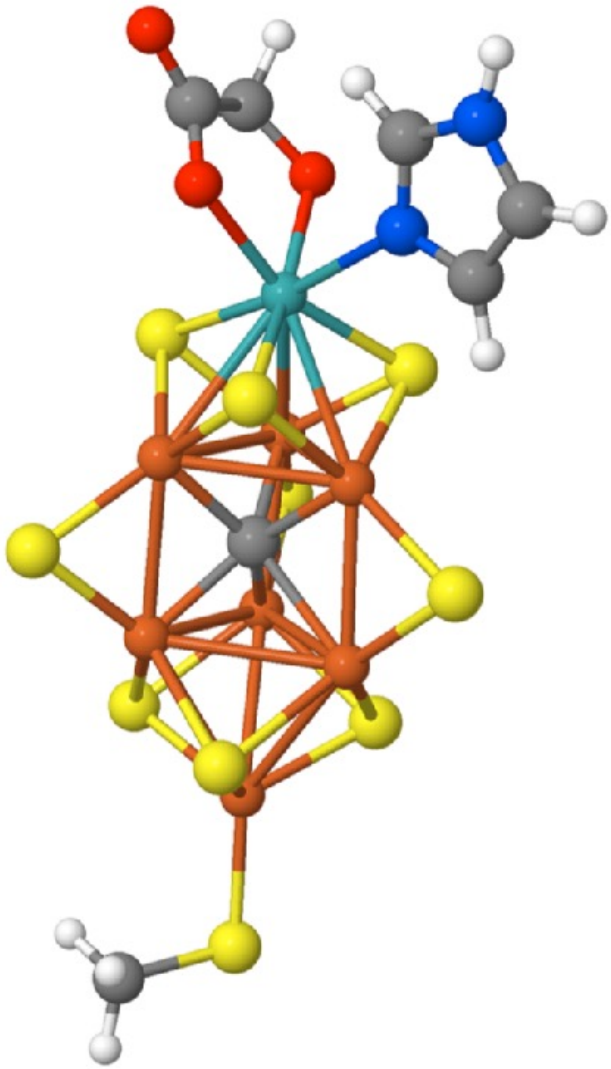


Lecture 11:
Variational Quantum Algorithms and Quantum Chemistry

COMP3366
Quantum algorithms & computing architecture

Instructor: Yuxiang Yang

Department of Computer Science, HKU

Objectives:

- **[O1] Concepts:** VQA, VQE, Ansatz, cost functions, quantum chemistry
- **[O3] Algorithm design:** Implementing VQE with Qiskit.

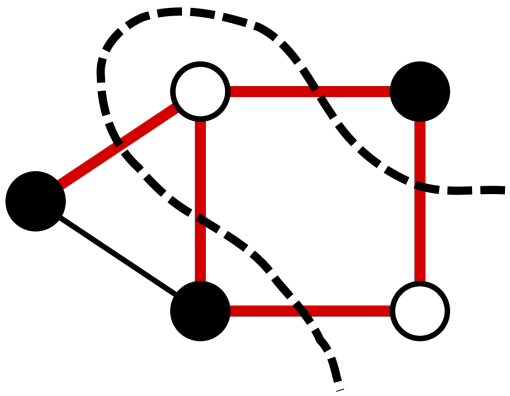
Previously in COMP3366: “QAOA”

Max-Cut problem

- Max-Cut:
  For a graph $G = (V, E)$, split the set of vertices V of a graph into 2 disjoint subsets, such that the number of edges connecting the 2 subsets is maximized.
- Maximum cut as an optimization problem:

$$\max_{s} \frac{1}{2} \sum_{(i,j)\in E} \left(1 - s_i s_j\right) \quad s_i \in \{1, -1\}$$

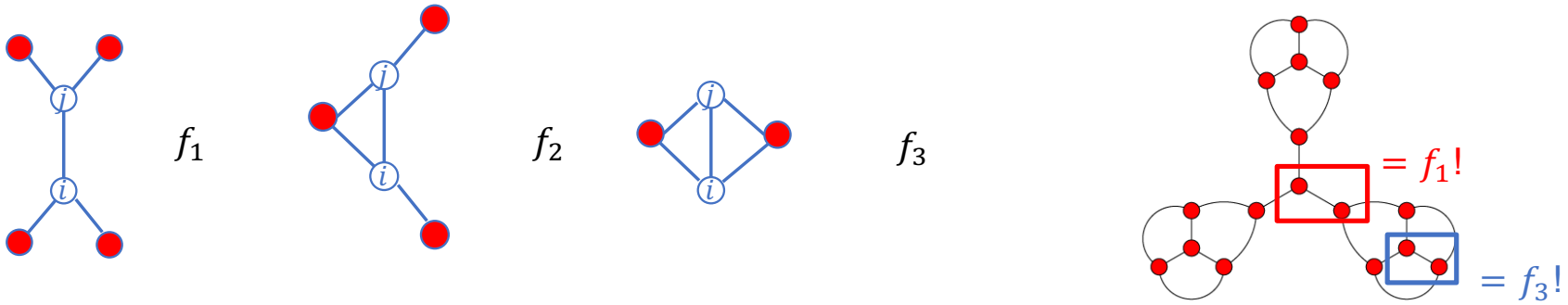
- Max-Cut is a prototypical NP-complete problem:
  Easy to verify a solution, but extremely hard to solve!



5 edges in the maximum cut

Quantum Approximate Optimization Algorithm

- QAOA is a classical-quantum variational algorithm for getting approximate solutions to the Max-Cut problem.
- For any 3-regular graph, a 6-qubit quantum computer is sufficient for executing a QAOA with depth $p = 1$ that yields a 0.6924-approximate solution. The depth of quantum circuits depends only on p and not on n!

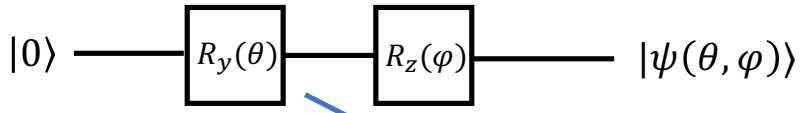


- QAOA may be extended to other optimization methods and be applied, for instance, to finance and engineering.
  It is a good proposal for demonstrating the power of near-term quantum computers.

Part I: Variational Quantum Algorithms

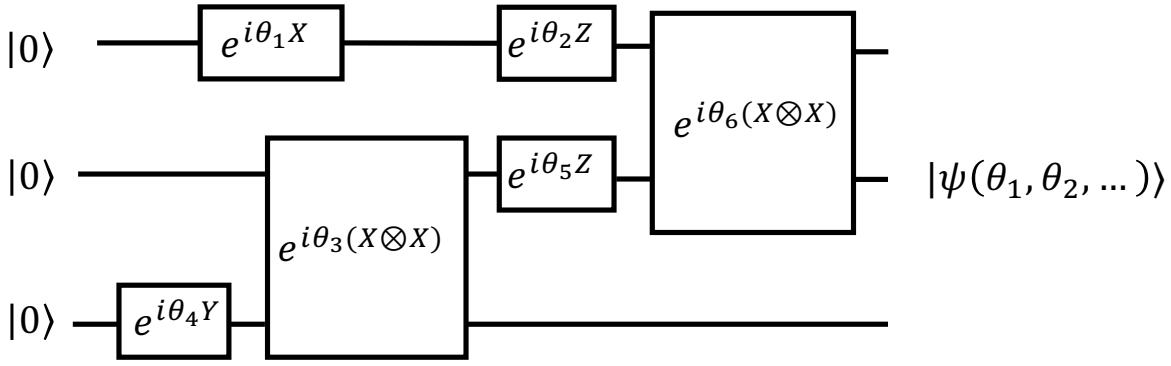
Variational states & circuits

- A single qubit state, generated by a variational circuit:



Gate types are fixed ($X/Y/Z$ rotation); parameters vary

- A general state: $|\psi(\theta)\rangle = U(\theta)|0\rangle^{\otimes n}$ where $\theta = (\theta_1, \theta_2, \dots)$ is a vector of variable parameters

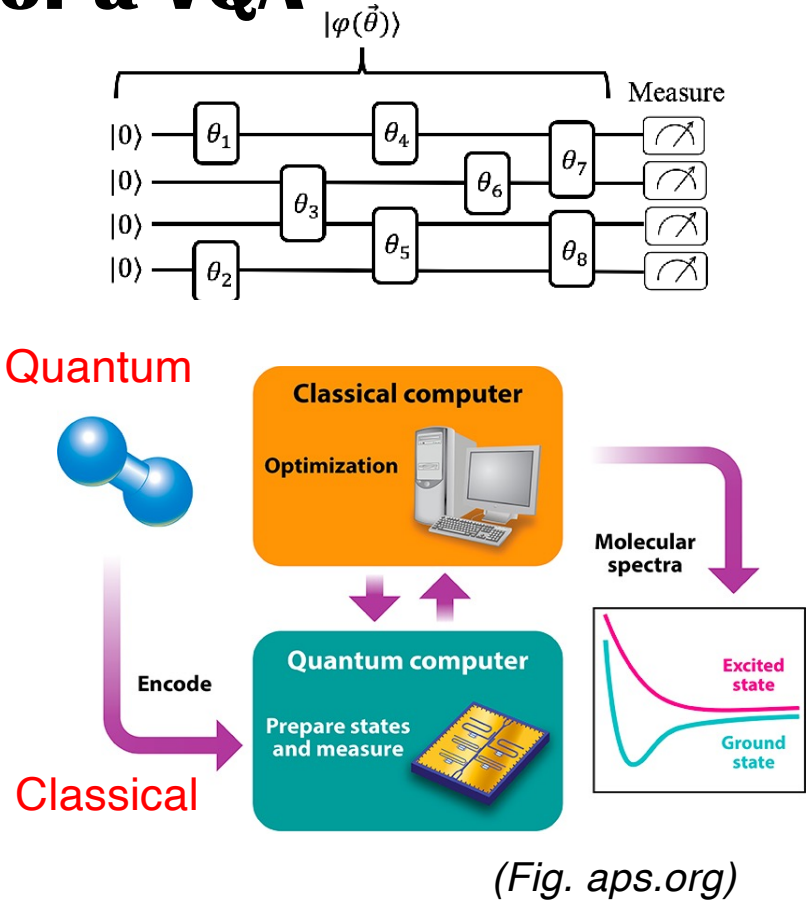


What is a VQA

- A VQA (Variational Quantum Algorithm) is an algorithm where a variational quantum circuit is trained step-by-step to minimize a cost function $C(\theta)$.
- VQAs are classical-quantum hybrid algorithms, where classical and quantum computers join force to tackle a hard problem.
- In the near-term future, the classical-quantum hybrid is the most promising structure of quantum computing!

Standard procedure of a VQA

1. Select an ansatz of the circuit and initiate it in a random configuration $\vec{\theta}_0$
2. Run the circuit in the current configuration $\vec{\theta}$.
3. Evaluate the cost function $C(\vec{\theta})$ with the output state of the circuit.
4. If $C(\vec{\theta}) \leq C_{\text{term}}$, terminate and output $\vec{\theta}$; otherwise, go to Step 5.
5. Repeat Steps 2-3 for multiple times to get the values of C in a neighborhood of $\vec{\theta}$. Compute the gradients $\frac{\partial C}{\partial \theta_1}, \frac{\partial C}{\partial \theta_2}, \ldots$
6. Update $\vec{\theta} \rightarrow \vec{\theta}'$ using gradient descent. Go to Step 2.



(Fig. aps.org)

Example: QAOA (Lecture 10)

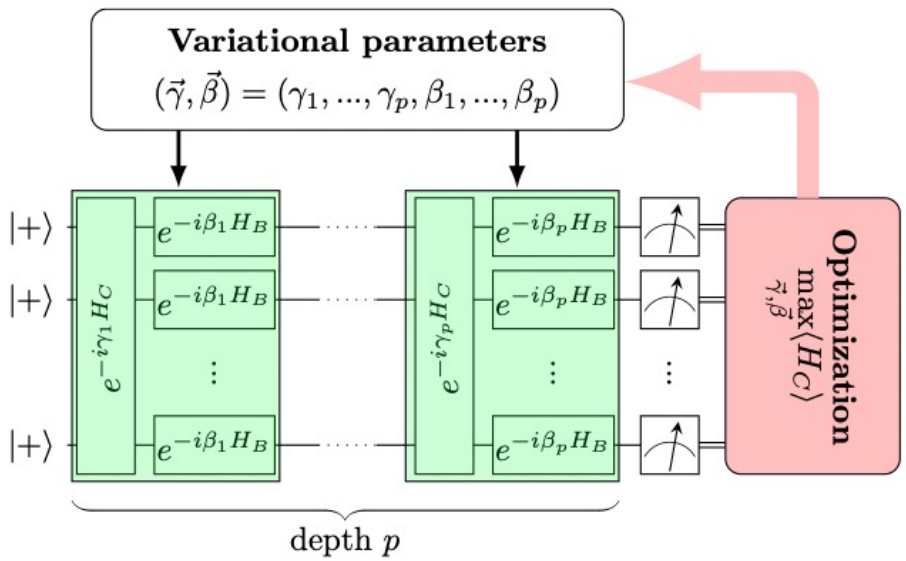
- QAOA is a VQA.
- Ansatz:

$$\left|\psi\left(\vec{\beta},\vec{\gamma}\right)\right\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \dots e^{-i\gamma_1 H_M}\ e^{-i\beta_1 H_C} |+\rangle^{\otimes n}$$

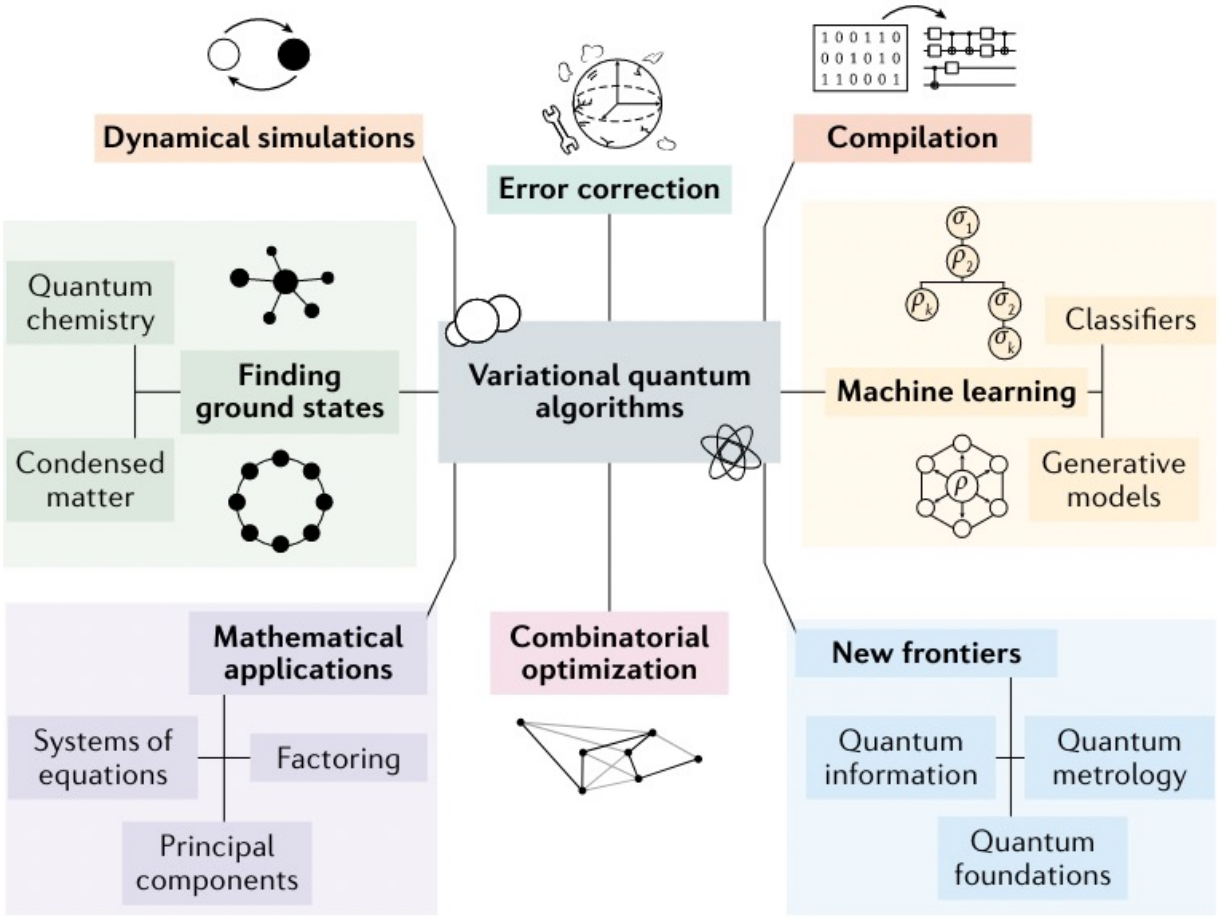
- Cost function:

$$C\left(\vec{\beta},\vec{\gamma}\right) = \left\langle\psi\left(\vec{\beta},\vec{\gamma}\right)\right| H_C \left|\psi\left(\vec{\beta},\vec{\gamma}\right)\right\rangle$$

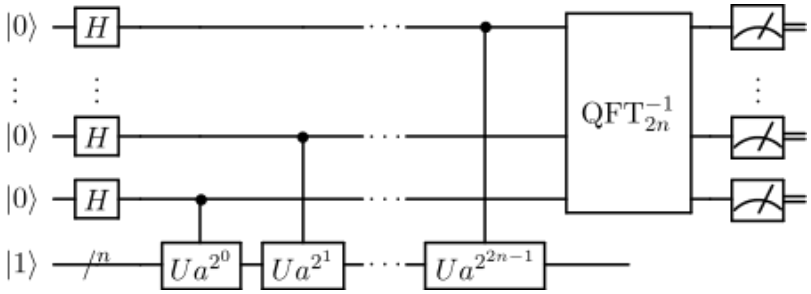
where H_C is the Hamiltonian
that encodes a Max-Cut problem.



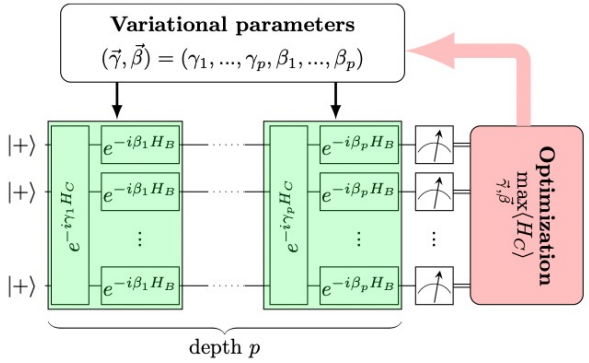
Applications of VQA



VQA vs Fully quantum algorithms



- “One-shot”.
- Fully coherent.
  (No conversion of quantum-classical information until the end.)
- Large scale and deep circuits.
- Usually no ancillary variables (except for the input).



- Iterative.
- Low depth.
  Circuit depth is brought down by introducing multiple iterations.
- Use classical computers as assistants.
- Involve ancillary variables.

There are 2 crucial ingredients of a VQA:
The cost function, and the ansatz!

Cost function

- A VQA is trained via step-by-step reducing a cost function $C(\theta)$.
- The cost function of a VQA is a function that maps the trainable parameters θ to real numbers:

$$C(\theta) = f(\{|\psi_k\rangle\}, \{O_k\}, U(\theta)) \in \mathbb{R}$$

- Here $\{|\psi_k\rangle\}$ are input states from a training set and $\{O_k\}$ is a set of observables.
- Example:

$$C(\theta) = \sum_k f_k \langle\psi_k|U(\theta)^\dagger O_k U(\theta)|\psi_k\rangle$$

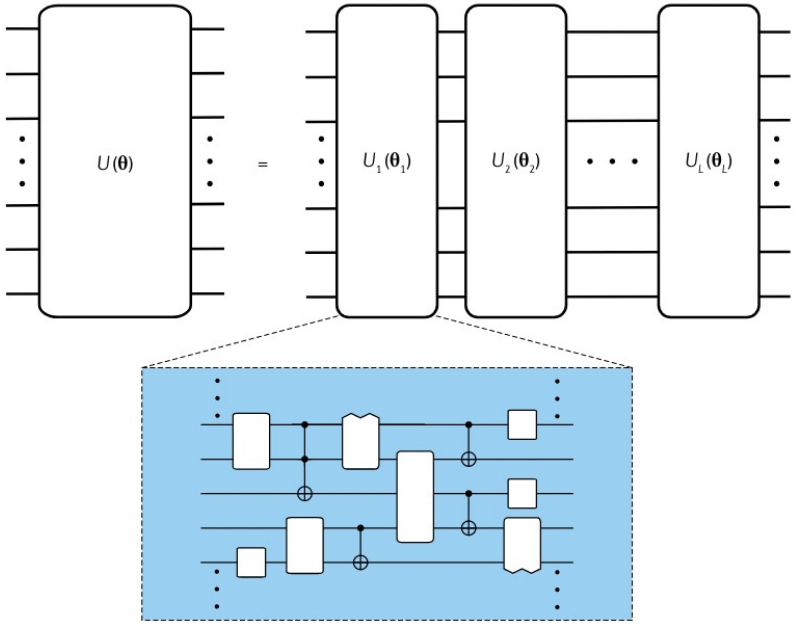
Criteria for a good cost

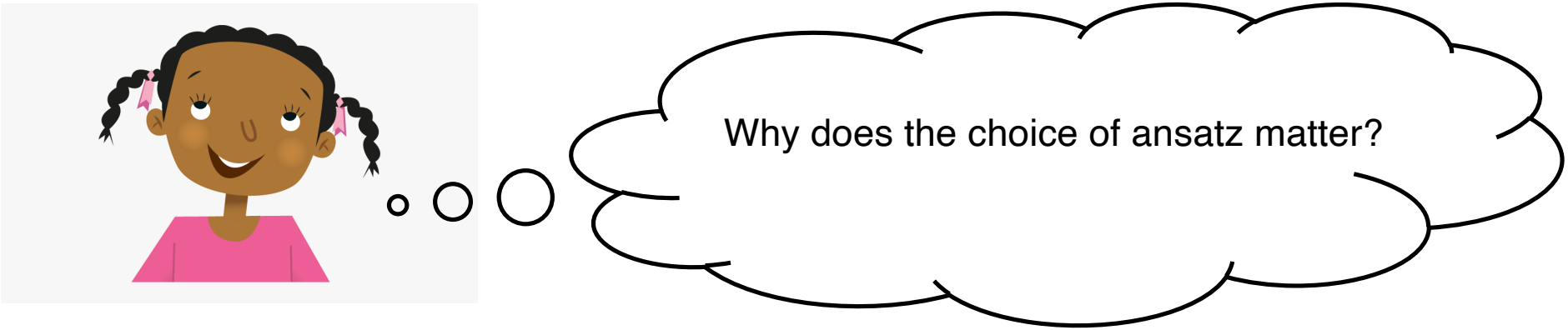
- Usually, a cost function $C(\theta)$ should satisfy the following criteria:

1. **Trainability:**
   It should be possible to efficiently optimize the parameters θ.
2. **Faithfulness:**
   The minimum of $C(\theta)$ should correspond to the solution of the problem.
3. **Measurability:**
   It should be efficiently estimable by performing measurements on a quantum computer and post-processing the classical data.

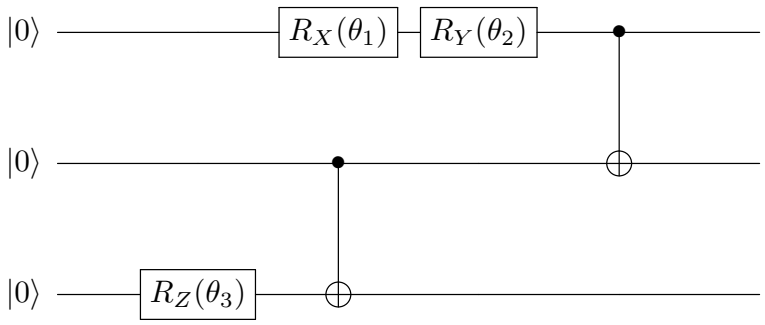
Ansatzes

- Usually, a variational circuit consists of a sequence of circuit blocks. Each block admits the same structure as specified by the ansatz.
- The choice of ansatzes is very important and may affect the accuracy of the solution as well as the efficiency of convergence.





Is this a good ansatz?

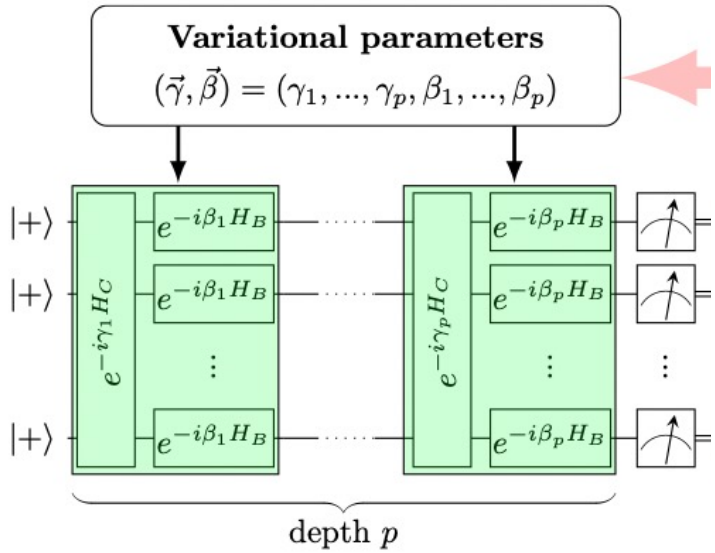


- No. For example, the last two qubits are wasted!
- Then, is there any trick or common-sense when choosing the ansatz?

- Two common types of good ansatz: problem-inspired and hardware-inspired.

Problem-inspired ansatz

- Why “problem”-inspired:
  The ansatz is designed based on information about the problem.
- Example: the quantum alternating operator ansatz in QAOA is a problem-inspired ansatz. It is based on the idea of simulated adiabatic quantum computing:

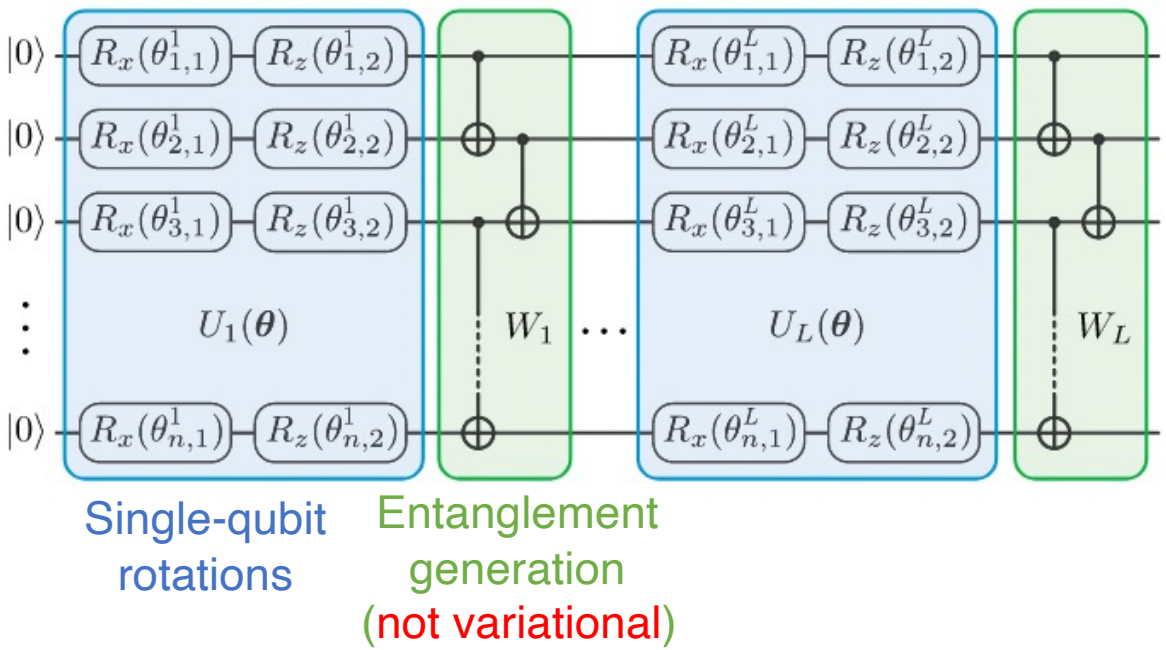


Hardware-efficient ansatz

- Problem-inspired ansatzes ensure fast convergence to a satisfying solution, but they could lead to a deep circuit or hard-to-implement gate sets.
- Besides, it might be hard to find such an inspiration for some problems.
- Alternatively, we can consider using hardware-efficient ansatzes.
- The choice of ansatz involves a trade-off:
  - Good and fast convergence to the solution → problem-inspired ansatzes
  - More implementable & “bullet-proof” → hardware-efficient ansatzes
- In the luckily case, you may find an ansatz with both merits.
  (There are fair and smart partners, but not many …)

Hardware-efficient ansatz

- A prototype of a hardware-efficient ansatz:



- Single-qubit rotations and CNOT are among the universal gate set and have high fidelity in most implementations.

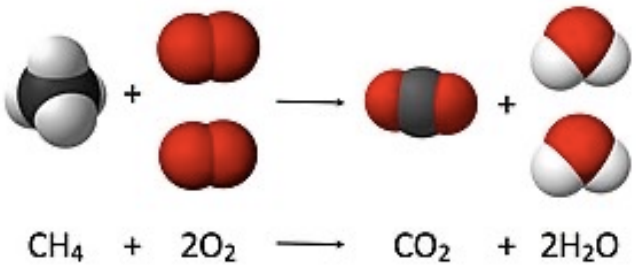
Part II:
Variational quantum eigensolvers

Hamiltonians in quantum chemistry

- Every quantum system, e.g., a molecule or an atom, has an associated Hamiltonian H that determines its feature via Schrödinger's equation:

$$\frac{i\partial|\psi\rangle}{\partial t} = H|\psi\rangle.$$

- In quantum chemistry, many fundamental tasks (e.g., calculating the reaction rate) boil down to finding the ground state associated with the Hamiltonian of a molecule.



The ground state problem

- In general, H is a Hermitian operator that can be diagonalized as

$$H = \sum_i E_i \, |\psi_i\rangle\langle\psi_i|$$

  with $E_0 \leq E_1 \leq \cdots$. Here $\{|\psi_i\rangle\}$ are the energy eigenstates.

- Given a complex H, it is one of the most important questions in modern physics to find the ground state $|\psi_0\rangle$ as well as the (minimum) ground energy E_0.

- Goal: We want to solve the ground state problem in quantum chemistry with a near-term quantum computer.
- Question: How to map “states of a molecule” to states of a quantum computer?

Hamiltonians in quantum chemistry

- Example: In quantum chemistry, we would like to know the ground state of a molecule, and the particles (electrons) are fermions.
- Hamiltonian: $H = \sum_j h_j a_j^\dagger a_j + \sum_{ij} h_{ij} a_j^\dagger a_i^\dagger a_i a_j$
- Here $a_j, a_j^\dagger$ are the annihilation and creation operators of the j-th orbital.

- Fermions are indistinguishable whereas qubits are distinguishable. They obey Pauli exclusion principle:

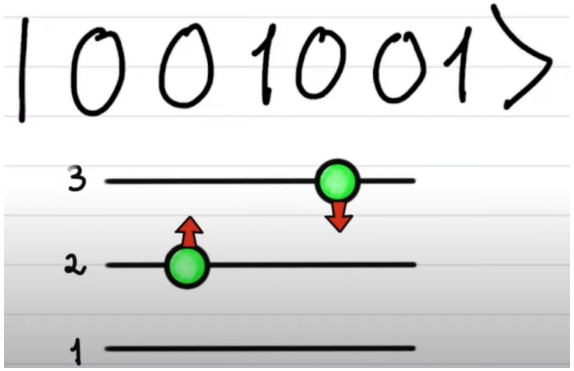
$$\{a_i, a_j\} = 0, \{a_i^\dagger, a_j^\dagger\} = 0, \{a_i, a_j^\dagger\} = \delta_{i,j}$$

- To simulate Fermionic systems on a quantum computer, we need to construct "simulated" annihilation and creation operators satisfying the above relation.

From Fermions to qubits

- Mapping occupations to qubit states:
  Spin orbital → qubits;
  $|0\rangle, |1\rangle$ → unoccupied, occupied
- (Jordan-Wigner) Operators of qubit systems

$$\hat{a}_j^\dagger := \left(\prod_{m<j} Z_m\right)\sigma_{+,j}$$



2 electrons, 3 orbitals → 6 qubits

where $\sigma_{+,j} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}_j = \frac{1}{2}(X_j + iY_j)$ obey Pauli exclusion principle: $\{\hat{a}_i, \hat{a}_j\} = 0, \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0, \{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{i,j}$.

- Conclusion: We can map a fermionic Hamiltonian to a Hamiltonian of multiple qubits by replacing every $a_j^\dagger$ (a_j) by $\hat{a}_j^\dagger$ $(\hat{a}_j)$.
- The number of qubits = the number of spin orbitals > the number of fermions.

Ground state via optimization

- Rayleigh-Ritz quotient formulation of the ground state energy:

$$E_0 = \min_{|\psi\rangle} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

- We can determine the ground state by solving the above optimization with H being the system's Hamiltonian.
- Idea:
  - Use a variational circuit to produce trial states $|\psi(\theta)\rangle$ with θ being variational parameter(s).
  - Measure the Hamiltonian for $|\psi(\theta)\rangle$ as the cost function $C_{VQE}(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$.
  - Update θ to make the cost function smaller.

Challenges

- We need to verify the following for $C_{VQE}(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$:

1. **Trainability:**
   It should be possible to efficiently optimize the parameters θ.
2. **Faithfulness:**
   If $|\psi_0\rangle$ is contained in the family $\{|\psi(\theta)\rangle\}_\theta$, then the faithfulness follows immediately from the definition. ✅
3. **Measurability:**
   This is a highly non-trivial problem! ⚠️
   It may not be possible to efficiently measure a generic H.

Measurement of energy

- Consider Hamiltonians of the following form:

$$H = \sum_{i,\alpha} h_\alpha^i \sigma_\alpha^i + \sum_{i,j,\alpha,\beta} h_{\alpha\beta}^{ij} \sigma_\alpha^i \otimes \sigma_\beta^j + \cdots$$

  where σ_α^i denotes a Pauli operator σ_α ($\sigma_{0,1,2,3} = I, X, Y, Z$) on the ith qubit and h_α^i ... are coefficients.

- By linearity, we have $\langle H \rangle = \sum_{i,\alpha} h_\alpha^i \langle \sigma_\alpha^i \rangle + \sum_{i,j,\alpha,\beta} h_{\alpha\beta}^{ij} \left\langle \sigma_\alpha^i \otimes \sigma_\beta^j \right\rangle + \cdots$

- Pauli operators can be efficiently estimated (by measurement in the bases $\{|0/1\rangle\}, \{|\pm\rangle\}, \{|\pm i\rangle\}$) ⇒ each $\langle \sigma_\alpha^i \rangle$ can be efficiently measured.

Energy measurability

- Any Hamiltonian can be expressed as

$$H = \sum_{i=1}^{m} c_i H_i$$

  with each H_i being tensor product of Paulis (i.e., a Pauli string) and c_i being real coefficients.
- To measure $\langle H \rangle$, we can measure each $\langle H_i \rangle$ separately and evaluate $\langle H \rangle = \sum_i c_i \langle H_i \rangle$ on a classical computer.
- Complexity grows as $O(m)$: the state preparation must be repeated for several times to measure $\langle H_i \rangle$, for $i = 1, \ldots, m$.
- Conclusion: $\langle H \rangle$ can be measured efficiently if H can be written as the sum of "not too many" Pauli strings.

Procedure of a VQE

- VQE (i.e., the Variational Quantum Eigensolver) is a VQA that is tailor-made for the ground state problem.
- Input: $H = \sum_{i=1}^{m} c_i H_i$, each H_i being tensor product of single-qubit Paulis. Output: ground state of H and the ground energy (up to a small error).
- VQE routine:
  1. **For $i = 1, 2, \ldots, m$, do:**
     State preparation:
     Prepare a parametrized quantum state $|\psi(\theta)\rangle$ via a variational circuit $U(\theta)|0\rangle^{\otimes n}$.
     Energy estimation: Measure H_i for $|\psi(\theta)\rangle$.
  2. **Classical feedback**:
     Evaluate $\langle H \rangle = \sum_i c_i \langle H_i \rangle$ and update θ using a classical optimization method.
  3. **Repeat** the above procedure until a convergence criteria is satisfied.

Example: finding the ground state of H_2

- Step 1: Setting up the equivalent Hamiltonian
- Configuration of the molecule (atom types, nuclei distance …)

```
from pennylane import numpy as np

symbols = ["H", "H"]
coordinates = np.array([0.0, 0.0, -0.6614, 0.0, 0.0, 0.6614])
```

- Set the number of qubits (= 4)

  A built-in function does the Jordan-Wigner transform for us, outputting a 4-qubit Hamiltonian that is equivalent to the original Fermionic Hamiltonian.

```
import pennylane as qml

H, qubits = qml.qchem.molecular_hamiltonian(symbols, coordinates)
print("Number of qubits = ", qubits)
print("The Hamiltonian is ", H)
```

Out:

```
Number of qubits =  4
The Hamiltonian is    (-0.2427450126096431) [Z2]
+ (-0.2427450126096431) [Z3]
+ (-0.04207255194723672) [I0]
+ (0.1777135822907777) [Z1]
+ (0.17771358229077772) [Z0]
+ (0.12293330449306411) [Z0 Z2]
+ (0.12293330449306411) [Z1 Z3]
+ (0.16768338855610976) [Z0 Z3]
+ (0.16768338855610976) [Z1 Z2]
+ (0.17059759276839878) [Z0 Z1]
+ (0.17627661394198885) [Z2 Z3]
+ (-0.04475008406304564) [Y0 Y1 X2 X3]
+ (-0.04475008406304564) [X0 X1 Y2 Y3]
+ (0.04475008406304564) [Y0 X1 X2 Y3]
+ (0.04475008406304564) [X0 Y1 Y2 X3]
```

- Step 2: Initialization of the variational circuit.
- We want to prepare a simple variational state of the form

$$|\Psi(\theta)\rangle = \cos\frac{\theta}{2}|1100\rangle - \sin\frac{\theta}{2}|0011\rangle$$

- There is only one variable θ.

```
dev = qml.device('default.qubit', wires=4)

@qml.qnode(dev)
def circuit(phi):
    qml.PauliX(wires=0)
    qml.PauliX(wires=1)
    qml.DoubleExcitation(phi, wires=[0, 1, 2, 3])
    return qml.state()

circuit(0.1)
```

- This is realized by a simple circuit:

- Set $\theta = 0$ as the initial configuration.

- Step 3: Update θ to minimize the energy.

```python
# store the values of the cost function
energy = [cost_fn(theta)]

# store the values of the circuit parameter
angle = [theta]

max_iterations = 100
conv_tol = 1e-06

for n in range(max_iterations):
    theta, prev_energy = opt.step_and_cost(cost_fn, theta)

    energy.append(cost_fn(theta))
    angle.append(theta)

    conv = np.abs(energy[-1] - prev_energy)

    if n % 2 == 0:
        print(f"Step = {n},  Energy = {energy[-1]:.8f} Ha")

    if conv <= conv_tol:
        break

print("\n" f"Final value of the ground-state energy = {energy[-1]:.8f} Ha")
print("\n" f"Optimal value of the circuit parameter = {angle[-1]:.4f}")
```

```
Step = 0,  Energy = -1.12799983 Ha
Step = 2,  Energy = -1.13466246 Ha
Step = 4,  Energy = -1.13590595 Ha
Step = 6,  Energy = -1.13613667 Ha
Step = 8,  Energy = -1.13617944 Ha
Step = 10,  Energy = -1.13618736 Ha
Step = 12,  Energy = -1.13618883 Ha
```

We can see that the energy goes down step-by-step.

- Step 4: Output the ground state & energy.

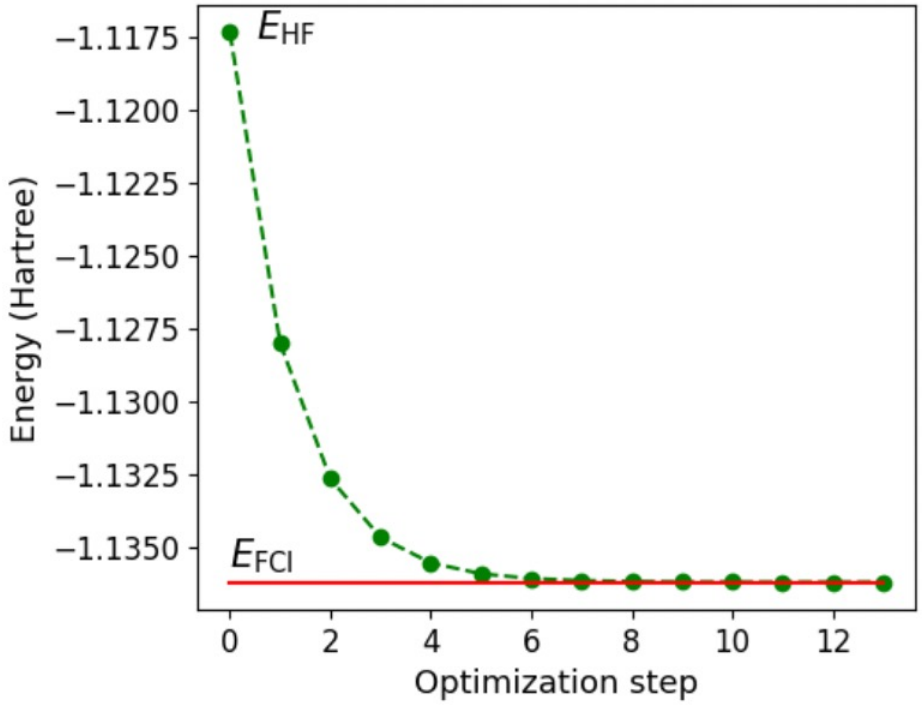
```
Final value of the ground-state energy = -1.13618883 Ha

Optimal value of the circuit parameter = 0.2089
```

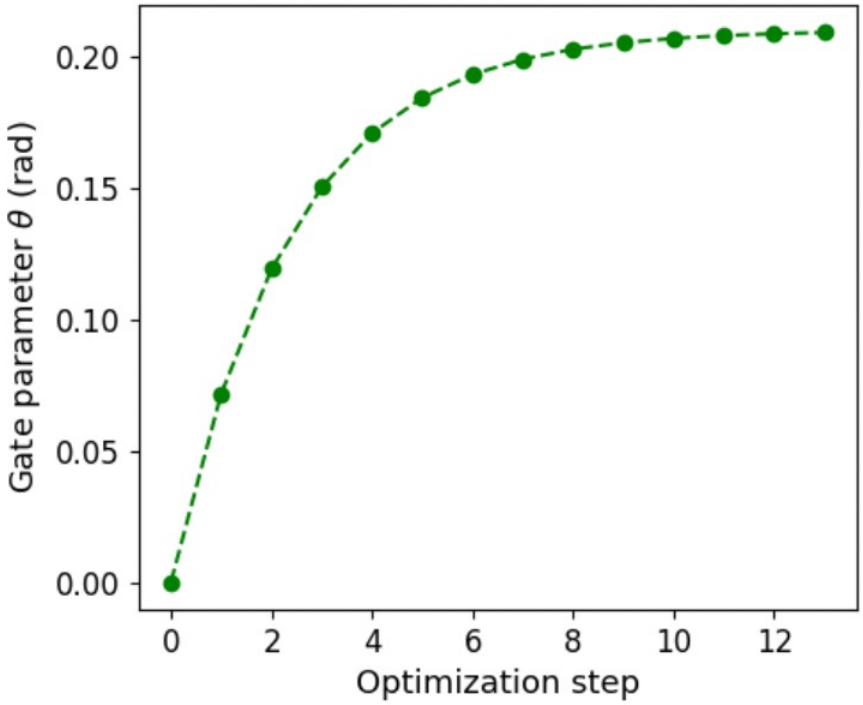
- The minimizer is identified

$$|\Psi(\theta^*)\rangle = 0.994\ |1100\rangle - 0.104\ |0011\rangle,$$

which can be mapped back to the ground state of the original (Fermionic) Hamiltonian of H_2 by reversing the Jordan-Wigner transform.

- Changes of E and θ during the iteration:





See https://pennylane.ai/qml/demos/tutorial_vqe.html for the complete code.

Summary

- VQA: cost function and ansatz.
- VQE: a variational solver for eigenstate problems.
- Application of VQE in quantum chemistry:

  mapping spin orbitals to qubits, and energy measurement.

Homework

- Review the lecture slides; you may find the review questions in the next slides helpful.
- Read the Qiskit tutorial on VQE https://qiskit.org/textbook/ch-applications/vqe-molecules.html.

Review questions

- What are the key ingredients of a VQA?
- What are the criteria for a good cost function?
- Give an example of a problem-inspired/hardware-efficient ansatz.
- When simulating a fermionic system, we often require more qubits than the number of feromions. Why?
- For the toy example of H_2 ground state, is the ansatz problem-inspired or hardware-efficient?