-end accuracy on QNN

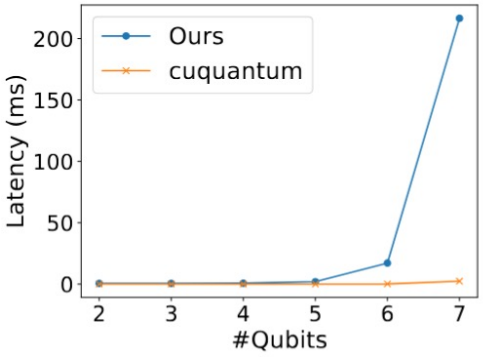
- Dataset: IRIS 2-classification



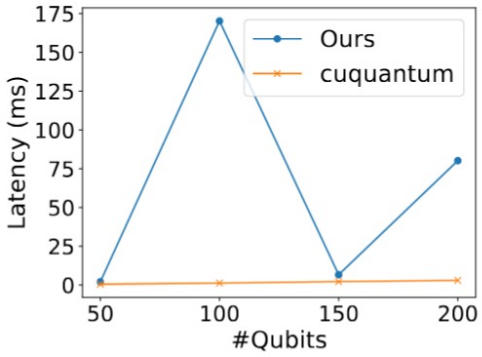
QNN



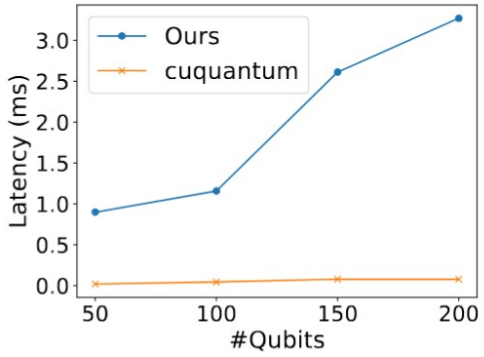
Classical postprocessing



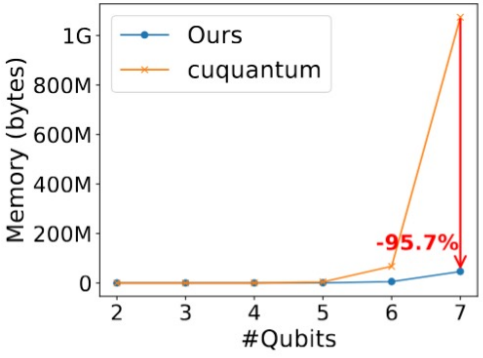
(g) QFT latency



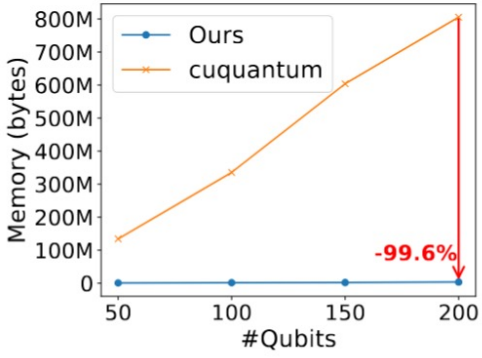
(h) AQFT latency



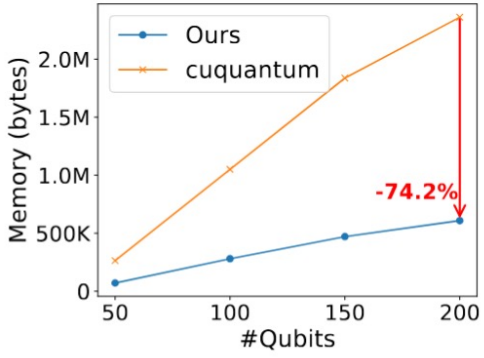
(i) 3-local HWEA (#Qubits,1,20) latency



(j) QFT memory footprint



(k) AQFT memory footprint

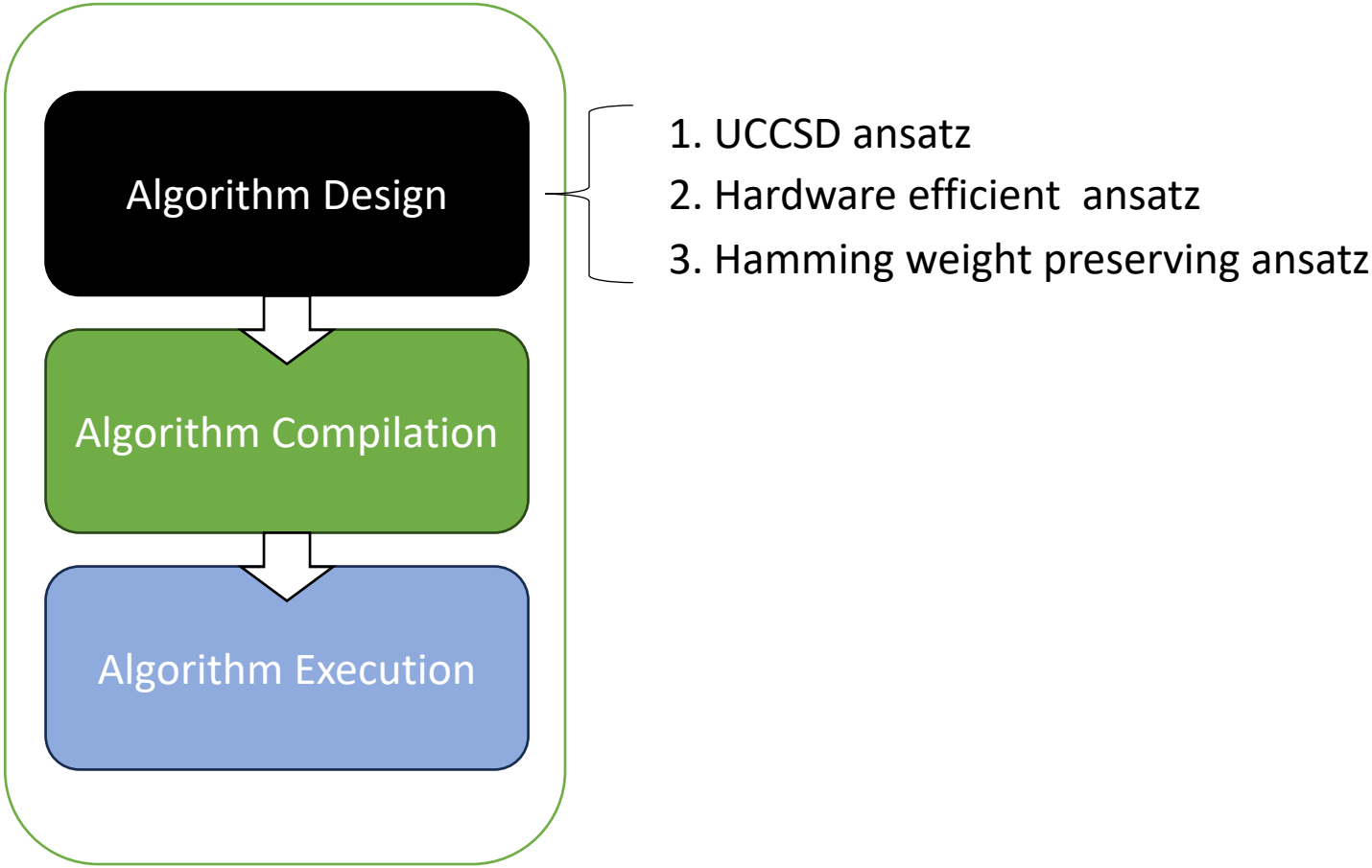


(l) 3-local HWEA (#Qubits,1,20) mem. footprint

Outline

- Background of quantum chemistry
- Quantum chemistry algorithm compilation
- Quantum chemistry algorithm execution
- Quantum chemistry algorithm design

Quantum chemistry simulation stack



What’s a good ansatz?

- Accuracy
- Trainability
- Noise-resilience

Different ansatzes

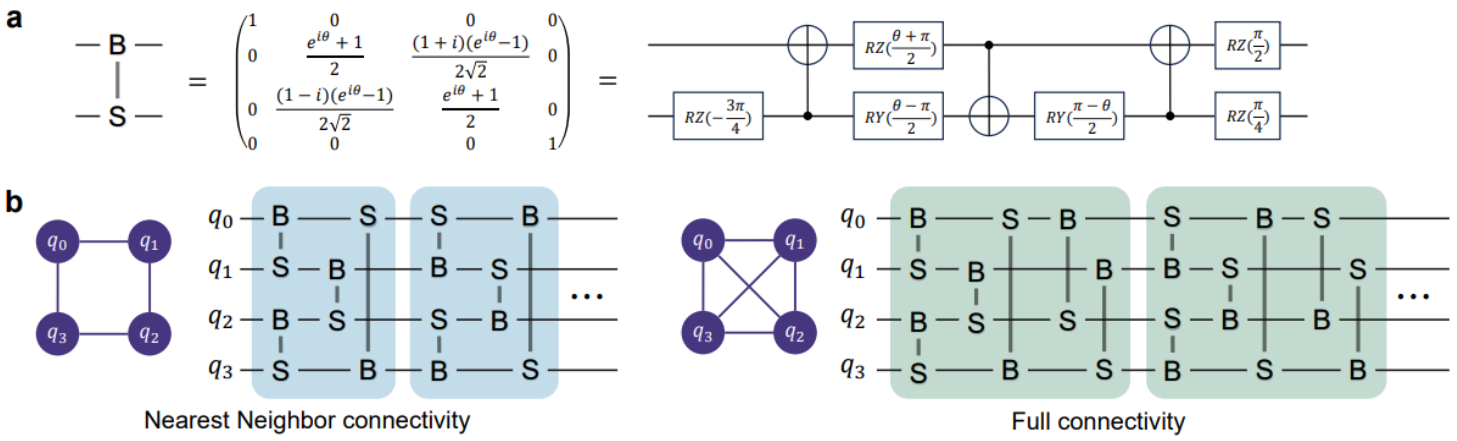
- $E_0 = \min_{\vec{\theta}} \langle \Psi(\vec{\theta}) | \hat{H} | \Psi(\vec{\theta}) \rangle$
- $|\Psi(\vec{\theta})\rangle$ is generated by ansatz.
- UCCSD:
  - chemistry inspired, high accuracy,
  - but too deep, hard to implement.
- HWEA:
  - hardware efficient, easy to implement,
  - but too weak, low accuracy.

Hamming weight preserving ansatz

N is number of orbitals

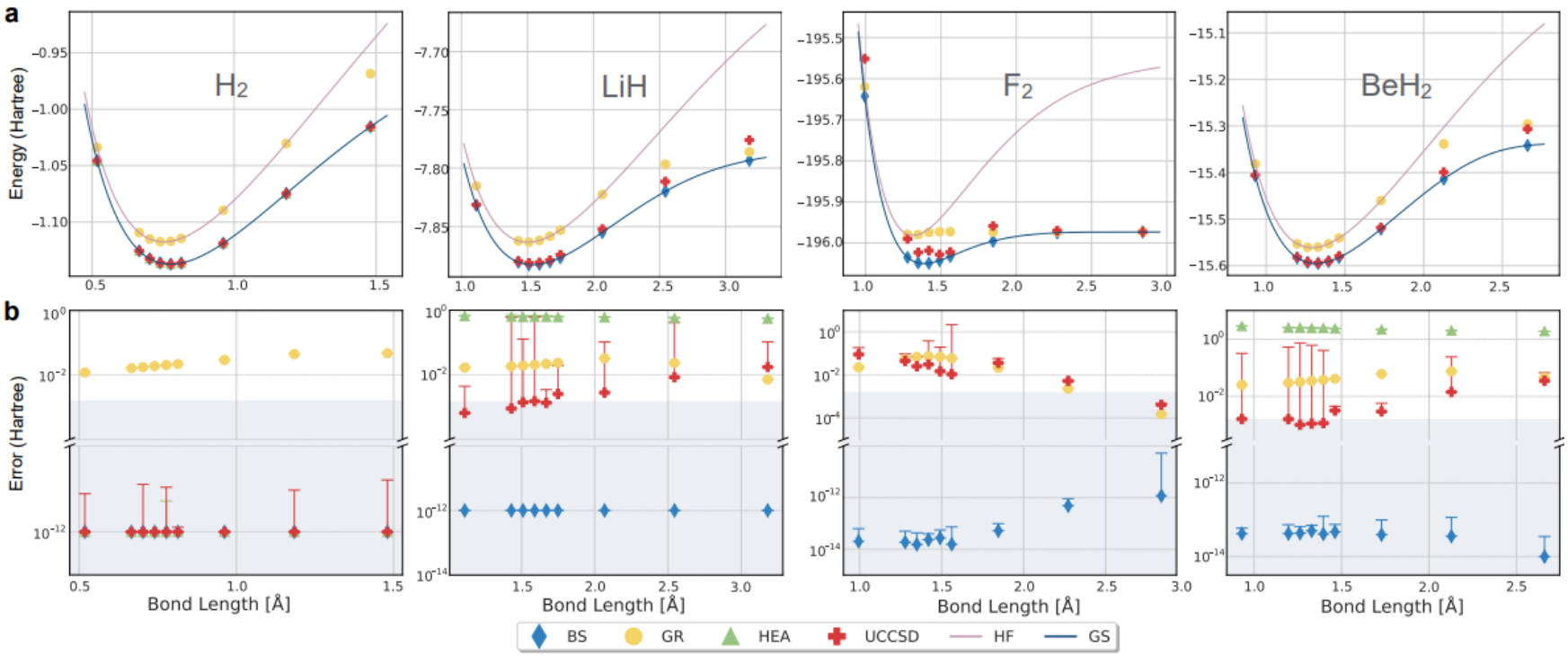
M is the number of particles

- Only explore the subspace expanded by $\binom{N}{M}$ basis.



Hamming weight preserving ansatz

- Preserve the number of particles.
- Accuracy: outperforms UCCSD



Hamming weight preserving ansatz

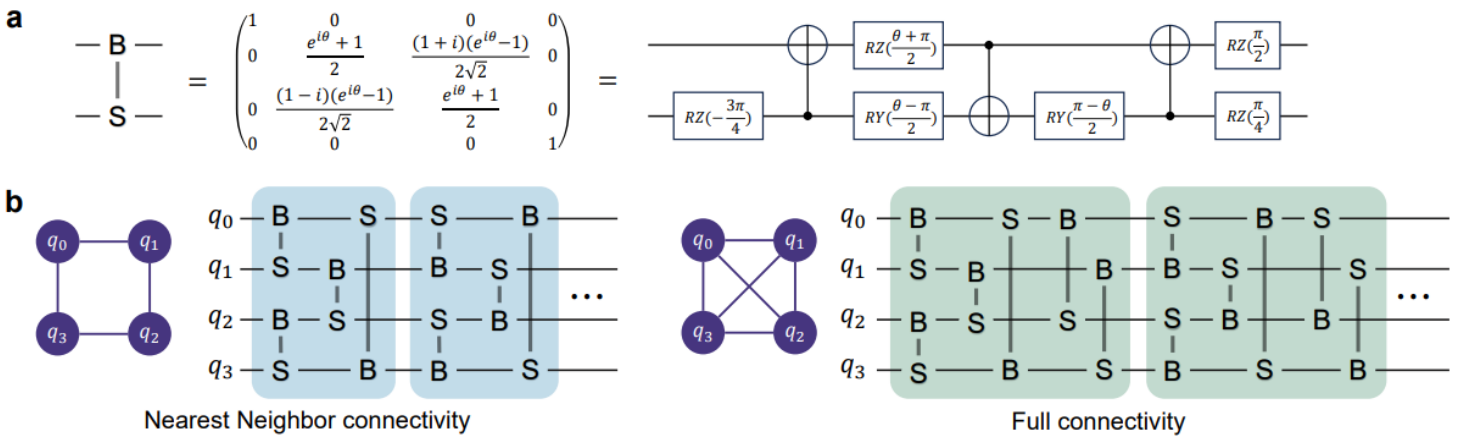
- Trainability:
  - N is number of orbitals
  - M is the number of particles
  - $Var_\theta\left[\partial_l tr\left(\rho\hat{O}\right)\right] \approx \frac{1}{\binom{N}{N/2}}$ when $M = \frac{N}{2}$
  - $Var_\theta\left[\partial_l tr\left(\rho\hat{O}\right)\right] \approx \frac{16}{N^3}$ when $M = 1$

  (Credit: https://arxiv.org/pdf/2412.04825)

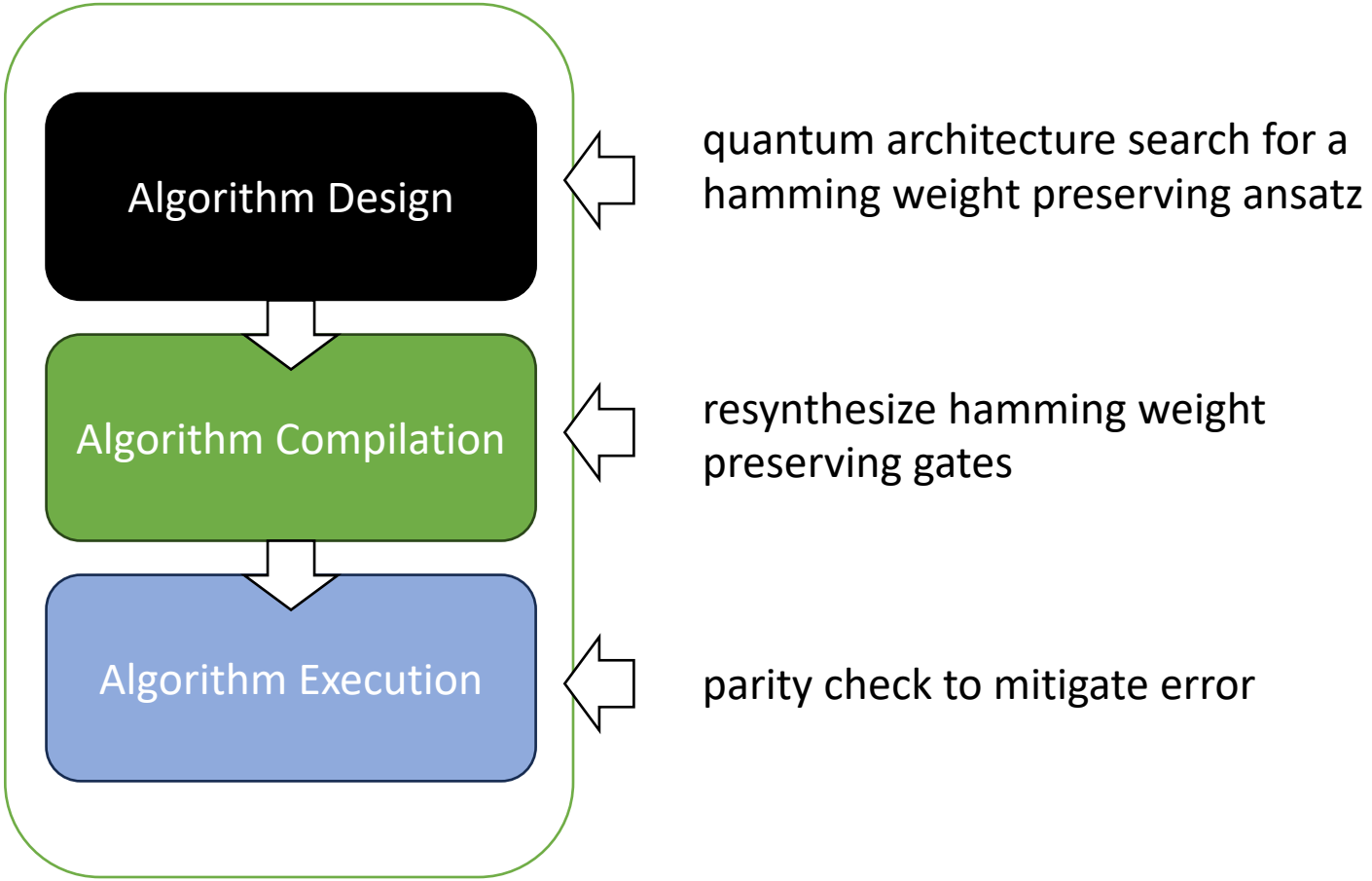
  - For reference: trainability for 2-design HWEA :
    - $Var_\theta\left[\partial_l tr\left(\rho\hat{O}\right)\right] \approx \frac{1}{2^{3n-1}} tr(H^2)tr(\rho^2)tr\left(\hat{O}^2\right)$

Hamming weight preserving ansatz

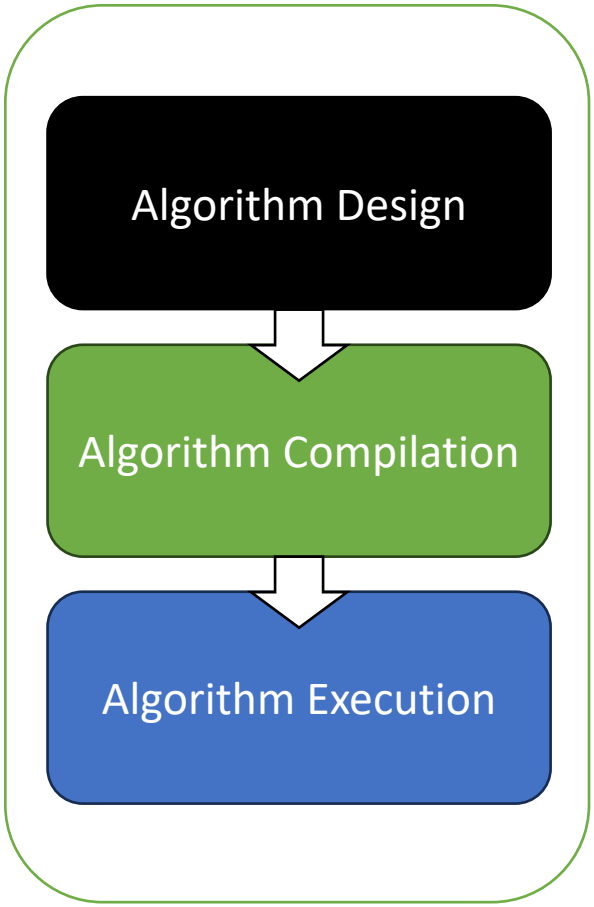
- Noise resilience:
  - Implement 1 beam-splitter (BS) gate costs 3 CNOT gates.



Future Work


Algorithm Design
Algorithm Compilation
Algorithm Execution
quantum architecture search for a hamming weight preserving ansatz
resynthesize hamming weight preserving gates
parity check to mitigate error

Q&A.



An ongoing project on Hamming weight preserving ansatz.

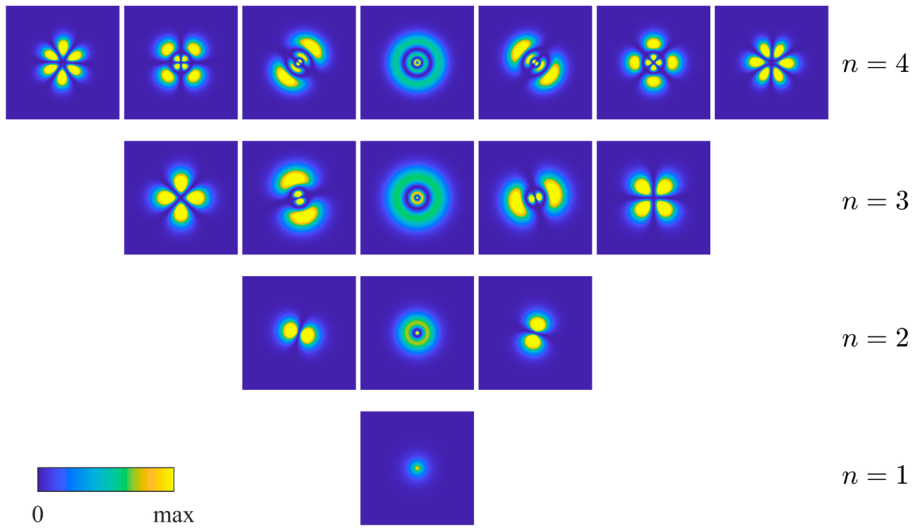
Y. Jin *et al.*, "Tetris: A Compilation Framework for VQA Applications in Quantum Computing," *ISCA 2024*

Z. Li *et al.*, "A Case for Quantum Circuit Cutting for NISQ Applications: Impact of topology, determinism, and sparsity" preprint@arxiv2412.17929

Supplementary materials

Hydrogen Atom

- $\left(-\frac{1}{2}\nabla^2 - \frac{1}{\mathrm{r}}\right)|\Psi\rangle = E|\Psi\rangle$
- $E_n = -\frac{1}{2n^2}$, n is the energy level.
- Each eigenstate ϕ_1, ϕ_2, … is an orbital.

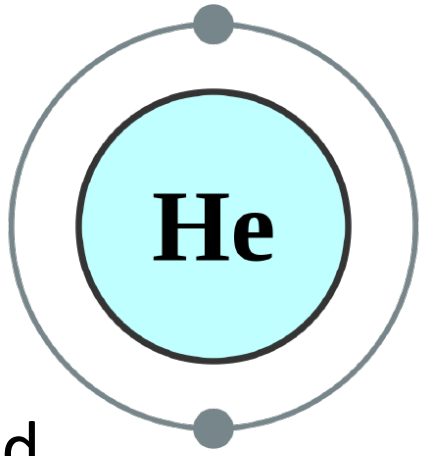


Hydrogen Atom

- Takeaway:
  - ϕ_1, ϕ_2, … are eigenstates of the Hamiltonian.
  - ϕ_1, ϕ_2, … form a complete orthonormal basis.
  - Any valid wavefunction:$|\Psi\rangle = c_1\phi_1 + c_2\phi_2 + \cdots$

Helium Atom

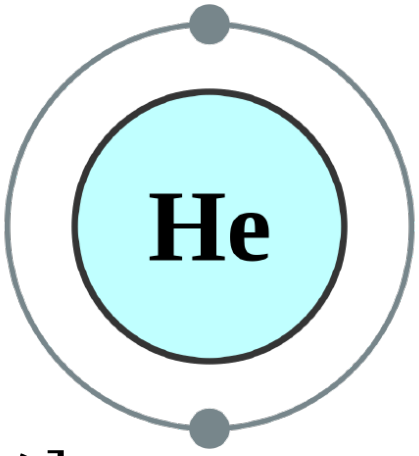
- Two electrons correlated.
- The wavefunction $\Psi(r_1, r_2)$
  is hard to solve.



- Approximation: Hartree-Fock method
  - Ignore the correlation.
  - Solve each single-electron orbitals independently: $\phi_1, \phi_2, \ldots$
  - Slater determinant to combine the orbitals

$$\Psi_{HF}(r_1, r_2) = \frac{1}{\sqrt{2!}} \begin{bmatrix} \phi_1(r_1) & \phi_2(r_1) \\ \phi_1(r_2) & \phi_2(r_2) \end{bmatrix}$$

Helium Atom



- Takeaway:
  - In a two-electron system
  - Solve single-electron orbitals:
    - $\phi_1, \phi_2, \phi_3, \phi_4 \ldots$
  - The Slater determinant of ϕ_i and ϕ_j

$$\Psi_{ij}(r_1, r_2) = \frac{1}{\sqrt{2!}} \begin{bmatrix} \phi_i(r_1) & \phi_j(r_1) \\ \phi_i(r_2) & \phi_j(r_2) \end{bmatrix}$$

  form a complete orthonormal basis for two-electron wavefunction space.
  - Any $\Psi(r_1, r_2) = \sum_{i<j} c_{ij} \Psi_{ij}$

First quantization

- N-electron wavefunction $\Psi(r_1, r_2, \dots, r_N)$.
  - Solve single-electron orbitals $\phi_1, \phi_2, \phi_3, \phi_4$ …
  - Slater determinant of n orbitals to form the basis.
  - Linear combination of basis to form any wavefunction.
- Redundancy in first quantization:
  - Electrons are identical.
  - $\Psi(r_1, r_2, r_3, \dots, r_N) = -\Psi(r_2, r_1, r_3, \dots, r_N)$
  - Use bitstring to represent occupation of orbitals.
  - 01100100… means the basis that's the Slater determinant of ϕ_2, ϕ_3, ϕ_6.

Second quantization

- First quantization to second quantization
  - Cartesian space to Fock space
    - A Fock space basis: $|0101\rangle_F$ means among the 4 orbitals, orbital 2 and 4 occupied, orbital 1 and 3 unoccupied
    - A Cartesian space wavefunction:

$$\Psi(r_1, r_2) = \frac{1}{\sqrt{2!}} \begin{bmatrix} \phi_2(r_1) & \phi_4(r_1) \\ \phi_2(r_2) & \phi_4(r_2) \end{bmatrix}$$

  - Hamiltonian: gradient and position operator to creation and annihilation operator
    - Creation/annihilation operator: $a^\dagger$ and a.

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$$

Second Quantization

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$$

- Creation operator: $a^\dagger |0\rangle_F = |1\rangle_F,\ a^\dagger |1\rangle_F = 0$
- Annihilation operator: $a|0\rangle_F = 0,\ a|1\rangle_F = |0\rangle_F$

$$h_{pq} = \int \phi_p^*(r) \left(-\frac{1}{2}\nabla^2 - \sum_A \frac{Z_A}{|r - R_A|} \right) \phi_q(r) d^3r$$

$$V_{pqrs} = \int \int \phi_p^*(r_1) \phi_q^*(r_2) \frac{1}{|r_1 - r_2|} \phi_r(r_1) \phi_s(r_2)\, d^3r_1 d^3r_2$$

- In practice: when doing integrals, use “STO-3g” to approximate the wavefunction with 3 Gaussians.

Second Quantization

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} V_{pqrs} a_p^\dagger a_q^\dagger a_s a_r$$

- Takeaway:
  - In practice, set a limit for M orbitals.
  - The Hamiltonian becomes a $2^M \times 2^M$ Hermitian matrix.
  - Solve Schrödinger equation: $H|\Psi\rangle = E|\Psi\rangle$
    - A PDE problem to a Matrix eigenvalue problem.

Quantum chemistry algorithm

$$E_0 = \min_{\vec{\theta}} \langle \Psi(\vec{\theta}) | \hat{H} | \Psi(\vec{\theta}) \rangle$$

- Unitary Coupled Cluster with Singles and Doubles
  - $|\Psi(\vec{\theta})\rangle = e^{T(\vec{\theta}) - T^{\dagger}(\vec{\theta})} |\Psi_{HF}\rangle$
  - $T(\vec{\theta}) = \sum_{p,q} \theta_{pq} a_p^{\dagger} a_q + \sum_{p<q,r<s} \theta_{pqrs} a_p^{\dagger} a_q^{\dagger} a_s a_r$

- $A = e^B$, B is anti-Hermitian⇒ A is unitary
- First-Order Trotter Approximation

$$e^{A+B} = \lim_{n \to \infty} \left(e^{A/n} e^{B/n} \right)^n$$

Unitary Coupled Cluster with Singles and Doubles

- After first-order trotter approximation with n=1:

$$|\Psi(\vec{\theta})\rangle = \prod_{p,q} e^{\theta_{pq}\left(a_p^\dagger a_q - a_q^\dagger a_p\right)} \prod_{p<q,r<s} e^{\theta_{pqrs}\left(a_p^\dagger a_q^\dagger a_s a_r - a_r^\dagger a_s^\dagger a_q a_p\right)} |\Psi_{HF}\rangle$$

- Each p,q or p,q,r,s corresponding exponential operation is unitary.
- How to implement on quantum computer?

Creation/Annihilation operator

- Anticommutation: $\{A, B\} = AB + BA$
  - $\left\{a_p^\dagger, a_q^\dagger\right\} = 0, \{a_p, a_q\} = 0, \forall p, q$
  - $\left\{a_p^\dagger, a_q\right\} = \begin{cases} 0 \; if \; p \neq q \\ 1 \; if \; p = q \end{cases}$
- You can't create more than two electrons on one orbital.
  - $a_p^\dagger a_p^\dagger = 0$ because $\left\{a_p^\dagger, a_p^\dagger\right\} = 0$
- Swap two electrons, create a negative sign
  - $a_p^\dagger a_q^\dagger = -a_q^\dagger a_p^\dagger$ because $\left\{a_p^\dagger, a_q^\dagger\right\} = 0$

Jordan-Wigner Encoder

- Pauli matrices:

$$X = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}, Y = \begin{bmatrix} & -i \\ i & \end{bmatrix}, Z = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}$$

- $a_p \coloneqq Z_0 \dots Z_{p-1} \frac{X_p + iY_p}{2}$
- $a_p^\dagger \coloneqq Z_0 \dots Z_{p-1} \frac{X_p - iY_p}{2}$

- Example: Pauli string $YYXY \coloneqq Y \otimes Y \otimes X \otimes Y$

Jordan-Wigner Encoder

- Example: $a_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, a_0^\dagger = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
- Quantum state: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- Annihilation operator:
  - $a_0|0\rangle = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0, a_0|1\rangle = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = |0\rangle$
- Creation operator:
  - $a_0^\dagger|0\rangle = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = |1\rangle, a_0^\dagger|1\rangle = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$