arxiv: **2306.15415**

Quantum Fourier Networks for Solving Parametric PDEs

Jonas Landman
Postdoctoral Research Associate
University of Edinburgh
QC Ware

Natansh Mathur
PhD Student
Université Paris-Cité
QC Ware

Joint Work with:

Iordanis Kerenidis
CNRS, QC Ware

Nishant Jain
IIT Roorkee, QC Ware

1/ Hamming Weight Preserving QML

Theory

a - Another approach to QML

b - Amplitude Encoding

c - Hamming Weight Preserving Circuits

d - Quantum Neural Networks Applications

2/ Quantum Fourier Neural Networks for solving PDEs

Application
(this paper)

a - The Classical Fourier Neural Operator

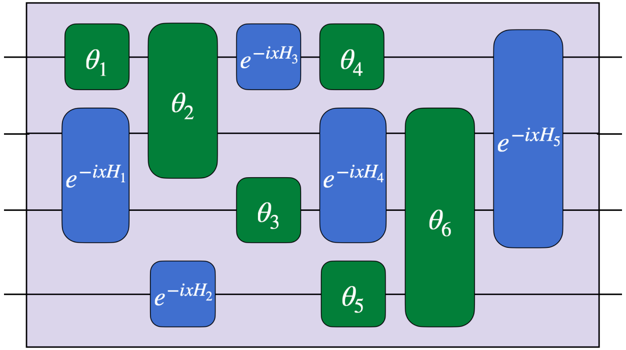
b - Quantum FourierNN

c - Experimental Results

1.a) Another Approach to QML

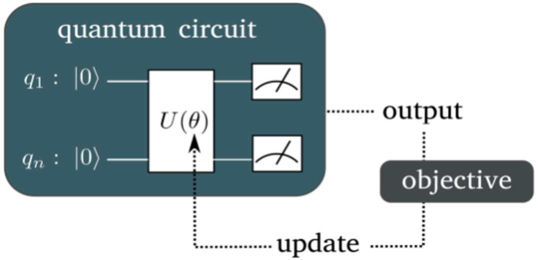
Most QML **Today**

Hamiltonian Encoding



$$e^{-ixH}$$

Variational Circuit



$$f_{\boldsymbol{\theta}}(x) = \langle 0| U^\dagger(x,\boldsymbol{\theta}) M U(x,\boldsymbol{\theta}) |0\rangle$$

Pros

- Easy to **implement**
- Potentially **universal**

Cons

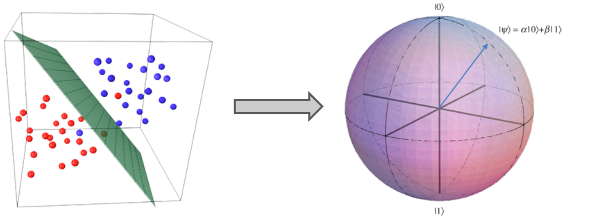
- **Hard to prove speedup / scaling**
- Hard to **compare** with Classical
- Classical Surrogates (Dequantisation)
- **Trainability** issues (BP)
- Hard to interpret **Expressivity**

« Variational quantum algorithms » M.Cerezo et al. Nature Reviews Physics Vol.3 (2021)
« Classically Approximating Variational Quantum Machine Learning with Random Fourier Features » J.Landman et al. ICLR 2022

1.a) Another Approach to QML

Most QML **Before**

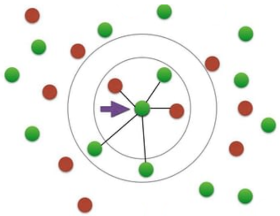
Amplitude encoding



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

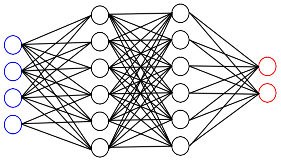
$$|x\rangle = \frac{1}{\|x\|}\sum_{i=1}^{n} x_i |i\rangle$$

Quantum Algorithms
Linear Algebra



Sing. Val. Transform
Phase Estimation
Q-SVM, Q-means
etc.

$$x = A^{-1}b$$



Pros

- **Provable Speedups**
- **Comparable** to classical
- Rigorous & interpretable
- **No Barren Plateaus** etc.

Cons

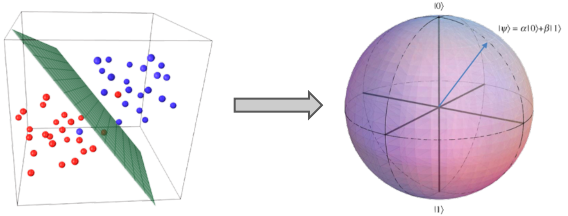
- **Long Term** (FTQC)
- **QRAM**
- Dequantisation

« Quantum Algorithms for Unsupervised Machine Learning and Neural Networks » J.Landman arXiv:2111.03598

1.a) Another Approach to QML

The best of **both worlds** ?

Amplitude encoding

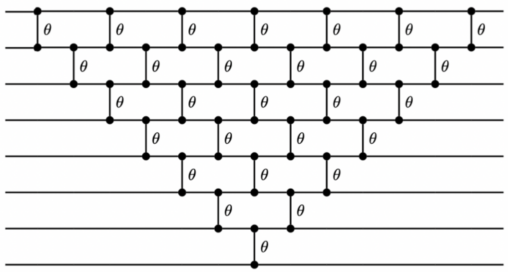


$x \in \mathbb{R}^n$

$$|x\rangle = \frac{1}{\|x\|}\sum_{i=1}^{n} x_i |i\rangle$$

In which basis ?

Variational Circuit



What is it good for ?

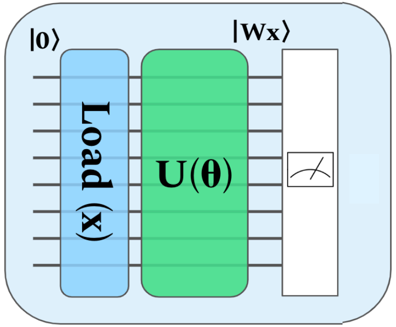
Pros

- Interpretable / Comparable to classical
- Provable Speedups
- Easy to implement
- No Barren Plateaus
- Modular

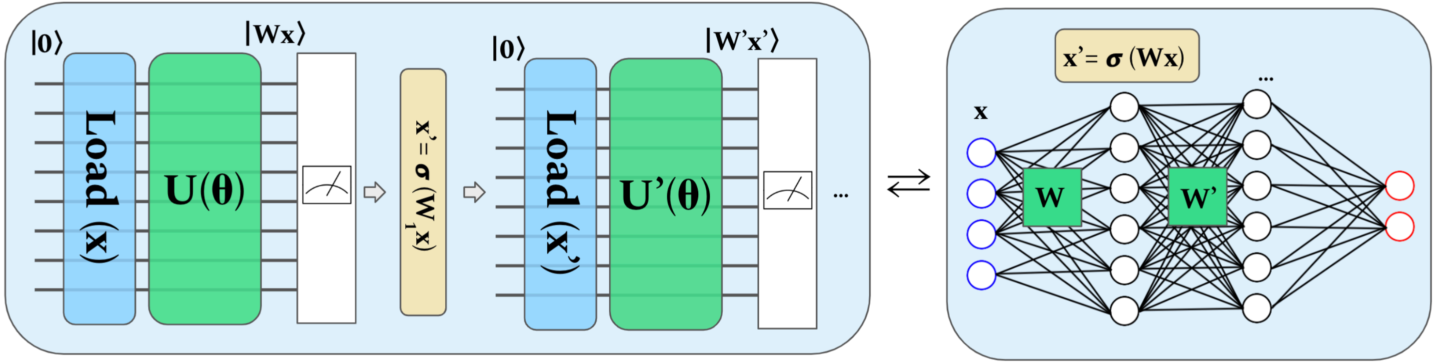
Cons

- Cost of non-linearities
- Depth depends on HW connectivity

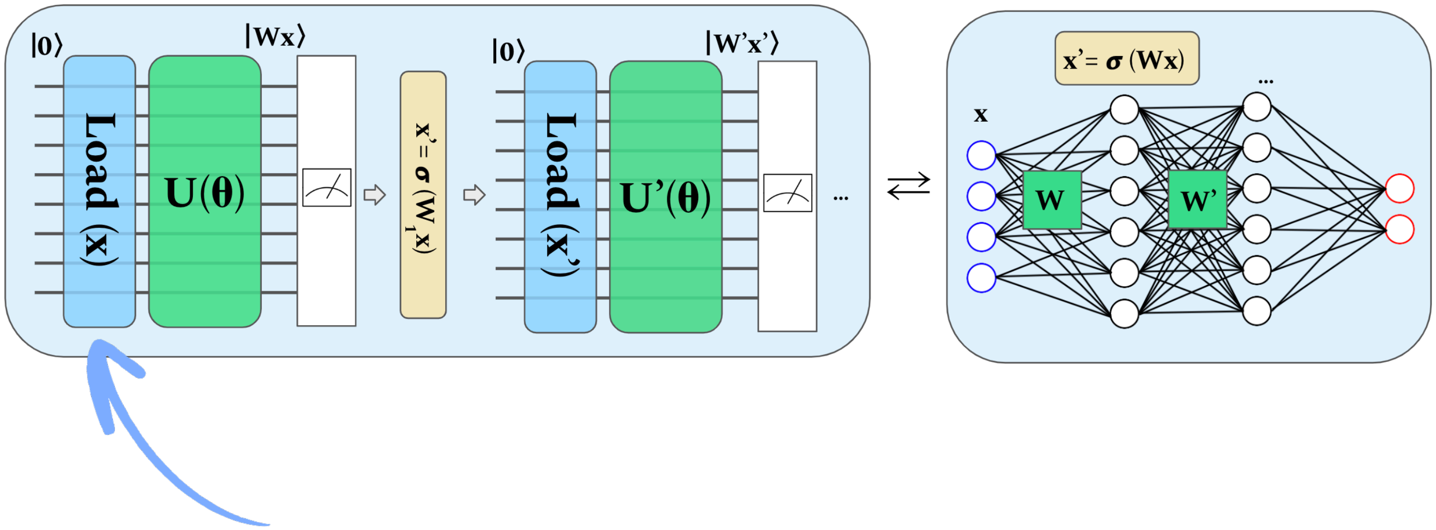
1.a) Another Approach to QML



1.a) Another Approach to QML



1.a) Another Approach to QML



1.b) Amplitude Encoding

Amplitude encoding

A vector $x \in \mathbb{R}^n$ is encoded using the amplitudes of a quantum superposition

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \longrightarrow |x\rangle = x_1|e_1\rangle + x_2|e_2\rangle + \cdots + x_n|e_n\rangle = \sum_i x_i|e_i\rangle \qquad \|x\|_2 = 1$$

1.b) Amplitude Encoding

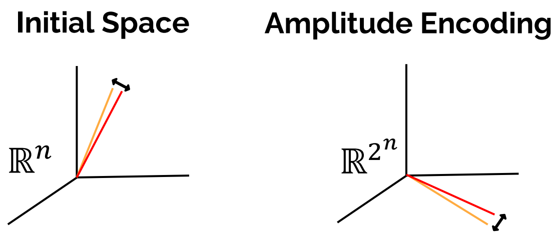
Amplitude encoding

A vector $x \in \mathbb{R}^n$ is encoded using the amplitudes of a quantum superposition

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \longrightarrow |x\rangle = x_1|e_1\rangle + x_2|e_2\rangle + \cdots + x_n|e_n\rangle = \sum_i x_i|e_i\rangle \qquad \|x\|_2 = 1$$

If $\{|e_1\rangle, \cdots, |e_n\rangle\}$ forms an ortho basis, this encoding preserves the metric

$$\| x - y \|_2 = \| \, |x\rangle - |y\rangle \, \|_2$$



1.b) Amplitude Encoding

Amplitude encoding $$|x\rangle = \frac{1}{\|x\|}\sum_i x_i|e_i\rangle$$

Basis choice with **n qubits ?**

1.b) Amplitude Encoding

Amplitude encoding $|x\rangle = \frac{1}{\|x\|}\sum_i x_i|e_i\rangle$

Basis choice with **n qubits**

	States	Maximum dimension for x	Method
- Computational basis	$\|00001\rangle, \|00010\rangle, \|00011\rangle, \cdots$	2^n in total	QRAM

1.b) Amplitude Encoding

Amplitude encoding $|x\rangle = \frac{1}{\|x\|}\sum_i x_i |e_i\rangle$

Basis choice with **n qubits**

	States	Maximum dimension for x	Method
- Computational basis	$\|00001\rangle, \|00010\rangle, \|00011\rangle, \cdots$	2^n in total	QRAM
- Unary basis	$\|00001\rangle, \|00010\rangle, \|00100\rangle, \cdots$	n in total	Unary Data Loaders

1.b) Amplitude Encoding

Amplitude encoding $|x\rangle = \frac{1}{\|x\|}\sum_i x_i |e_i\rangle$

Basis choice with **n qubits**

	States	Maximum dimension for x	Method
- Computational basis	$\|00001\rangle, \|00010\rangle, \|00011\rangle, \cdots$	2^n in total	QRAM
- Unary basis	$\|00001\rangle, \|00010\rangle, \|00100\rangle, \cdots$	n in total	Unary Data Loaders
- Hamming Weight k basis	$\|00111\rangle, \|01101\rangle, \|11001\rangle, \cdots$ ($k = 3$)	$\binom{n}{k}$ in total	HW-k Data Loaders

1.b) Amplitude Encoding

Use two-qubit gates that **preserve the Hamming Weight**

RBS gates

Reconfigurable Beam Splitter

$$RBS(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$RBS(\theta) : \begin{cases} |01\rangle \mapsto \cos\theta\,|01\rangle - \sin\theta\,|10\rangle \\ |10\rangle \mapsto \sin\theta\,|01\rangle + \cos\theta\,|10\rangle \end{cases}$$

1.b) Amplitude Encoding

Use two-qubit gates that **preserve the Hamming Weight**

RBS gates 

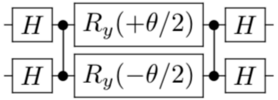
Reconfigurable Beam Splitter

$$RBS(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$RBS(\theta) : \begin{cases} |01\rangle \mapsto \cos\theta\,|01\rangle - \sin\theta\,|10\rangle \\ |10\rangle \mapsto \sin\theta\,|01\rangle + \cos\theta\,|10\rangle \end{cases}$$

Easy to implement

Decomposition:



Native:

"Demonstrating a continuous set of two-qubit gates for near term quantum algorithms"
B. Foxen et al. Physical Review Letters, Vol.125, n°12

1.b) Amplitude Encoding

Use two-qubit gates that **preserve the Hamming Weight**

RBS gates

Reconfigurable Beam Splitter

$$RBS(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$RBS(\theta) : \begin{cases} |01\rangle \mapsto \cos\theta\,|01\rangle - \sin\theta\,|10\rangle \\ |10\rangle \mapsto \sin\theta\,|01\rangle + \cos\theta\,|10\rangle \end{cases}$$

FBS gates

Fermionic Beam Splitter

$$FBS_{ij}(\theta)\,|S\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & (-1)^{f(i,j,S)}\sin(\theta) & 0 \\ 0 & (-1)^{f(i,j,S)+1}\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

« Quantum machine learning with subspace states » I.Kerenidis et al. arxiv:2202.00054

1.b) Amplitude Encoding

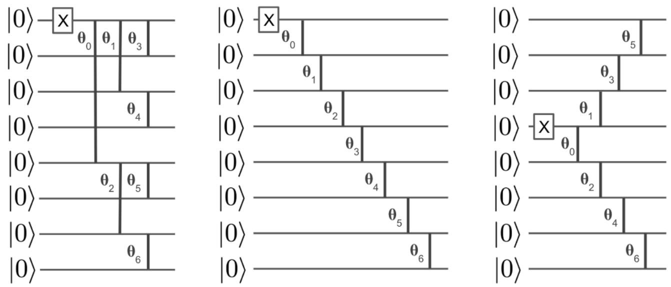
Unary

Data Loaders

Vector

$x \in \mathbb{R}^n$, n qubits

$$|x\rangle = \frac{1}{\|x\|}\sum_i^n x_i |e_i\rangle$$



parallel — $O(\log(n))$

diag — $O(n)$

semi-diag — $O(n/2)$

diag

$$\begin{cases} \theta_0 = \arccos(x_0) \\ \theta_1 = \arccos\left(\frac{x_1}{\sin(\theta_0)}\right) \\ \theta_2 = \arccos\left(\frac{x_2}{\sin(\theta_0)\sin(\theta_1)}\right) \\ \dots \end{cases}$$

« Nearest Centroid Classification on a Trapped Ion Quantum Computer » S.Johri et al. NPJ Quantum Information Vol.7,122 (2021)

1.b) Amplitude Encoding

Unary

Data Loaders

Vector

$x \in \mathbb{R}^n$, n qubits

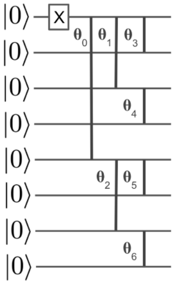
$$|x\rangle = \frac{1}{\|x\|}\sum_{i}^{n} x_i |e_i\rangle$$

Matrix

$$\begin{pmatrix} X_{11} & \cdots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mn} \end{pmatrix}$$

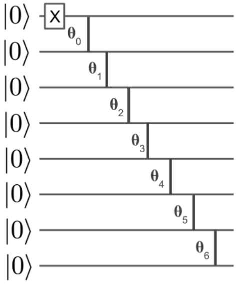
$X \in \mathbb{R}^{m\times n}$, $m+n$ qubits

$$|X\rangle = \frac{1}{\|X\|}\sum_{i}^{m}\sum_{j}^{n} X_{ij} |e_i\rangle |e_j\rangle$$



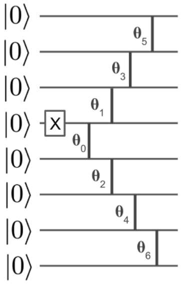
parallel

$O(\log(n))$



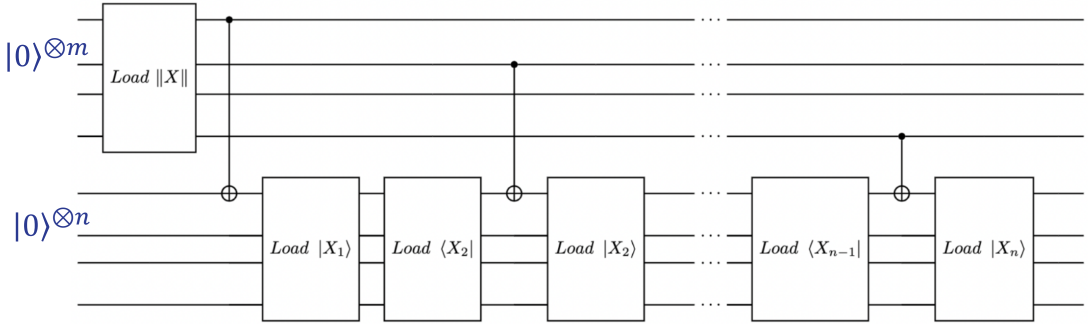
diag

$O(n)$



semi-diag

$O(n/2)$



$O(\log(m) + 2m\log(n))$

« Nearest Centroid Classification on a Trapped Ion Quantum Computer » S.Johri et al. NPJ Quantum Information Vol.7,122 (2021)

« Quantum Vision Transformers » E.A.Cherrat et al. arxiv:2209.08167

1.b) Amplitude Encoding

Hamming Weight k

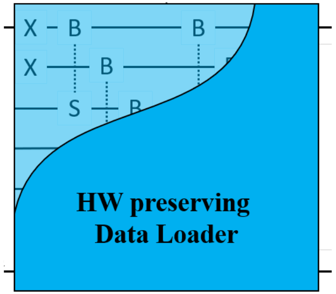
Data Loaders

Heuristic

$x \in \mathbb{R}^{\binom{n}{k}}$, n qubits

$$|x\rangle = \frac{1}{\|x\|}\sum_{i}^{\binom{n}{k}} x_i |e_i\rangle$$

$|0\rangle^{\otimes d}$



$|x\rangle$

Trainable
With guarantees

« Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning » L.Monbroussou et al. arxiv:2309.15547

Clifford Loader

$X \in \mathbb{R}^{n \times k}$, n qubits

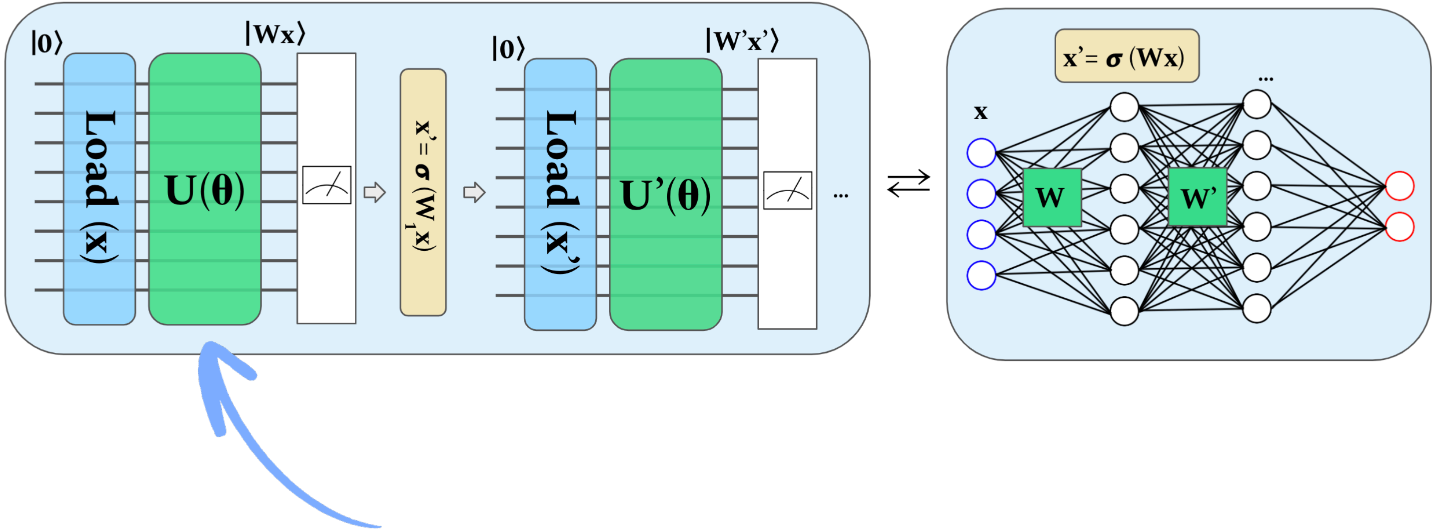
$$|X\rangle = \sum_{|S|=k} \det(X_S)|e_S\rangle$$

```
q0 : -B-Z-B-Z---Z-B-Z-X-Z-B--Z-----Z--B--Z--B--
      | | | |   | | |   | |  |     |  |  |  |
q1 : -S-@-|-@-X-@-|-@---@-|--@--X--@--|--@--S--
          |   |   |       |     |     |
q2 : -B---S-X-@---|-------|-----@--X--S-----B--
      |     |     |       |        |        |
q3 : -S-----@-----|-------|--------@--------S--
                  |       |
q4 : -B-Z-B-Z-----S-------S--------Z--B--Z--B--
      | | | |                      |  |  |  |
q5 : -S-@-|-@----------------------@--|--@--S--
          |                           |
q6 : -B---S---------------------------S-----B--
      |                                     |
q7 : -S-------------------------------------S--
```

$O(k\log(n))$

« Quantum machine learning with subspace states » I.Kerenidis et al. arxiv:2202.00054

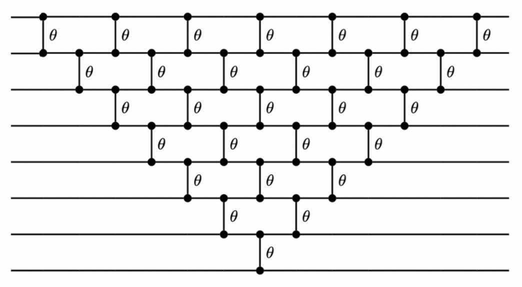
1.c) Hamming Weight Preserving Circuits



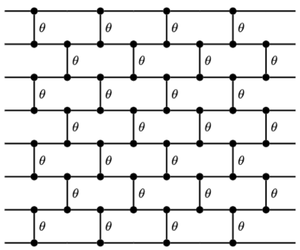
1.c) Hamming Weight Preserving Circuits

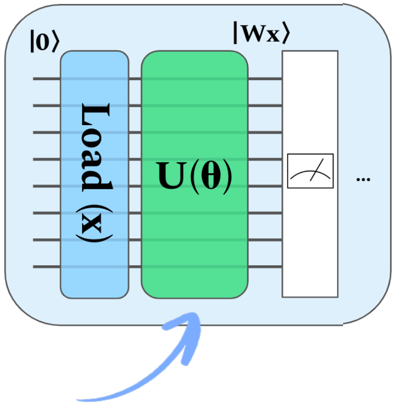
Quantum Orthogonal Layers

Pyramid

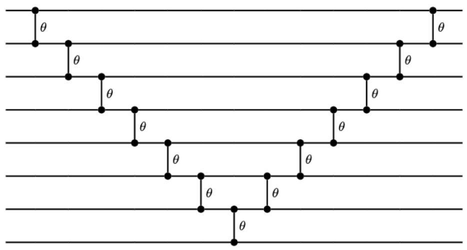


Block



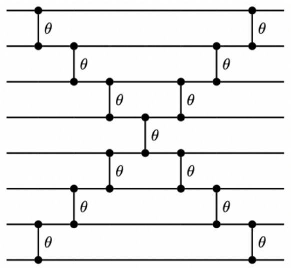


V Circuit



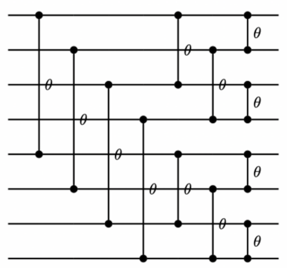
$\sim 2n$

X Circuit



$\sim n$

Butterfly

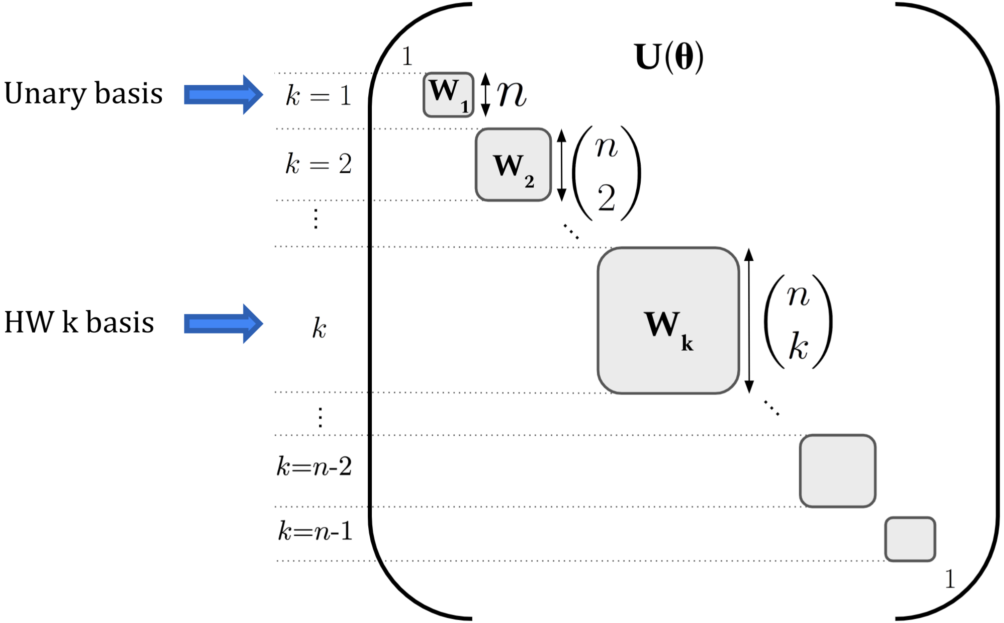


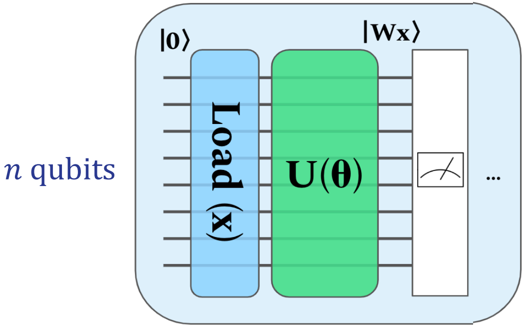
$\sim \log(n)$

« Quantum Vision Transformers » E.A.Cherrat et al. arxiv:2209.08167

1.c) Hamming Weight Preserving Circuits

Properties of **Hamming Weight (HW) Preserving** Quantum Circuits

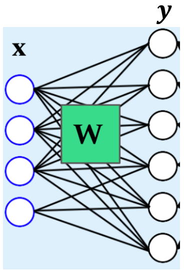




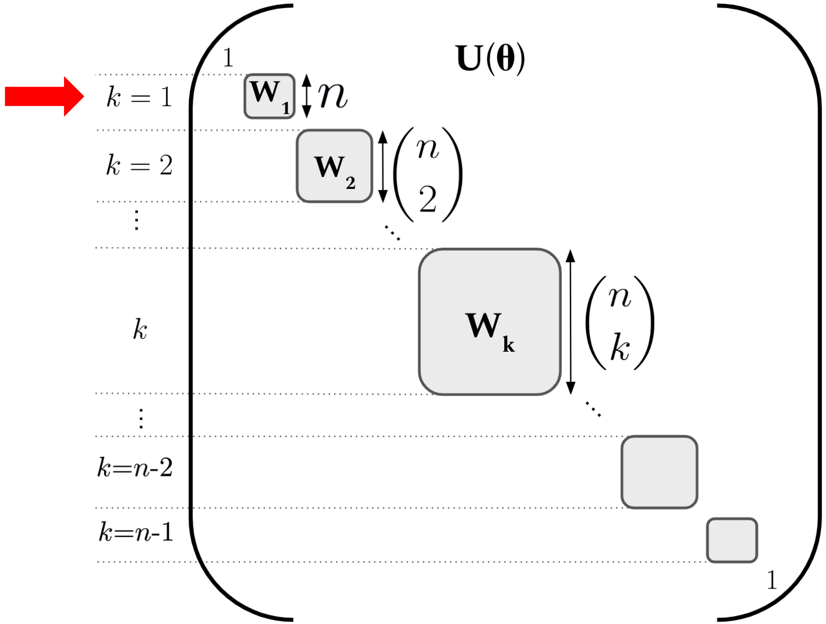
If we load $|x\rangle$ in HW k basis

The output state $|y\rangle$ is in the same basis

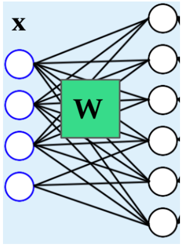
$$|y\rangle = |W_k x\rangle$$



Properties of **Hamming Weight (HW) Preserving** Quantum Circuits



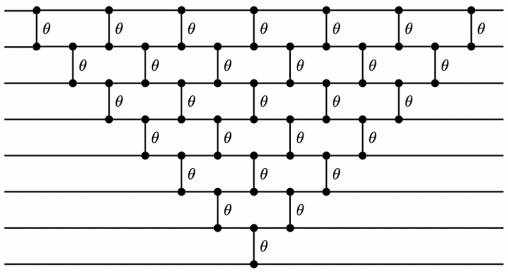
Orthogonal Neural Nets



$$\mathbf{W_1} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$

SO(n) : Orthogonal matrices (with Det = +1)

$\frac{n(n-1)}{2}$ Free parameters

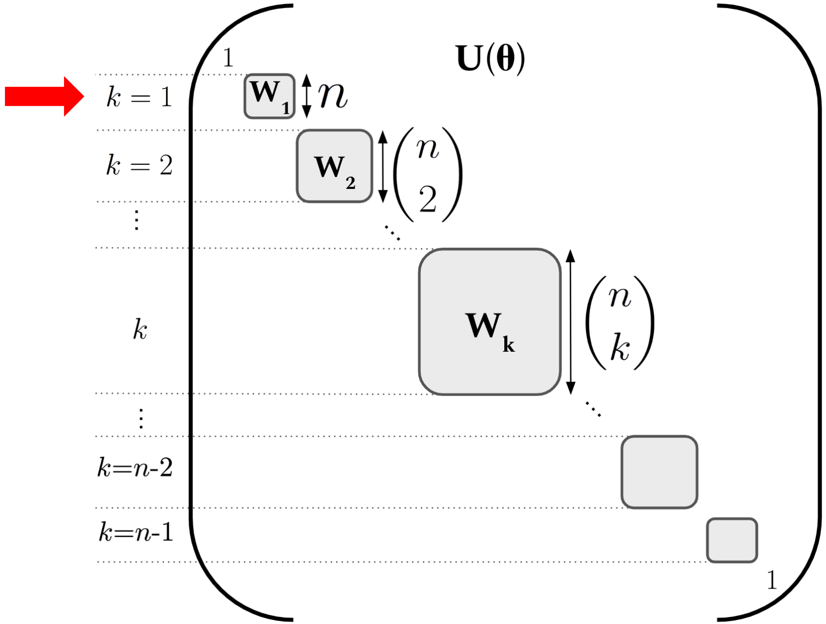


Quantum: $O(n)$

Classical: $O(n^2)$

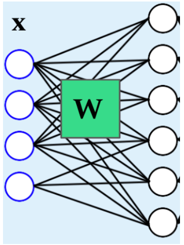
« Orthogonal Deep Neural Networks », K.Jia IEEE transactions on pattern analysis and machine intelligence, 43(4)

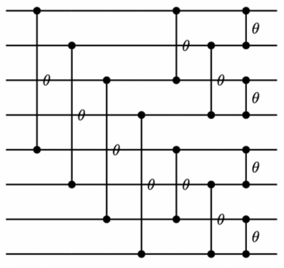
Properties of **Hamming Weight (HW) Preserving** Quantum Circuits



$$\mathbf{W_1} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$

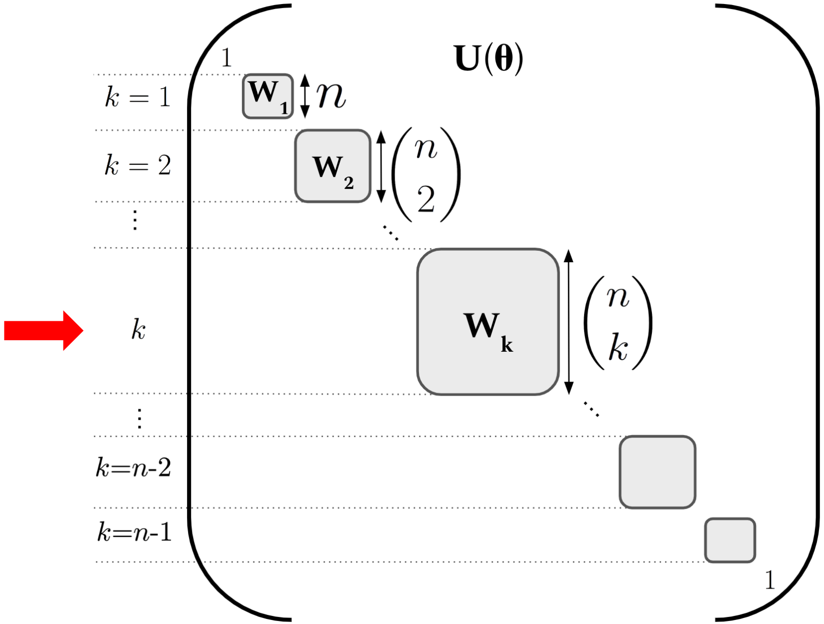
Butterfly Neural Nets





Quantum: $O(\log(n))$

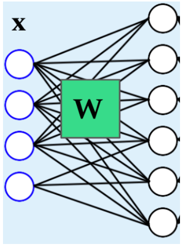
Properties of **Hamming Weight (HW) Preserving** Quantum Circuits

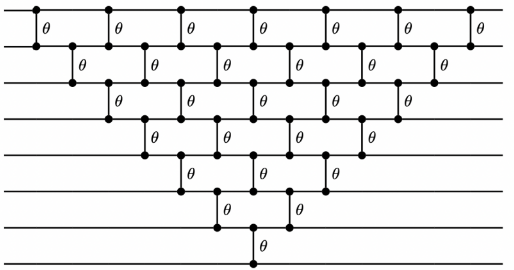


$\boldsymbol{W_k} = ?$

How many degrees of freedom?

DLA dimension?

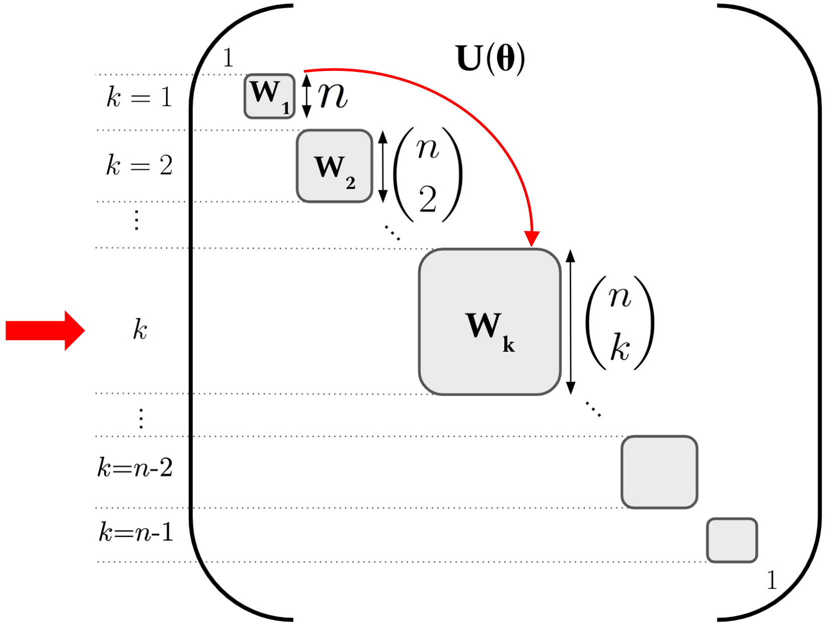




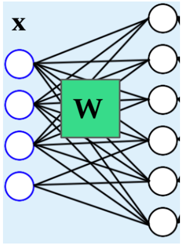
Quantum: $O(n)$

Classical: $O(n^{2k})$

Properties of **Hamming Weight (HW) Preserving** Quantum Circuits



Compound Neural Nets ?

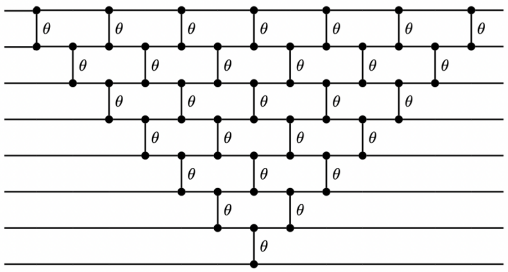


Using nearest neighbour connectivity
(or FBS gates)

$\boldsymbol{W_k}$ = ***k*-Compound** matrix of $\boldsymbol{W_1}$

$$W_{k\,IJ} = \det(W[IJ])$$

where **I** and **J** are subsets of rows/col with size **k**

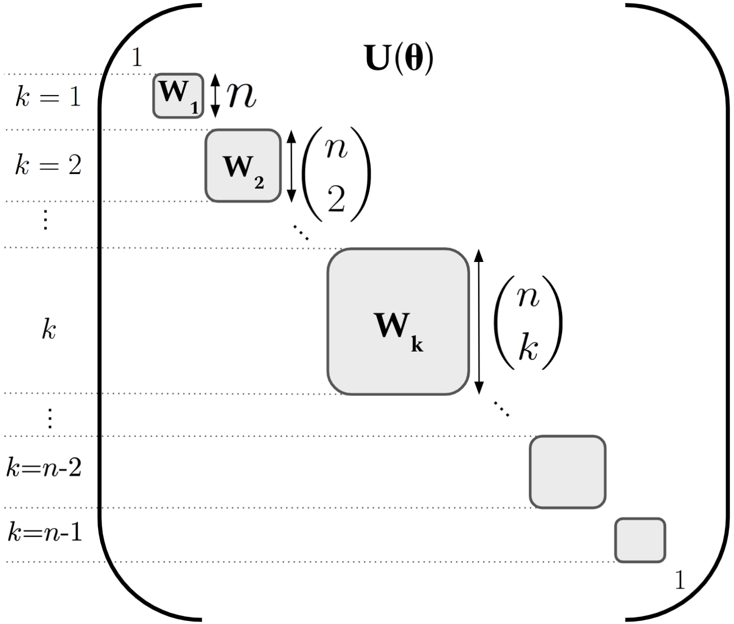


Quantum: $O(n)$

Classical: $O(n^{2k})$

« Quantum machine learning with subspace states » I.Kerenidis et al. arxiv:2202.00054

1.c) Hamming Weight Preserving Circuits



Absence of Barren Plateaus

$\boldsymbol{d_k}$: dimension of subspace k, $d_k = \binom{n}{k}$
$\mathbf{dim}(\boldsymbol{g_k})$: dimension of Lie Algebra of subspace k

$$d_k \neq \dim(g_k)$$

For **LASA** circuits (Lie Algebraic Supported Ansatz) :

$$\mathbb{V}(\partial_\theta C) = O\left(\frac{1}{poly(\dim(g_k))}\right)$$

« The Adjoint Is All You Need: Characterizing Barren Plateaus in Quantum Ansätze » E.Fontana et al. arxiv:2309.07902

For **RBS/FBS** circuits (≠LASA) :

$$\mathbb{V}(\partial_\theta C) = O\left(\frac{1}{poly(d_k)}\right) = O\left(\frac{1}{n^k}\right) \neq O\left(\frac{1}{2^n}\right)$$

« Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning » L.Monbroussou et al. arxiv:2309.15547

1.c) Hamming Weight Preserving Circuits

Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning

Léo Monbroussou,[1,2] Jonas Landman,[3,4] Alex B. Grilo,[1] Romain Kukla,[2] and Elham Kashefi[1,3]

[1] *Laboratoire d'Informatique de Paris 6, CNRS, Sorbonne Université, 4 Place Jussieu, 75005 Paris, France*
[2] *CEMIS, Direction Technique, Naval Group, 83190 Ollioules, France*
[3] *School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, United Kingdom*
[4] *QC Ware, Palo Alto, USA and Paris, France*
(Dated: September 28, 2023)

Quantum machine learning has become a promising area for real world applications of quantum computers, but near-term methods and their scalability are still important research topics. In this context, we analyze the trainability and controllability of specific Hamming weight preserving quantum circuits. These circuits use gates that preserve subspaces of the Hilbert space, spanned by basis states with fixed Hamming weight k. They are good candidates for mimicking neural networks, by both loading classical data and performing trainable layers. In this work, we first design and prove the feasibility of new heuristic data loaders, performing quantum amplitude encoding of $\binom{n}{k}$-dimensional vectors by training a n-qubit quantum circuit. Then, we analyze more generally the trainability of Hamming weight preserving circuits, and show that the variance of their gradients is bounded according to the size of the preserved subspace. This proves the conditions of existence of Barren Plateaus for these circuits, and highlights a setting where a recent conjecture on the link between controllability and trainability of variational quantum circuits does not apply.

27 Sep 2023

Showcasing a Barren Plateau Theory Beyond the Dynamical Lie Algebra

N. L. Diaz,[1,2] Diego García-Martín,[1] Sujay Kazi,[3] Martin Larocca,[4,5] and M. Cerezo[1]

[1] *Information Sciences, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[2] *Departamento de Física-IFLP/CONICET, Universidad Nacional de La Plata, C.C. 67, La Plata 1900, Argentina*
[3] *Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA*
[4] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[5] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

Barren plateaus have emerged as a pivotal challenge for variational quantum computing. Our understanding of this phenomenon underwent a transformative shift with the recent introduction of a Lie algebraic theory capable of explaining most sources of barren plateaus. However, this theory requires either initial states or observables that lie in the circuit's Lie algebra. Focusing on parametrized matchgate circuits, in this work we are able to go beyond this assumption and provide an exact formula for the loss function variance that is valid for arbitrary input states and measurements. Our results reveal that new phenomena emerge when the Lie algebra constraint is relaxed. For instance, we find that the variance does not necessarily vanish inversely with the Lie algebra's dimension. Instead, this measure of expressiveness is replaced by a generalized expressiveness quantity: The dimension of the Lie group modules. By characterizing the operators in these modules as products of Majorana operators, we can introduce a precise notion of generalized globality and show that measuring generalized-global operators leads to barren plateaus. Our work also provides operational meaning to the generalized entanglement as we connect it with known fermionic entanglement measures, and show that it satisfies a monogamy relation. Finally, while parameterized matchgate circuits are not efficiently simulable in general, our results suggest that the structure allowing for trainability may also lead to classical simulability.

ant-ph] 17 Oct 2023

The Adjoint Is All You Need: Characterizing Barren Plateaus in Quantum Ansätze

Enrico Fontana,[1,2] Dylan Herman,[1,*] Shouvanik Chakrabarti,[1] Niraj Kumar,[1] Romina Yalovetzky,[1] Jamie Heredge,[1,3] Shree Hari Sureshbabu,[1] and Marco Pistoia[1]

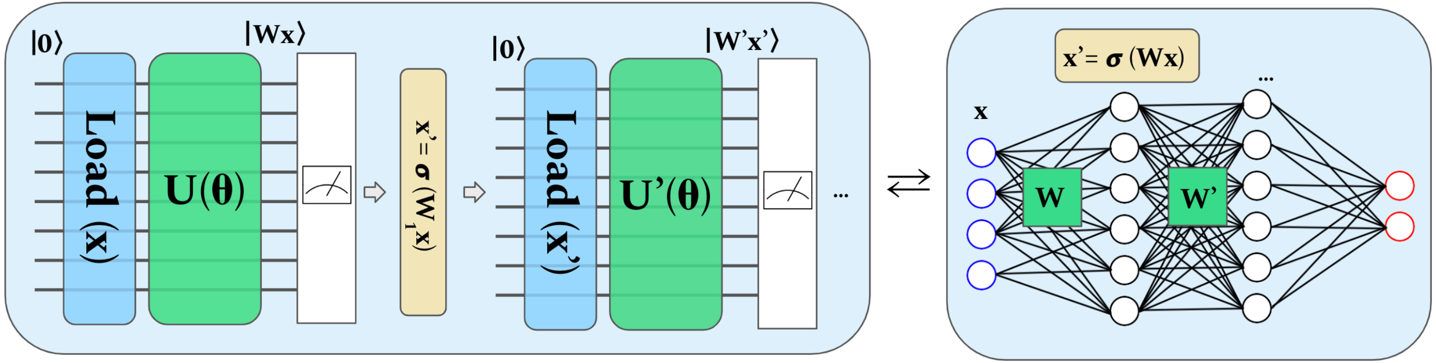
[1] *Global Technology Applied Research, JPMorgan Chase*
[2] *Computer and Information Sciences, University of Strathclyde*
[3] *School of Physics, The University of Melbourne*

Using tools from the representation theory of compact Lie groups, we formulate a theory of Barren Plateaus for parameterized quantum circuits whose observables lie in their dynamical Lie algebra (DLA), a setting that we term Lie algebra Supported Ansatz (LASA). A large variety of commonly used ansätze such as the Hamiltonian Variational Ansatz, Quantum Alternating Operator Ansatz, and many equivariant quantum neural networks are LASAs. In particular, our theory provides for the first time the ability to compute the variance of the gradient of the cost function for a non-trivial, subspace uncontrollable family of quantum circuits, the quantum compound ansätze. We rigorously prove that the variance of the gradient of the cost function, under Haar initialization, scales inversely with the dimension of the DLA, which agrees with existing numerical observations. Lastly, we include potential extensions for handling cases when the observable lies outside of the DLA and the implications of our results.

21 Sep 2023

Quantum Fully Connected Neural Networks

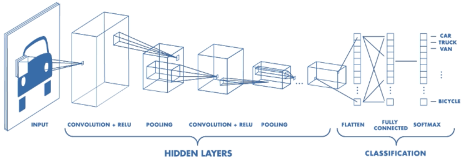
One layer = 1 loader + 1 ortho circuit + 1 measurement + 1 classical non linearity



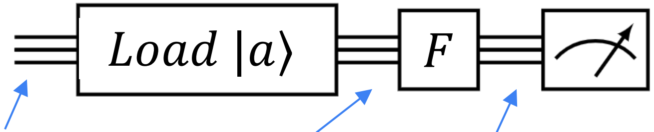
« Quantum Methods for Neural Networks and Application to Medical Image Classification » J.Landman et al. Quantum, 2022-12-22 Vol.6

1.d) - Quantum Neural Networks Applications

Quantum Convolutional Neural Networks



A convolution product can be rewritten as a (larger) matrix multiplication

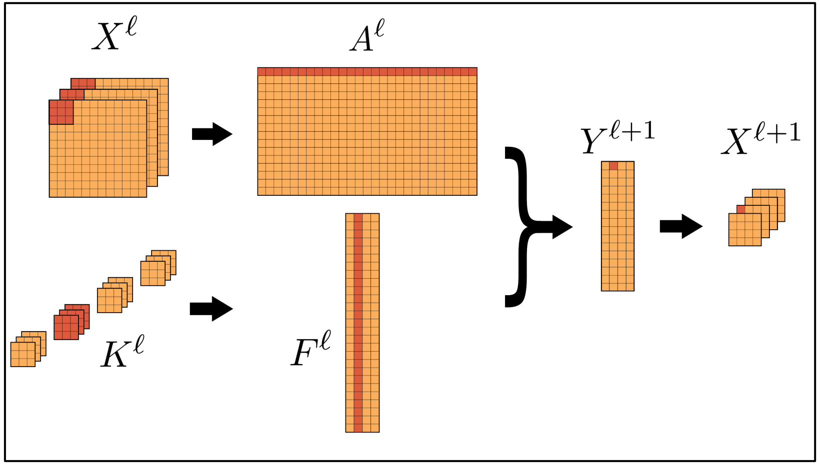


$|0\rangle$

$\mapsto |a\rangle = \sum_i a_i |e_i\rangle$

$$\mapsto \sum_i F a_i |e_i\rangle = \sum_i y_i |e_i\rangle = |y\rangle$$

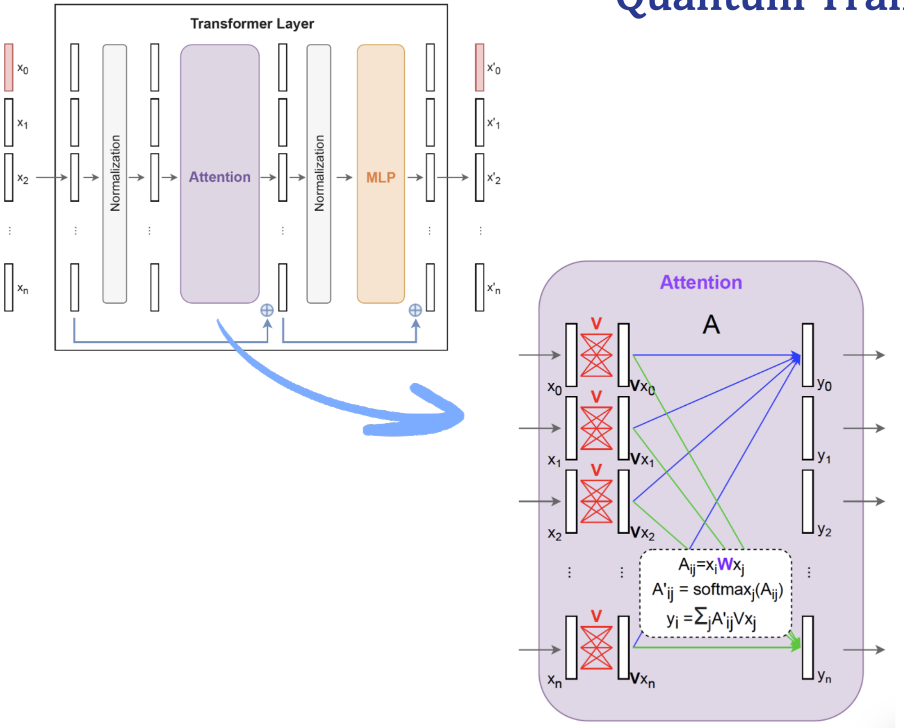
$y = Fa$

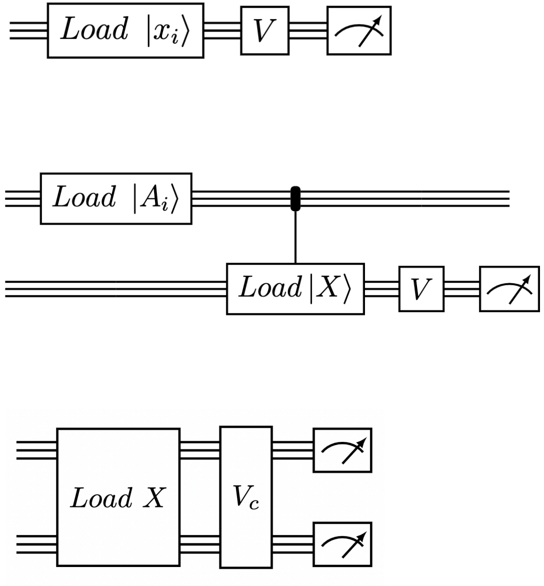


$$Y^{l+1} = F^l \cdot A^l$$

1.d) - Quantum Neural Networks Applications

Quantum Transformers





« Quantum Vision Transformers » E.A.Cherrat et al. arxiv:2209.08167

Papers !

« **Nearest Centroid Classification on a Trapped Ion Quantum Computer** »
S.Johri et al. NPJ Quantum Information Vol.7,122 (2021)
- Data Loader (unary)

« **Quantum Vision Transformers** » E.A.Cherrat et al. arxiv:2209.08167
- Data Loader (Matrix), Quantum Tranformers

POSTER

« **Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning** » L.Monbroussou et al. arxiv:2309.15547
- No Barren Plateaus in HW Preserving QML
- Data Loader (Heuristic HW-k)

QTML 22

« **Quantum Methods for Neural Networks and Application to Medical Image Classification** » J.Landman et al. Quantum, 2022-12-22 Vol.6
- Quantum Fully connected NN

« **Quantum machine learning with subspace states** »
I.Kerenidis et al. arxiv:2202.00054
- Data Loader (Clifford)
- Quantum Subspace Learning, Compound Matrix

« **The Adjoint Is All You Need: Characterizing Barren Plateaus in Quantum Ansätze** » E.Fontana et al. arxiv:2309.07902

« **Showcasing a Barren Plateau Theory Beyond the Dynamical Lie Algebra** » NL.Diaz et al. arxiv:2310.11505
- No Barren Plateaus « Beyond LASA »

QTML 23

« **Quantum Fourier Networks for Solving Parametric PDEs** » N.Jain et al. arxiv 2306.15415
- Unary Q Fourier Transform
- PDE solver with Quantum NN

NOW !

From Theory to Application

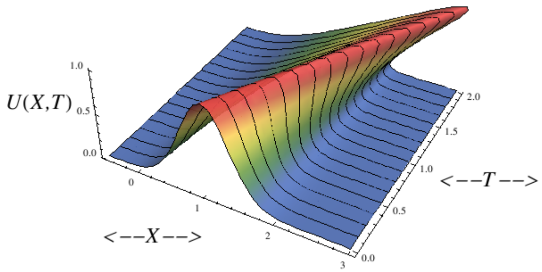
Quantum Fourier Neural Operator for Partial Differential Equations

Introduction to Partial Differential Equations (PDEs)

Burgers' Equation:

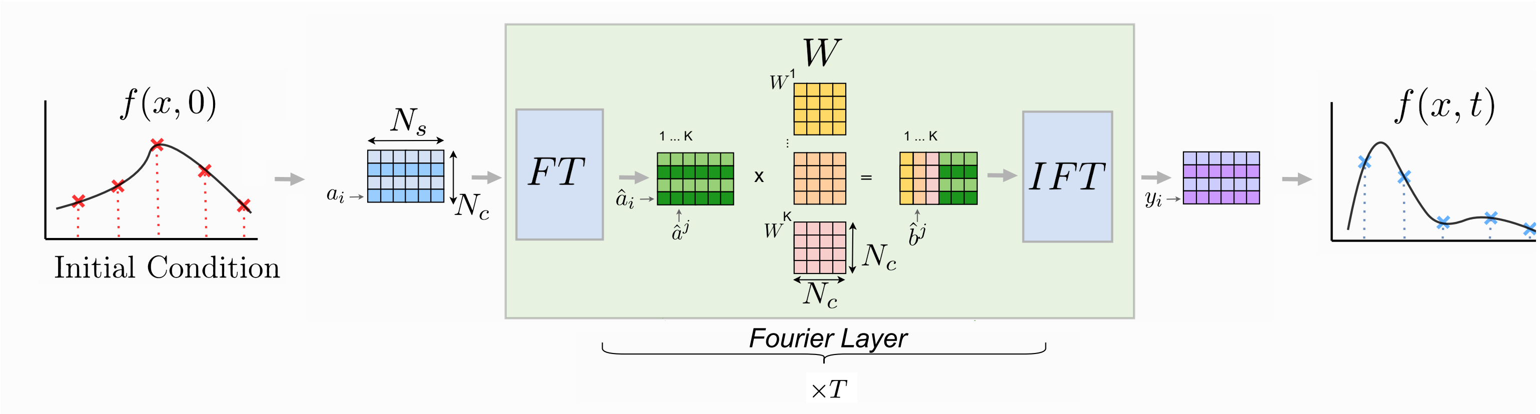
$$\partial_t u(x,t) + \partial_x \left(\frac{u^2(x,t)}{2} \right) = \nu \partial_x^2 u(x,t) \qquad \text{for } x \in (0,1), t \in (0,1]$$

$$u(x,0) = u_0(x), \qquad \text{for } x \in (0,1)$$



Goal: **Learn $\boldsymbol{u(x,0) \mapsto u(x,t)}$**

Classical Fourier Neural Operator (FNO)



FOURIER NEURAL OPERATOR FOR PARAMETRIC PARTIAL DIFFERENTIAL EQUATIONS

Zongyi Li zongyili@caltech.edu
Nikola Kovachki nkovachki@caltech.edu
Kamyar Azizzadenesheli kamyar@purdue.edu
Burigede Liu bgl@caltech.edu
Kaushik Bhattacharya bhatta@caltech.edu
Andrew Stuart astuart@caltech.edu
Anima Anandkumar anima@caltech.edu

ICLR 2021

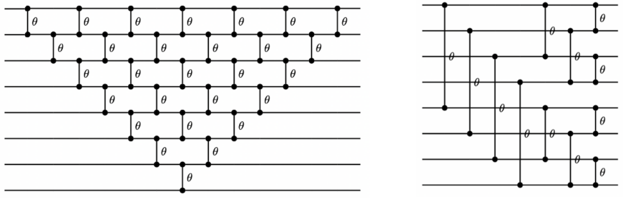
Quantum Toolbox

Already Discussed!

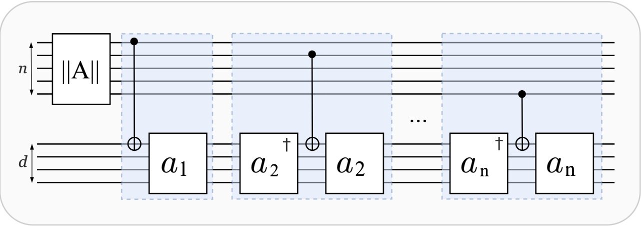
HW-Preserving gates

$$RBS(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

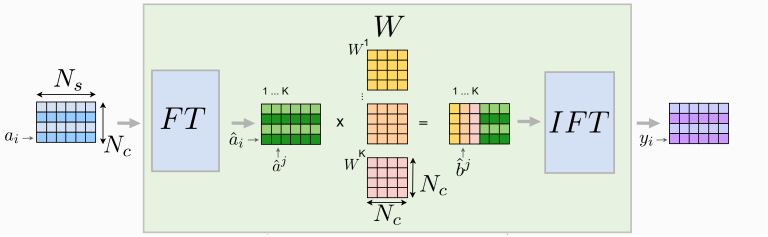
Orthogonal Quantum Layers

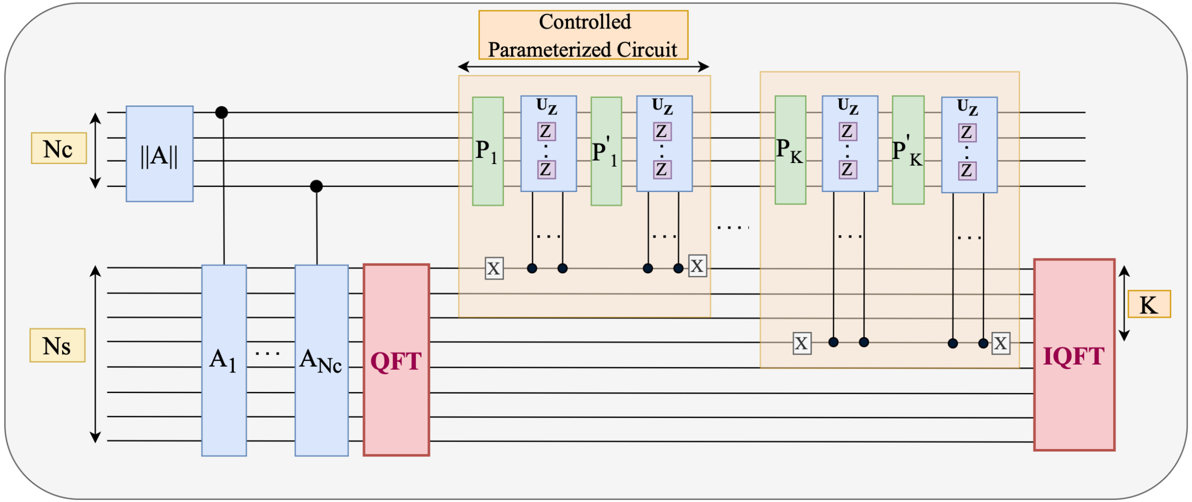


Unary Data Loader for Matrix

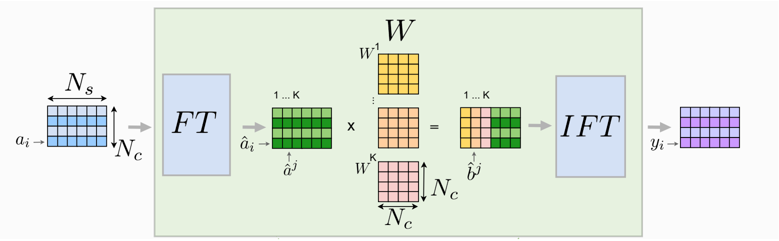


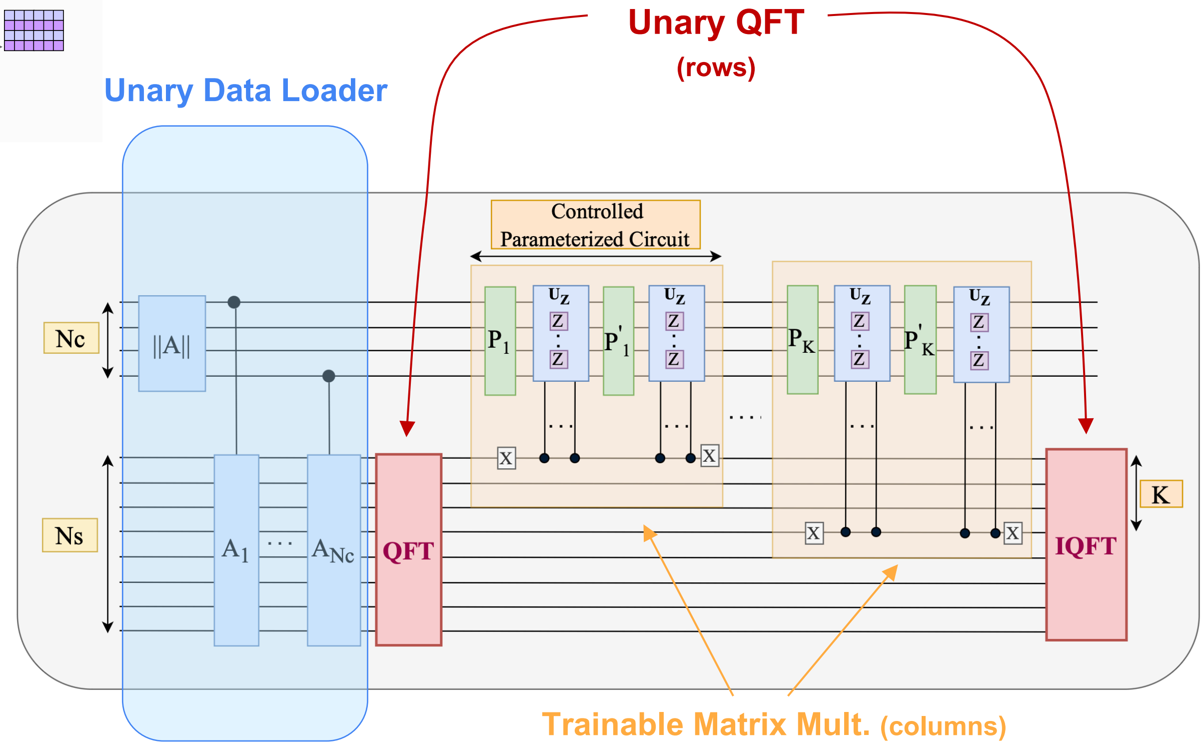
From Classical to Quantum FNO





From Classical to Quantum FNO





Unary Quantum Fourier Transform (QFT)

Classical FFT algorithm

$$F = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix}$$

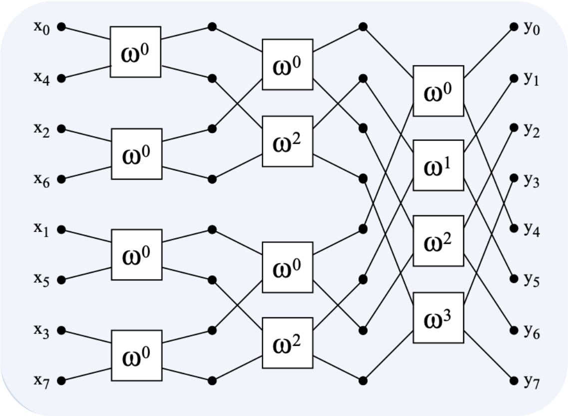
$$\omega = e^{-\frac{2\pi i}{n}}$$

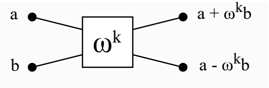
n^{th} square root of unity

Unary Quantum Fourier Transform (QFT)

Classical

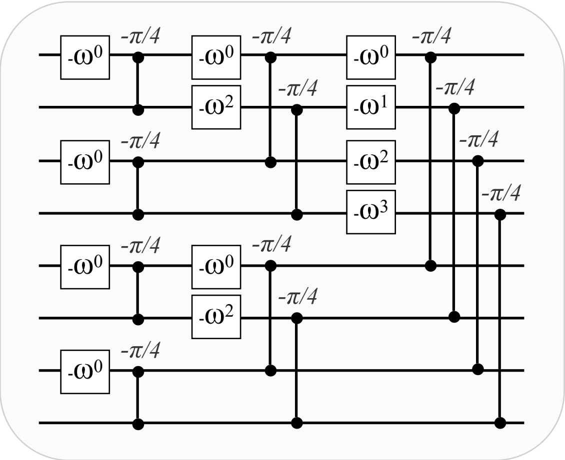
$$x \mapsto Fx$$

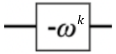




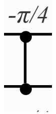
Quantum

$$|x\rangle \mapsto |Fx\rangle$$



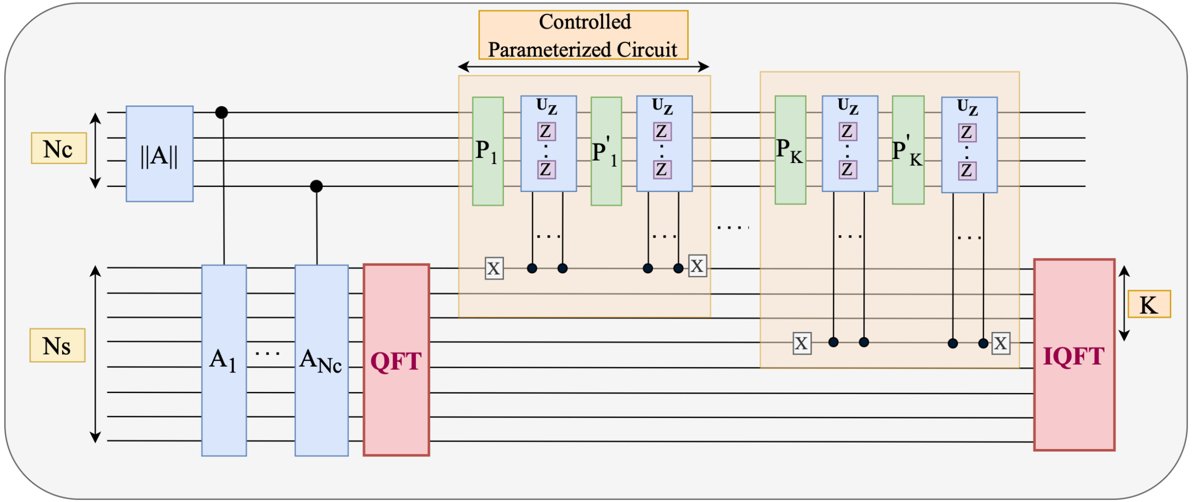


$$= \begin{pmatrix} 1 & 0 \\ 0 & -e^{-i\frac{2\pi k}{N}} \end{pmatrix}$$



$$= RBS\left(\frac{\pi}{4}\right)$$

Sequential QFNO



Resource analysis:

#circuits = 1

#qubits = $N_C + N_S$

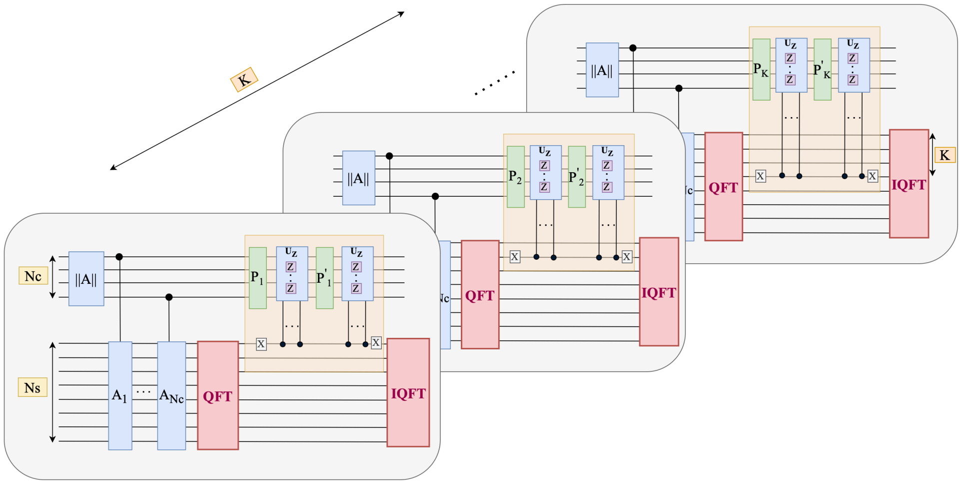
#gates = $KN_C \log N_C + N_C N_S \log N_S$

Depth = $\mathbf{N_C \log N_S} + K \log N_C + KN_C$

Classical Complexity = $N_C + \mathbf{N_S \log N_S}$

Three Different Quantum Circuits

Parallel QFNO



Resource analysis:

#circuits = **K**

#qubits = $N_C + N_S$

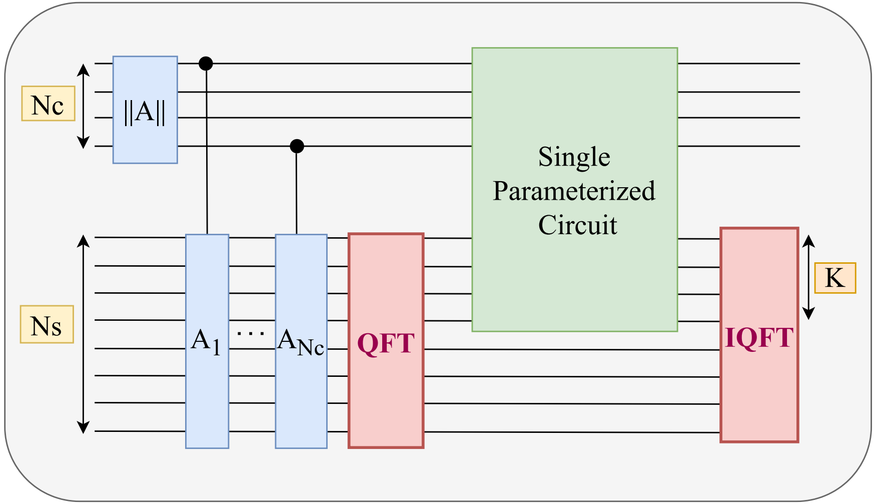
#gates = $KN_C \log N_C + KN_C N_S \log N_S$

Depth = $N_C \log N_S + \boldsymbol{\log N_C} + N_C$

Classical Complexity = $N_C + N_S \log N_S$

Three Different Quantum Circuits

Composite QFNO



Resource analysis:

#circuits = **1**

#qubits = $N_C + N_S$

#gate = $(K + N_C)\log(N_C{+}K) + N_C N_S \log N_S$

Depth = $N_C \log N_S + \boldsymbol{\log(N_C{+}K)} + N_C + K$

Classical Complexity = $N_C + N_S \log N_S$

Experimental Results

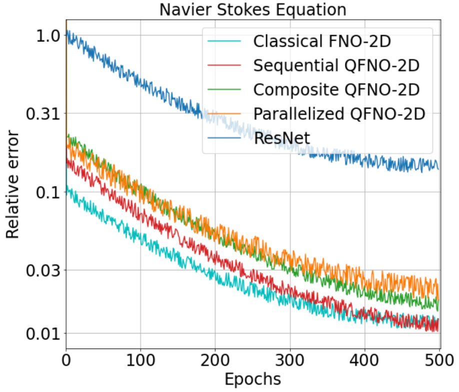


Figure 12: Convergence comparison for the Navier-Stokes equation with $v = 1e{-}3$, trained for 500 epochs.

Method	Classical FNO	Sequential QFNO	Parallelised QFNO	Composite QFNO
Parameters	294,912	23,040	23,040	6,144
$\nu = 1e - 3;\ T = 50$	0.0139	0.0148	0.0167	0.0186
$\nu = 1e - 4;\ T = 30$	0.1603	0.1618	0.1633	0.1660
$\nu = 1e - 5;\ T = 20$	0.1601	0.1615	0.1626	0.1638

Table 2: Comparison of parameters required by one layer of the proposed circuits and the existing classical Fourier Layer along with error analysis for different ν and T values for the 2D case of a Navier-Stokes equation.

Conclusion

Conclusion

Hamming Weight Preserving QML is cool

Thank you