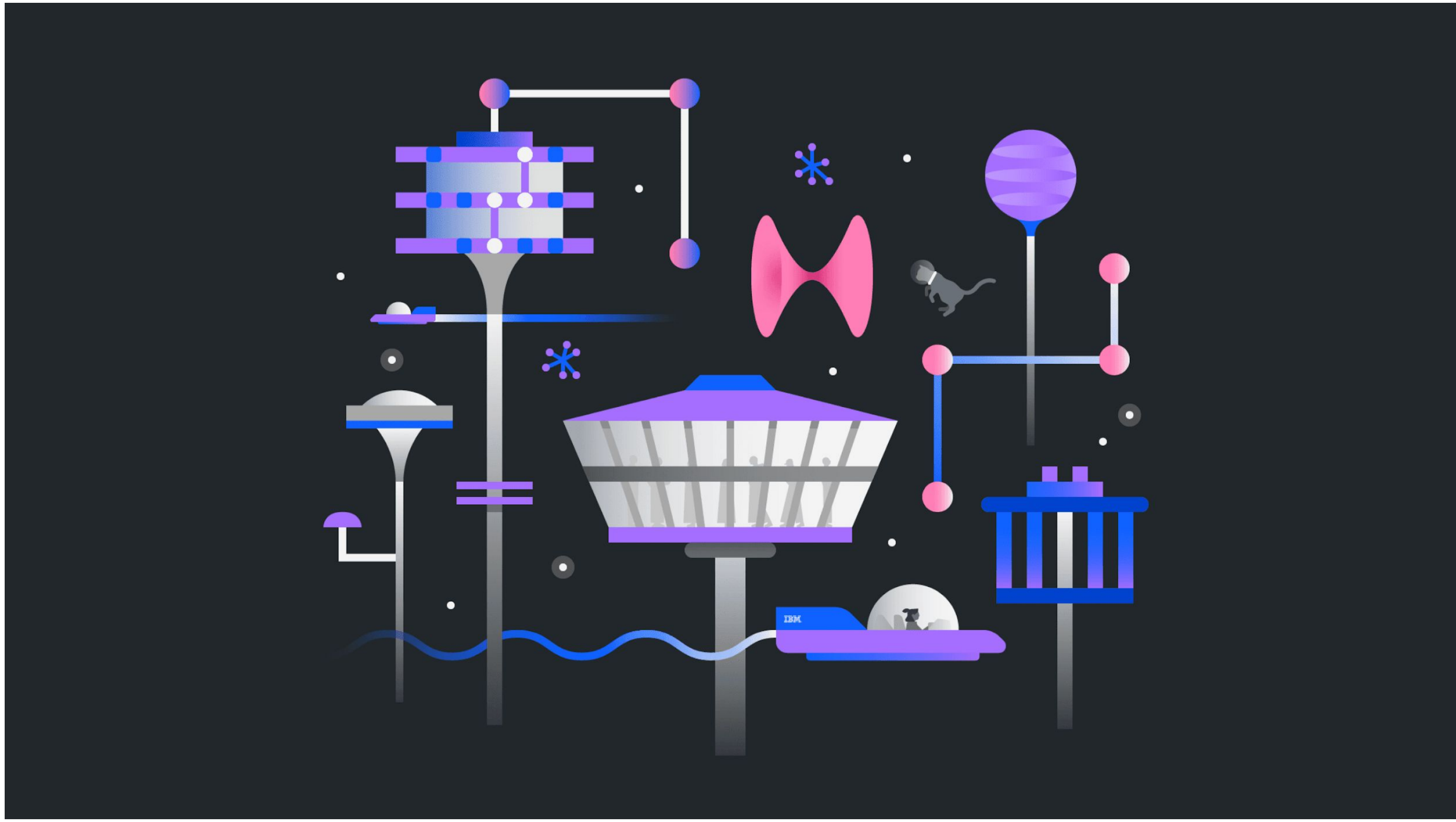
BasQ Qiskit Fall Fest 2024

Towards implementation of ADAPT-VQE in real quantum hardware



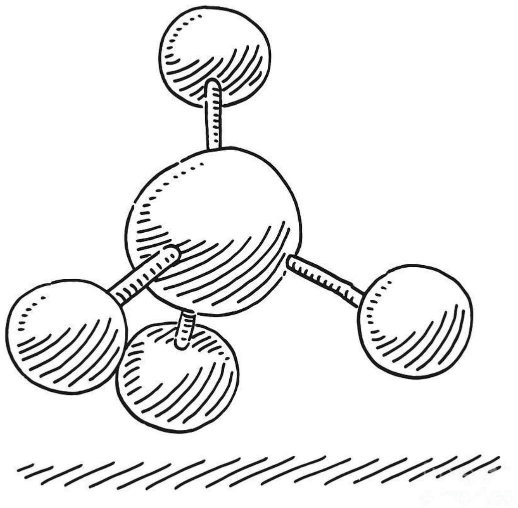
Nonia Vaquero Sabater
PhD Student

BasQ
Basque Quantum

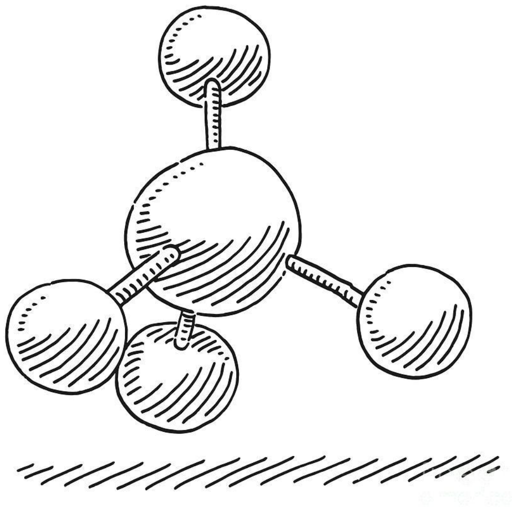
QISKIT 2024 2024 FALL FEST

Quantum Chemistry

- Ground state energy $\Longrightarrow$ Minimum eigenvalue



Quantum Chemistry

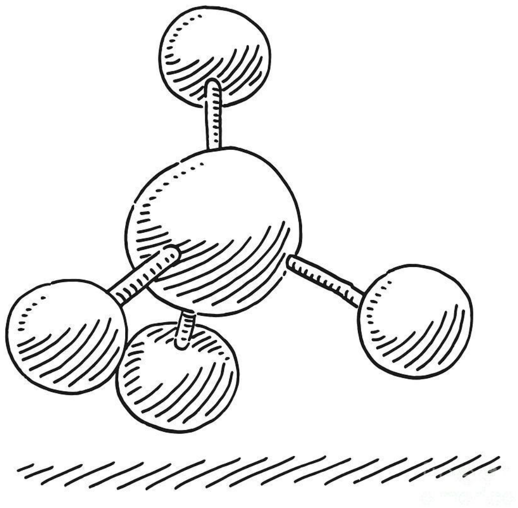


- Ground state energy $\Longrightarrow$ Minimum eigenvalue

- Molecular Hamiltonian in second quantization

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

Quantum Chemistry



- Ground state energy $\Longrightarrow$ Minimum eigenvalue

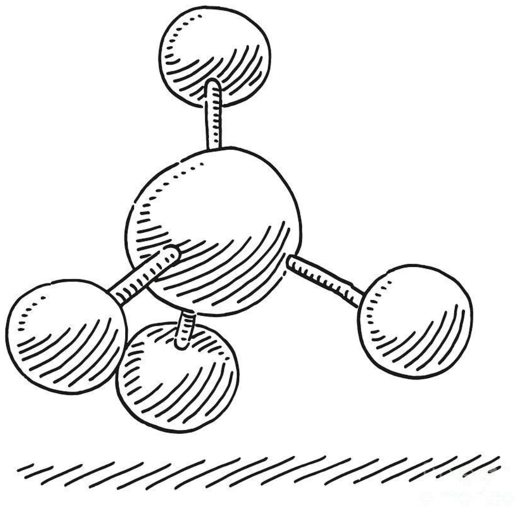
- Molecular Hamiltonian in second quantization

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

- Define a fermion to qubit mapping

Jordan-Wigner: $|\chi_i\rangle \rightarrow |q_i\rangle$

Quantum Chemistry



- Ground state energy $\Longrightarrow$ Minimum eigenvalue
- Molecular Hamiltonian in second quantization

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \Longrightarrow H = \sum_i h_i P_i \quad with \quad P_i = \prod_j p_j, \quad p_j \in \{X, Y, Z\}$$

- Define a fermion to qubit mapping

Jordan-Wigner: $|\chi_i\rangle \rightarrow |q_i\rangle$

BasQ Basque Quantum

QISKIT 2024 2024 FALL FEST

Variational Quantum Eigensolver (VQE)

Cost function:

Ansatz:

Variational Quantum Eigensolver (VQE)

Cost function:

$$\langle \psi(\theta) | H | \psi(\theta) \rangle$$

Ansatz:

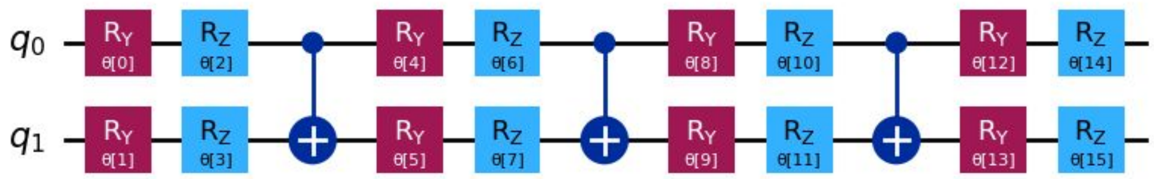
Variational Quantum Eigensolver (VQE)

Cost function:

$$\langle\psi(\theta)|H|\psi(\theta)\rangle$$

Ansatz:

$$|\psi(\theta)\rangle$$



Variational Quantum Eigensolver (VQE)

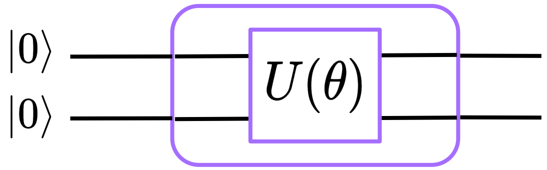
Cost function:

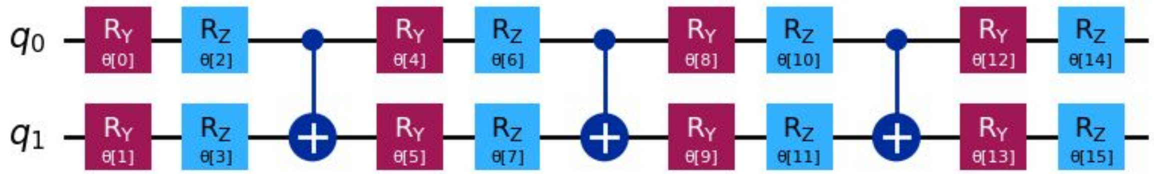
$$\langle \psi(\theta) | H | \psi(\theta) \rangle$$

Ansatz:

$$|\psi(\theta)\rangle$$

1. Prepare trial state $|\psi(\theta)\rangle$





Variational Quantum Eigensolver (VQE)

Cost function:

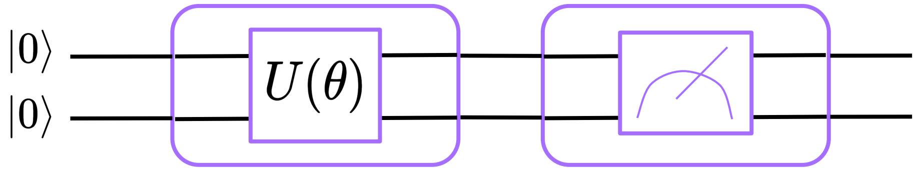
$$\langle \psi(\theta) | H | \psi(\theta) \rangle$$

Ansatz:

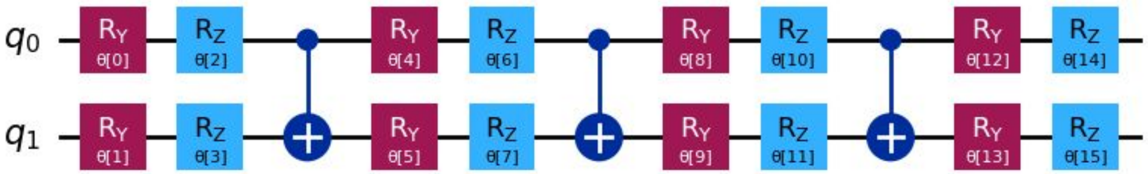
$$|\psi(\theta)\rangle$$

1. Prepare trial state $|\psi(\theta)\rangle$

2. Measure cost function



$$\langle \psi(\theta) | H | \psi(\theta) \rangle$$



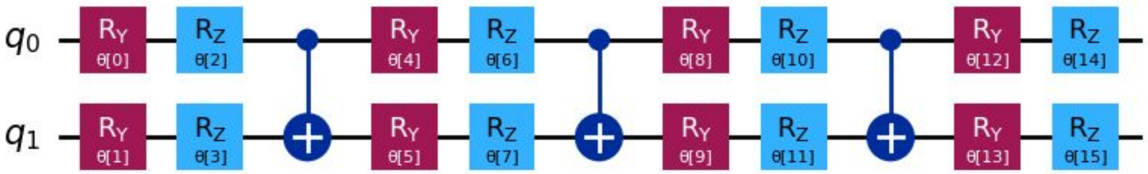
Variational Quantum Eigensolver (VQE)

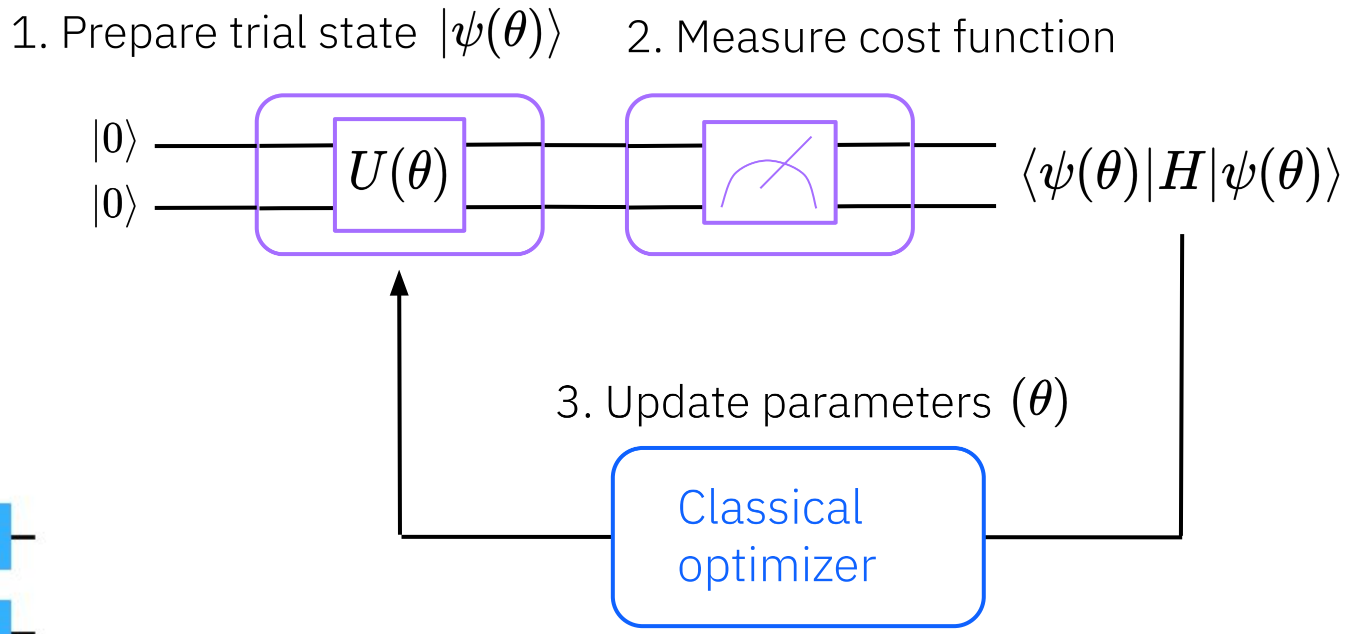
Cost function:

$$\langle \psi(\theta) | H | \psi(\theta) \rangle$$

Ansatz:

$$|\psi(\theta)\rangle$$



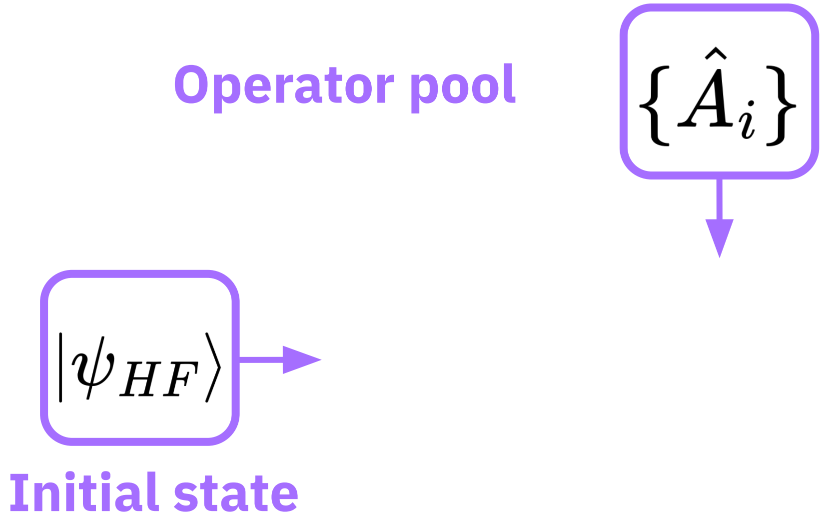


ADAPT-VQE

$$\boxed{|\psi_{HF}\rangle} \rightarrow$$

Initial state

ADAPT-VQE



ADAPT-VQE

Operator pool $\{\hat{A}_i\}$

Operator selection

$|\psi_{HF}\rangle \rightarrow$ $$\frac{\partial E}{\partial \theta_i} = \langle \psi | [\hat{H}, \hat{A}_i] | \psi \rangle$$ $\rightarrow$

Initial state

ADAPT-VQE

Operator pool

$$\{\hat{A}_i\}$$

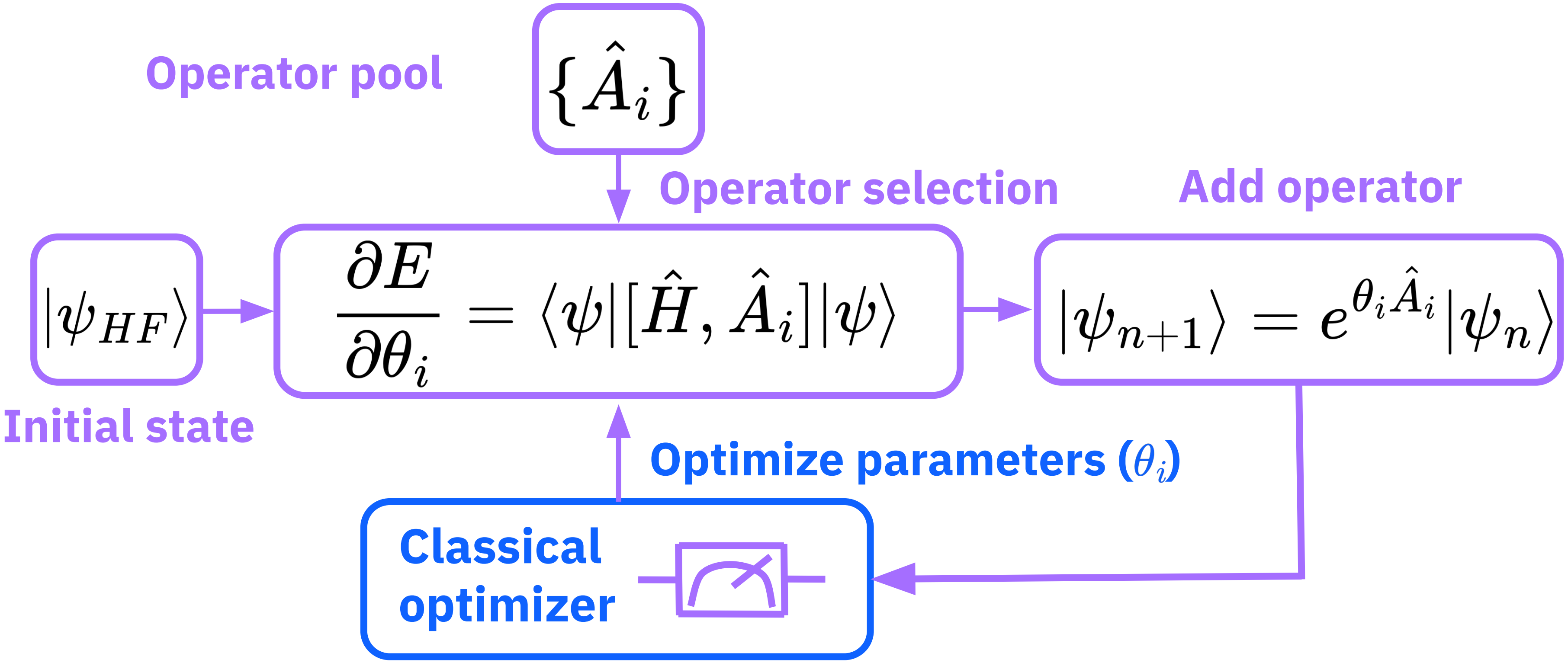
Operator selection

Add operator

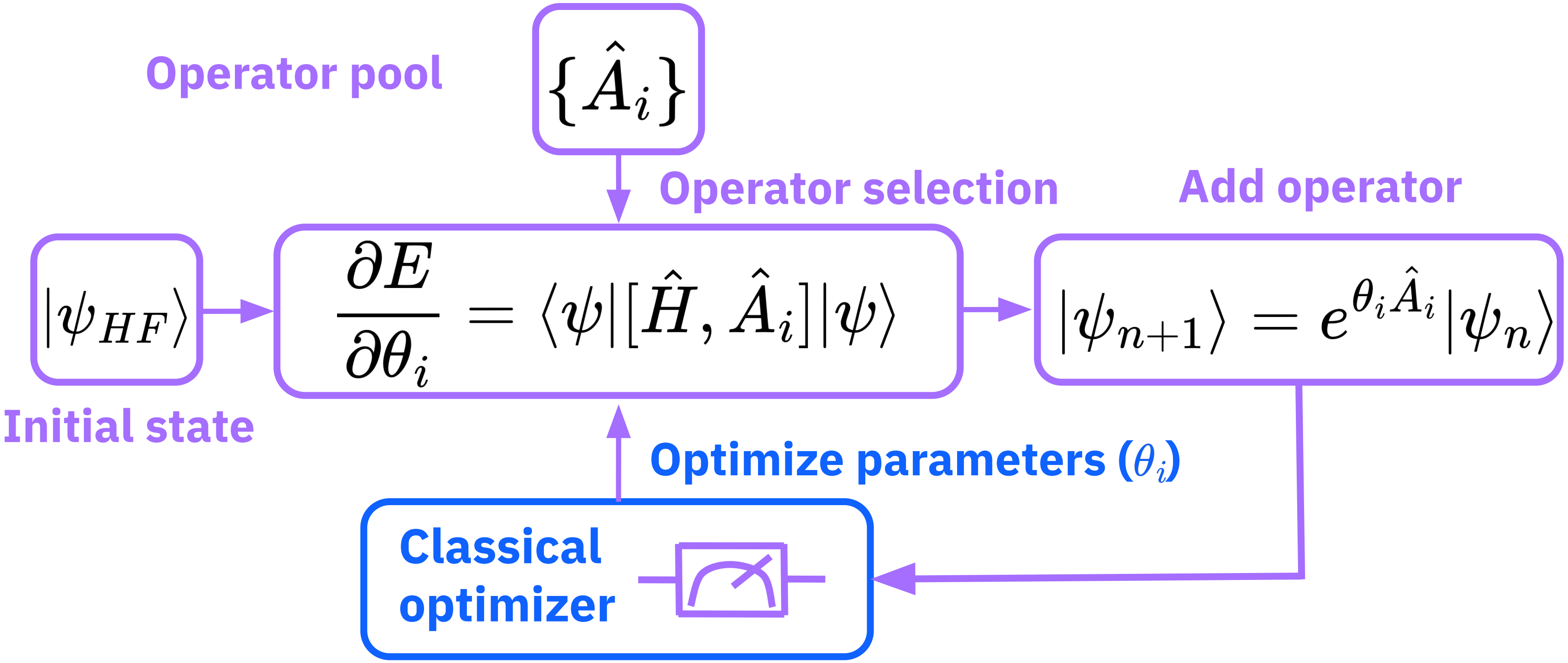
$$|\psi_{HF}\rangle \rightarrow \frac{\partial E}{\partial \theta_i} = \langle \psi | [\hat{H}, \hat{A}_i] | \psi \rangle \rightarrow |\psi_{n+1}\rangle = e^{\theta_i \hat{A}_i} |\psi_n\rangle$$

Initial state

ADAPT-VQE


Operator pool
$\{\hat{A}_i\}$
Operator selection
Add operator
$|\psi_{HF}\rangle$
$\frac{\partial E}{\partial \theta_i} = \langle\psi|[\hat{H}, \hat{A}_i]|\psi\rangle$
$|\psi_{n+1}\rangle = e^{\theta_i \hat{A}_i}|\psi_n\rangle$
Initial state
Optimize parameters (θ_i)
Classical optimizer

ADAPT-VQE



Final state:

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i}|\psi_{HF}\rangle$$

Improving Initial State

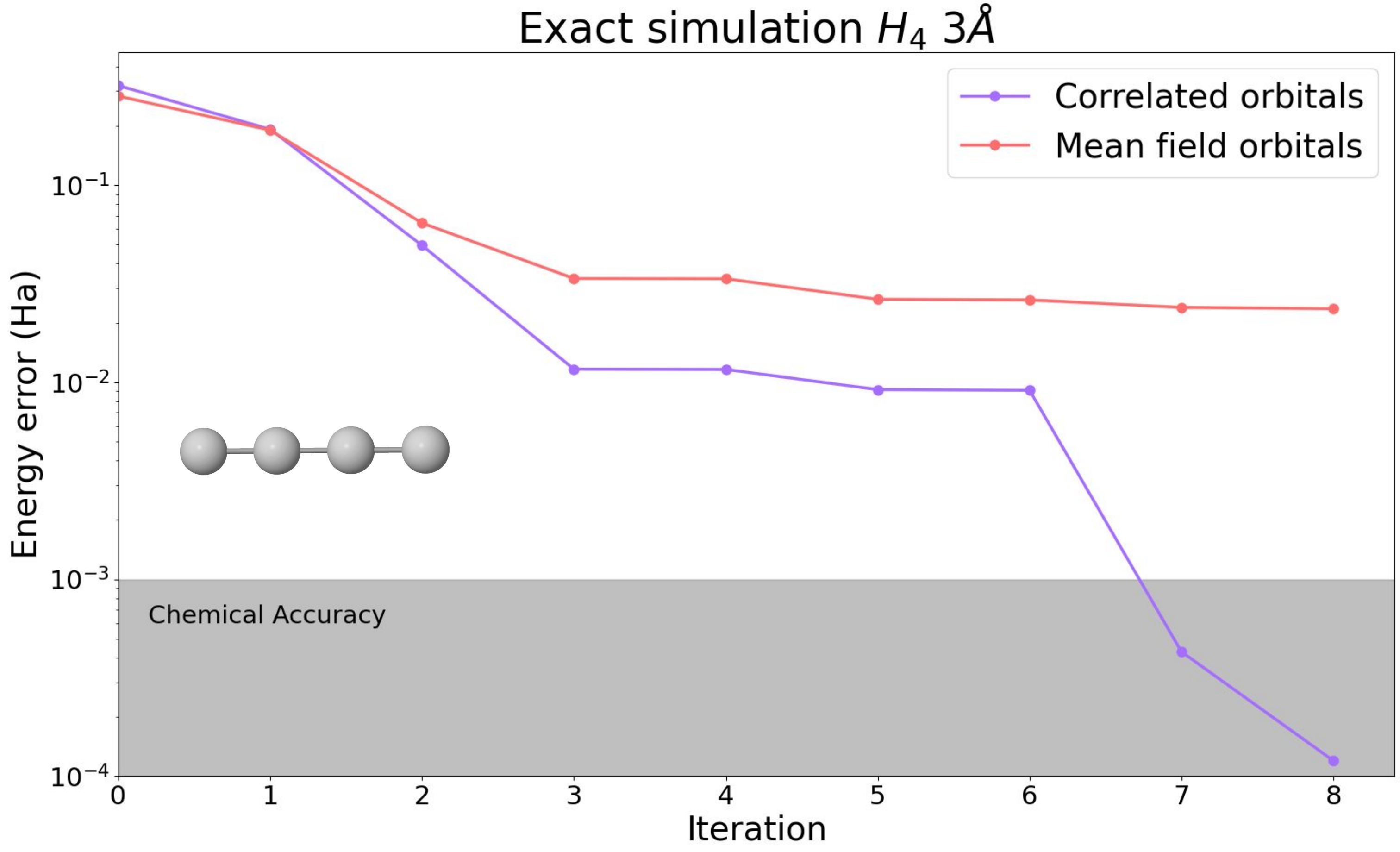
$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

Improving Initial State

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle \Longrightarrow$$ Correlated orbitals basis set

Improving Initial State

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle \Longrightarrow \text{Correlated orbitals}$$



Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

$$\hat{A} = i \sum_j P_j \quad with \quad P_j = \prod_k p_k, \quad p_k \in \{X, Y, Z\}$$

Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

$$\hat{A} = i \sum_j P_j \quad with \quad P_j = \prod_k p_k, \quad p_k \in \{X, Y, Z\}$$

$$e^{i\theta_i \sum_j P_j} \neq \prod_j e^{i\theta_i P_j}$$

when terms do not commute

Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

$$\hat{A} = i\sum_j P_j \quad with \quad P_j = \prod_k p_k, \quad p_k \in \{X, Y, Z\}$$

$$e^{i\theta_i \sum_j P_j} \approx \prod_j e^{i\theta_i P_j}$$

Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

$$e^{i\theta_i \sum_j P_j} \approx \prod_j e^{i\theta_i P_j}$$

$$e^{i\theta IXYZ}$$

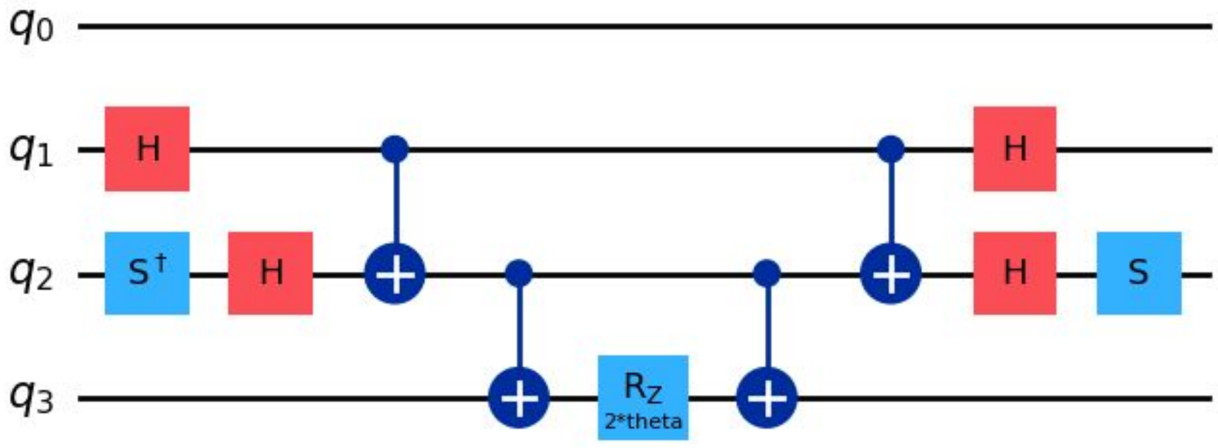
Simulation cost

- Trotterization

$$|\psi\rangle = \prod_i e^{\theta_i \hat{A}_i} |\psi_{HF}\rangle$$

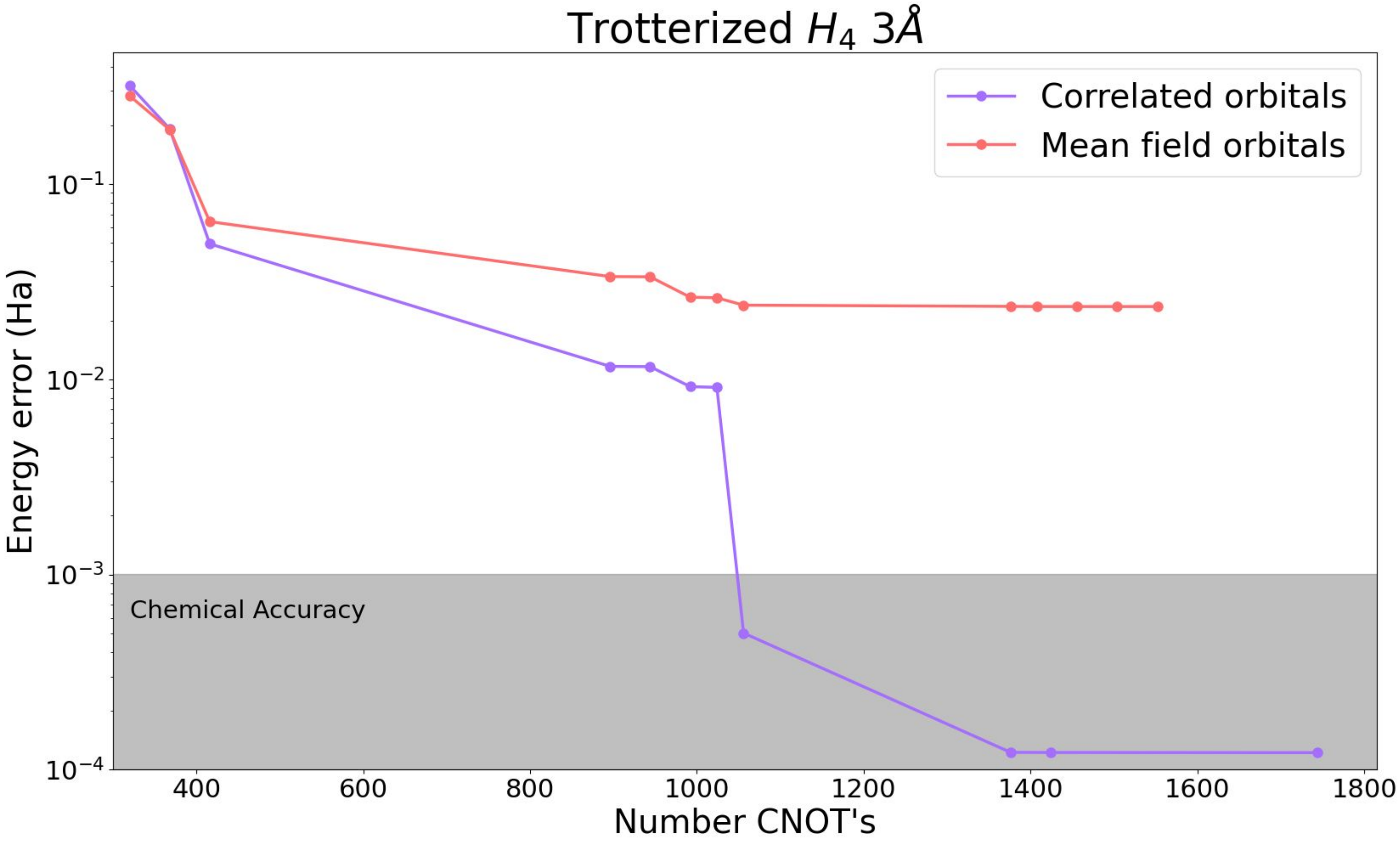
$$e^{i\theta_i \sum_j P_j} \approx \prod_j e^{i\theta_i P_j}$$

Staircase algorithm



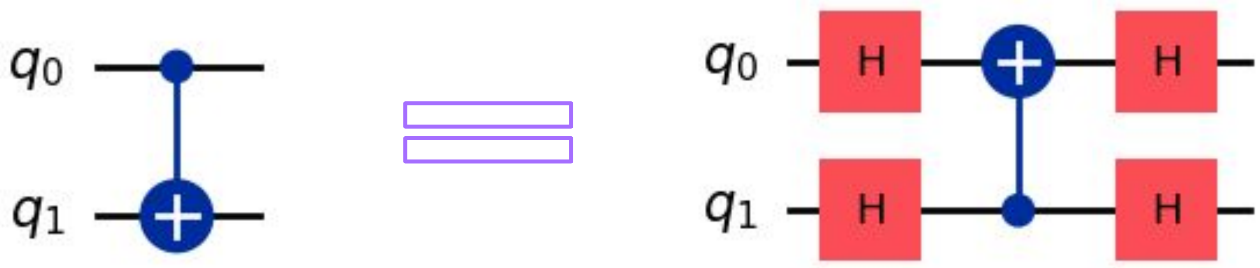
$$e^{i\theta IXYZ}$$

Simulation cost



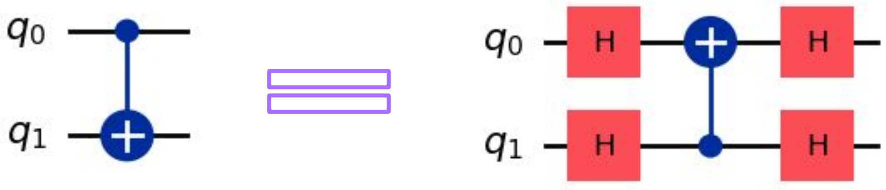
CNOT’s optimization

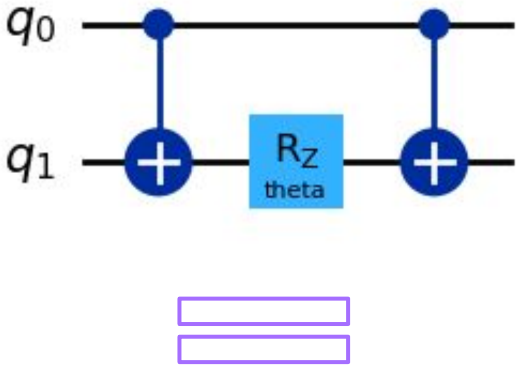
Inverse staircase algorithm

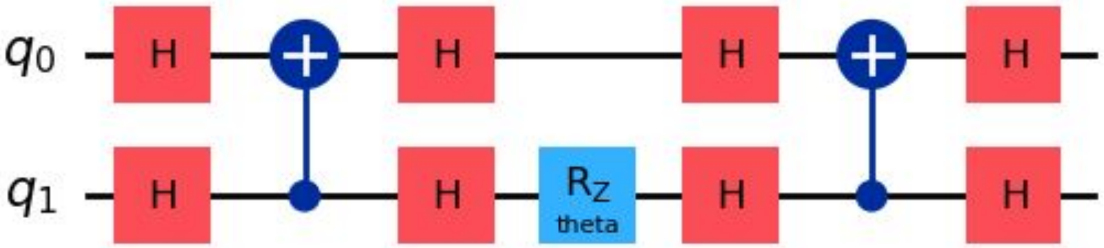


CNOT's optimization

Inverse staircase algorithm

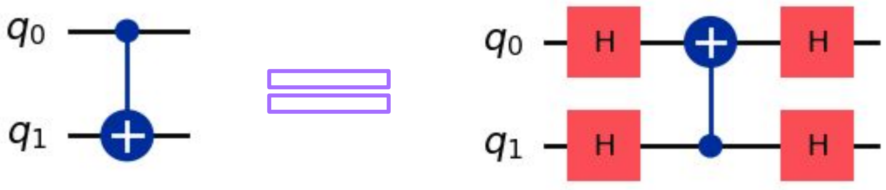


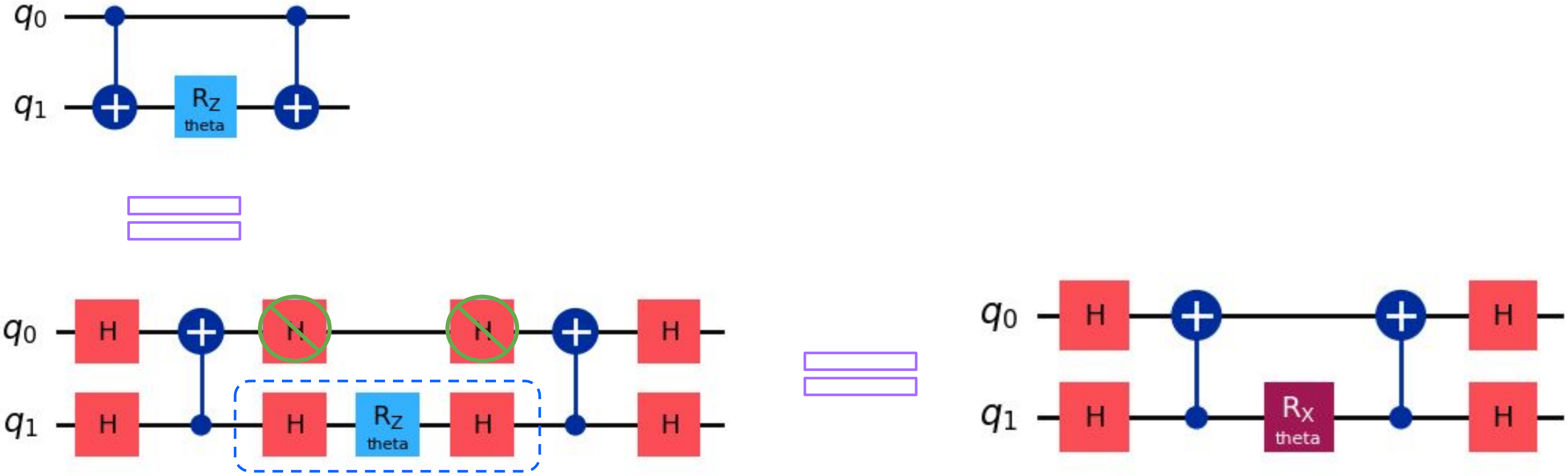




CNOT's optimization

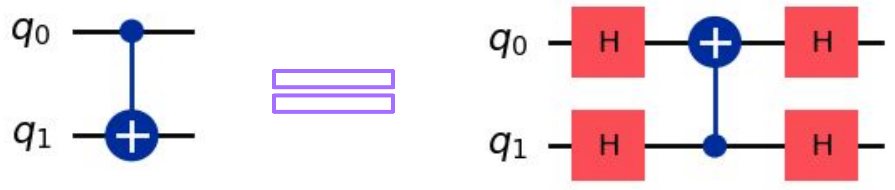
Inverse staircase algorithm



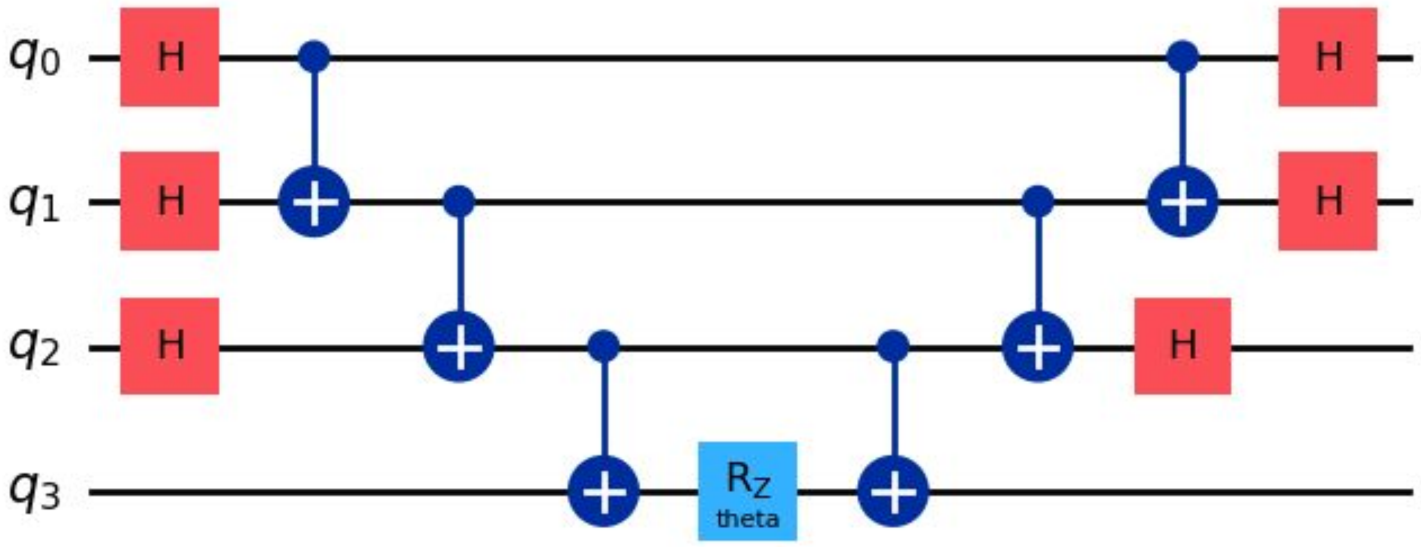


CNOT's optimization

Inverse staircase algorithm

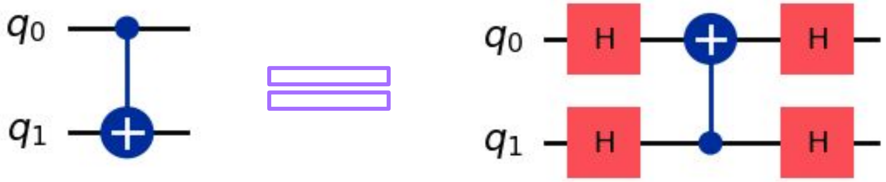


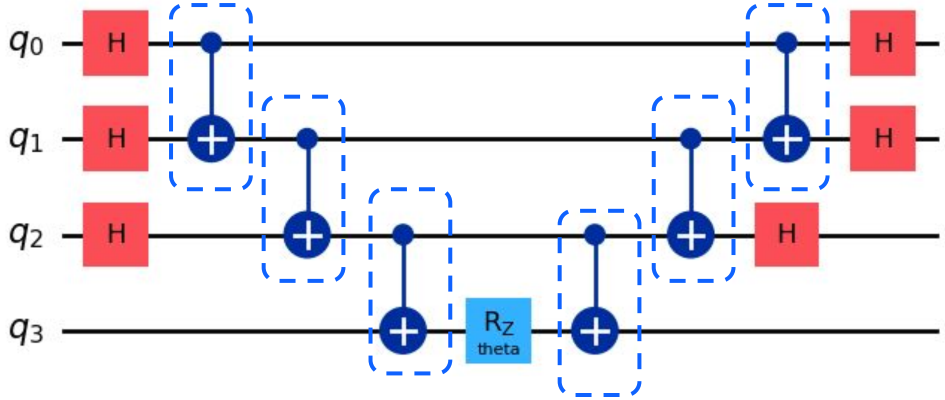
$$e^{i\theta XXXZ}$$



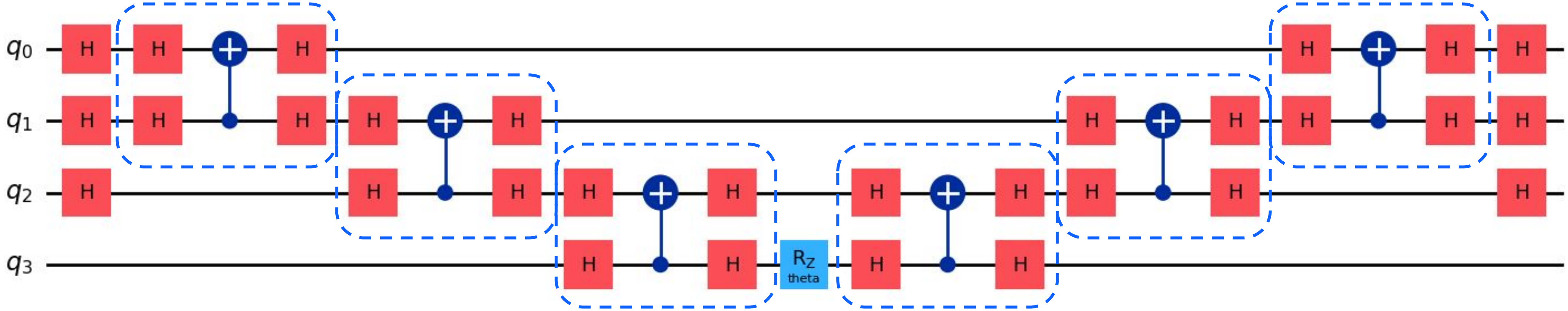
CNOT’s optimization

Inverse staircase algorithm



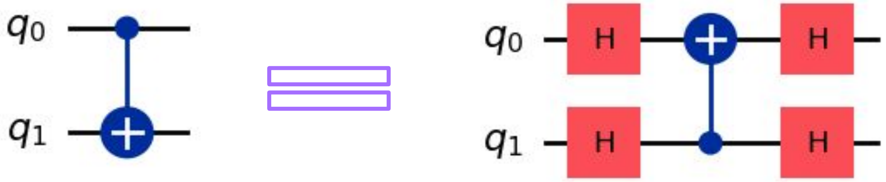


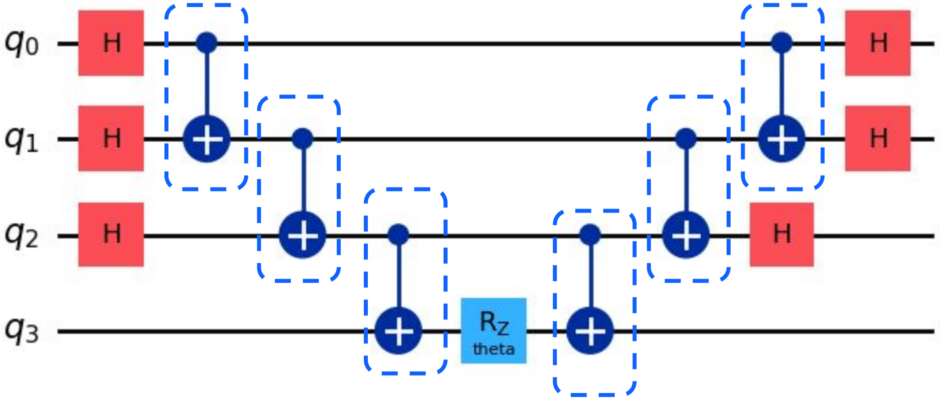
$$e^{i\theta XXXZ}$$



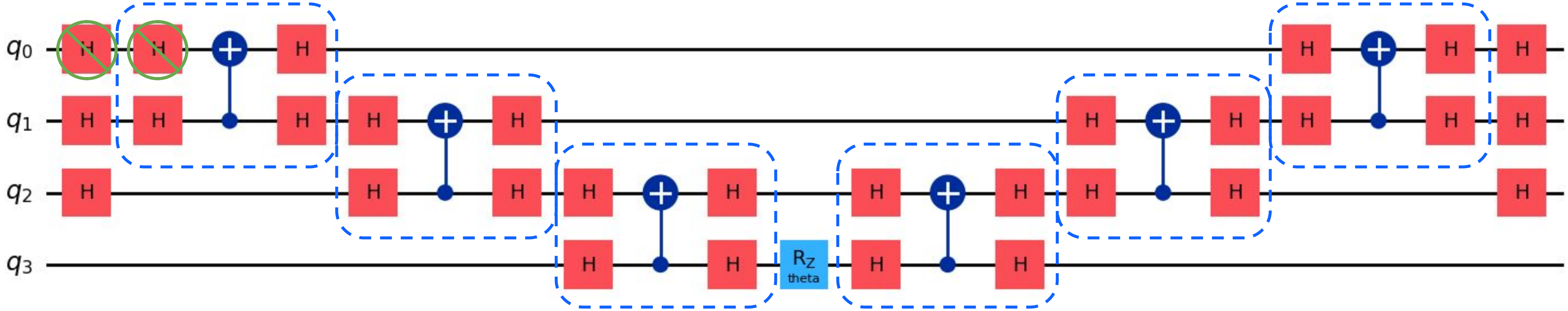
CNOT’s optimization

Inverse staircase algorithm





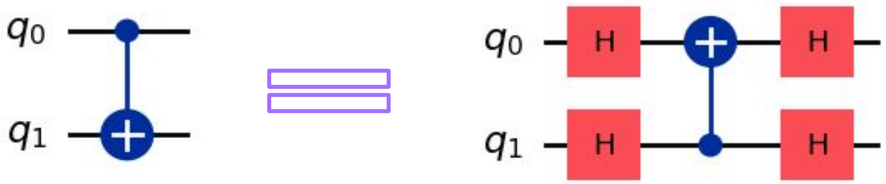
$e^{i\theta XXXZ}$

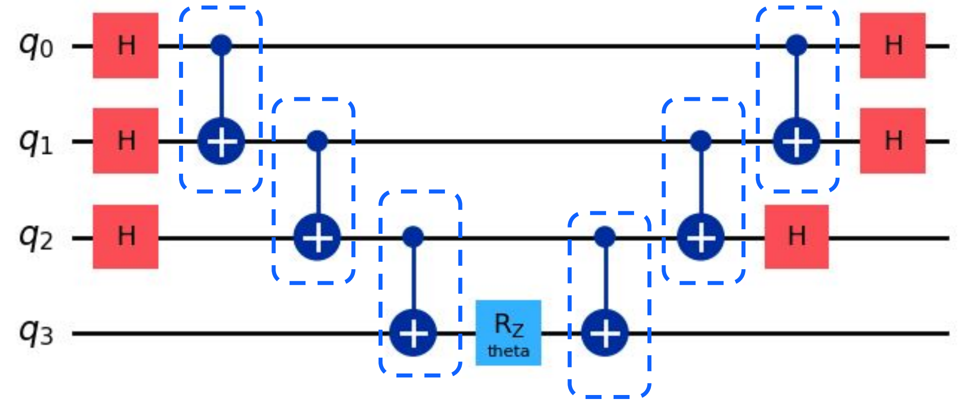


BasQ Basque Quantum

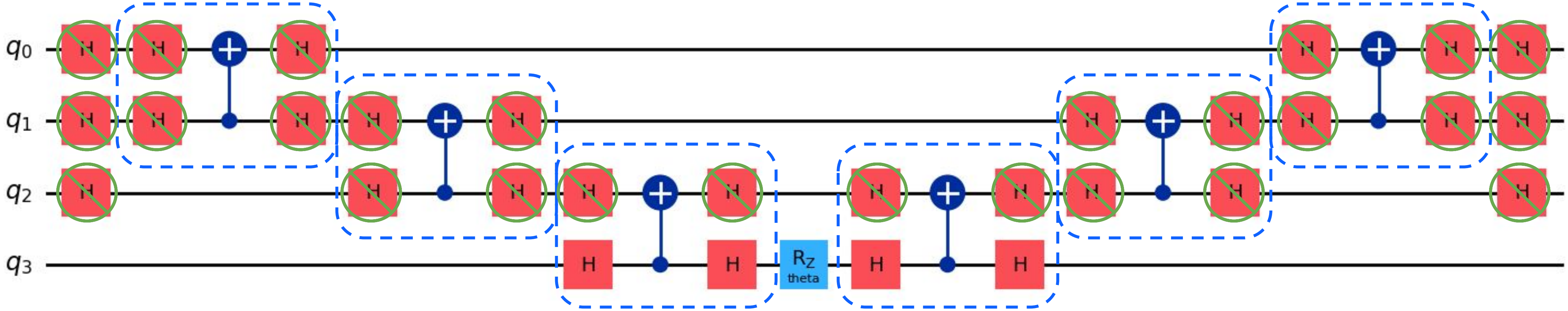
CNOT’s optimization

Inverse staircase algorithm



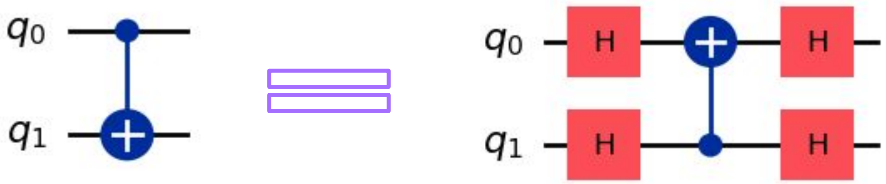


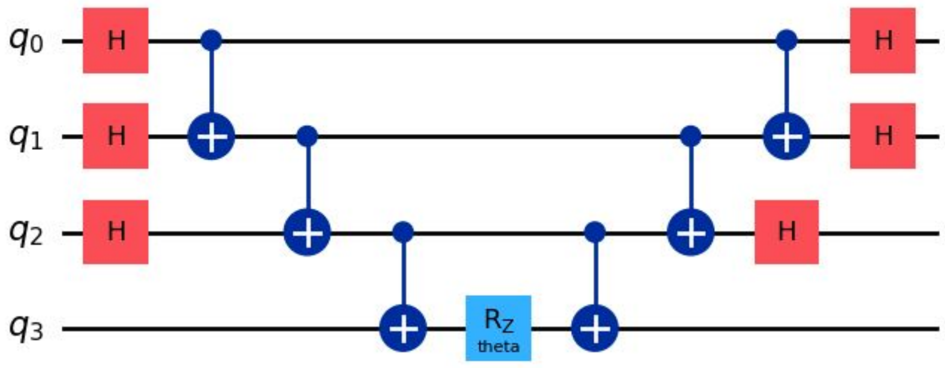
$e^{i\theta XXXZ}$



CNOT's optimization

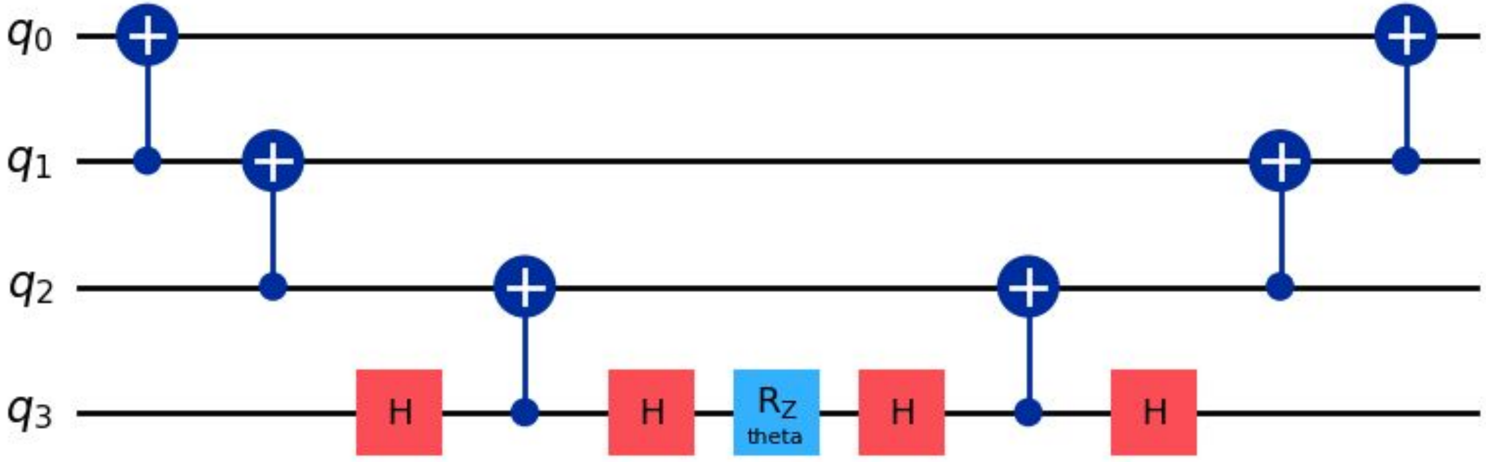
Inverse staircase algorithm





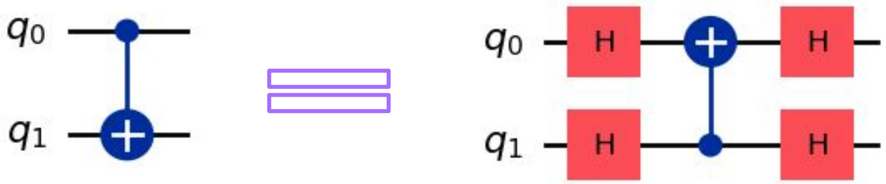
=

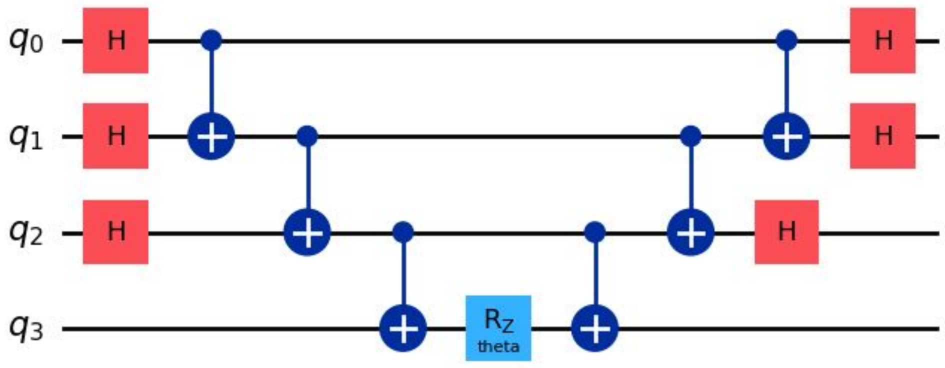
$$e^{i\theta XXXZ}$$



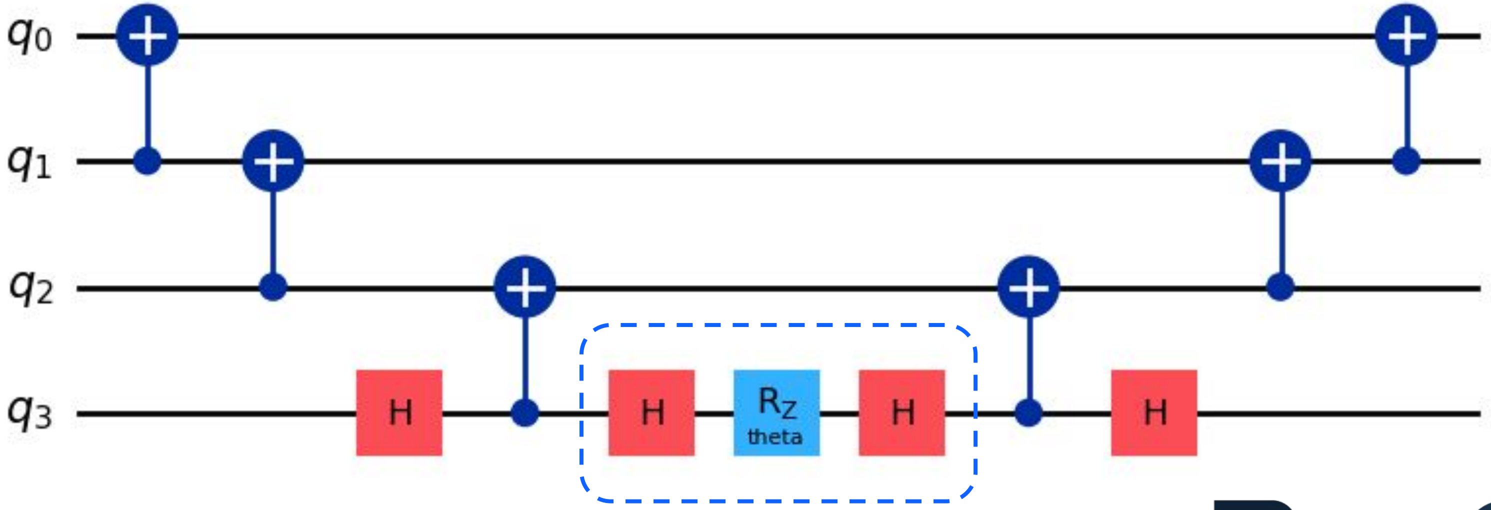
CNOT's optimization

Inverse staircase algorithm



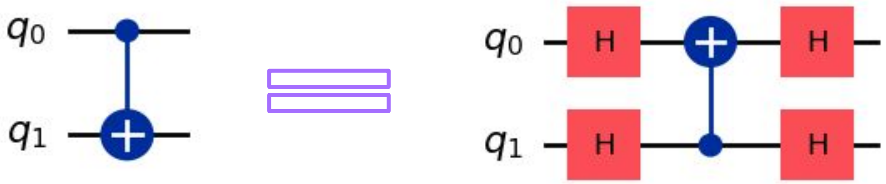


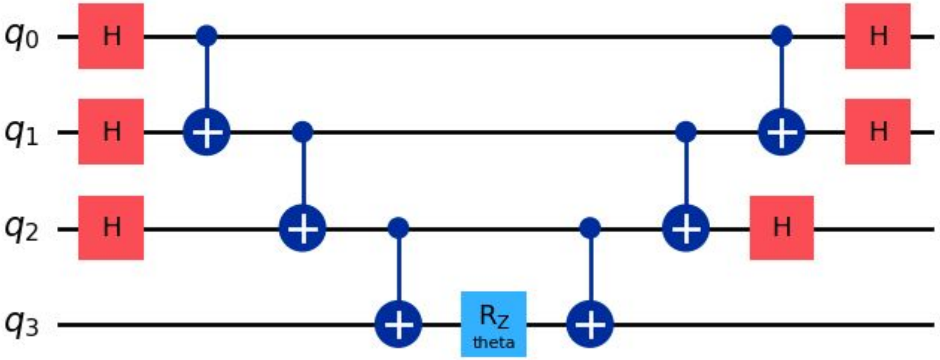
$$e^{i\theta XXXZ}$$



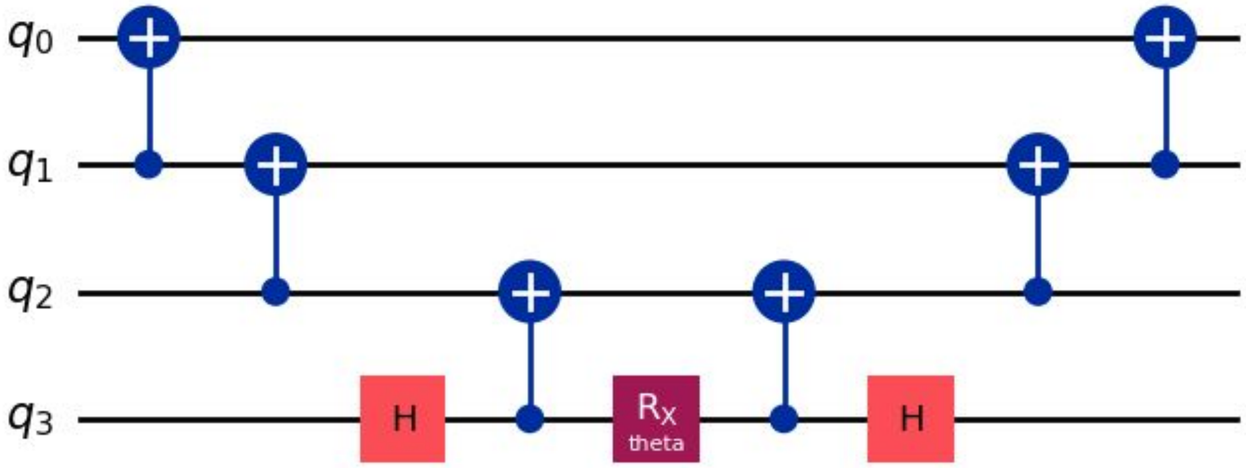
CNOT's optimization

Inverse staircase algorithm



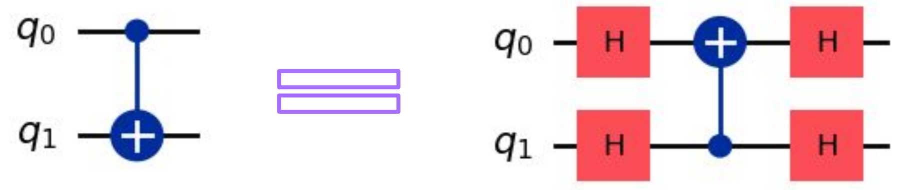


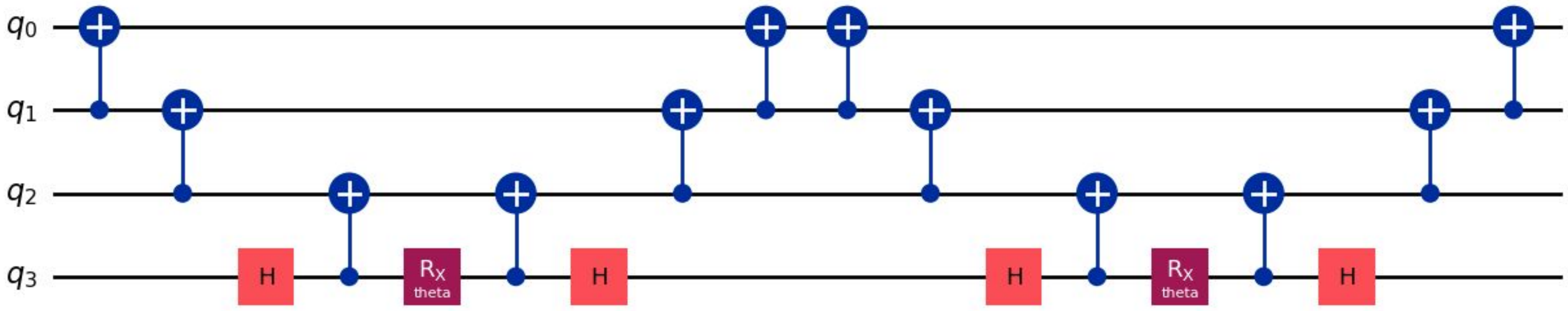
$$e^{i\theta XXXZ}$$



CNOT's optimization

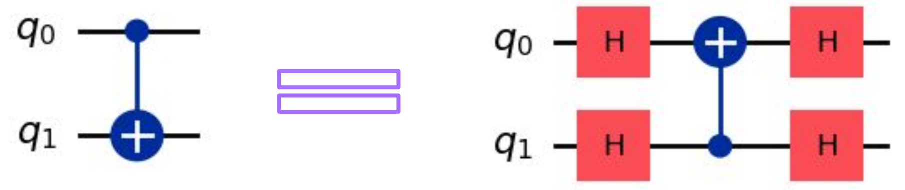
Inverse staircase algorithm

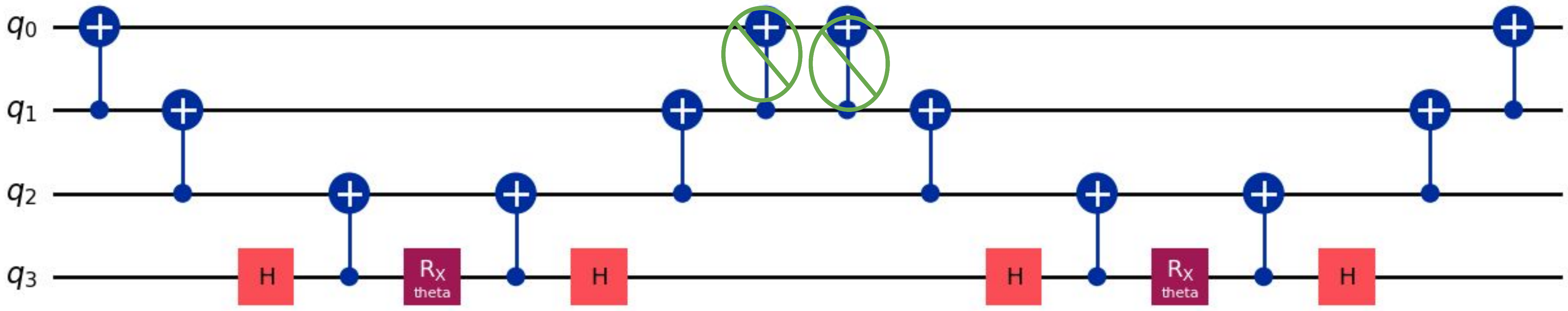




CNOT's optimization

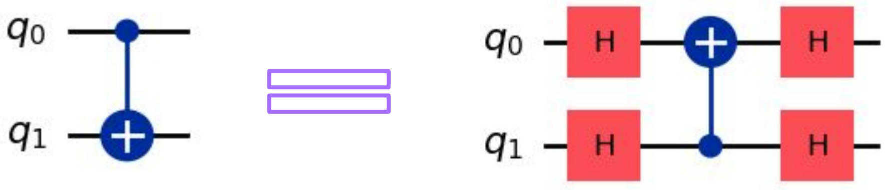
Inverse staircase algorithm

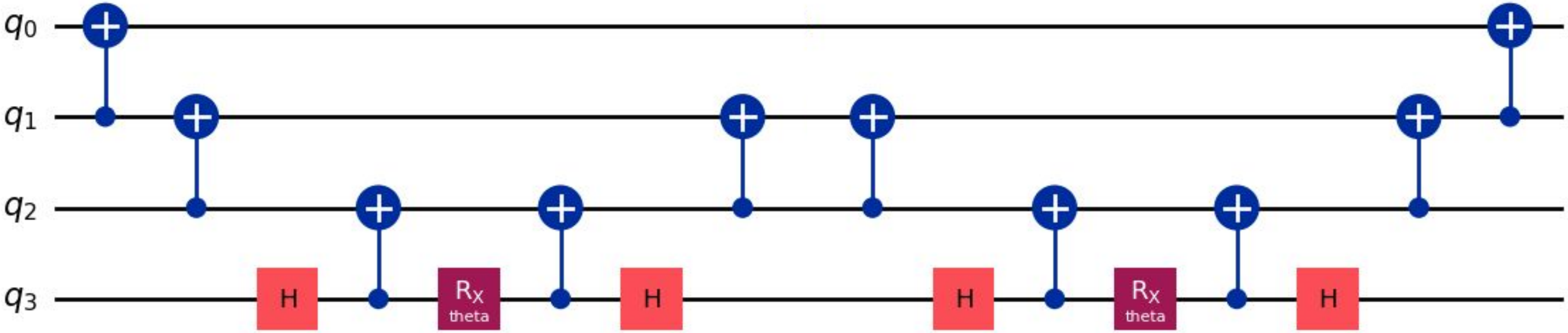




CNOT’s optimization

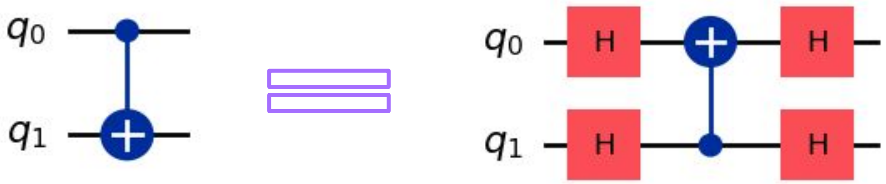
Inverse staircase algorithm

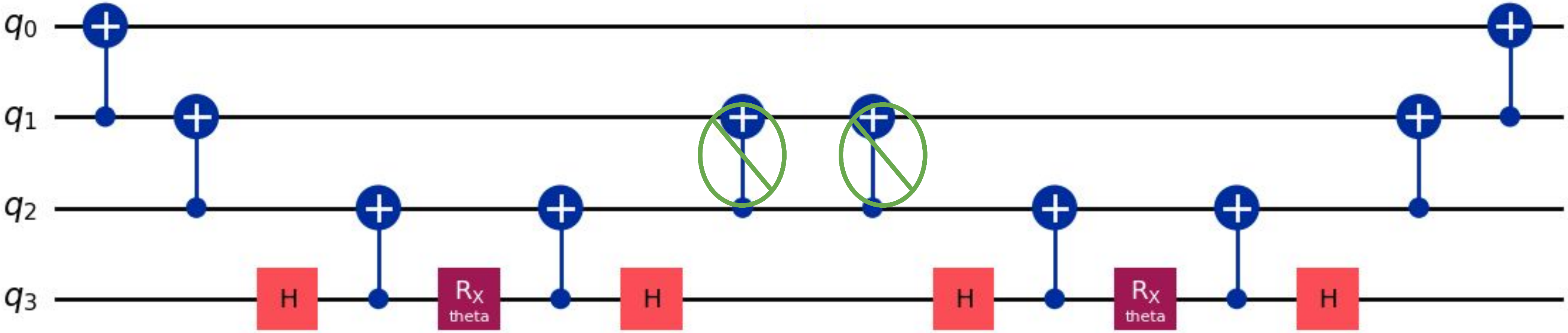




CNOT's optimization

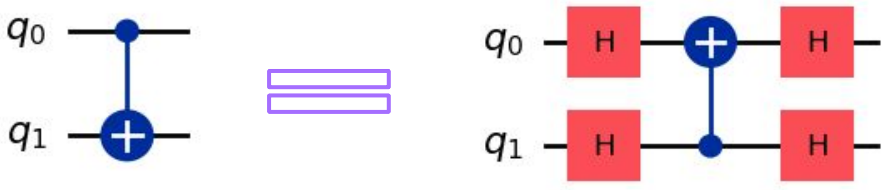
Inverse staircase algorithm

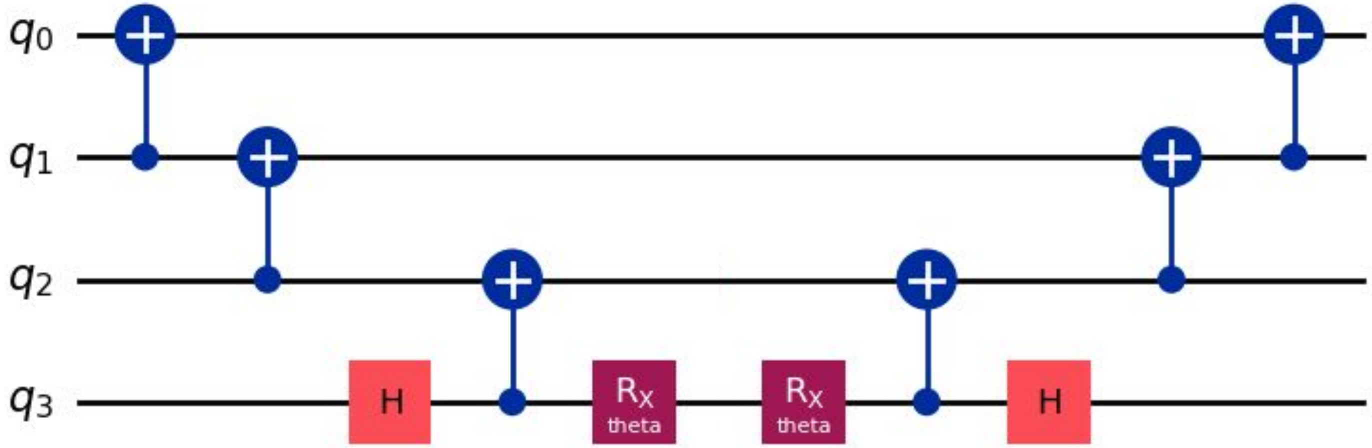




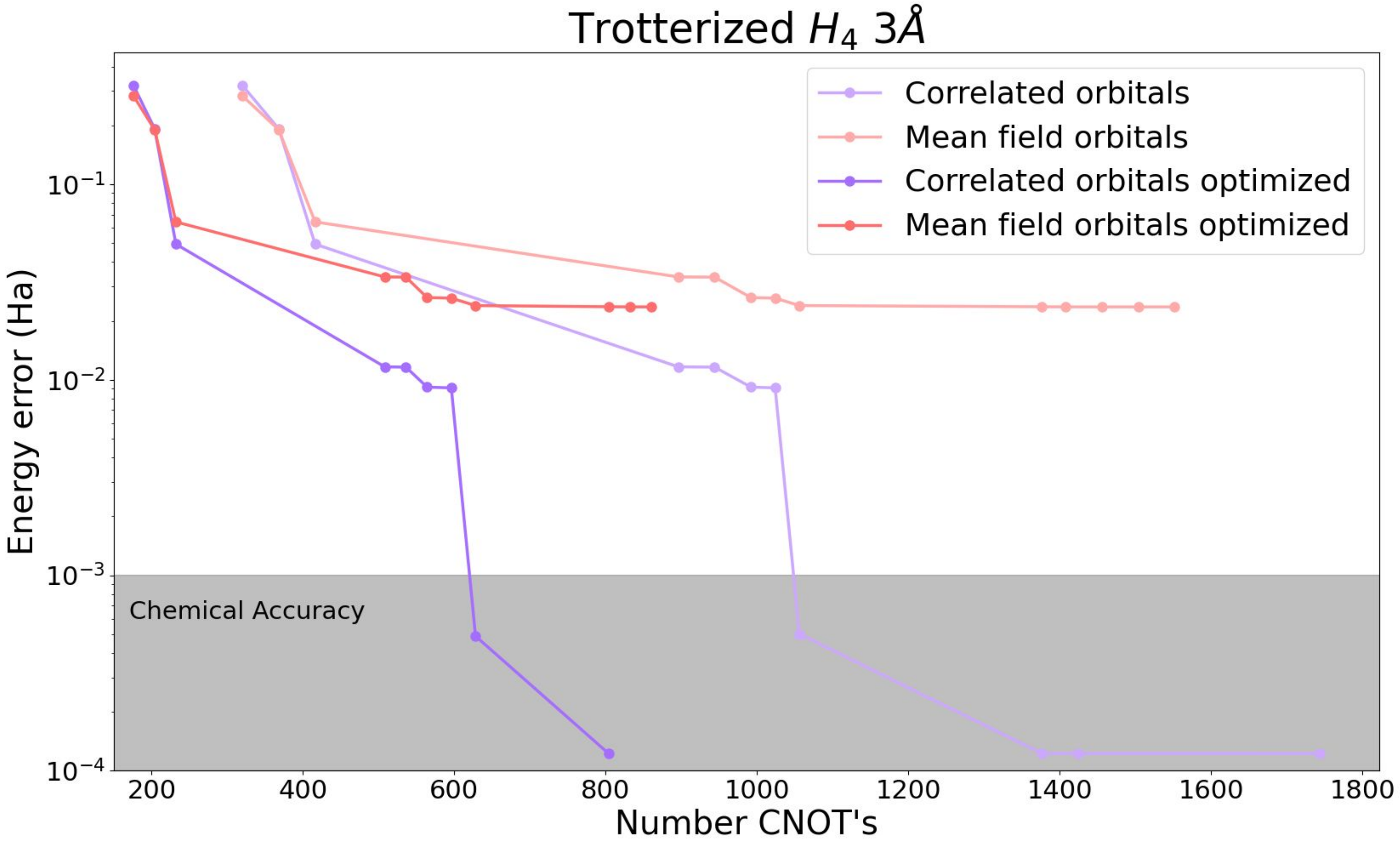
CNOT’s optimization

Inverse staircase algorithm





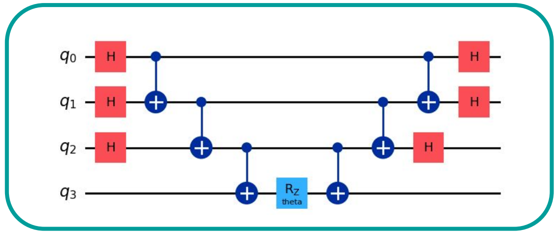
CNOT’s optimization



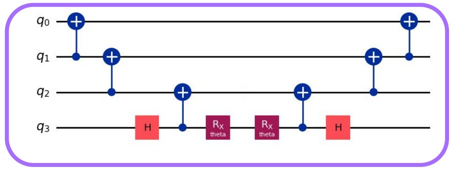
Transpiling

- FakeTorino()

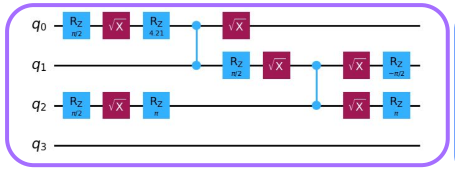
Initial circuit

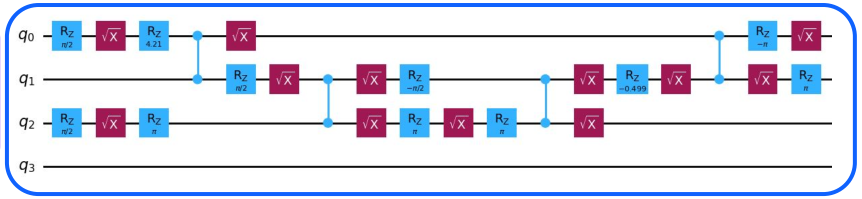


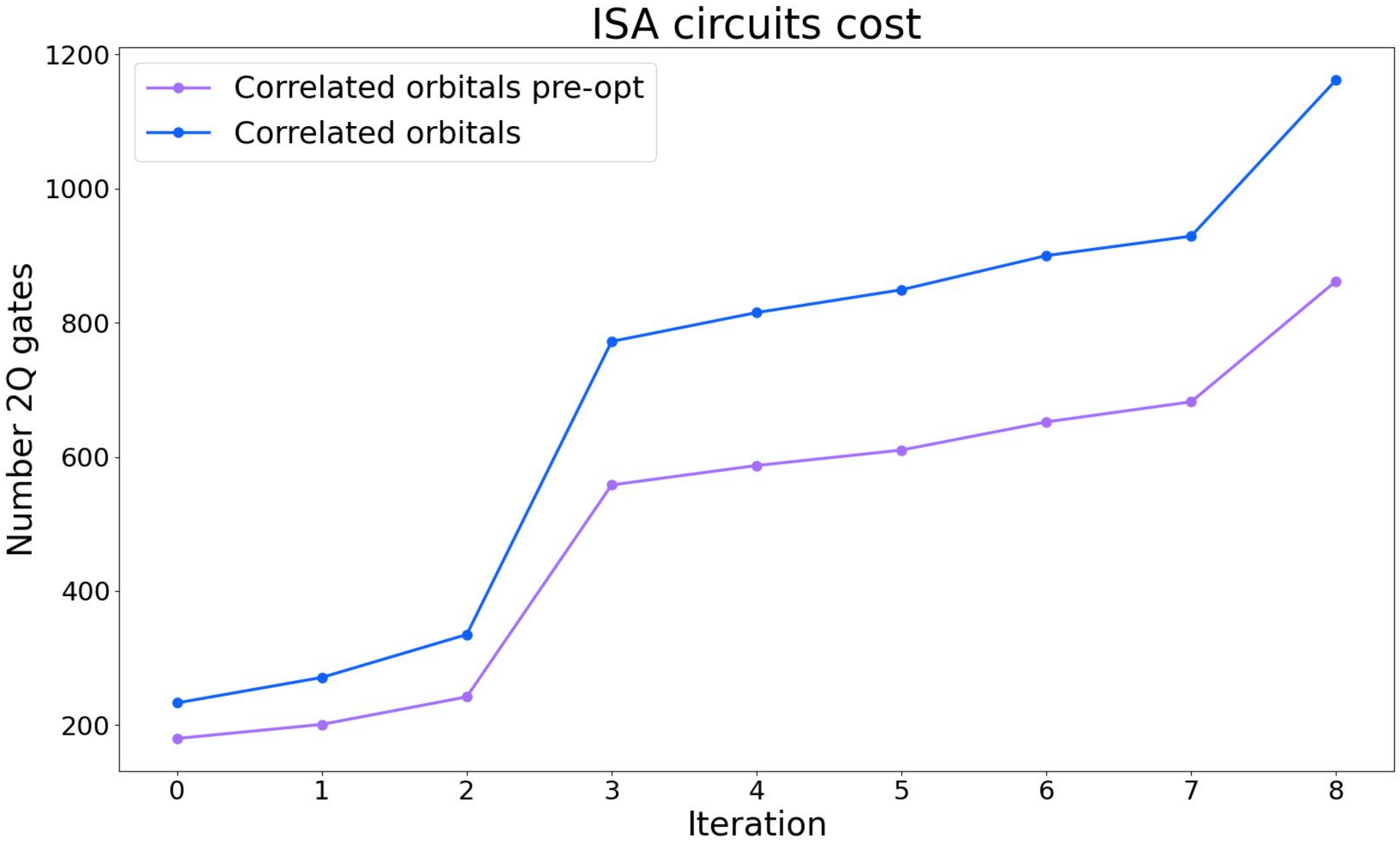
Pre-optimization



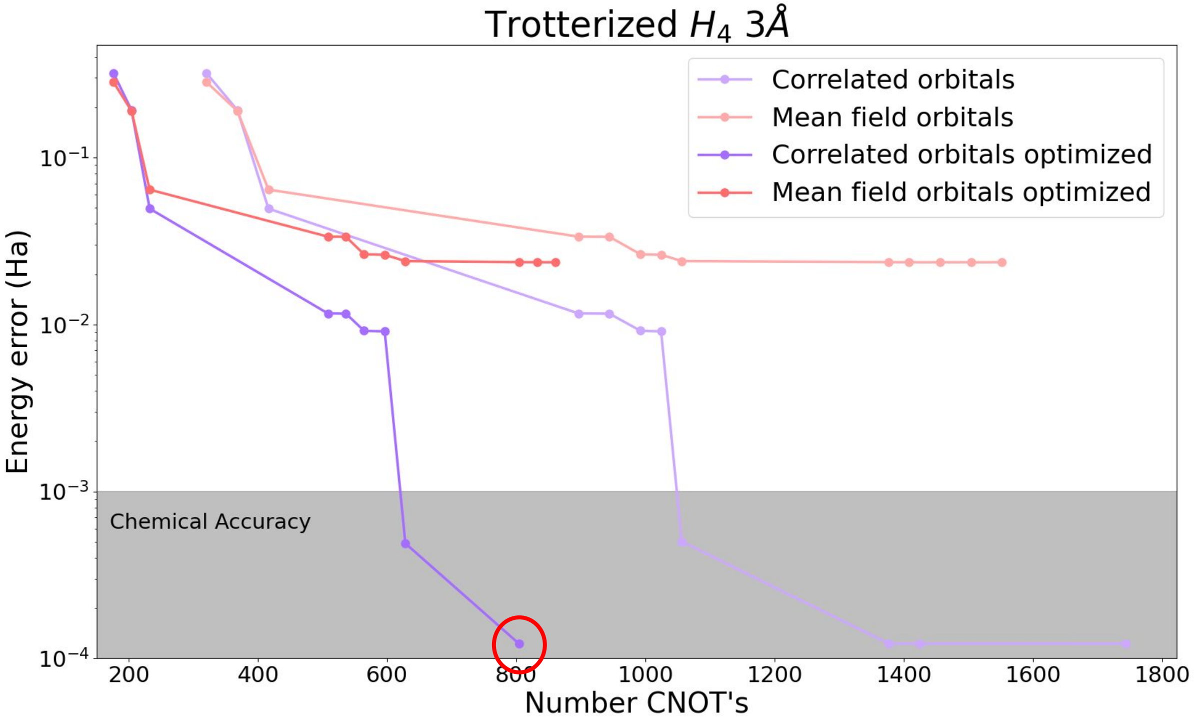
ISA circuit

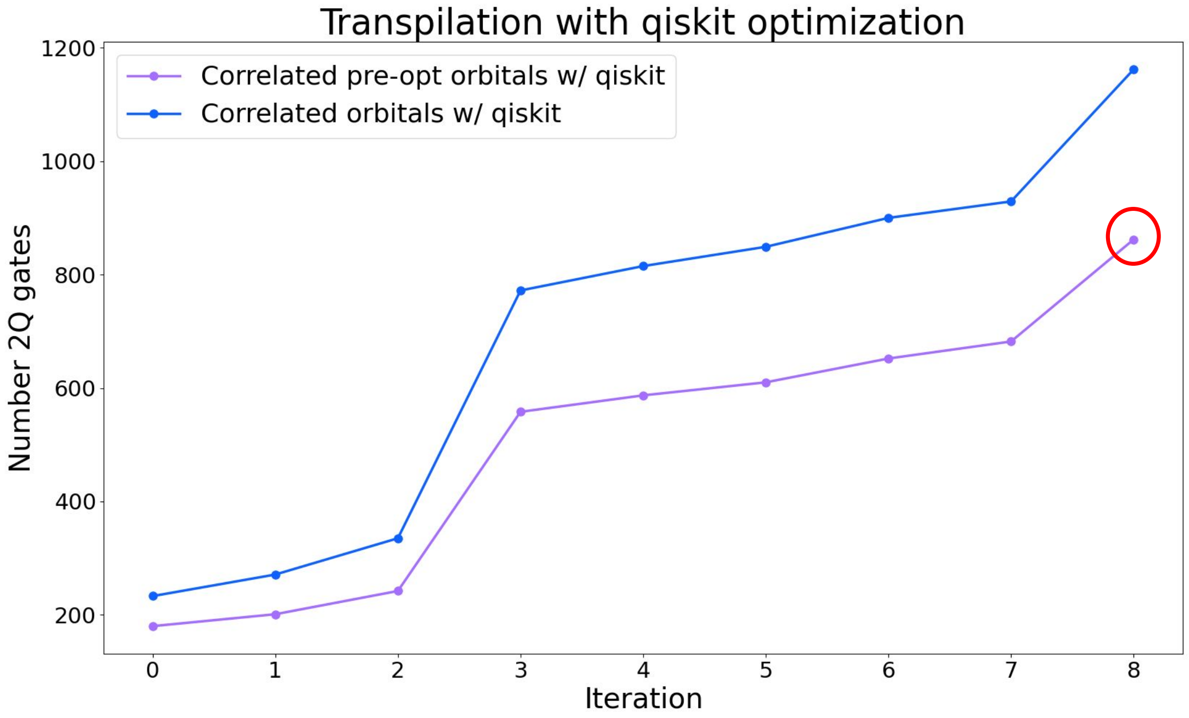






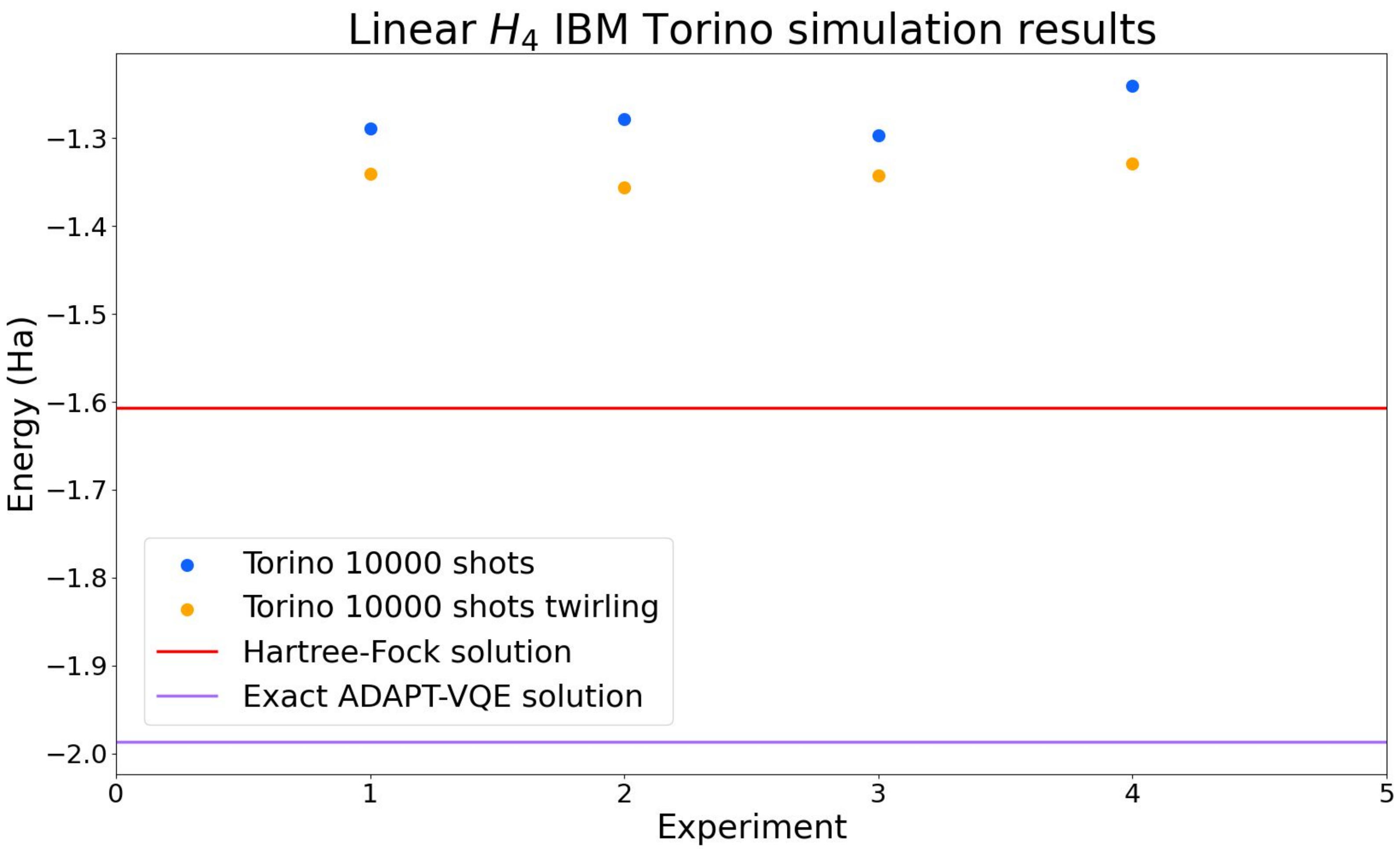
Real Hardware Simulation Result





Real Hardware Simulation Result

- IBM Torino



BasQ

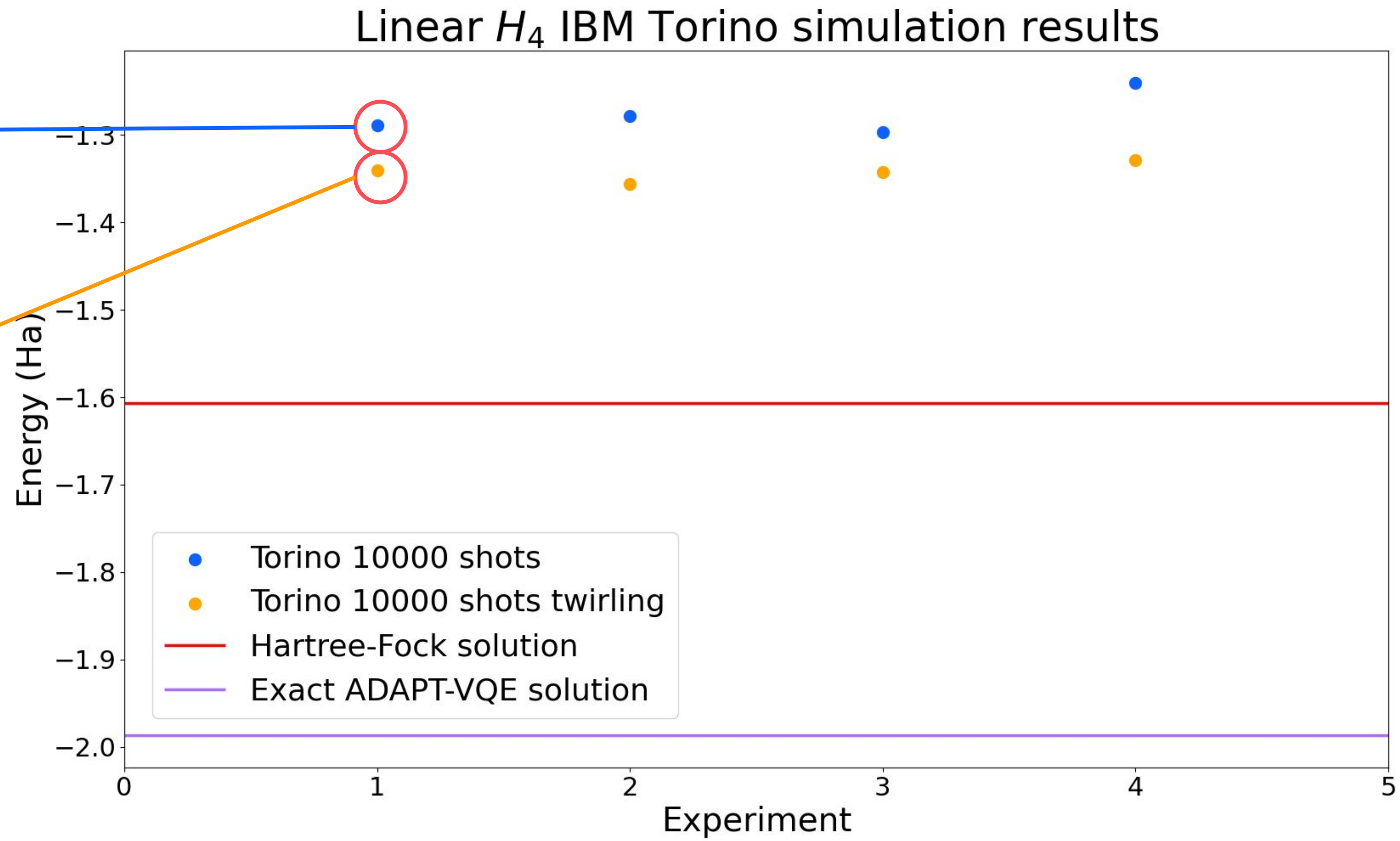
Basque Quantum

What error mitigation I have?

zne mitigation and measure mitigation

(resilience level 2)

zne mitigation, measure mitigation and twirling



What error mitigation I have?

Estimator Sampler

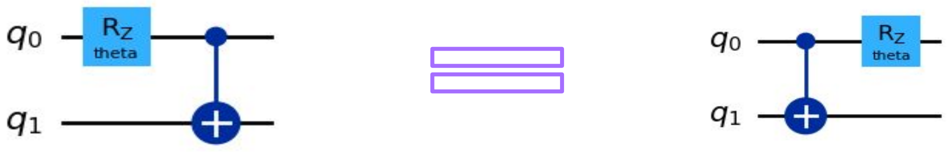
- Resilience: Advanced options for configuring error mitigation methods such as measurement error mitigation, ZNE, and PEC.
- Dynamical decoupling: Options for dynamical decoupling.
- Execution: Primitive execution options, including whether to initialize qubits and the repetition delay.
- Twirling: Twirling options, such as whether to apply two-qubit gate twirling and the number of shots to run for each random sample.
- Environment: Execution environment options, such as the logging level to set and job tags to add.
- Simulator: Simulator options, such as the basis gates, simulator seed, and coupling map. Applies to local testing mode only.

Take into account incompatibilities, such as PEC and ZNE

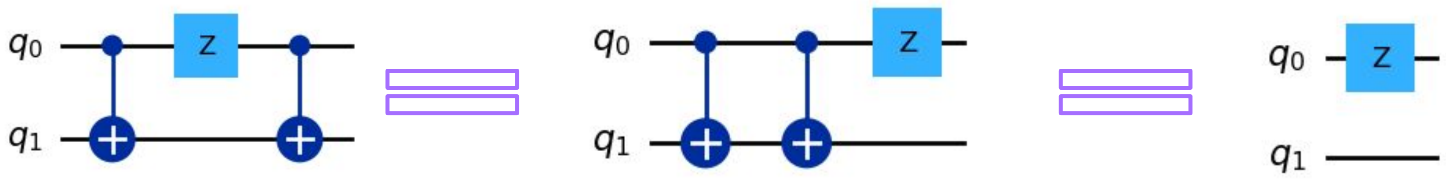
https://docs.quantum.ibm.com/guides/runtime-options-overview#options-compatibility-table

How does qiskit optimize?

- Using known equalities reduces the number of gates in the circuit
- Using known commutation relations



Example:



About the pass manager I used

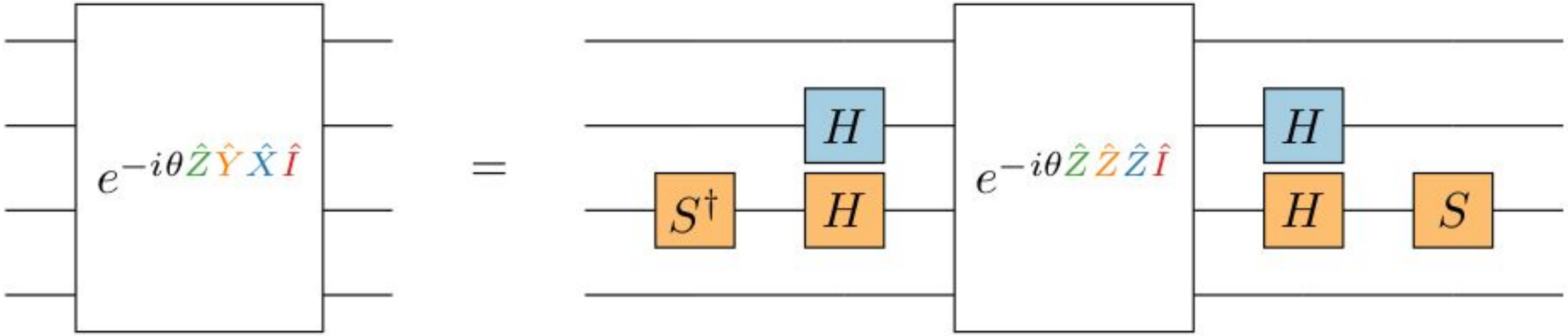
```
self.pm = generate_preset_pass_manager(backend=backend,
                                       optimization_level=3,
                                       initial_layout=layout,
                                       # layout_method='dense'
                                       )
```

Pass manager stages:

- Init stage: decomposition into 1 and 2 qubit gates
- Layout stage: mapping circuit qubits to qubits in the target
- Routing stage: fixes connectivity introducing SWAP when needed
- Translation stage: transforms operations into ones supported by the target
- Optimization stage: optimization, eliminates not needed gates
- Scheduling stage: scheduling the circuit to account for the timing of operations in the circuit

About trotter circuits implementation

$$e^{-i\theta \hat{I}\hat{X}\hat{Y}\hat{Z}}$$

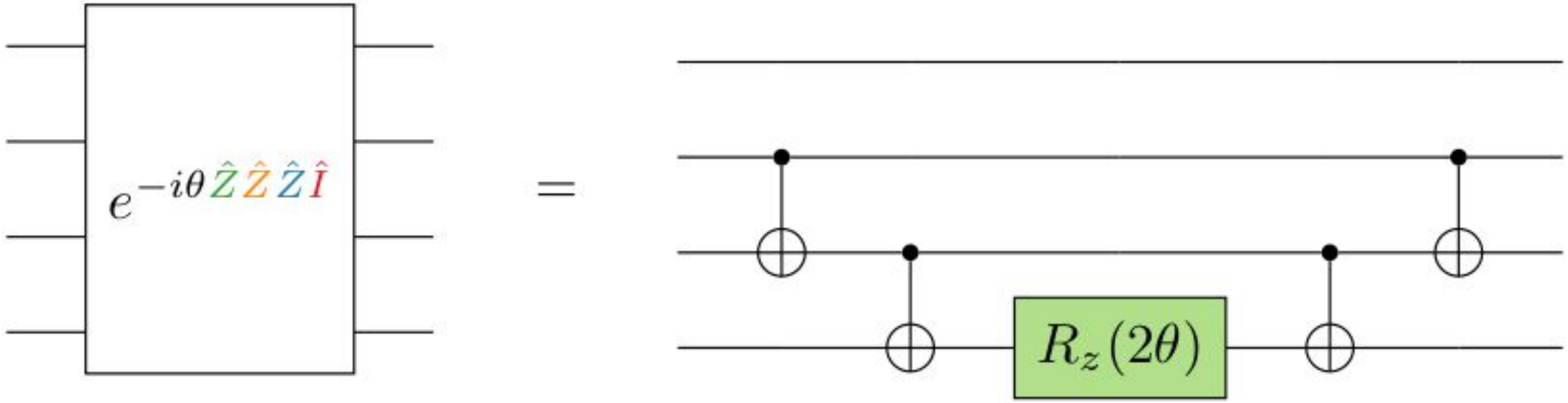


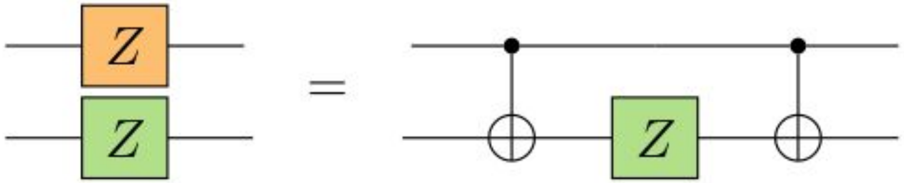
$$\hat{X} = \hat{H}\hat{Z}\hat{H}$$

$$\hat{Y} = \hat{S}\hat{X}\hat{S}^\dagger = \hat{S}\hat{H}\hat{Z}\hat{H}\hat{S}^\dagger$$

$$\hat{Y} = \hat{R}_x(-\pi/2)\hat{Z}\hat{R}_x(\pi/2)$$

About trotter circuits implementation





$$\hat{Z}\hat{Z} = \widehat{\mathrm{CNOT}}\,\hat{Z}\hat{I}\,\widehat{\mathrm{CNOT}}$$

CCSD ansatz

$$\left\langle \psi^{\text{CCSD}} \right\rangle = e^{\hat{T}_1+\hat{T}_2} \left| \psi^{\text{HF}} \right\rangle$$

$$\hat{T}_1 = \sum_{ia} \hat{t}_i^a = \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i$$

$$\hat{T}_2 = \sum_{i<j,a,b} \hat{t}_{ij}^{ab} = \sum_{i<j,a<b} t_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_i \hat{a}_j$$

Grimsley, H. R., Economou, S. E., Barnes, E., & Mayhall, N. J. (2019). An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, *10*(1). https://doi.org/10.1038/s41467-019-10988-2

UCCSD ansatz

$$|\Psi_{\text{UCCSD}}\rangle = e^{\hat{\mathcal{T}}_1+\hat{\mathcal{T}}_2}|0\rangle$$

$$\hat{\mathcal{T}}_1 = \sum_{ia} \theta_{ia}\left(a_a^\dagger a_i - a_i^\dagger a_a\right)$$

$$\hat{\mathcal{T}}_2 = \sum_{ijab} \theta_{ijab}\left(a_a^\dagger a_b^\dagger a_i a_j - a_j^\dagger a_i^\dagger a_b a_a\right)$$

Grimsley, H. R., Claudino, D., Economou, S. E., Barnes, E., & Mayhall, N. J. (2019). Is the Trotterized UCCSD Ansatz Chemically Well-Defined? *Journal of Chemical Theory and Computation*, *16*(1), 1–6. https://doi.org/10.1021/acs.jctc.9b01083

UCCSD ansatz trotterized

$$\hat{U}(\vec{\theta}) = \prod_{p>r} \exp\Big\{\theta_{pr}(\hat{c}_p^\dagger \hat{c}_r - \mathrm{H.\,c.})\Big\} \prod_{p>q>r>s} \exp\Big\{\theta_{pqrs}(\hat{c}_p^\dagger \hat{c}_q^\dagger \hat{c}_r \hat{c}_s - \mathrm{H.\,c.})\Big\}$$

https://docs.pennylane.ai/en/stable/code/api/pennylane.UCCSD.html

About VQE size

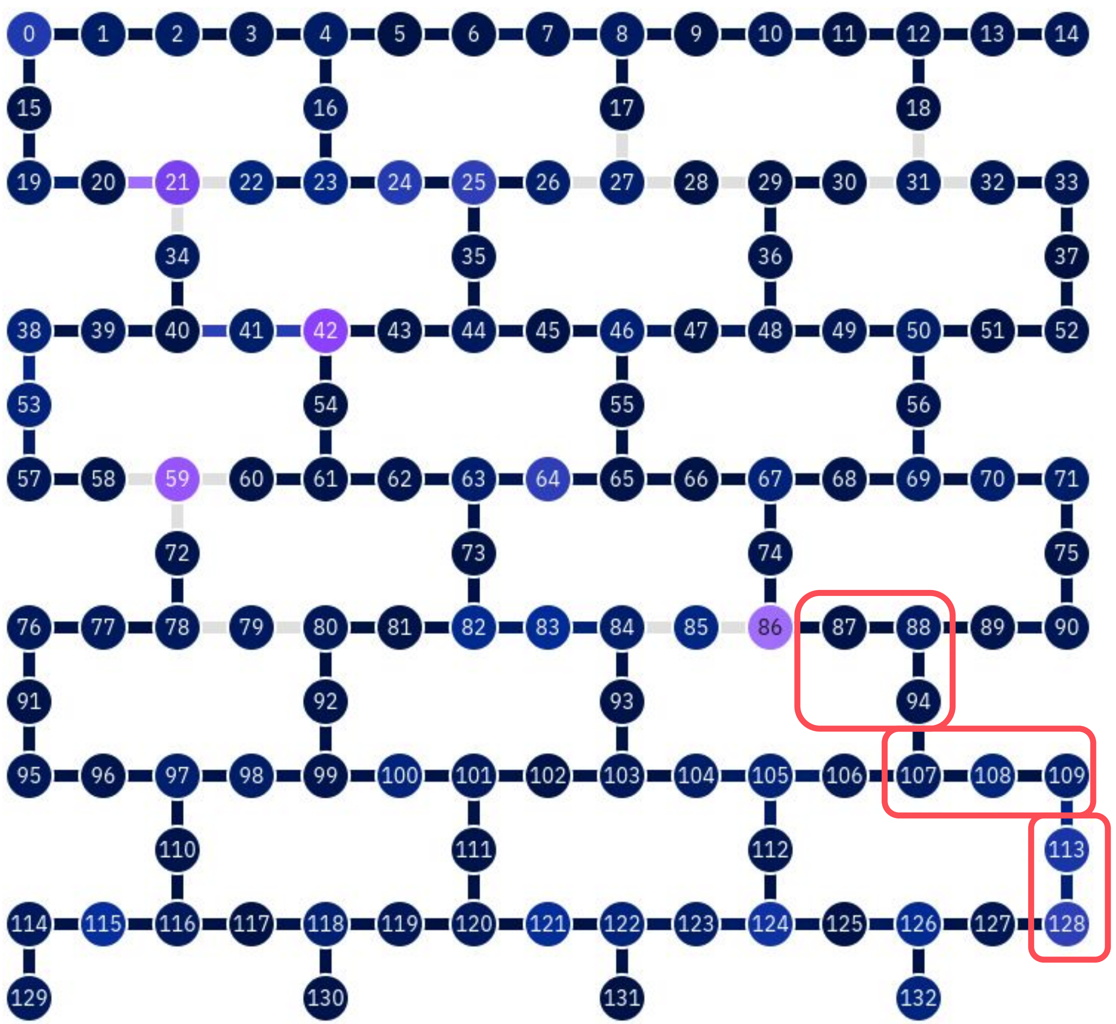
Orbitals	UCCSD operators
8	14
10	27
12	44
14	65
16	90

https://docs.pennylane.ai/en/stable/code/api/pennylane.UCCSD.html

About Torino

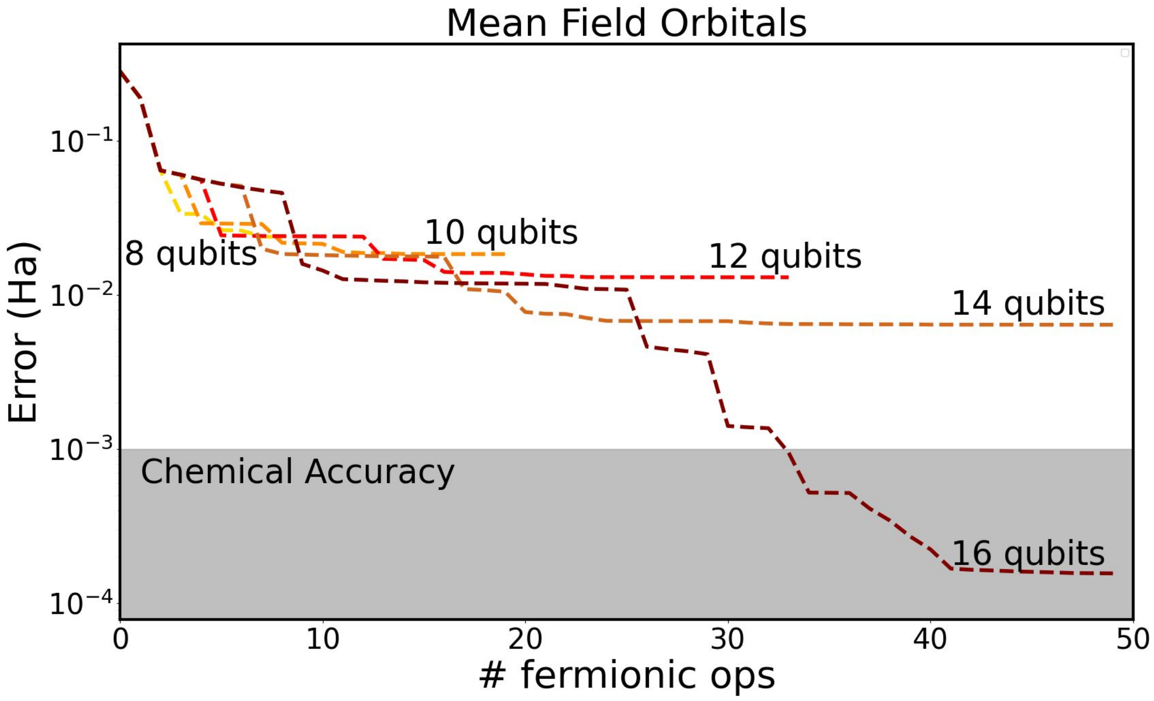
- Available gates

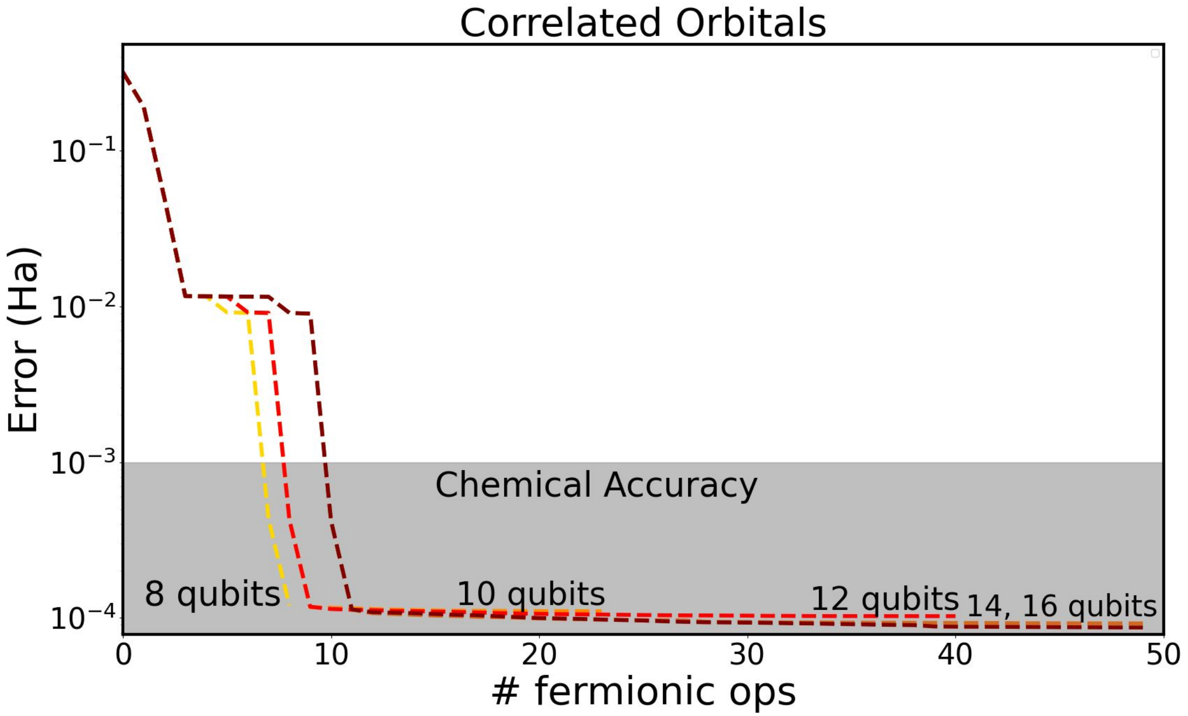
  CZ, ID, RX, RZ, RZZ, SX, X



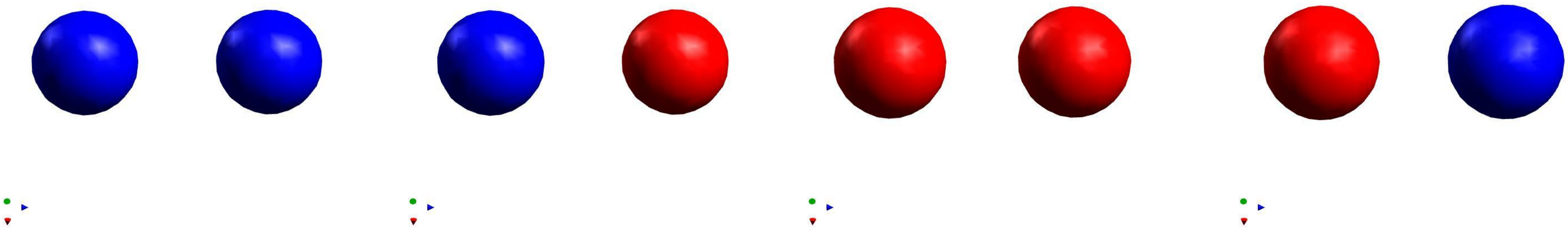
BasQ
Basque Quantum

About basis sets

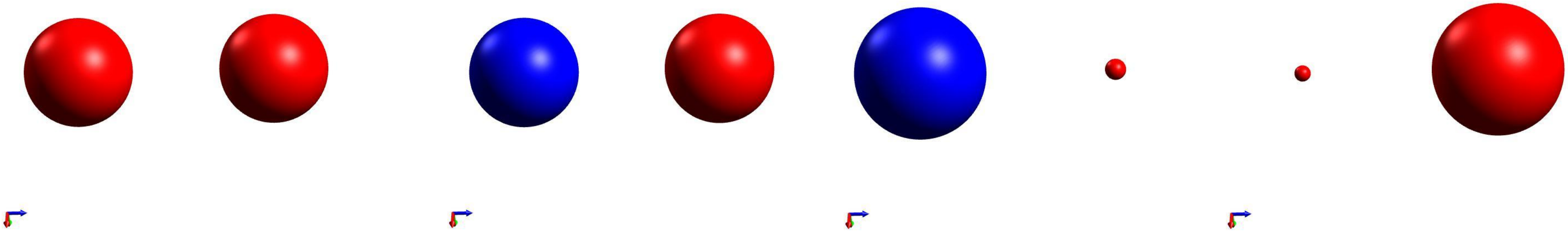




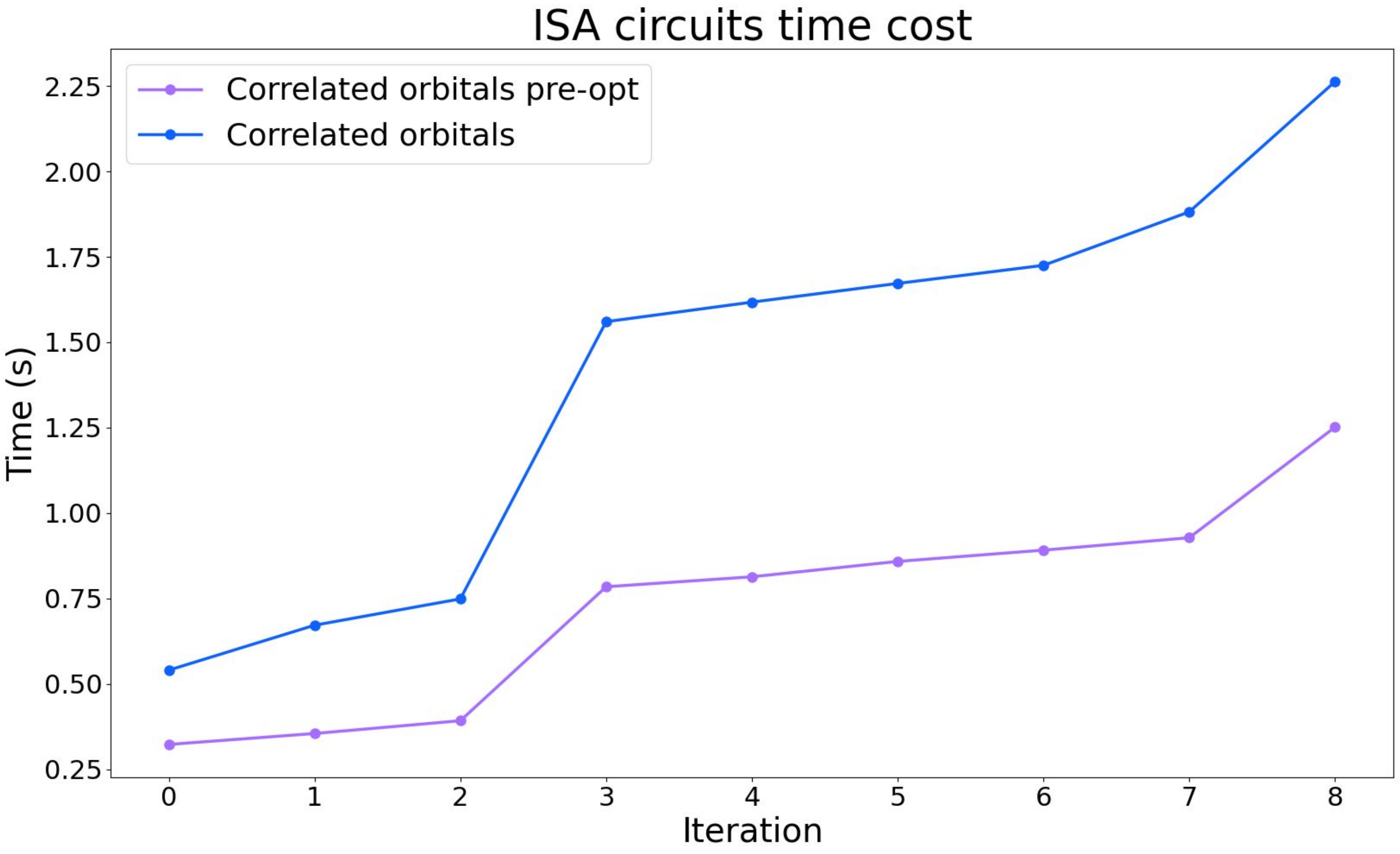
CO’s visualization



NO's visualization



Time transpilation



S gate

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$