Intro to Quantum Computing ML for Neutrino Astronomy

August 24th, 2023
NPML 2023
Tufts University

Pavel Zhelnin
With Jeff L. Et al.
Harvard University
pzhelnin@g.harvard.edu

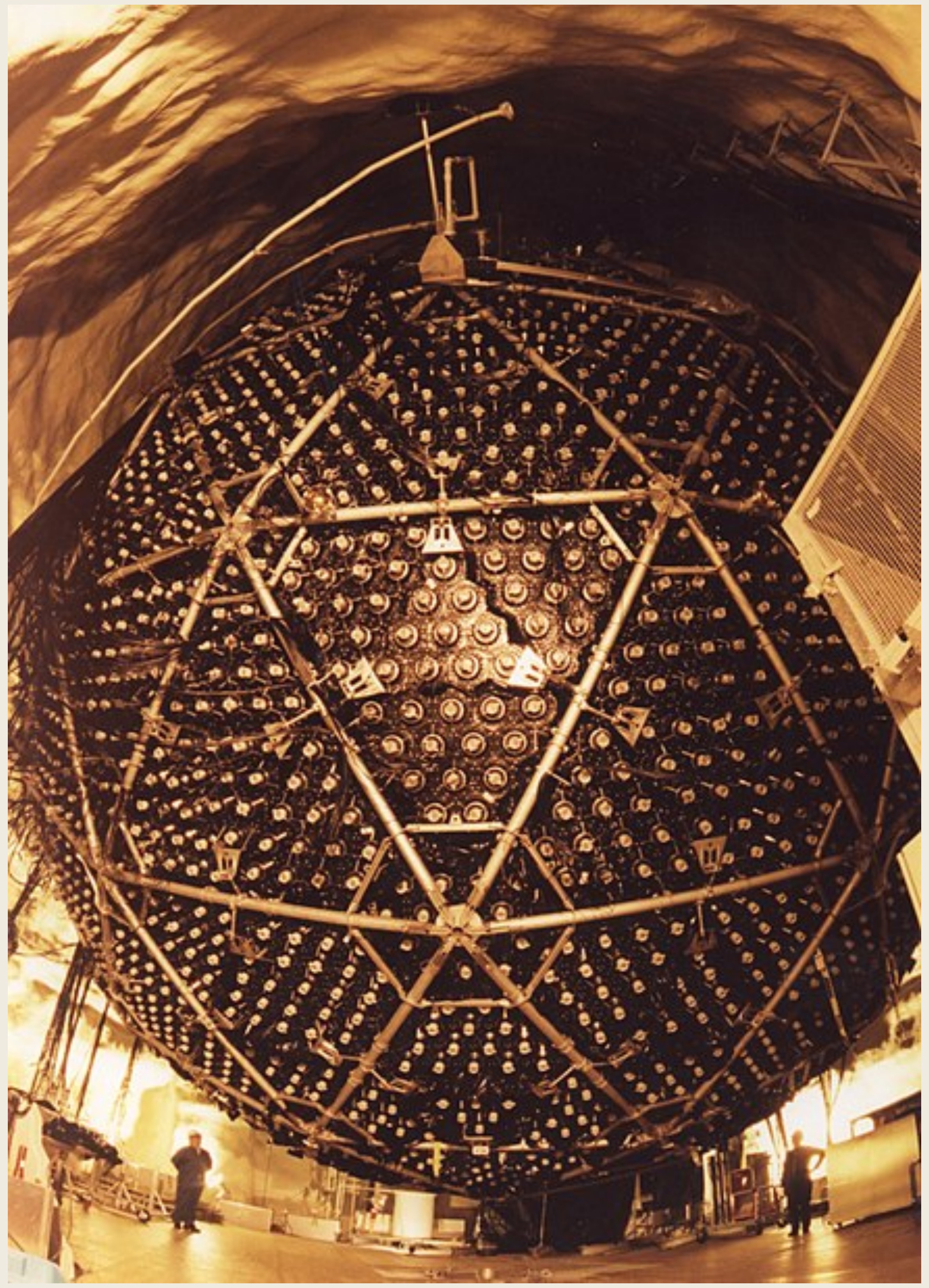
Machine Learning

- In the 1990s the initial exploration of ML in particle physics began: SNO experiment
- At the beginning, these neural networks did not outperform other statistical techniques but they did demonstrate capabilities
- However as expertise grew ML techniques began to surpass traditional reco
- Now ML has played a role in nearly every particle physics discovery and measurement since

Observation of high-energy neutrinos from the Galactic plane

ICECUBE COLLABORATION, R. ABBASI, M. ACKERMANN, J. ADAMS, J. A. AGUILAR, M. AHLERS, M. AHRENS, J. M. ALAMEDDINE, A. A. ALVES JR., [...], AND P. ZHELNIN

+380 authors Authors Info & Affiliations



https://en.wikipedia.org/wiki/Sudbury_Neutrino_Observatory

Machine Learning

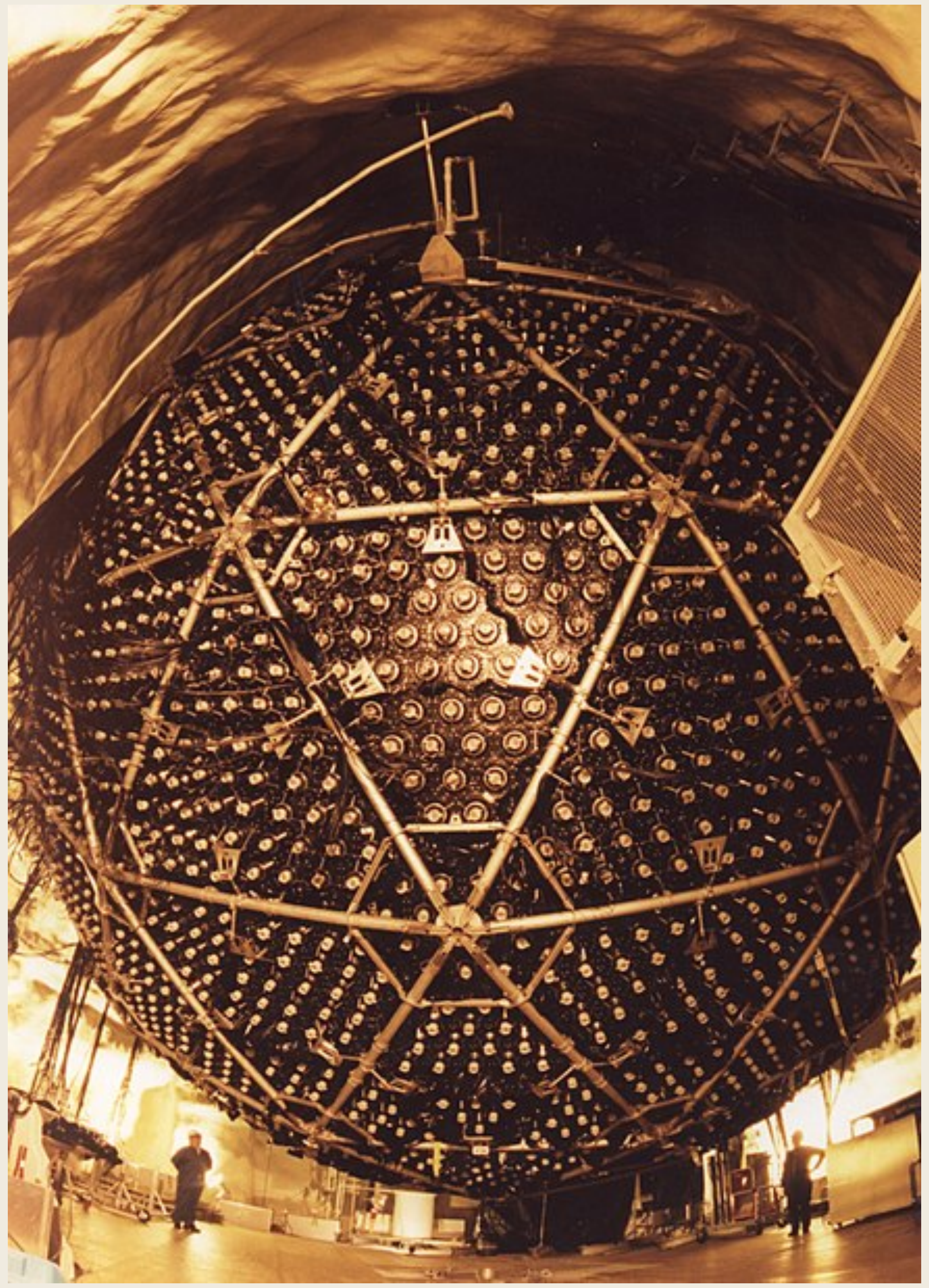
- In the 1990s the initial exploration of ML in particle physics began: SNO experiment
- At the beginning, these neural networks did not outperform other statistical techniques but they did demonstrate capabilities
- However as expertise grew ML techniques began to surpass traditional reco
- Now ML has played a role in nearly every particle physics discovery and measurement since

Observation of high-energy neutrinos from the Galactic plane

ICECUBE COLLABORATION, R. ABBASI, M. ACKERM

+380 authors Authors Info & Affiliati

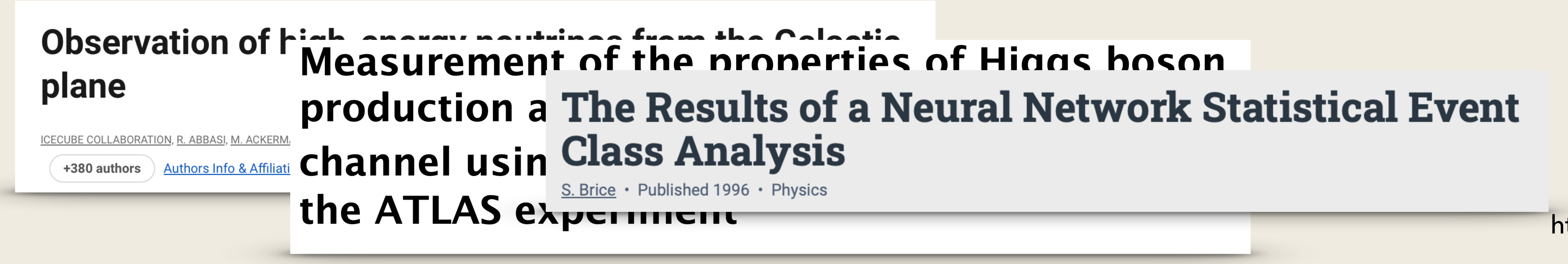
Measurement of the properties of Higgs boson production at $\sqrt{s} = 13$ TeV in the $H \rightarrow \gamma\gamma$ channel using 139 fb^{-1} of pp collision data with the ATLAS experiment

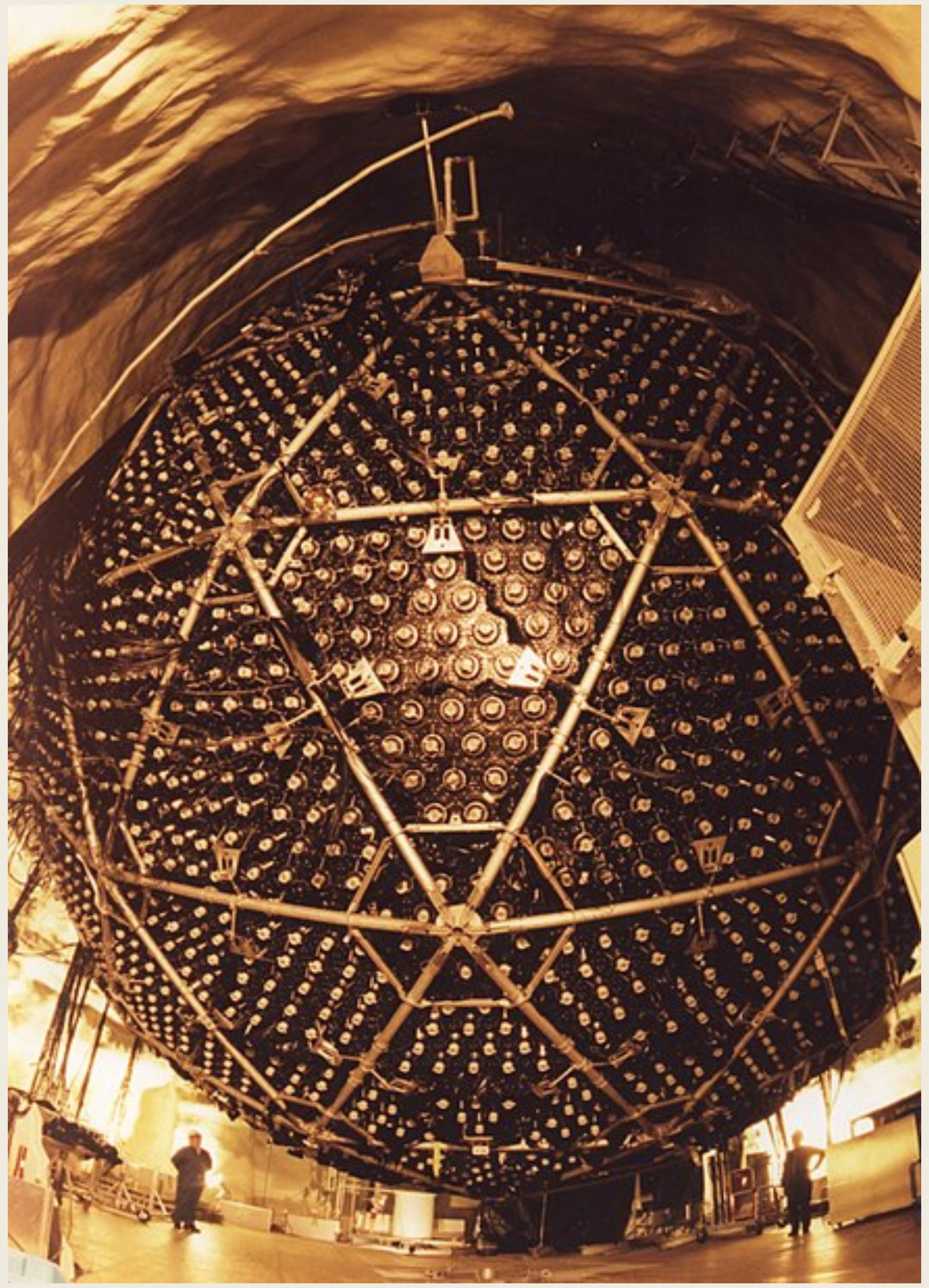


https://en.wikipedia.org/wiki/Sudbury_Neutrino_Observatory

Machine Learning

- In the 1990s the initial exploration of ML in particle physics began: SNO experiment
- At the beginning, these neural networks did not outperform other statistical techniques but they did demonstrate capabilities
- However as expertise grew ML techniques began to surpass traditional reco
- Now ML has played a role in nearly every particle physics discovery and measurement since
- What's the next iteration?

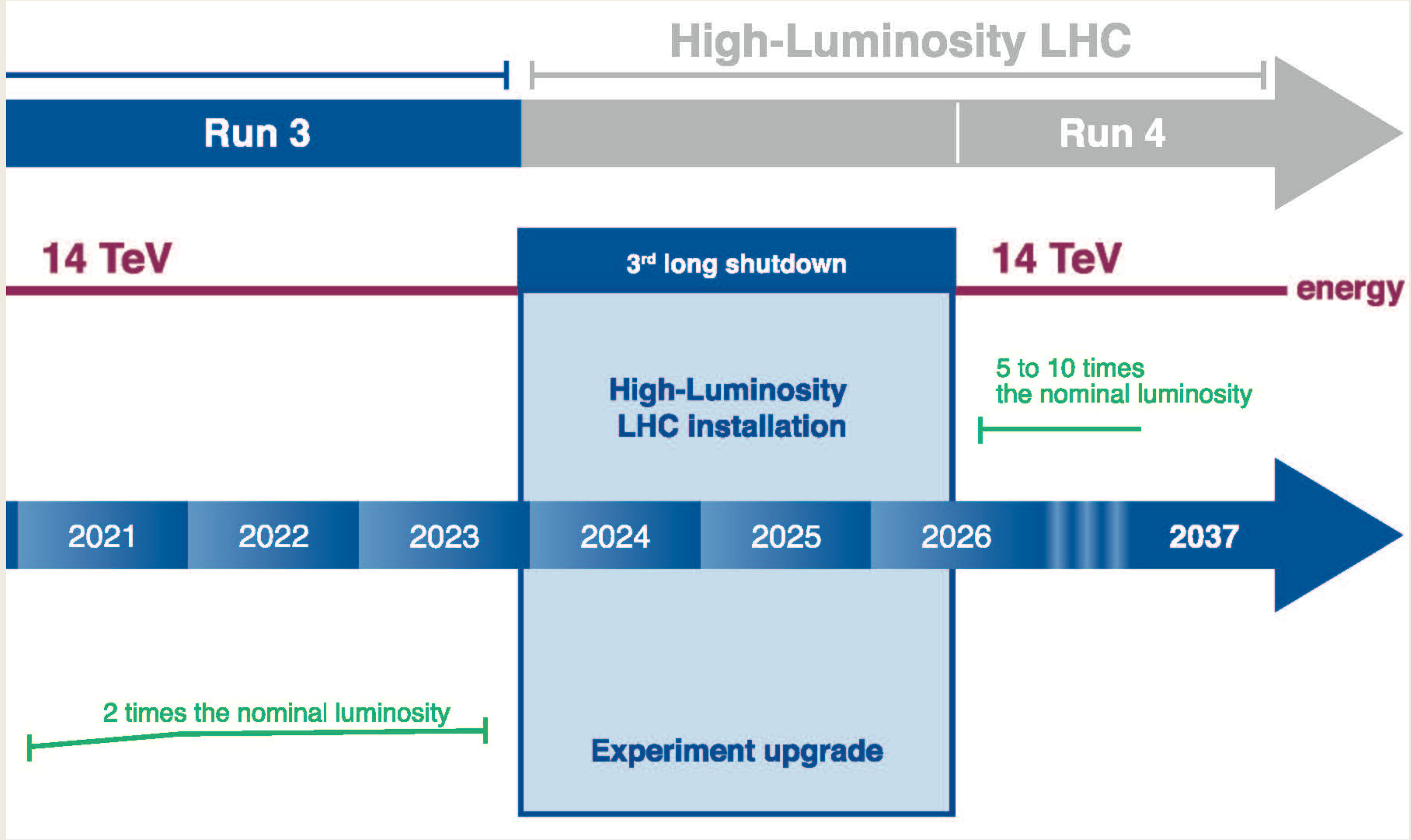

Observation of h
plane
ICECUBE COLLABORATION, R. ABBASI, M. ACKERM
+380 authors
Authors Info & Affiliati
Measurement of the properties of Higgs boson
production a
channel usin
the ATLAS ex
The Results of a Neural Network Statistical Event Class Analysis
S. Brice • Published 1996 • Physics



https://en.wikipedia.org/wiki/Sudbury_Neutrino_Observatory

A Growing Data Challenge

- ML is essential in analyzing a commonality among experiments now: large data size
- In fact cuts are needed to manage modern experiments
- Even after cuts, datasets are huge
  - CERN produces > 300 TB of data per day
  - IceCube produces ~ 1 TB
- Templates based on our current understanding filters data
- Furthermore, next generation experiments will increase data output by an order of magnitude
- Could new physics be hiding in cut data?

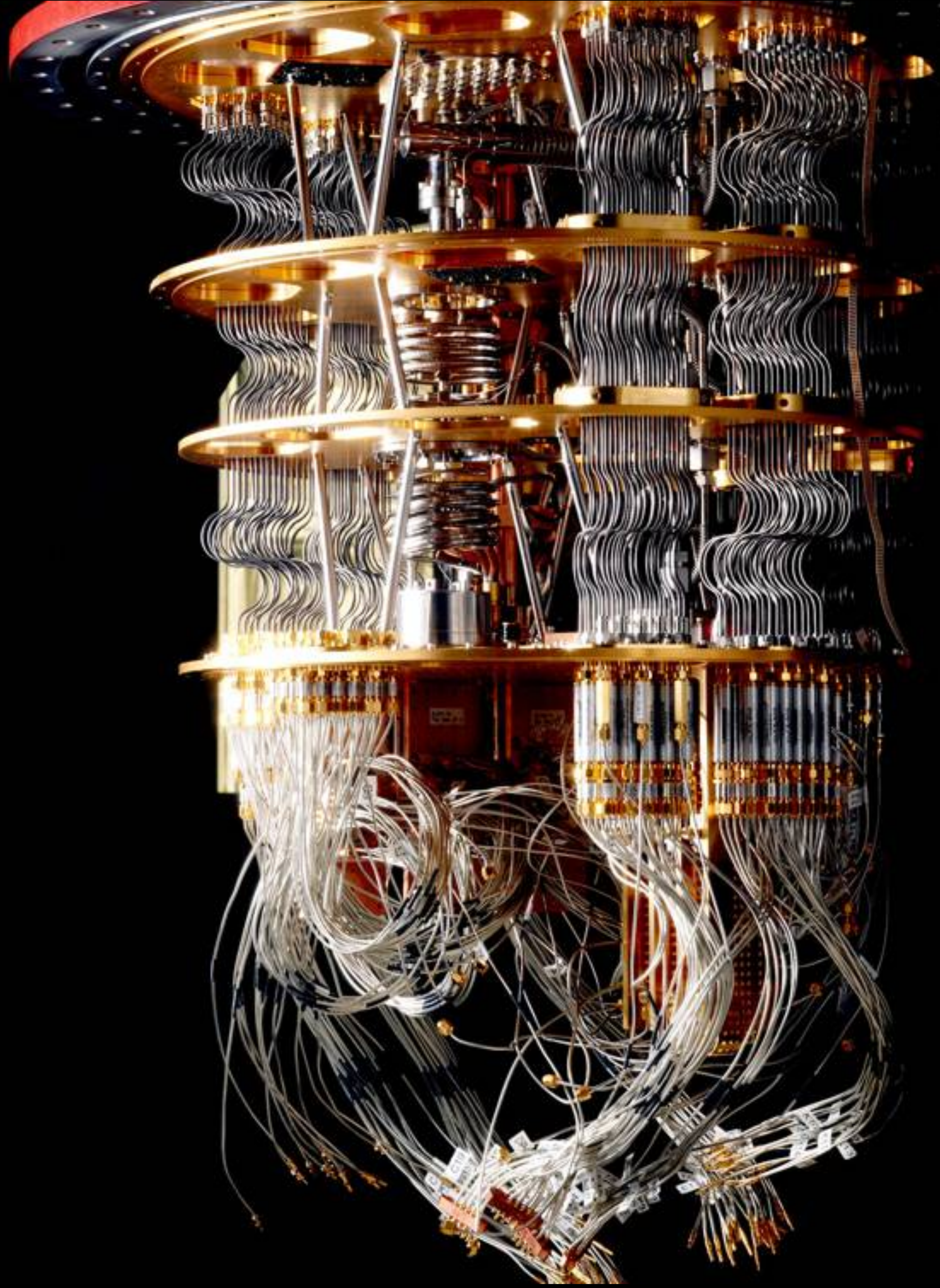


A Growing Challenge cont.

- We could be missing new physics due to un-modeled interactions **(streetlight effect)**
- Allowing additional data may be necessary for new physics
- For this, a paradigmatic shift is needed in data management to process trigger-level data

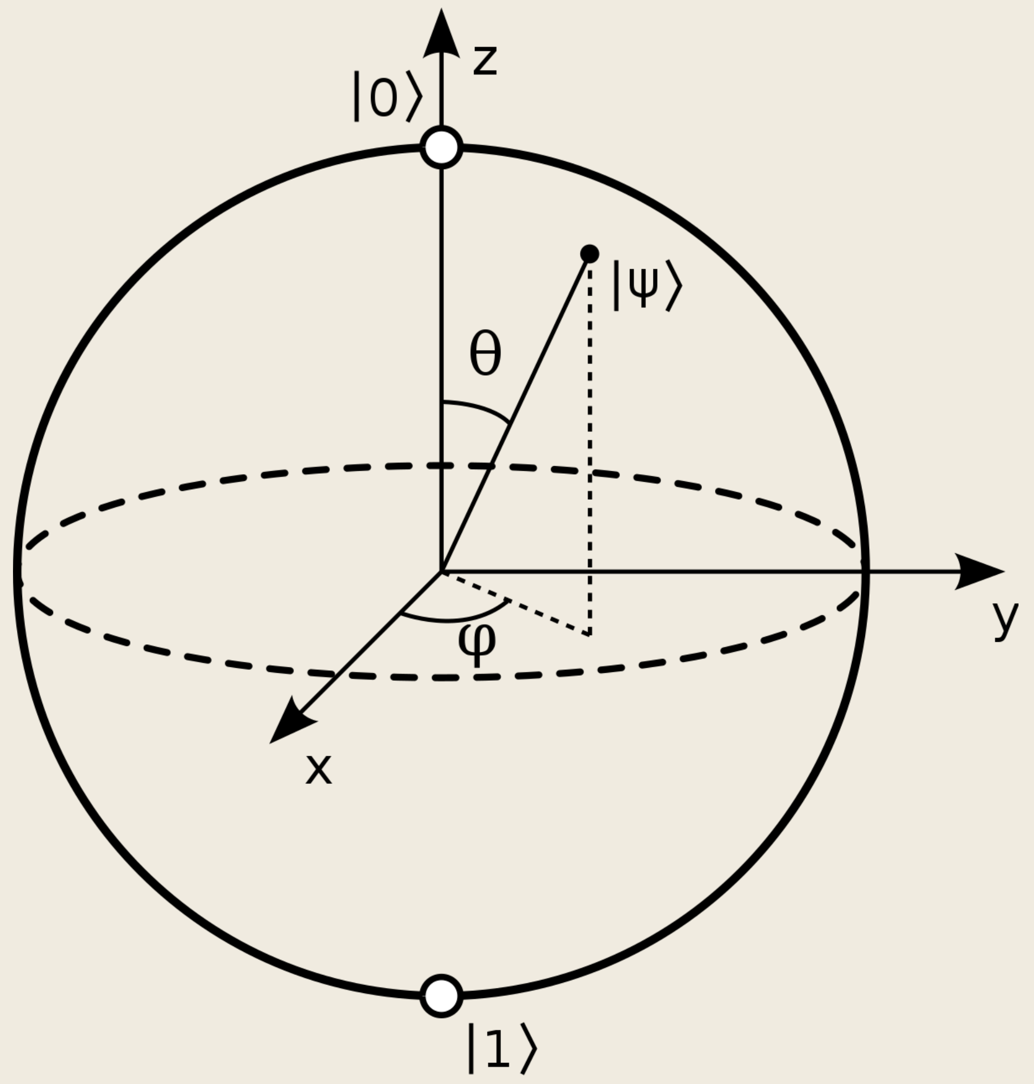
Quantum Computing

- A computer whose computations can only be described with the laws of quantum theory
  - **Exponentially large Hilbert space**
  - Entanglement
  - Superposition
  - Interference
- 2^N advantage over classical computers
- E.g. 8 classical bits → 3 "qubits", 64 bits → 6 "qubits", can store all of Google Drive cloud storage in ~60 qubits



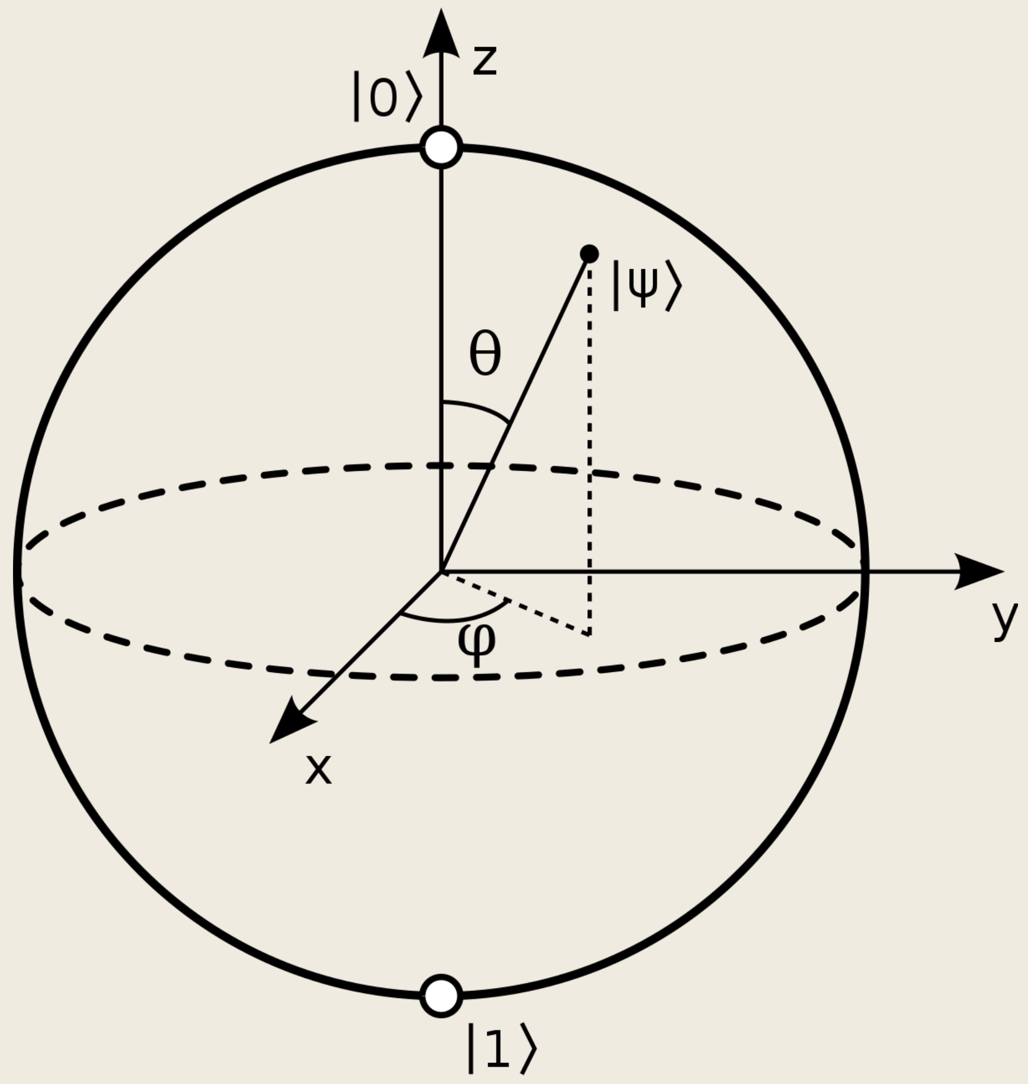
Basics of Quantum Computing

- **Qubits** = basic unit of information in a QC (akin to a bit)
- Often represented by a Bloch sphere

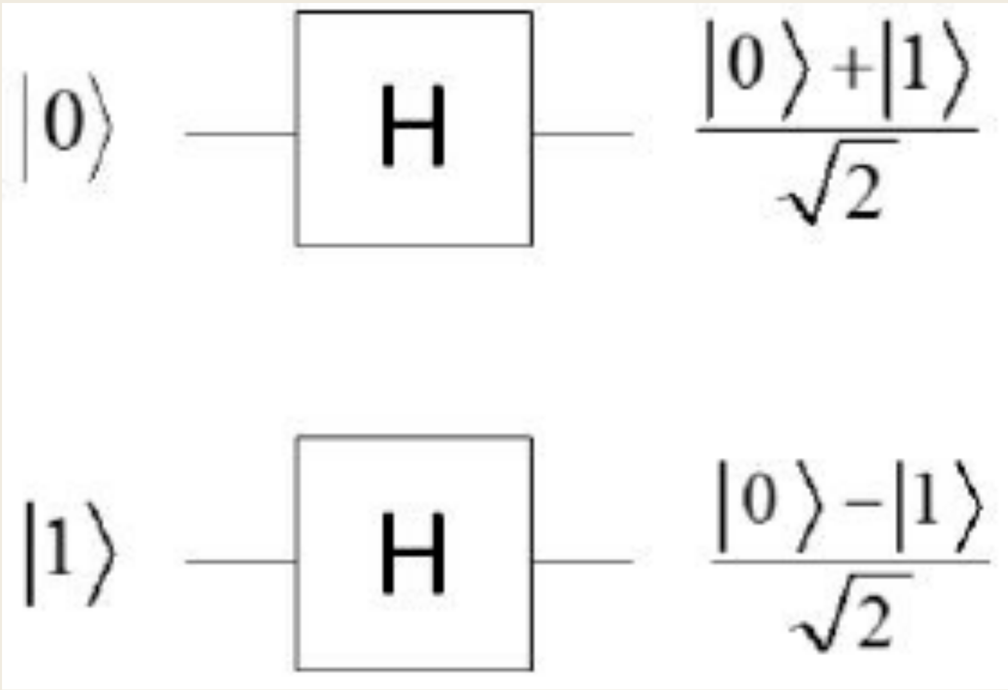


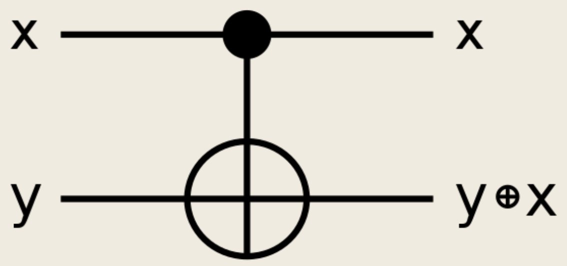
Basics of Quantum Computing

- **Qubits** = basic unit of information in a QC (akin to a bit)
- Often represented by a Bloch sphere

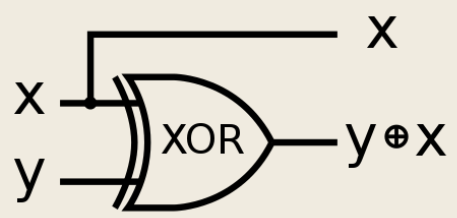


- **Quantum gates** = most basic operation that can performed on a qubit (or set of qubits)
- Two basic quantum gates: Hadamard/CNOT
- Hadamard creates superpositions
- CNOT entangles
- Combination makes a Bell State





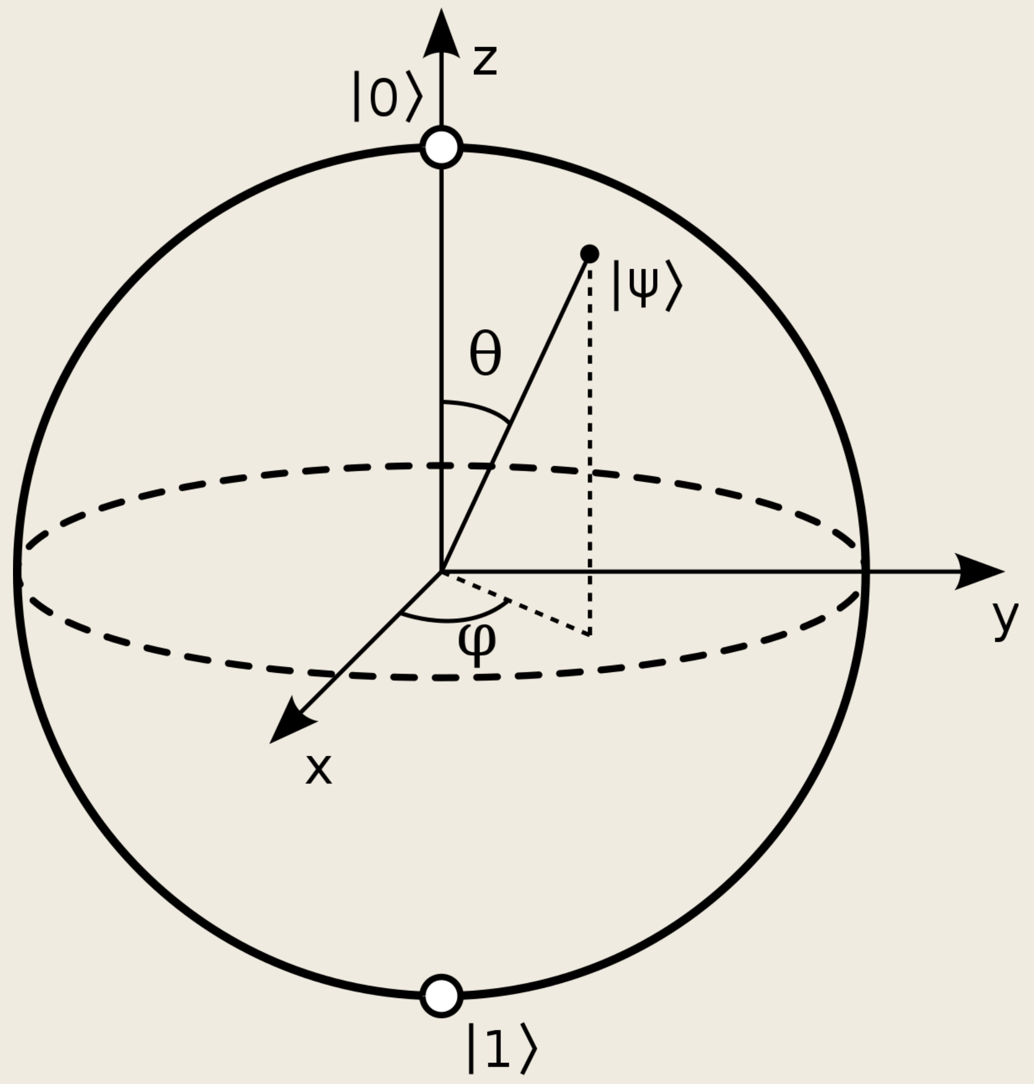
input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩



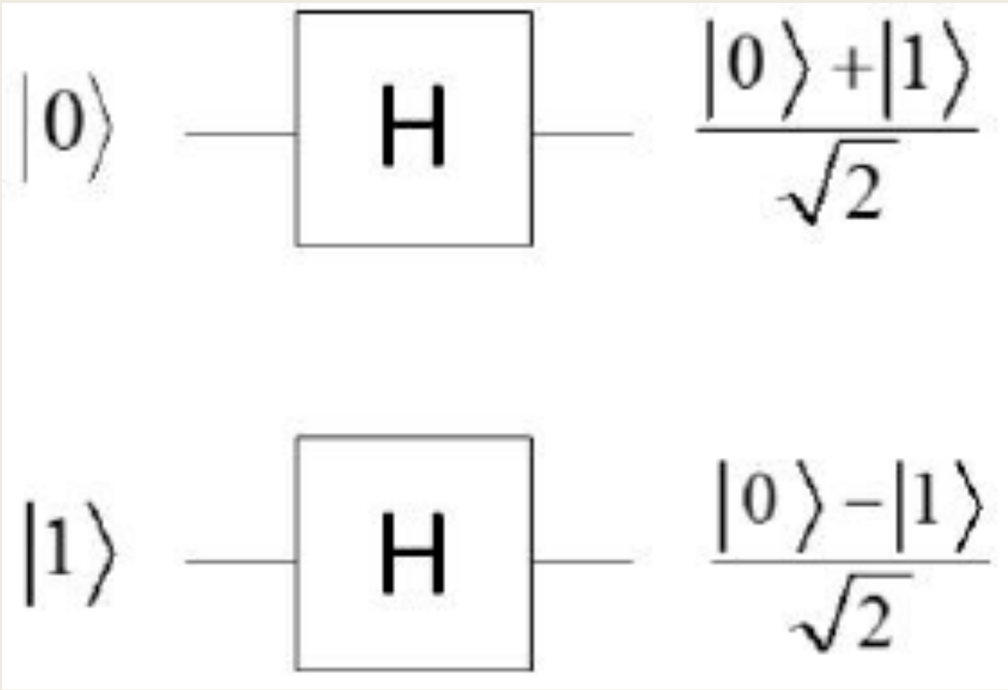
input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

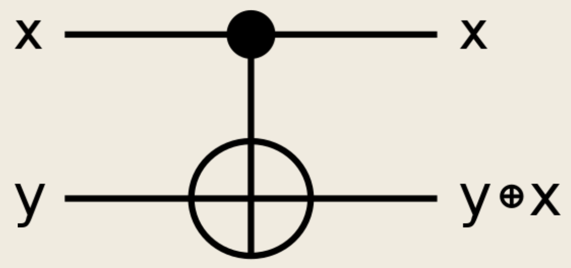
Basics of Quantum Computing

- **Qubits** = basic unit of information in a QC (akin to a bit)
- Often represented by a Bloch sphere

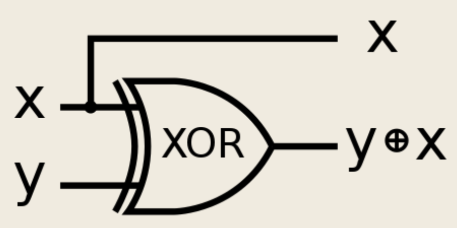


- **Quantum gates** = most basic operation that can performed on a qubit (or set of qubits)
- Two basic quantum gates: Hadamard/CNOT
- Hadamard creates superpositions
- CNOT entangles
- Combination makes a Bell State



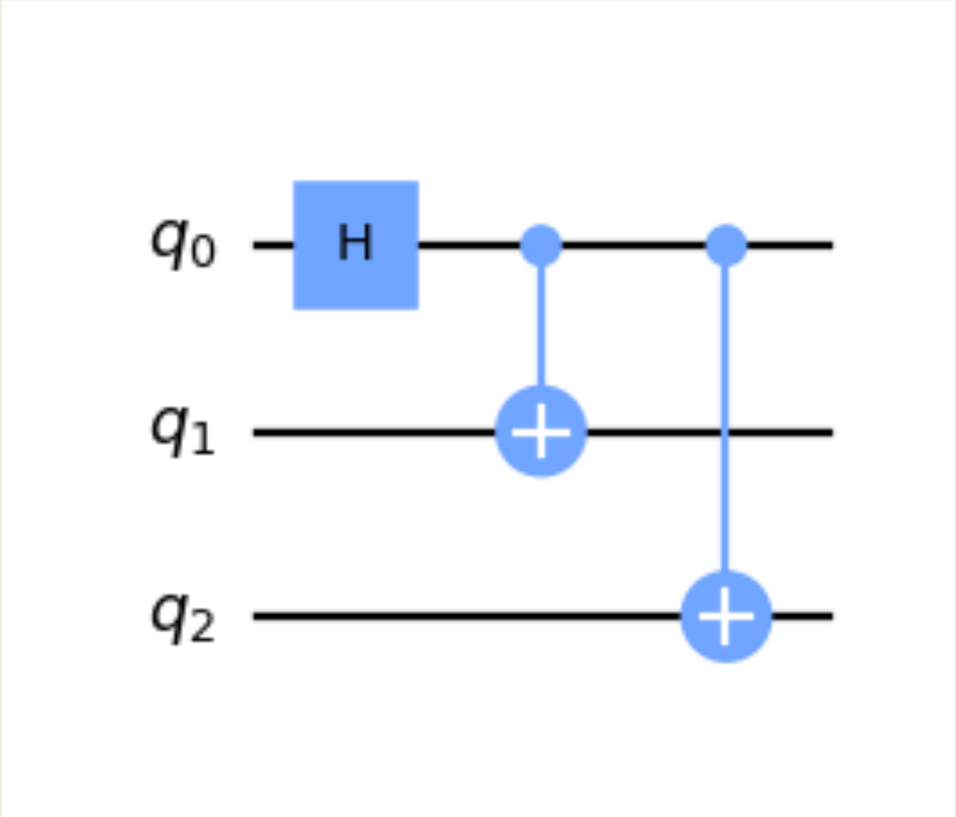


input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩



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

- **Quantum circuit** = a model for quantum computation in which a sequence of quantum gates are applied to a set of n qubits

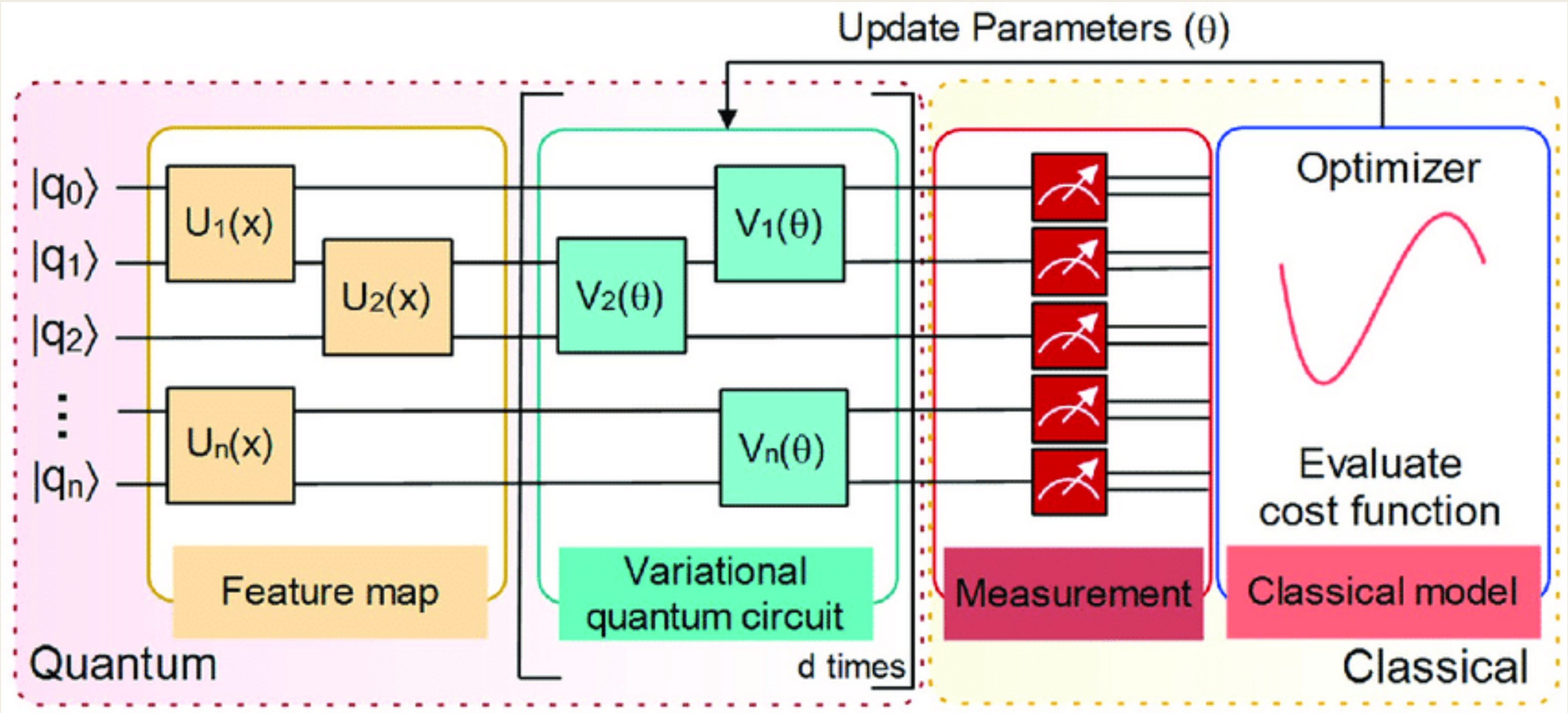


A Data Processing Pipeline Using QC

- We don't want to just store data on a QC, we want to process it as well
- Its runtime is costly to do: classical → quantum or quantum → classical transfers of data
- We want a fully "quantum pipeline", no classical preprocessing

E.g. …

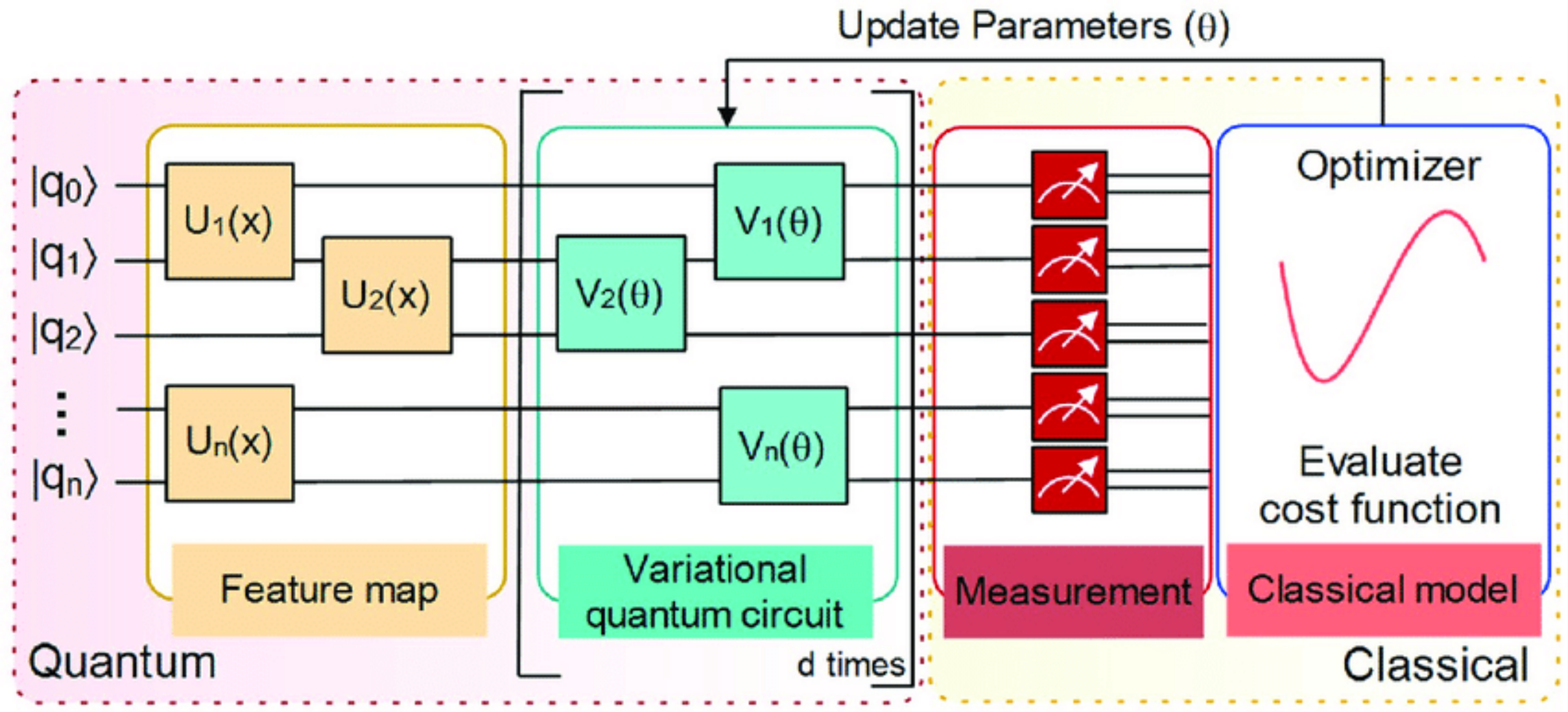
Classical Data →



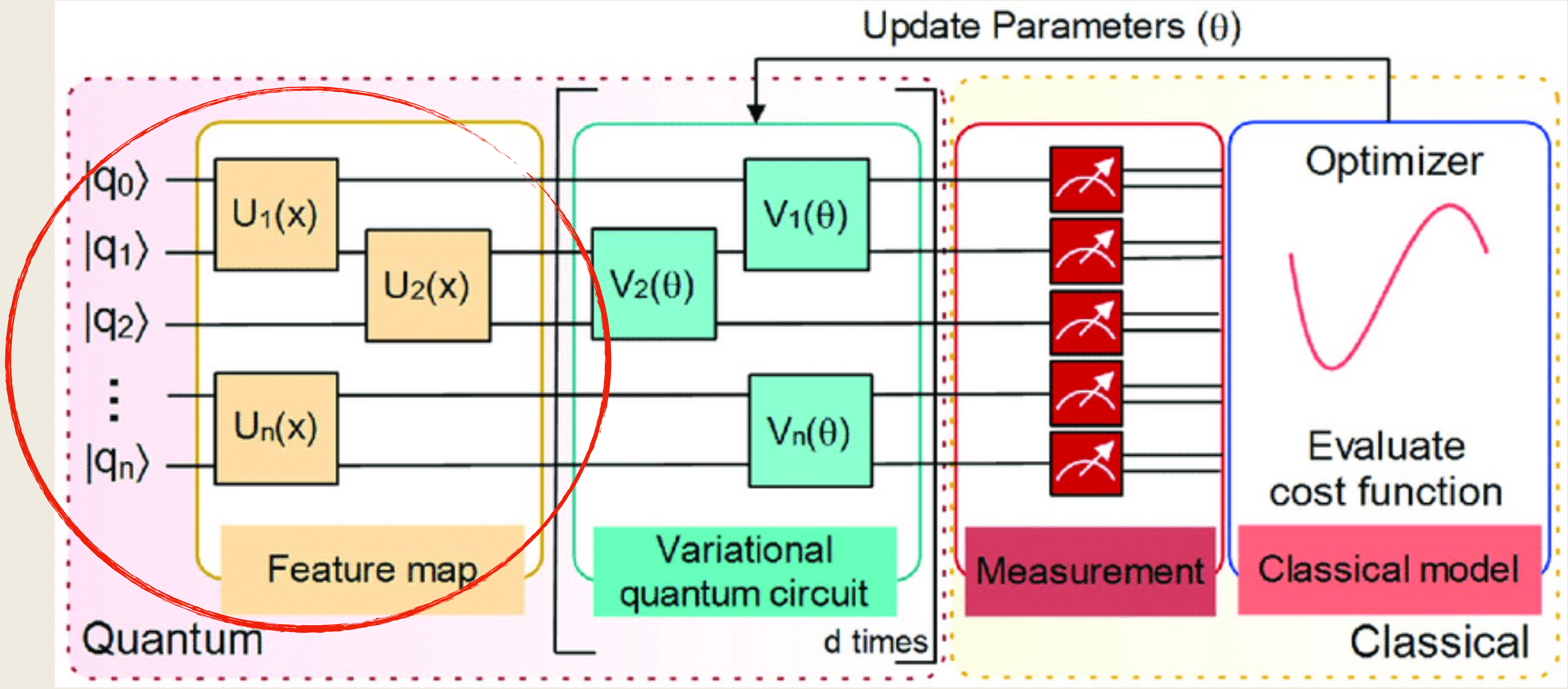
→ Classification

Rest of the talk: QML with a Variational Quantum Circuit

A VQC is a low depth, low width choice suitable for ML applications on current quantum computers.



Rest of the talk: QML



Data Encoding

- "Data encoding is often the most crucial step in QML with classical data: it influences potential quantum advantage, learning performance and runtime."
- Most other QML encoding schemes involve some classical preprocessing then using either amplitude/basis encoding (arXiv:2012.11560, arXiv.1907.00397, arXiv.2010.07335)
- We want to avoid classical preprocessing while still working within the constraints of Near-Intermediate Scale Quantum (NISQ) computers

Shuld et al. Machine Learning with Quantum Computers

Background: Data Encoding

- **Amplitude encoding** can store information with 2^N efficiency
  - Susceptible to decoherence

$$\mathbf{x} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$|\psi_{\mathbf{x}}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle.$$

$$|\mathbf{x}\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

Background: Data Encoding

- **Amplitude encoding** can store information with 2^N efficiency
  - Susceptible to decoherence
- **Basis encoding** is the simplest encoding
  - No quantum advantage, a 1 to 1 mapping

$$\mathbf{x} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$|\psi_{\mathbf{x}}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle.$$

$$|\mathbf{x}\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

$$x = 1011 \rightarrow |1011\rangle$$

Background: Data Encoding

- **Amplitude encoding** can store information with 2^N efficiency
  - Susceptible to decoherence

$$\mathbf{x} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$|\psi_{\mathbf{x}}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle.$$

$$|\mathbf{x}\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

- **Basis encoding** is the simplest encoding
  - No quantum advantage, a 1 to 1 mapping

$$x = 1011 \rightarrow |1011\rangle$$

- **Angle encoding**
  - Rotations around principle axes of Bloch sphere
  - Principle encoding scheme by others in QML HEP

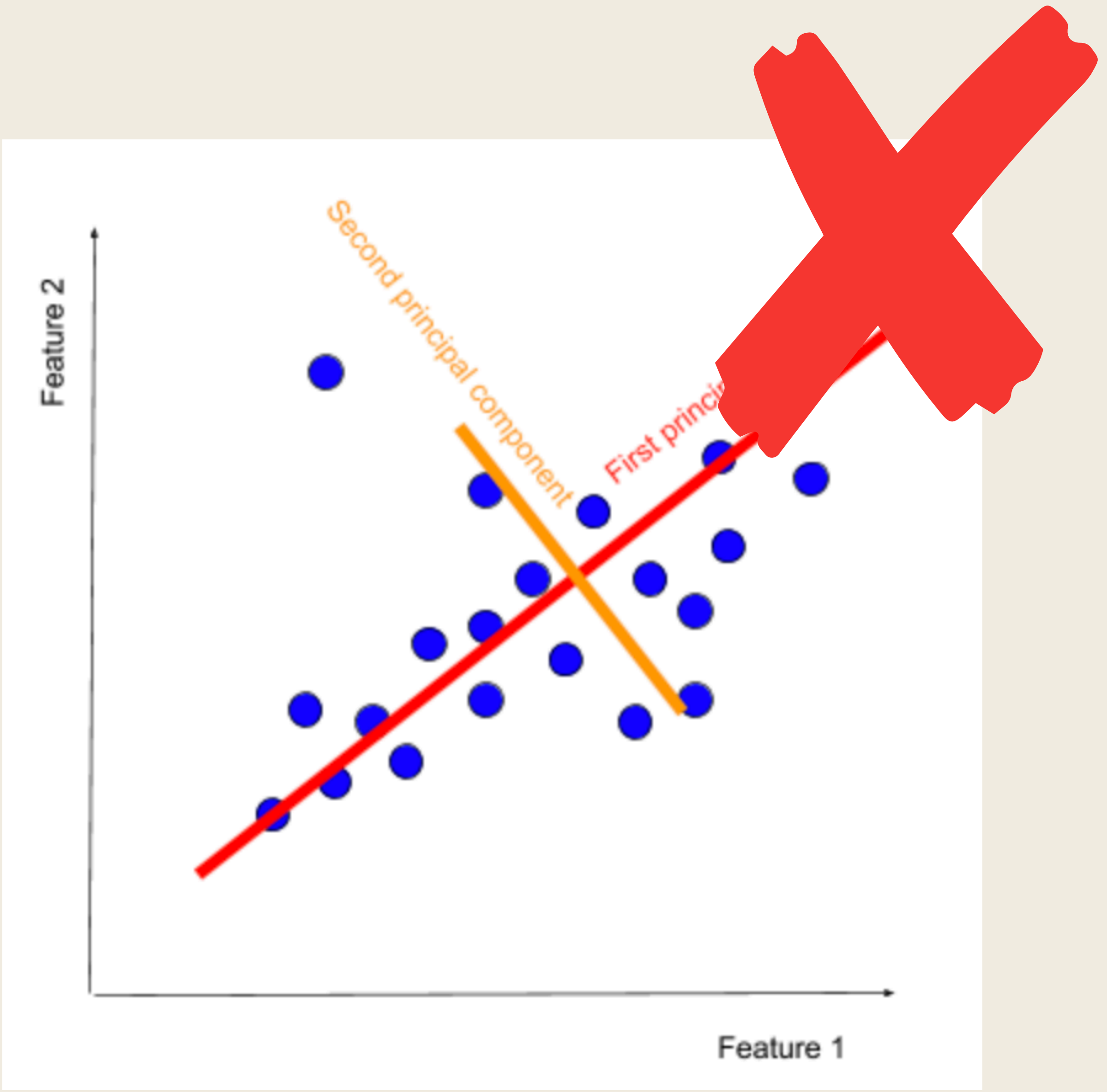
$$|\mathbf{x}\rangle = \bigotimes_{i}^{n} R(\mathbf{x}_i)\,|0^n\rangle$$

Rotation by pi around y axis on Bloch sphere

$$\mathbf{x} = \begin{bmatrix} \pi \\ \pi \\ \pi \end{bmatrix} \rightarrow |111\rangle$$

Recap so far

- We want to use quantum computers because they can handle computational challenges of increase data loads of upcoming experiments
- This way we can investigate more data
- We don’t want to reduce the complexity of our data (no PCA, no classical dimension reduction)
- Traditional quantum encoding schemes either don’t use quantum advantage or are overly susceptible to decoherence
- So we want a near lossless quantum encoding scheme with quantum advantage

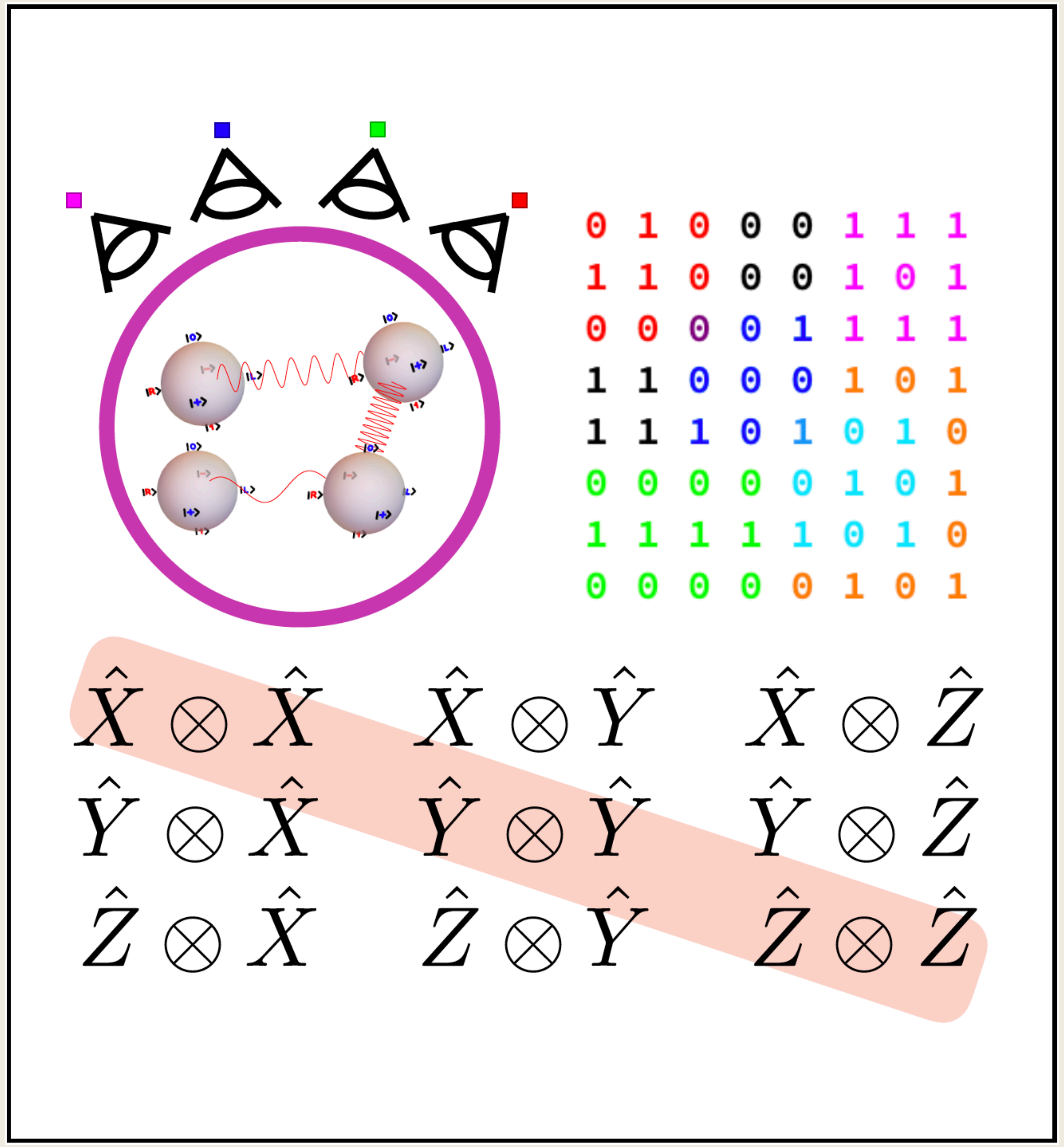


Quantum Random Access Codes (QRAC)

- We want a resilient data encoding scheme that still occupies some advantage over classical systems
- Encodes digital information in correlations between qubits
- ~a 1.5^N advantage over classical systems
- It's resilient: for N $\geq$ 18 nearly lossless (0.999) recovery rate
- For N $\geq$ 14 QRAC has greater success than classical counterparts

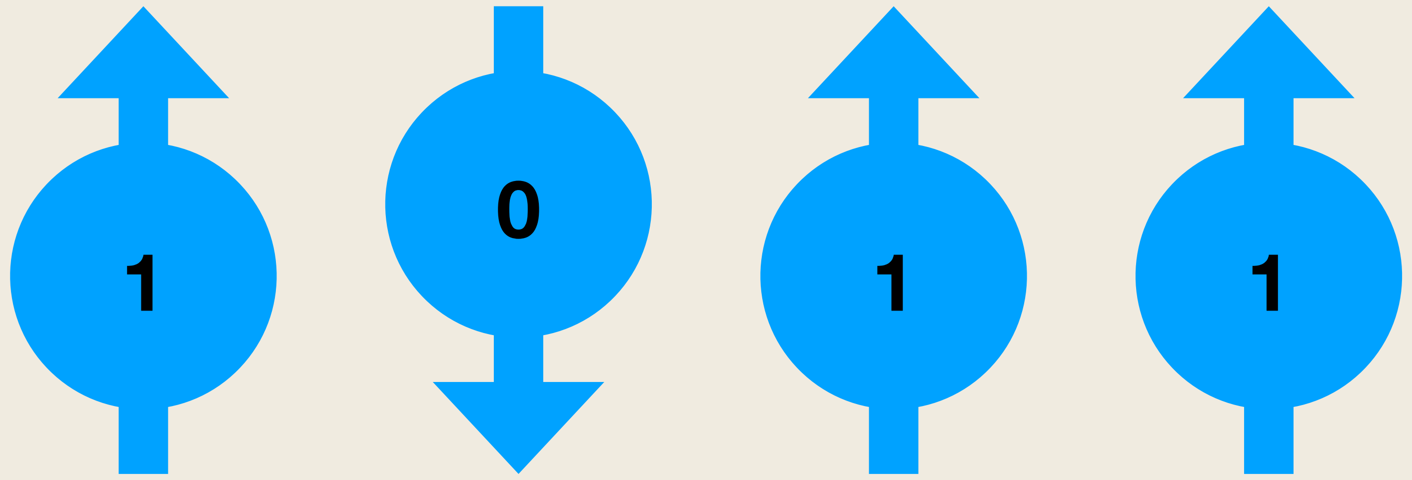
Random access codes via quantum contextual redundancy

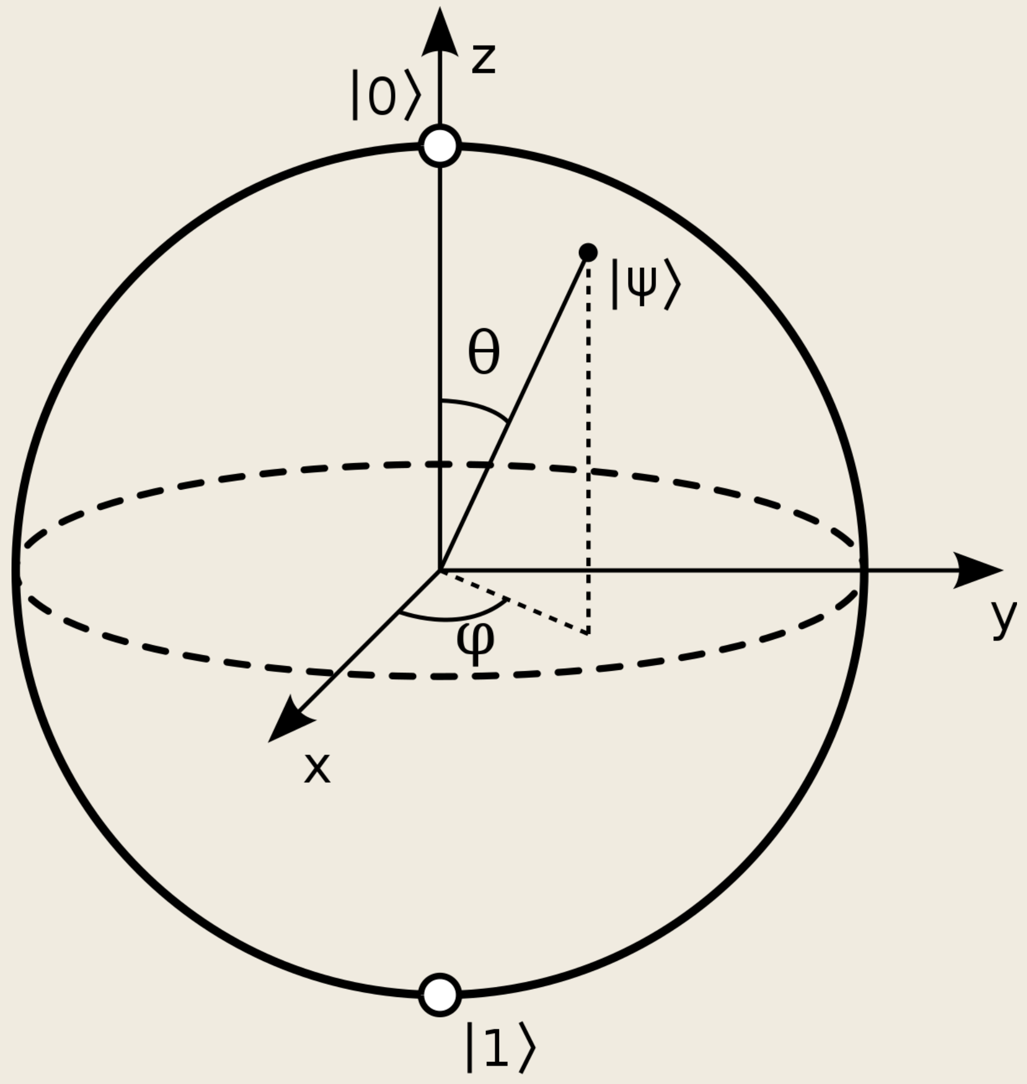
Giancarlo Gatti,[1,2,*] Daniel Huerga,[1,†] Enrique Solano,[1,3,4,5,‡] and Mikel Sanz[1,4,5,§]



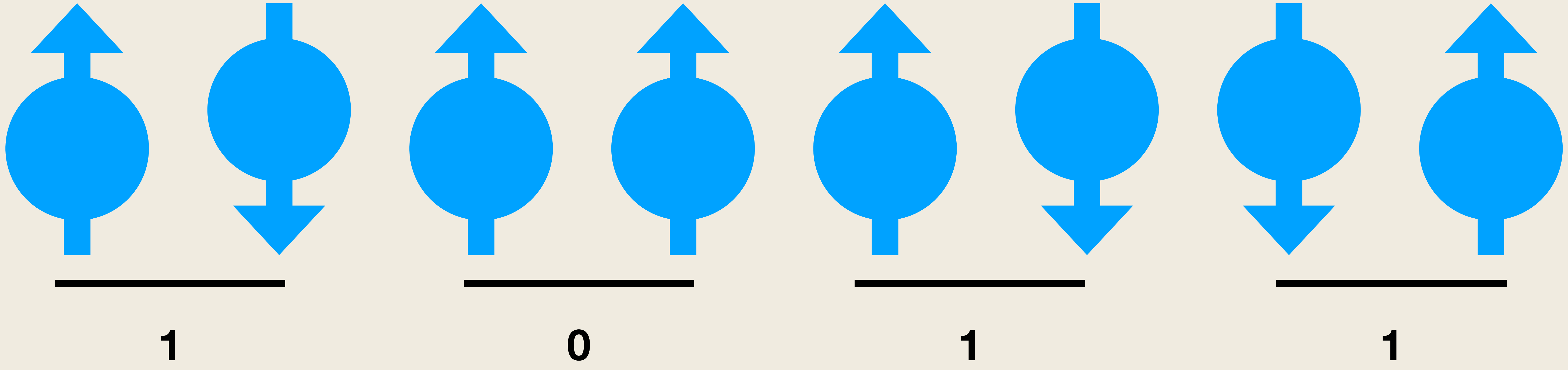
Digital Quantum Encoding

- For example, I want to encode the bit string '1011'
- Option 0: One-to-one mapping to z-spin





- Option 1: Two-to-one mapping to z-spins (have two options to encode)



Correlation-based digital encoding

- The set $\{X, Y, Z\}^{\otimes N}$ are parity observables (POs) where N is number of qubits
- Measuring using POs always yield a ± 1
- Instead of assigning a bit to each PO, assign a bit to each pair of POs, $0 \leftrightarrow =$ and $1 \leftrightarrow \neq$
- We create eigenstates of sets of commuting POs, these are our compressed states that when measured later recover our input data

b^{target}	1		0		1		1		
PO relationship	$\neq$		$=$		$\neq$		$\neq$		ZZ
POs	XX	XY	XZ	YX	YY	YZ	ZX	ZY	NA
Option 1	$+1$	-1	$+1$	$+1$	$+1$	-1	$+1$	-1	NA
Option 2	-1	$+1$	-1	-1	-1	$+1$	-1	$+1$	NA

Example for encoding '1011' in two qubits

0 Let us use n-qubit systems
(and draw figures for n=4)

1 Alice has m bits of data

1	1	0	0	1	0	0	1
0	0	0	0	1	1	1	1
1	1	0	1	1	0	1	0
1	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1

that she wants Bob to randomly access

$m = (3^n - 1)/2$

2 Alice maps her data to outcomes of n-body Pauli observables

1 $\begin{cases} X_1X_2X_3X_4 & \text{yields +1 (more than -1)} \\ X_1X_2X_3Y_4 & \text{yields -1 (more than +1)} \end{cases}$

1 $\begin{cases} X_1X_2X_3Z_4 & \text{yields -1 (more than +1)} \\ X_1X_2Y_3X_4 & \text{yields +1 (more than -1)} \end{cases}$

0 $\begin{cases} X_1X_2Y_3Y_4 & \text{yields -1 (more than +1)} \\ X_1X_2Y_3Z_4 & \text{yields -1 (more than +1)} \end{cases}$

⋮ 3^n observables

3a Alice prepares a group of n-qubit states which collectively have those outcome preferences

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$$
$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0001\rangle + i\,|1110\rangle)$$
$$|\psi_3\rangle = \frac{1}{\sqrt{2}}(|000+\rangle - |111-\rangle)$$

⋮ $O(n(3/2)^n)$ states

She sends a few copies of each state to Bob (k/n states in total)

3b E.g., the state $|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$ yields

+1 in $X_1X_2X_3X_4$, -1 in $X_1X_2Y_3Y_4$,
-1 in $X_1Y_2X_3Y_4$, -1 in $X_1Y_2Y_3X_4$,
-1 in $Y_1X_2X_3Y_4$, -1 in $Y_1X_2Y_3X_4$,
-1 in $Y_1Y_2X_3X_4$, +1 in $Y_1Y_2Y_3Y_4$
and +1 in $Z_1Z_2Z_3Z_4$, with probability 1

4 Bob measures the states with some of the observables

$|\psi_1\rangle$ $\mathcal{O}$ $|\psi_2\rangle$ $\mathcal{O}$ $|\psi_3\rangle$ $\mathcal{O}$

He finds their preferred outcomes

5 Bob reconstructs a fragment of Alice's original data

1	1	0

$m \rightarrow k$ QRAC

Optimizations

- Which set of preferred parities to chose from out of 2^N choices where N are number of couples of POs
  - First compute all compatible well-defined outcomes for all possible eigenstates
  - Compare eigenvalues from this optimization to compatible preferred parities
- Which eigenstates best represents that chosen preferred parity order
  - We want a low sampling requirement
  - Least number of states

Neutrino Astronomy: IceCube Events

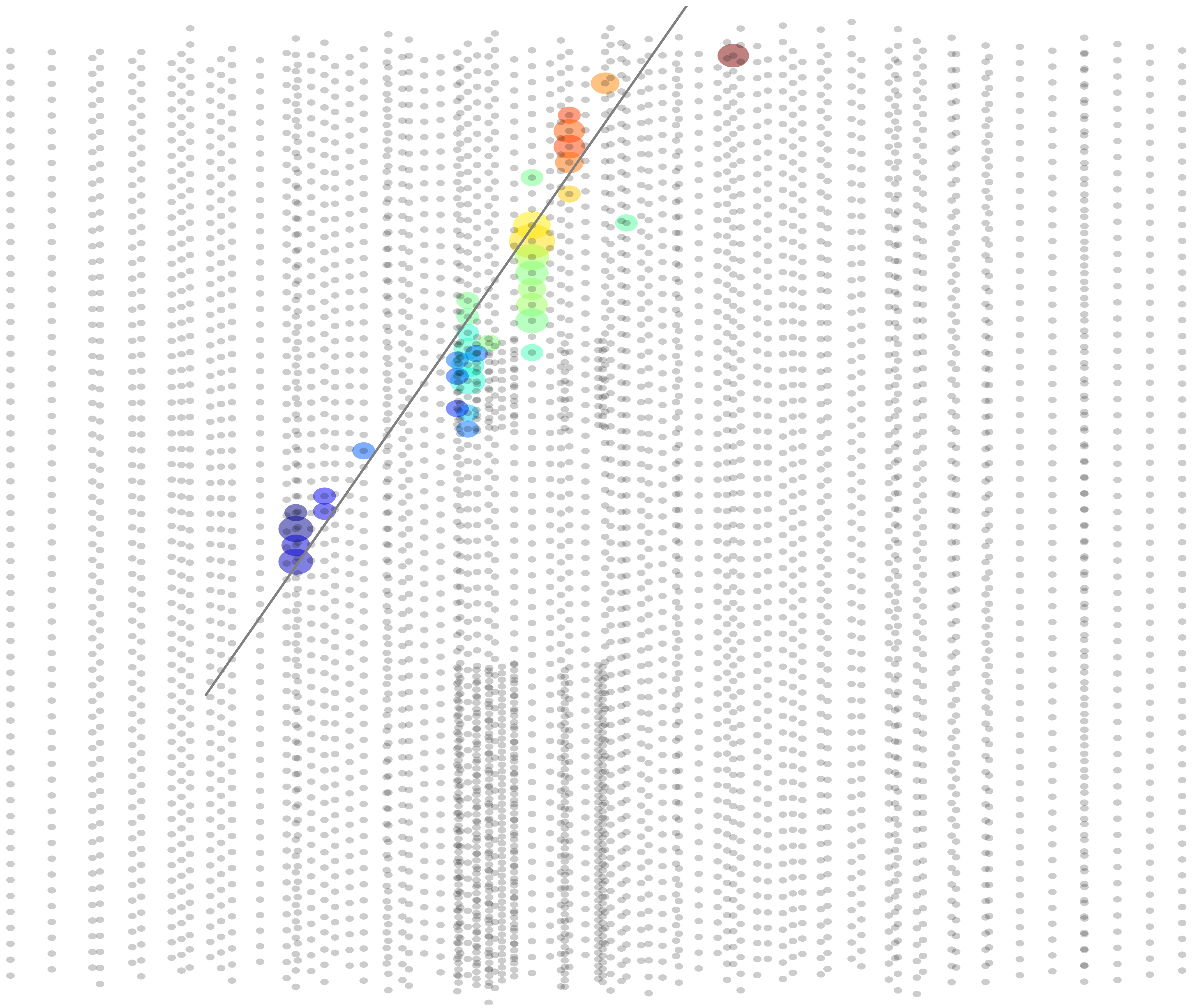
x	95.3 m	01000010101111101001100110011010
y	75.8 m	11000010100101111001100110011010
z	484.6 m	01000011111100100100110011001101
Q_{tot}	2.84 PE	01000000001101011100001010001111
$\bar{t}$	26.2 ns	01000001110100011001100110011010

0100001010111110100110011001101011000010100
1011110011001100110100100001111110010010011
0011001101010000000011010111000010100011110
1000001110100011001100110011010

Digitization Scheme: takes Optical Module (OM) position, light and time information and converts to binary. Each circle in image is a OM, size of circle indicates amount of light, color indicates time (red → purple)

This is our input to our QRAC

Event example:



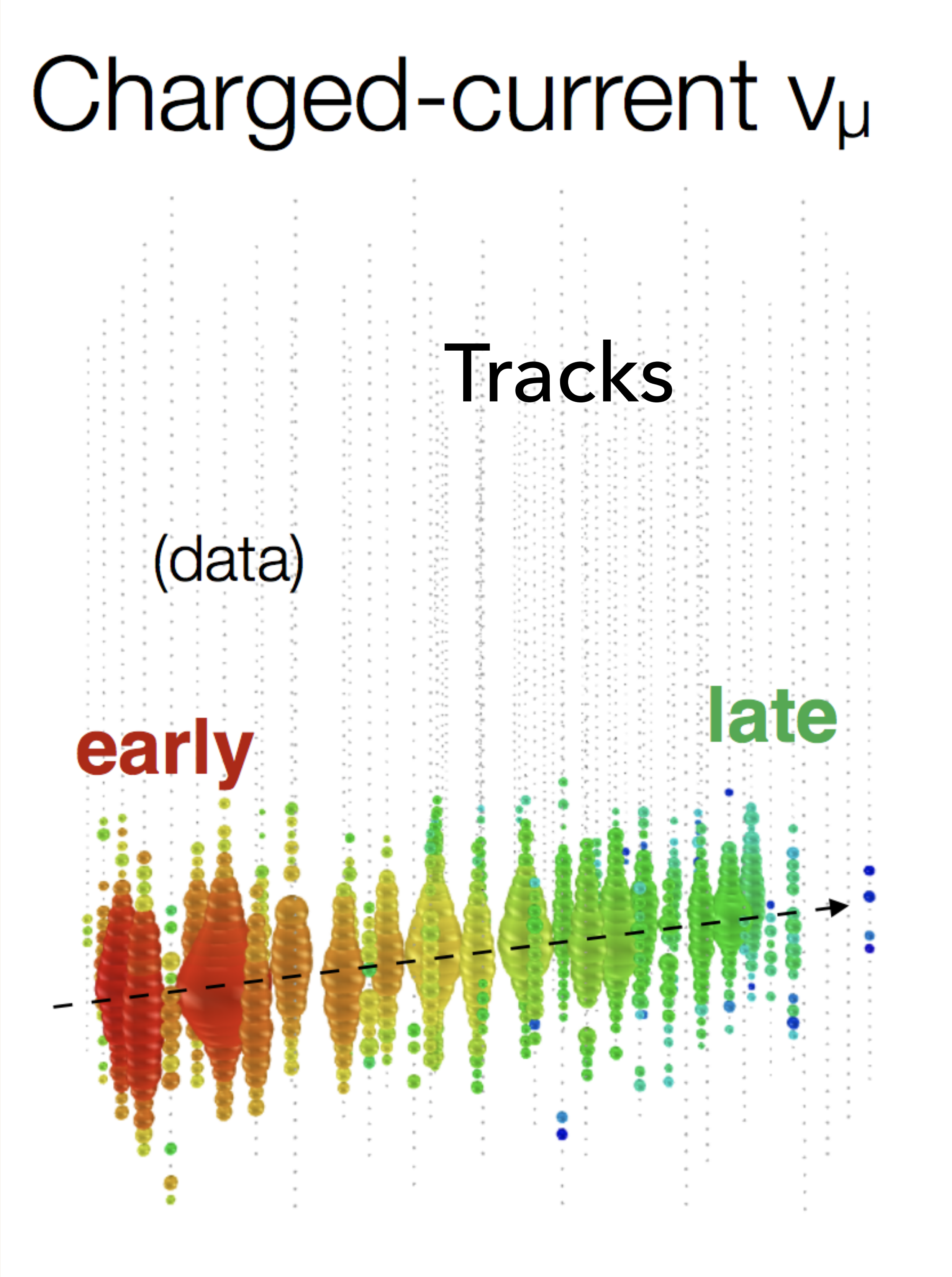
- 01000010101111101001100110011010110000101001011110011001100110100100001111110010010011001100110101
000000001101011100001010001111010000011101000110011001100110101010001111000000101100001111000001 10
01110001000011010100001100001110111011101100001101111011110000110111001000100010001011011001100011
01011010001101011110111101001100000110101000111011010010010010001111111110001011000110000001100101
00001101110000001110100010111100111010011001110111111000100010101001000011001111001100000100010100
00111010001001100010110110001101001100011001100110100100001001010111101000101100100011111010110000
011111010100111111101000110010001100101110011110010010101100010110111001010100010000110000101001101
01111000011101000111101001010010001001000100111001100100010100011001010011111101000110010110101010
00111110000111011110111100010100110100010111111111011110111001110100111100011000111001011111011001
00100110101100100010101000100101001111011110111110010100011011001001111011111011000000010010011001
11001110001011001010101011001001111011110101100111010011001010011110101000001111011101111001100100
10100010100001101010010100001000101110011100100001100100111010111010110100001010011011110110010011
10001101000100101101100100101100110111101111100110011110100110111000111010100110101111000000010100
0001001001110011111001101011101011100101110111001110101101111000010100011101010000001101110101000
000011101011000101101101010110101000111001001000101000100101100100000000000110111011001100101111001
10001100001000111101011000001010011111100111101111011111010010110100100000111111010001000010101110
010011111111101101011000110010011111110010011111111001100000001110011001001011101000100011101000110
01000001110001010000110101000111011101100000011111100011011001001010000110001110101101010111100001
10000000000100000010010001010100010100000000000111000000110101011110100011010001001111100101011010
10010111001100110011110111101111001000000100011001001001101111110011101010011010100110001110110111
01010100101010110110100101101110011001010111001000110000111010111101110011000100101011001001111
1111010110111010101001101111100111010011111011000111100101011101111100011001010010010000101101001
100001010011101101110110000001100111010011010000100011110111110110001010100101110101010010010000011
01001000101001000011110010001011001101001000000100000011110000000111111100101111010110011111100
1111101110101111001111001010100110100011111110101011011000001110101111010000011101011010101010001
110001000100110110011101111000101000000000101010101100100100101101110011100000110010101010010000010
1100111110111010010001000111110110000101010000101101001011101111111110100100000001011011101010010
11010011011001110000010001001101011011100100011001000100001110101111100001100110101011101001000001
101001111111100100101011011010001011011100001111010011011000000000101000111101101011010100101001111
1001101110111000100011001011000100101010000100011110110011110100110001011001001110011110100000101
011111111010011111100000010001000110001011101001001011000001101000000010111111001010111101001111000
011111010101010010110010110010000010010010111011110000110000011010011001100110101111001111001111111
10110100000000100010101010001101010111100011111001111110111000110110111101011001010110111001110001
010000010101001111111011010001100100110101101010111111000001111111000101001111100010111100100110010
00001100101100011001000000011110000100100001111010101011000100100000101010010101111101001101101110
000010001111000100100000001110100000001100111010011100111101011000011111101001001111101101000111100
00011001001100111111111110001011000100110110100000001010011110101001000000000100101001011010100001000
11000101000110111011010010011101010110111010111001001001110011101111010010100011000100100010000000000
1110101101110110001010101011010011101001111001010111101111101111101001011001011110110010111101010010
0111001100100110101111001010101110110111101011000011111000000000110110000100101000011001101011000001
0111011111111011001011110000010111111011110011011100101011010100101000110000010010111110110111010001
01000110101001011010101000111010011110100001000100111000011010101011000000111101010011101110001 1
110000010000000001000011111010010010011000000111111011100100011100100100100011111110011100011100000
100010110111000000110001100100100001001110000000011011111101011011000001010110010000101100001110101
1101000001000111110110010010001110110010010100110011111111011000011110111000000101111011111111011100
001000100001100011000010101111001000000011100100011001110000101110111110101100101100011100111000011
000010101011101101000011011010101011001010001111001010010100010001111011011010010111100000010101101
10111110011110001100111011011010001011110111100001010100110110000001100100111100101010011101001000
01001001011100001101001011111101101001011111011011101101111001000011111010100111000100010110110101
110110100001101001011110111110001100000001100001011110011101110011110101100101100110001001101000101
1001000110011101111111011000100001001111011111110110100111101010110110100110011101111011111001011101
000001011111001110001100110011001110101101011110100110100001100010101011011010101101001010110100100
00100001000010101010010110100111111000100001000011100001100101010010010000010001011100000011111010
1000110101011001000100010010101010010101110101110011011001100000010111001100011000010111000101011
11010100001101100010011110110001110111010011101111010011001101100101101010101110101010011100011101
01011000110110010110001111000111100001010001111010111100111000110001011011111101110011101011101011
00011101010001111011101111111100010110001100000100011001011011000111100001010100111110100001111011011
01001100011111111110111101111000110010111110100101010011010111010011001011000111110011011110101100
01010110100011110010111010000110001110000001111001111110110101100111101010100000010001110000100111
11110000001001100011001000011011010101101100110110101110001101001011101010001011010011000010000011001
1101001011010011100100001111110001010110000000000000.....

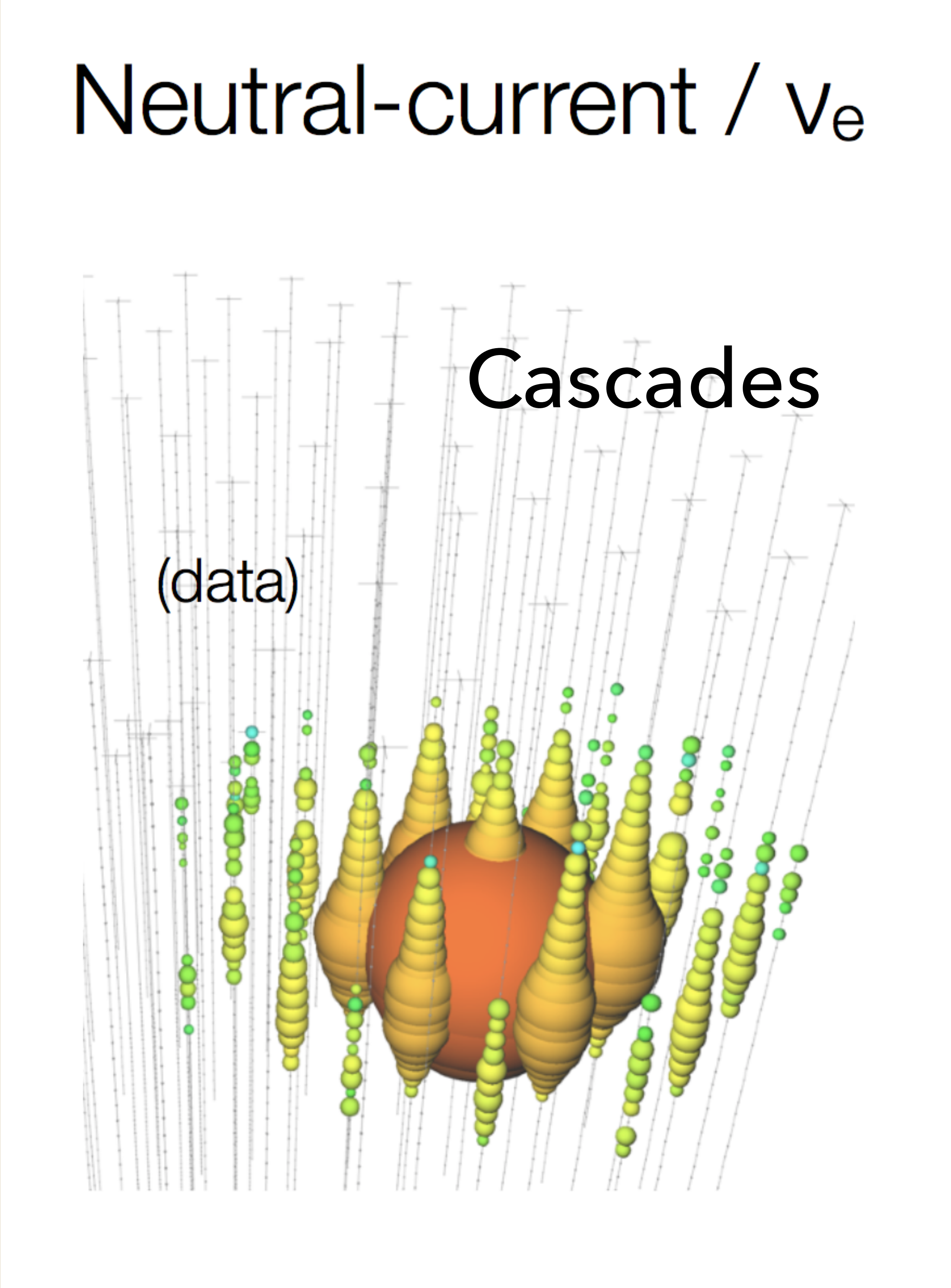
QRAC: What has been done so far

- Have achieved compression with 8 qubits
- Almost demonstrated storage/retrieval of IceCube simulation data with 14 qubits

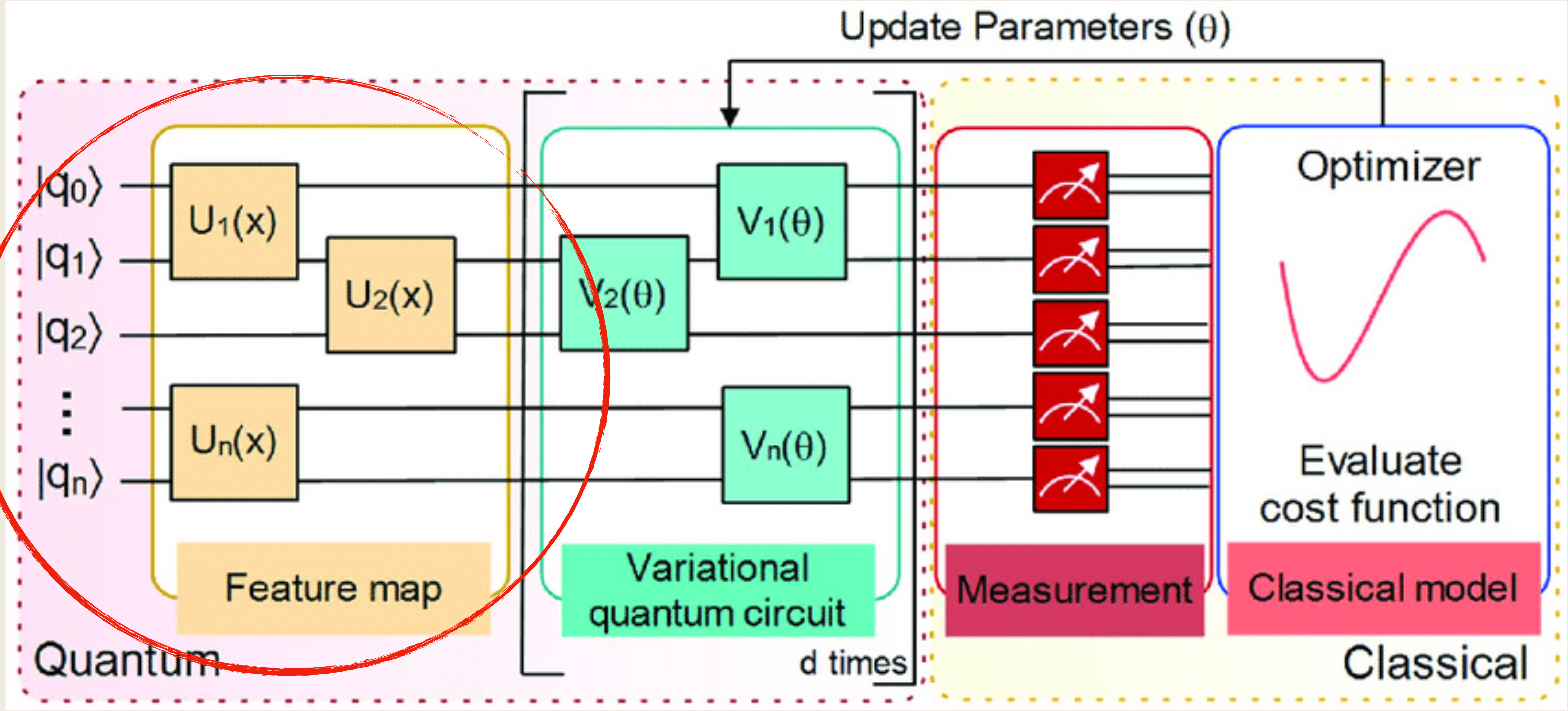
Plan with IceCube data

- Take set of IceCube data: tracks/cascades ($\nu_\mu CC/\nu_e CC$)
- Input into our QRAC
- Take states from QRAC, input them to train Variational Quantum Circuit
- First step, will be to investigate if VQC can classify tracks/cascades

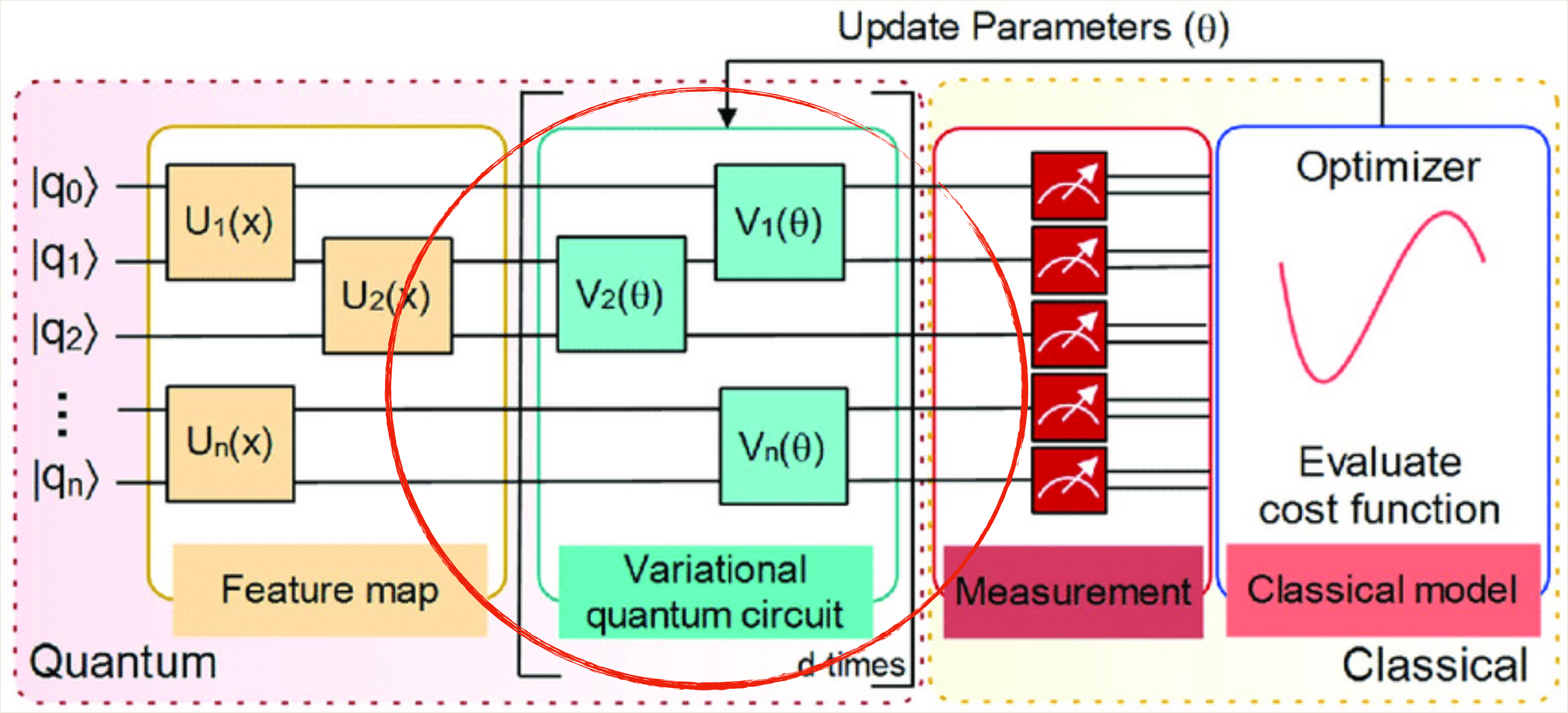




Variational Quantum Circuit

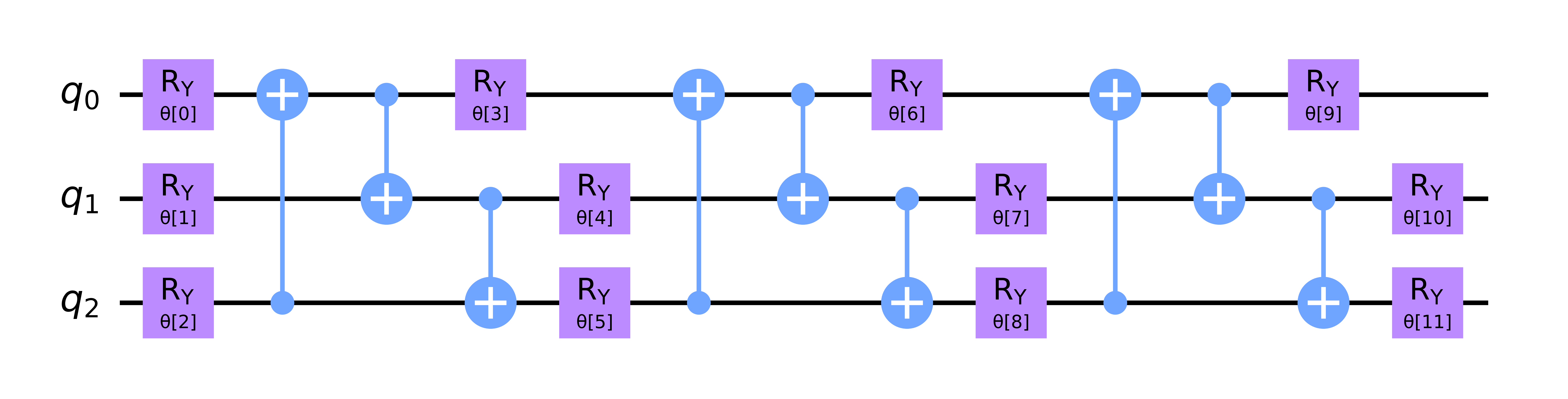


Variational Quantum Circuit



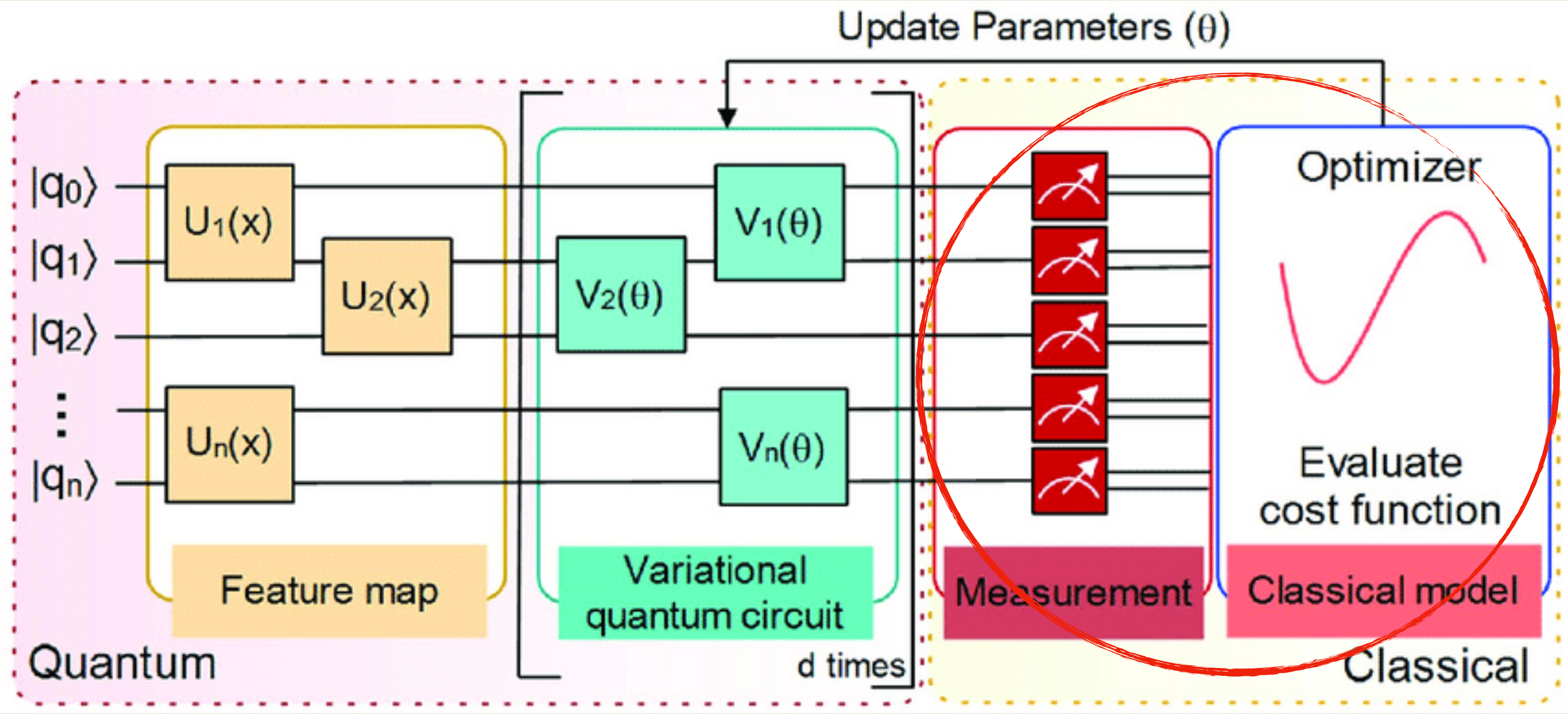
Model Circuit

- Is the variational part, where machine learning happens, called the "ansatz"
- The ansatz is parameterized by a set of free parameters θ that will be updated during training
- The structure of the ansatz, entanglement, type of rotations, number of parameters, number of gates, are all tunable



A 3 qubit example of a "RealAmplitudes" ansatz from qiskit.

Variational Quantum Circuit



Loss Function

- Default: is cross-entropy loss

$$\text{CrossEntropyLoss}(predict, target) = -\sum_{i=0}^{N_{\text{classes}}} target_i * log(predict_i).$$

  - The difference between the ideal distribution (the true labels/target) and the measured distribution
- Calculated across a batch of samples, and the average is taken across the batch to obtain the final loss value for that iteration
- Others like MSE are also used
- Inherently a classical operation

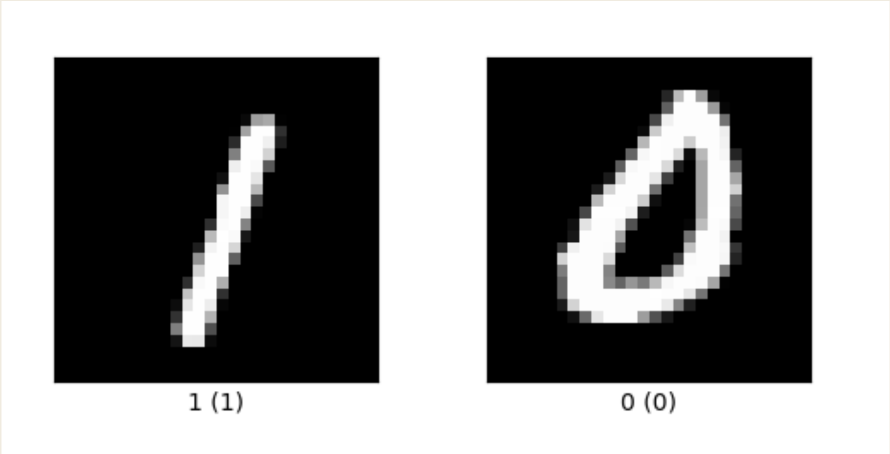
Optimization

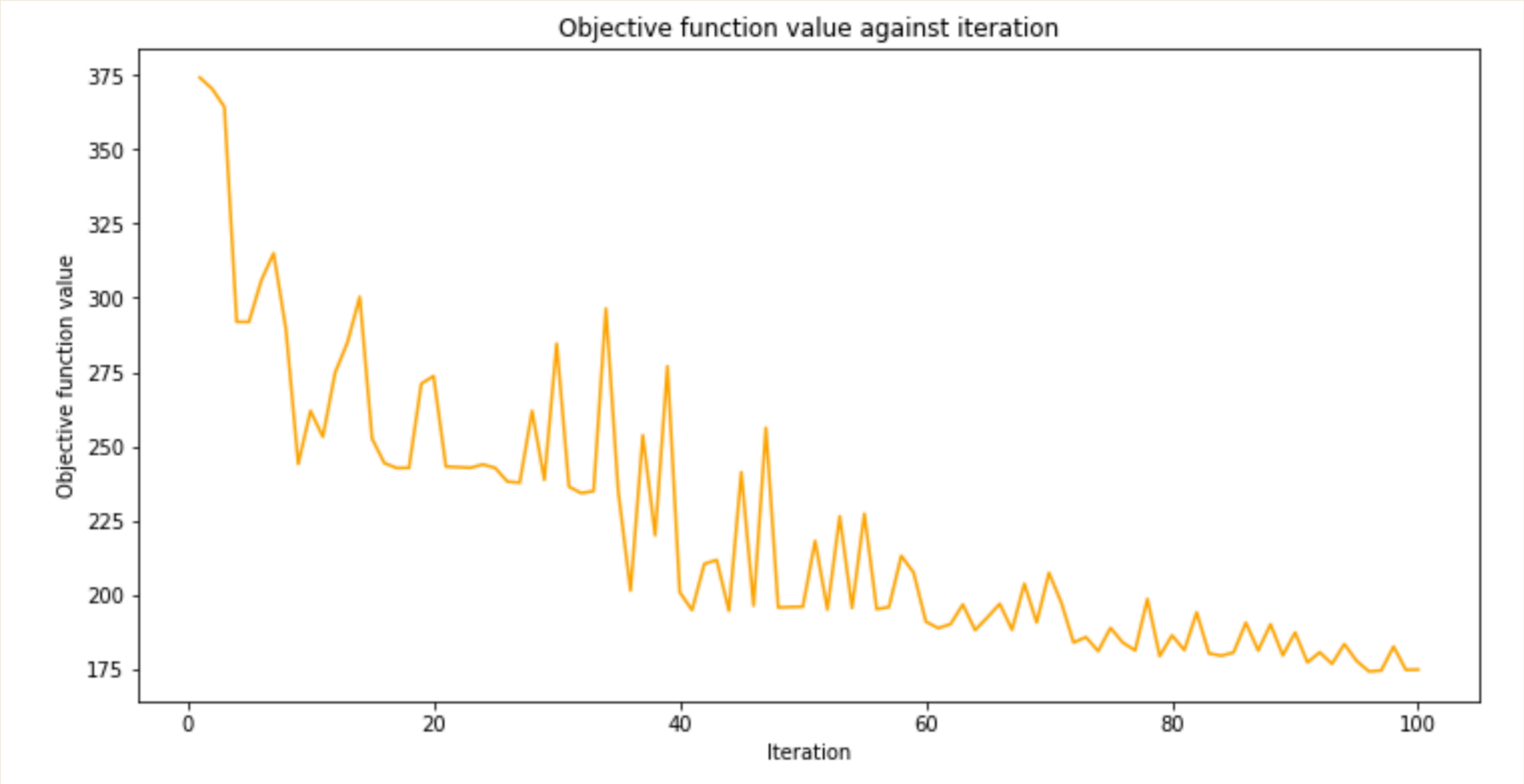
- Optimization performed using classical algorithms: COBYLA, ADAM, SPSA, etc.
  - Numerical Differentiation has been the norm for a while using parameter-shift differentiation
  - It's unclear how intermediate derivatives could be stored/reused inside a quantum computation
- Investigating if we can use quantum optimization techniques like quantum annealing
- A lot is still unknown about what is optimal, more of an art to test
- "There is often little theoretical motivation for any given optimizer, although recent work has been to analyze the training dynamics of QVCs (VQCs)"

arXiv:2202.01389v3

Proof of Concept using Qiskit

- MNIST: image classification 0s and 1s
- Input image via amplitude encoding into 8 qubit VQC
- ~about 100 tunable parameters
- Iterations are a hyper parameter
- COBYLA, cross-entropy loss function
- Training score: 97%
- Testing score: 98%





What's next?

- Comparing VQC structure with amplitude encoding of IceCube simulation data with QRAC encoding
- Takes ~16 qubits to encode largest IceCube events via amplitudes
- Need larger quantum computers than publicly available (applied for ORNL grant)
- Now have access to ~100 qubit QCs
- Choose optimal qubits in target QCs, implement custom feature maps, testing ansatz types, optimizations, loss functions etc.
- Transition away from Qiskit to PennyLane (a QML framework)

ICECUBE SOUTH POLE NEUTRINO OBSERVATORY

Qiskit

PENNYLANE

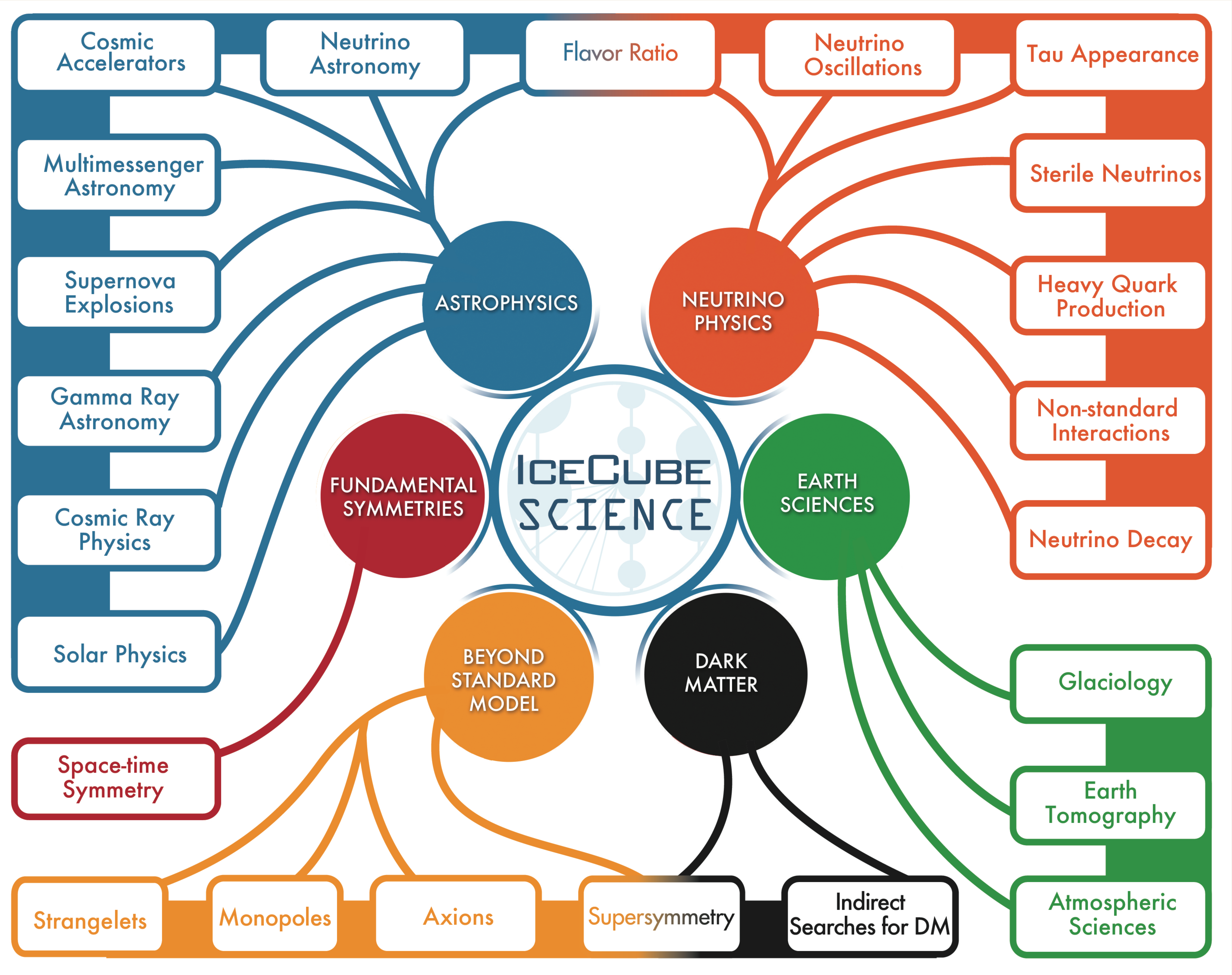
Conclusion/Takeaway

- ML in neutrino physics has been very successful the past two decades and will continue to contribute
- Yet with concerns about the streetlight effect and increasing data loads in next generation experiments/observatories, we might need new tools
- QC can take advantage of additional degrees of freedom to enhance data compression
- QCML is still a burgeoning field; much is still unknown about optimizations, actual quantum speedups, feasibility on current NISQ computers, etc.
- ***"The question on whether quantum computers can really play a role in identifying practical ML applications is still wide open, and it is unlikely to be decided by theoretical proofs…"***

Based on the Perspective Manuscript by M. Schuld and N. Killoran, PRX Quantum 3, 030101 (2022)

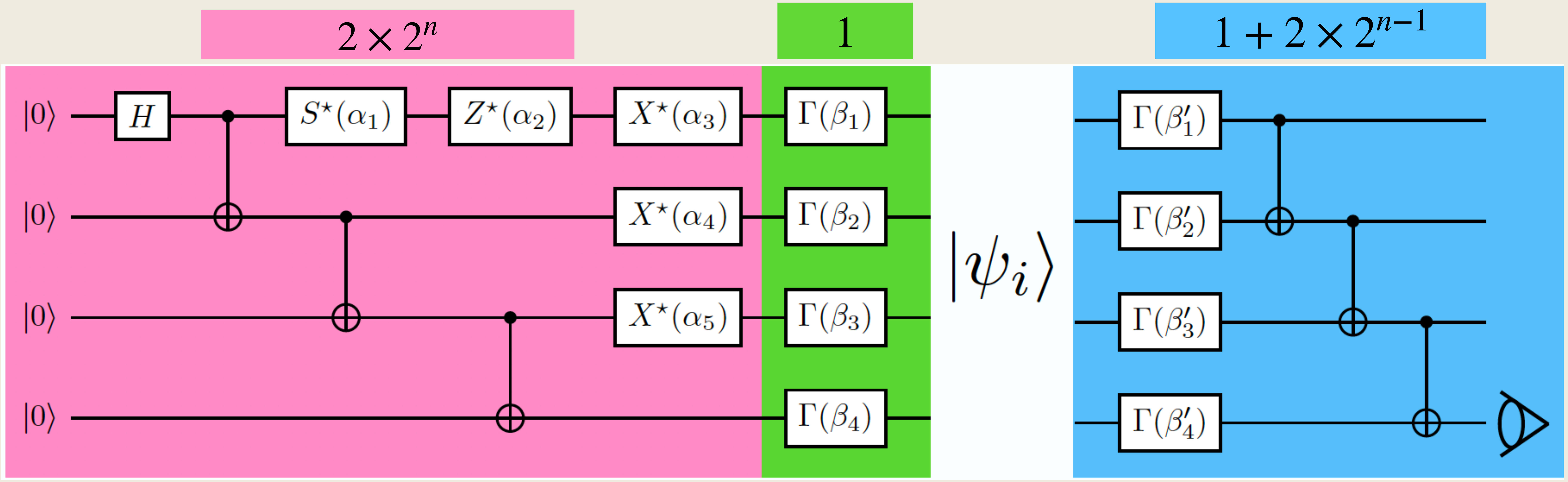
Gracias! Thank you! Preguntas? Questions?

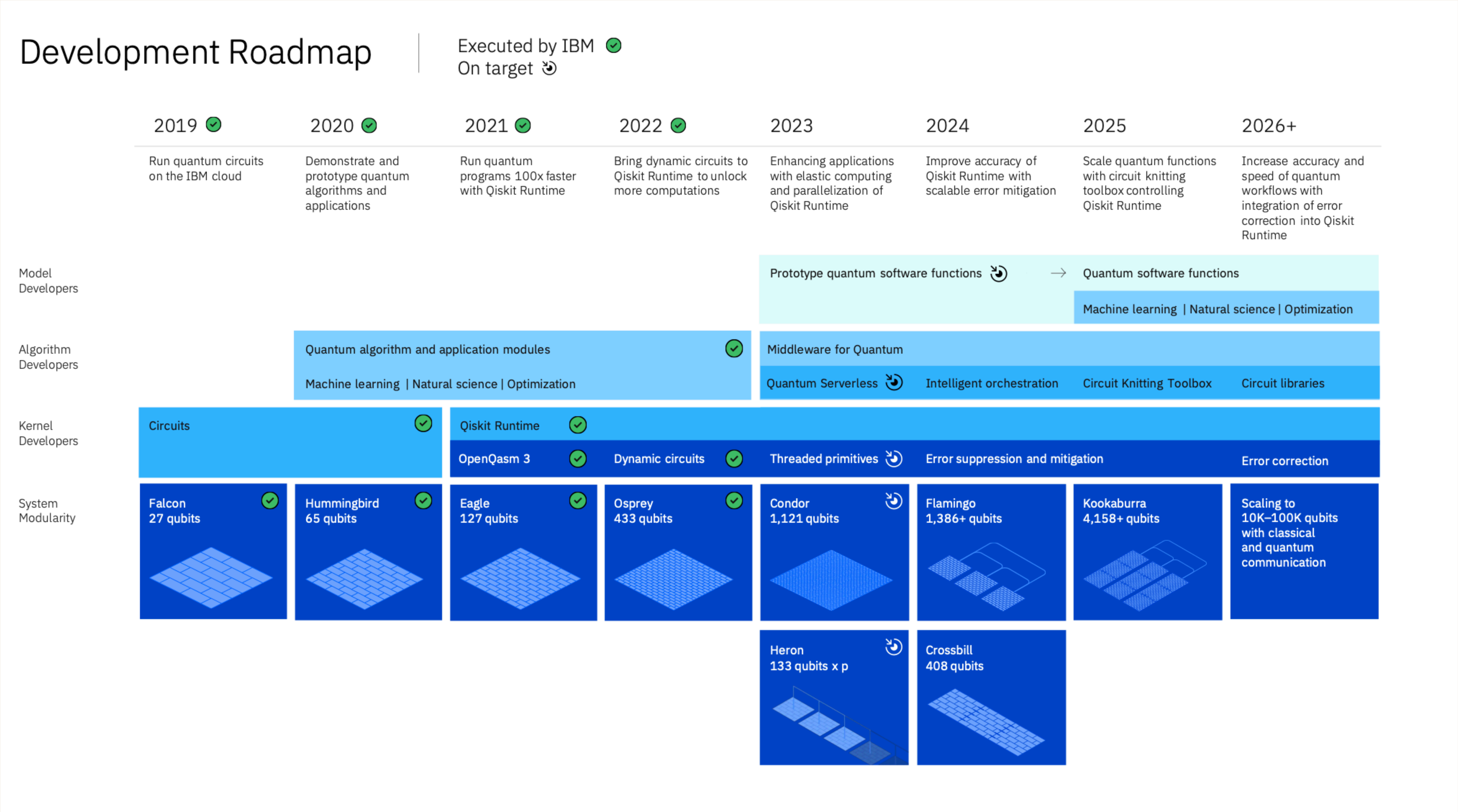
New arm of hectopus:
Quantum Computing!



Measuring Context Eigenstates

- The values for one context are all related. Only need to calculate one “fingerprint” and figure out mapping
- Scales like 4^N
- For 12 qubits this can be done in a few hours and stored in 2 MB




Development Roadmap
Executed by IBM
On target
2019
2020
2021
2022
2023
2024
2025
2026+
Run quantum circuits on the IBM cloud
Demonstrate and prototype quantum algorithms and applications
Run quantum programs 100x faster with Qiskit Runtime
Bring dynamic circuits to Qiskit Runtime to unlock more computations
Enhancing applications with elastic computing and parallelization of Qiskit Runtime
Improve accuracy of Qiskit Runtime with scalable error mitigation
Scale quantum functions with circuit knitting toolbox controlling Qiskit Runtime
Increase accuracy and speed of quantum workflows with integration of error correction into Qiskit Runtime
Model Developers
Prototype quantum software functions
Quantum software functions
Machine learning | Natural science | Optimization
Algorithm Developers
Quantum algorithm and application modules
Machine learning | Natural science | Optimization
Middleware for Quantum
Quantum Serverless
Intelligent orchestration
Circuit Knitting Toolbox
Circuit libraries
Kernel Developers
Circuits
Qiskit Runtime
OpenQasm 3
Dynamic circuits
Threaded primitives
Error suppression and mitigation
Error correction
System Modularity
Falcon 27 qubits
Hummingbird 65 qubits
Eagle 127 qubits
Osprey 433 qubits
Condor 1,121 qubits
Flamingo 1,386+ qubits
Kookaburra 4,158+ qubits
Scaling to 10K–100K qubits with classical and quantum communication
Heron 133 qubits x p
Crossbill 408 qubits

Overview

- Introduction to Quantum Computing
- Application of QC to HEP
- QCML in HEP
- QCML for neutrino astronomy
- Variational Quantum Circuits
- Conclusion