Learning and Training in Quantum Environments

Mohsen Heidari

CS Department, Indiana University

Quantum Computing System Lecture Series
Dec. 2022

Evolution of Quantum Computing

1980s **Early suggestions of QC:**
Manin ’80, Feynman ’82, Benioff ’82

1990s **Models of QC and first Q algorithms:**
- Deutsch and Jozsa
- Shor’s factoring algorithm
- Grover’s search algorithm,
- Simon, Bernstein and Vazirani
- quantum perceptron by Lewenstein

1997 Early 2-qubit QC

2001 Shor’s algorithm on a 7-qubit QC for factoring 15

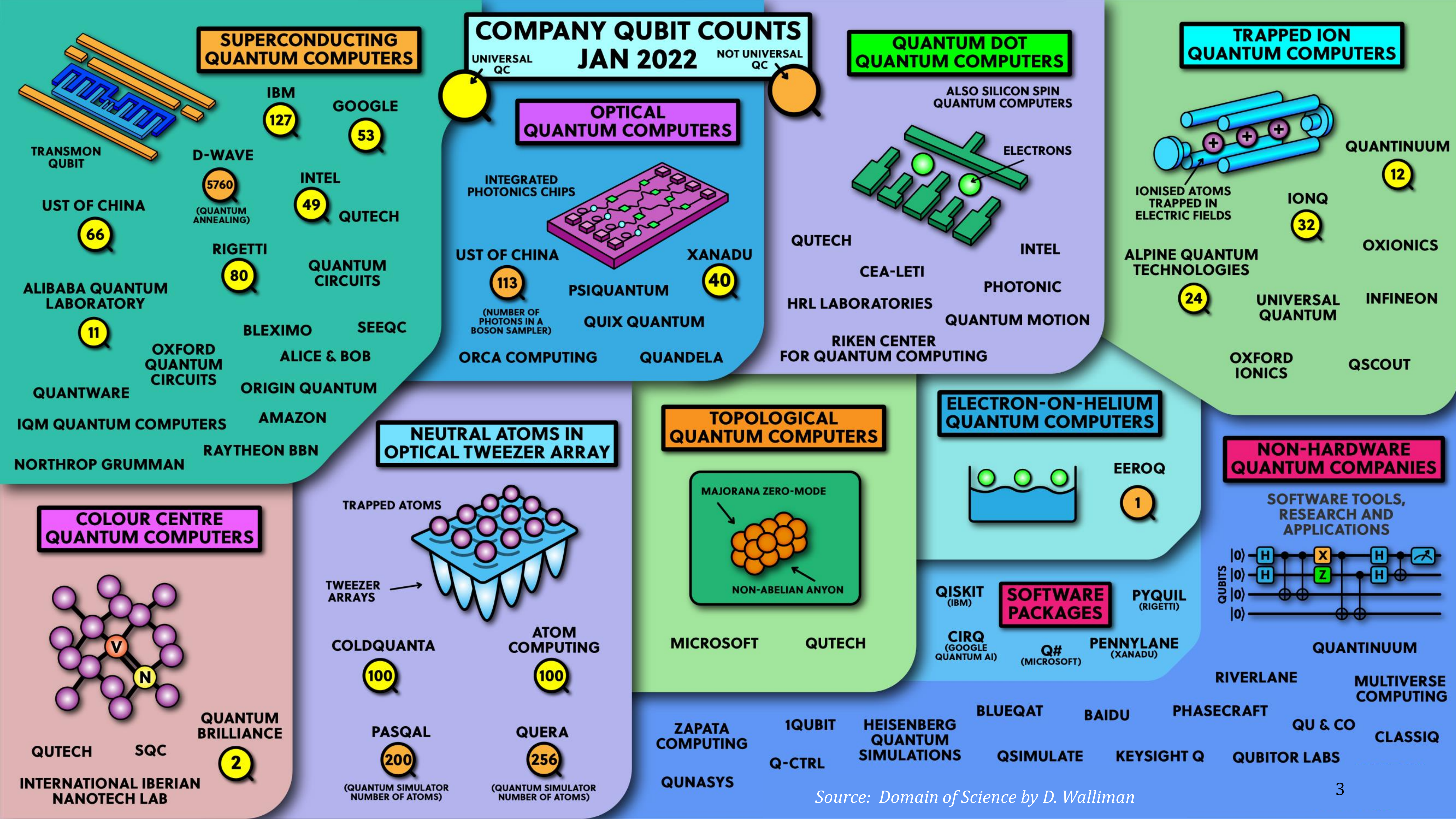
2021 Beyond 100 qubits [IBM]

~2024 Beyond 1000 qubits [IBM]

⋮ ⋮

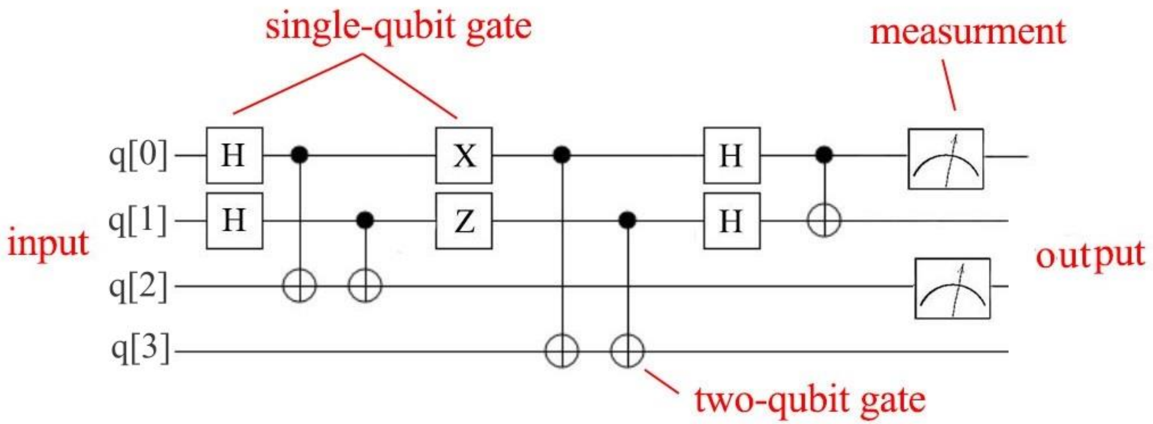
???? Fault tolerant QCs

IBM: Osprey 433-qubit (2022)

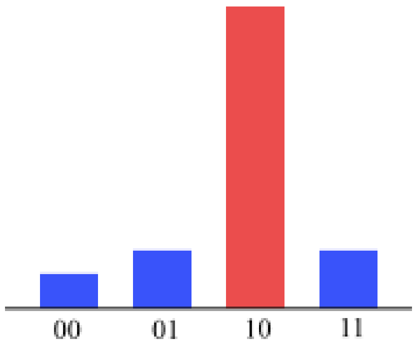


Source: Domain of Science by D. Walliman

Quantum Computer



Gate-based Model



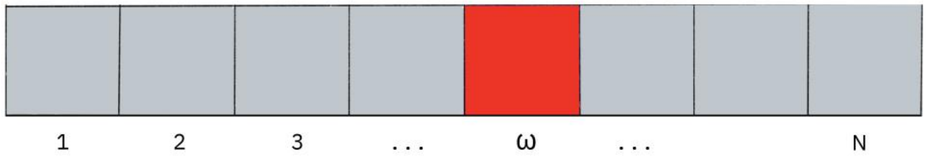
Quantum algorithm generates a probability of possible outcomes

- Quantum gates ≡ logical gates in classical computers
- Input: qubits
- Output: classical

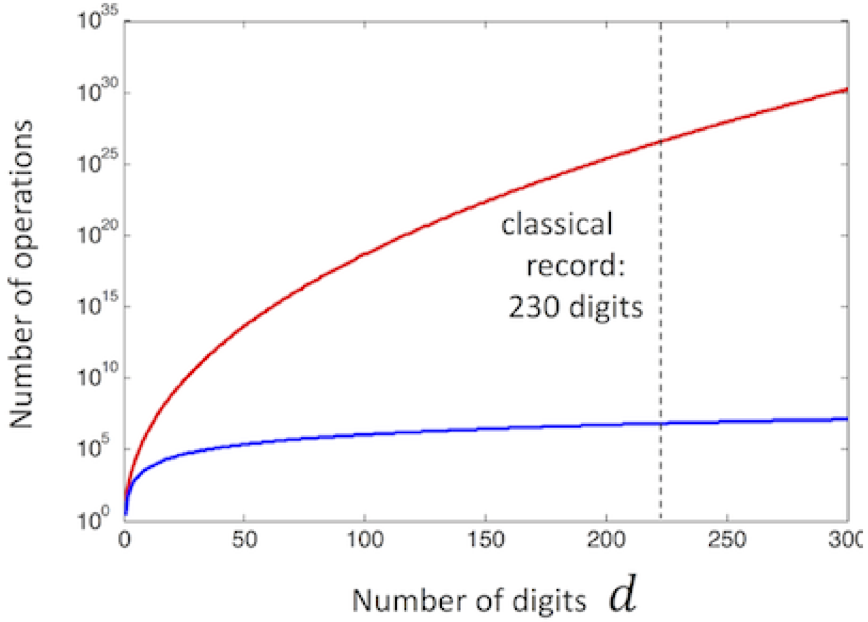
Why Quantum Computing?

- **Speed ups classical problems:**
  - Integer factoring ➔ NP
    - Best known classical algorithm $2^{O(d^{1/3})}$
    - Shor's algorithm $O(d^3)$ ➔ BQP complexity class.

    ➔ breaking RSA encryption
  - Simon's problem: $\Omega(2^{n/2})$➔ $\Omega(n)$ query complexity
  - Bernstein–Vazirani, Deutsch-Jozsa, …
  - Polynomial speedups in several problems:



Grover's search algorithm: $O(\sqrt{N})$
Classical search: $O(N)$

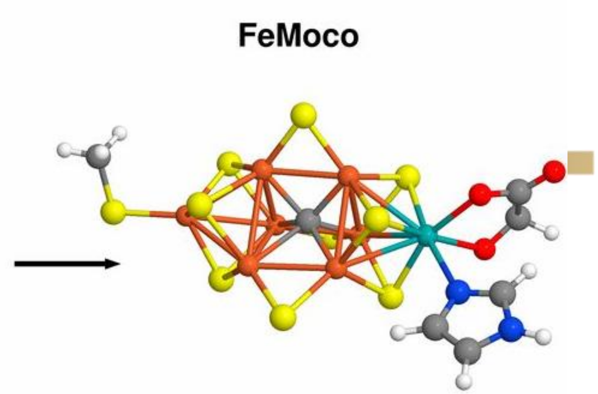


- Best known classical (field sieve)
- Shor's algorithm

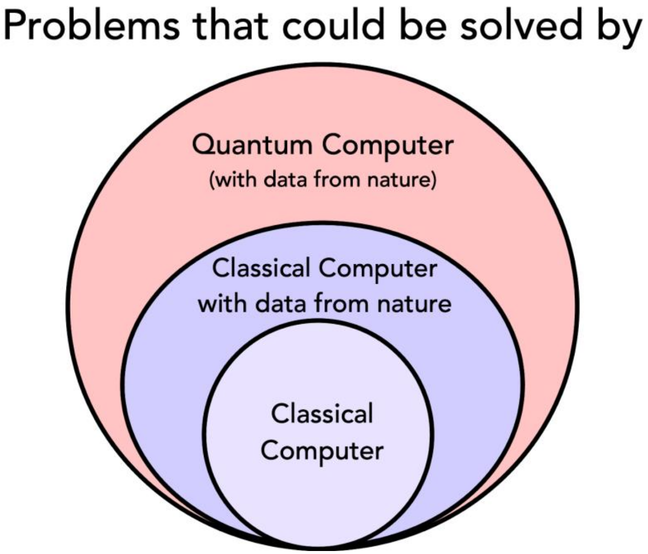
Source: IBM

Why Quantum Computing?

- **Leveraging quantum data**
  - Directly operating on quantum states of physical systems
    - Optical systems, sensors with quantum effects, …
    - Not accessible to classical computers.
  - Simulating quantum physical processes
    - Exponentially hard for classical computers as $dim = 2^q$
- Wide range of applications:
    - Simulations in chemistry
      - Drug discovery, material design (batteries, solar panels), …
      - Faster and cheaper prototyping than physically making and testing
    - Quantum many-body systems
    - Phonic circuits
    - Social sciences



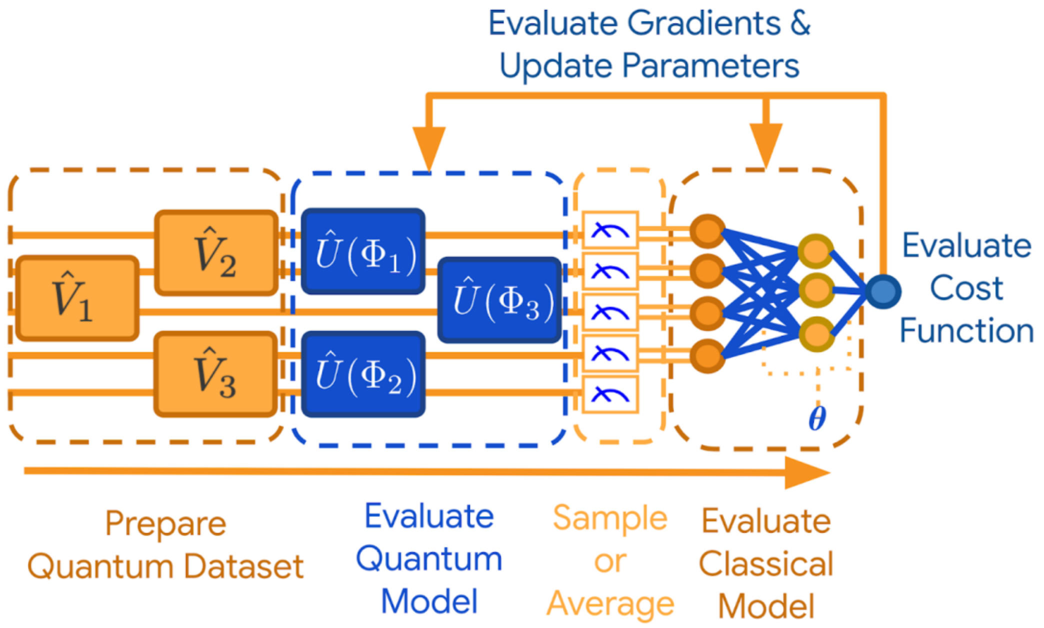
Better catalyst for fertilizer production
(2% of global CO_2 emission)
Source: Microsoft Research



Why Quantum Computing?

Enhancing learning models

- Exploring larger classes of probability distributions because of entanglement.
- Quantum machine learning.

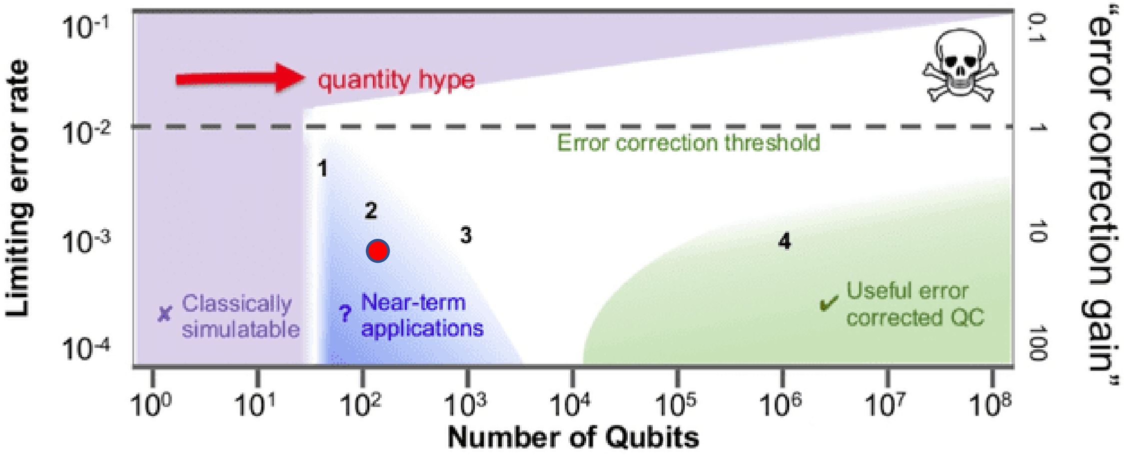


Source: Google Research

Challenges

- **Qubit decoherence:** interactions with the environment.
- Coherence time needs to be much longer than gate operations time.
- **Infidelity**: quantum operations are erroneous (NISQ era).
- **Scalability**: number of qubits (entangled) in the device.
- **Speed:** number of operations per second

Need Both Quality and Quantity



Source: Google Research

Approaches

- Hardware:
  - Higher fidelity quantum operations
  - Longer coherence time
- Quantum Error Correcting Codes
  - Fault tolerant QCs
- Algorithms on NISQ devices
  - Handling the challenges with the algorithms
  - Infidelities as extra sources of noise/randomness
  - One-shot approach for dealing with no-cloning and state collapse

Outline:

Part 1: Introduction

Part 2: Learning with quantum computers

Part 3: Band-limited QNNs

Postulate 1: Quantum State

Classical bit

- belongs to $\{0,1\}$, either 0 or 1.
- Can make several copies

Qubit

- Lives in Hilbert Space ($\mathbb{C}^2$)
- In superposition of $|0\rangle$ and $|1\rangle$:

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

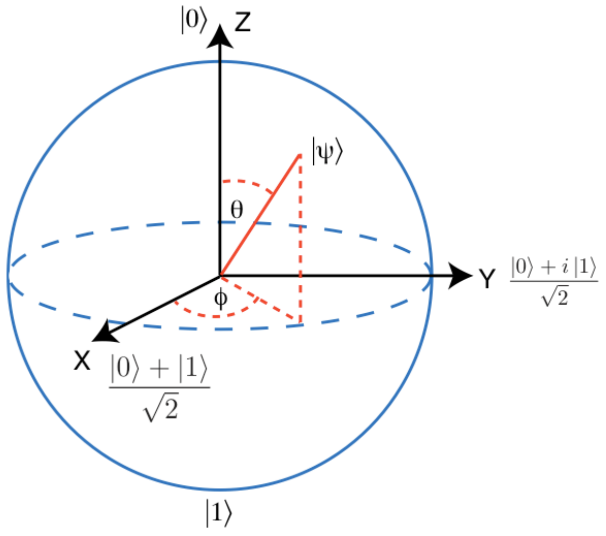
- Each qubit doubles the dimension:
  - d qubits live in $\mathbb{C}^D$, where $D = 2^d$
- Impossible to clone a qubit

Quantum State

- Quantum state contains everything we can possibly know about a quantum system.
- More general than qubits ➔ Hilbert Space
- For this talk, we simply assume states are qubits ➔ $\mathbb{C}^D$
- A qubit is represented by a complex vector $(\alpha, \beta) \in \mathbb{C}^2$
  - Unite norm: $|\alpha|^2 + |\beta|^2 = 1$
- Dirac's Notation:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \qquad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

- Superposition state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$



Bloch Sphere

Postulate 2: Evolution of Quantum Systems

- An isolated (closed) quantum systems evolves only through a unitary transformation.
- State at time t_1: $|\psi(t_1)\rangle$
- State at time t_2: $|\psi(t_2)\rangle$

Unitary transformation

$$|\psi(t_2)\rangle = U(t_1, t_2)|\psi(t_1)\rangle$$

- Unitary transformations: $U^t U = UU^t = I$

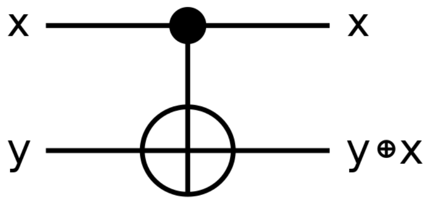
Hadamard matrix: $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

- Unitary ➔ reversible

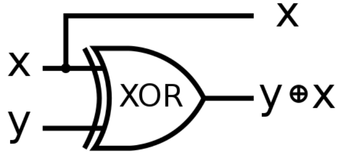
Reversible Gates

- Quantum gates must be reversible. ➔ reversible logic!
- The input and output dimension should match!

a_1	a_2	a_1 AND a_2
0	0	0
0	1	0
1	0	0
1	1	1



input		output	
x	y	x	y+x
$\vert 0\rangle$	$\vert 0\rangle$	$\vert 0\rangle$	$\vert 0\rangle$
$\vert 0\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert 1\rangle$
$\vert 1\rangle$	$\vert 0\rangle$	$\vert 1\rangle$	$\vert 1\rangle$
$\vert 1\rangle$	$\vert 1\rangle$	$\vert 1\rangle$	$\vert 0\rangle$

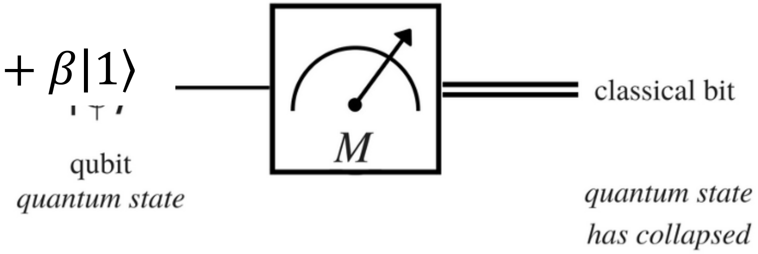


input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Postulate 3: Quantum Measurements

- Quantum measurements gives us classical information about quantum systems
  - Example: position of an electron, polarization of a photon.
- Typically appear at the end of a quantum computer to output a classical answer.
- Quantum measurements change the state itself.

$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$

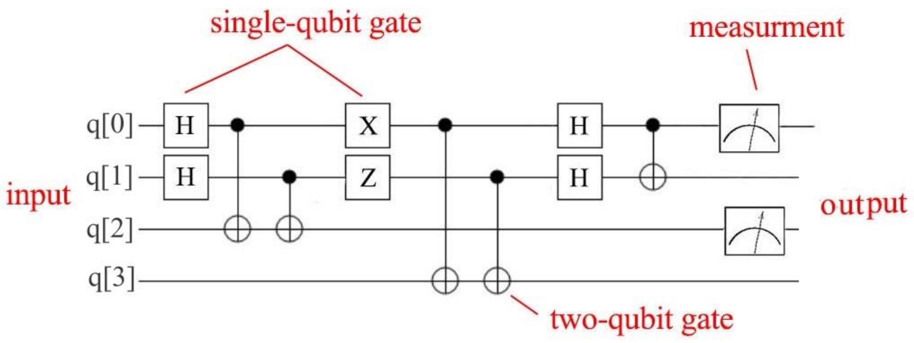


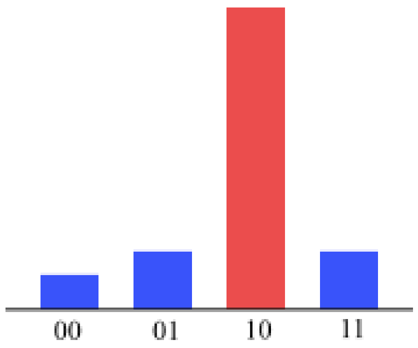
$$\begin{cases} 0 & \text{with prob } |\alpha|^2 \\ 1 & \text{with prob } |\beta|^2 \end{cases}$$

- Measurement output is probabilistic!

Postulate 3: Quantum Measurements

- Reading qubits destroys them:
  - Good for security and privacy
  - Bad for computation and learning





Quantum algorithm generates a probability of possible outcomes

Consequences

- State collapse has important consequences:
  - Measurements change the state
- Uncertainty Principle:
    - Some observables cannot be measured simultaneously!
    - Example: position and momentum of an electron!
- In quantum ML, we can measure either training loss or the gradient!
  - Even the gradient's components may not be simultaneously measurable.

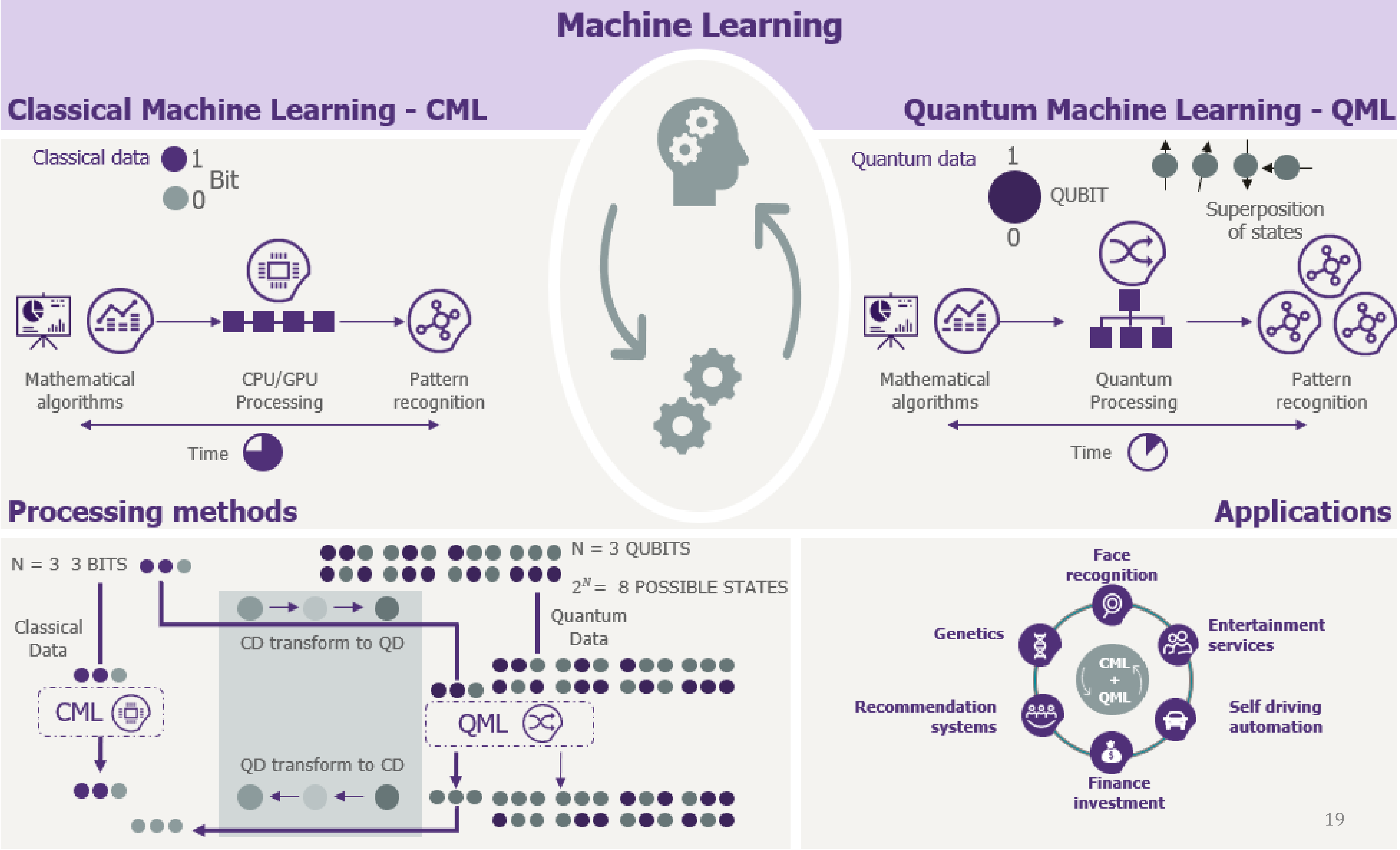
$$\nabla L\left(\vec{\theta}\right) = \left(\frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2}, \dots, \frac{\partial L}{\partial \theta_m}\right)$$

Outline:

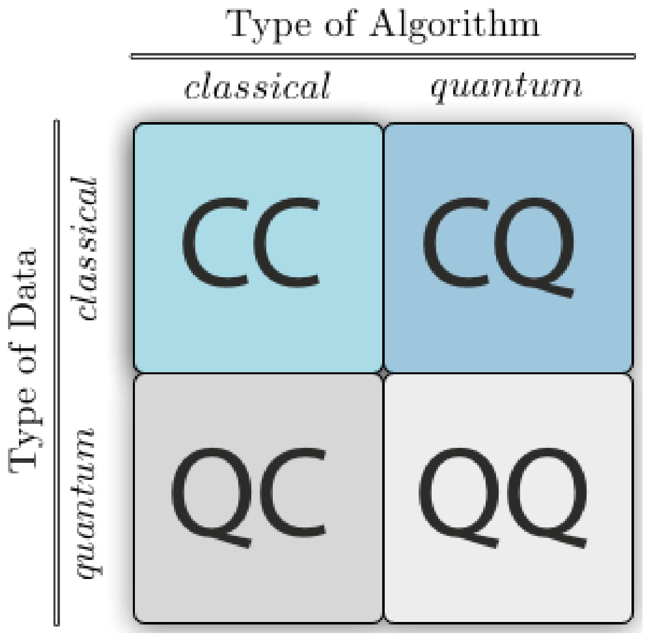
Part 1: Introduction

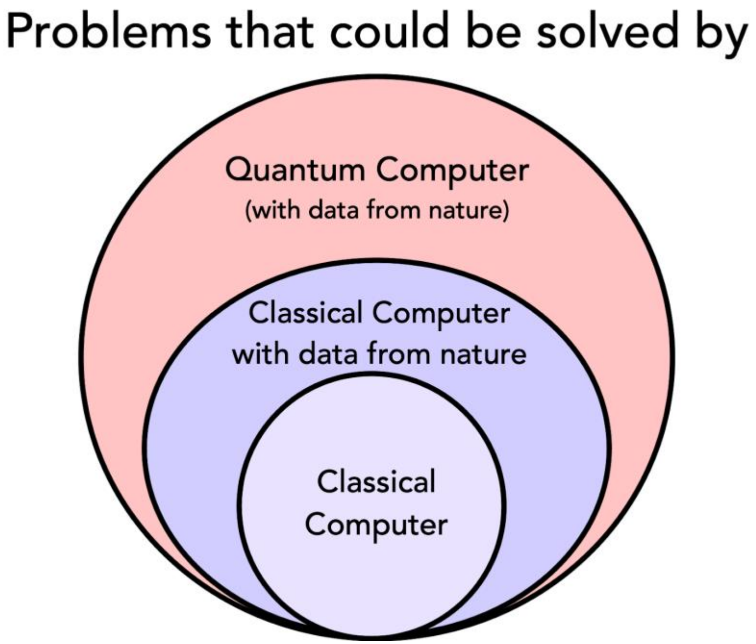
Part 2: Learning with quantum computers

Part 3: Band-limited QNNs


Machine Learning
Classical Machine Learning - CML
Quantum Machine Learning - QML
Classical data
1
0
Bit
Mathematical algorithms
CPU/GPU Processing
Pattern recognition
Time
Quantum data
1
QUBIT
0
Superposition of states
Mathematical algorithms
Quantum Processing
Pattern recognition
Time
Processing methods
Applications
N = 3 3 BITS
Classical Data
CML
CD transform to QD
QD transform to CD
N = 3 QUBITS
2^N = 8 POSSIBLE STATES
Quantum Data
QML
Face recognition
Entertainment services
Self driving automation
Finance investment
Recommendation systems
Genetics
CML + QML

Types of Learning





Quantum ML

× No-cloning of training samples.

× Measuring destroys the samples.

× Uncertainty Principle:

- Gradient and training loss cannot be measured simultanec

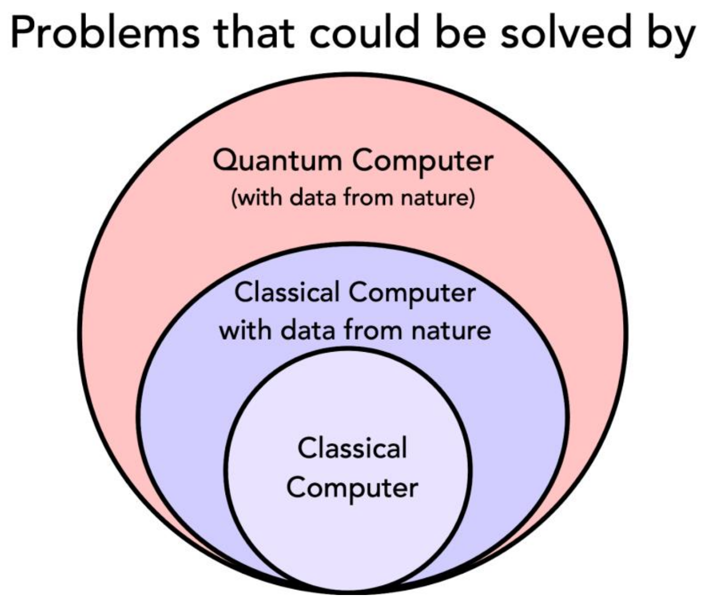
❖Stochasticity: the training loss is random.

✓Entanglement: more powerful patterns

✓Exponential state space: richer models!

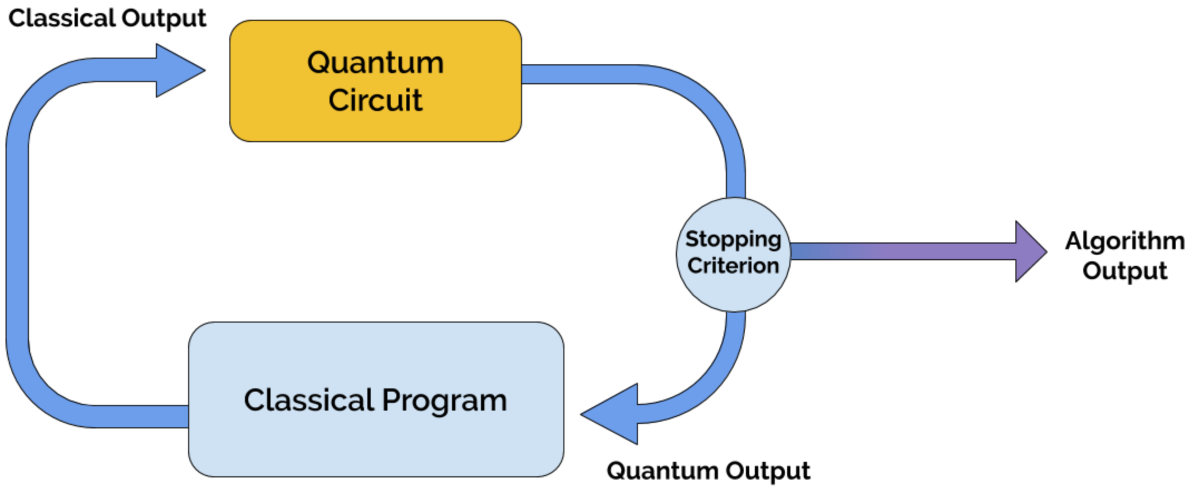
- ✓ 300 qubit ➔ $\dim = 2^{300}$

✓Quantum Data not accessible to classical.



Optimization and Training

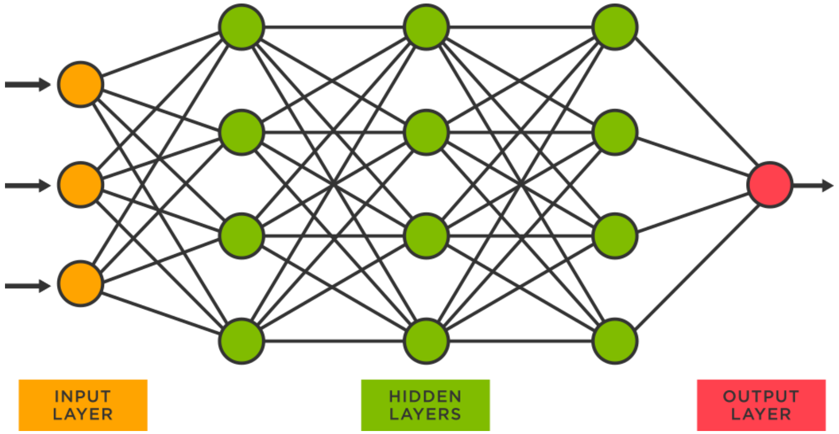
- Applications:
  - Dynamical simulations,
  - Finding ground states (condensed matter, …),
  - Photonic circuits, …
  - Machine learning (classification, generative models, …),
  - Combinatorial optimization.
- Iterative optimization
- Quantum-classical hybrid approach
- Prior works:
  - QAOA (Farhi, Goldstone, and Gutmann ‘14),
  - Gradient approximation (Farhi and Neven ’18 Rebentrost et al., ‘18),
  - Other variational algorithms (Peruzzo et al ‘14, McClean et al ’16), …

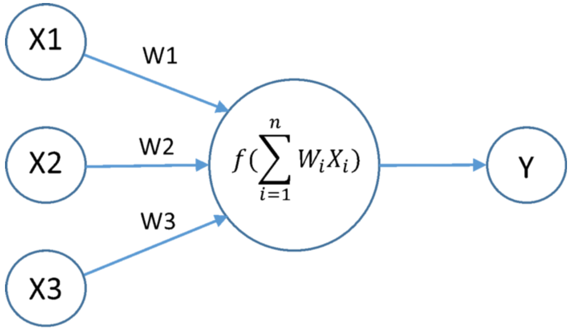


$$\vec{a}^* = \operatorname*{argmin}_{\vec{a} \in \mathcal{D}} L(\vec{a})$$

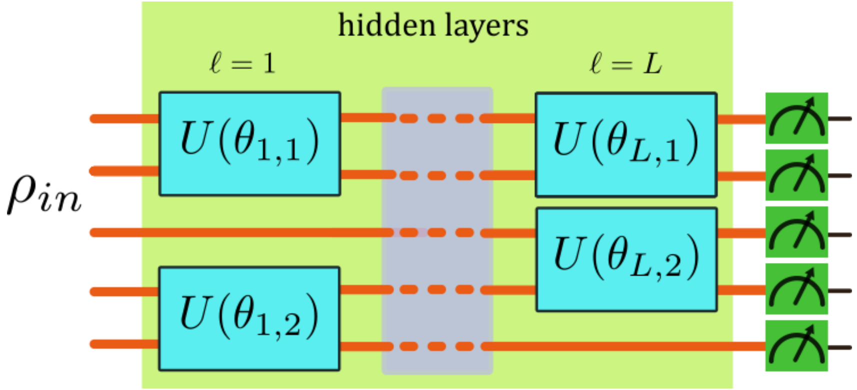
QNNs

Classical Neural Networks



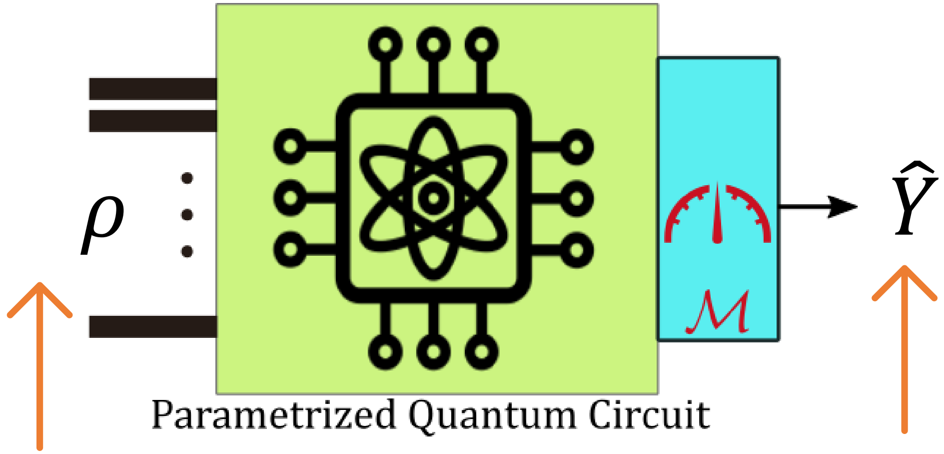


Activation function



- QNNs are network of small quantum circuits as activation functions, as in CNNs.
- Input: qubits, output: classical
- Measurements at the output layer ➔ making the decision
- Quantum perceptron (Lewenstein 1994, Tothet al. 1996).
- Applications in classical/quantum machine learning:
- (Schuld et al. 2014, 2020)(Mitarai et al. 2018)(Farhi and Neven 2018)(Torrontegui and Garcia-Ripoll 2018)(Beer et al. 2020), …

Problem Formulation



input state

output

- Parametrized circuit as a unitary operator $U(\vec{a})$.
- Fixed measurement at the end: $\mathcal{M} = \{M_{\hat{y}} : \hat{y} \in \mathcal{Y}\}$
- Quantum states with classical attributes
  - Training samples: $\{(\rho_i, y_i)\}_{i=1}^{n}$ generated randomly
    - ρ_i: state of d qubits ➔ dim $= 2^d$
    - y_i: the true outcome
- Expected Loss:

$$L(\vec{a}) = \sum_{y,\rho} D(y,\rho) \sum_{\hat{y}} \ell(y,\hat{y})\, tr\{M_{\hat{y}}\, U(\vec{a})\rho U(\vec{a})^{\dagger}\}$$

loss function

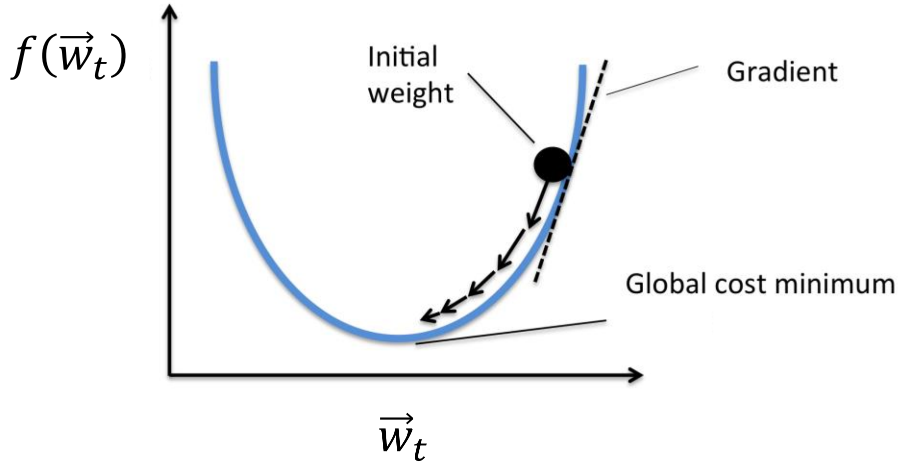
Probability of $\hat{Y}$
given the sample

$$\vec{a}^* = \underset{\vec{a}\in\mathcal{D}}{\operatorname{argmin}}\, L(\vec{a})$$

- Example: misclassification probability

Gradient-Based Methods

- Iterative method to find the local minimum
  $f: \mathbb{R}^n \rightarrow \mathbb{R}$
- If f is convex local min = global min
- Studied extensively even for non-convex functions.



Gradient Descent:

1. Initial point $\vec{w}_0$
2. Gradient at each step $\nabla f(\vec{w}_t)$
3. Update rule
   $$\vec{w}_{t+1} = \vec{w}_t - \eta_t \nabla f(\vec{w}_t)$$
4. Go to step 2

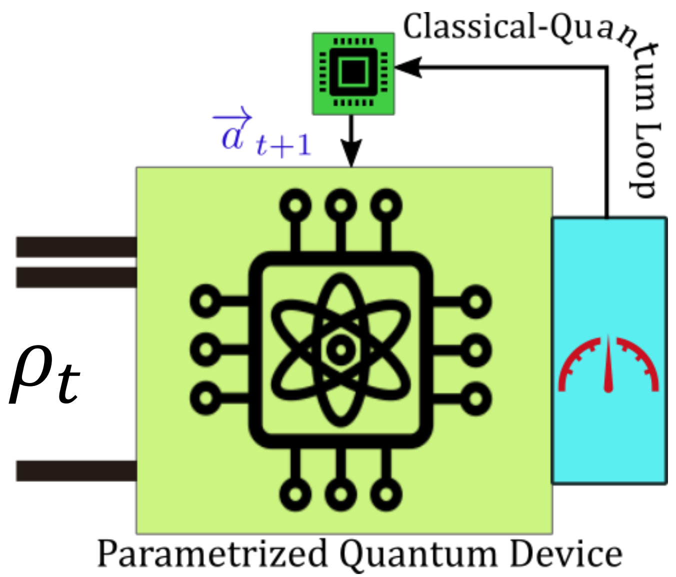
Gradient Descent in Quantum

- Quantum operation $U(\vec{a})$ with parameters $\vec{a} = (a_1, a_2, \dots, a_c)$
- Expected loss at iteration t:

$$L(\vec{a}, \rho_t, y_t) = \sum_{\hat{y}} \ell(y_t, \hat{y})\, tr\{M_{\hat{y}}\, U(\vec{a}) \rho_t U(\vec{a})^\dagger\}$$

- Update rule if gradient was known:

$$\boxed{\vec{a}_{t+1} = \vec{a}_t - \eta_t \nabla L(\vec{a}_t, \rho_t, y_t)}$$



But $L(\vec{a}, \rho, y)$ is unknown!

1) We cannot "see" the samples (state-collapse)
2) The loss is random (stochasticity of quantum measurements)

Derivative of the loss

- Parametrized unitary:

$$U(\vec{a}) = \exp\{i \sum_s a_s \sigma^s\}$$

- Expected Loss:

$$L(\vec{a}, \rho, y) = \sum_{\hat{y}} \ell(y, \hat{y})\, tr\{M_{\hat{y}}\, U(\vec{a}) \rho_y U(\vec{a})^\dagger\}$$

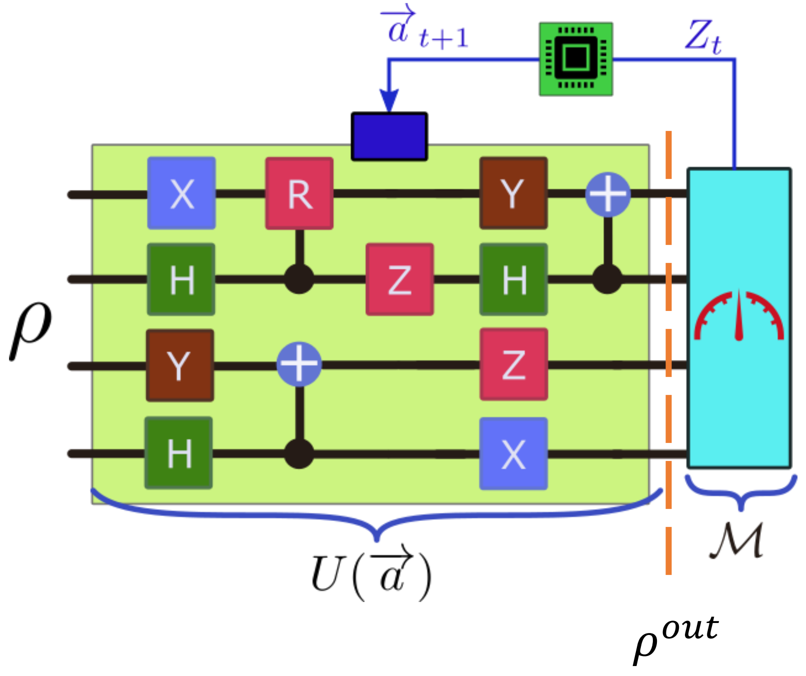
- Derivative of the loss:

$$\frac{\partial L(\vec{a}, \rho, y)}{\partial a_s} = \sum_{\hat{y}} \ell(y, \hat{y})\, tr\left\{M_{\hat{y}} \frac{\partial\left(U(\vec{a}) \rho U(\vec{a})^\dagger\right)}{\partial a_s}\right\}$$

$$= \sum_{\hat{y}} \ell(y, \hat{y})\, tr\{M_{\hat{y}}\, i\left(\sigma^s U(\vec{a}) \rho U(\vec{a})^\dagger - U(\vec{a}) \rho U(\vec{a})^\dagger \sigma^s\right)\}$$

$$= \sum_{\hat{y}} \ell(y, \hat{y})\, tr\{M_{\hat{y}}\, (\sigma^s \rho^{out} - \rho^{out} \sigma^s)\}$$

Unknown!



- State ρ is unknown
- Output is random
- Asymmetric

Gradient-based training

- Approximations with several copies of each sample (Farhi and Neven '18, Rebentrost et al., '18).
- $O(\frac{c \log c}{\epsilon^2})$ copies per-sample needed for approximation error up to ϵ.
- $O(\frac{Tc \log c}{\epsilon^2})$ copies for training a device with c parameters in T iterations.
- Fewer copies? What about no-cloning?

Our work:

- Can we do the gradient-based training without copying states?
- Yes: designing a circuit to measure the derivative
- (MH, Grama and Szpankowski): Randomized QSGD with $O(T)$ samples without the need for exact copies.

Idea: one-shot measurement

Step 1: Qubit Embedding: $\rho^{out} \otimes |+\rangle\langle+|$

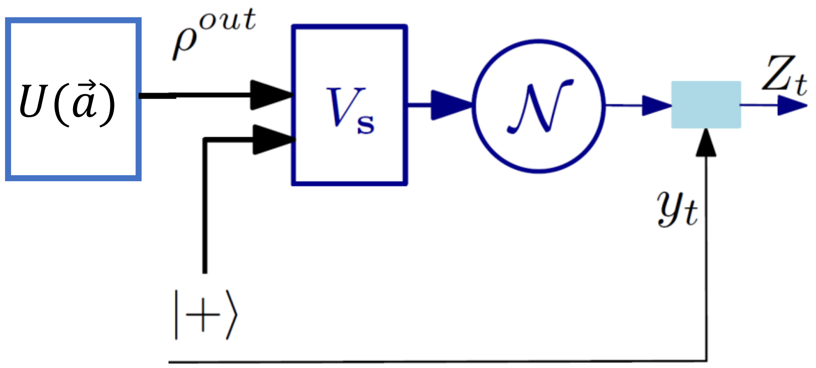
$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\frac{\partial L(\vec{a},\rho,y)}{\partial a_s} = \sum_{\hat{y}} \ell(y,\hat{y})\, tr\{M_{\hat{y}}(\sigma^s\rho^{out} - \rho^{out}\sigma^s)\}$$

Step 2: Controlled Rotations

$$V_s = \exp\left\{\frac{i\pi}{4}\sigma^s\right\} \otimes |0\rangle\langle 0| + \exp\left\{-\frac{i\pi}{4}\sigma^s\right\} \otimes |1\rangle\langle 1|$$

Output qubit



Step 3: Measurement:

$$\mathcal{N} = \mathcal{M} \otimes \{|0\rangle\langle 0|, |1\rangle\langle 1|\} \Longrightarrow \text{Outcome: } (\tilde{y}, b)$$

Step 4: Classical processing:

$$z_t = -2(-1)^b \ell(y_t, \tilde{y})$$

Circuit for measuring the derivative of per-sample loss

Idea

Lemma: The derivative measurement is unbiased

$$\mathbb{E}[Z|\rho, y] = \frac{\partial L(\vec{a},\rho,y)}{\partial a_s}$$

What about Gradient?

- Randomized approach
- Each time randomly select a component of $\vec{a} = (a_1, a_2, \dots, a_c)$, say a_s
- Measure the derivative and create a vector $\vec{Z} = (0,0,\dots,0,Z_t,0,\dots 0)$

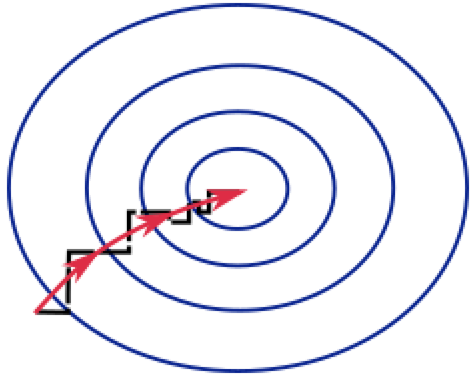
Theorem: The gradient measurement satisfies

$$\mathbb{E}\left[\vec{Z}\middle|\rho, y\right] = \frac{1}{c}\nabla L(\vec{a}, \rho, y)$$

Update Rule

- One-shot gradient update:

$$\vec{a}_{t+1} = \vec{a}_t - \eta_t \vec{Z}_t$$



- Not an accurate estimate of the gradient at each point.
- But pushing the system statistically in the direction of the gradient over a time interval.
- Potentially handle any other unbiased sources of noise or randomness (NISQ)

Theorem (Convergence Rate): Suppose that the loss function is bounded by $\gamma \in \mathbb{R}$ and is convex. Then, after T iterations of the randomized QSGD with learning rate $\eta = \frac{1}{2\gamma\sqrt{T}}$:

$$|\mathbb{E}[L(\vec{a}_{ave}, \rho, y)] - L(\vec{a}^*)| \leq \frac{2\gamma c}{\sqrt{T}}$$

where $\vec{a}_{ave} = \frac{1}{T}\sum_t \vec{a}_t$ and c is the number of parameters.

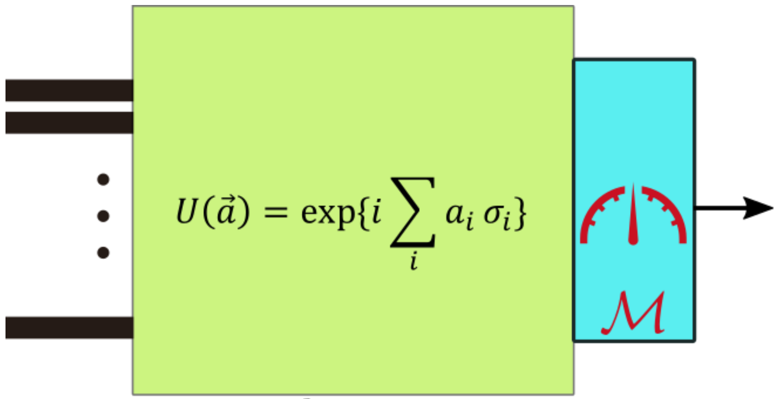
Comparison to Gradient Approximation

Comparison for a fixed number of sample/copies, say n.

- c: number of parameters.
- The randomized QSGD has excess loss $O(\frac{c}{\sqrt{n}})$.
- The gradient approximation algorithm using exact copies has excess loss $O\left(\frac{\sqrt{c \log c}}{\epsilon\sqrt{n}}\right)$, where $\epsilon \ll 1$.
- ➔ Faster convergence with randomized QSGD when $\frac{\log c}{c} = O(\epsilon^2)$

Numerical Results

- Binary classification of quantum states
- Dataset: (Mohseni,Steinberg, and Bergou 2004)(Chen et al. 2018)(Patterson et al. 2021)(Li, Song, and Wang 2021)
- Pure vs Mixed State
  - Pure states with label $y = 0$:
    $$\rho_0(u) = |\phi_u\rangle\langle\phi_u|, \qquad u \sim unif([0,1])$$
  - Mixed states with label $y = 1$:
    $$\rho_1(v) = \frac{1}{2}(|\psi_{+v}\rangle\langle\psi_{+v}| + |\psi_{-v}\rangle\langle\psi_{-v}|), \qquad v \sim unif([0,1])$$

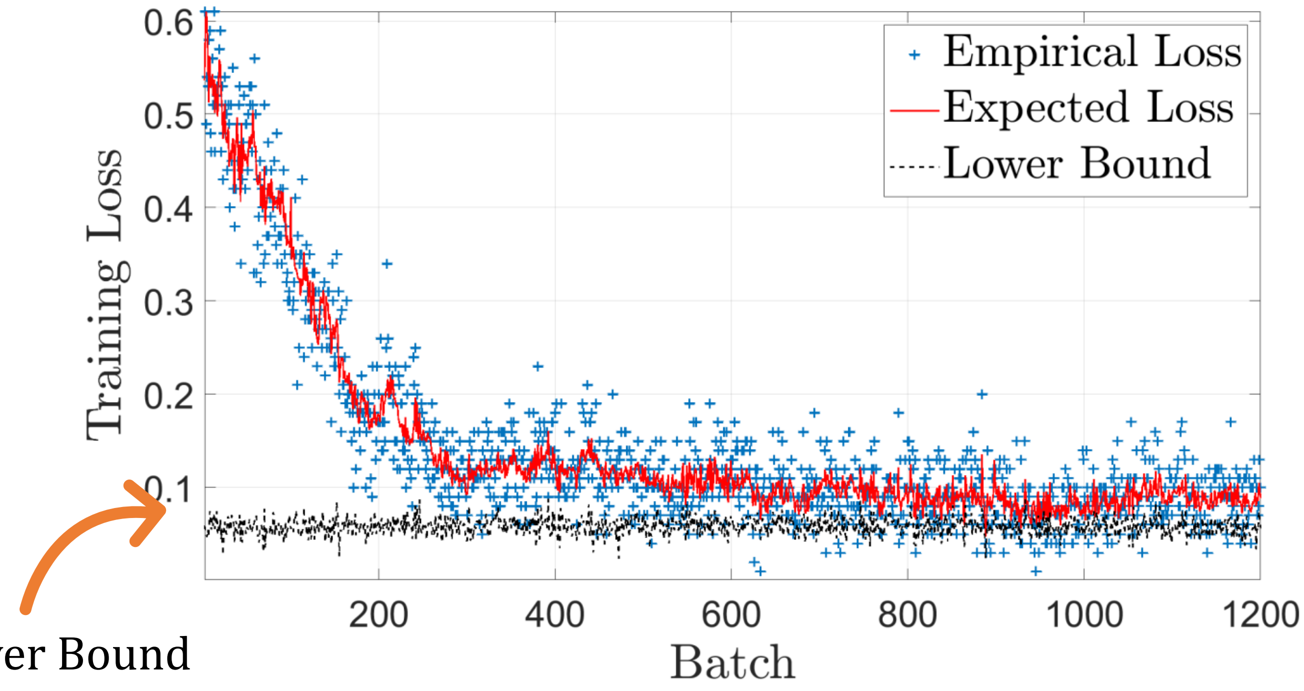


Parametrized Quantum Circuit

- Each sample is either:
  - $\rho_0(u)$, with prob $p = \frac{1}{3}$
  - $\rho_1(v)$, with prob $(1 - p) = \frac{2}{3}$

Numerical Results

Randomized QSGD with gradient measuring

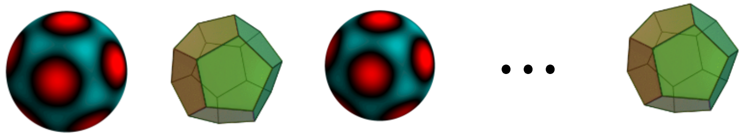


Holevo–Helstrom Lower Bound

	QSGD	Known gradient	Lower-bound
Acc	$91\% \pm 1$	$93.48\% \pm 1$	$93.51\% \pm 1$

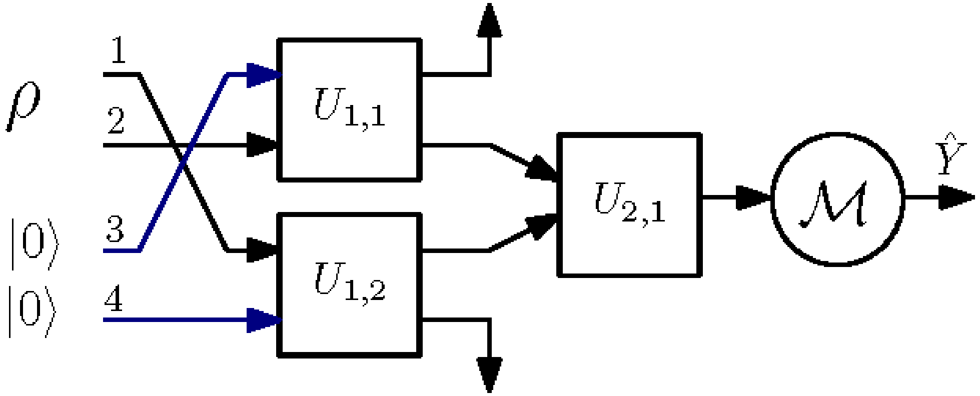
Entangled vs Separable

ρ_i:



y_i: 0 1 0 … 1

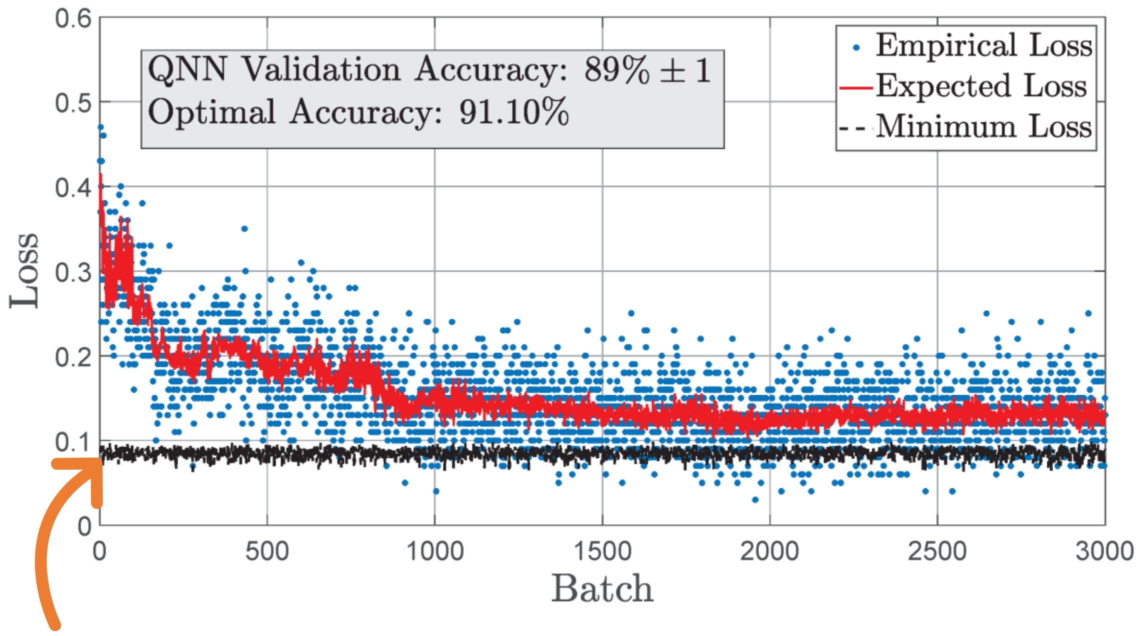
- A random unknown quantum state generated that is either:
  - ρ_0: *maximally entangled* qubit, generated with probability $p = \frac{1}{2}$
  - ρ_1: *separable* qubit, generated with probability $(1-p) = \frac{1}{2}$



QNN

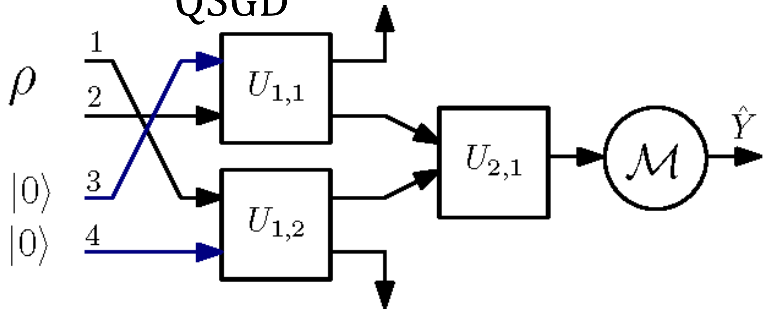
$$U_{\ell,j}\left(\vec{a}_{(\ell,j)}\right) = \exp\{i(a_0 I + a_1 X + a_2 Y + a_3 Z)\}$$

Numerical Results



Holevo–Helstrom
Theoretical Lower Bound

Randomized
QSGD



Outline:

Part 1: Introduction

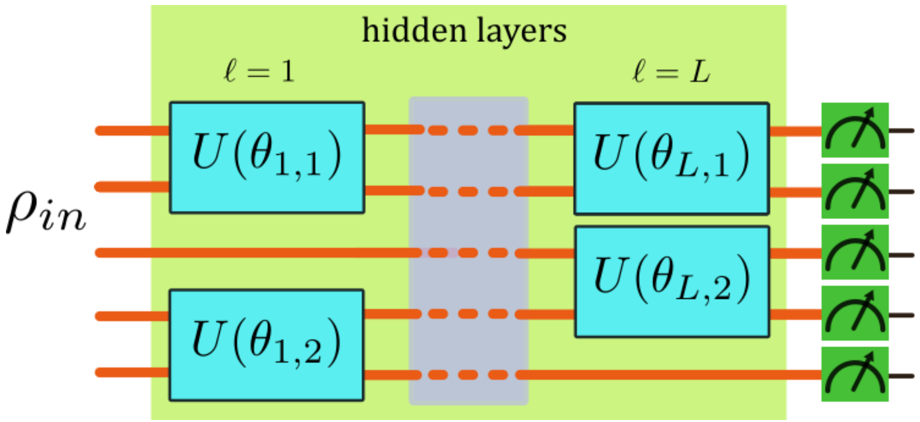
Part 2: Learning with quantum computers

Part 3: Band-limited QNNs

QNNs

Challenges:

- Scalability & exponential parameters as $dim = 2^q$
- Barren Plateau issue: certain random QNNs structures do not lead to learning as gradient vanishes everywhere.



This Work:

- New approach: band-limited QNNs
- Controlling the number of parameters with bandwidth limitation.
- Training with randomized QSGD

The Notion of Bandwidth

- “Bandwidth” in classical learning:
  - Based on the Fourier expansion for functions on the Boolean cube.
  - Studied in computational learning (O’Donnell ‘14), (Wolf, ‘08), …
  - Characterizing the non-linear complexity of Boolean functions.
  - A proxy to derive learning-theoretic bounds.
  - Supervised and unsupervised feature selection [ICMl’21, ISIT’21]
  - Information theory [ISIT’19]

Boolean Fourier Expansion

Standard Fourier on the Boolean Cube:

Any bounded $g: \{+1, -1\}^d \to \mathbb{R}$ is uniquely written as:

$$g(\boldsymbol{x}) = \sum_{\mathcal{S} \subseteq [\boldsymbol{d}]} g_{\mathcal{S}}\, \boldsymbol{\chi}_{\mathcal{S}}(\boldsymbol{x})$$

Fourier Coefficients:

$$g_{\mathcal{S}} = \frac{1}{2^d} \sum_{x} g(\boldsymbol{x})\, \boldsymbol{\chi}_{\mathcal{S}}(\boldsymbol{x})$$

Monomials:

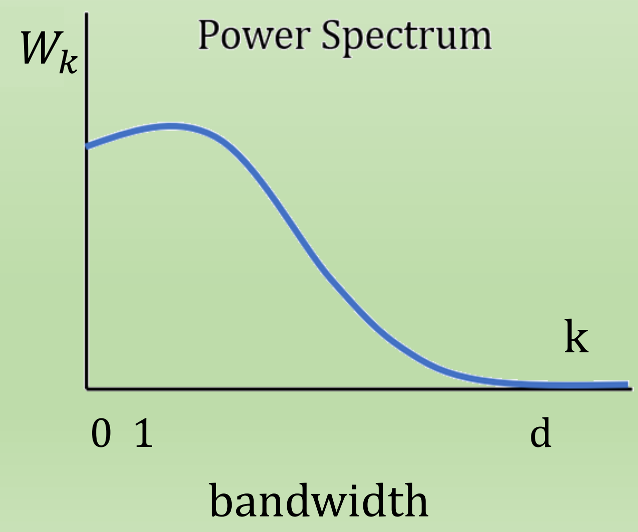
$$\boldsymbol{\chi}_{\mathcal{S}} = \prod_{j \in \mathcal{S}} x_j$$

E.g., Logical $\boldsymbol{OR}$ on two $\{\pm 1\}$ bits:

$$OR(x_1, x_2) = \frac{1}{2} + \frac{1}{2} x_1 + \frac{1}{2} x_2 - \frac{1}{2} x_1 x_2$$

- Bandwidth for a Boolean function?
- Influence of k-element groups of inputs:

$$W_k = \sum_{\mathcal{S}:\text{k-element}} g_{\mathcal{S}}^2$$

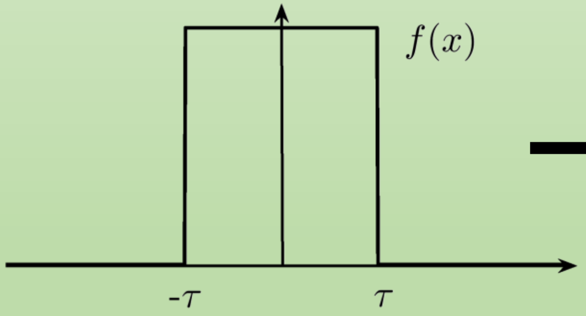


Comparison with the Fourier Transform

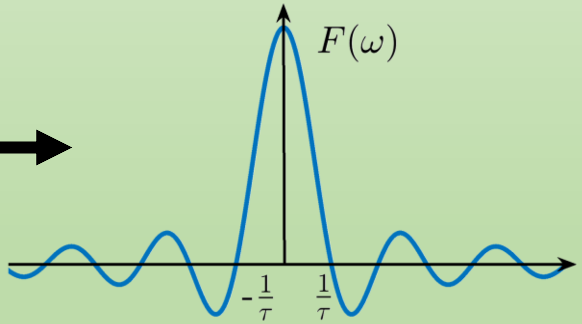
How does the concept of bandwidth different from the standard Fourier analysis?

Classical Fourier Transform

longer in time ⟺ Narrower bandwidth



Rectangular pulse



Spectrum

Fourier Expansion on Boolean Cube

Narrower bandwidth ⟺ influenced by smaller group of inputs

$$f(x^d) = x_1$$

$$g(x^d) = x_1 \oplus x_2 \oplus \cdots \oplus x_d$$

New Connections to Machine Learning

- Machine learning with *feature selection:*
  - Selecting a subset of features in the dataset while maintaining the same level of accuracy.
- A Fourier measure of features' redundancy and relevancy to the label.

➔ Our work: probabilistic Fourier expansion:

Gram-Schmidt-Type Orthogonalization [ISIT'21]:

- Creating an empirically orthogonal basis.
- A measure of the effective dimension of dataset.

$$\tilde{\psi}_{\mathcal{S}_i} \equiv \chi_{\mathcal{S}_i} - \sum_{j=1}^{i-1} \langle \psi_{\mathcal{S}_j}, \chi_{\mathcal{S}_i} \rangle_D \, \psi_{\mathcal{S}_j},$$

$$\psi_{\mathcal{S}_i} \equiv \begin{cases} \frac{\tilde{\psi}_{\mathcal{S}_i}}{\|\tilde{\psi}_{\mathcal{S}_i}\|_{2,D}} & \text{if } \|\tilde{\psi}_{\mathcal{S}_i}\|_{2,D} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Probabilistic Fourier Expansion

- Binary input with arbitrary distribution $\boldsymbol{X}^d \sim D_X$.

$$g(\boldsymbol{x}) = \sum_{\mathcal{S} \in \mathcal{T}} g_{\mathcal{S}} \, \psi_{\mathcal{S}}(\boldsymbol{x})$$

non-redundant features

Orthogonalized Parities

$$g_{\mathcal{S}} = \mathbb{E}_D[g(\boldsymbol{X})\psi_{\mathcal{S}}(\boldsymbol{X})]$$

Unsupervised FS
Orthogonalization Algorithm

[ISIT '19, 21]

Supervised FS
Estimate $\hat{g}_{\mathcal{S}}$
SFFS Algorithm

[ICML '21]

Numerical Results

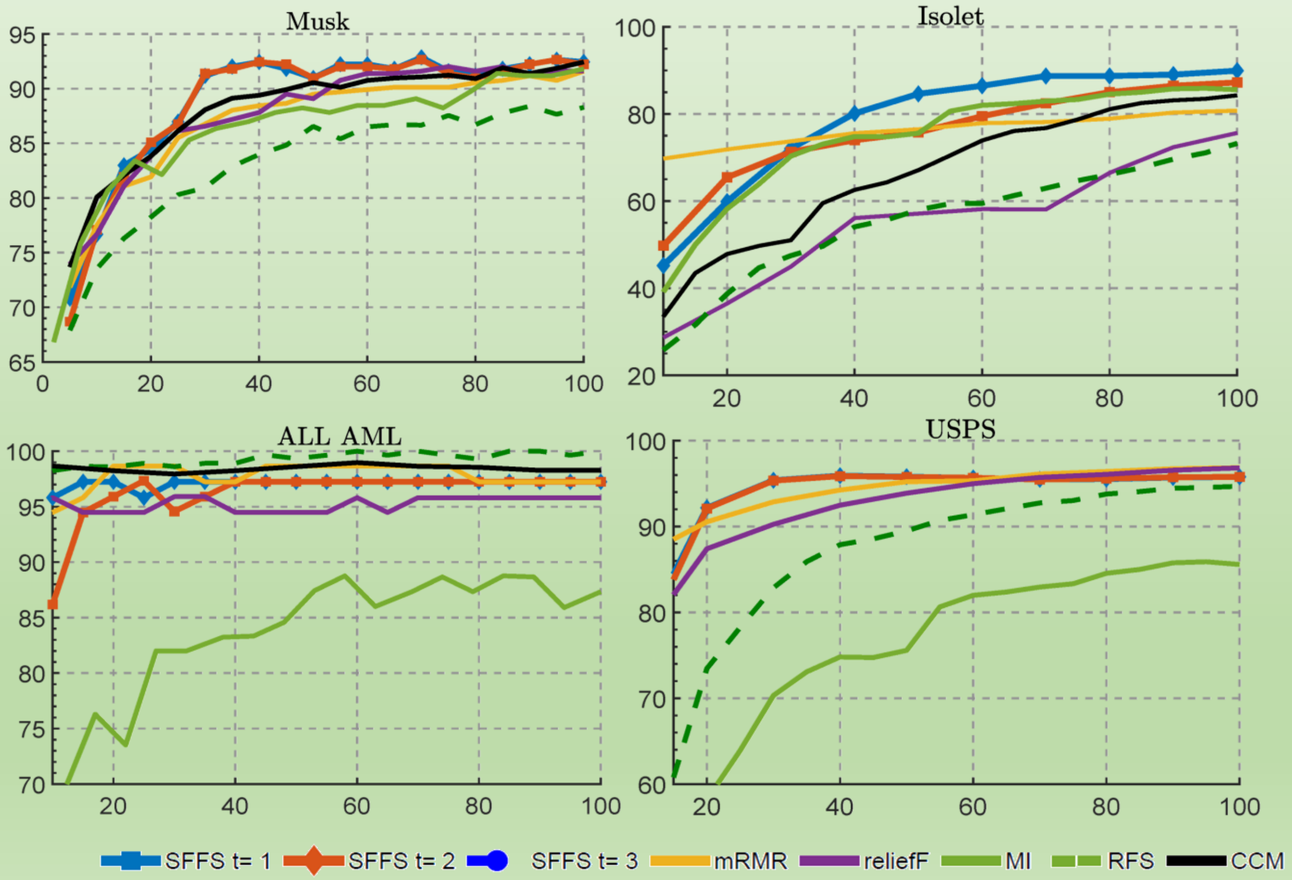
Unsupervised Feature Selection:

Data set	USPS	Lung	Covertype	Australian	Musk	ALL AML
Features	256	3312	54	14	166	7128
Samples	9298	203	581	690	467	72

	S1	S2	S3	USPS	Covertype	Australian	Musk	ALL AML	Lung
No FS	77.9	75.0	87.0	97.3	75.6	84.9	92.2	94.3	94.6
UFFS k	11	12	11	93	34	12	35	39	114
UFFS	**80.3**	**76.8**	**86.2**	**97.0**	**76.9**	**85.1**	**85.7**	**97.1**	**94.6**
LS	55.1	61.2	71.0	95.6	72.8	85.4	84.5	97.2	93.6
MCFS	56.6	59.0	65.8	93.9	72.3	84.8	84.2	95.9	94.1
UDFS	64.0	60.6	64.3	80.8	72.0	84.9	80	86.2	92.6

Validation acc. on a SVM with radial kernel.

Supervised Feature Selection:



Back to The Quantum

Classical Fourier Transform

$$f(x) = \int F(\omega) e^{2\pi i x \omega}\, d\omega$$

$$F(\omega) = \int f(x) e^{-2\pi i x \omega}\, dx$$

Fourier Expansion on Boolean Cube

$$g(x^d) = \sum_{S \subseteq [d]} g_S\, \psi_S(x^d)$$

$$g_S = \mathbb{E}_D\left[g(X^d)\psi_S(X^d)\right]$$

Quantum Fourier Transform

- Shor's algorithm

$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk}\, |k\rangle$$

Pauli Decomposition

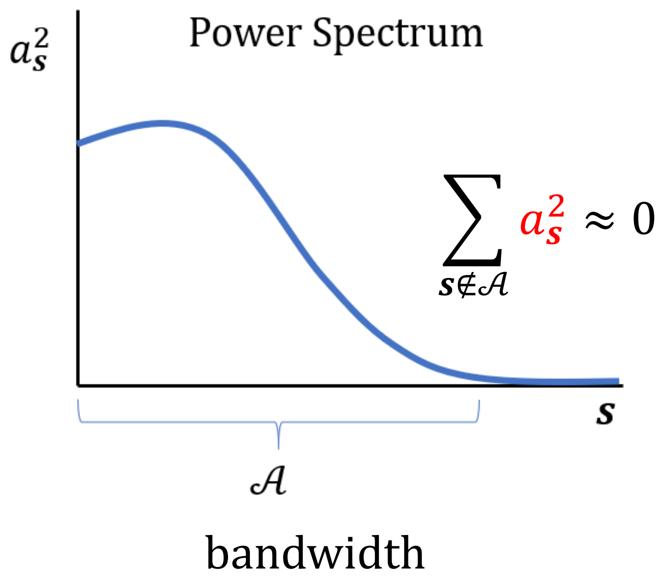
Elementary Pauli operators (with the identity):

$$\sigma^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \sigma^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma^2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma^3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Any operator A on the Hilbert space of d qubits admits the decomposition

$$A = \sum_{s \in \{0,1,2,3\}^d} a_s \, \sigma^{s_1} \otimes \sigma^{s_2} \otimes \cdots \otimes \sigma^{s_d}$$

where $a_s \in \mathbb{C}$ are the Pauli/Fourier coefficients and given by $a_s = \frac{1}{2^d} tr\{A\sigma^s\}$.



Back to the Quantum

Classical Fourier

$$f(x) = \int F(\omega) e^{2\pi i x \omega}\, d\omega$$

$$F(\omega) = \int f(x) e^{-2\pi i x \omega}\, dx$$

Fourier Expansion on Boolean Cube

$$g(x^d) = \sum_{\mathcal{S} \subseteq [d]} g_{\mathcal{S}}\, \psi_{\mathcal{S}}(x^d)$$

$$g_{\mathcal{S}} = \mathbb{E}_D\left[g(X^d)\psi_{\mathcal{S}}(X^d)\right]$$

Quantum Fourier Transform

- Shor’s algorithm

$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk}\, |k\rangle$$

Pauli Decomposition

$$A = \sum_{\boldsymbol{s}} a_s\, \sigma^{\boldsymbol{s}}$$

$$a_{\boldsymbol{s}} = \frac{1}{2^d}\, tr\{A\sigma^{\boldsymbol{s}}\}$$

Band-limited QNN

- Each QP is of the form $U(a) = \exp\{iA\}$ acting on a small subsystem.
- Coordinate of that subsystem: $\mathcal{J}$
- ➔ the Fourier expansion of A is zero outside of $\mathcal{J}$:

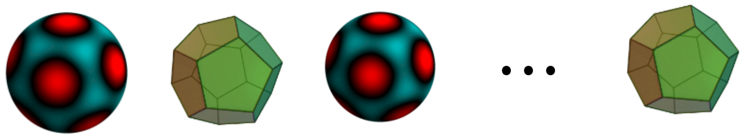
$$A = \sum_{\boldsymbol{s}: s_j = 0, \forall j \notin \mathcal{J}} a_{\boldsymbol{s}}\, \sigma^{s_1} \otimes \sigma^{s_2} \otimes \cdots \otimes \sigma^{s_d}$$

- Bandwidth parameter: $|\mathcal{J}| \leq k$.
- Number of parameters: $c_{QNN} = m4^k$, with m number of QPs.
- Expressive power of band-limited QNNs:

Theorem (AAAI'22): With $k = 2$ and enough QPs any quantum measurement (predictor) can be implemented.

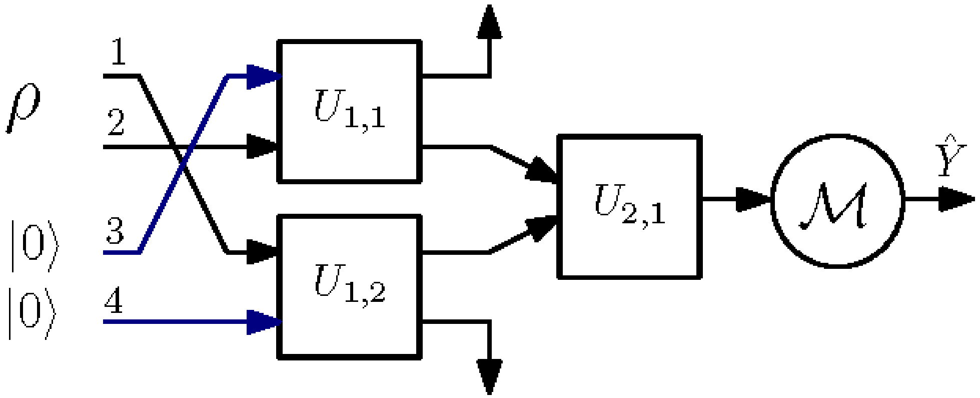
Entangled vs Separable

ρ_i:



y_i: 0 1 0 ⋯ 1

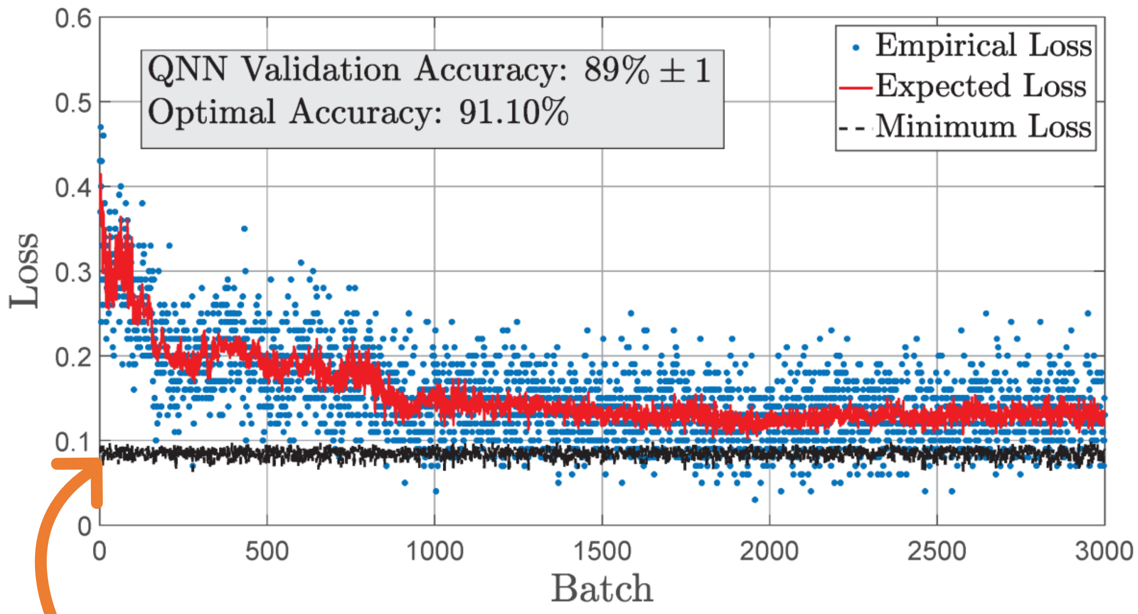
- A random unknown quantum state generated that is either:
  - ρ_0: *maximally entangled* qubit, generated with probability $p = \frac{1}{2}$
  - ρ_1: *separable* qubit, generated with probability $(1-p) = \frac{1}{2}$



QNN with $k = 2, m = 3$

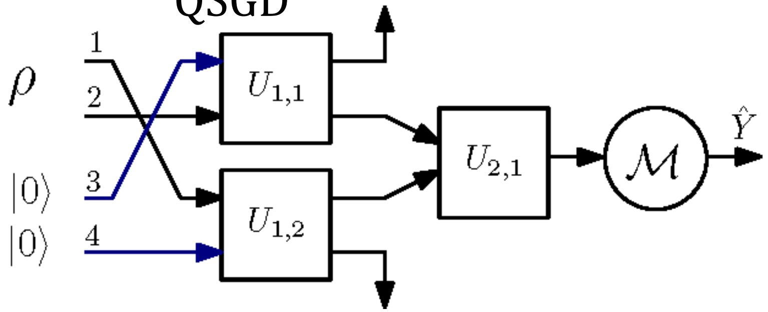
$$U_{\ell,j}\left(\vec{a}_{(\ell,j)}\right) = \exp\{i(a_0 I + a_1 X + a_2 Y + a_3 Z)\}$$

Numerical Results



Holevo–Helstrom
Lower Bound

Randomized
QSGD



$$\vec{a}_{t+1} = \vec{a}_t + \frac{0.77}{\sqrt{t}} \vec{Z}_t$$