Quantum Machine Learning

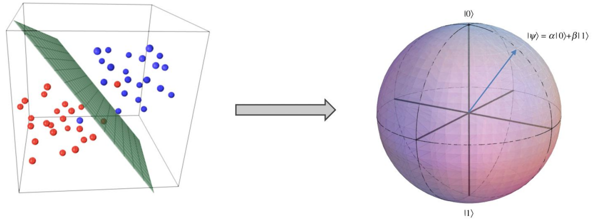
TQCI Seminar

Jonas Landman
Postdoctoral Research Associate - University of Edinburgh / LIP6

Why Quantum Machine Learning?

Can quantum algorithms solve machine learning problems?

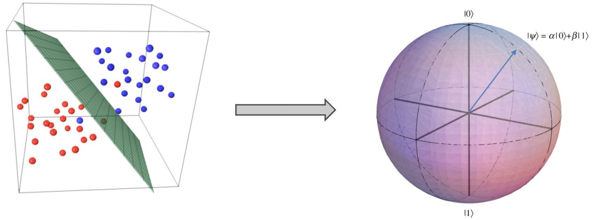
Provable/comparable guarantees?
Speedup or any other advantage?
Control over quantum effects?



Why Quantum Machine Learning?

Can quantum algorithms solve machine learning problems?

Provable/comparable guarantees?
Speedup or any other advantage?
Control over quantum effects?

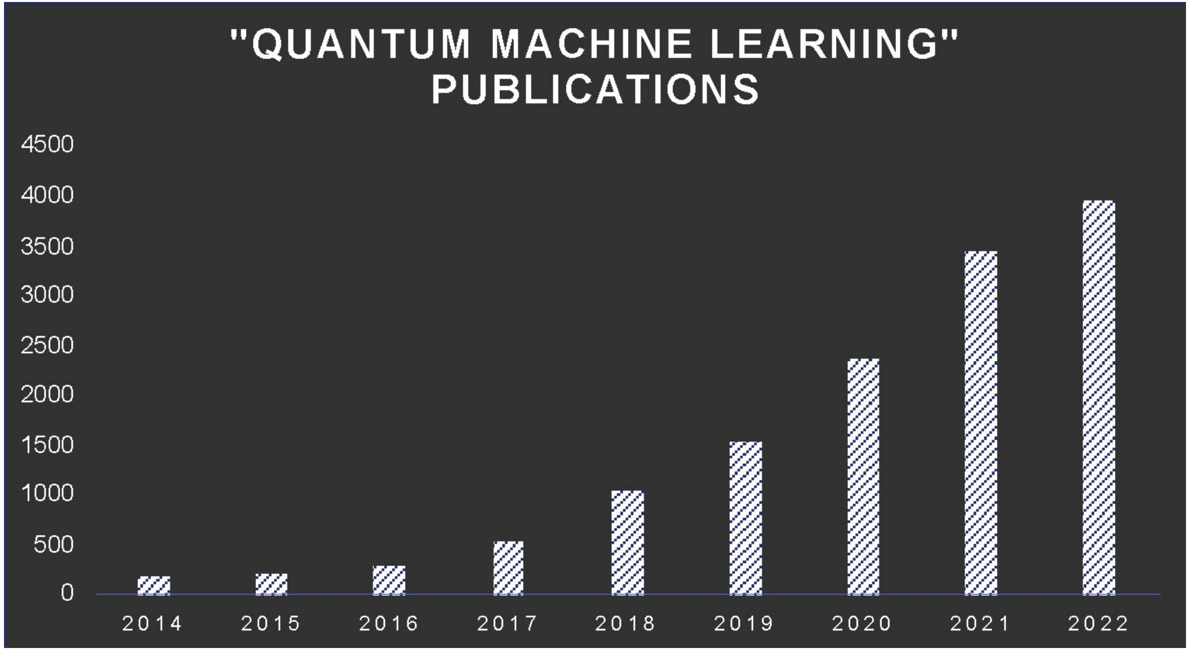


There is a strong link between ML and QC : Linear Algebra

There are different approaches to QML : Long Term & Short Term

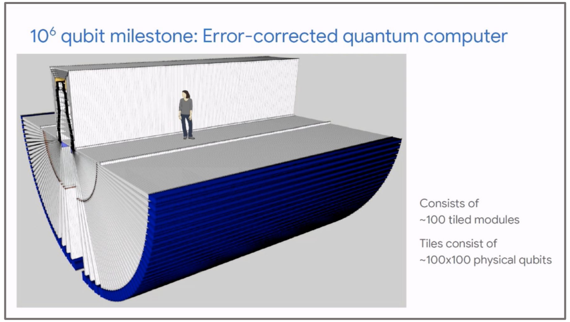
A lot of theory is still needed to understand Advantages and Caveats

Research publications



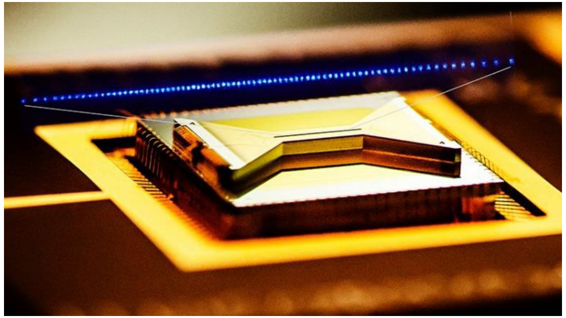
Long Term vs Near Term

Long Term



Google Quantum©

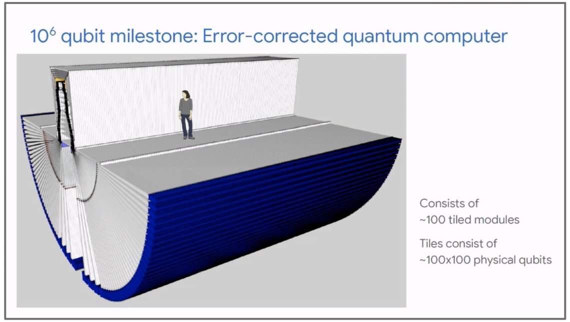
Near Term



IonQ©

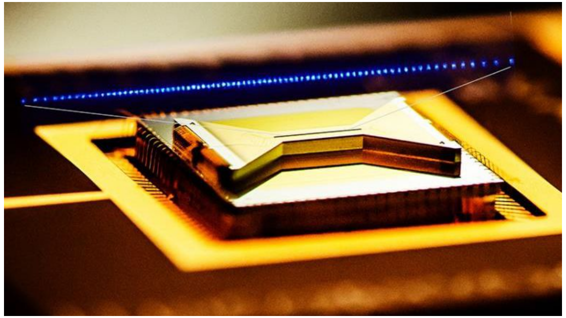
Long Term vs Near Term

Long Term



Google Quantum©

Near Term



IonQ©

In the long term, Quantum provide a theoretical advantage

- **Matrix Inversion** (Ax = b)
- **Linear Algebra** (SVD, projections, inner product)
- **Topology** (distance estimation) **etc.**

Quantum Machine Learning will be provably faster

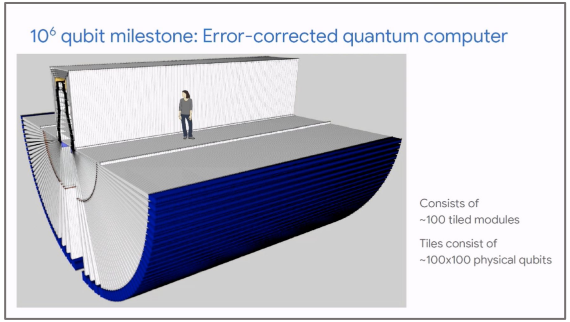
- **Clustering, Neural Networks**
- **Recommendation Systems, SVM, etc.**

Many Requirements

- **Loading Data (QRAM), Error Correction, De-quantization**

Long Term vs Near Term

Long Term



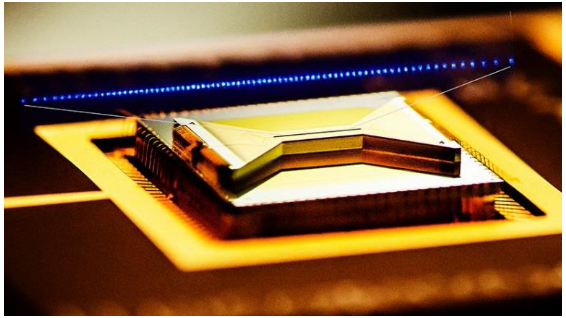
Google Quantum©

In the long term, Quantum provide a theoretical advantage
- **Matrix Inversion** (Ax = b)
- **Linear Algebra** (SVD, projections, inner product)
- **Topology** (distance estimation) **etc.**

Quantum Machine Learning will be provably faster
- **Clustering, Neural Networks**
- **Recommendation Systems, SVM, etc.**

Many Requirements
- **Loading Data (QRAM), Error Correction, De-quantization**

Near Term



IonQ©

In the near term, several approaches exist

Variational Quantum Circuits are used
- Require **classical optimization** of quantum gates
- Project data in **large feature space**
- **Not many proof** of advantage
- **Gradients** are vanishing
- **Expressivity** could be reproduced classically

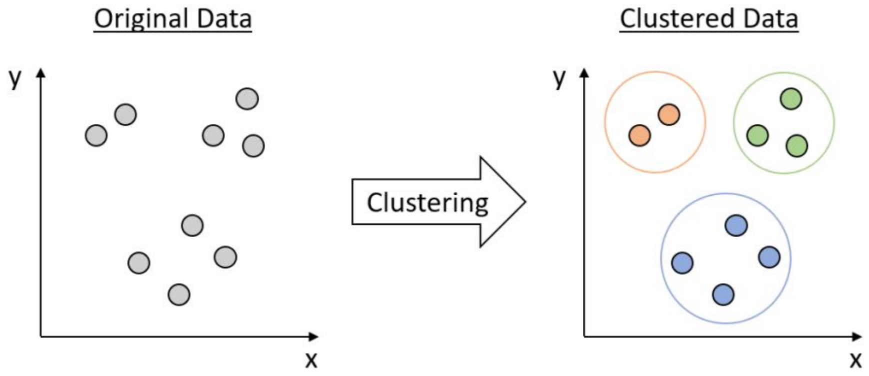
Easy to implement
Unclear Scaling
Harder to interpret

Long Term QML

Long Term QML : A Recipe

Long Term QML : A Recipe

- Choose your favorite classical ML algorithm



Long Term QML : A Recipe

- Choose your favorite classical ML algorithm
- Understand it at the linear algebra level

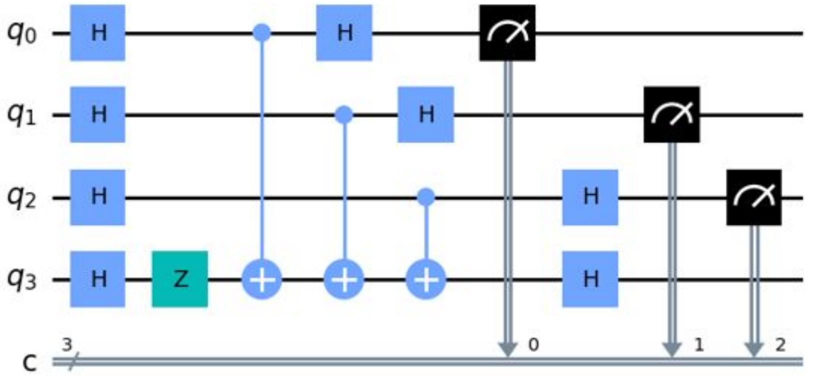
Input: The data set $Y = \{y_s\}_{s=1}^{S} \subset \mathbb{R}^M$
Input: Number of clusters : K
Output: Clusters: $\{Y_k^{(i)}\}_{k=1}^{K}$
Output: Cluster labels array: $L[1 : K]$
// Initialization
$i \leftarrow 0$; // Iteration counter
foreach $k \in \{1, \ldots, K\}$ **do**
  // for k-th cluster
  Initialize $\mu_k^{(0)}$ and $\Sigma_k^{(0)}$ suitably ; Set $Y_k^{(0)}$ to empty set ;
end
repeat
  // Segmentation:
  foreach $y_s \in Y$ **do**
    $L(k) \leftarrow l \leftarrow \arg\min_{k=1,\ldots,K} \|y_s - \mu_k^{(i)}\|^2_{\Sigma_k^{(i)}}$; assign y_s to $Y_l^{(i+1)}$;
  end
  // Estimation:
  foreach $k \in \{1, \ldots, K\}$ **do**
    $\mu_k^{(i+1)} \leftarrow \text{mean}(Y_k^{(i+1)})$;
    $\Sigma_k^{(i+1)} \leftarrow \text{cov}(Y_k^{(i+1)})$;
  end
  $i \leftarrow i + 1$;
until *the segmentation has stopped changing*;

Long Term QML : A Recipe

- Choose your favorite classical ML algorithm

- Understand it at the linear algebra level

- Imagine a quantum circuit that
    - Loads the **data**
    - Perform the **same operations**
    - **Retrieve** the results from the Q state



Assuming many qubits
+ large depth circuits
+ error correction !

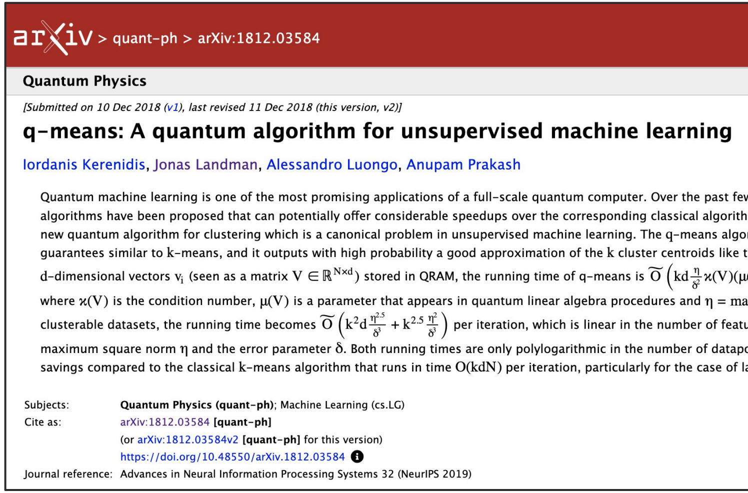
Long Term QML : A Recipe

- Choose your favorite classical ML algorithm

- Understand it at the linear algebra level

- Imagine a quantum circuit that
  - Loads the **data**
  - Perform the **same operations**
  - **Retrieve** the results from the Q state

- Compare "speed" of C vs. Q
  - If good : Claim "**Q Advantage**" but be honest about all the issues



arXiv > quant-ph > arXiv:1812.03584

Quantum Physics

[Submitted on 10 Dec 2018 (v1), last revised 11 Dec 2018 (this version, v2)]

q-means: A quantum algorithm for unsupervised machine learning

Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, Anupam Prakash

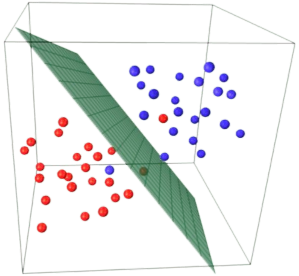
Quantum machine learning is one of the most promising applications of a full-scale quantum computer. Over the past fev algorithms have been proposed that can potentially offer considerable speedups over the corresponding classical algorith new quantum algorithm for clustering which is a canonical problem in unsupervised machine learning. The q-means algo guarantees similar to k-means, and it outputs with high probability a good approximation of the k cluster centroids like t d-dimensional vectors v_i (seen as a matrix $V \in \mathbb{R}^{N\times d}$) stored in QRAM, the running time of q-means is $\widetilde{O}\left(kd\frac{\eta}{\delta^2}\varkappa(V)(\mu$ where $\varkappa(V)$ is the condition number, $\mu(V)$ is a parameter that appears in quantum linear algebra procedures and $\eta =$ ma clusterable datasets, the running time becomes $\widetilde{O}\left(k^2 d\frac{\eta^{2.5}}{\delta^3} + k^{2.5}\frac{\eta^2}{\delta^3}\right)$ per iteration, which is linear in the number of featu maximum square norm η and the error parameter δ. Both running times are only polylogarithmic in the number of datapo savings compared to the classical k-means algorithm that runs in time $O(kdN)$ per iteration, particularly for the case of la

Subjects: **Quantum Physics (quant-ph)**; Machine Learning (cs.LG)
Cite as: arXiv:1812.03584 **[quant-ph]**
(or arXiv:1812.03584v2 **[quant-ph]** for this version)
https://doi.org/10.48550/arXiv.1812.03584
Journal reference: Advances in Neural Information Processing Systems 32 (NeurIPS 2019)

I.Kerenids, JL, A.Luongo, A.Prakash: "**q-means**" NeurIPS 2019

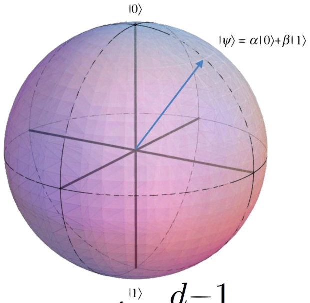
Long Term QML
Where does the advantage come from?

1) Load data via Amplitude Encoding



$$x \in \mathbb{R}^d$$

QRAM



$$|x\rangle = \frac{1}{\|x\|} \sum_{j=0}^{d-1} x_j \, |j\rangle$$

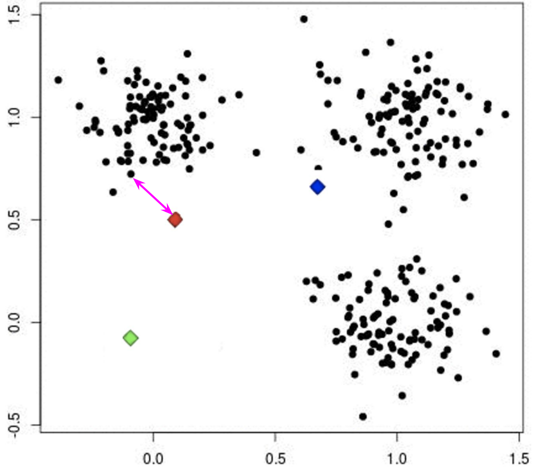
Exponentially small !

computational basis $|x\rangle = x_1|0\cdots01\rangle + x_2|0\cdots11\rangle + \cdots + x_d|11\cdots1\rangle$

$\log(d)$ qubits

Long Term QML
Where does the advantage come from?

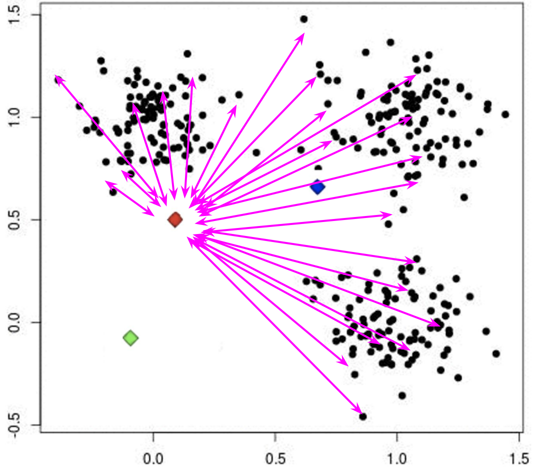
2) Compute in parallel



$d_1, \ d_2, \ d_3,$
$d_4, \ \ldots \ , d_N$

Classical values

C: Repeat N times



$|d_1\rangle + |d_2\rangle + |d_3\rangle +$
$|d_4\rangle + \ \ldots \ + |d_N\rangle$

Quantum state

Q: All in 1 time

Long Term QML : Main Issues

1) Loading the data in the quantum state is harder than you think

2) Extract the result from a quantum state is harder than you think

3) Quantum processes are random by nature, are you ok with it ?

commentary

Read the fine print

Scott Aaronson

New quantum algorithms promise an exponential speed-up for machine learning, clustering and finding patterns in big data. But to achieve a real speed-up, we need to delve into the details.

For twenty years, quantum computing has been catnip to science journalists. Not only would a quantum computer harness the notorious weirdness of quantum mechanics, but it would do so for a practical purpose: solving certain problems exponentially faster than we know how to solve them with any existing computer. But,

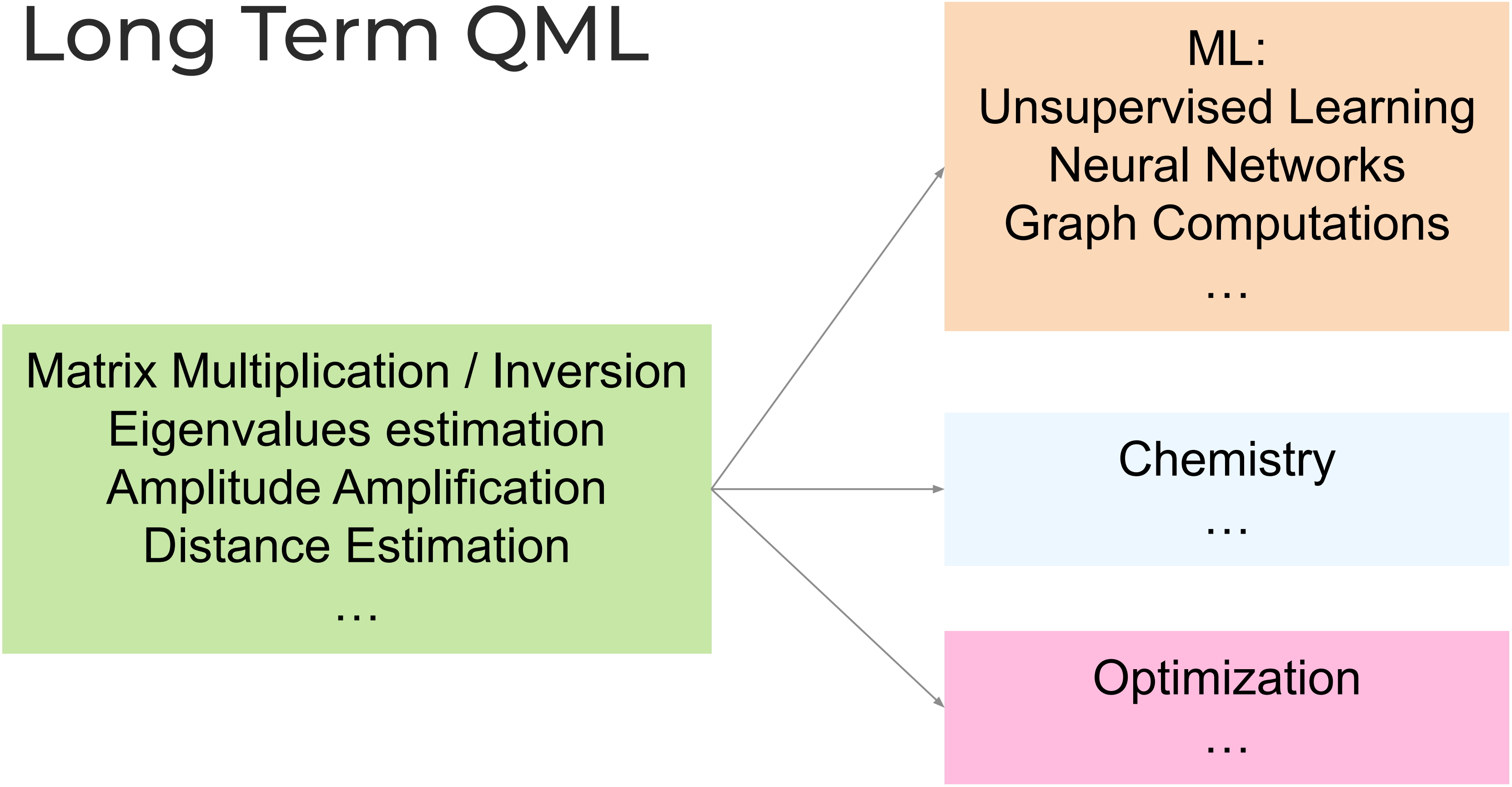
HHL attacks one of the most basic problems in all of science: solving a system of linear equations. Given an $n \times n$ real matrix, A, and a vector, $\mathbf{b}$, the goal of HHL is to (approximately) solve the system $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$, and to do so in an amount of time that scales only logarithmically with n, the number of equations and unknowns. Classically, this

of interest, and then carefully analyses the resulting performance against that of the best-known classical algorithm for that case.

To my knowledge, so far there have been two attempts to work out potential applications of the HHL template from start to finish. Clader, Jacobs, and Sprouse[5] argued that HHL could be used to speed up

Scott Aaronson: "**Read The Fine Print**" Nature Physics 2015

Long Term QML

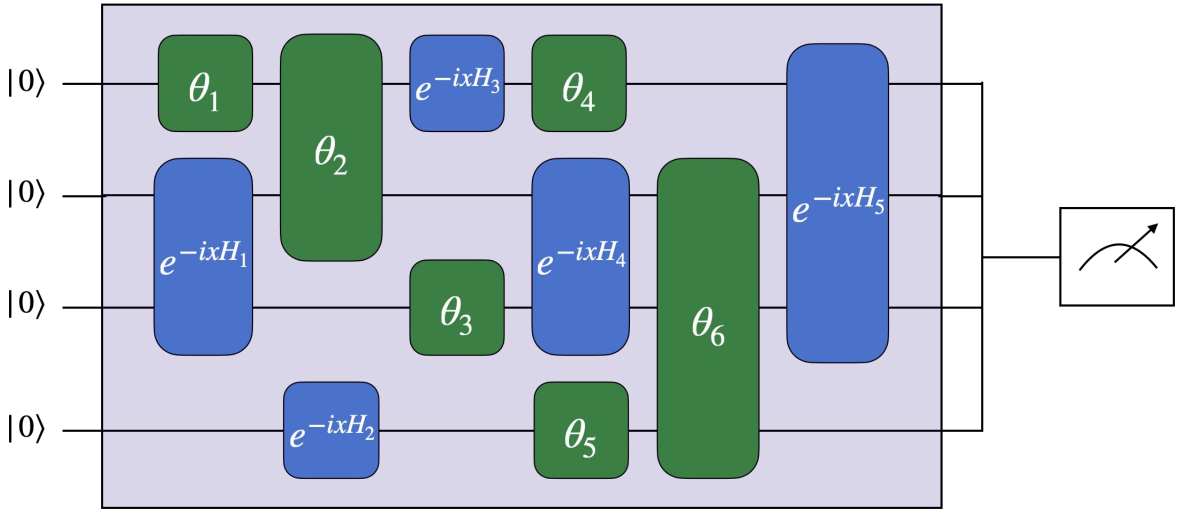


Near Term QML

Near Term QML : A Recipe

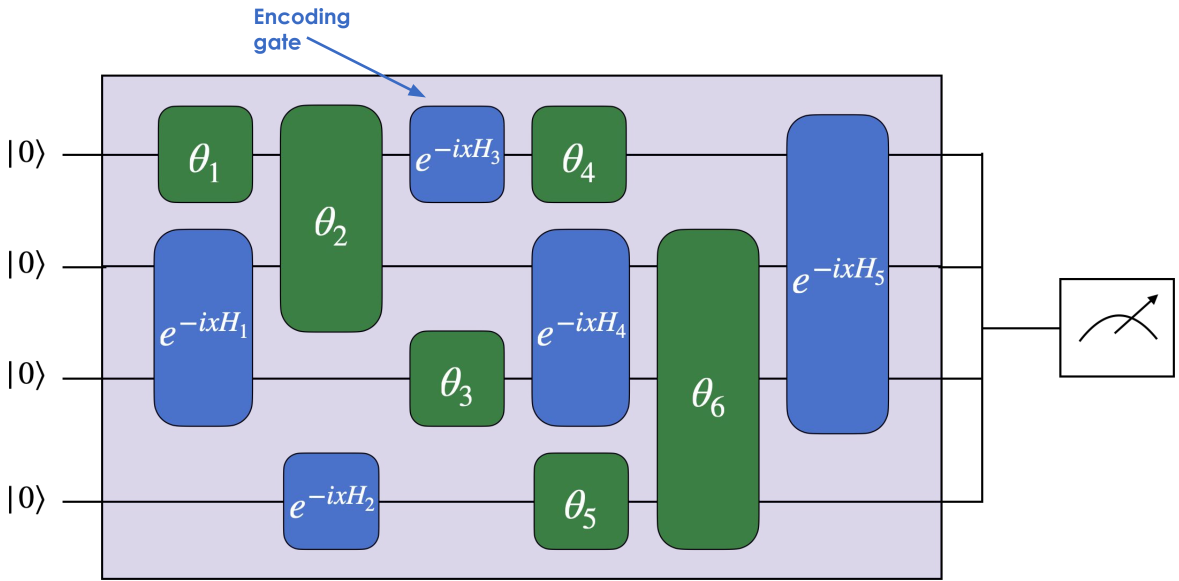
Near Term QML : A Recipe

- Take a *variational* quantum circuit that looks like this



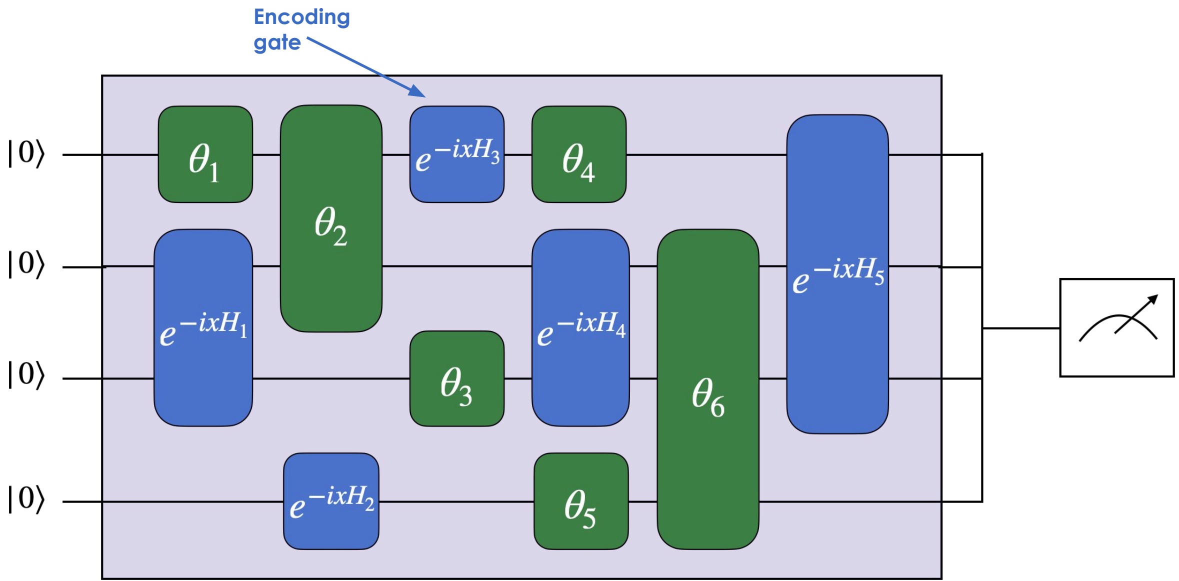
Near Term QML : A Recipe

- Take a *variational* quantum circuit that looks like this
- Input your data (x) as parameters of some gates



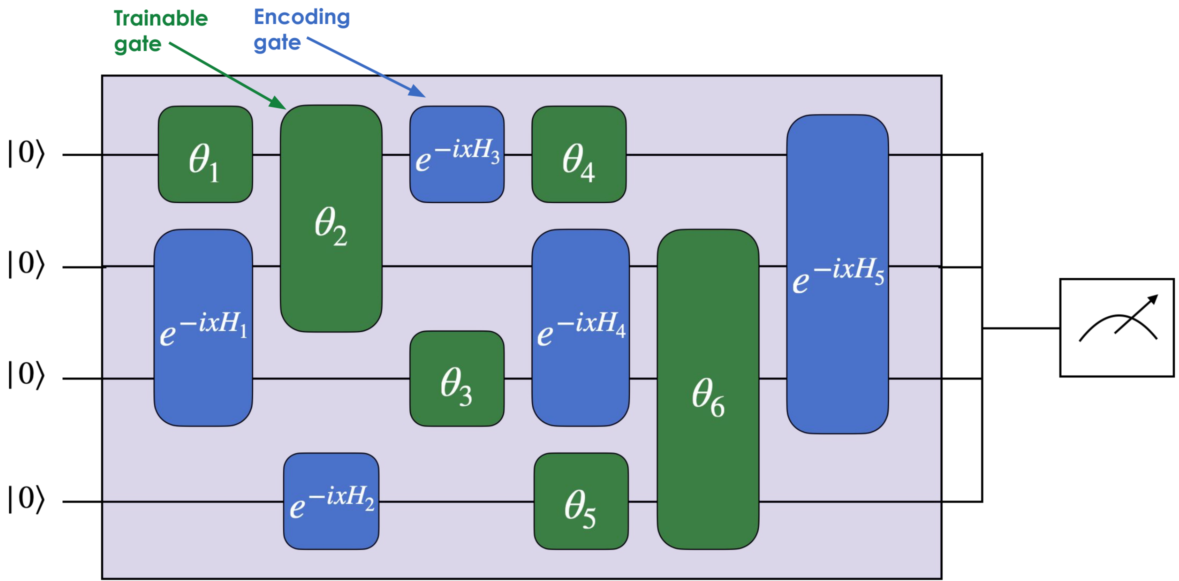
Near Term QML : A Recipe

- Take a *variational* quantum circuit that looks like this
- Input your data (x) as parameters of some gates
- Hope that what you measure at the end is the right answer

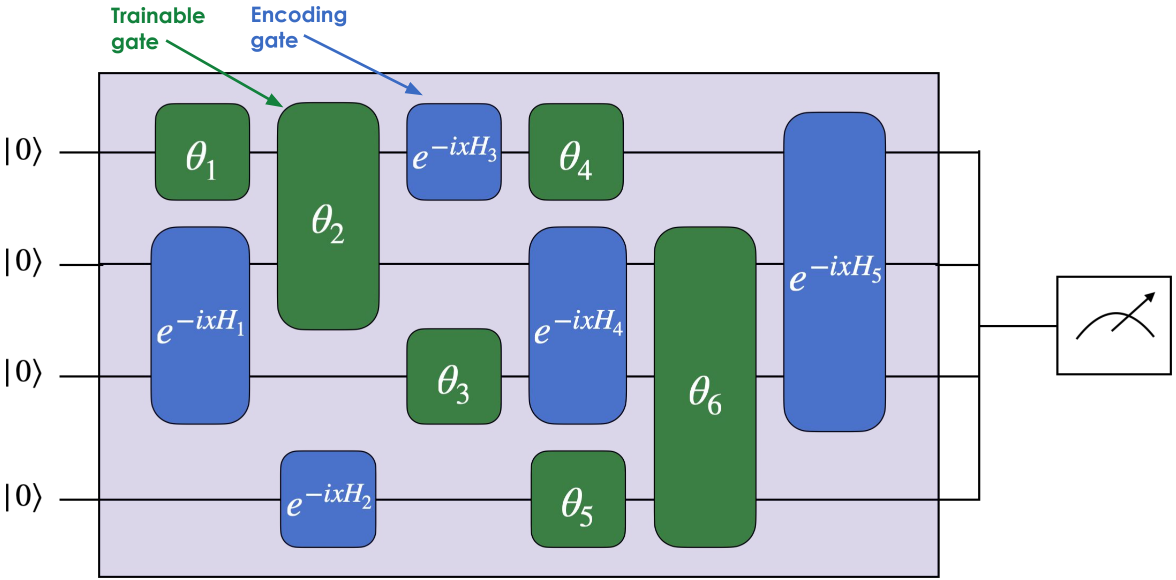


Near Term QML : A Recipe

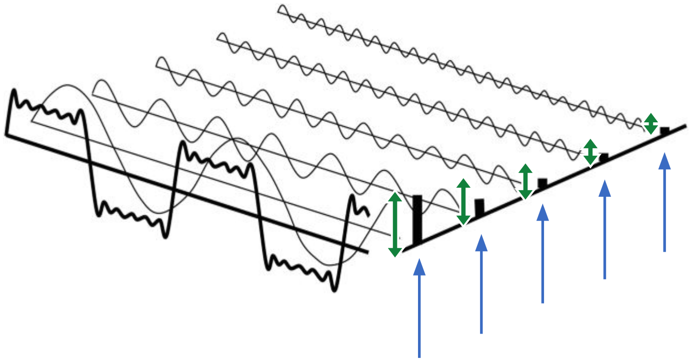
- Take a *variational* quantum circuit that looks like this
- Input your data (x) as parameters of some gates
- Hope that what you measure at the end is the right answer
- Tune the trainable gates (θ) until your hope becomes true



Near Term QML : What is going on?



$$f(x;\theta) = \sum_{\omega \in \Omega} c_\omega(\theta) e^{i\omega x}$$



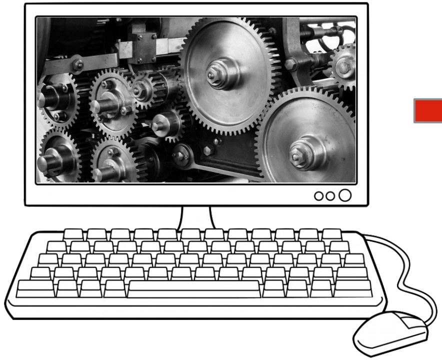
It learns (exponentially large) Fourier Series

M. Schuld: “**Supervised quantum machine learning models are kernel methods**”

Near Term QML : What is going on?

Learning in Exponentially Large Spaces

$$\begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix}$$

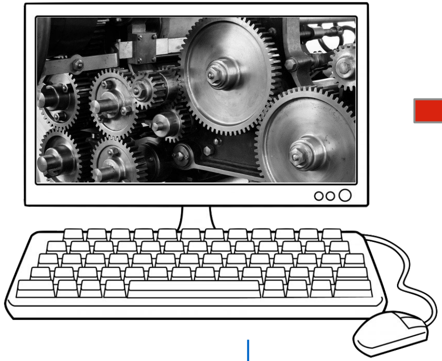


$$answer \begin{cases} +1\ (dog) \\ -1\ (cat) \end{cases}$$

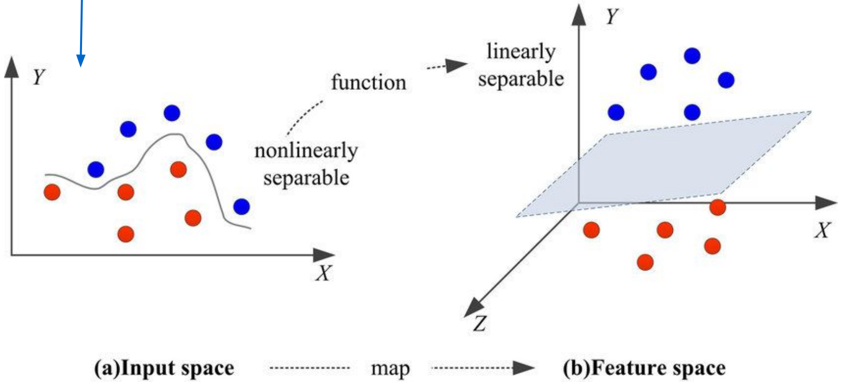
Near Term QML : What is going on?

Learning in Exponentially Large Spaces

$$\begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix}$$



$$answer \begin{cases} +1\ (dog) \\ -1\ (cat) \end{cases}$$

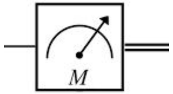


Near Term QML : What is going on?

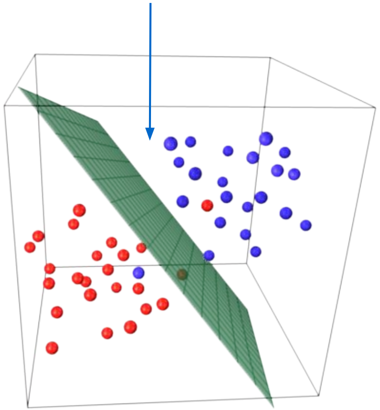
Learning in Exponentially Large Spaces

$$\begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix}$$

loader



$$answer \begin{cases} +1\ (dog) \\ -1\ (cat) \end{cases}$$



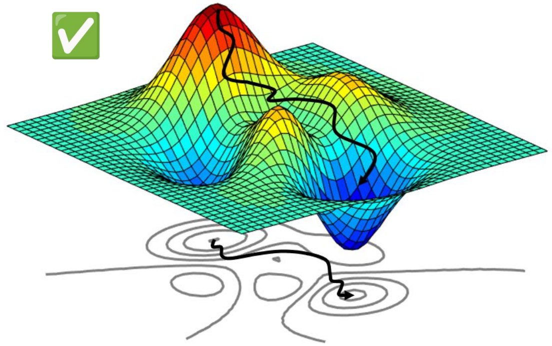
#Dimensions : Exponential

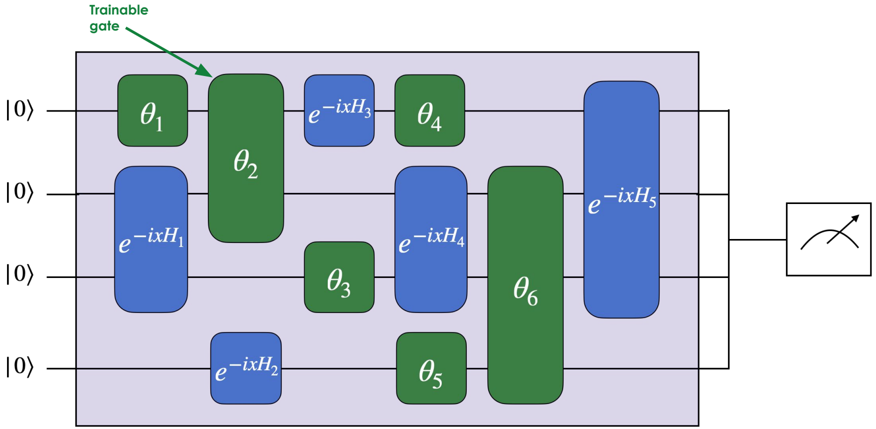
Near Term QML : Main Issues

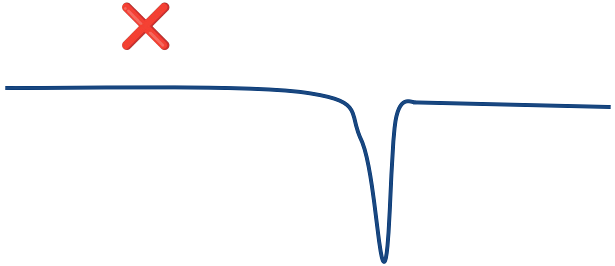
- Barren Plateaus: impossibility to train

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta)$$

Exponentially small value!



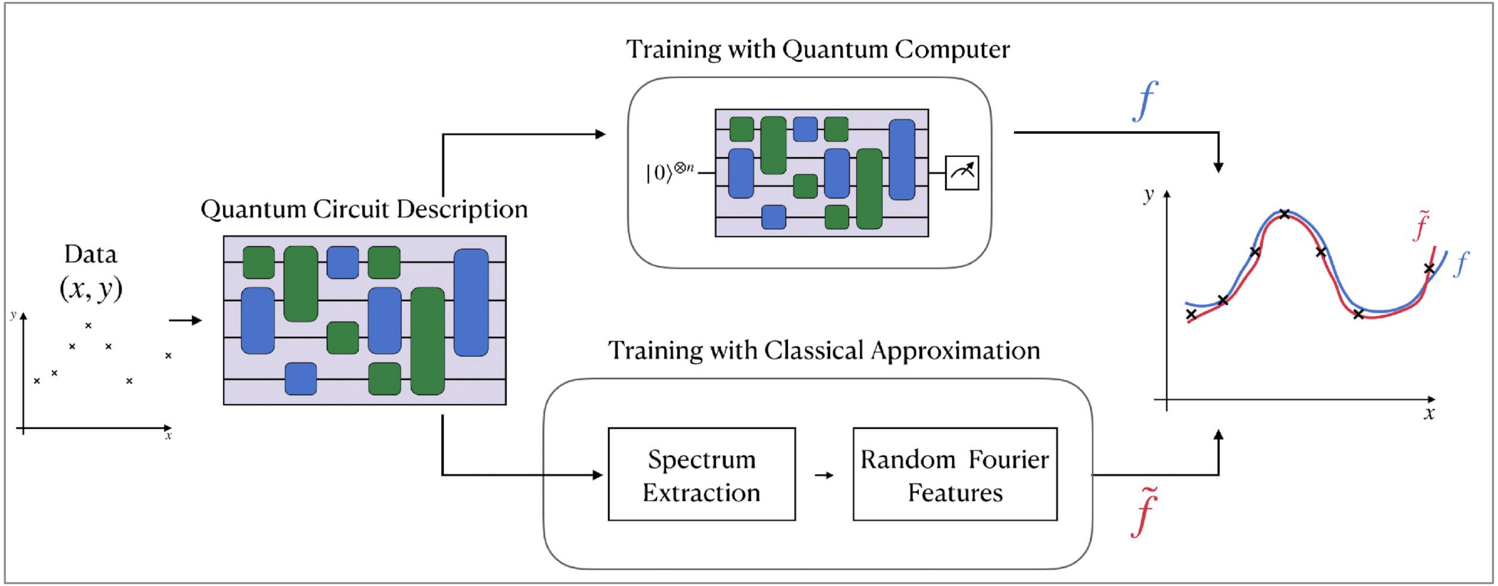




M. Larroca et al: "**A Review of Barren Plateaus in Variational Quantum Computing**"

Near Term QML : Main Issues

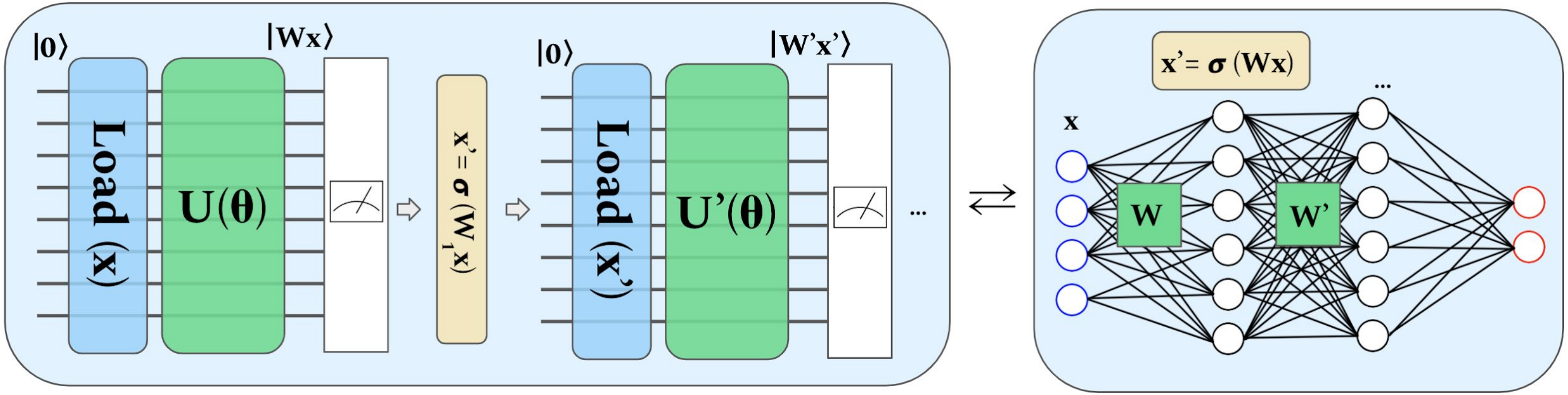
- Barren Plateaus: impossibility to train
- Classical Approximations



JL, S.Thabet, C.Dalyac, H.Mhiri, E.Kashefi: "**Classically Approximating Variational Quantum Machine Learning with Random Fourier Features**" ICLR 2022

Alternative: Subspace Preserving QML

- Variational, low depth q circuits
- Reproduce Neural Networks rigorously
- Does not explore exponentially large spaces



Trade off between:

- Loaders Feasibility
- Classical Approximations
- Barren Plateau

L.Monbroussou, E.Mamon, **JL**, A.Grilo, R.Kukla, E. Kashefi "**Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning**"

Qonqlusion

- QML is not so simple to implement
- Near / Long term are very different approaches
- Devil is in the details
- We might be surprise when Q Computers will be big enough
- Meanwhile: keep doing research !

Quantum Machine Learning

TQCI Seminar

Jonas Landman
Postdoctoral Research Associate - University of Edinburgh / LIP6