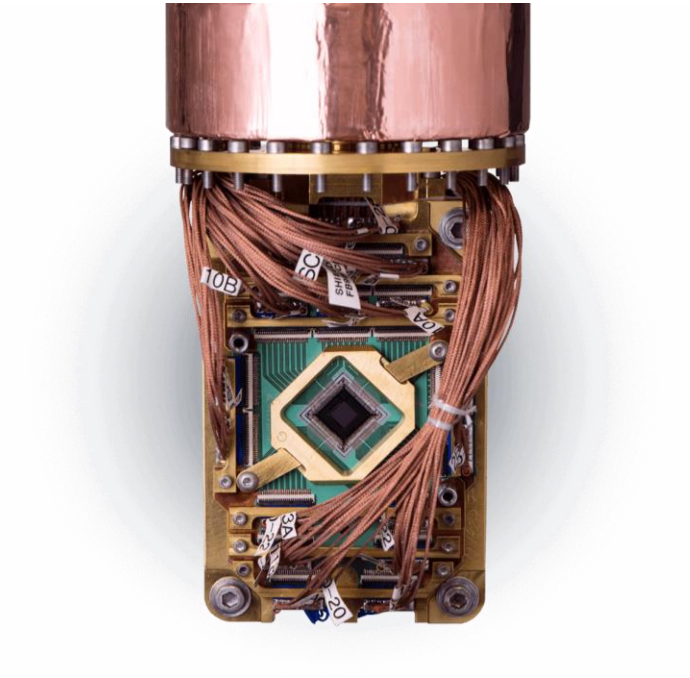
Advanced Parallel School 2022
Quantum Computing – Day 4
NISQ Algorithms

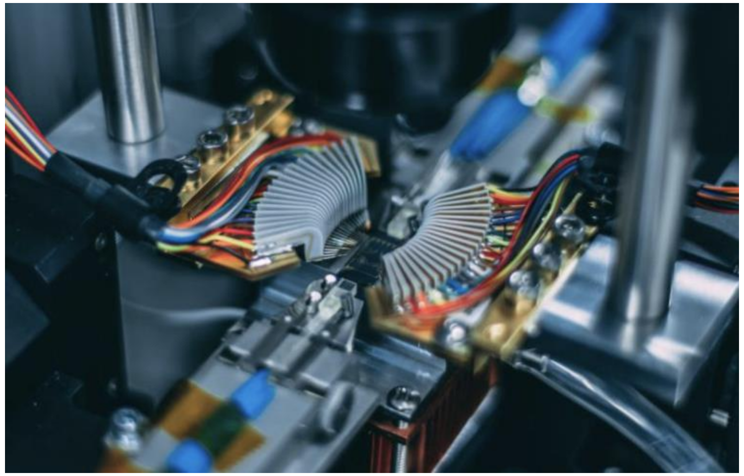
Mengoni Riccardo, PhD

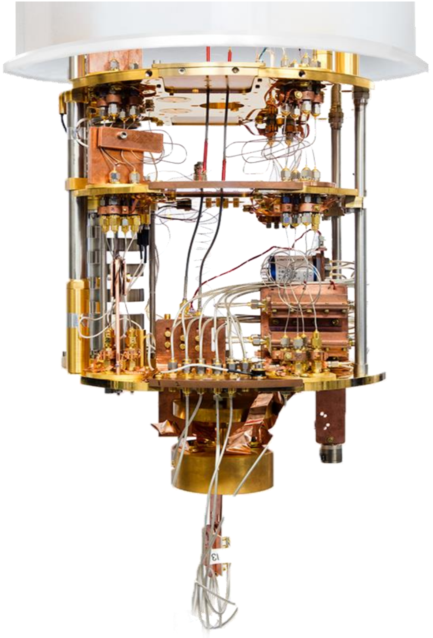
17 Feb 2022

CINECA QUANTUM COMPUTING LAB

How can we use the small and imperfect Quantum Devices (NISQ) we have today?



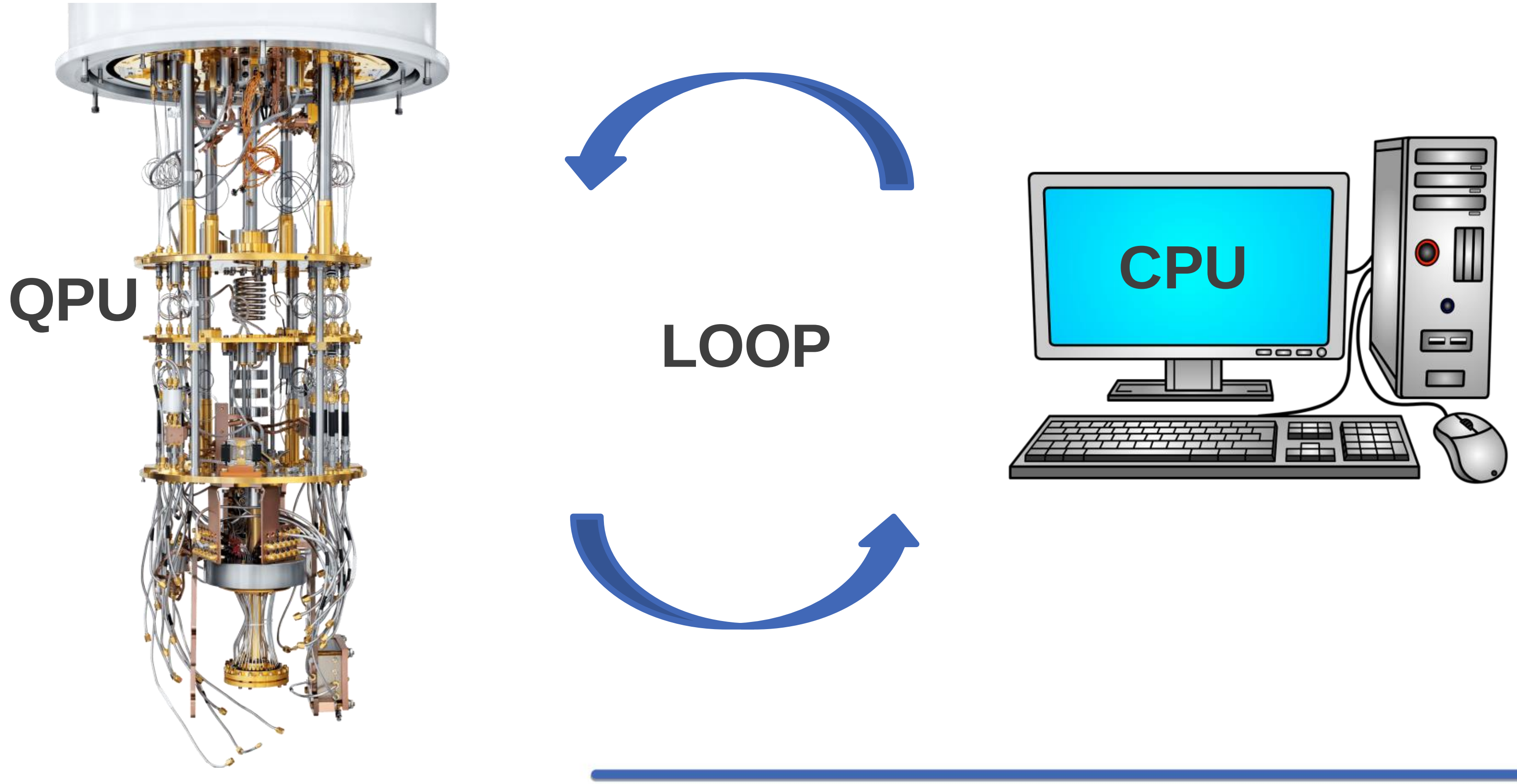




Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

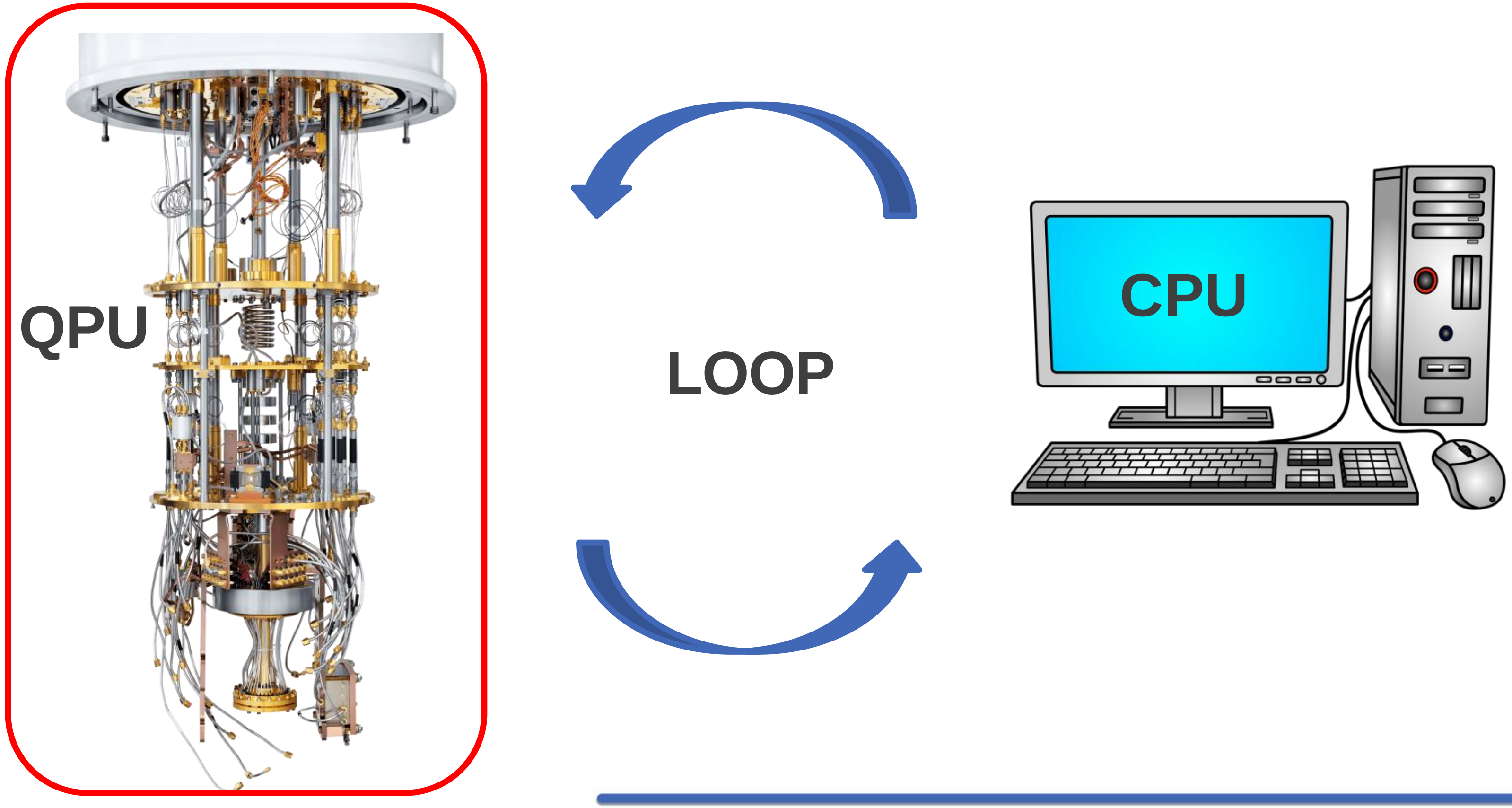
Hybrid Quantum-Classical algorithms



Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

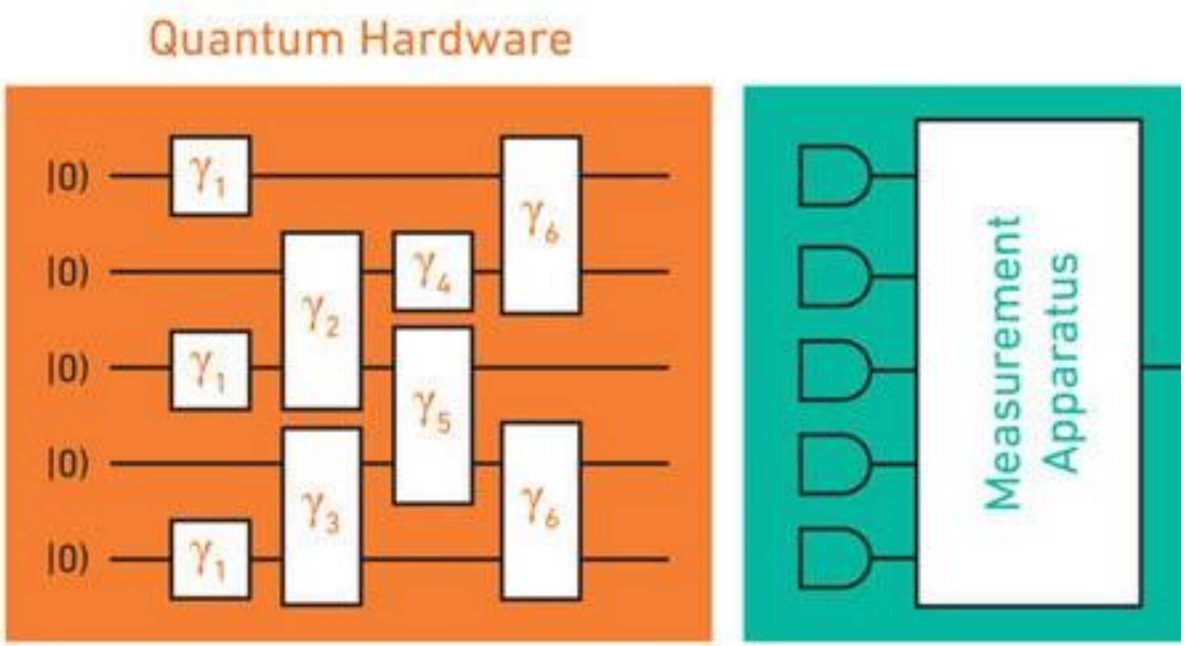
Hybrid Quantum-Classical algorithms



Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

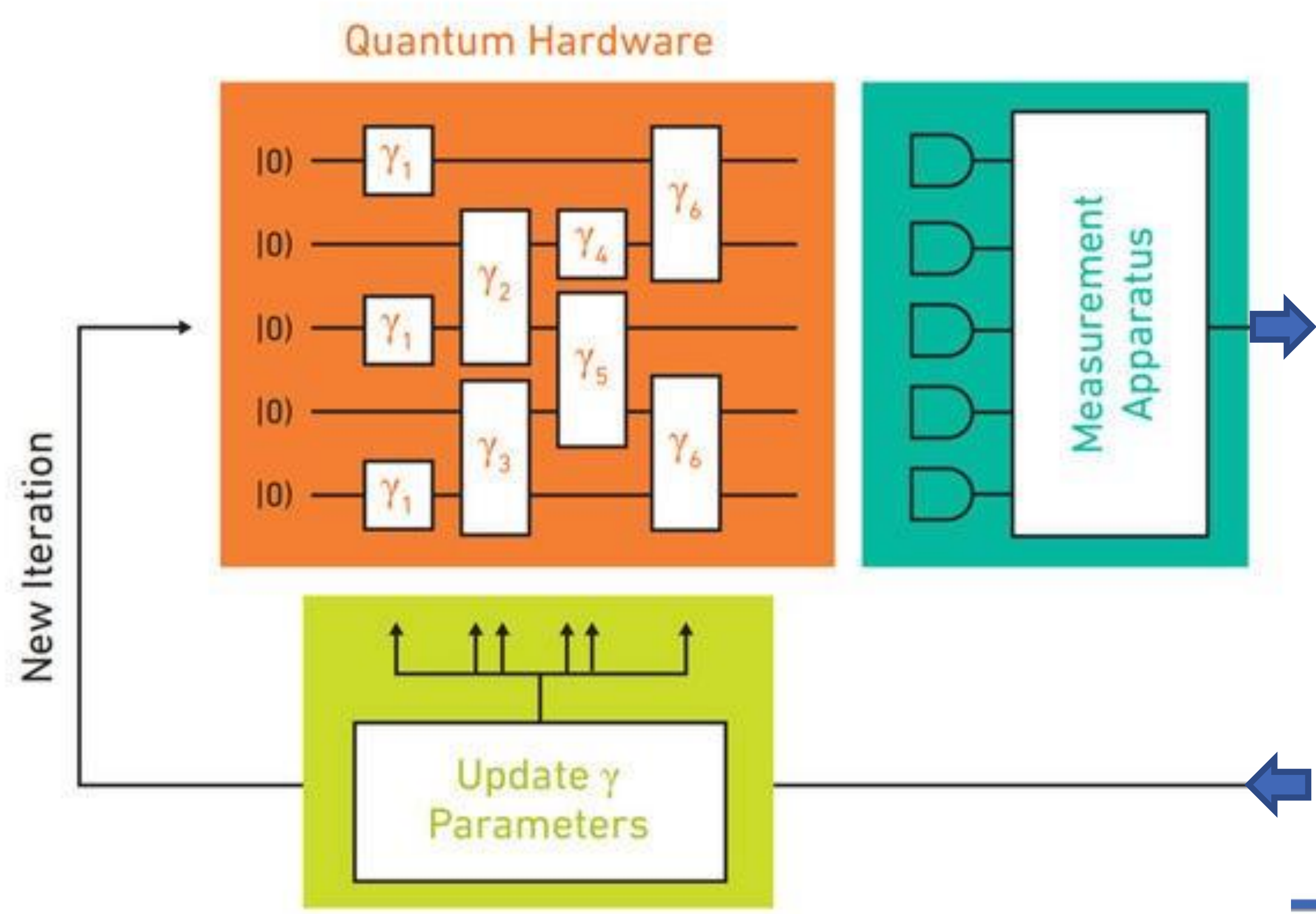
Parametric Quantum Circuits



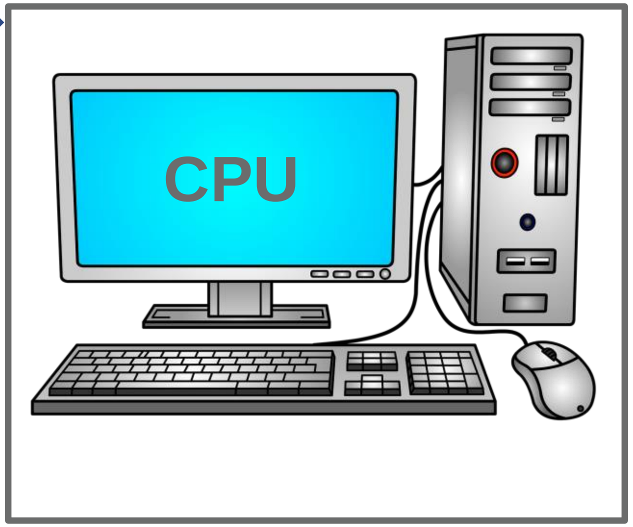
- Circuits that **use gates**, or in general, that apply **parameter-dependent operations** to qubits (e.g. Arbitrary rotations of angle γ)
- Circuits in which the **error is not corrected**
- **Shallow circuits**, i.e. of **limited depth** (1000 gates maximum, due to limited coherence times)

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU



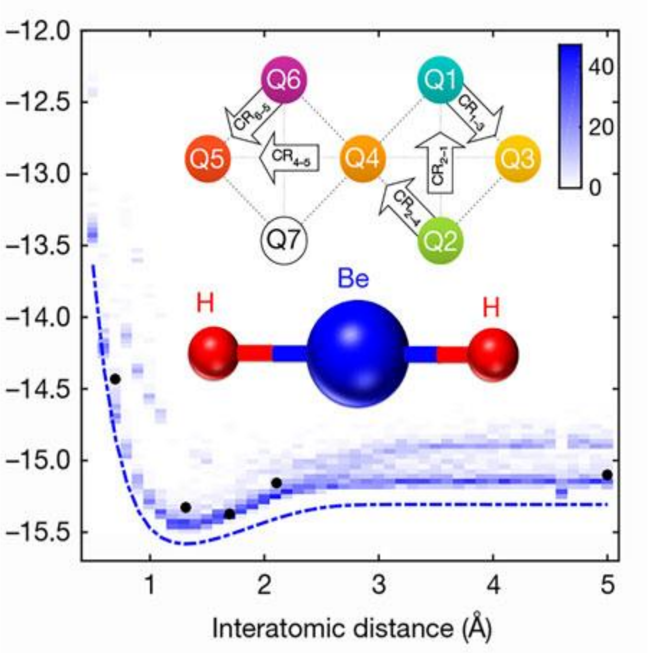
4. Classical optimization of parameters gamma



Quantum algorithms for NISQ Devices

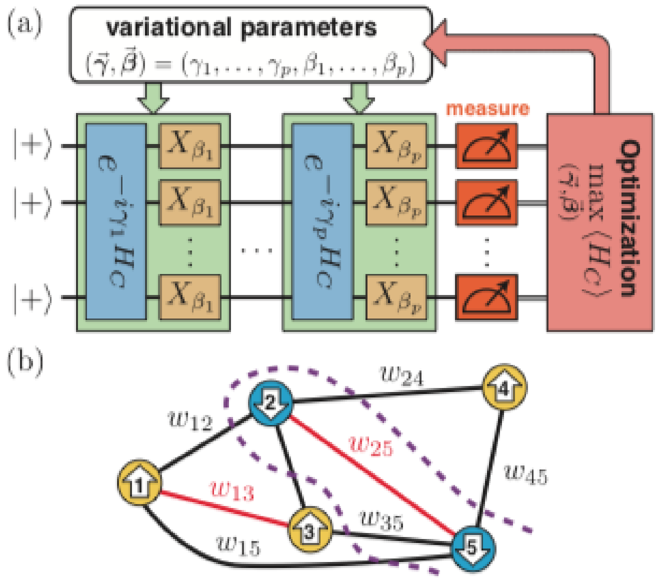
NISQ-ready algorithms for general purpose QPU

VQE



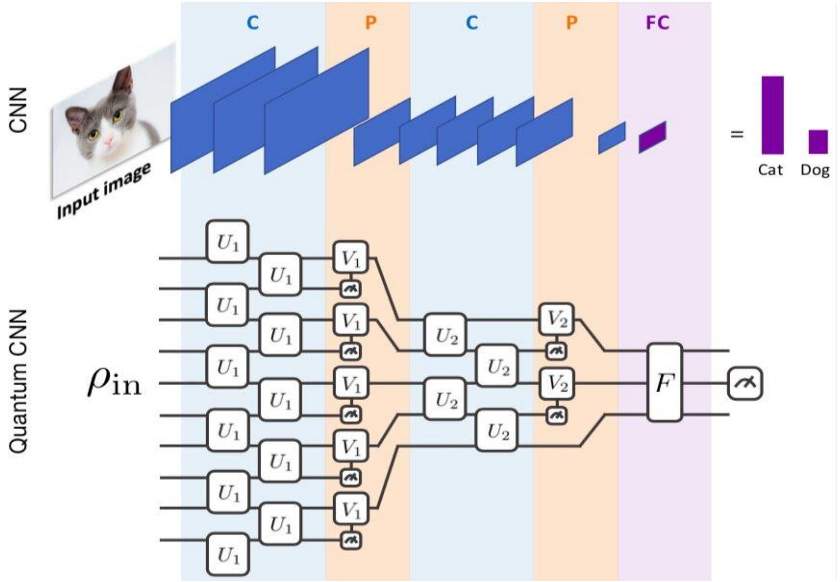
Quantum Chemistry

QAOA



Quantum Optimization

QNN



Quantum Machine Learning

Quantum algorithms for NISQ Devices

NISQ-ready algorithms for general purpose QPU

VQE

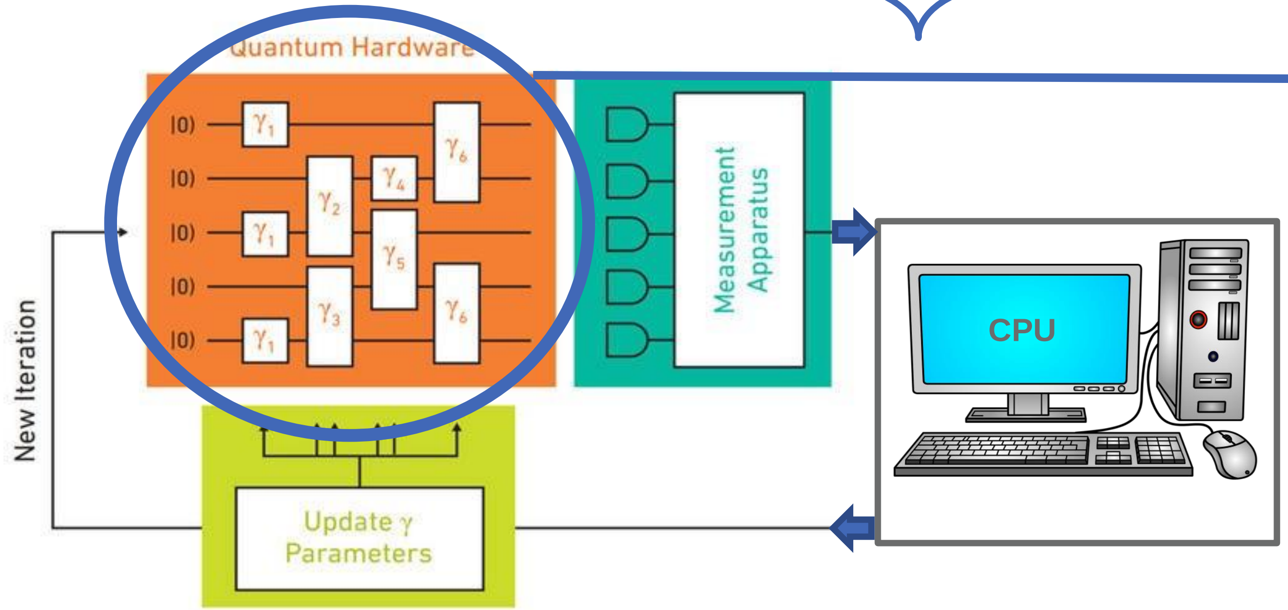
Quantum Chemistry

QAOA

Quantum Optimization

QNN

Quantum Machine Learning



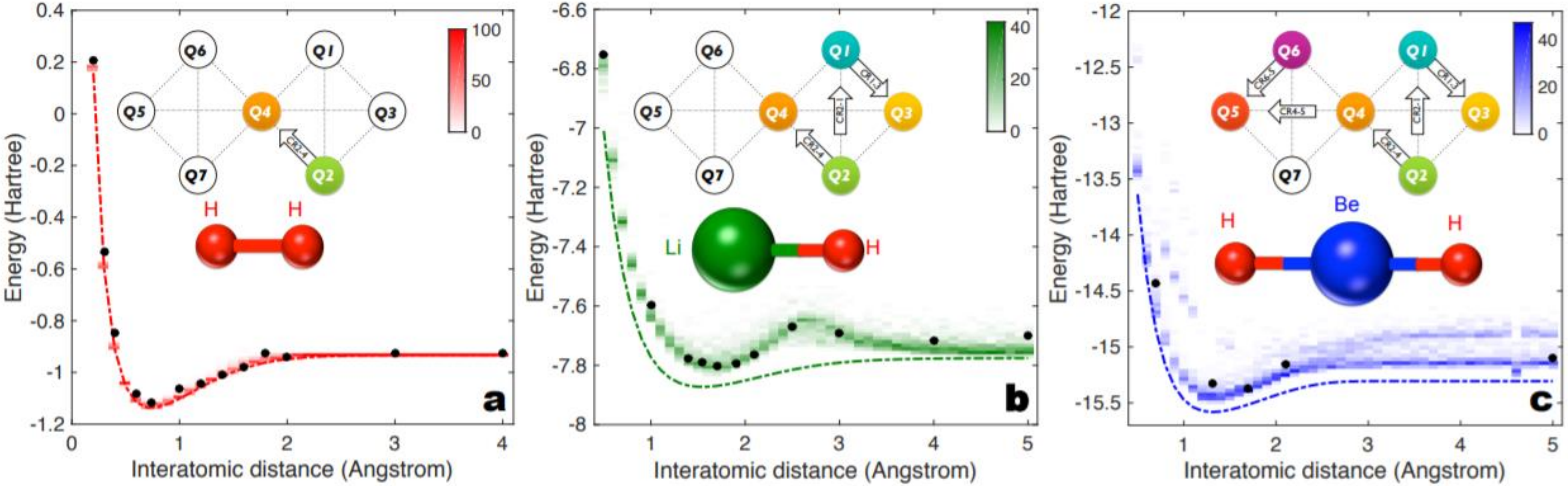
The **main difference** between **VQE, QAOA** and **QNN** concerns the **choice of the parametric quantum circuit** (Variational Ansatz)

Variational Quantum Eigensolver (VQE)

Quantum algorithms for NISQ Devices

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Objective: finding the ground state energy of molecules



Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

In chemistry, the **minimum eigenvalue of the Hamiltonian** characterizing the molecule is the **ground state energy of that system**

$$\mathcal{H} = -\sum_I \frac{\nabla^2_{R_I}}{M_I} - \sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}$$

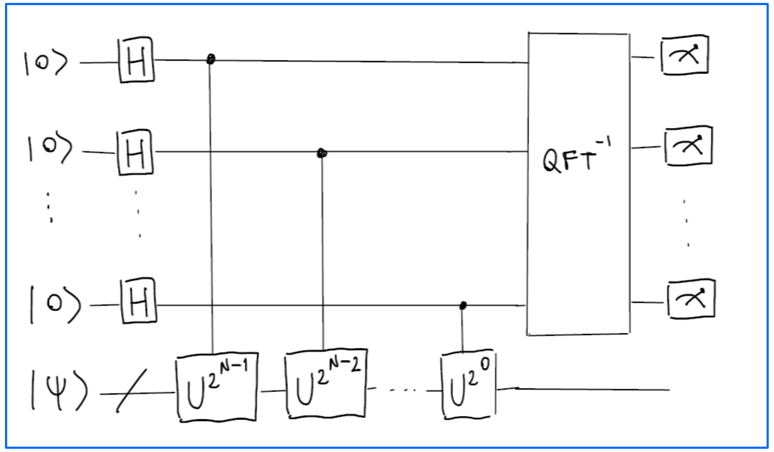
Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

In chemistry, the **minimum eigenvalue of the Hamiltonian** characterizing the molecule is the **ground state energy of that system**

$$\mathcal{H} = -\sum_I \frac{\nabla^2_{R_I}}{M_I} - \sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}$$

In the future, the **quantum phase estimation** algorithm may be **used to find the minimum eigenvalue**

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$$



Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

In chemistry, the **minimum eigenvalue of the Hamiltonian** characterizing the molecule is the **ground state energy of that system**

$$\mathcal{H} = -\sum_I \frac{\nabla^2_{R_I}}{M_I} - \sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}$$

In the future, the **quantum phase estimation** algorithm may be **used to find the minimum eigenvalue** → **its implementation on useful problems requires circuit depths exceeding the limits of hardware available in the NISQ era**

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

In chemistry, the **minimum eigenvalue of the Hamiltonian** characterizing the molecule is the **ground state energy of that system**

$$\mathcal{H} = -\sum_I \frac{\nabla^2_{R_I}}{M_I} - \sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}$$

In the future, the **quantum phase estimation** algorithm may be **used to find the minimum eigenvalue** ➡

VQE
(NISQ Ready)

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Molecular Hamiltonian is

$$\mathcal{H} = -\sum_I \frac{\nabla^2_{R_I}}{M_I} - \sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}$$

Because the **nuclei are much heavier than the electrons** they do **not move on the same time scale** and therefore, the behavior of **nuclei and electrons can be decoupled**

$$\mathcal{H}_{\mathrm{el}} = -\sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|}.$$

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Electronic Hamiltonian is

$$\mathcal{H}_{\mathrm{el}} = -\sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|}.$$

Applied to a **given state**, will return the **energy associated** to that state!

$$\mathcal{H}_{\mathrm{el}}|\Psi_n\rangle = E_n|\Psi_n\rangle$$

Minimum energy is associate to the **ground state** ➡ $E_0 = \frac{\langle\Psi_0|H_{\mathrm{el}}|\Psi_0\rangle}{\langle\Psi_0|\Psi_0\rangle}$

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Hartree-Fock (HF) method.

This method **approximates a N-body problem into N one-body problems** where each electron evolves in the **mean-field** of the others.

$$\mathcal{H}_{\mathrm{el}} = -\sum_i \frac{\nabla^2_{r_i}}{m_e} - \sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|}.$$

$$\Big\Downarrow \text{ HF}$$

$$\boxed{\hat{H}_{elec} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s}$$

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Hartree-Fock (HF) method.

This method **approximates a N-body problem into N one-body problems** where each electron evolves in the **mean-field** of the others.

$$\hat{H}_{elec} = \sum_{pq} \boxed{h_{pq}} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} \boxed{h_{pqrs}} \boxed{\hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s}$$

Creation and annihilation operators

$$h_{pq} = \int \phi_p^*(r) \left(-\frac{1}{2}\nabla^2 - \sum_I \frac{Z_I}{R_I - r} \right) \phi_q(r)$$

$$h_{pqrs} = \int \frac{\phi_p^*(r_1)\phi_q^*(r_2)\phi_r(r_2)\phi_s(r_1)}{|r_1 - r_2|}.$$

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

The **electronic wave function is represented in the occupation number basis**. For **M *spin* molecular orbitals**, the elements of this basis are labelled

$$|n_0, n_1, \ldots, n_{M-1}\rangle$$

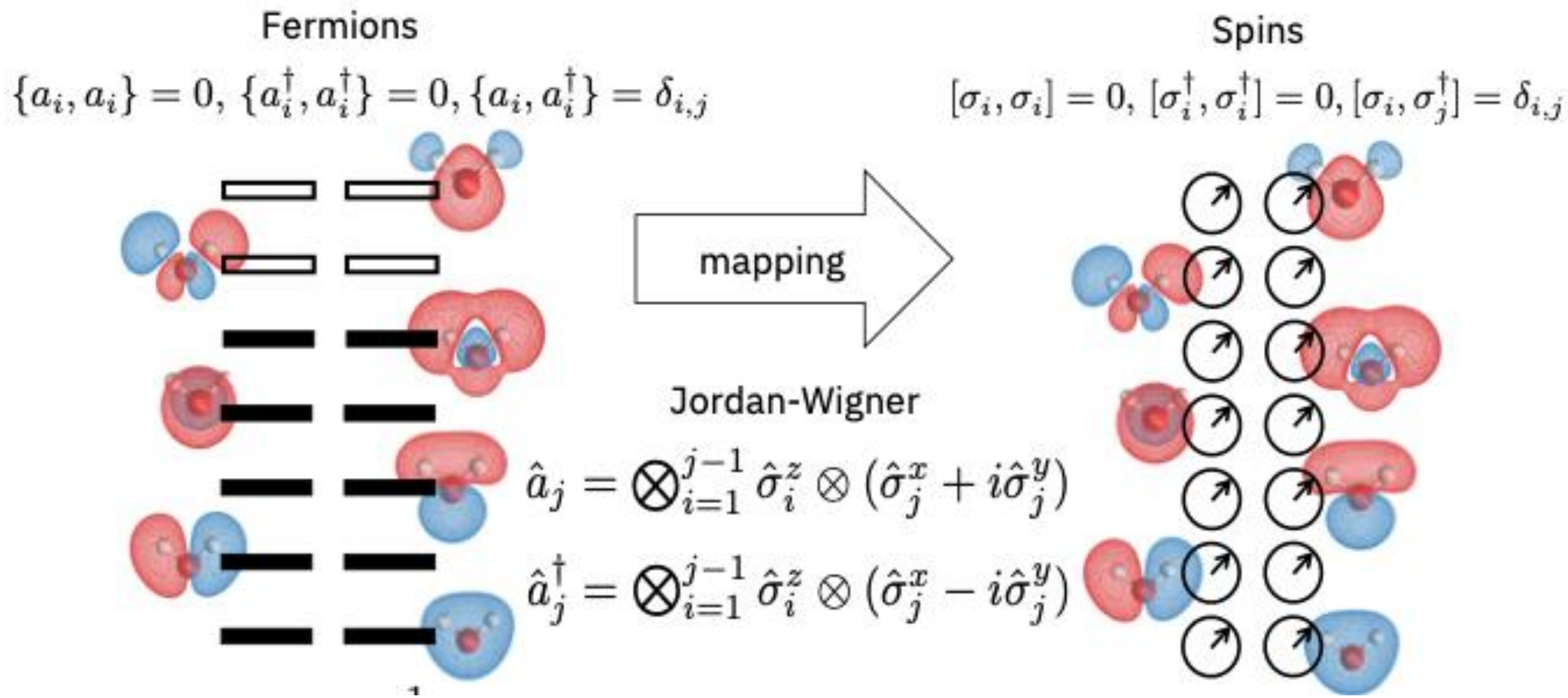
where **n_i=0,1 indicates the occupation of each orbital**. Thus **Spin Orbitals can contain one electron or be unoccupied.**

$$|n_0, n_1, \ldots, n_{M-1}\rangle \rightarrow |q_0 q_1 \cdots q_{M-1}\rangle.$$

We can use the **states of M qubits to encode any element of the occupation number basis**

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

We need to **map the fermionic operators onto operators that act on qubits.**
Done via **Jordan-Wigner transformation**



Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

Jordan-Wigner transformation **decompose the fermionic Hamiltonian into a linear combination of the tensor product of Pauli operators**

$$\hat{H}_{elec} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$$

Jordan-Wigner

$$\hat{a}_j = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x + i\hat{\sigma}_j^y)$$

$$\hat{a}_j^\dagger = \bigotimes_{i=1}^{j-1} \hat{\sigma}_i^z \otimes (\hat{\sigma}_j^x - i\hat{\sigma}_j^y)$$

$$\boxed{\hat{H}_{elec} = \sum_i c_i \hat{P}_i}$$

Variational Quantum Eigensolver (VQE) – Variational Principle

Hamiltonian can be written as follows, where each **λi is the eigenvalue corresponding to the eigenvector |ψi⟩**

$$H = \sum_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i|$$

the **expectation value** of the observable **H on an arbitrary quantum state** is given by

$$\langle H \rangle_\psi \equiv \langle \psi | H | \psi \rangle$$

Variational Quantum Eigensolver (VQE) – Variational Principle

$$\langle H \rangle_\psi = \langle \psi | H | \psi \rangle$$

Variational Quantum Eigensolver (VQE) – Variational Principle

$$\langle H \rangle_\psi = \langle \psi | H | \psi \rangle = \langle \psi | \left(\sum_{i=1}^{N} \lambda_i |\psi_i\rangle\langle\psi_i| \right) |\psi\rangle$$

$$= \sum_{i=1}^{N} \lambda_i \langle\psi|\psi_i\rangle\langle\psi_i|\psi\rangle \;\; = \sum_{i=1}^{N} \lambda_i |\langle\psi_i|\psi\rangle|^2$$

Variational Quantum Eigensolver (VQE) – Variational Principle

$$\langle H\rangle_\psi = \langle\psi|H|\psi\rangle = \langle\psi|\left(\sum_{i=1}^{N}\lambda_i|\psi_i\rangle\langle\psi_i|\right)|\psi\rangle$$

$$= \sum_{i=1}^{N}\lambda_i\langle\psi|\psi_i\rangle\langle\psi_i|\psi\rangle = \sum_{i=1}^{N}\lambda_i|\langle\psi_i|\psi\rangle|^2$$

each of **the weights in the linear combination is greater than or equal to 0**

$$\lambda_{min} \leq \langle H\rangle_\psi = \langle\psi|H|\psi\rangle = \sum_{i=1}^{N}\lambda_i|\langle\psi_i|\psi\rangle|^2$$

Variational Quantum Eigensolver (VQE) – Variational Principle

Formally stated, given a Hermitian matrix H with an unknown minimum eigenvalue λ_{min}, associated with the eigenstate $|\psi_{min}\rangle$, VQE provides an estimate λ_θ bounding λ_{min}:

$$\lambda_{min} \leq \lambda_\theta \equiv \langle\psi(\theta)|H|\psi(\theta)\rangle$$

where $|\psi(\theta)\rangle$ is the eigenstate associated with λ_θ. By applying a parameterized circuit, represented by $U(\theta)$, to some arbitrary starting state $|\psi\rangle$, the algorithm obtains an estimate $U(\theta)|\psi\rangle \equiv |\psi(\theta)\rangle$ on $|\psi_{min}\rangle$. The estimate is iteratively optimized by a classical controller changing the parameter θ minimizing the expectation value of $\langle\psi(\theta)|H|\psi(\theta)\rangle$.

Variational Quantum Eigensolver (VQE) – Ansatz

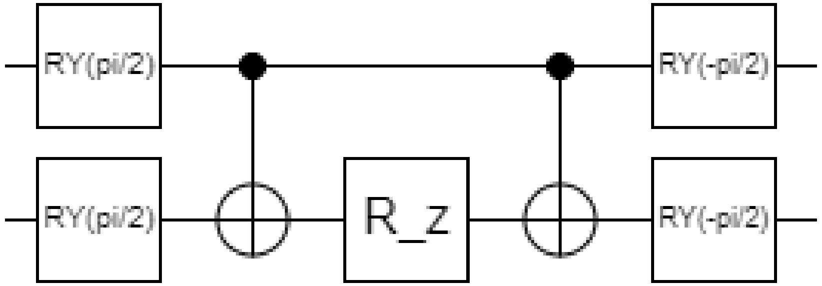
- **Ansatz** is a **provisional molecular ground state**

$$|\psi\rangle \;—\!\!\boxed{U(\theta,\phi,\lambda)}\!\!—\; U(\theta,\phi,\lambda)\,|\psi\rangle$$

- The **classic optimizer evaluates the suitability of candidate solution** based on its **energy**.

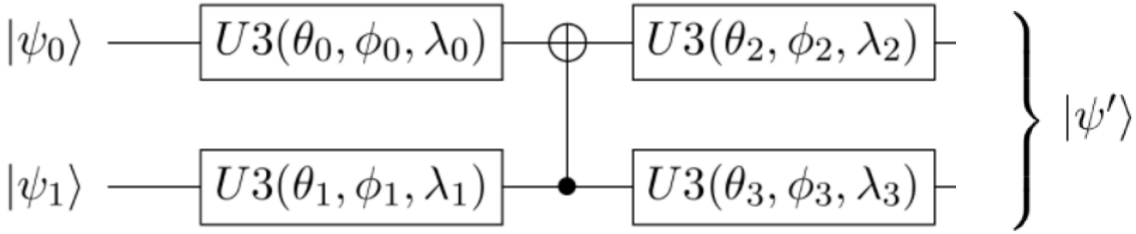
Variational Quantum Eigensolver (VQE) – Ansatz

VQE uses:



Chemical-inspired Ansatz, such as the **Unitary Coupled Cluster (UCC)** method

(may be harder to implement on real hardware)



or a **Hardware-efficient Ansatz**

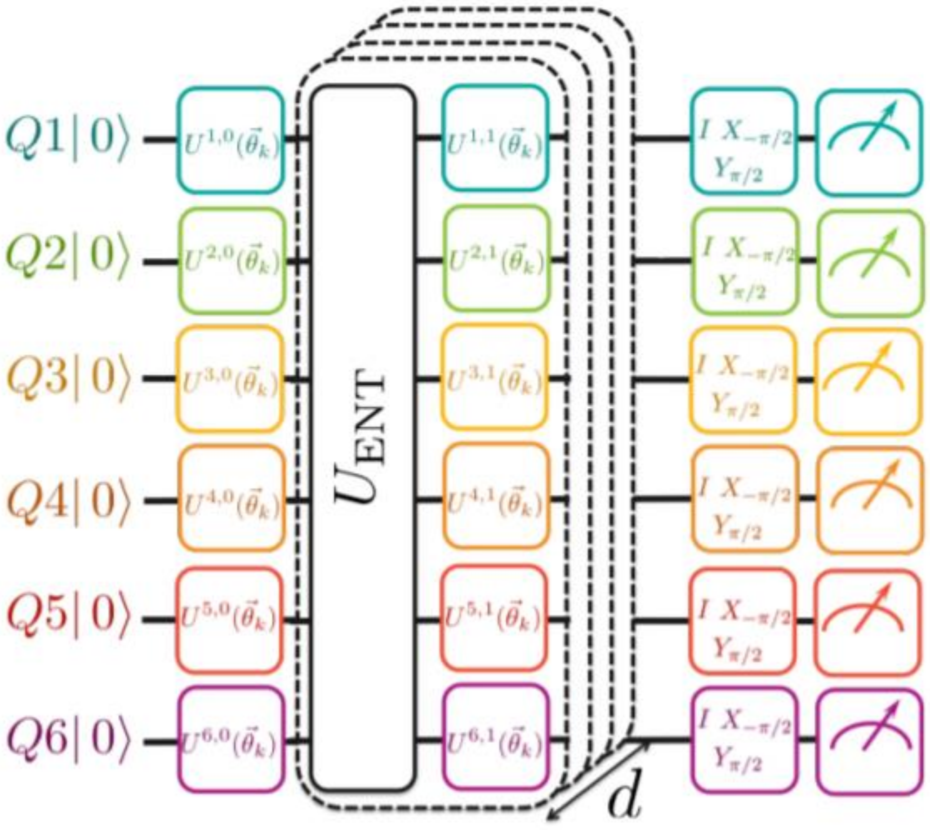
(easy to implement on hardware but lack of any physical meaning)

Quantum algorithms for NISQ Devices

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

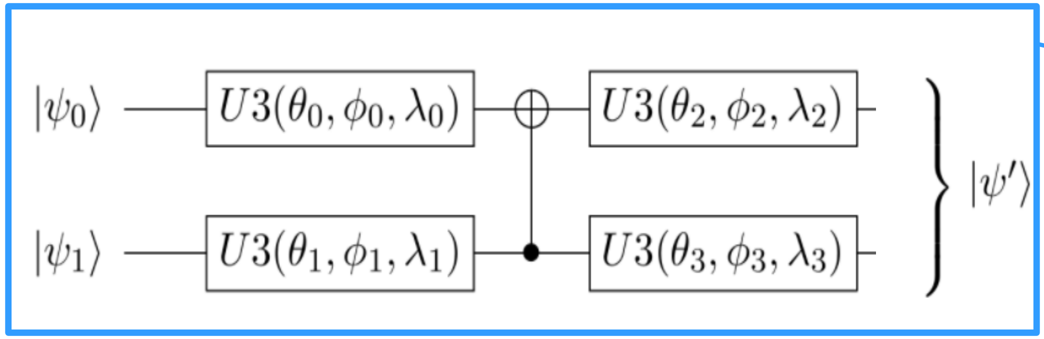
Hardware-efficient Ansatz

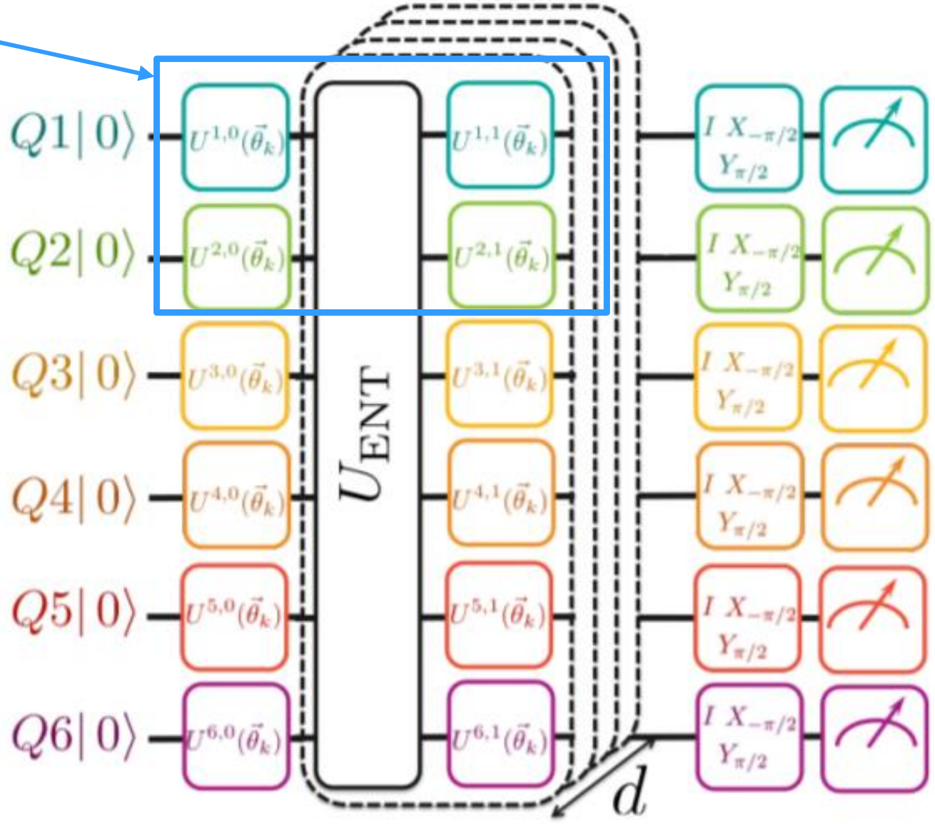
https://arxiv.org/abs/1704.05018



Quantum algorithms for NISQ Devices

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY



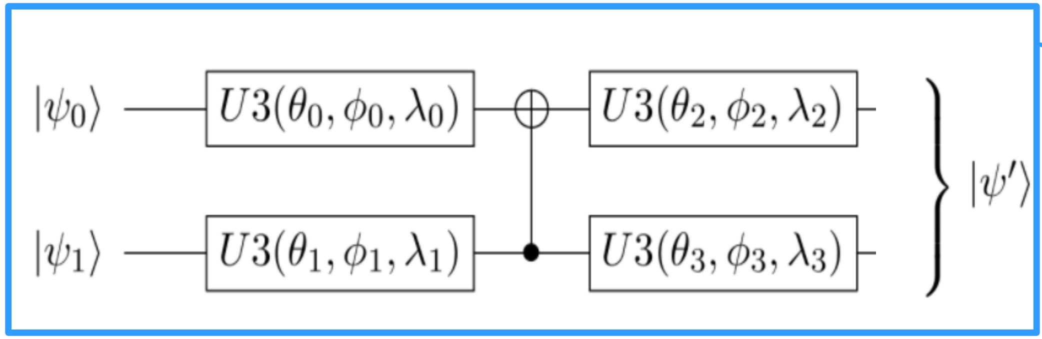


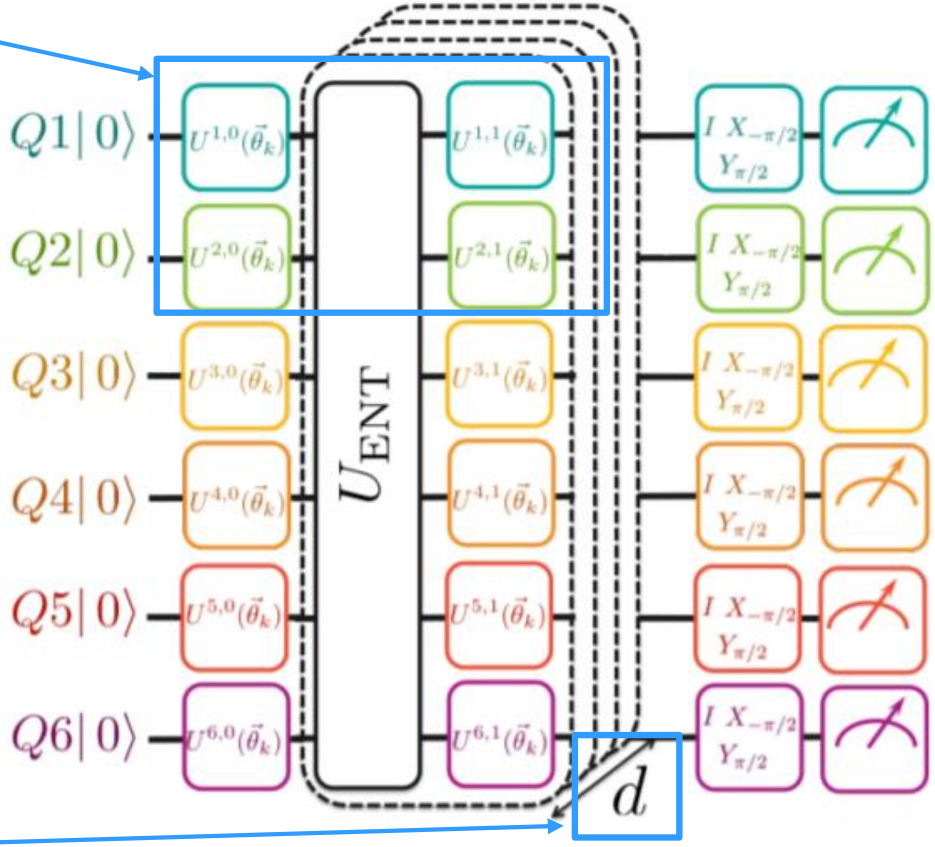
Hardware-efficient Ansatz

https://arxiv.org/abs/1704.05018

Quantum algorithms for NISQ Devices

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY





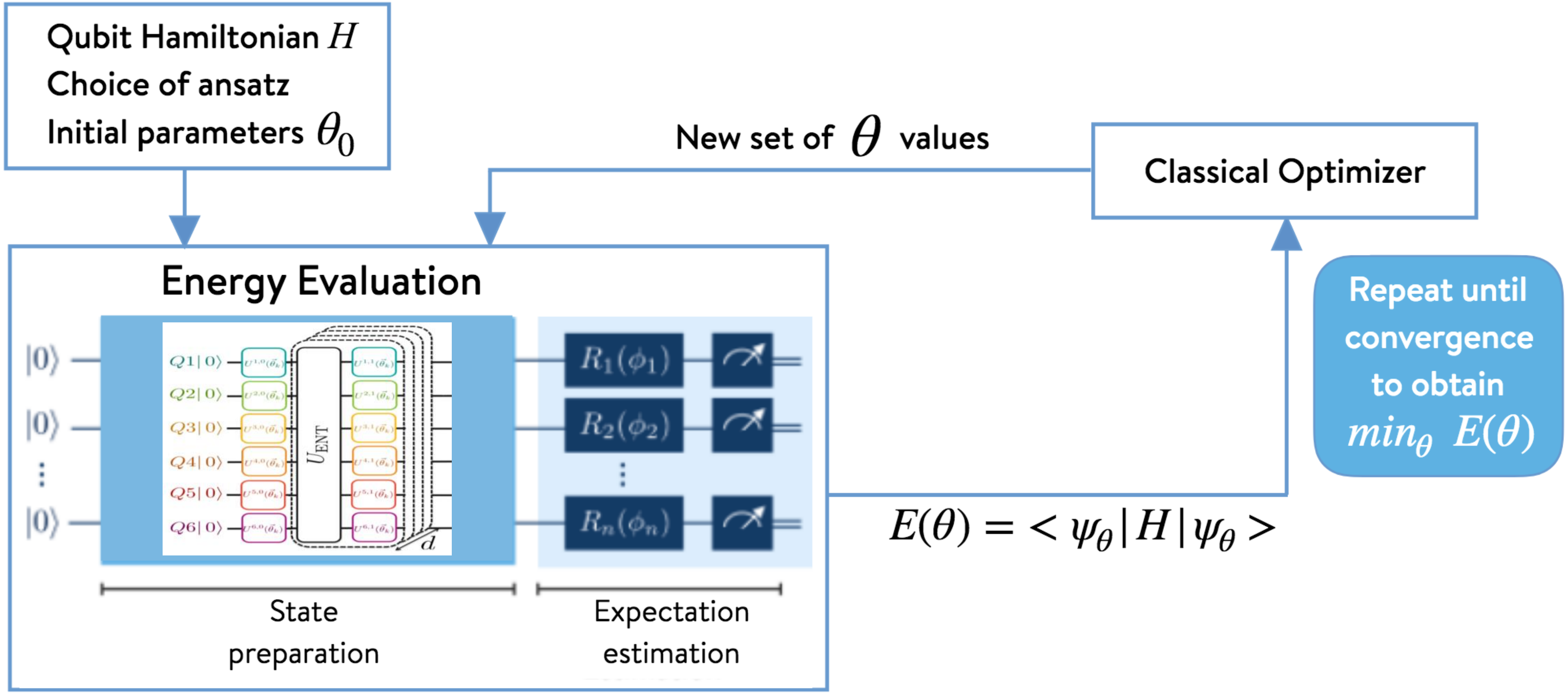
Hardware-efficient Ansatz

https://arxiv.org/abs/1704.05018

Repeated multiple times in **d layers**

Quantum algorithms for NISQ Devices

Variational Quantum Eigensolver (VQE) – QUANTUM CHEMISTRY

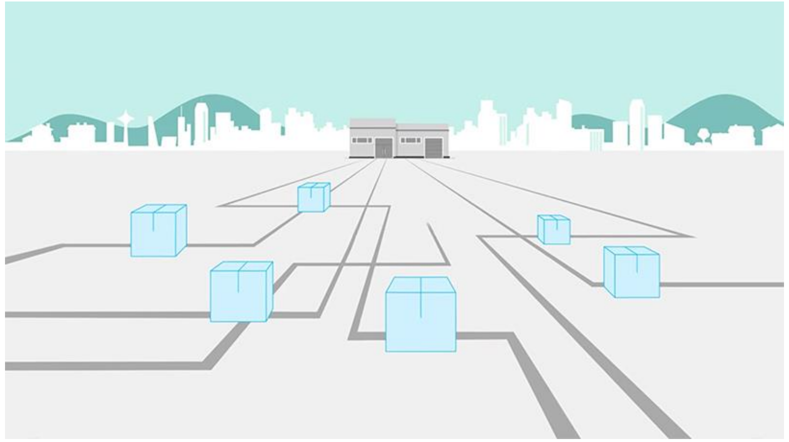


Quantum Approximate Optimization Algorithm (QAOA)

Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

Optimization Problems



Routing

Scheduling

Portfolio Optimization

Quantum algorithms for NISQ Devices

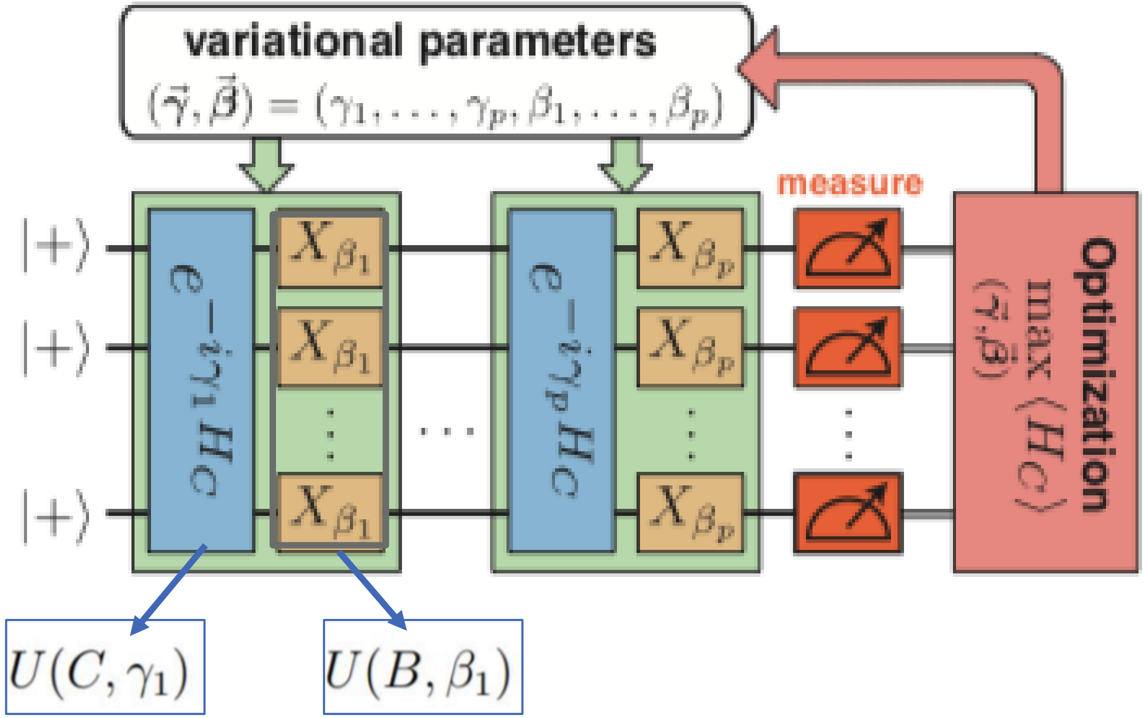
Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

<u>Objective</u>: to **solve a combinatorial optimization problem**

<u>Method</u>: **Ansatz encodes two alternating circuits, U(C) and U(B)**, each **parameterized** by a number, **γ and β**.

Ideally, the circuit provides the **solution** **|γ,β>** to a **combinatorial problem implicit in** the definition of **U(C)**.

$$|\gamma, \beta\rangle = U(B,\beta_p)\,U(C,\gamma_p)\cdots U(B,\beta_1)\,U(C,\gamma_1)\;|s\rangle$$



Quantum algorithms for NISQ Devices

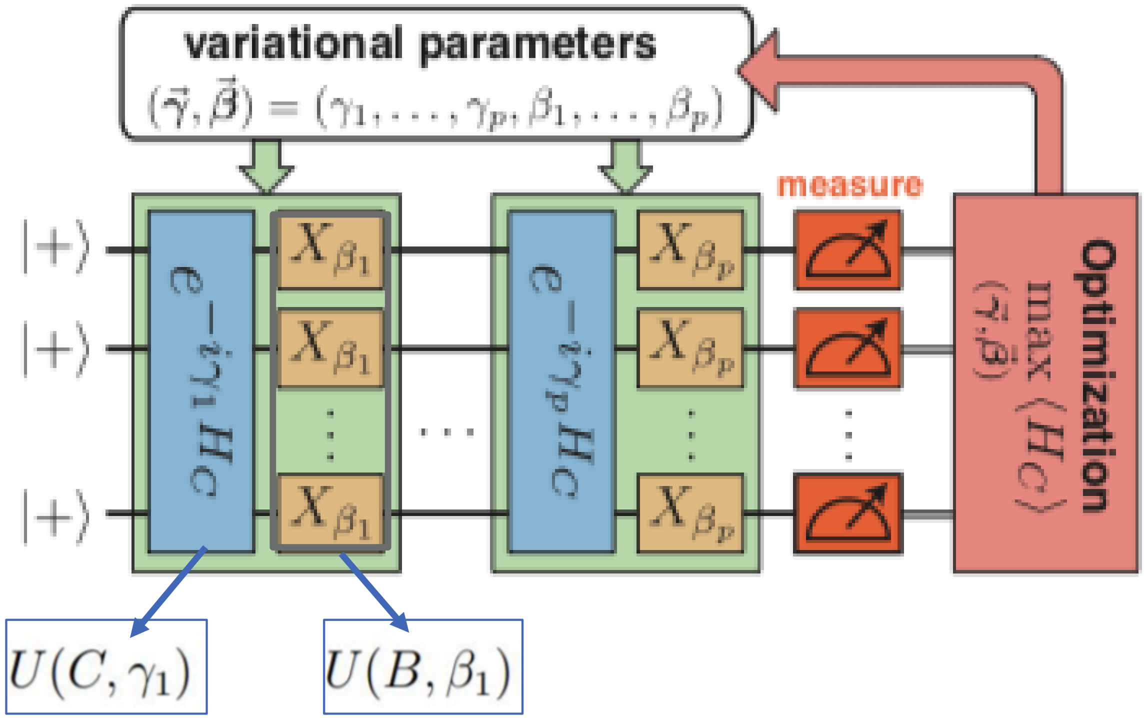
Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

It is a heuristic optimization algorithm

$$\underbrace{|\gamma, \beta\rangle}_{\text{solution}} = \underbrace{U(B,\beta_p)\,U(C,\gamma_p)\cdots U(B,\beta_1)\,U(C,\gamma_1)}_{\text{Circuit (alternating circuits)}}\ \underbrace{|s\rangle}_{\text{initial state}}$$

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x}$$



Quantum algorithms for NISQ Devices

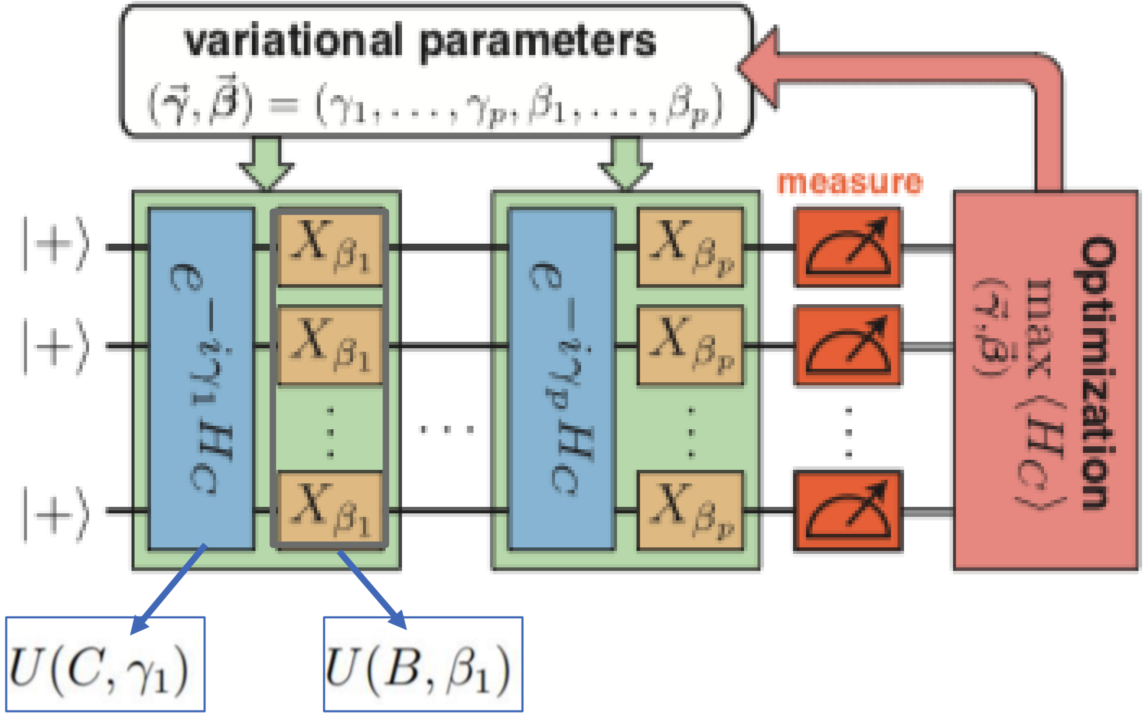
Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

Encodes the optimization problem to solve (e.g. C could be some Qubo problem)

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$

Allow the **exploration of the solution space**

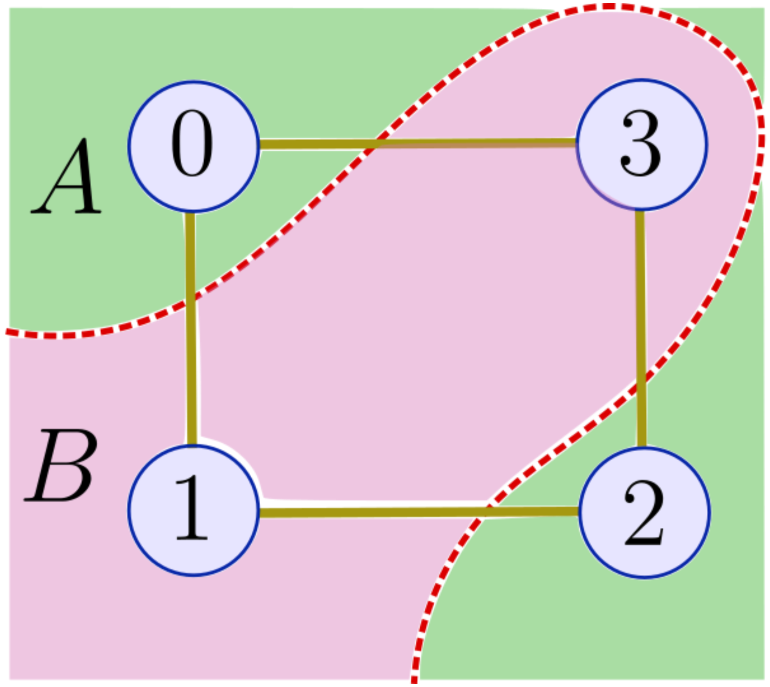


https://arxiv.org/abs/1411.4028

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

The MaxCut problem

The aim of MaxCut is to maximize the number of edges in a graph that are “cut” by a given partition of the vertices into two sets



Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

Consider a **graph with m edges and n vertices**. We seek the **partition z** of the **vertices into two sets A and B** which **maximizes**

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z), \qquad C_\alpha = \frac{1}{2}\left(1 - \sigma_z^j \sigma_z^k\right)$$

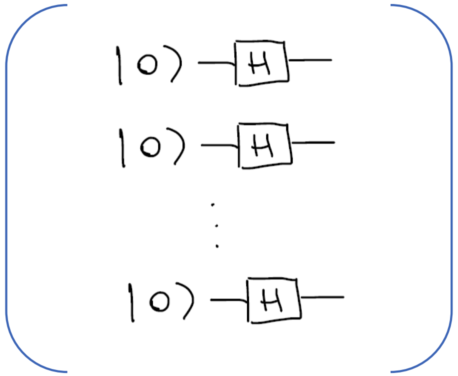
$$\begin{cases} C_\alpha(z) = 1 & \text{When } \textbf{vertex j belong to set A} \text{ and } \textbf{vertex k to set B or viceversa} \\ C_\alpha(z) = 0 & \text{otherwise} \end{cases}$$

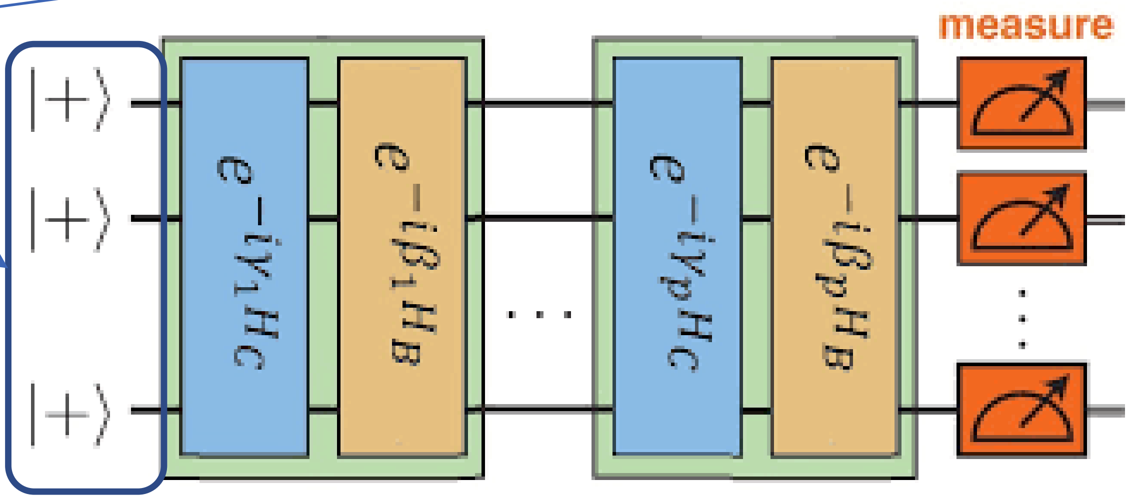
Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$|\gamma, \beta\rangle = U_{B_p} U_{C_p} U_{B_{p-1}} U_{C_{p-1}} \dots U_{B_1} U_{C_1} \boxed{|+_n\rangle}$$

$$\boxed{|+_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle}$$



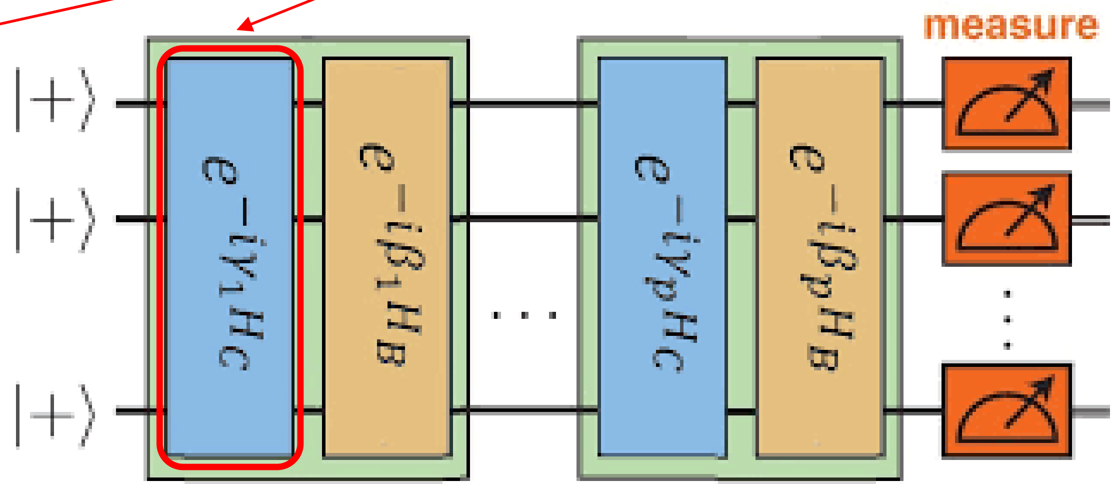


Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle = U_{B_p}U_{C_p}U_{B_{p-1}}U_{C_{p-1}}\ldots U_{B_1}\boxed{U_{C_1}}|+_n\rangle$$

$$\boxed{U_{C_l} = e^{-i\gamma_l C} = \prod_{\text{edge (j,k)}} e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}}$$

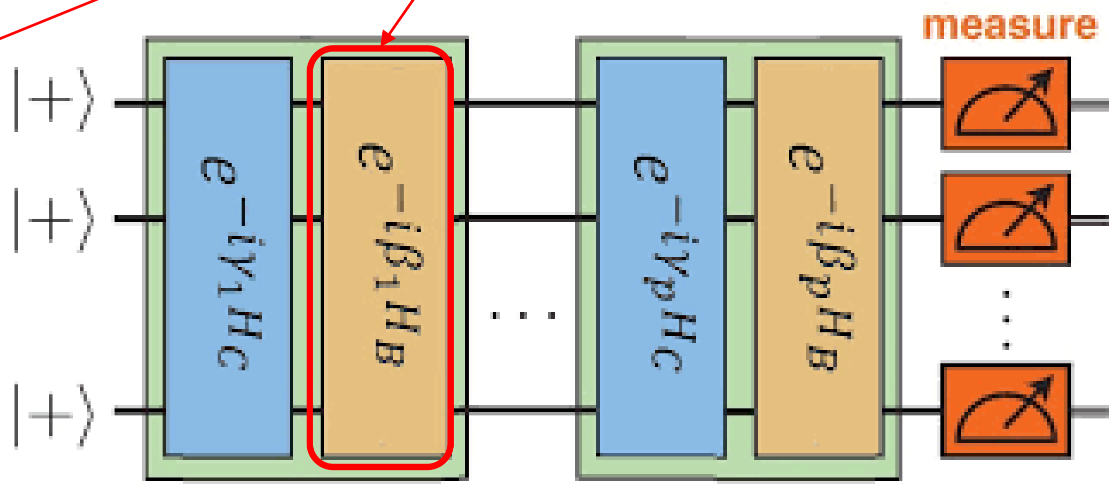


Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U_{B_p} U_{C_p} U_{B_{p-1}} U_{C_{p-1}} \cdots \boxed{U_{B_1}} U_{C_1} |+_n\rangle$$

$$\boxed{U_{B_l} = e^{-i\beta_l B} = \prod_{j=1}^{n} e^{-i\beta_l \sigma_x^j},}$$



Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U_{C_l} = e^{-i\gamma_l C} = \prod_{\text{edge (j,k)}} e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}$$

$$U_{B_l} = e^{-i\beta_l B} = \prod_{j=1}^{n} e^{-i\beta_l \sigma_x^j},$$

How can we represent these unitaries by means of quantum gates?

Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U_{B_l} = e^{-i\beta_l B} = \prod_{j=1}^{n} \boxed{e^{-i\beta_l \sigma_x^j}},$$

$$e^{-i\beta_l \sigma_x^j}$$

Recall that..

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$R_x(\theta) \equiv e^{-i\frac{\theta}{2}X} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

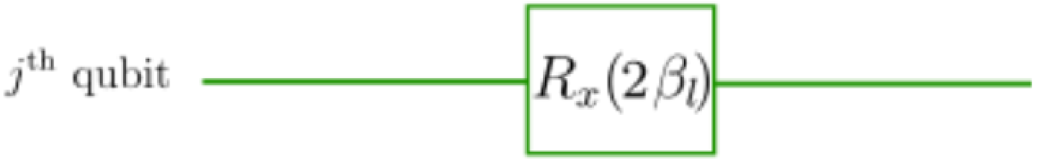
$$U_{B_l} = e^{-i\beta_l B} = \prod_{j=1}^{n} \boxed{e^{-i\beta_l \sigma_x^j}},$$

Recall that..

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$R_x(\theta) \equiv e^{-i\frac{\theta}{2}X} = \cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

$$e^{-i\beta_l \sigma_x^j}$$ =



Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U_{C_l} = e^{-i\gamma_l C} = \prod_{\text{edge (j,k)}} \boxed{e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}}$$

$$e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}$$

Recall that..

$$R_z(\theta) \equiv e^{-i\frac{\theta}{2}Z} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

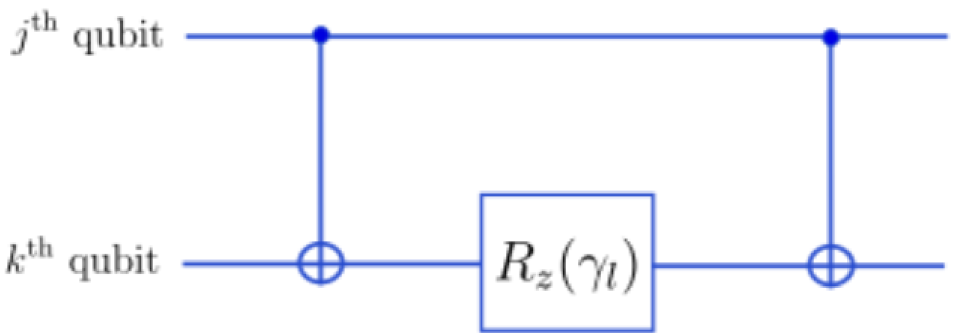
$$U_{C_l} = e^{-i\gamma_l C} = \prod_{\text{edge (j,k)}} \boxed{e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}}$$

Recall that..

$$R_z(\theta) \equiv e^{-i\frac{\theta}{2}Z} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

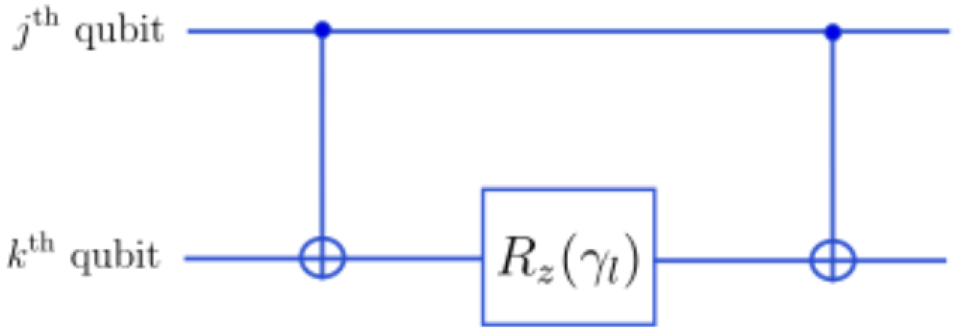
$$e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2} =$$



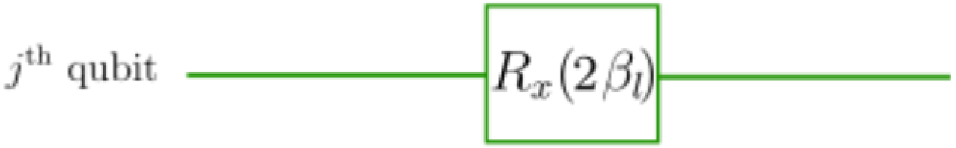
Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

$$U_{C_l} = e^{-i\gamma_l C} = \prod_{\text{edge (j,k)}} e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}$$



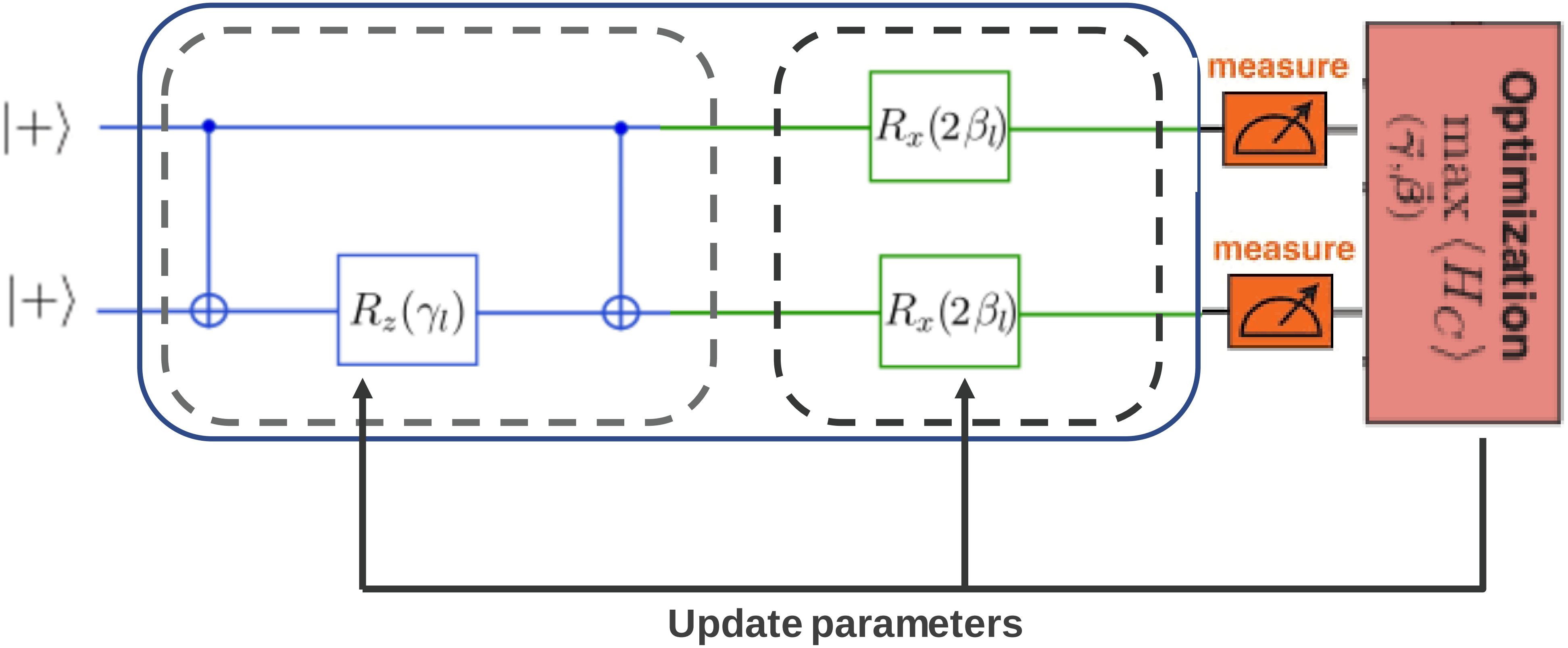
$$U_{B_l} = e^{-i\beta_l B} = \prod_{j=1}^{n} e^{-i\beta_l\sigma_x^j},$$



Quantum algorithms for NISQ Devices

Quantum Approximate Optimization Algorithm (QAOA) – QUANTUM OPTIMIZATION

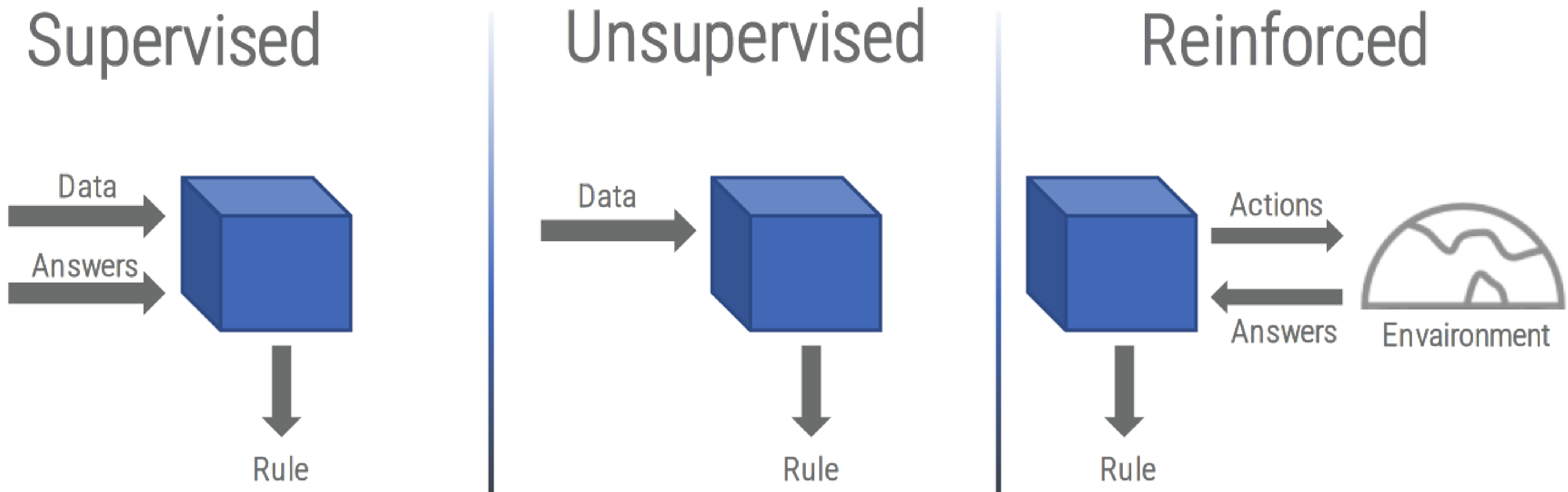
QAOA for The MaxCut problem (simple case with 2 qubits)



Quantum Machine Learning (QML)

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Classical ML



Labeled data

Correlations

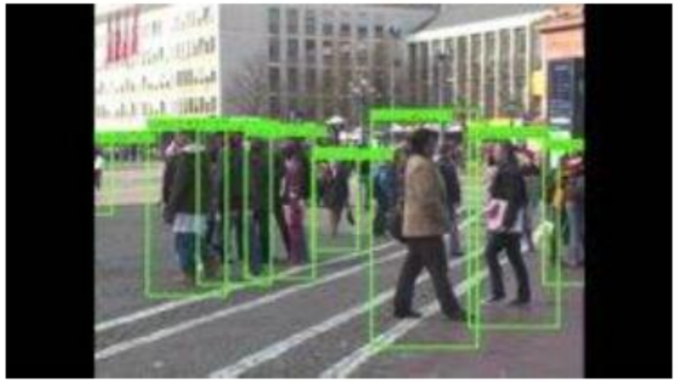
Strategy

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Classical ML

Supervised

- Classification
- Regression



Unsupervised

- Clustering
- Data Generation

Reinforced

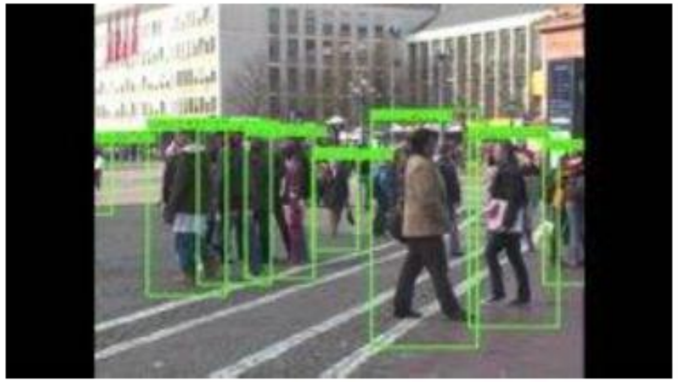
- Game theory
- Robotics

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Classical ML

Supervised

- Classification
- Regression



Unsupervised

- Clustering
- Data Generation

Reinforced

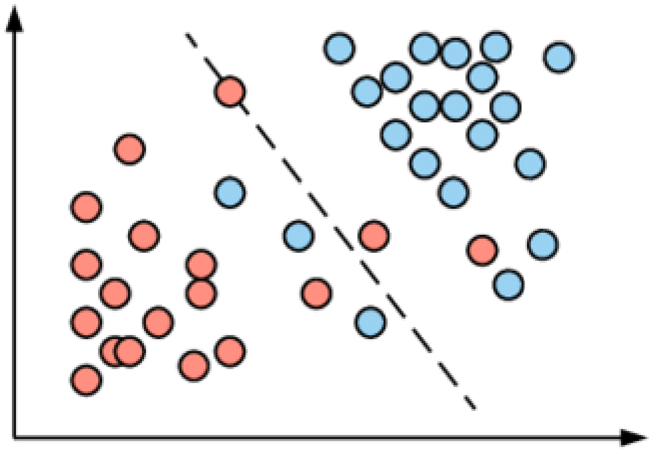
- Game theory
- Robotics

Quantum Machine Learning (QML) – Classification

It is given an **ensemble of labelled data points** usually called **training set**.

The learning algorithm has the task **to induce a classifier from these labelled samples**.

The **classifier is a function that allocate labels to inputs**, including those that are out of the training set which have never been previously analyzed by the algorithm.



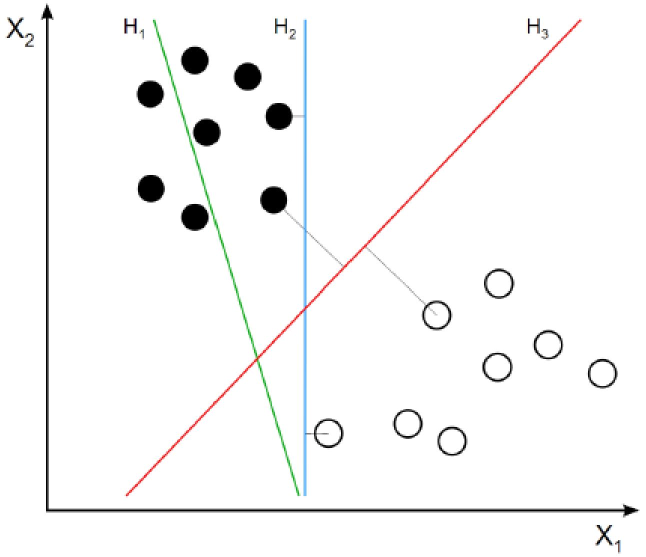
Supervised learning

- **Classification**
- **Regression**

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – SVM

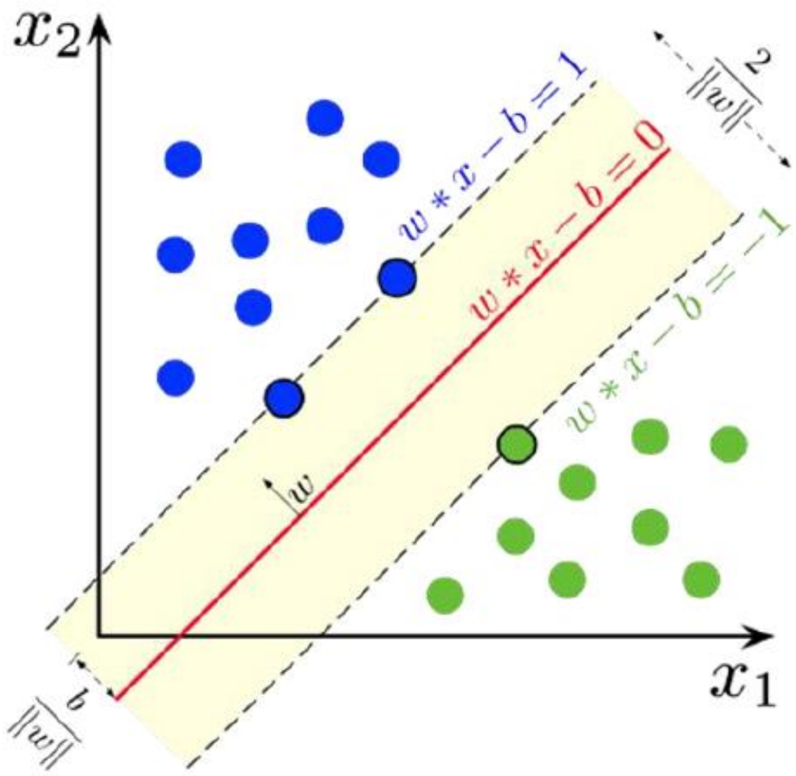
The task is to find a **hyperplane** that is the best discrimination between two class regions and serves as a decision boundary for future classification tasks.



input: M training vectors $\{(\vec{x}, y) \mid \vec{x} \in \mathbb{R}^N,\ y \in \{-1, +1\}\}$

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – SVM



Objective:

minimize $\quad \|\boldsymbol{w}\|$

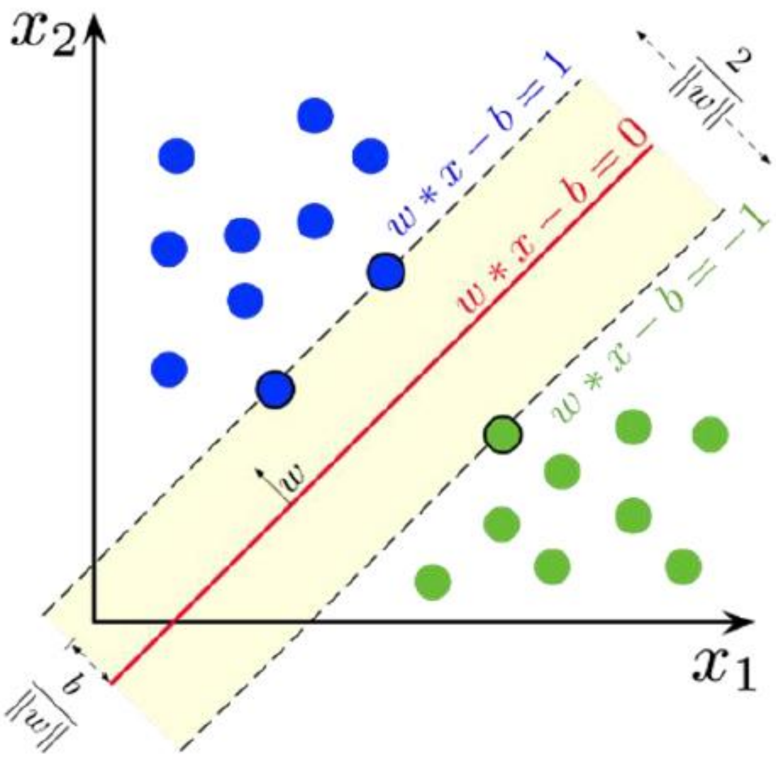
Constraints

$$\boldsymbol{x}_i \cdot \boldsymbol{w} - b \leq -1 \quad \text{for} \quad y_i = -1$$

$$\boldsymbol{x}_i \cdot \boldsymbol{w} - b \geq 1 \quad \text{for} \quad y_i = 1$$

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – SVM

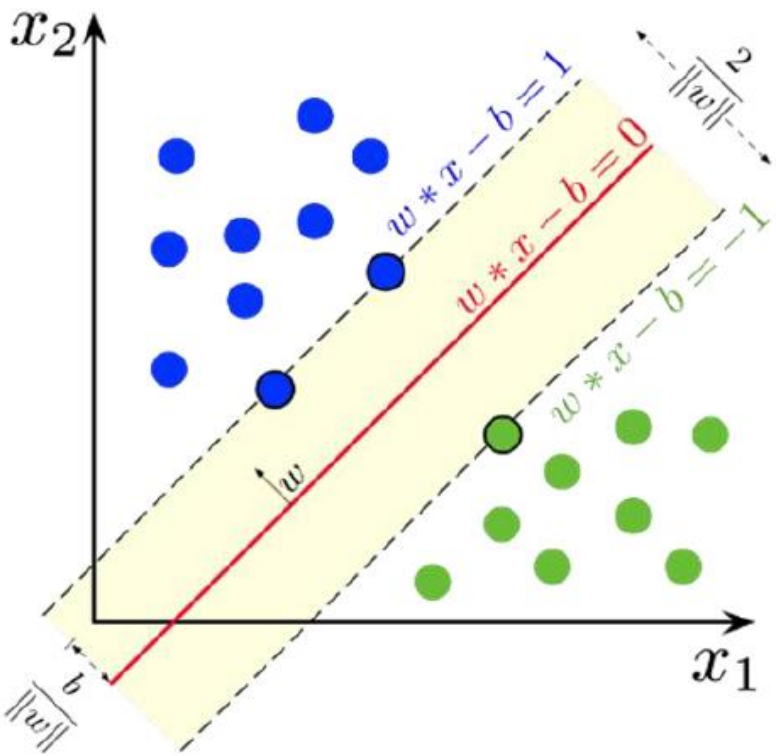


Can be seen as an optimization problem

$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\boldsymbol{x}_i \cdot \boldsymbol{w} + b \right) - 1 \right]$$

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i$$

Quantum Machine Learning (QML) – SVM



Can be seen as an optimization problem

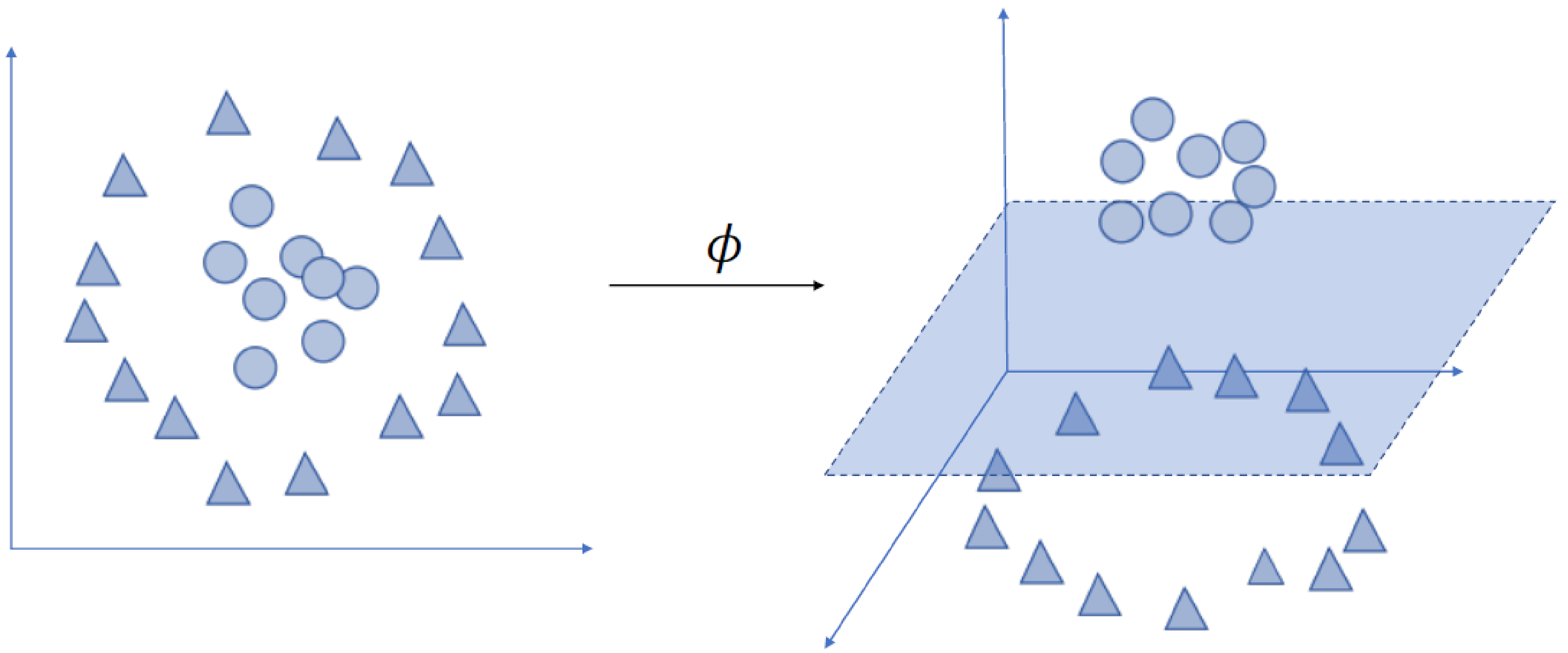
$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\boldsymbol{x}_i \cdot \boldsymbol{w} + b \right) - 1 \right]$$

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i$$

$$\boxed{L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j}$$

Quantum Machine Learning (QML) – SVM

Kernel methods [115] are classification algorithms in ML that use a kernel function K in order to map data points, living in the input space V, to a higher dimensional feature space V', where separability between classes of data becomes clearer. Kernel methods avoid the calculation of points in the new coordinates but rather perform the so called kernel trick that allow to work in the feature space V' simply computing the kernel of pairs of data points [115] (see. Fig.1.2).

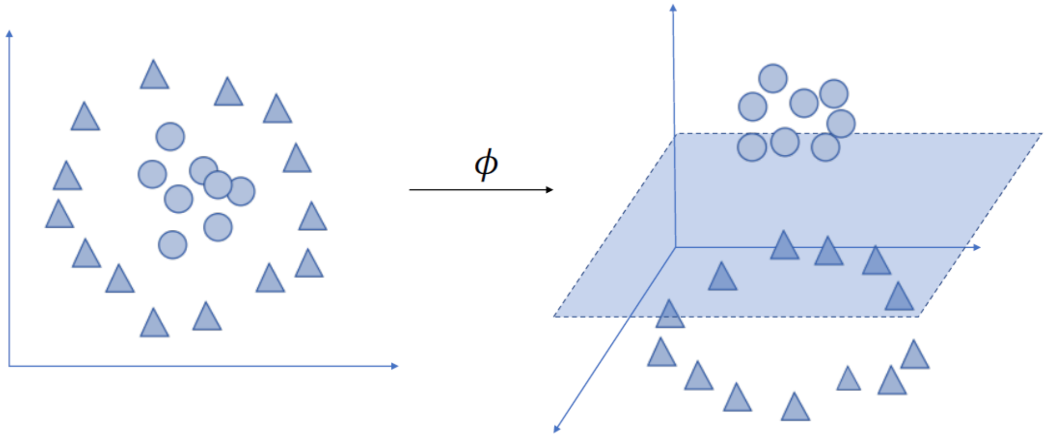


Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – SVM

Input space

$$\boldsymbol{x}_i = (x_{0,i}, x_{1,i}, \ldots)$$



Feature space

$$\boldsymbol{\varphi}(\boldsymbol{x}_i) = \left(x_{0,i}, x_{1,i}, \ldots, \phi\left(x_{0,i}, x_{1,i}, \ldots\right)\right)$$

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

ϕ

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boxed{\boldsymbol{\varphi}(\boldsymbol{x}_i) \cdot \boldsymbol{\varphi}(\boldsymbol{x}_j)}$$

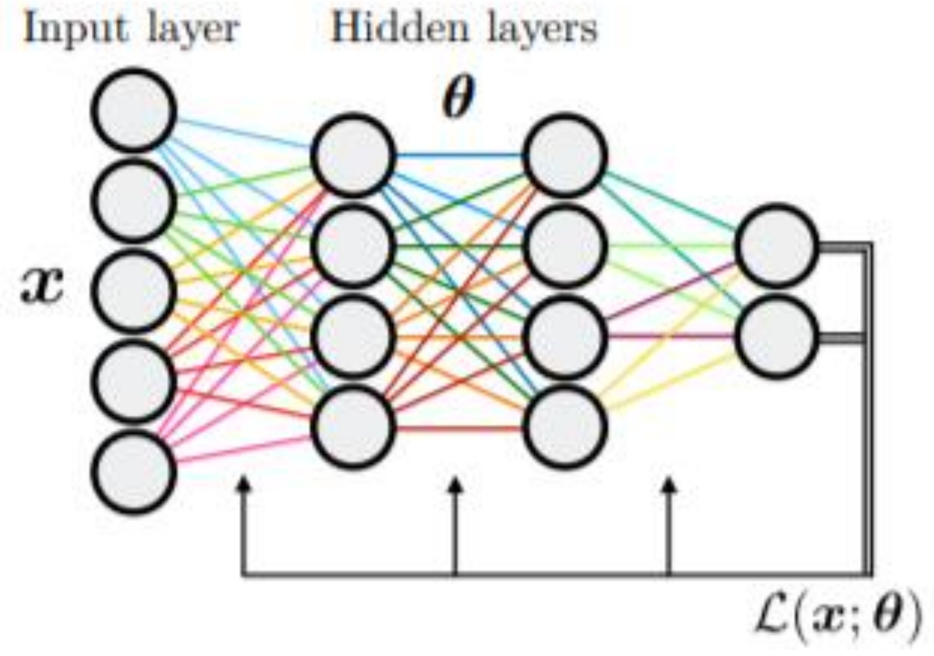
$$\boxed{K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i) \cdot \boldsymbol{\varphi}(\boldsymbol{x}_j)}$$

Kernel function

Quantum Machine Learning (QML) – Neural networks

the algorithm is asked to **reproduce the relations** between some inputs and outputs.

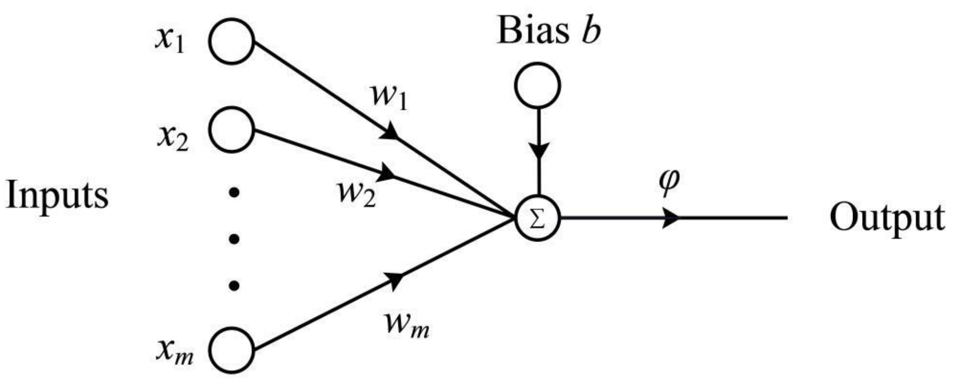
If properly trained, the **NN is able to classify new data**, i.e. data that was not used during training



Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Neural networks

Each **neuron**

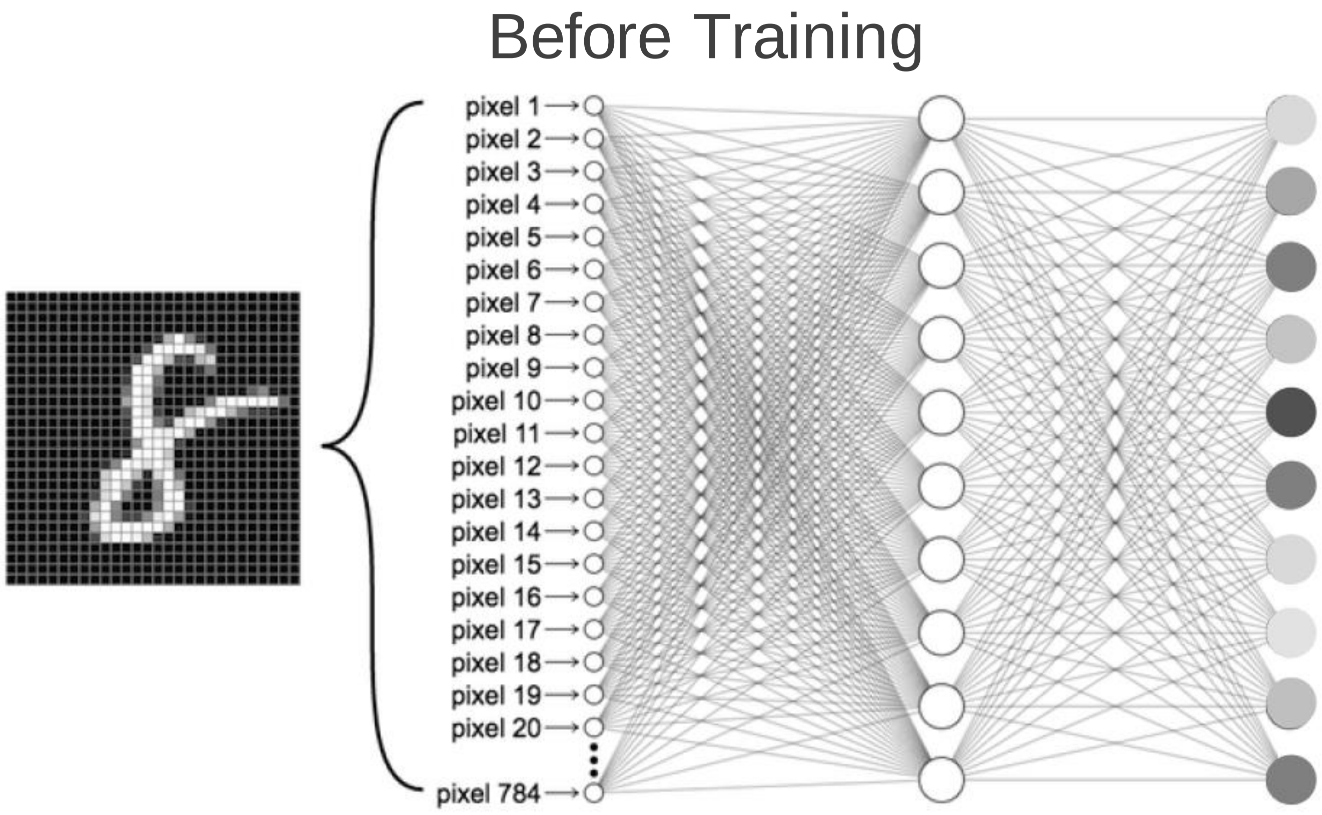


Computes a **non linear function**

$$F(\boldsymbol{x}) = \varphi\left(\sum_{i=0}^{m} x_i w_i + b\right) = \begin{cases} 1, & \text{if } \sum_{i=0}^{m} x_i w_i + b > 0 \\ 0, & \text{otherwise} \end{cases}$$

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Neural networks



LOSS function

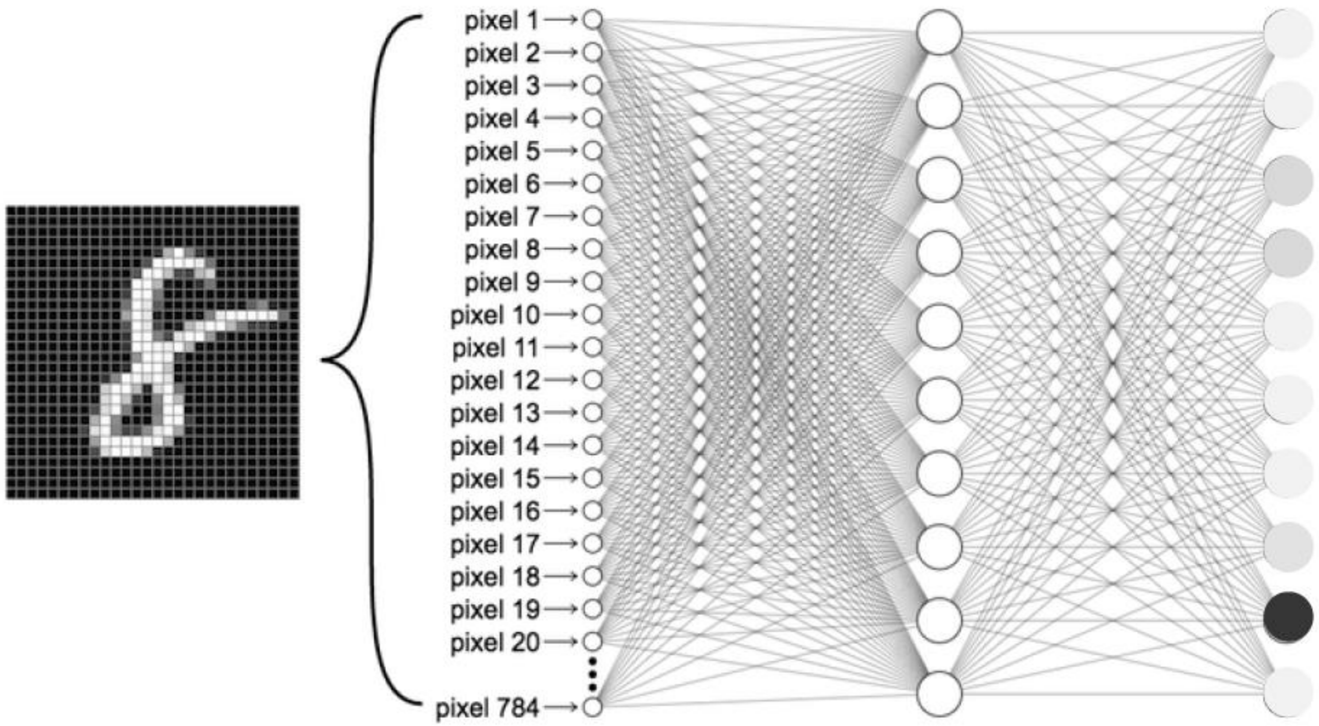
The **distance between the neural network predictions and the labels of the training set**

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2$$

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Neural networks

After Training



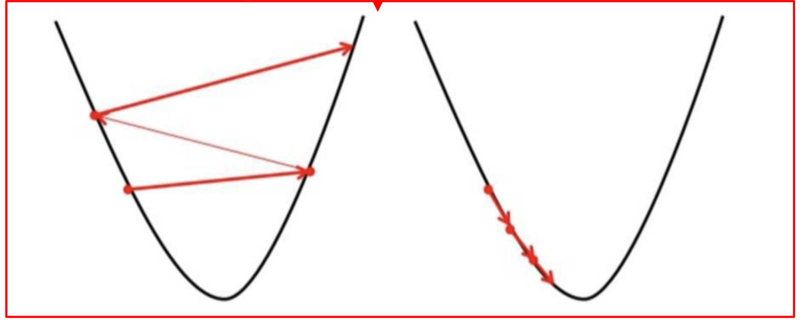
Training

Parameters are updated using **gradient descent** and **backpropagation**

$$\theta \leftarrow \theta - \boxed{\alpha} \boxed{\nabla \mathcal{L}(\theta)}$$

Gradient

Learning rate



Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML)

QISKIT

Most used quantum SDK with access of quantum devices

Machine learning modules

PENNYLANE

For hybrid quantum-classical computation

Integration of pytorch and tensorflow with different quantum SDK

TF QUANTUM

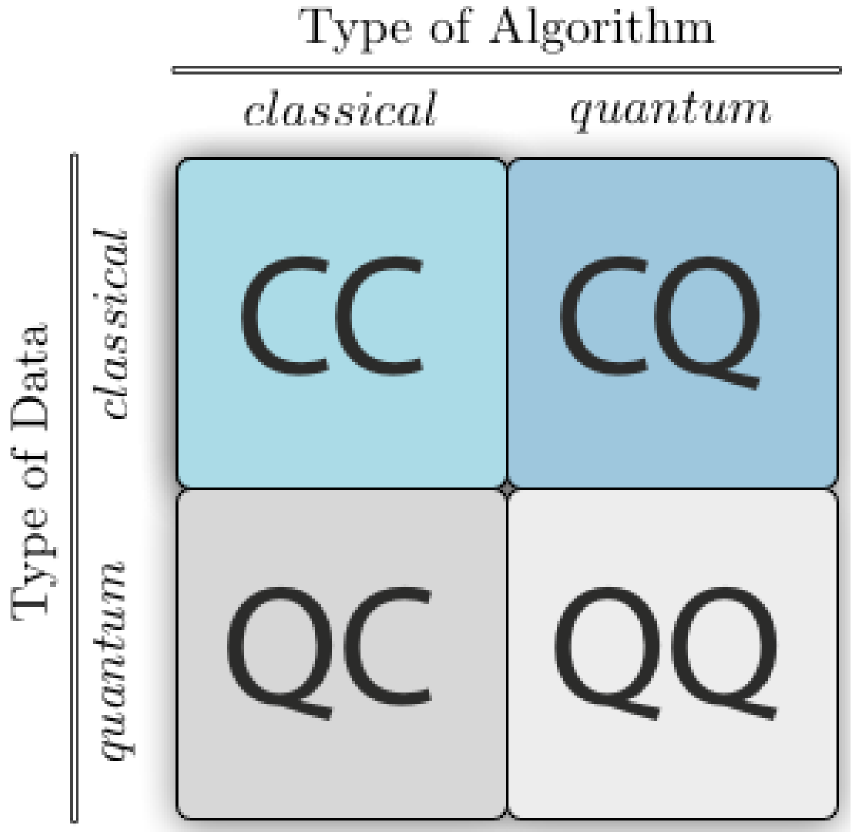
= Cirq + TensorFlow

For hybrid quantum-classical computation

Equivalence between quantum and tensor operations

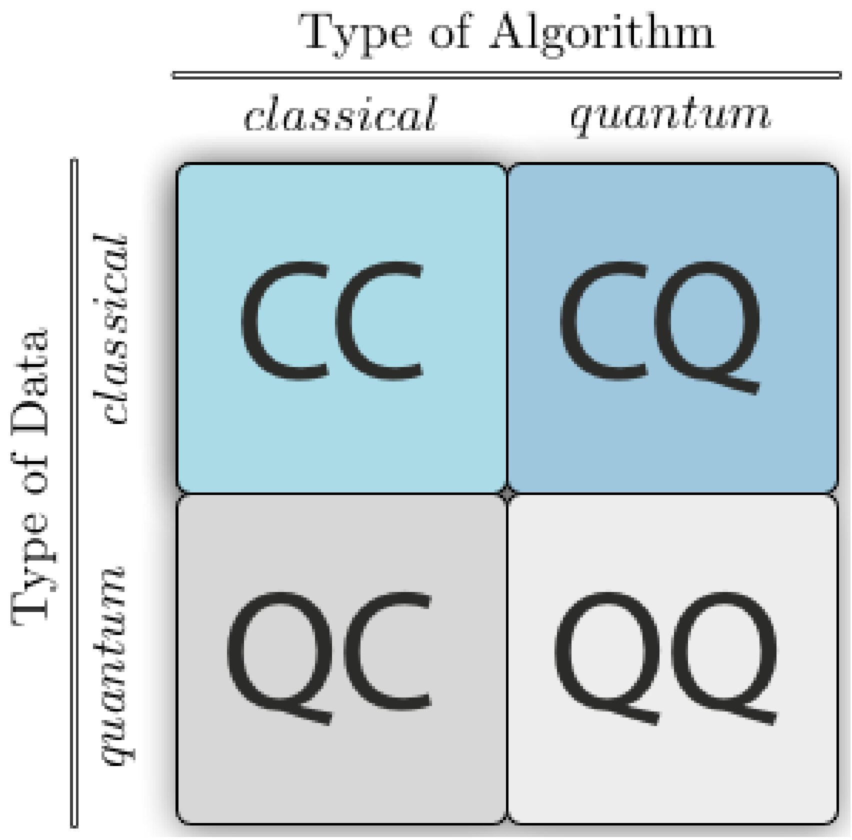
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML)



Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML)



First generation of QML

- Accelerated linear algebra on quantum computers
- Only applies to fault-tolerant quantum computers
- Assumed to have a QRAM to encode Data

Second generation of QML

- Low-depth quantum circuit learning
- Applicable to NISQ devices
- No QRAM but encoding Data is generally hard

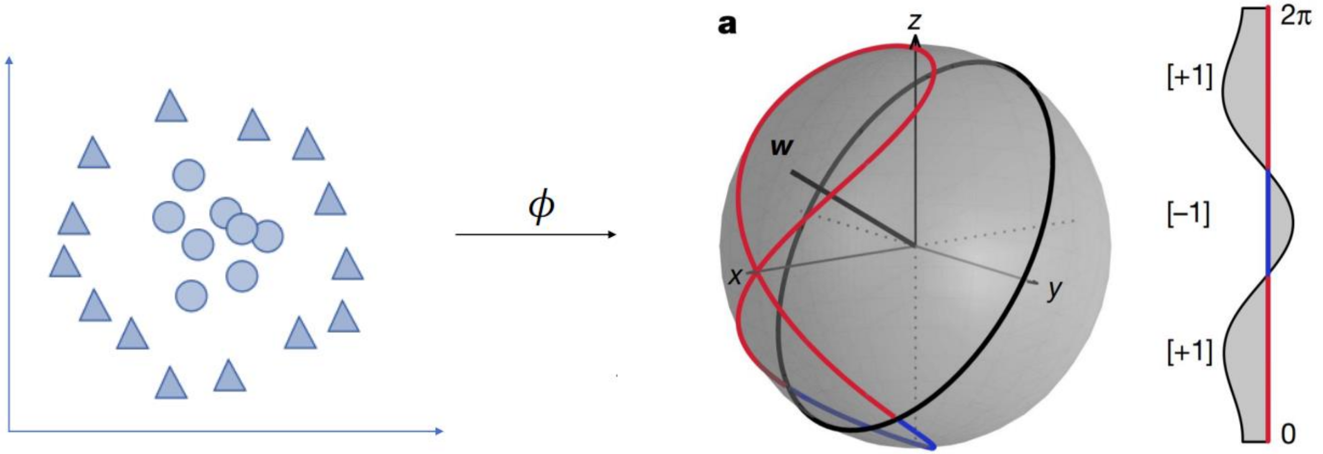
Quantum Support Vector Machine (QSVM)

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

Quantum Feature map
(maps classical vector into a quantum state)

$$\vec{x} \mapsto |\Phi(\vec{x})\rangle = \mathcal{U}_{\Phi(\vec{x})}|0\rangle^{\otimes n}$$



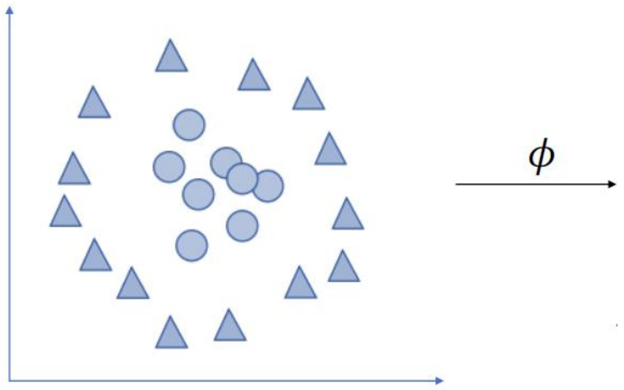
https://arxiv.org/abs/1804.11326

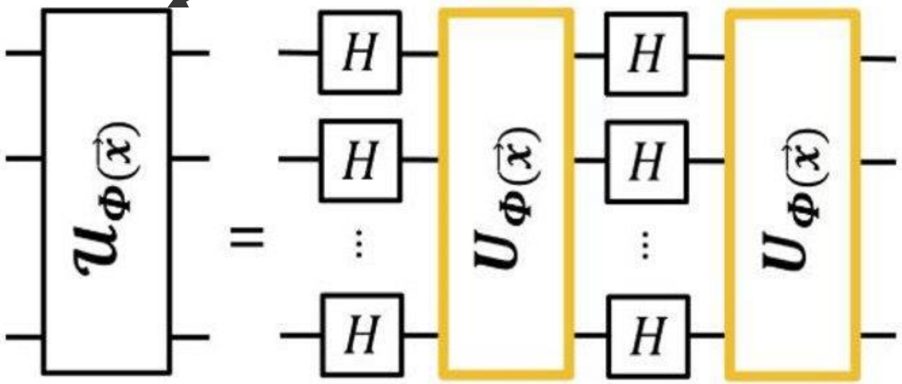
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

Quantum Feature map
(maps classical vector
into a quantum state)

$$\vec{x} \mapsto |\Phi(\vec{x})\rangle = \boxed{\mathcal{U}_{\Phi(\vec{x})}}|0\rangle^{\otimes n}$$





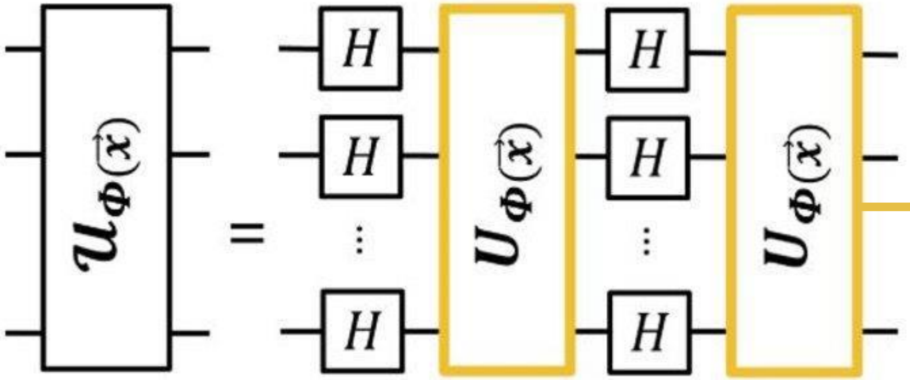
https://arxiv.org/abs/1804.11326

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

Quantum Feature map
(maps classical vector into a quantum state)

$$\vec{x} \mapsto |\Phi(\vec{x})\rangle = \mathcal{U}_{\Phi(\vec{x})}|0\rangle^{\otimes n}$$



$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{j \in S} Z_j \right)$$

https://arxiv.org/abs/1804.11326

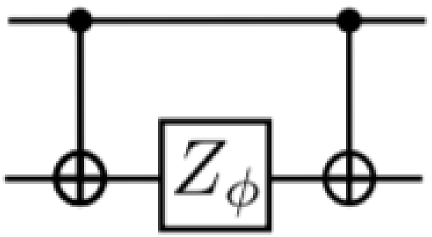
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{j \in S} Z_j \right)$$

Second order expansion (entangling gates)

$$e^{i\phi_{\{l,m\}}(\vec{x}) Z_l Z_m} =$$

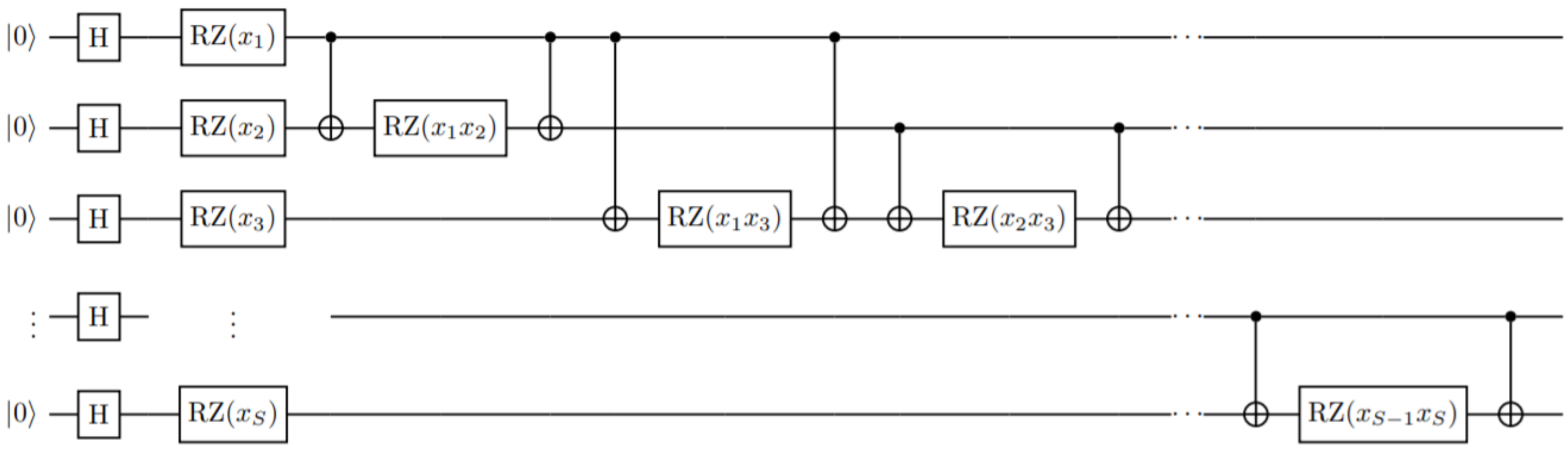


https://arxiv.org/abs/1804.11326

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{j \in S} Z_j \right)$$



Feature Map

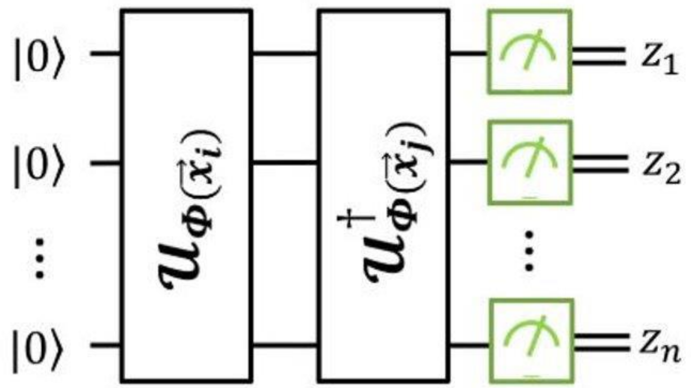
https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

Quantum Feature map
(maps classical vector into a quantum state)

$$\vec{x} \mapsto |\Phi(\vec{x})\rangle = \mathcal{U}_{\Phi(\vec{x})}|0\rangle^{\otimes n}$$



$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\varphi}(\boldsymbol{x}_i) \cdot \boldsymbol{\varphi}(\boldsymbol{x}_j)$$

Quantum Kernel

$$K(\vec{x}_i, \vec{x}_j) = |\langle \Phi(\vec{x}_i)|\Phi(\vec{x}_j)\rangle|^2 = \left|\langle 0|\mathcal{U}^{\dagger}_{\Phi(\vec{x}_j)}\mathcal{U}_{\Phi(\vec{x}_i)}|0\rangle^{\otimes n}\right|^2$$

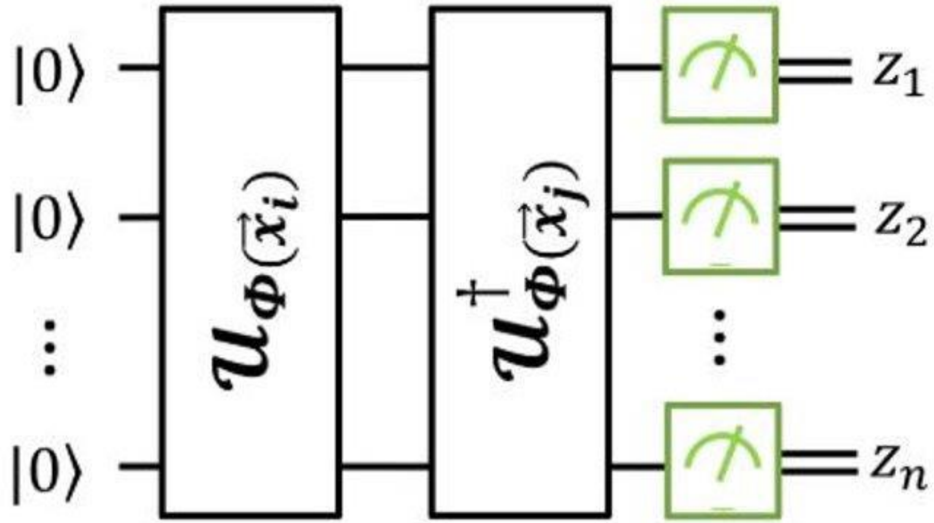
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

Quantum Kernel ➡

$$K(\vec{x}_i, \vec{x}_j) = |\langle \Phi(\vec{x}_i)|\Phi(\vec{x}_j)\rangle|^2 = \left|\langle 0|\mathcal{U}^\dagger_{\Phi(\vec{x}_j)}\mathcal{U}_{\Phi(\vec{x}_i)}|0\rangle^{\otimes n}\right|^2$$

The exact evaluation of the inner-product between two states generated from a similar circuit is #P - hard



https://arxiv.org/abs/1804.11326

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

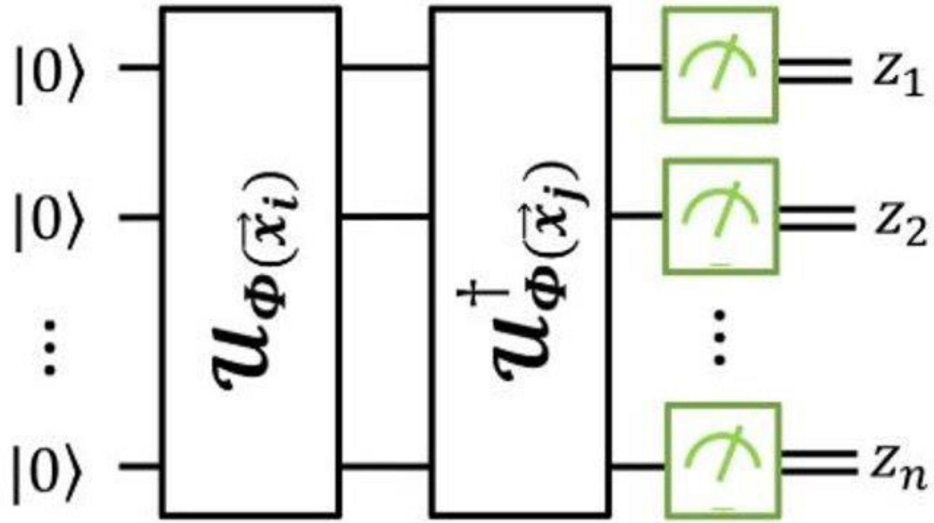
Quantum Kernel ➡

$$K(\vec{x}_i, \vec{x}_j) = |\langle \Phi(\vec{x}_i) | \Phi(\vec{x}_j) \rangle|^2 = \left| \langle 0 | \mathcal{U}^\dagger_{\Phi(\vec{x}_j)} \mathcal{U}_{\Phi(\vec{x}_i)} | 0 \rangle^{\otimes n} \right|^2$$

The exact evaluation of the inner-product between two states generated from a similar circuit is #P - hard

⬇

Quantum advantage: More complex feature map at low computational cost



https://arxiv.org/abs/1804.11326

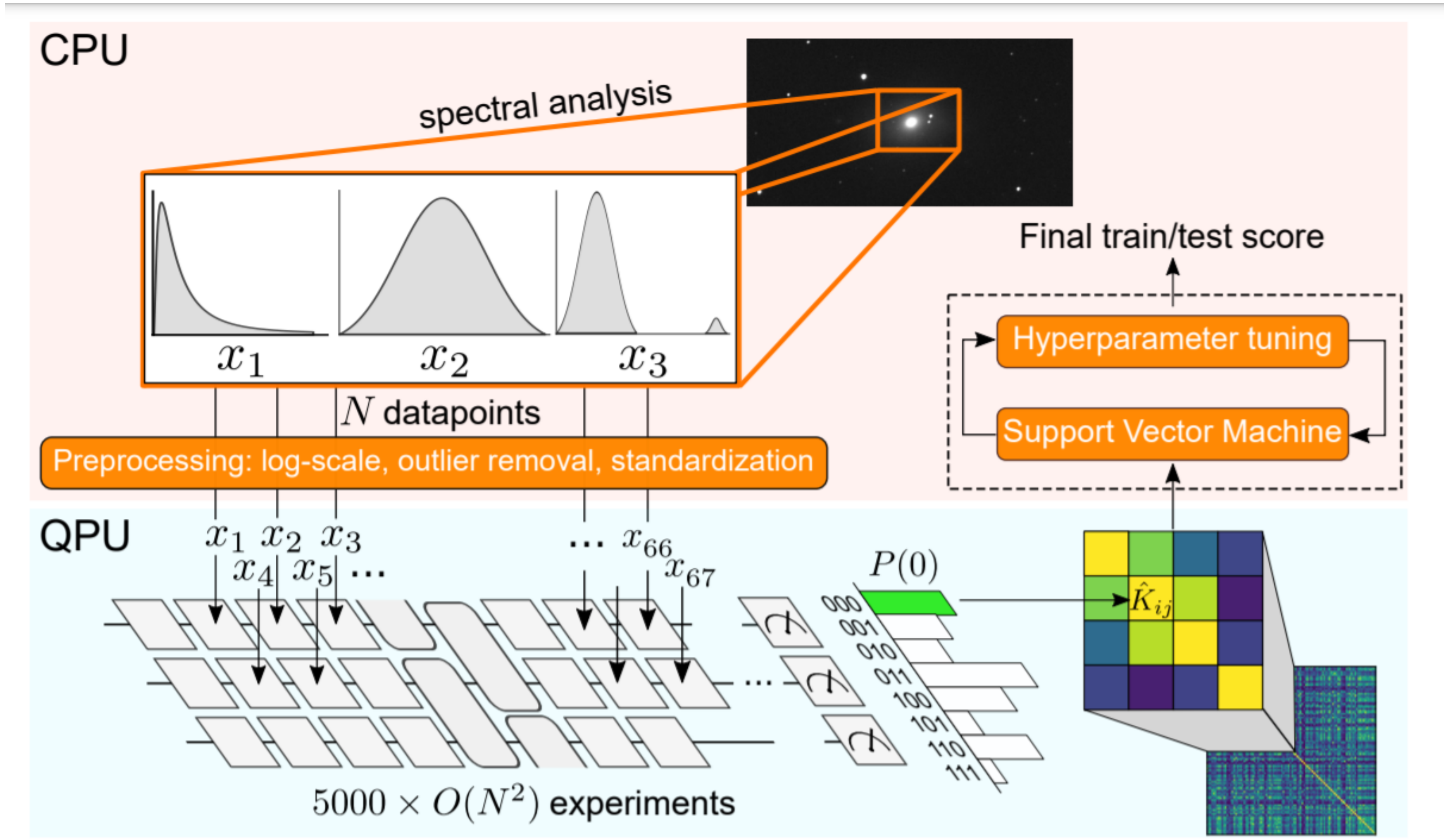
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

In order to **encode high dimensional data**, a **compression scheme** is applied

Each **qubit** encode **multiple datapoints via parametrized rotations**

https://arxiv.org/abs/2101.09581



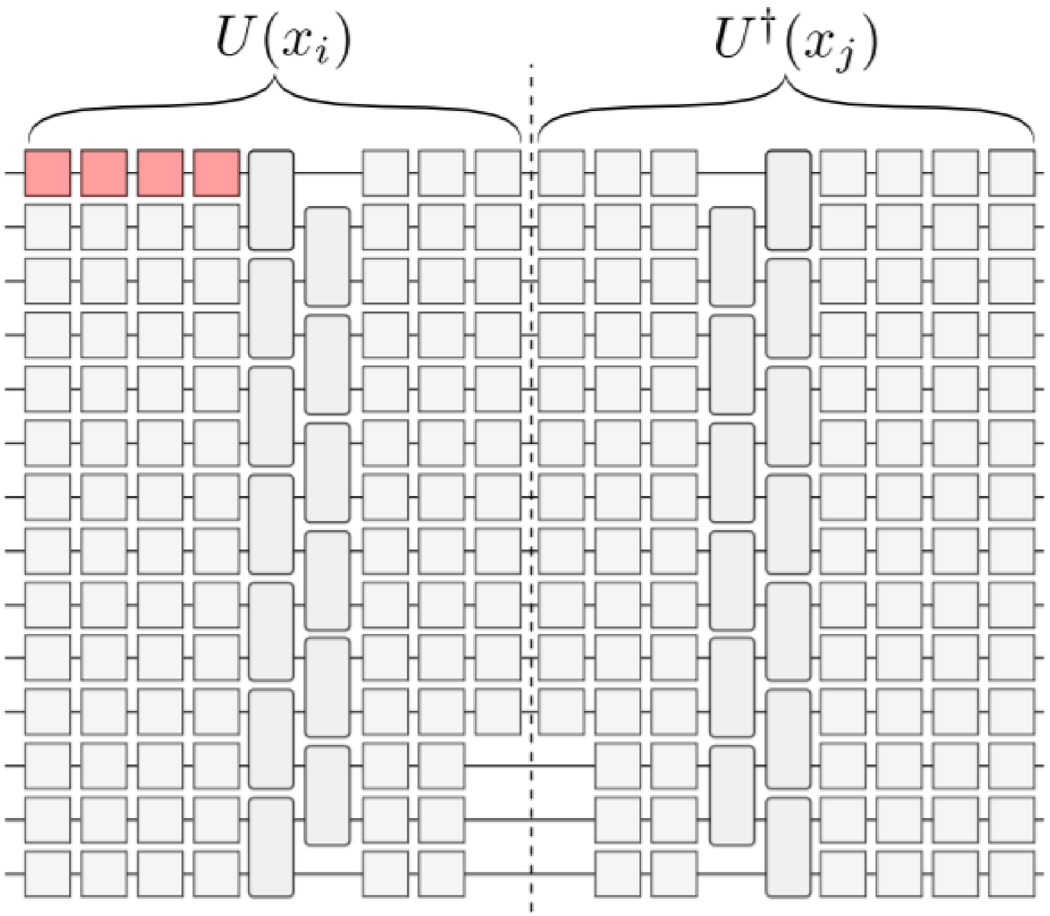
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

In order to **encode high dimensional data**, a **compression scheme** is applied

Each **qubit** encode **multiple datapoints via parametrized rotations**

https://arxiv.org/abs/2101.09581



CINECA QUANTUM COMPUTING LAB

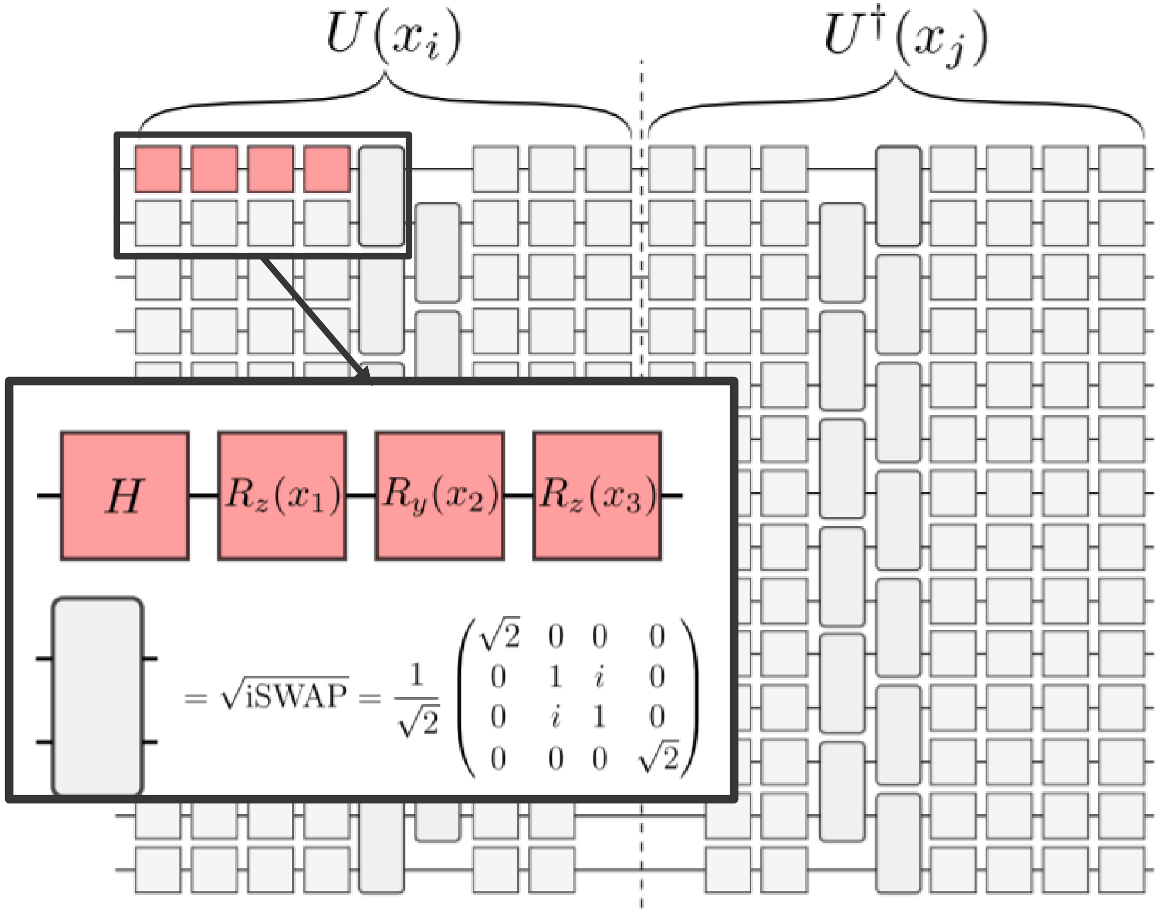
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum SVM

In order to **encode high dimensional data**, a **compression scheme** is applied

Each **qubit** encode **multiple datapoints via parametrized rotations**

https://arxiv.org/abs/2101.09581



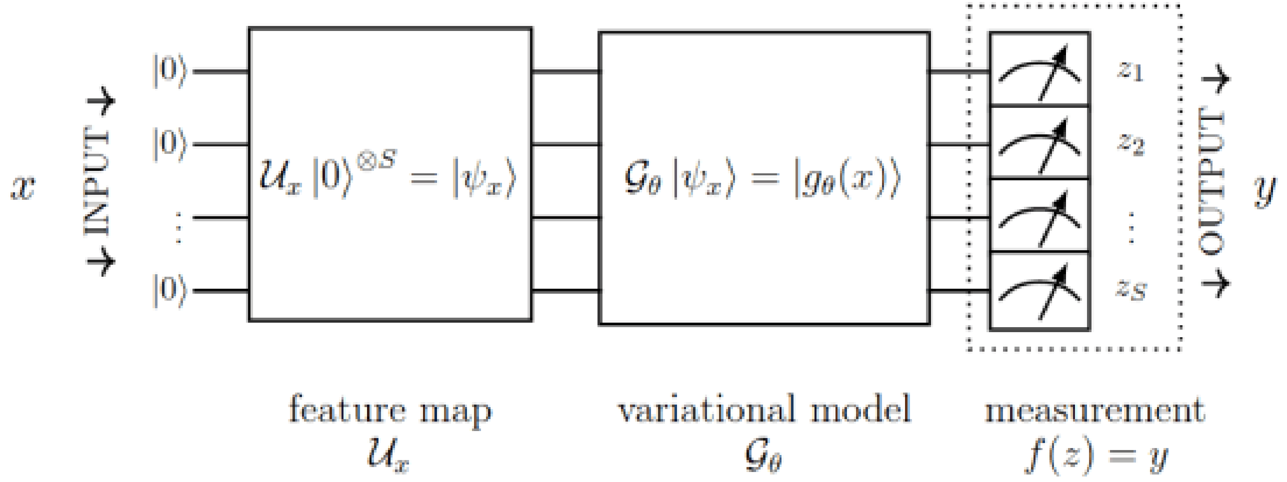
Quantum Neural Networks (QNN)

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

<u>Goal</u>: Address a supervised machine learning problem

<u>Method</u>: **Ansatz** consists of a **feature map** that serves to represent classical data and a **variational part** for learning



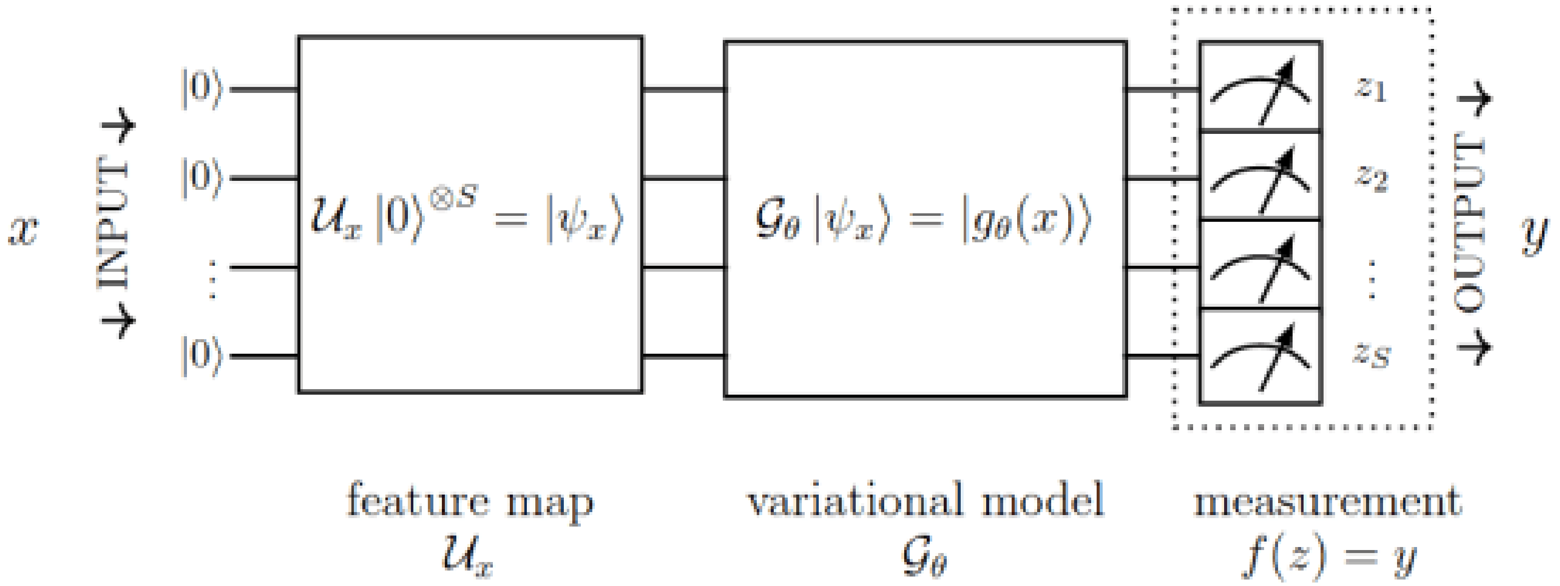
The circuit learns to classify new inputs based on the examples seen in the training phase

https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

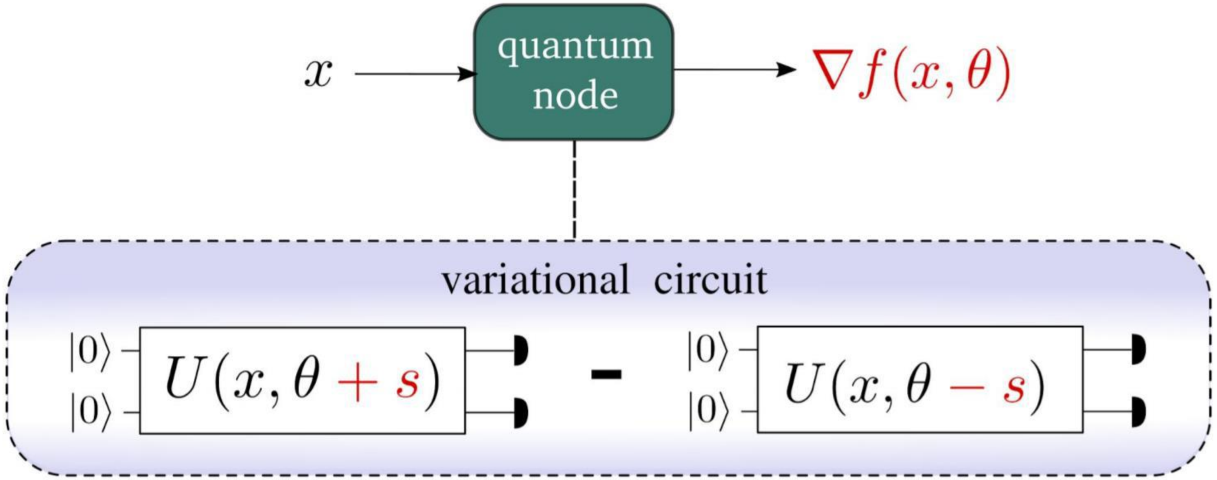
Quantum Machine Learning (QML) – Quantum NN

- **Feature map**: Store the inputs in a quantum state
- **Variational circuit**: Learnable parameter circuit
- **Expectation value**: Measurements introducing non-linearity



Quantum Machine Learning (QML) – Quantum NN

Training: Gradient of a quantum circuit can be exactly calculated



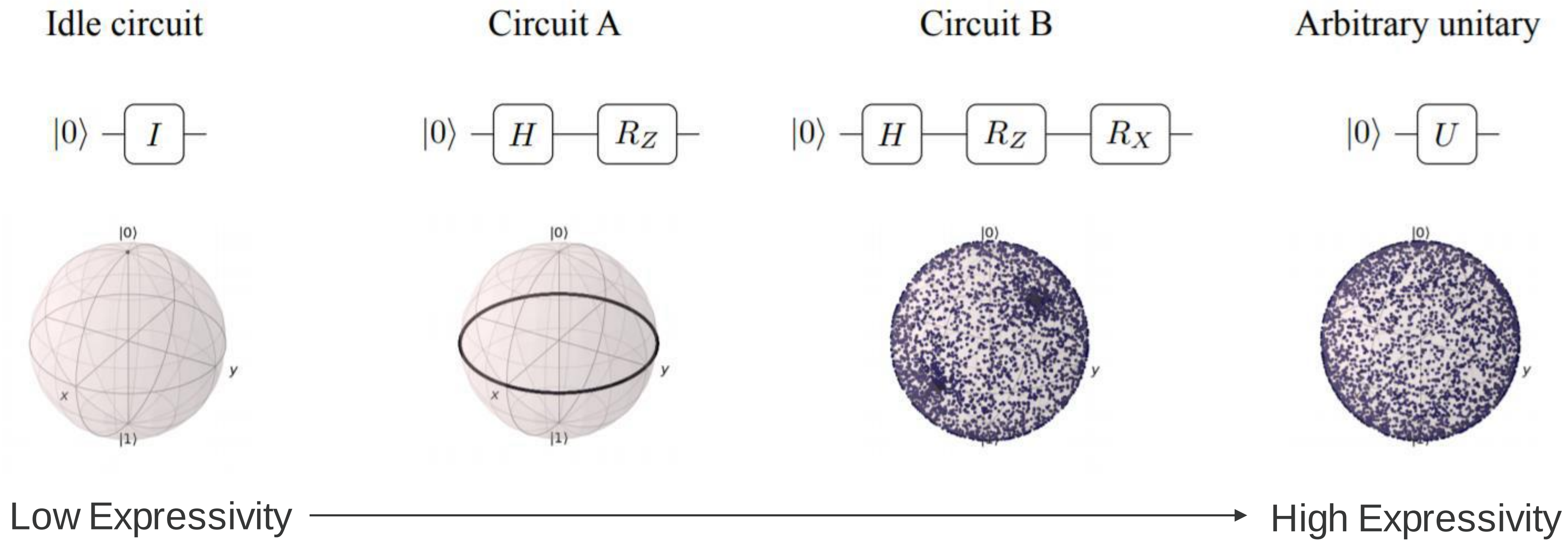
Parameter shift rule:
derivatives of the circuit can be computed by using the same variational circuit architecture

$$\partial_\theta f(\theta) = c[f(\theta + s) - f(\theta - s)]$$

Quantum Machine Learning (QML) – Quantum NN

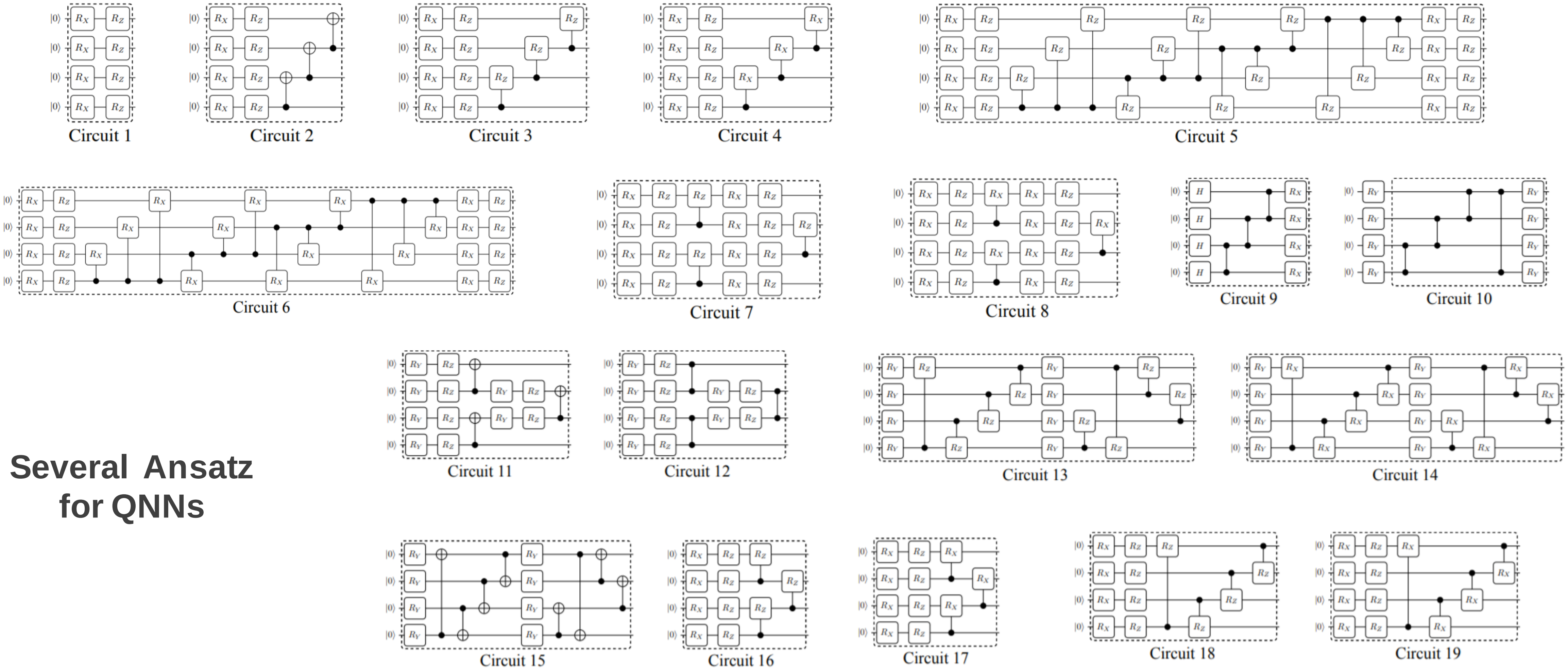
Ansatz - Variational circuit should take into account:

- Computational cost (depth of the quantum circuit)
- Expressiveness of the unitary operator



Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN



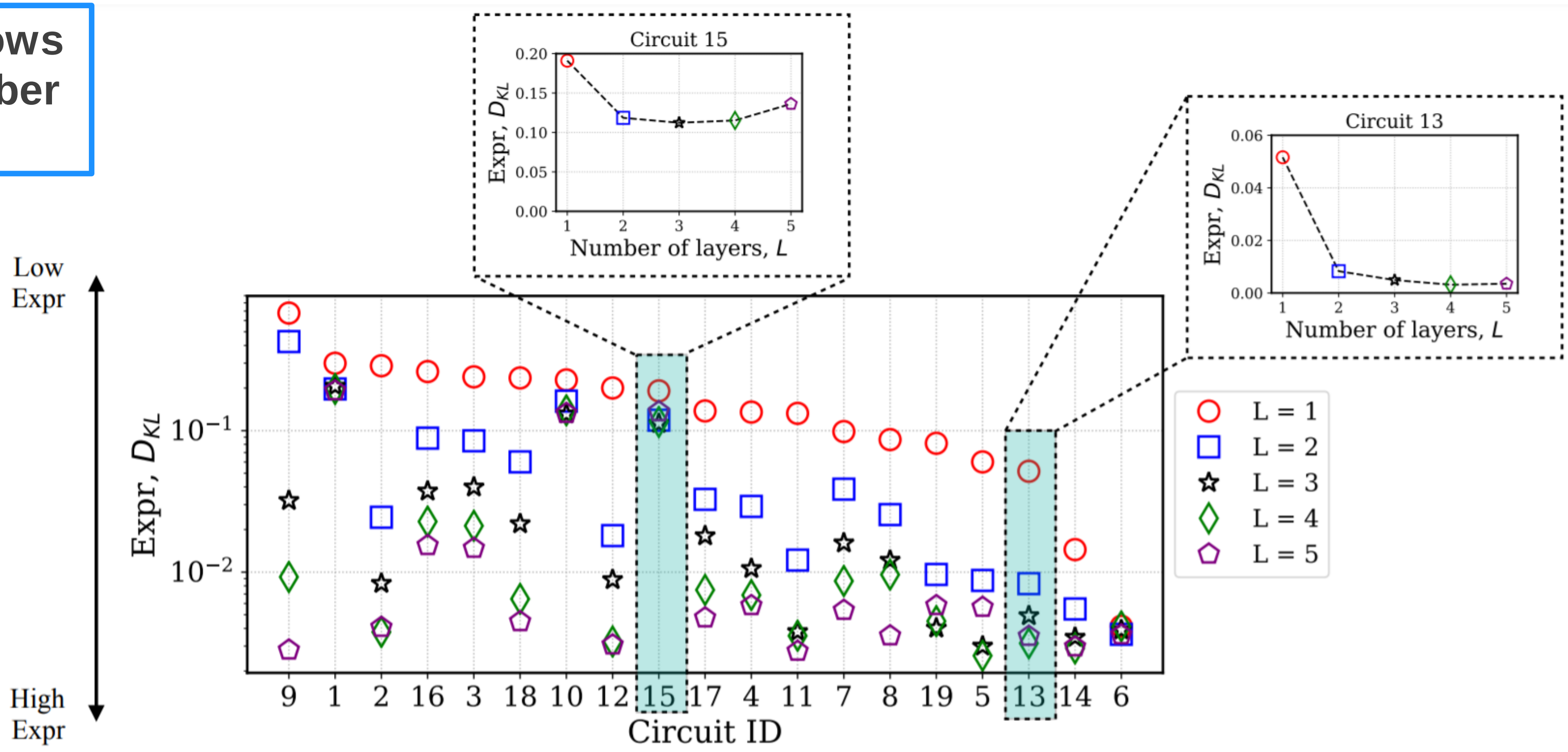
Several Ansatz for QNNs

https://arxiv.org/abs/1905.10876

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

Expressivity grows increasing number of layers

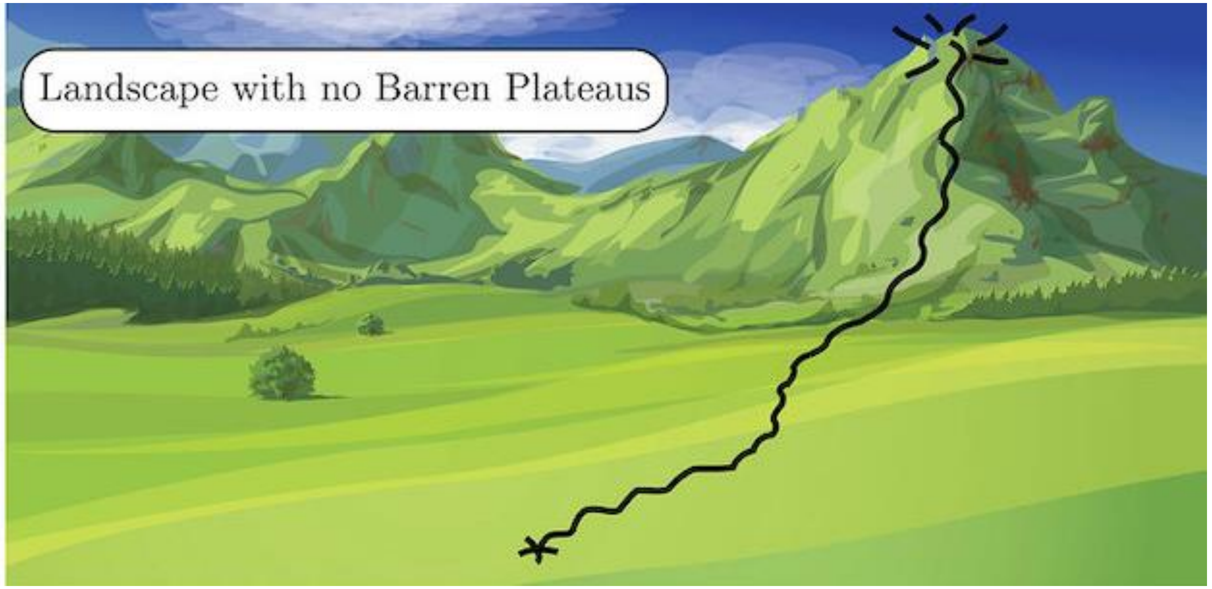


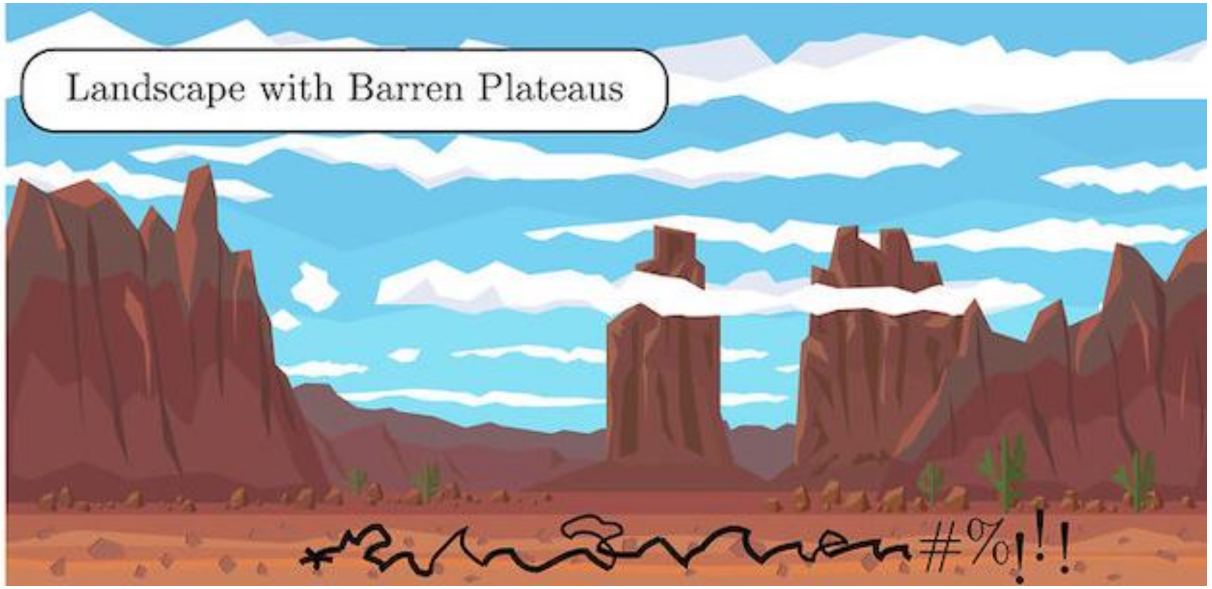
https://arxiv.org/abs/1905.10876

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

Barren Plateaus: Vanishing loss function Gradient that make it hard to train the QNN

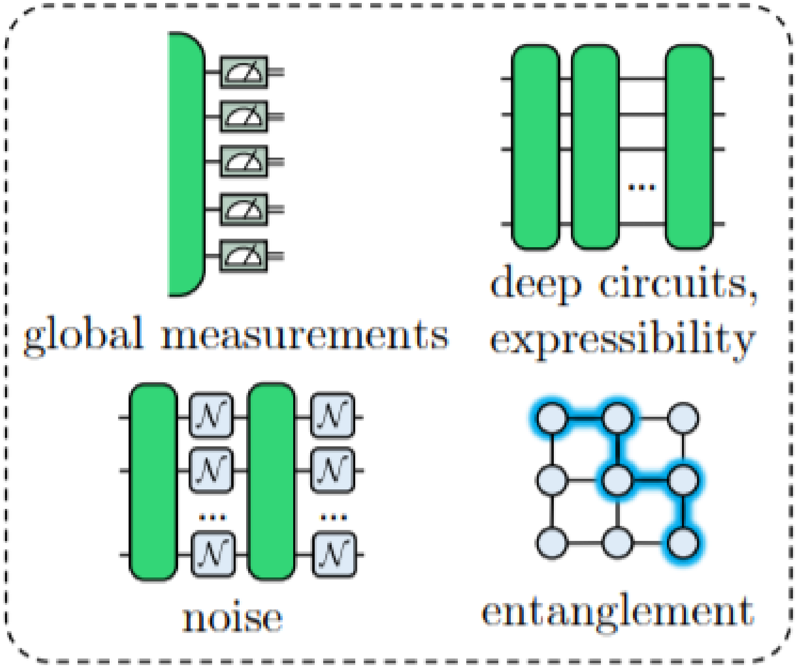




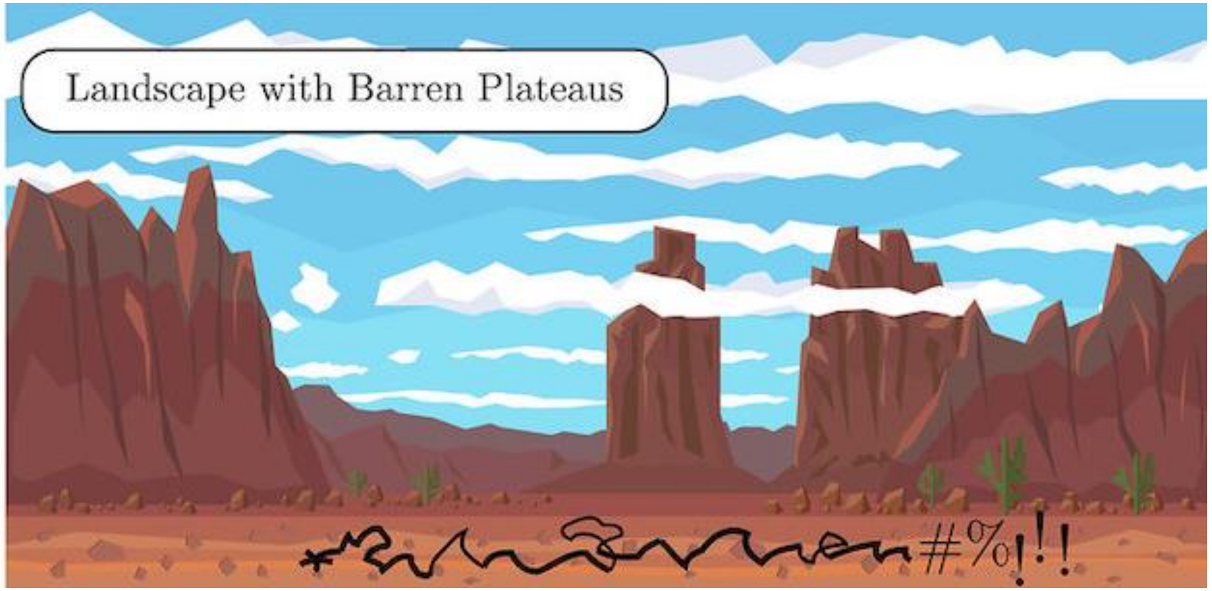
Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

Barren Plateaus: Vanishing loss function Gradient that make it hard to train the QNN



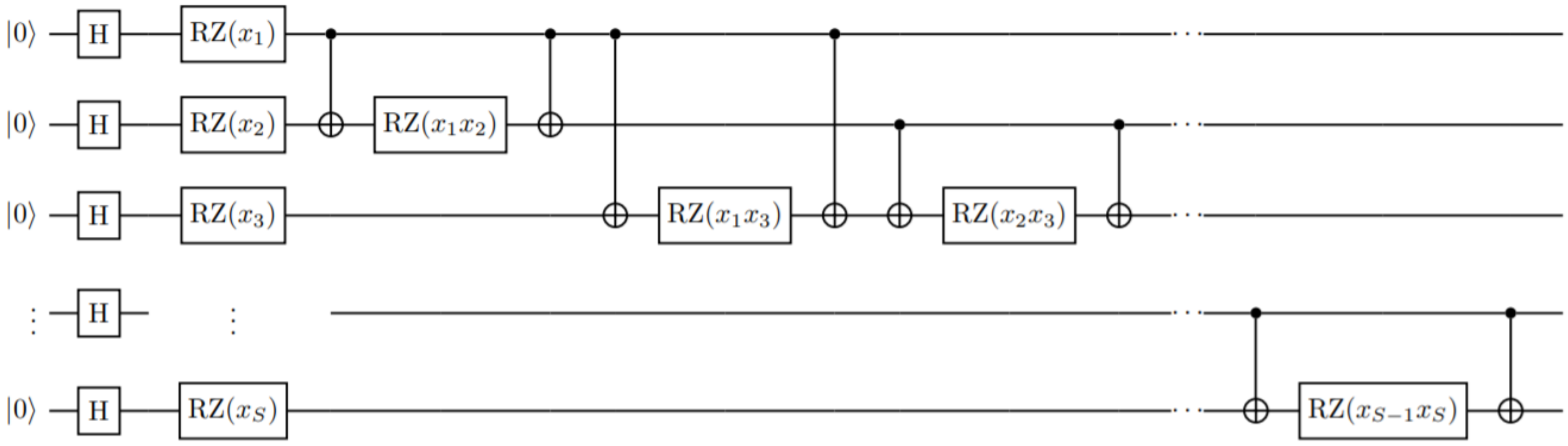
Features that may induce Barren Plateaus



Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

The Power of QNNs



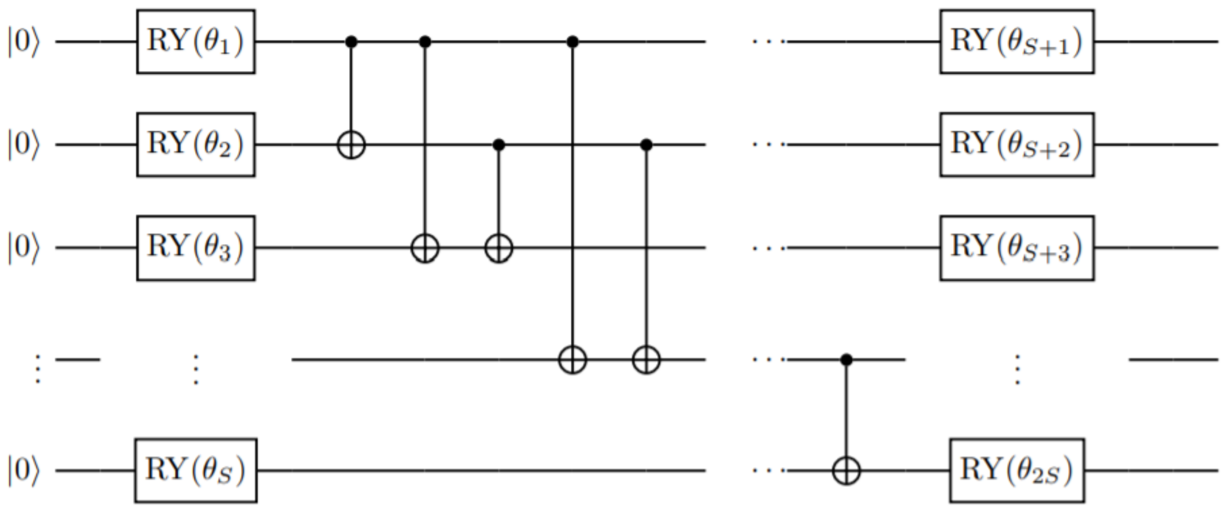
Feature Map

https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

The Power of QNNs



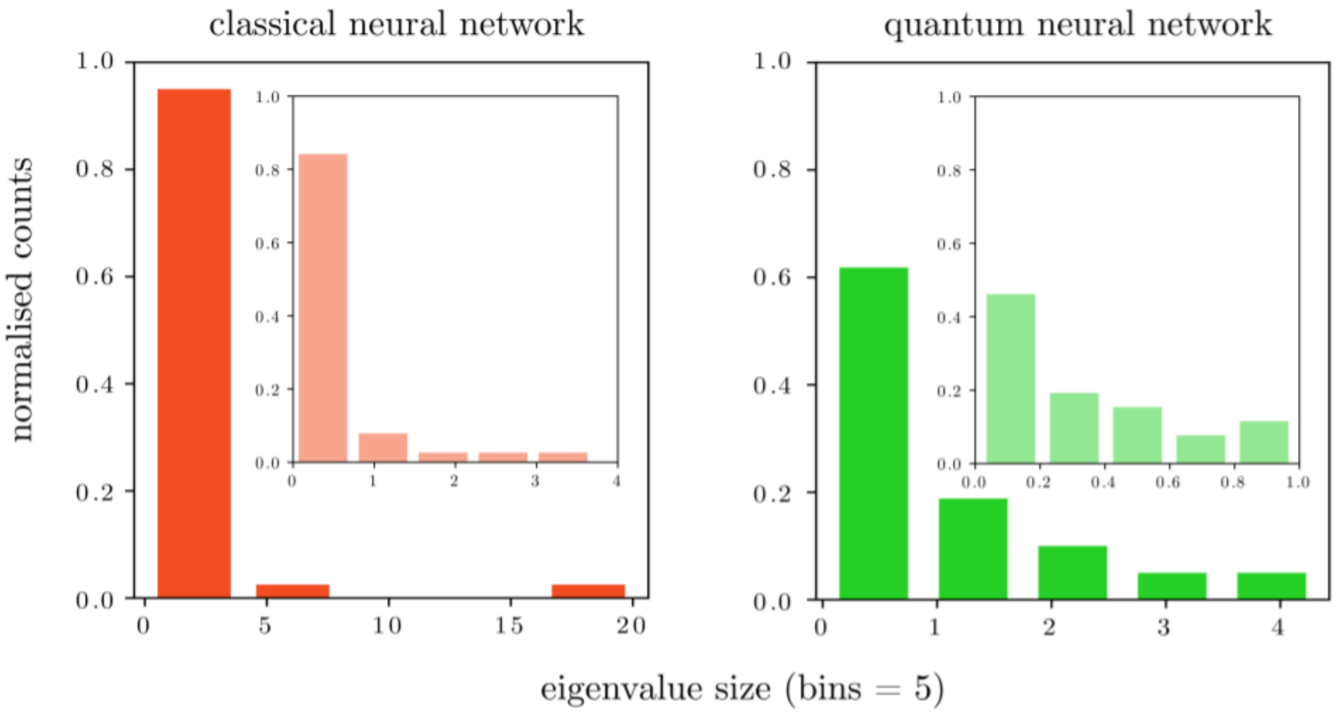
Variational Circuit

https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

Quantum Machine Learning (QML) – Quantum NN

The Power of QNNs



More evenly spread eigenvalues of the Fisher information for the QNN wrt classical NN with same number of parameters

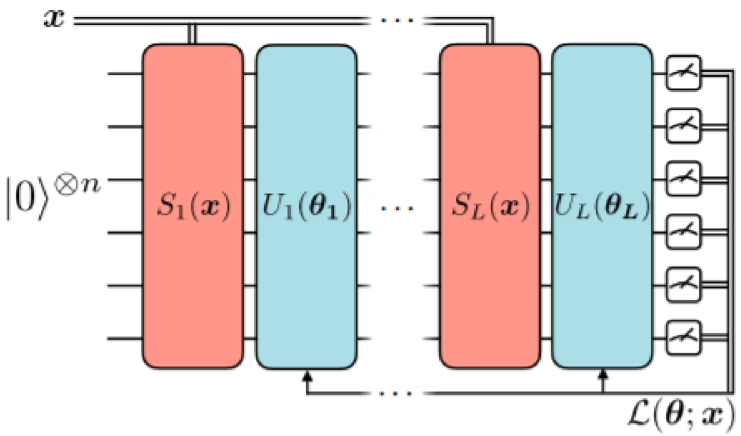
⇓

Better Generalization
(how accurately the algorithm is able to predict outcome values for previously unseen data.)

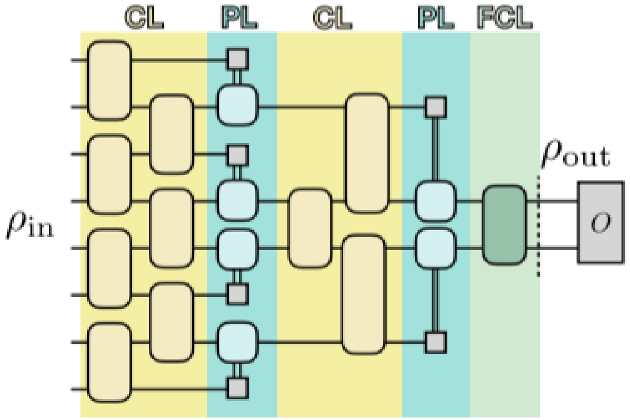
https://arxiv.org/abs/2011.00027

Quantum algorithms for NISQ Devices

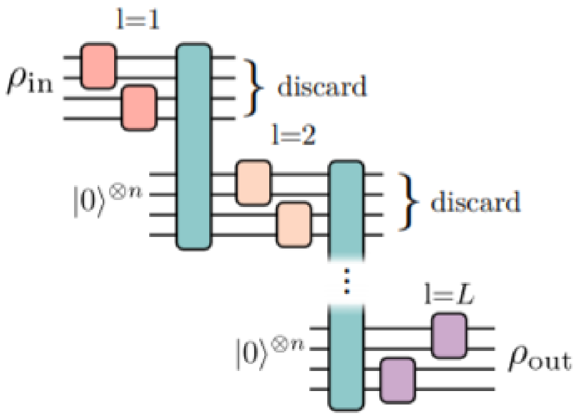
Quantum Machine Learning (QML) – Quantum NN



Standard QNN



Convolutional QNN



Dissipative QNN

https://arxiv.org/abs/2102.03879

NISQ-ready algorithms for general purpose QPU

Main Challenges

- **Trainability / optimization of parameters:** best optimization scheme or technique
- **Barren plateaus:** Vanishing gradients that make it hard to optimize
- **Ansatz and initialization strategies**: structure of the parametric circuit
- **Efficiency:** precision required in the output per the amount of resources consumed
- **Accuracy:** the degree to which output conforms to the correct value or a standard.
- **Hardware noise / Error mitigation:** optimal techniques to reduce errors without an overhead in resources

NISQ-ready algorithms

QUANTUM ADVANTAGE IN THE NISQ ERA?

Quantum Computing @ CINECA

CINECA: Italian HPC center

CINECA Quantum Computing Lab:

- Research with Universities, Industries and QC startups
- Internship programs, Courses and Conference (HPCQC)

https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it