Noisy Intermediate-Scale Quantum algorithms

Alba Cervera-Lierta
Machine Learning in Quantum Physics and Chemistry
Warsaw, August 25, 2021
Outlook

0. What is Quantum Computation?
1. Quantum computing in the NISQ era
2. Variational Quantum Algorithms
3. Squeezing the NISQ lemon

Break

4. NISQ algorithms
5. NISQ horizon

Coding time! (if we have time)

Slides: albacl.github.io

Tutorials (Tequila):
github.com/AlbaCL/VQA_tutorials
The basics of Quantum Computation

- What
- How
- Why
- Where
- Who
What is a quantum computer

A device capable of processing data in a quantum mechanical form. A device that uses the properties of quantum mechanics to process data.

- **SOFTWARE**
  - Classic
- **HARDWARE**
  - Qubits
- **Instructions**
- **Result**

The minimal quantum information units
How does it work

Bit

\[ |0\rangle = (1) \]
\[ |1\rangle = (0) \]

Qubit

\[ |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \]

\[ \alpha, \beta \in \mathbb{C} \]

Entanglement

\[ |\psi\rangle \neq |\psi\rangle_1 \otimes |\psi\rangle_2 \]
A new paradigm in computation

A single operation (logic gate) affects all possible qubit states.

\[ |\psi_0\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \]

\[ CNOT |\psi_0\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \delta|10\rangle \]

4 “sums” with a single physical operation!
Some math...

- **Pure states** (those that can be written in state form): surface of the Bloch sphere
- **Mixed states** (can only be written with the density matrix formalism): inside of the Bloch sphere

**SU(2):**
- Three generators: $\sigma_x$, $\sigma_y$, and $\sigma_z$ (the Pauli matrices)
- Isomorphic to SO(3), meaning the evolution of the qubit states can be represented with rotations on the Bloch sphere

$$R_x(\theta) = e^{i \theta \sigma_x}, \text{ etc}$$

In general (n-qubits) quantum logic operations are represented by Hermitian matrices (unitary complex matrices) [SU($2^n$) group]

Qubit states have $2^n - 2$ degrees of freedom (you need this number of complex variables to fully represent an arbitrary state).
How does it work

Qubit: physical system that 1) is quantum and 2) have two well-defined states

Example: atomic orbitals

Example: superconducting circuit (transmon qubit)
Why do we need a quantum computer

MareNostrum supercomputer (BSC)
Why do we need a quantum computer

Less time... and less energy!


“Quantum computational advantage using photons”, Science eabe8770 (2020)
Where are they being constructed

Image: "Overview on quantum initiatives worldwide", Araceli Venegas-Gomez (Qureca Ltd)
Quantum computing in the NISQ era

Quantum Computing in the NISQ era and beyond

John Preskill

Noisy intermediate-scale quantum (NISQ) algorithms

Kishor Bharti,1, * Alba Cervera-Lierta,2, 3, * Thi Ha Kyaw,2, 3, * Tobias Haug,4 Sumner Alperin-Lea,3 Abhinav Anand,3 Matthias Degroote,2, 3, 5 Hermanni Heimonen,1 Jakob S. Kottmann,2, 3 Tim Menke,6, 7, 8 Wai-Keong Mok,1 Sukin Sim,9 Leong-Chuan Kwek,1, 10, 11, † and Alán Aspuru-Guzik2, 3, 12, 13, ‡

arXiv:2101.08448
The power of quantum

Why do we need a quantum computer?
- Quantum simulation
- Solve problems beyond P and BPP

Quantum computers are powerful but not limitless

Which problems are BQP?
Approximate solutions to NP problems?
The power of quantum

qubit, April '95, Schumacher, Quantum coding. PRA 51, 2738–2747

Predicting research trends with semantic and neural networks with an application in quantum physics, M. Krenn, A. Zeilinger, PNAS 117 (4) 1910-1916 (2020)

Trapped ions companies: IonQ, Honeywell, Alpine QT

Quantum supremacy using a programmable superconducting processor, Google AI, Nature 574, 505(2019).

Quantum computational advantage using photons, USTC (Chao-Yang Lu, Jian-Wei Pan’s group), Science 370, 1460 (2020).

Quantum Computing Paradigms

- Measurement-based (one-way)
- Quantum annealing
- Adiabatic Quantum Computing
- Quantum simulators
- Boson Sampling
- Digital-analog quantum computation
- Gate-based quantum computation

## Gate-based Quantum Computing

<table>
<thead>
<tr>
<th>Name</th>
<th>Definition</th>
<th>Mathematically</th>
<th>Diagramatically</th>
<th>Experimentally</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qubits</td>
<td>2-level quantum systems</td>
<td>Two-dimensional complex vector</td>
<td>Lines</td>
<td>Photons, superconducting circuits, trapped ions, ...</td>
</tr>
<tr>
<td>Gates</td>
<td>Interactions between qubits that generate superposition and entanglement in a controllable way</td>
<td>SU(2^n) matrices where n is the number of qubits involved in the operation</td>
<td>Boxes that specify the gate and some vertical symbols that represent particular entangling gates (CNOT and SWAP)</td>
<td>Laser pulses (ions, photons), optical devices (SPDC, PS, ...) microwave pulses</td>
</tr>
<tr>
<td>Measurement</td>
<td>Interaction with the individual qubits that forces its collapse to one of the two levels</td>
<td>Projector over the computational basis state</td>
<td>Box with a “meter” symbol</td>
<td>Coupling with a cavity, photon detectors, ...</td>
</tr>
</tbody>
</table>

![Quantum Circuit Diagram](image)

**Equations:**

\[ H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

\[ \text{CNOT} = CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]

This quantum circuit generates the GHZ state.

---

From theory to experiment

Integer factorization (Shor’s) algorithm

From theory to experiment

Example: \( n = 4 \) Ising model simulation.

- \(~35\) gates circuit depth
- \(~500\) ns entangling gates
- \(~100\) ns single-qubit

\(<17500\) ns = \(17.5\) \(\mu\)s

Qubits coherence time \(\sim 50\) \(\mu\)s

Errors coming from readout, cross-talk, relaxation, ... are relevant and difficult to track

March-May 2018

NISQ vs Fault-Tolerant

Who lives in the Fortress?
- Factorization algorithm
- Grover search algorithm
- ...

Who lives in the Plains?
- Variational Quantum Eigensolver
- QAOA
- ...

~1000 noisy qubits/logical qubit

Image: “Quantum computing: near- and far-term opportunities”, Ewan Munro, Medium @quantum_wa
NISQ circuits

Imperfect gate operations.

We cannot run:

- Algorithms that require perfect control (e.g. Grover, QFT, ...)
- Circuits that require many gates
NISQ circuits

Imperfect gate operations.

We can run:
- Algorithms that do not require particular gates
- Circuits that require a few gates

Can we design algorithm resistant to these imperfect operations?
Clever ways to explore the Hilbert space

We can not apply exact algorithms...

... but we can explore the Hilbert space in other ways.

We can use our quantum computer as a machine that generates variational states and find a way to converge towards the solution.
Noisy Intermediate-Scale Quantum

Why is QC hard experimentally?
- Qubits have to interact strongly (by means of the quantum logic gates)...
- ...but not with the environment...
- ...except if we want to measure them.

What is the state-of-the-art in digital quantum computing?
- ~50 qubit devices
- Error rates of ~10^-3
- No Quantum Error Correction (QEC)

Noisy Intermediate-Scale Quantum (NISQ) computing
- 50-100 qubits
- Low error rates
- No QEC

What can we do in NISQ?
- Good trial field to study physics
- Possible applications?
- A step in the path towards Fault Tolerant QC

Noisy Intermediate Scale Quantum computation

- A few qubits (~100)
- Noise
- Decoherence
- Classical optimizers
- Different qubits architectures

Quantum McGyver carrying a quantum advantage experiment

Something interesting:
Understanding quantum physics
Something useful:
Quantum advantage
Variational Quantum Algorithms is one of the most used NISQ paradigms, but it is not the only one.

The parents of VQA are the Variational Quantum Eigensolver (VQE) and the Quantum Approximate Optimization Algorithm (QAOA).
Variational Quantum Algorithms

e.g. Variational Quantum Eigensolver, Classifier, Autoencoder, QAOA...

Parameterized quantum circuit

Initial state $|\psi_0\rangle$

Quantum circuit that depend on $\theta$

Output $|\psi\rangle$

Expectation value $\langle \psi | H | \psi \rangle$

Objective $H$

Variational principle: $E = \langle \psi | H | \psi \rangle \geq E_0$

New $\theta$

Classical optimization

$E_0 + \epsilon$

Parameterized quantum circuits

\[ |\Phi(\theta)\rangle = U(\theta)|0\rangle \]

\[ E_0 = \min_{\theta} \langle\Phi(\theta)|H|\Phi(\theta)\rangle = \min_{\theta} \langle 0|U^\dagger(\theta)HU(\theta)|0\rangle \]

Assumptions:

1. There exist a set of parameters that approximates the ground state \( \exists \theta^* \ |\Phi(\theta^*)\rangle \simeq |gs\rangle \)
2. Our PQC can represent that solution
3. We can converge towards the solution (we do not get trapped in local minima)
4. The PQC can be run on a NISQ computer
Parameterized quantum circuits

How can we design $U(\theta)$?

Two strategies:

1. Problem-inspired PQC ansatz.
   a) Approximation to the solution by construction.
   b) High-circuit depth/# gates in general (not always hardware-friendly)

2. Hardware-efficient ansatz.
   a) Heuristic ansatz
   b) Low circuit depth/# gates in general (hardware-friendly)
Parameterized quantum circuits

**Problem-inspired ansatz:** UCCSD, QAOA, etc (see VQE section ahead).

**Hardware-efficient ansatz:**

- Low circuit depth
- Hardware-friendly gates (native gates that can be implemented experimentally)
- Respectful with qubit connectivity
- Useful for general problems (no problem-inspired ansatzes).

**Example:**
Layers of subcircuits.
Each layer: single-qubit gates + entangling gates
Objective function

It encodes the problem in a quantum operator, e.g. a Hamiltonian

$$\langle H \rangle_{U(\theta)} \equiv \langle 0 | U^\dagger(\theta) H U(\theta) | 0 \rangle$$

The objective is decomposed into Pauli strings which expectation value can be measured with the quantum computer.

$$H = \sum_{k=1}^{M} c_k \hat{P}_k \quad \Rightarrow \quad \langle H \rangle_U = \sum_{k=1}^{M} c_k \langle \hat{P}_k \rangle_U$$

An objective can also be the fidelity w.r.t. a particular target state that we are trying to match.

$$F(\Psi, \Psi_{U(\theta)}) \equiv |\langle \Psi | \Psi_{U(\theta)} \rangle|^2$$
Measurement

We need to find a way to extract information from our quantum computer.

In general, quantum devices project in a particular basis, normally the z-basis.

\[
\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
\]

\[
\sigma_z |0\rangle = +1 |0\rangle, \quad \sigma_z |1\rangle = -1 |1\rangle
\]

This means we only measure the eigenvalues of the \(\sigma_z\) operator, namely the “0”s and the “1”s.

\[
\langle \hat{\sigma}_z \rangle = 2p_0 - 1
\]

In other basis, we need to rotate the state to that particular basis first

\[
\hat{\sigma}_x = R_y^\dagger \left( \frac{\pi}{2} \right) \hat{\sigma}_z R_y \left( \frac{\pi}{2} \right) = H_d \hat{\sigma}_z H_d,
\]

\[
\hat{\sigma}_y = R_x^\dagger \left( \frac{\pi}{2} \right) \hat{\sigma}_z R_x \left( \frac{\pi}{2} \right) = S H_d \hat{\sigma}_z H_d S^\dagger.
\]

...and measure how many 0 we obtain as in the \(\sigma_z\) case.

Classical optimization

We need to navigate the quantum circuit parameter space, e.g. by using gradient-based methods

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta \partial_i f(\theta)$$

The gradients are expectation values of the quantum circuit derivatives w.r.t. a parameter.

Example: parameter-shift rule

$$U(\theta) = V(\theta_{-i})G(\theta_i)W(\theta_{-i})$$

$$G = e^{-i\theta_i g}$$

Eigenvalues of $g$ are $\pm \lambda$

$$\partial_i \langle f(\theta) \rangle = \lambda \left( \langle f(\theta_+) \rangle - \langle f(\theta_-) \rangle \right)$$

$$\theta_{\pm} = \theta \pm (\pi/4\lambda) e_i$$

Gradient-free: genetic algorithms, reinforcement learning, ...
Squeezing the NISQ lemon

My perfect quantum algorithm
Quantum Error Mitigation

A set of classical post-processing techniques and active operations on hardware that allow to correct or compensate the errors from a noisy quantum computer.

Zero-noise extrapolation

Instead of running our circuit unitary $U$, we run different circuits $U(UU^\dagger)^n$ (increasingly noisy). Extrapolate the result for zero-noise $U$.

Stabilizer based approach

relies on the information associated with conserved quantities such as spin and particle number conserving ansatz. If any change in such quantities is detected, one can pinpoint an error in the circuit.
Quantum Error Mitigation

Quantum Optimal Control strategies

Dynamical Decoupling:

Designed to suppress decoherence via fancy pulses to the system so that it cancels the system-bath interaction to a given order in time dependent perturbation theory

Pulse shaping technique:

passive cancellation of system-bath interaction.

Among many others...

The **barren-plateaux** problem

Compute the gradients with the quantum circuit and use these values to run a classical minimizer, e.g. Nelder-Mead, Adam, ...

With no prior knowledge about the solution, $\hat{\theta}$ parameters are initialized at random.

**Consequence:** **barren-plateaux**

- The expected value of the gradient is zero!
- The expected value of the variance is also zero!

**Solutions**

- Use parameters close to the solution.
- Use local cost functions instead of global ones.
- Introduce correlations between parameters.

Ref.: M. Cerezo et. al. Nature Communications 12, 1791 (2021)
Expressibility

When setting a PQC ansatz we have to be careful to not narrow the Hilbert space accessible by the PQC so we can reach a good approximation of the solution state.

We can quantify the expressibility of a PQC by computing the distance between a Haar distribution of the states and states generated by the PQC.

\[ A_U^{(t)} = \| \int_{\text{Haar}} (|\psi\rangle \langle \psi|)^{\otimes t} d\psi - \int_{\theta} (|\psi_\theta\rangle \langle \psi_\theta|)^{\otimes t} d\psi_\theta \| \]


Circuit compilation

Native and universal gate sets:

**Solovay-Kitaev theorem:** With a universal gate set we can approximate with epsilon accuracy any SU(N) with a circuit of polynomial depth.

**Gottesman–Knill theorem:** Circuits composed by gates from the Clifford group (Clifford circuits) can be simulated efficiently with a classical computer.

Gate sets are usually composed by Clifford gates + one non-clifford gate, e.g. \{H, S, CNOT\} + T

However, depending on the hardware implementation, some gates are easier to control.

*The more native gates, the shorter and simpler the circuit*
## Applications

<table>
<thead>
<tr>
<th>A. Many-body physics and chemistry</th>
<th>35</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Qubit encodings</td>
<td>35</td>
</tr>
<tr>
<td>2. Constructing electronic Hamiltonians</td>
<td>36</td>
</tr>
<tr>
<td>3. Variational quantum eigensolver</td>
<td>37</td>
</tr>
<tr>
<td>4. Variational quantum eigensolver for excited states</td>
<td>38</td>
</tr>
<tr>
<td>5. Hamiltonian simulation</td>
<td>40</td>
</tr>
<tr>
<td>6. Quantum information scrambling and thermalization</td>
<td>41</td>
</tr>
<tr>
<td>7. Simulating open quantum systems</td>
<td>41</td>
</tr>
<tr>
<td>8. Nonequilibrium steady state</td>
<td>42</td>
</tr>
<tr>
<td>9. Gibbs state preparation</td>
<td>43</td>
</tr>
<tr>
<td>10. Many-body ground state preparation</td>
<td>43</td>
</tr>
<tr>
<td>11. Quantum autoencoder</td>
<td>44</td>
</tr>
<tr>
<td>12. Quantum computer-aided design</td>
<td>44</td>
</tr>
<tr>
<td>B. Machine learning</td>
<td>45</td>
</tr>
<tr>
<td>1. Supervised learning</td>
<td>46</td>
</tr>
<tr>
<td>2. Unsupervised learning</td>
<td>48</td>
</tr>
<tr>
<td>3. Reinforcement learning</td>
<td>49</td>
</tr>
<tr>
<td>C. Combinatorial optimization</td>
<td>50</td>
</tr>
<tr>
<td>1. Max-Cut</td>
<td>50</td>
</tr>
<tr>
<td>2. Other combinatorial optimization problems</td>
<td>52</td>
</tr>
<tr>
<td>D. Numerical solvers</td>
<td>52</td>
</tr>
<tr>
<td>1. Variational quantum factoring</td>
<td>52</td>
</tr>
<tr>
<td>2. Singular value decomposition</td>
<td>53</td>
</tr>
<tr>
<td>3. Linear system problem</td>
<td>53</td>
</tr>
<tr>
<td>4. Non-linear differential equations</td>
<td>54</td>
</tr>
<tr>
<td>E. Finance</td>
<td>54</td>
</tr>
<tr>
<td>1. Portfolio optimization</td>
<td>55</td>
</tr>
<tr>
<td>2. Fraud detection</td>
<td>56</td>
</tr>
<tr>
<td>F. Other applications</td>
<td>56</td>
</tr>
<tr>
<td>1. Quantum foundations</td>
<td>56</td>
</tr>
<tr>
<td>2. Quantum optimal control</td>
<td>56</td>
</tr>
<tr>
<td>3. Quantum metrology</td>
<td>57</td>
</tr>
<tr>
<td>4. Fidelity estimation</td>
<td>57</td>
</tr>
<tr>
<td>5. Quantum error correction</td>
<td>57</td>
</tr>
<tr>
<td>6. Nuclear physics</td>
<td>57</td>
</tr>
<tr>
<td>7. Entanglement properties</td>
<td>58</td>
</tr>
</tbody>
</table>
Variational Quantum Eigensolver (VQE)

Resources:
- Artur Izmaylov “Quantum Chemistry on a Quantum Computer” course on Youtube

Tutorials:
- Qiskit: https://qiskit.org/textbook/ch-applications/vqe-molecules.html
- Tequila: https://github.com/aspuru-guzik-group/tequila-tutorials
- Pennylane: https://pennylane.ai/qml/demos/tutorial_vqe.html
Electronic structure problem

The electronic structure Hamiltonian describes the dynamics of an atom or a molecule.

In the Born-Oppenheimer approximation, it has two main terms:

$$H_{mol} = H_{nucl}(\vec{R}) + H_{elec}(\vec{R}, \vec{r})$$

The wavefunction can be factorized as well

$$\psi(\vec{R}, \vec{r}) = \phi_{nucl}(\vec{R})\chi_{elec}(\vec{R}, \vec{r})$$

The part of interest for chemistry is solving the electronic one:

$$H_{elec}\chi_{elec}(\vec{R}, \vec{r}) = E_{elec}(\vec{R})\chi_{elec}(\vec{R}, \vec{r})$$

$$\hat{H}_{elec} = -\sum_{i} \frac{\nabla_{\vec{r}_{i}}^{2}}{2} - \sum_{i,j} \frac{Z_{i}}{|\vec{R}_{i} - \vec{r}_{j}|} + \sum_{i,j>i} \frac{1}{|\vec{r}_{i} - \vec{r}_{j}|}$$

- Kinetic energy electrons
- Interaction electrons-nucleus
- Interaction between electrons

Electronic structure problem

How does the wave-function look like?

Single electrons wavefunction are the electronic orbitals.

Two-electron wavefunctions are a combinations of these orbitals in what are called Slater determinants.

Slater determinants manipulation in the first quantization might be cumbersome, so we move to the second quantization or Fock space:

$$|\psi\rangle = \sum_{\text{orbitals}} C_k |n_1, \ldots, n_k\rangle$$

Electronic wave-function

Occupation number of that orbital
= 0 (no orbital)
or 1 (there is an electron in that orbital)

Electronic structure problem

The electronic Hamiltonian in the second quantization becomes:

$$H_{2q} = \sum_{p,q} h_{pq} a_{p}^{\dagger} a_{q} + \sum_{p,q,r,s} h_{pqr}s a_{p}^{\dagger} a_{q}^{\dagger} a_{r} a_{s}$$

<table>
<thead>
<tr>
<th>Single-excitations</th>
<th>Double-excitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-electron moves from one orbital to another</td>
<td>2-electrons move from one orbital to another</td>
</tr>
</tbody>
</table>

“Couple-Cluster Single-Double” model (CCSD)

Creation and annihilation operators:
- $a_{p}^{\dagger}$: Adds an electron to the “p” orbital
- $a_{q}$: Removes an electron from the “q” orbital
CCSD on a quantum computer

\[ H_{2q} = \sum_{p,q} h_{pq} a_p^\dagger a_q + \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \]

We can not compute expectation values of the creation and annihilation operators.

We apply a unitary transformation that maps these operators into Pauli strings:

\[ a^\dagger_k \equiv j = 1 \quad k - 1 \quad j \quad \sigma_j^z - i \sigma_j^y \]

We have our objective function to minimize with our VQA!

Next, what do we use as a PQC ansatz?

We are looking for a quantum circuit (a.k.a. unitary operation) that generates the ground state of an electronic structure Hamiltonian:

\[ U_{\text{UCCSD}} \sim e^{i H_{\text{UCCSD}}} \]

Hartree-Fock approximation: single electron orbitals (first-order approximation)

\[ |\Psi(\theta)\rangle = e^{T(\theta) - T(\theta)^\dagger} |\Psi_{\text{HF}}\rangle \]

\[ T(\theta) = T_1(\theta) + T_2(\theta) + \cdots \]

\[ T_1(\theta) = \sum_{i \in \text{occ}, j \in \text{virt}} \theta_i^j \hat{a}_j^\dagger \hat{a}_i \]

\[ T_2(\theta) = \sum_{i_1, i_2 \in \text{occ}, j_1, j_2 \in \text{virt}} \theta_{i_1, i_2}^{j_1, j_2} \hat{a}_{j_2}^\dagger \hat{a}_{i_2}^\dagger \hat{a}_{j_1} \hat{a}_{i_1} \]

Remember that \( e^{i \theta \sigma_x} = R_x(\theta) \) etc. From Pauli strings we can obtain the necessary quantum gates.

Coefficients to be determined (with our VQA!)

Couple-cluster operators (single, double, triple, … excitations).

We transform it into spin operators (Jordan-Wigner, etc) and use it as a unitary generators.
The Variational Quantum Eigensolver

We will use the classical subroutting to obtain the parameters from the UCC operator.

Electronic structure $H$

Initial state $|\psi_{HF}\rangle$

Output $|\psi\rangle$

Expectation value $\langle\psi|H|\psi\rangle$

$E_0 + \epsilon$

Classical optimization
Quantum Machine Learning
Quantum algorithms feed with classical or quantum data

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
From classical to quantum NN

Classical

Quantum (circuit centric)

Input neurons | Hidden neurons | Output neurons

Encoding | Processing | Measure


The minimal QNN

What is the most simple (but universal) NN?

What is the most simple (but universal) QNN?

- Single hidden layer NN
- Single-qubit QNN

(a) Neural network
(b) Quantum classifier

A. Pérez-Salinas, ACL, E. Gil-Fuster and J. I. Latorre, Quantum 4, 226 (2020)
Encoding the data

A product of unitaries can be written with a single unitary

\[ U(\vec{\phi}_1) \ldots U(\vec{\phi}_N) \equiv U(\vec{\phi}) \]

If we add some fixed parameter dependency (the data), the operation becomes flexible and data-dependent.

\[ U(\vec{\phi}, \vec{x}) \equiv U(\vec{\phi}_N)U(\vec{x}) \ldots U(\vec{\phi}_1)U(\vec{x}) \]

The paths depend on the data \( x \)

Correct label

Data point 1

Correct label

Data point 2

A. Pérez-Salinas, ACL, E. Gil-Fuster and J. I. Latorre, Quantum 4, 226 (2020)
Supervised Learning

\[ |\psi_0\rangle \rightarrow |\psi(\vec{x}, \vec{\theta})\rangle \rightarrow |\psi(\vec{x}, \vec{\theta}, \vec{\phi})\rangle \]

Encode the data (quantum feature space)

\[ S(\vec{x}, \vec{\theta}) \]

Rotate to the correct measurement basis

\[ U(\phi) \]

We can then compute the Kernel

\[ \kappa(x_i, x_j) \equiv \langle \Phi(x_i) | \Phi(x_j) \rangle \]

Or minimize the fidelity w.r.t. target states

\[ C(\theta) = \sum_{i=1}^{D} (1 - |\langle y_i | \Psi(x_i, \theta) \rangle|^2) \]
Example 1: single-qubit classifier

Target states
Divide the Bloch sphere into \#classes sections

PQC
Layers of single-qubit gates where we encode the data and variational parameters into the angles.

Loss function
Overlap between the target state and the output state for all training points

Quantum classifier
Once trained, we introduce the test points and classify them according to the qubit state.

Check the Tequila tutorial!

A. Pérez-Salinas, ACL, E. Gil-Fuster and J. I. Latorre, Quantum 4, 226 (2020)
Example 2: single-qubit approximant

Quantum circuits can be theoretically written as partial Fourier series and, therefore, they can be universal function approximators. The more data re-uploading, the more precision can be achieved.

Same PQC as the quantum classifier but the loss function will be:

\[ \chi^2 = \frac{1}{M} \sum_{j=1}^{M} (\langle Z(x_j) \rangle - f(x_j))^2. \]

Check the Tequila tutorial!


What’s the true goal of VQE?

**GOAL**: find $|\psi\rangle$ that minimizes $\frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$.

Find the atomic separation that minimizes the energy

$\min \langle H(R) \rangle$

---

What’s the true goal of VQE?

To obtain this you need to scan from 0 to 300.

Each blue point is a VQE, that is, you have to prepare, run and optimize the quantum circuit.

Can we avoid to compute the uninteresting points?

Example 3: Meta-VQE

Parameterized Hamiltonian $H(\vec{\lambda})$

**Goal:** to find the quantum circuit that encodes the ground state of the Hamiltonian for any value of $\vec{\lambda}$

- Training points: $\vec{\lambda}_i$ for $i = 1, ..., M$
- Data re-uploading to encode the $\vec{\lambda}_i$ into the circuit
- Loss function with all $\langle H(\vec{\lambda}_i) \rangle$

**Standard VQE**

**Meta-VQE**

**Option 1:** run the circuit with test $\vec{\lambda}$ and obtain the g.s. energy profile.

**Option 2:** use $\Phi_{opt}$ and $\Theta_{opt}$ as starting point of a standard VQE optimization (opt-meta-VQE)

Check the Tequila tutorial!

ACL, J. Kottmann, A. Aspuru-Guzik, PRX Quantum 2, 020329 (2021)
NISQ horizon
What are the theoretical limits of NISQ quantum computation?

- Complexity theory
- New algorithms?
- What are the theoretical limits of NISQ quantum computation?

Software tools

Enabling technologies

Benchmark measures

Practical applications

Which technology?
- Scalability?
- Quantum control?
- Decoherence and cross-talk

How to program a NISQ algorithm

Next goal: fault-tolerant quantum computing

**Quantum Error Correction**: protect the quantum information in a highly entangled state.

QEC comes with a big qubit overhead: thousands (possible millions) of qubits to implement a quantum advantage experiment.

That’s why we have NISQ... but most of the NISQ algorithms can also be implemented in the Fault-Tolerant era.

Noise limits NISQ algorithms such as VQAs, we do not always have performance guarantees.

We don’t know how much time will it take or even if it’s possible to achieve F-T QC, but there is so much physics to explore along the way!
Acknowledgements

Alán Aspuru-Guzik
Leong-Chuan Kwek
Kishor Bharti
Thi Ha Kyaw
Tobias Haug
Matthias Degroote
Hermann Heimonen
Sumner Alperin
Abhinav Anand
Jakob Kottmann
Tim Menke
Wai-Keong Mok
Sukin (Hannah) Sim

~15 min Break...

Next: Coding time!
Backup slides
Circuit compilation

Circuit simplification: use identities or tools like the ZX calculus (graph representation of quantum circuits)

Circuit compilation

Theoretical circuit → Decomposition into native gates → Simplification (circuit identities) → Mapping to qubit chip topology

**Circuit simplification**: use identities or tools like the ZX calculi (graph representation of quantum circuits)

**Qubits connectivity problem**: not all qubits are physically connected, so we have to map our quantum circuits to the real devices.

---

The Variational Quantum Eigensolver

A. Peruzzo et al., Nature Comm. 5, 4213 (2014)

Bond dissociation curve of the He–H\(^+\) molecule.


Quantum Approximate Optimization Algorithm (QAOA)

Resources:
Musty Thoughts blog (Michał Stęchły):
https://www.mustythoughts.com/quantum-approximate-optimization-algorithm-explained

Tutorials:
- Qiskit: https://qiskit.org/textbook/ch-applications/qaoa.html
- Pennylane: https://pennylane.ai/qml/demos/tutorial_qaoa_intro.html,
  https://pennylane.ai/qml/demos/tutorial_qaoa_maxcut.html
Preliminaries

Time evolution:
\[ i\hbar \frac{d}{dt} |\Psi(t)\rangle = H |\Psi(t)\rangle \]
\[ |\Psi(t)\rangle = e^{-iHt} |\Psi(0)\rangle \]

Trotterization
\[ e^{A+B} = \lim_{n \to \infty} (e^{A/n} e^{B/n})^n \]
\[ H = H_1 + H_2 \]
\[ e^{-iHt} = e^{-itH_1-itH_2} = \lim_{n \to \infty} (e^{-itH_1/n} e^{-itH_2/n})^n \]
Apply alternatively \( e^{-itH_1} e^{-itH_2} \) in intervals of \( t/n \)

Adiabatic Quantum Evolution
\[ H = H_M + H_P \]
\[ H(s) = sH_M + (1 - s)H_P \]
If \( s \) small, we end up in the ground state of \( H_P \) (under certain assumptions)

Quantum Approximate Optimization Algorithm

Can be understood as an approximation of the Trotter decomposition of adiabatic evolution.

\[
H_M = \sum_{i=1}^{n} \sigma_x^i \\
H_P = \sum_{i=1}^{n} C(e_i) |e_i\rangle
\]

Mixing Hamiltonian

Problem Hamiltonian

Construct the circuit ansatz by alternating the problem and mixing Hamiltonians where \(\beta\) and \(\gamma\) are the variational parameters to be optimized classically.

\[
|\Psi(\gamma, \beta)\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_P} \ldots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P} |D\rangle
\]

Objective function: \(\langle \Psi(\gamma, \beta) | H_P(\gamma, \beta) |\Psi(\gamma, \beta)\rangle\)

full superposition state (in general)

<table>
<thead>
<tr>
<th></th>
<th>QAOA</th>
<th>VQE</th>
<th>Adiabatic Quantum Evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>Find an approximation of the ground state and its energy</td>
<td>Find an approximation of the ground state and its energy</td>
<td>End up in the ground state</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>$\beta, \gamma$ can take any value</td>
<td>$\theta$, can take any value</td>
<td>$s$ must be small</td>
</tr>
<tr>
<td><strong>Computational paradigm</strong></td>
<td>Digital</td>
<td>Digital</td>
<td>Analog</td>
</tr>
<tr>
<td><strong>Circuit ansatz</strong></td>
<td>Problem-specific, alternating</td>
<td>Problem-specific or other (e.g. Hardware-efficient)</td>
<td>-</td>
</tr>
</tbody>
</table>

Check Tequila QAOA vs VQE tutorial at github.com/AlbaCL/VQA_tutorials

Universal Approximation Theorem

Any continuous function \( f(x) \) can be approximated with \( \epsilon \) accuracy by the function

\[
h(\bar{x}) = \sum_{i=1}^{N} \alpha_i \varphi(\bar{w}_i \cdot \bar{x} + b_i)
\]

where \( \varphi \) is a nonconstant, bounded and continuous function.

A single-layer neural network can approximate any continuous function (providing enough neurons in the hidden layer).

Meta-VQE outlook

Parameterized Hamiltonian $H(\tilde{\lambda})$

**Goal:** to find the quantum circuit that encodes the ground state of the Hamiltonian for any value of $\tilde{\lambda}$

---

**Standard VQE**

- VQEs (individual minimizations)

**Energy**

---

**Meta-VQE**

- One minimization (training points)
- Evaluation curve (any point, without further minimizations)

---


ACL, J. Kottmann, A. Aspuru-Guzik, PRX Quantum 2, 020329 (2021)
The Meta-VQE

Parameterized Hamiltonian $H(\tilde{\lambda})$

Training points: $\tilde{\lambda}_i$ for $i = 1, \ldots, M$

Loss function with all $\left\{ H(\tilde{\lambda}_i) \right\}$

**Goal:** to find the quantum circuit that encodes the ground state of the Hamiltonian for any value of $\tilde{\lambda}$


ACL, J. Kottmann, A. Aspuru-Guzik, PRX Quantum 2, 020329 (2021)
The Meta-VQE output

Output: \( \Phi_{opt} \) and \( \Theta_{opt} \)

**Option 1:** run the circuit with test \( \lambda \) and obtain the g.s. energy profile.

\[
|0\rangle \otimes^n \xrightarrow{S(\lambda, \Phi_{opt})} U(\Theta_{opt}) \xrightarrow{} \langle H(\lambda) \rangle
\]

**Option 2:** use \( \Phi_{opt} \) and \( \Theta_{opt} \) as starting point of a standard VQE optimization (opt-meta-VQE)

\[
(\tilde{\Phi}_0, \tilde{\Theta}_0) \xrightarrow{\min_{\tilde{\Phi}, \tilde{\Theta}} \langle H(\tilde{\lambda}) \rangle} |0\rangle \otimes^n \xrightarrow{S(\lambda, \Phi)} U(\Theta) \xrightarrow{} \langle H(\lambda) \rangle
\]
1D XXZ spin chain

14 qubits simulation, $\lambda = 0.75$

Linear encoding: $R_z(w_1 \Delta + \phi_1)R_y(w_2 \Delta + \phi_2) \otimes$ Alternating CNOT

Processing layer: $R_z(\theta_1)R_y(\theta_2) \otimes$ Alternating CNOT

Results 2 encoding + 2 processing layers

$$H = \sum_{i=1}^{n} \sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \Delta \sigma_i^z \sigma_{i+1}^z + \lambda \sigma_i^z$$

ACL, J. Kottmann, A. Aspuru-Guzik, PRX Quantum 2, 020329 (2021)
$H_4$ molecule

$H_4$ molecule in 8 spin-orbitals (STO-3G basis set)

Ansatz: k-UpCCGSD (k=2 for these results)

Linear encoding: $\theta = \alpha + d\beta$
Non-linear encoding: $\theta = \alpha e^{\beta(y-d)} + \delta$ (floating Gaussians)