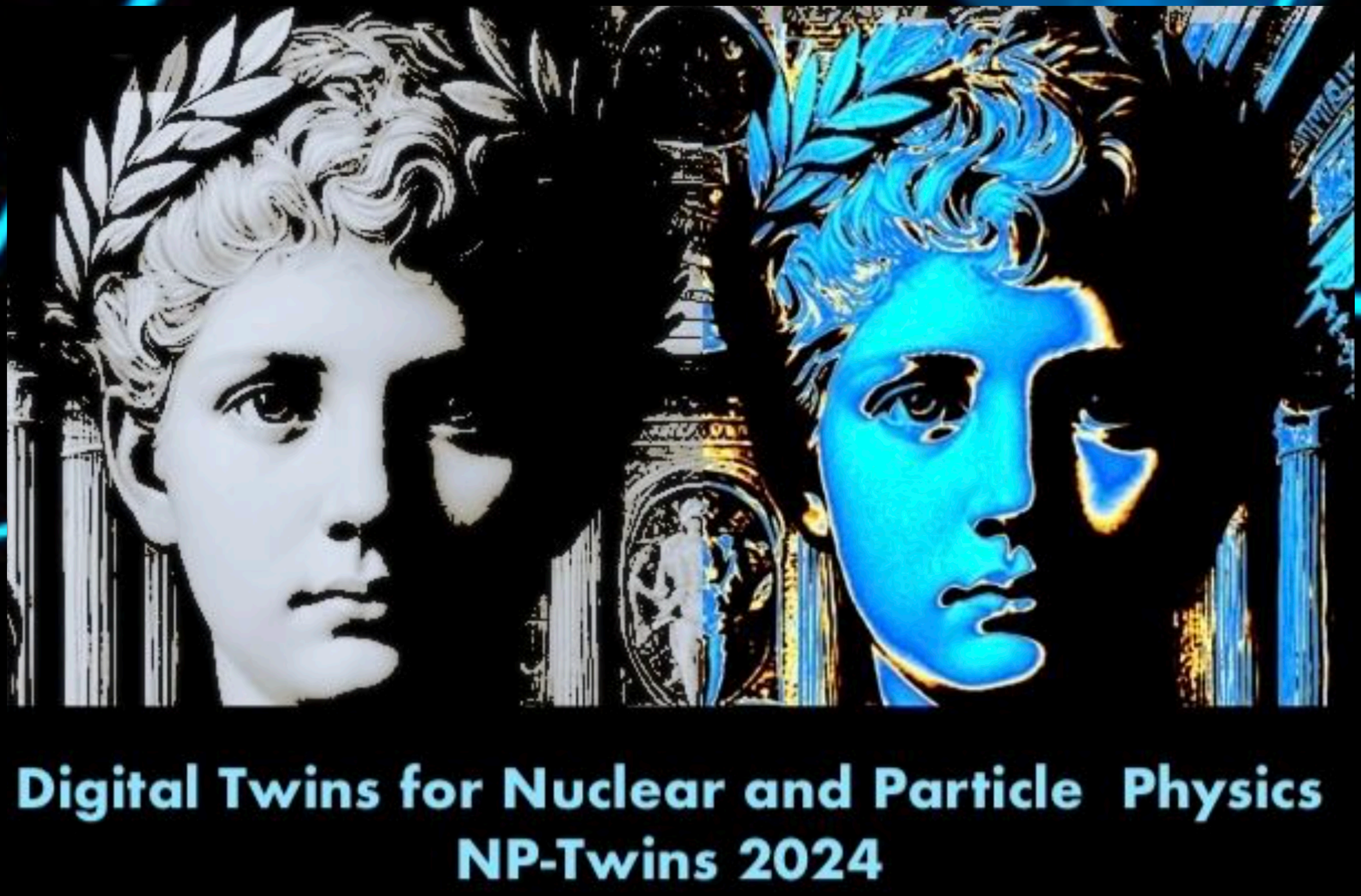
Quantum computing applications for HEP

Davide Zuliani
University and INFN Padova



Introduction

- The goal of this talk is to give an overview of **Quantum Computing (QC)** applications to High Energy Physics

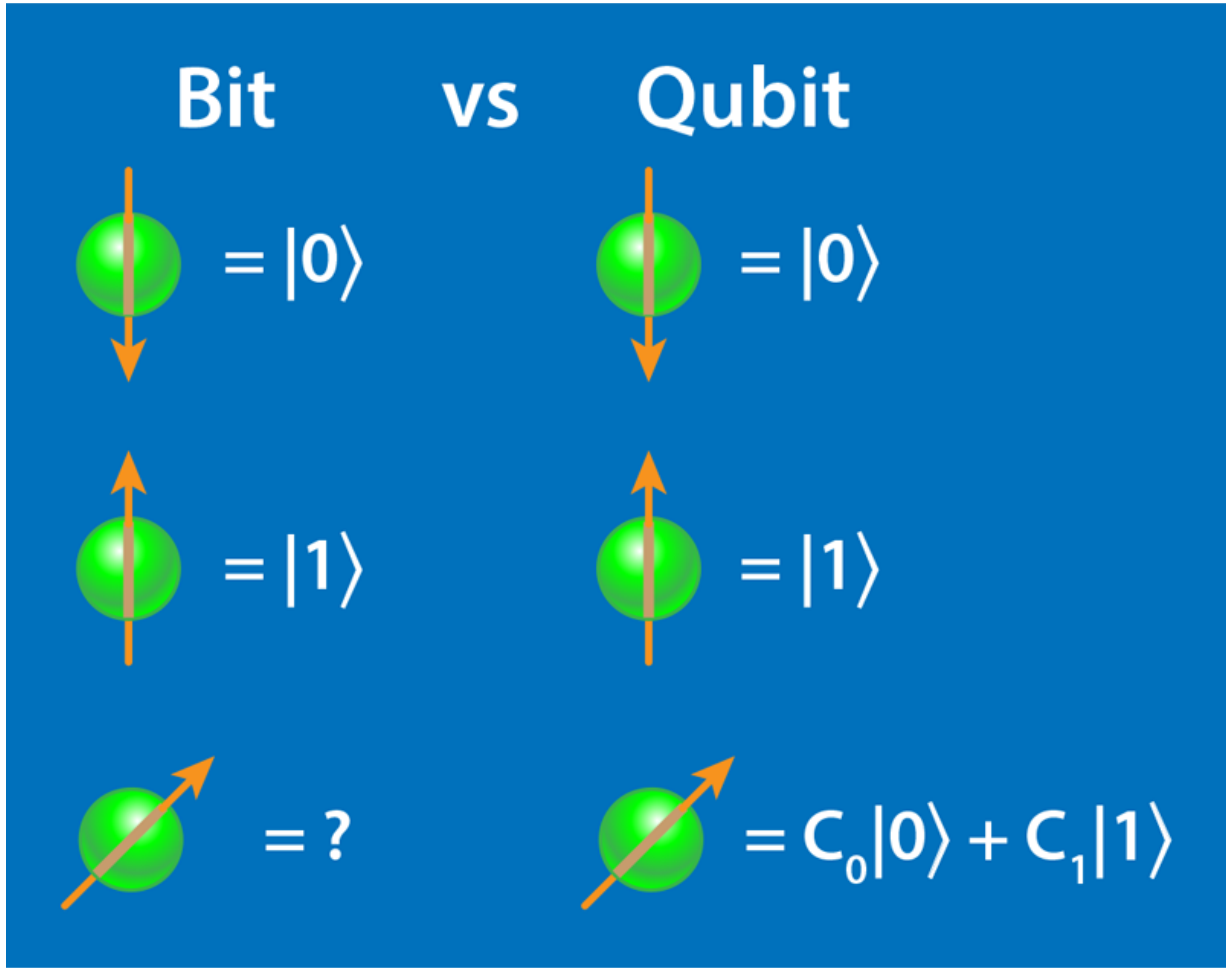
- Since I am a **user** from the **experimental side** (I work in the LHCb Collaboration), the examples I am going to show are definitely biased by my personal view (apologies for this)

- QC in HEP is now in an exploration and study phase, **you won't see any quantum supremacy in this talk**, just the state-of-the-art and prospects

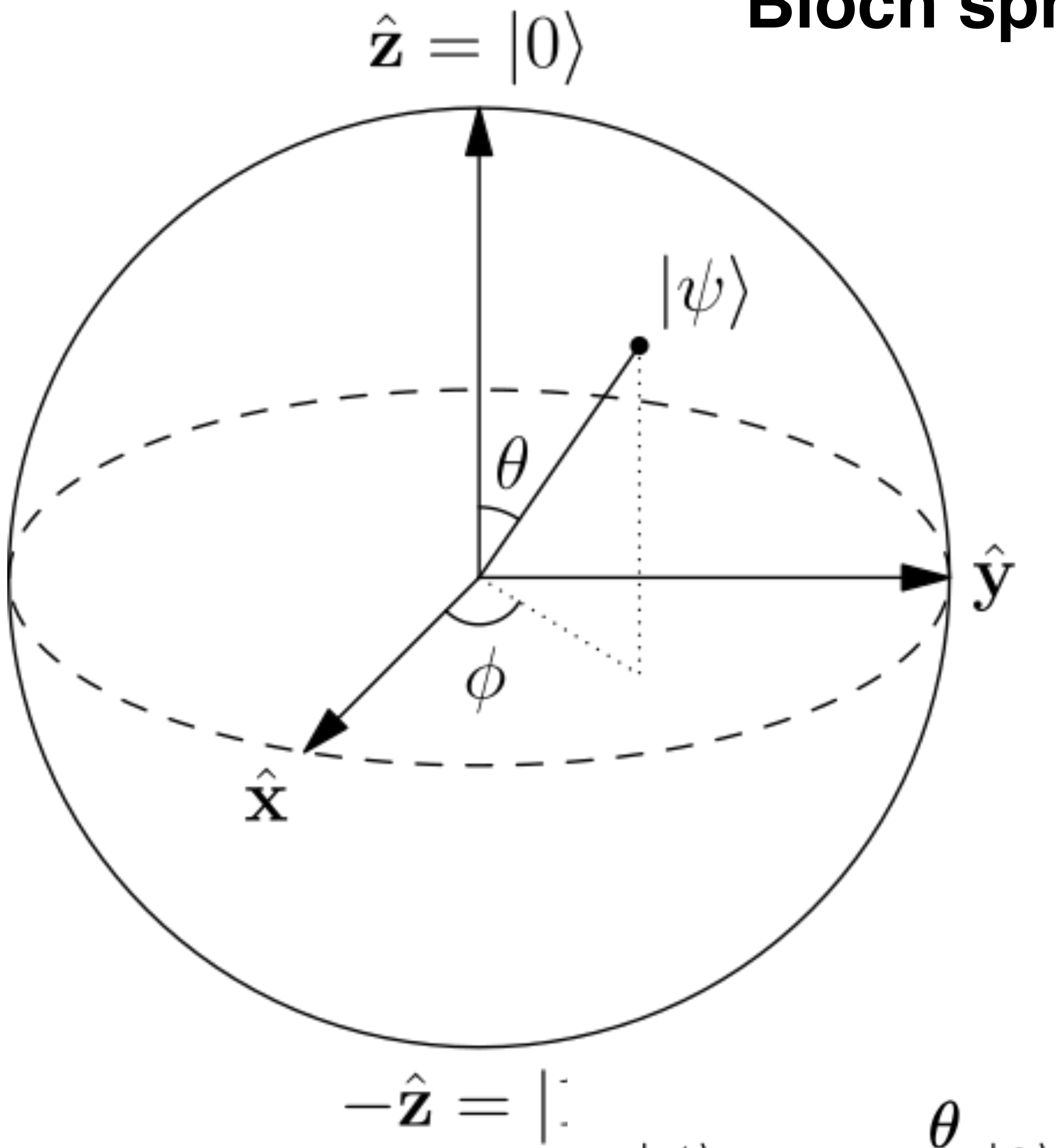
- In particular, in this presentation I will focus mostly on **Quantum Machine Learning (QML)** applications

- I will briefly introduce some **basics of QC**

Quantum computing: qubits



Bloch sphere



$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

1 Bit: two possible values, 0 o 1

1 Qubit: infinite values, one for each point in a sphere → **But when we read it we always find 0 or 1!**

Quantum computing: gates

- Evolution of isolated quantum states described by Hamiltonians

- Operations on qubits are unitary matrices

- **The operations are reversible**

- Some classical gates (like OR/AND) cannot be implemented directly

$$H(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

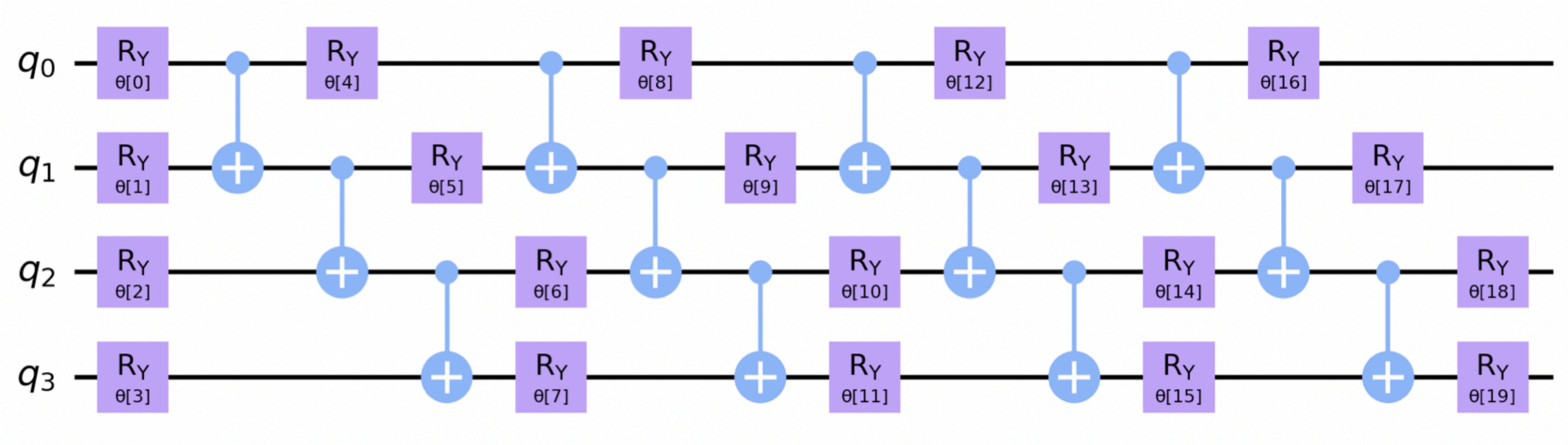
$$UU^\dagger = U^\dagger U = I$$

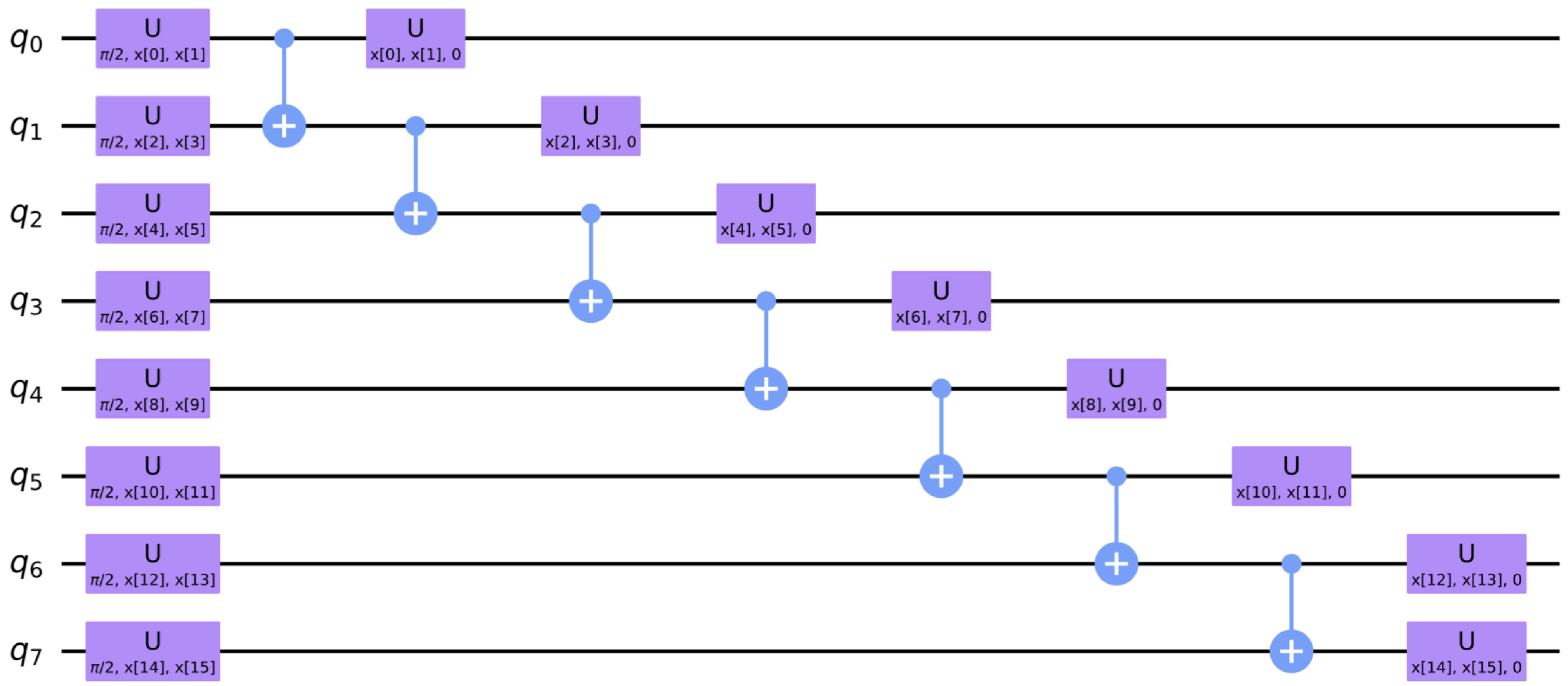
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix}$$

Operator	Gate(s)	Matrix
Pauli-X (X)	X, ⊕	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	Z	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Quantum circuits





- Circuits are composed by a sequence of operations on qubits
- Quantum software is programmed by building these circuits
- When they are ported to the quantum hardware they can look very different from the initial design (**transpiling**)

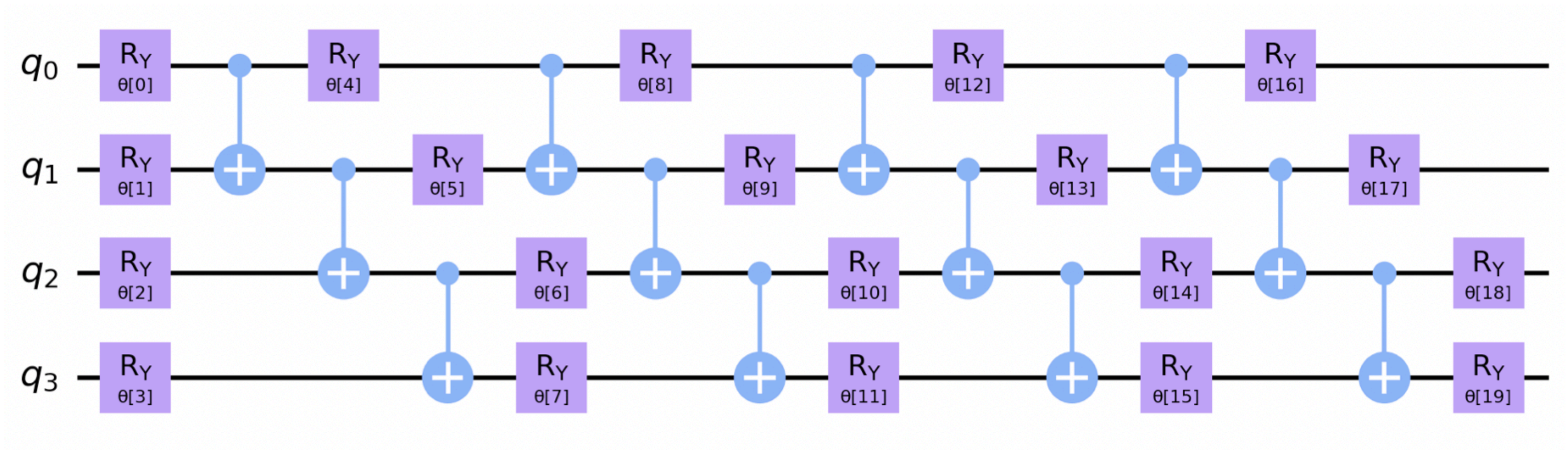
Popular python libraries for implementing Quantum Circuits are Pennylane/Qiskit

In particular **Qiskit** is used for tests on IBM hardwares

PENNYLANE Qiskit

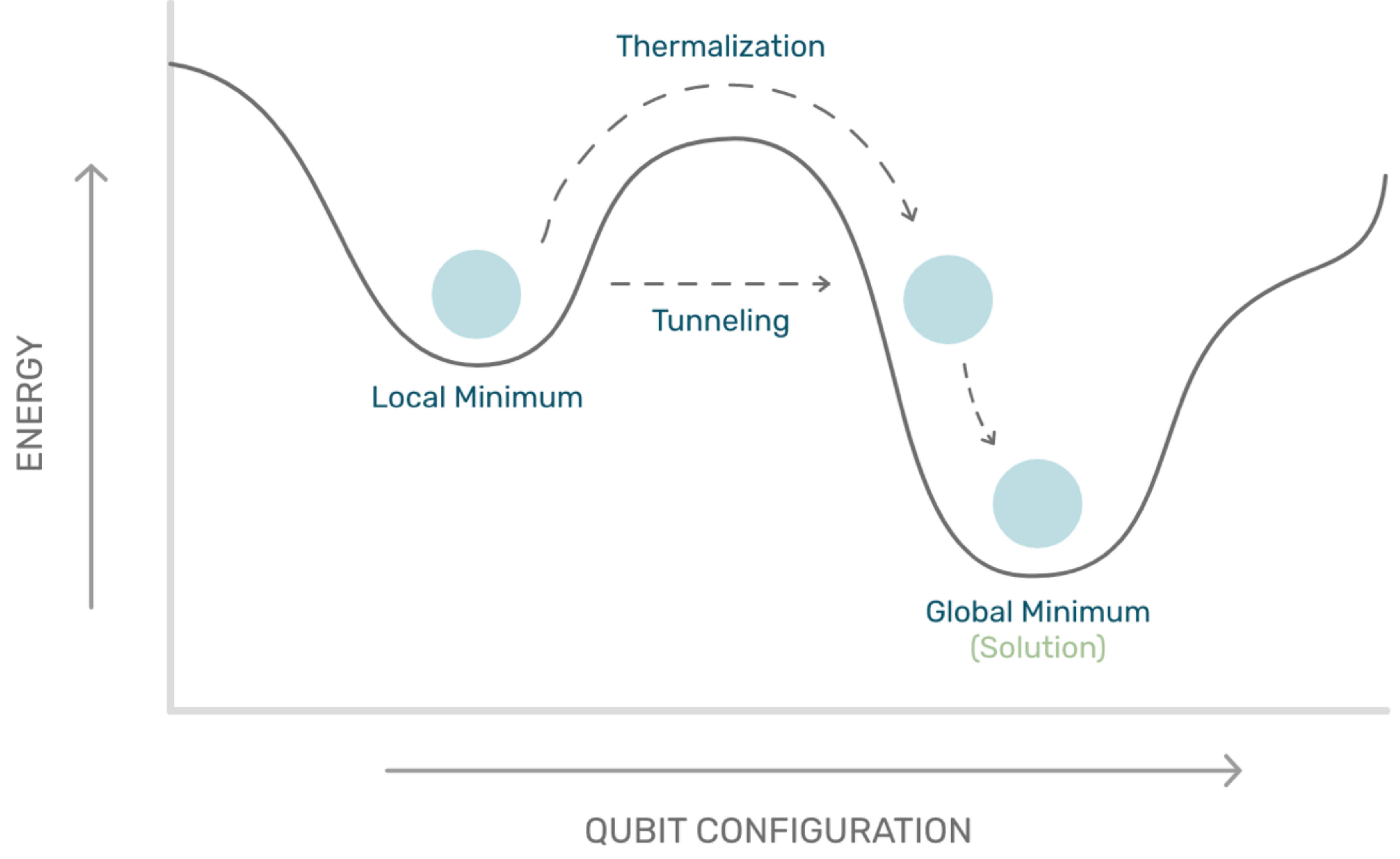
Gate-based vs quantum annealing

Gate based quantum computers



All kind of tasks

Quantum annealers



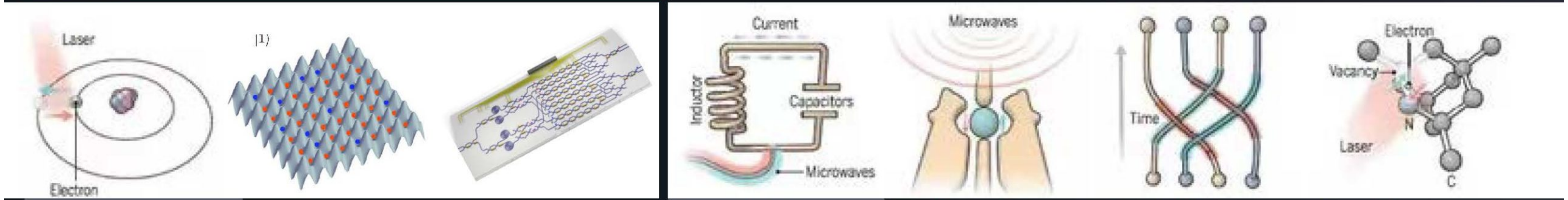
https://www.vesselproject.io/life-through-quantum-annealing

Dedicated to optimization problems

D:WAVE
The Quantum Computing Company™

Quantum computer technologies

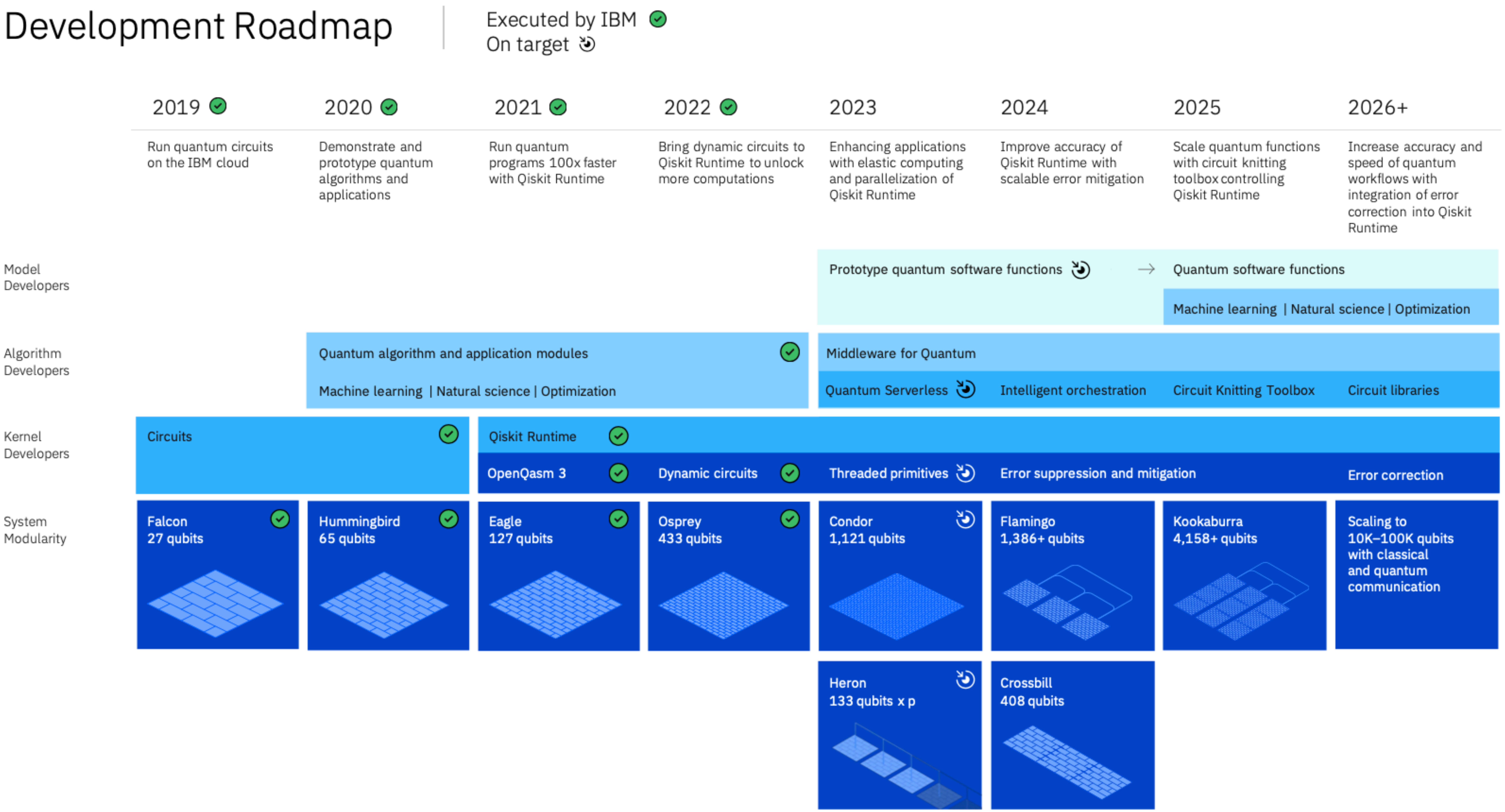
Quantum Computer Technologies



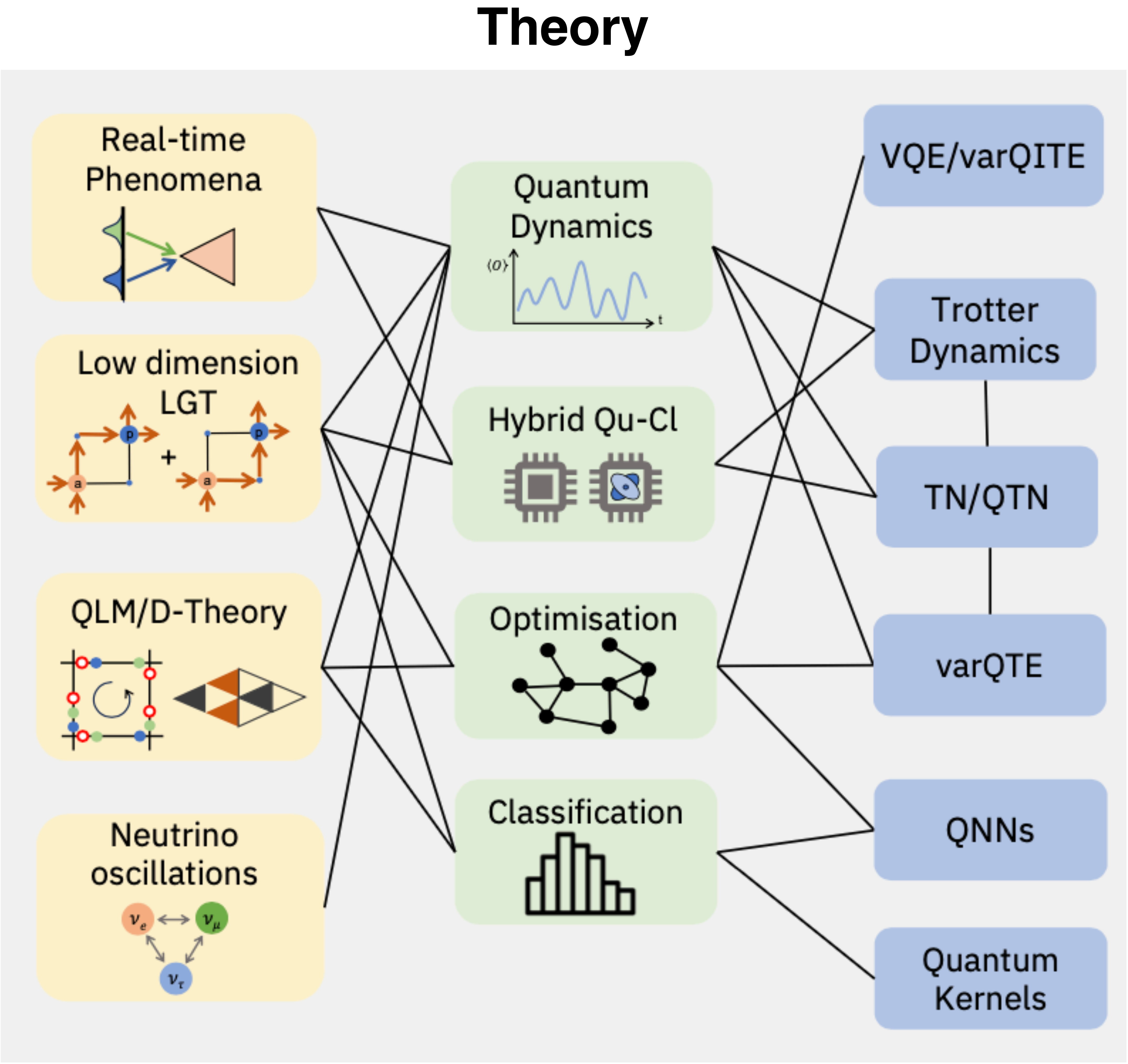
	Natural Qubits			Synthetic Qubits			
	Trapped Ions	Neutral Atoms	Photonics	Superconducting Loops	Silicon Quantum Dots	Topological Qubits	Diamond Vacancies
	Electrically charged atoms, or ions, are held in place with electric fields. Qubits are stored in electronic states. Ions are pushed with laser beams to allow the qubits to interact.	Neutral atoms, like ions, store qubits within electronic states. Laser activates the electrons to create interaction between qubits.	Photonic qubits (light particles) are sent through a maze of optical channels on a chip to interact. At the end of the maze, the distribution of photons is measured as an output.	A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.	These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.	Quasiparticles can be seen in the behavior of electrons channeled through semi-conductor structures. Their braided paths can encode quantum information.	A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.
Qubit Coherence Time (sec)	>1000	1	--	0.00005	0.03	N/A	10
Fidelity	99.9%	97%	--	99.4%	~99%	N/A	99.2%
Qubits Connected	High	Very high; low individual control	--	High	Very Low	N/A	Low
Company Support	IONQ, AQT, Honeywell, Oxford Ionics	Atom Computing, ColdQuanta, QuEra	Psiquantum, Xanadu	Google, IBM, QCI, Rigetti	HRL, Intel, SQC	Microsoft	Quantum Diamond Technologies
Pros	Very stable. Highest achieved gate fidelities.	Many qubits, 2D and maybe 3D.	Linear optical gates, integrated on-chip.	Can lay out physical circuits on chip.	Borrows from existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
Cons	Slow operation. Many lasers are needed.	Hard to program and control individual qubits; prone to noise.	Each program requires its own chip with unique optical channels. No memory.	Must be cooled to near absolute zero. High variability in fabrication. Lots of noise.	Only a few connected. Must be cooled to near absolute zero. High variability in fabrication.	Existence not yet confirmed.	Difficult to create high numbers of qubits, limiting compute capacity.

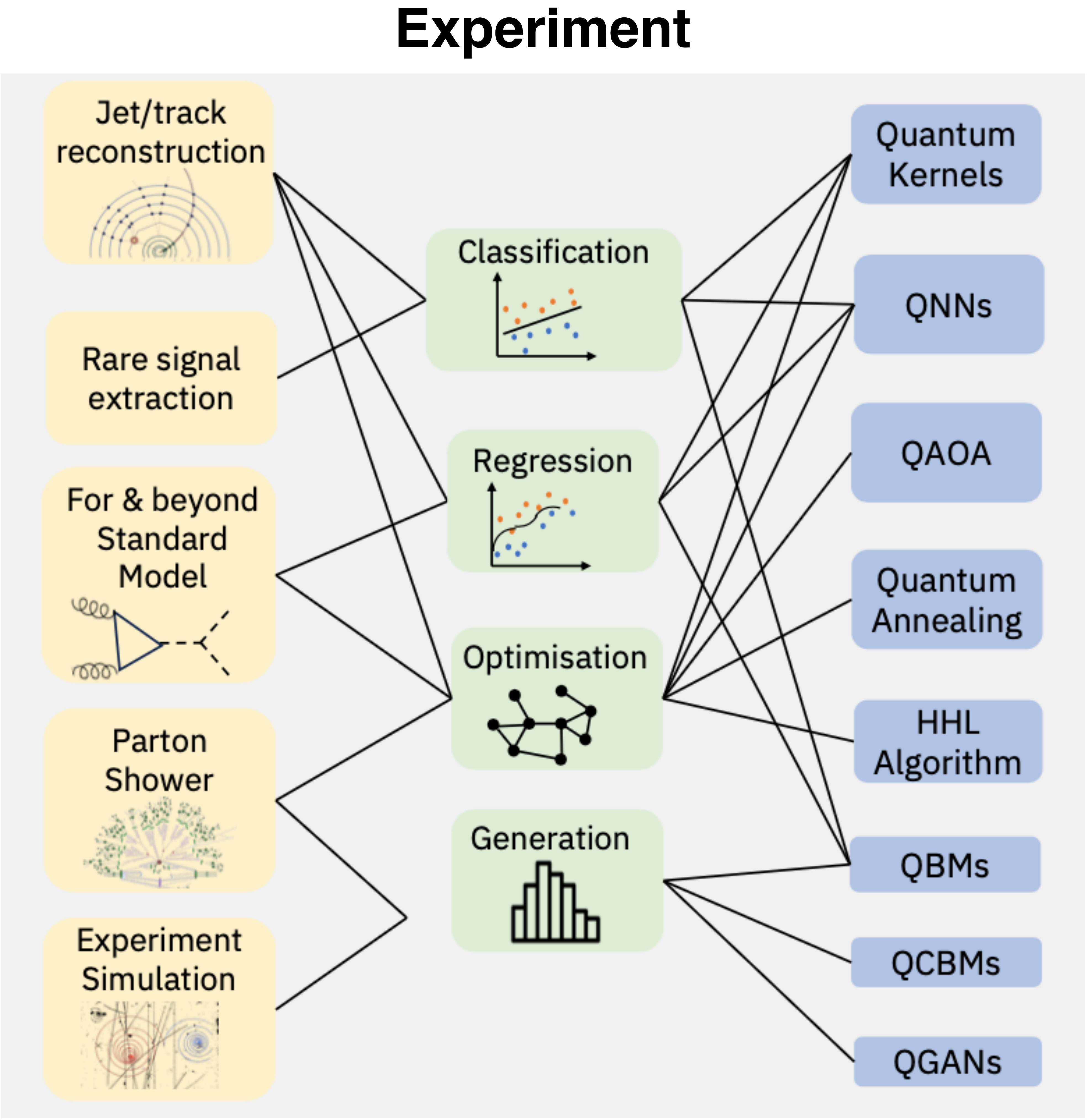
Source: Science, Dec. 2016

Quantum computers



Quantum computing in HEP





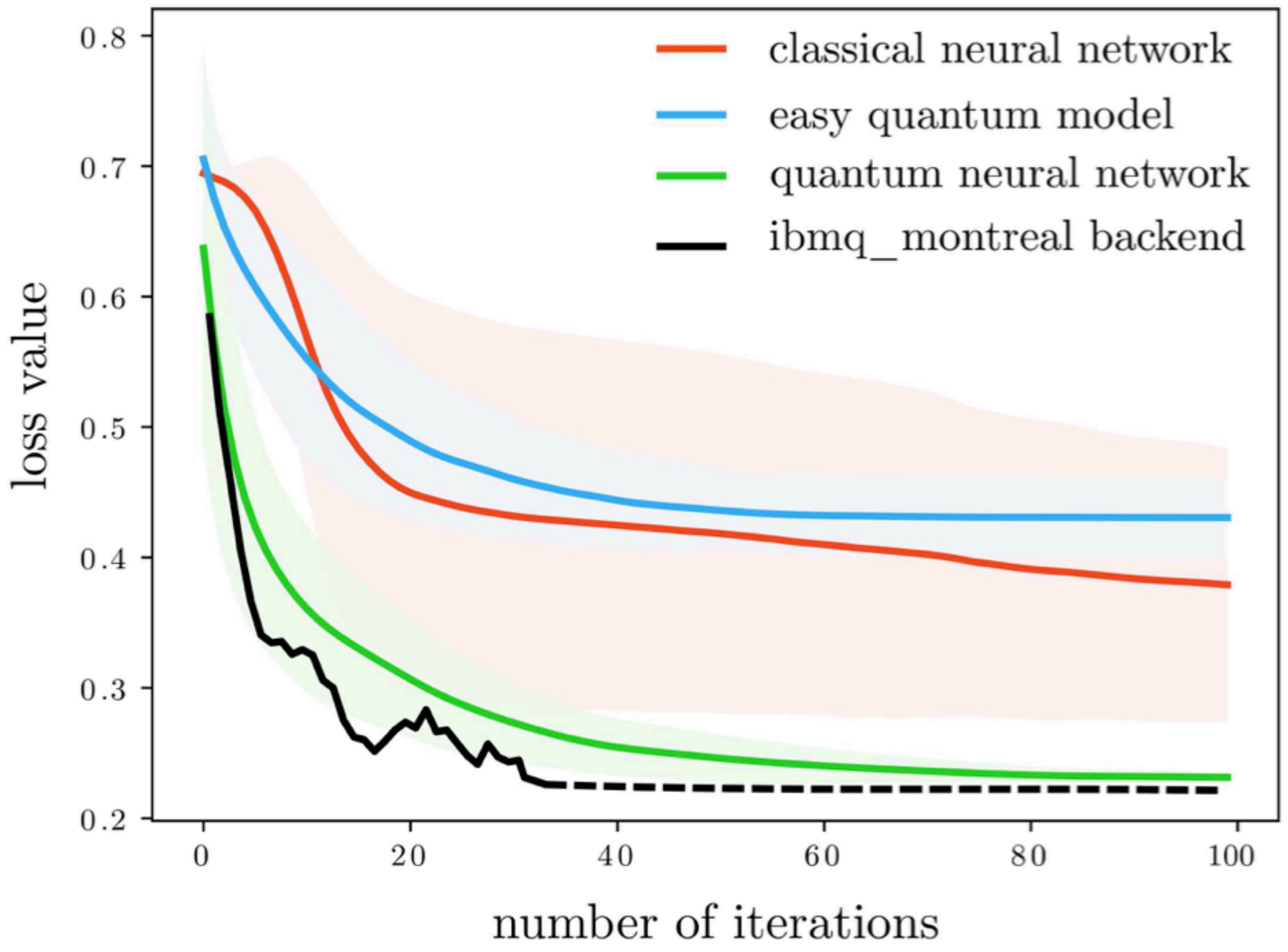
QC4HEP: https://arxiv.org/abs/2307.03236

Quantum machine learning (QML)

What could be the possible advantage of QML?

- **Runtime speedup**, both in training and inference
- **Representational power**: exponential advantage of Hilbert space
- **Explainability**: open the black box by measuring entanglement correlations
- Catch **unknown** (quantum?) **correlations** of our data

Nature Computational Science volume **1**, pages 403–409 (2021)



QML: a possible working flow

Data preparation → Data embedding: map data from classical to qubits

Model definition → Circuit (or Hamiltonian) definition

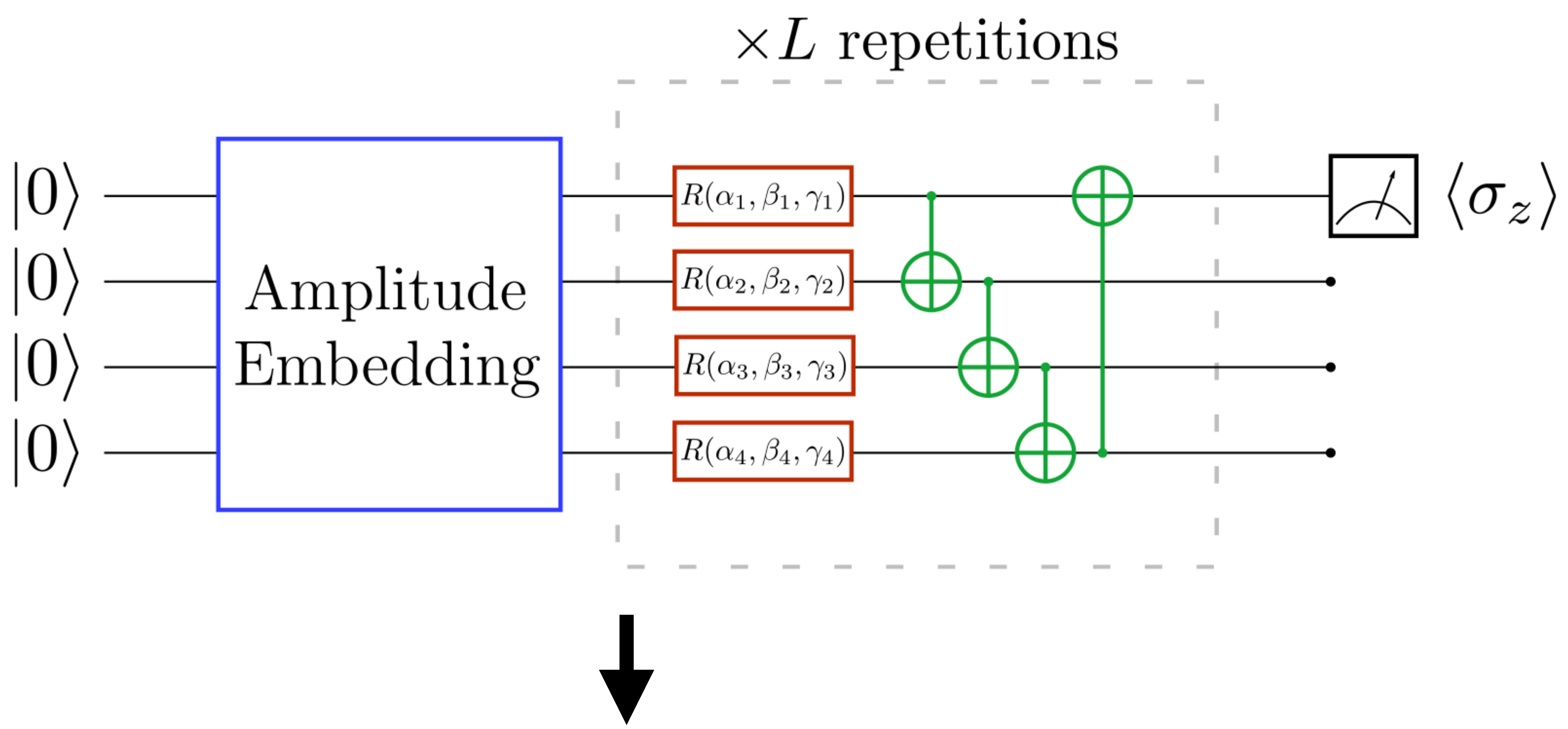
Training

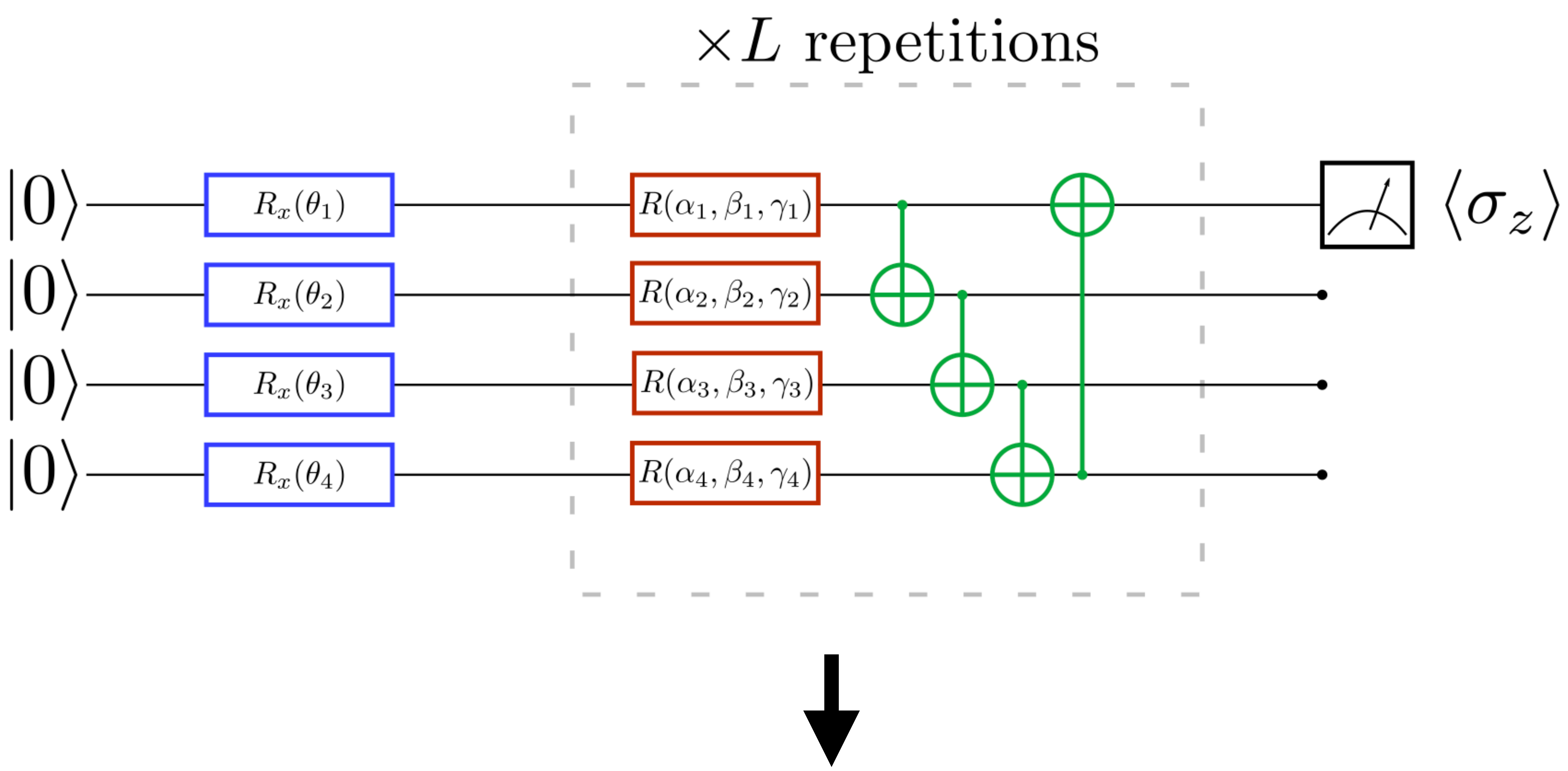
Testing

Training, Testing → **Readout**: measure the qubit state → The required output is usually the probability of measuring 0 (or 1) → Several measurements (**shots**) are necessary

Interpretation → Entanglement correlations, entropy

QML: data embedding (example)





Amplitude encoder: **2^n** features in **n** qubits

$$|x\rangle = \sum_{i=1}^{2^n} x_i \, |n_i\rangle$$

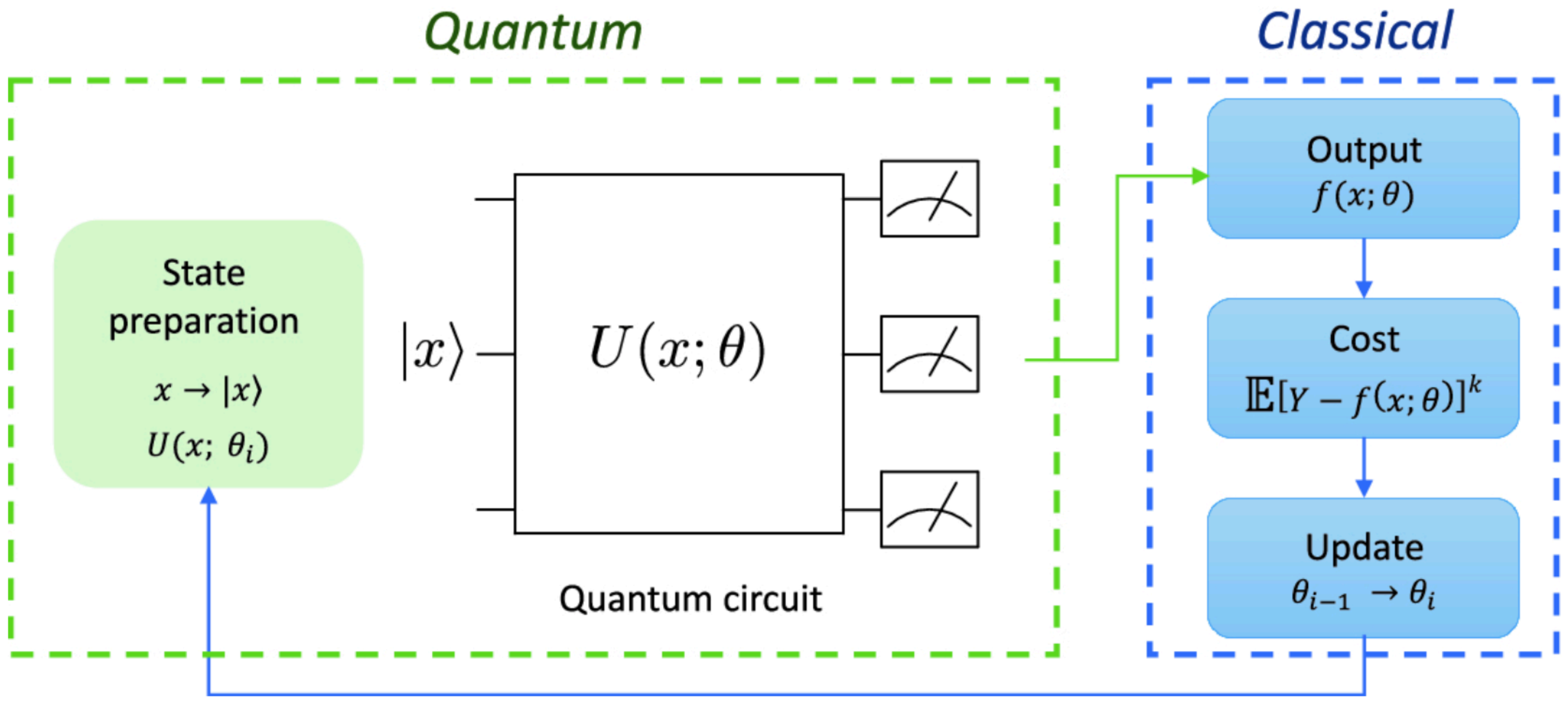
exponential compression

Angle embedding: one rotational gate per feature (#features=#qubits)

Polynomial compression

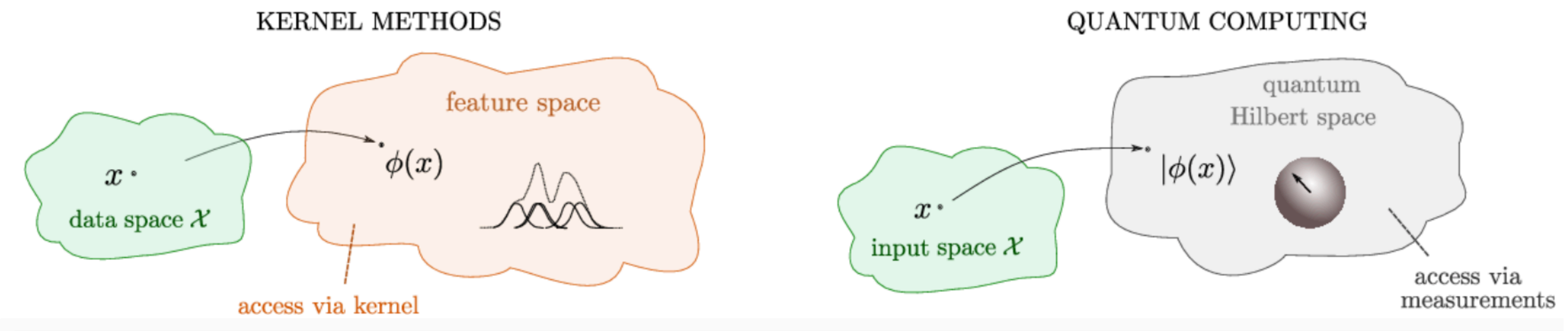
QML: models

Variational Quantum Circuit



Example: Quantum Neural Networks

Kernel methods

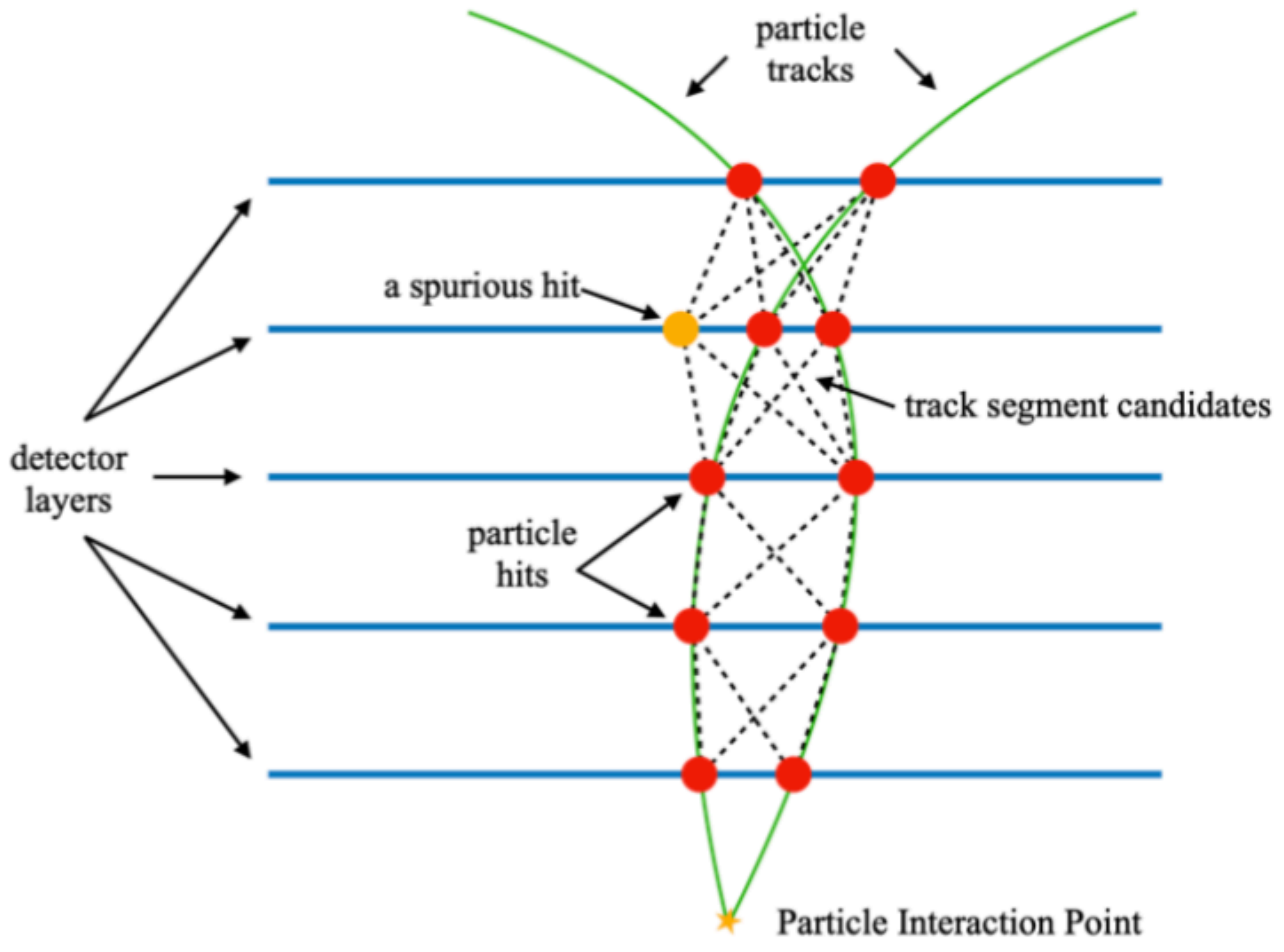


M. Schuld

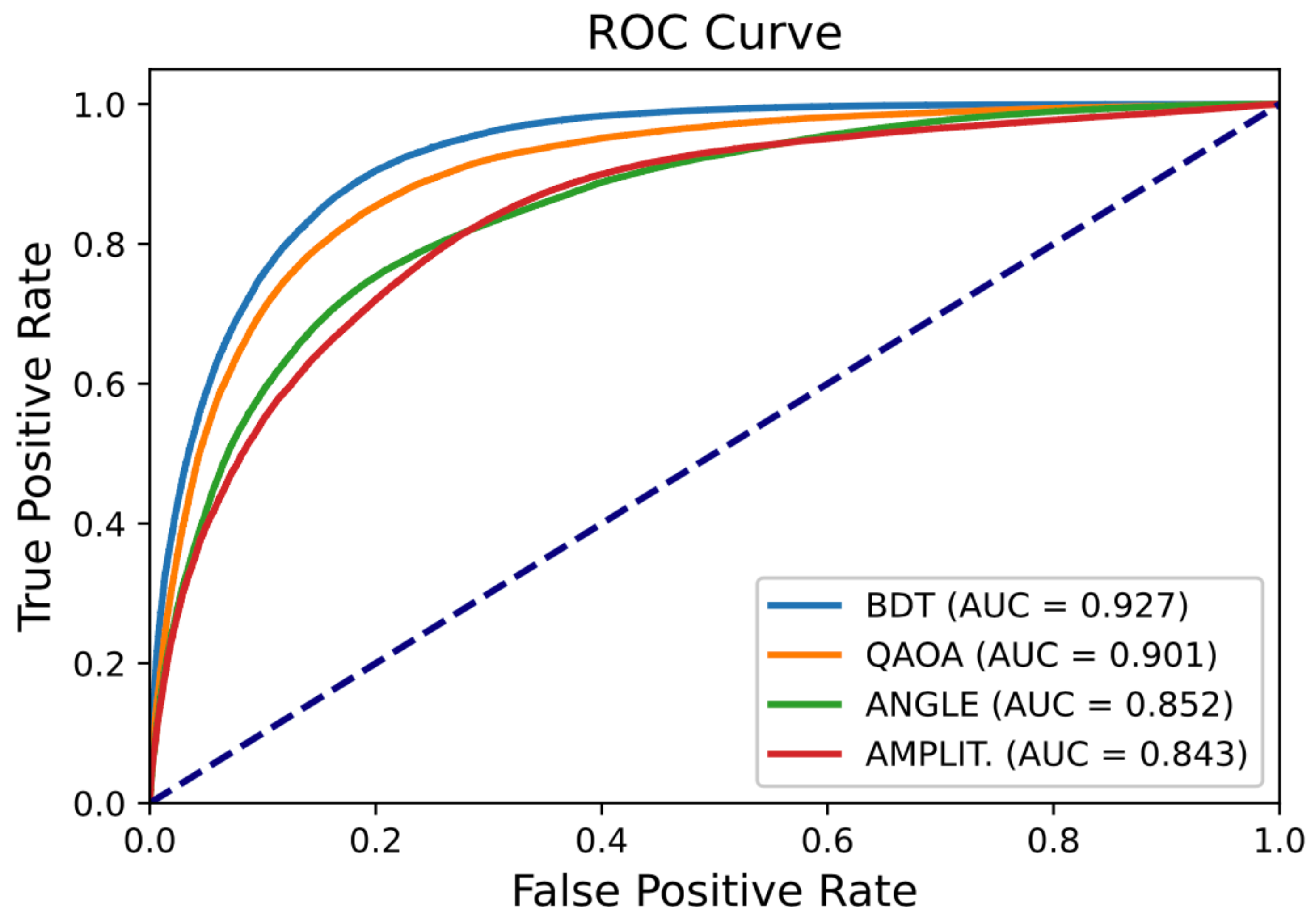
Example: Quantum Support Vector Machines

QML: examples in HEP

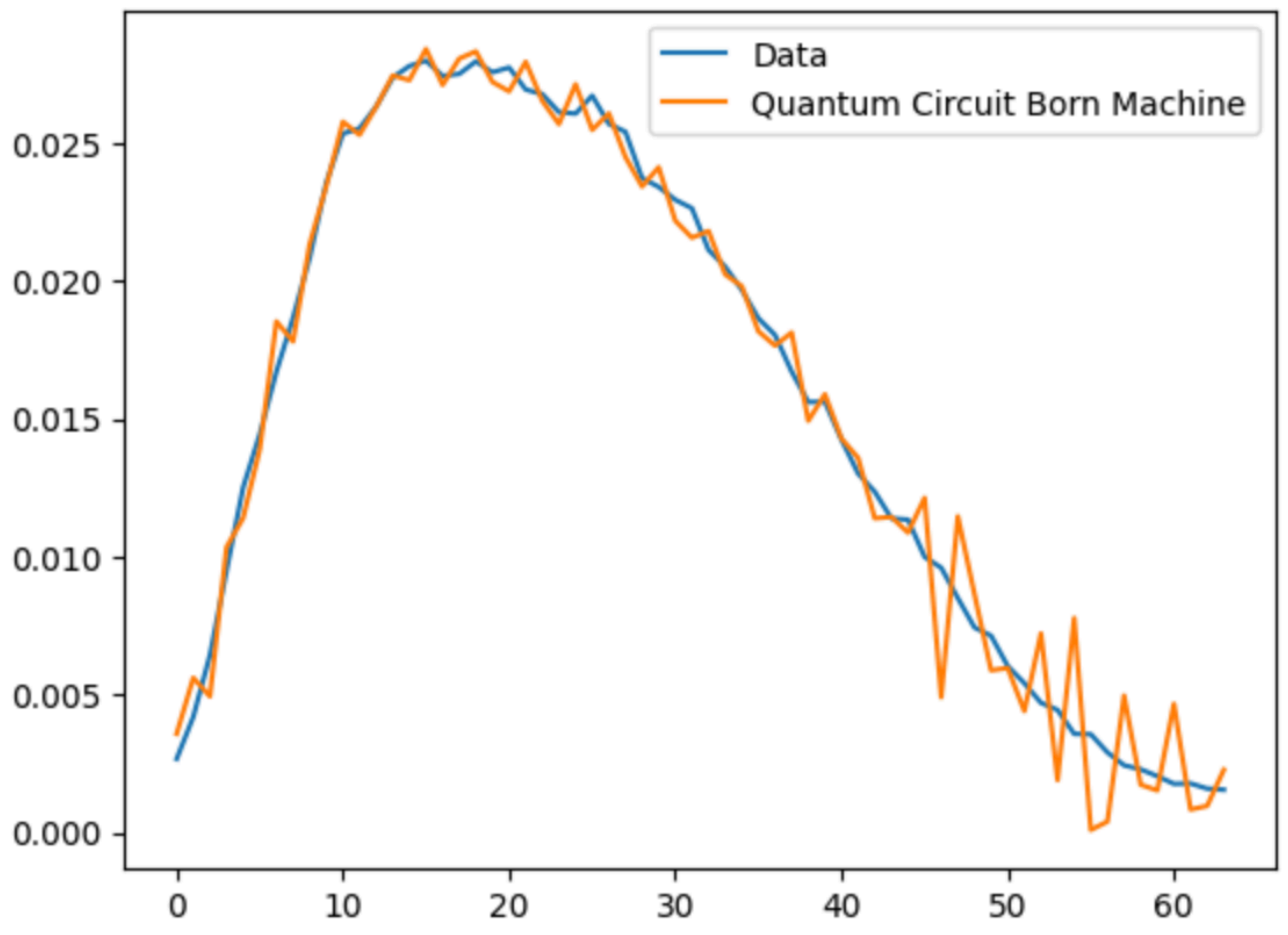
Tracking



Classification



Generative

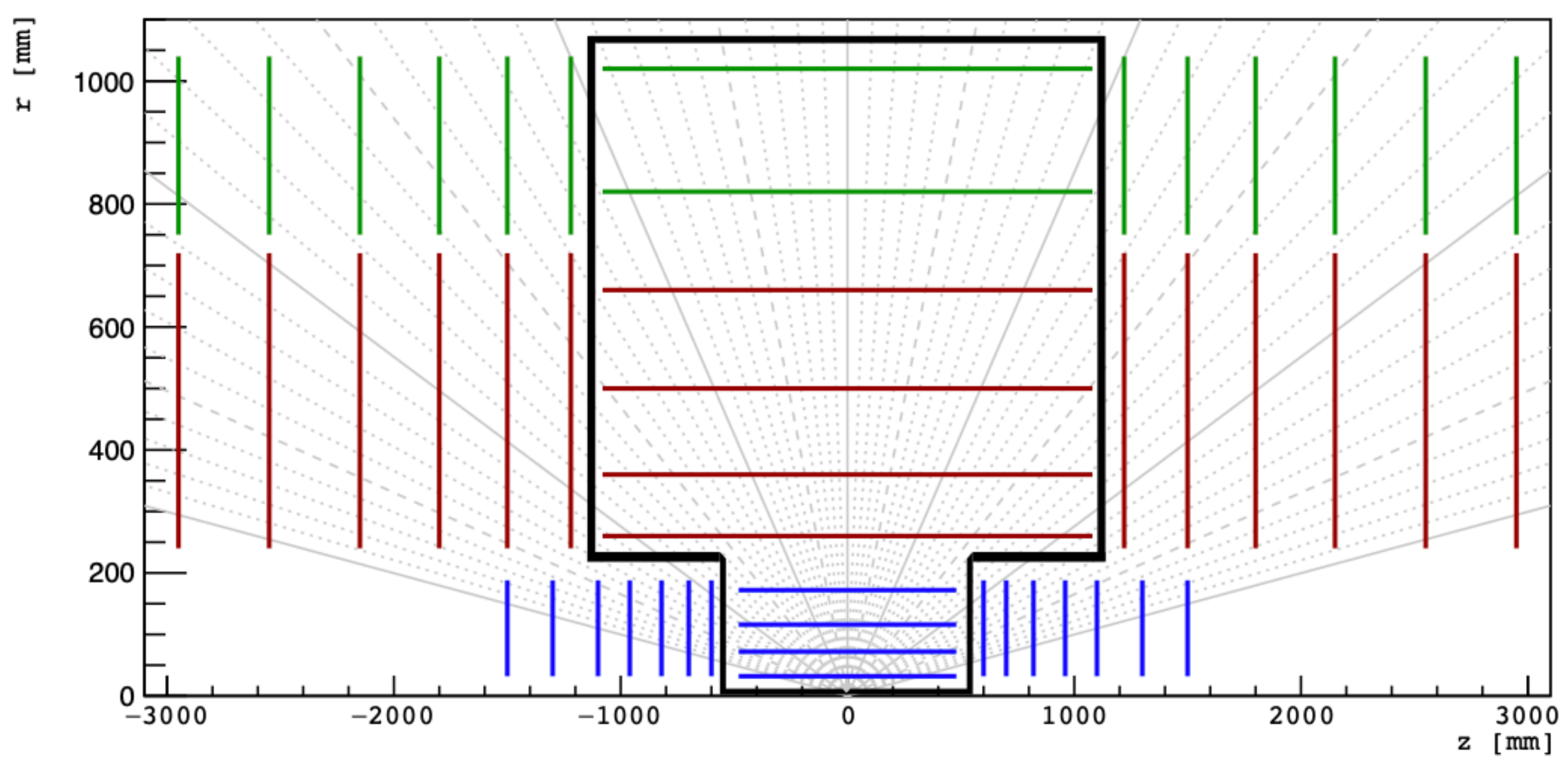


Tracking

QML: tracking with Quantum Graph Neural Networks

https://arxiv.org/pdf/2012.01379.pdf

TrackML dataset from CERN
Kaggle Tracking Machine
Learning challenge

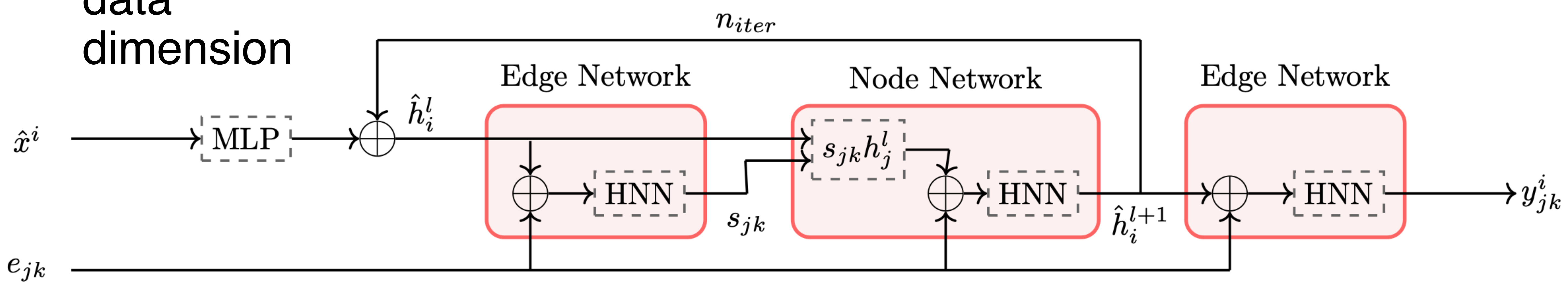


Data are graphs of connected hits

- Hits are **nodes**
- Tracks that connects hits (with geometric constraints) are **edges**

MLP: increase data dimension

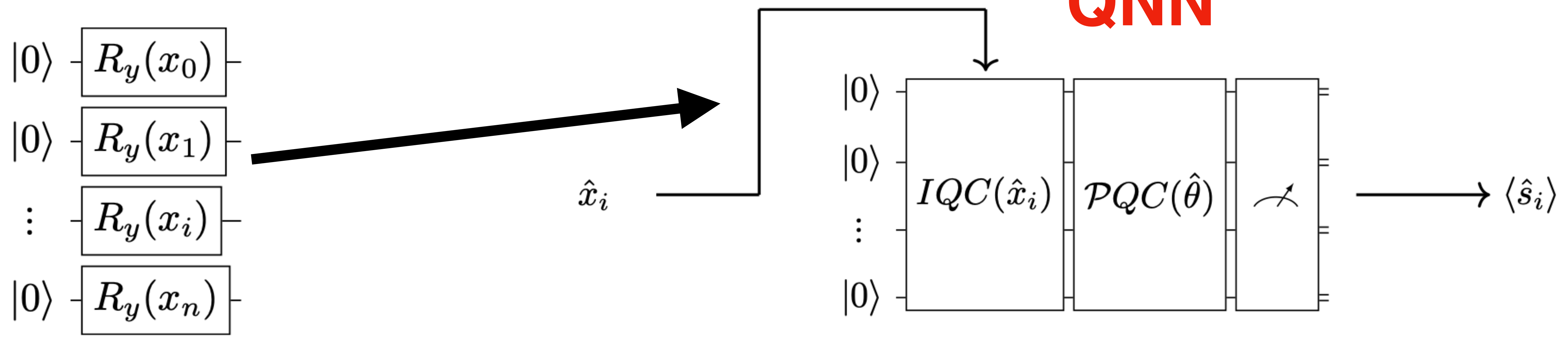
Quantum-classical hybrid architecture



Edge network: QNN with edges as inputs, and has as outputs probabilities for edges to be true (edge features)

Node network: Edges are weighted with edge features. Triplets of connected nodes are built, and fed to a QNN. QNN provides updated nodes as outputs.

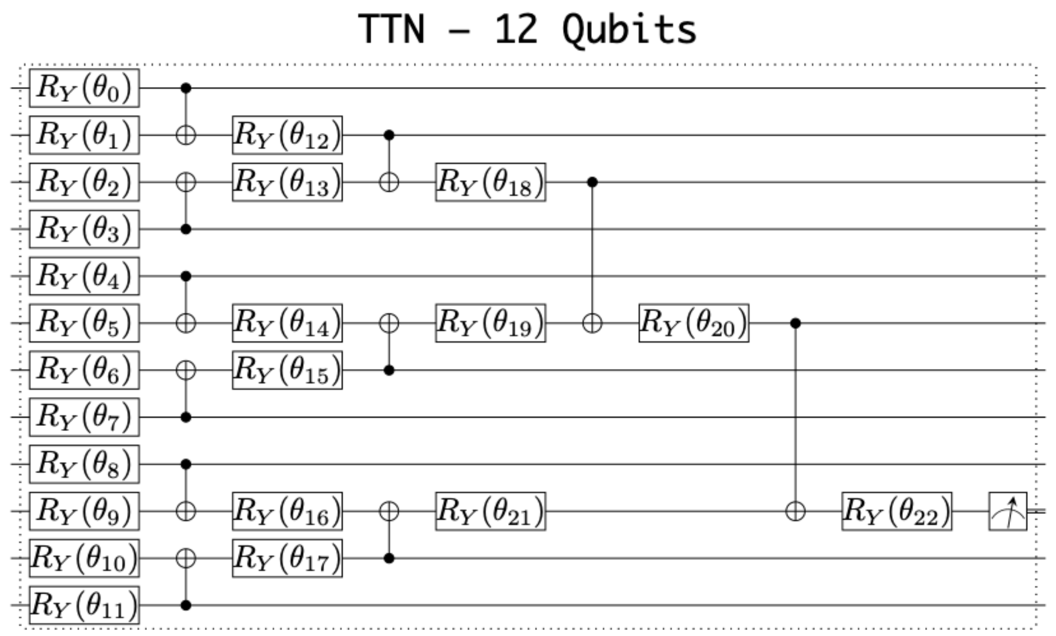
QNN

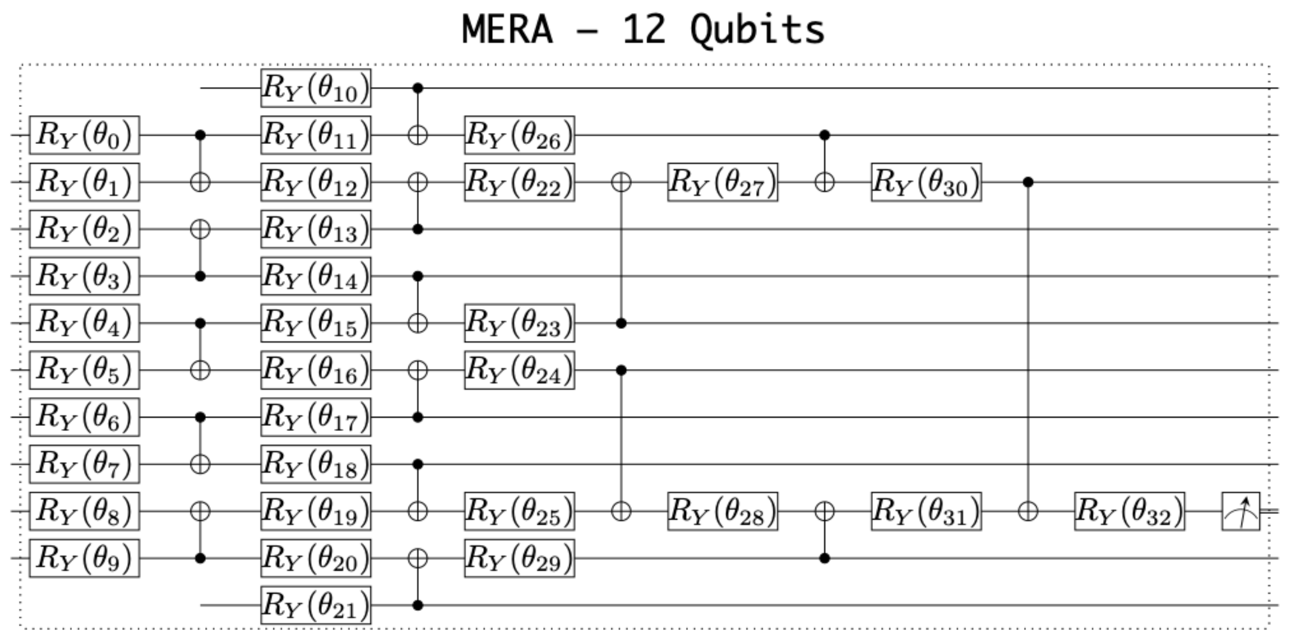


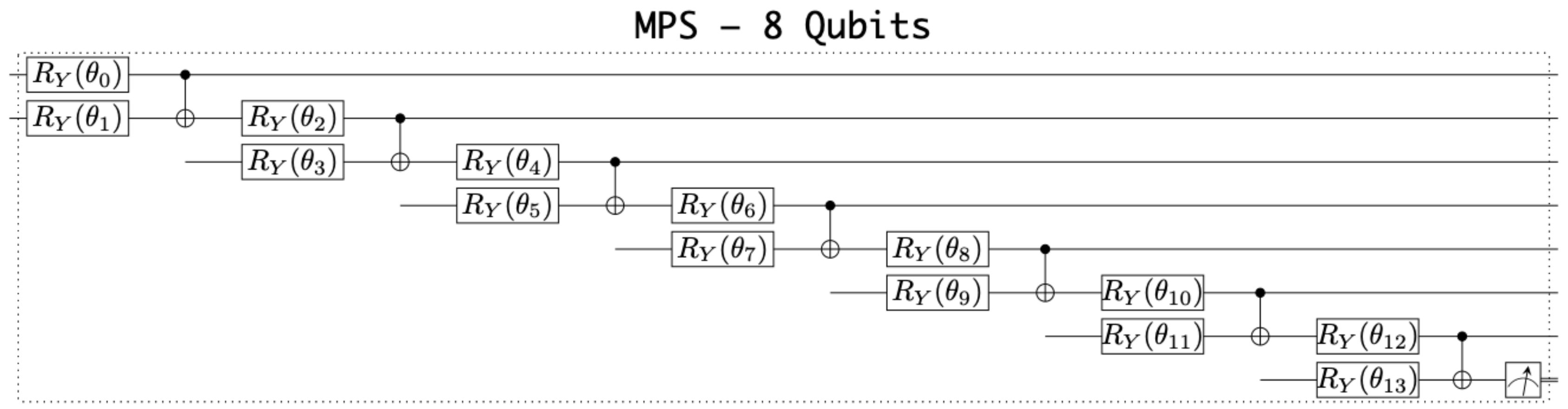
Embedding **Parametrized quantum circuit**

QML: tracking with Quantum Graph Neural Networks

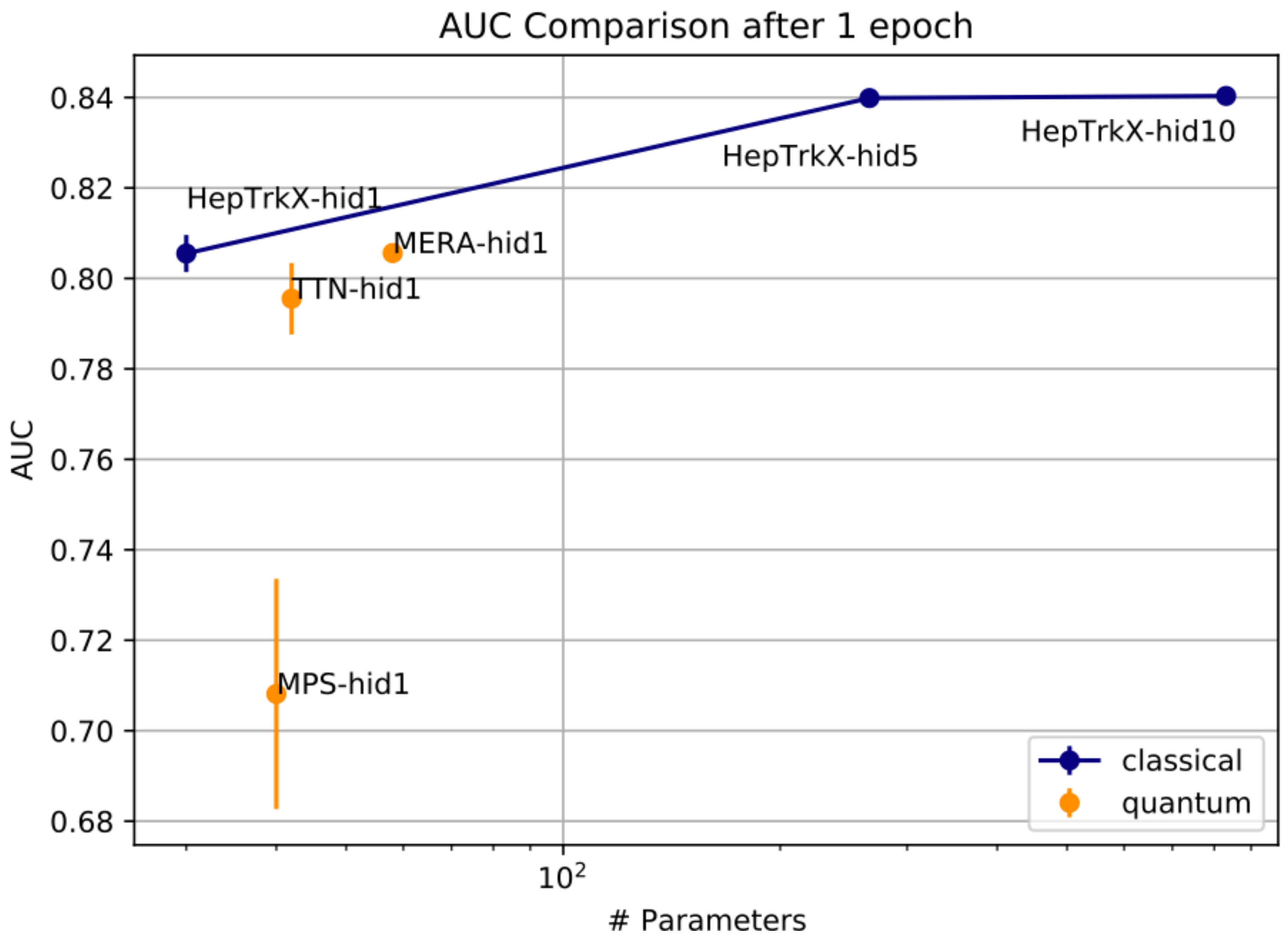
Different variational quantum circuits architectures are trained







Trained to obtain the best true-fake tracks separation

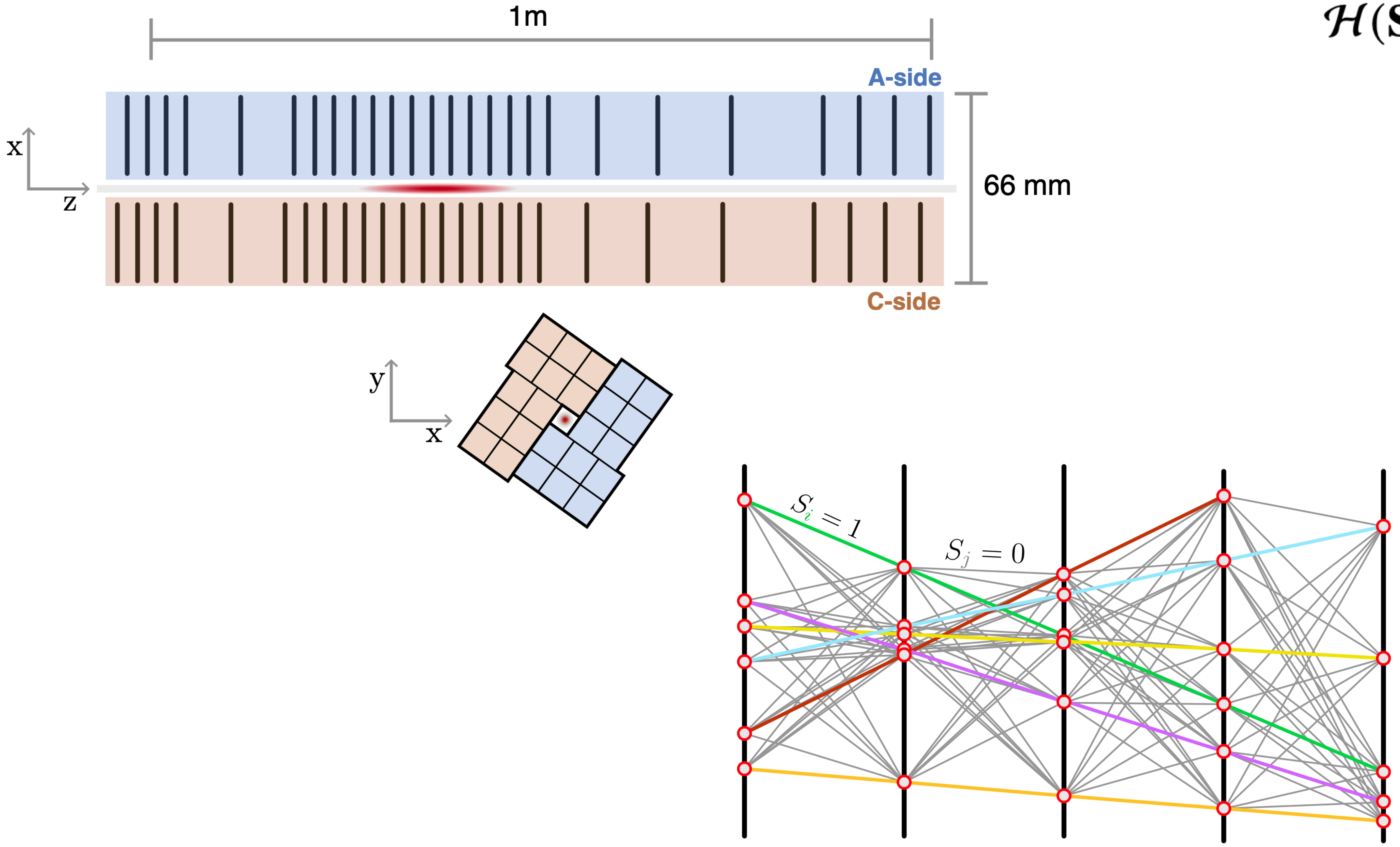


Comparison with classical GNN after 1 epoch.

QGNN trained on CPU/GPU (long training time)

Tracking at LHCb

Vertex detector tracking at LHCb



https://arxiv.org/pdf/2308.00619.pdf

$$\mathcal{H}(\mathbf{S}) = -\frac{1}{2}\sum_{i,j} A_{ij}S_iS_j + \sum_i b_iS_i = -\frac{1}{2}\mathbf{S}^{\mathrm{T}}A\mathbf{S} + \mathbf{b}^{\mathrm{T}}\mathbf{S},$$

$$S_i = \begin{cases} 1 & \text{if the doublet is part of a track} \\ 0 & \text{otherwise} \end{cases}$$

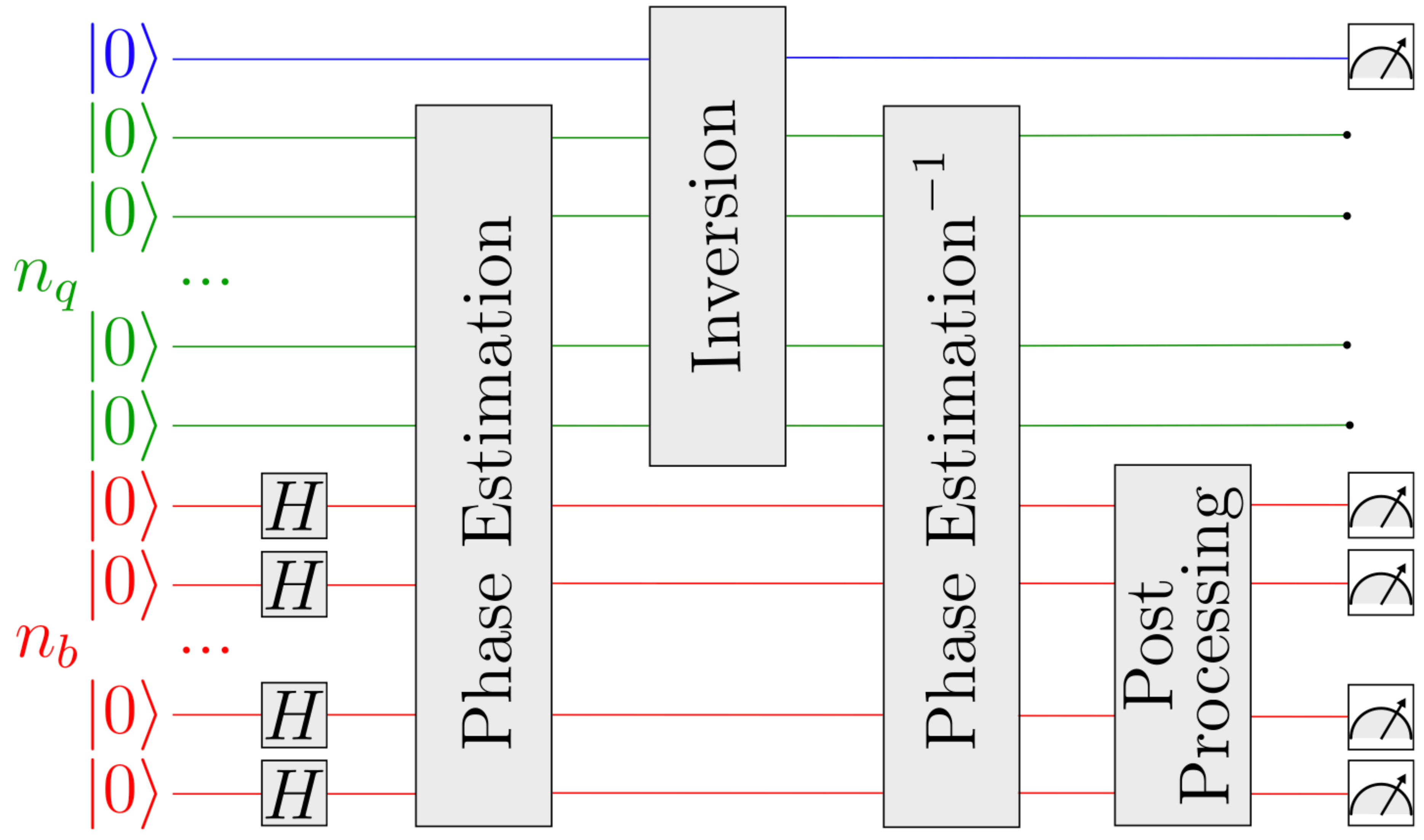
Ising Hamiltonian: the minimum is the solution of tracking problem

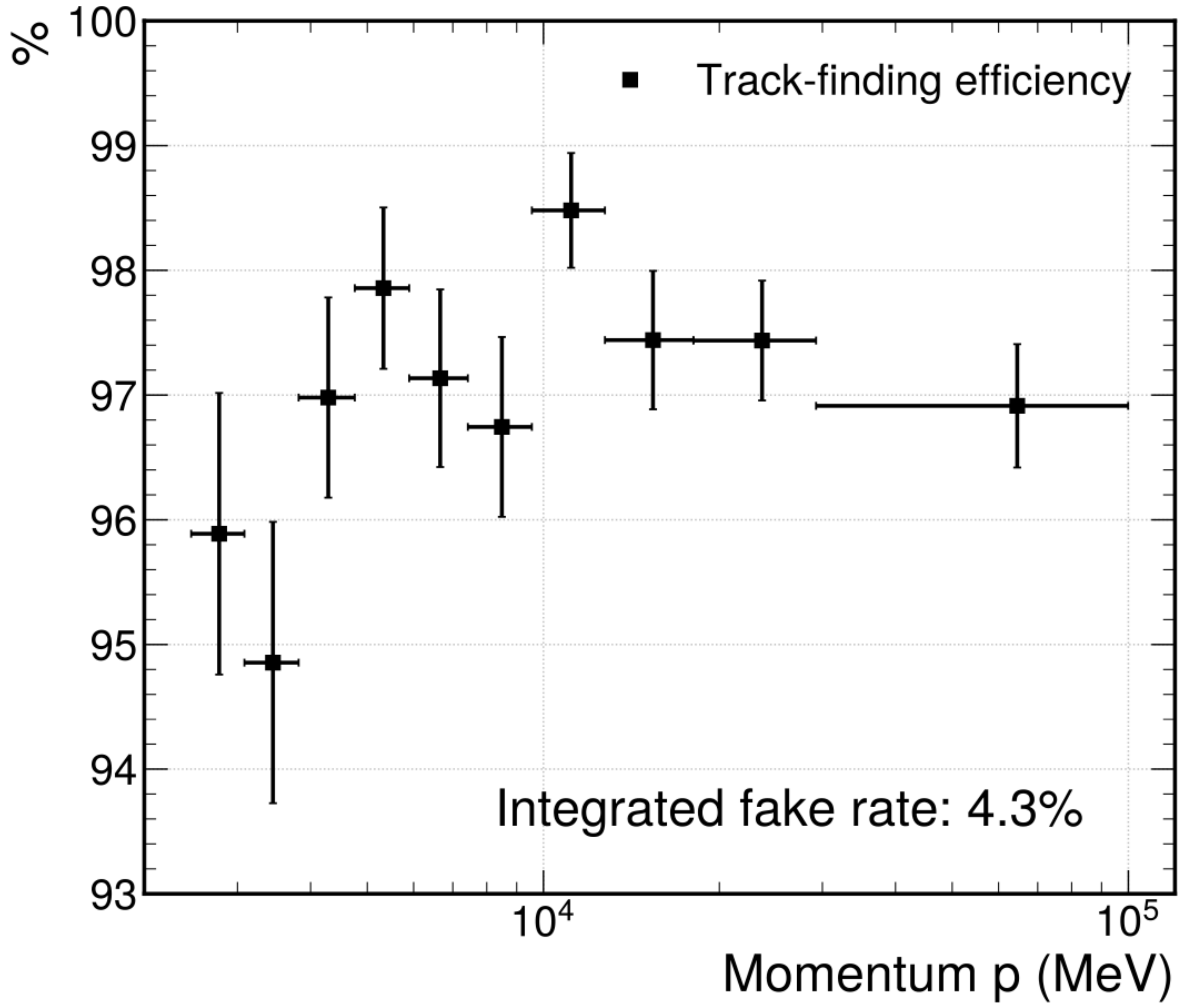
Probably not machine learning in the strict sense, because we are minimizing a Hamiltonian and not a loss function

It is necessary to solve a N x N linear system of equations, with N number of doublets

Tracking at LHCb

HHL quantum algorithm for solving linear problems



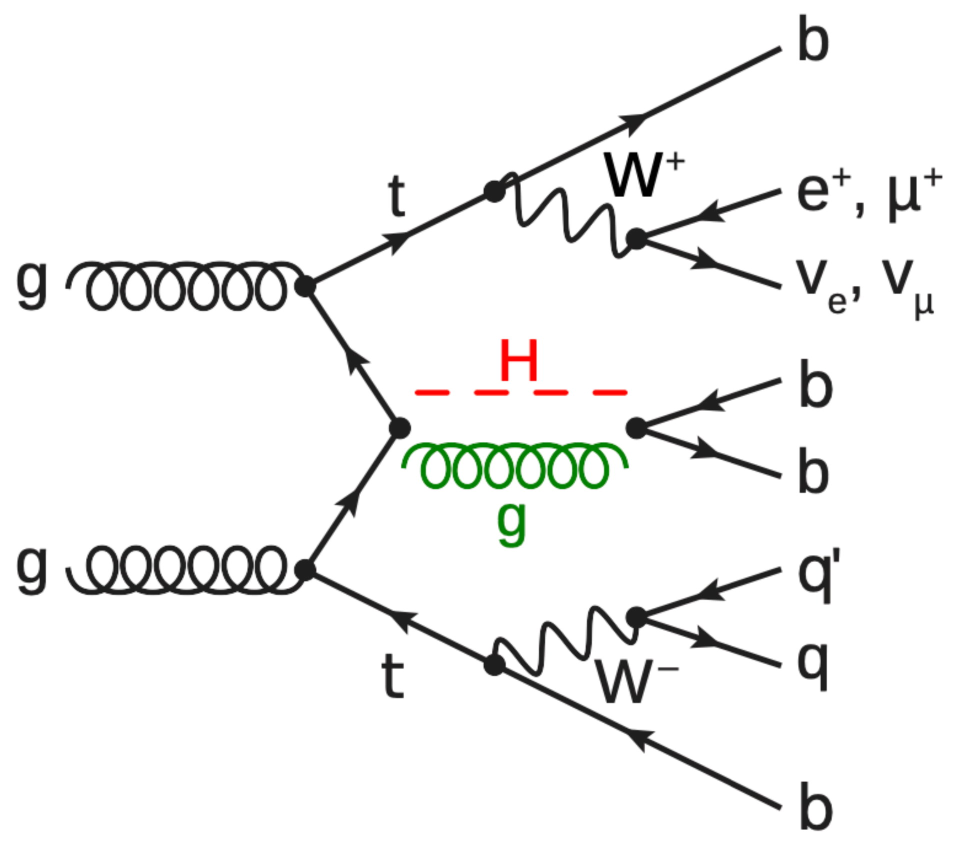


Classification

QML: Higgs classification

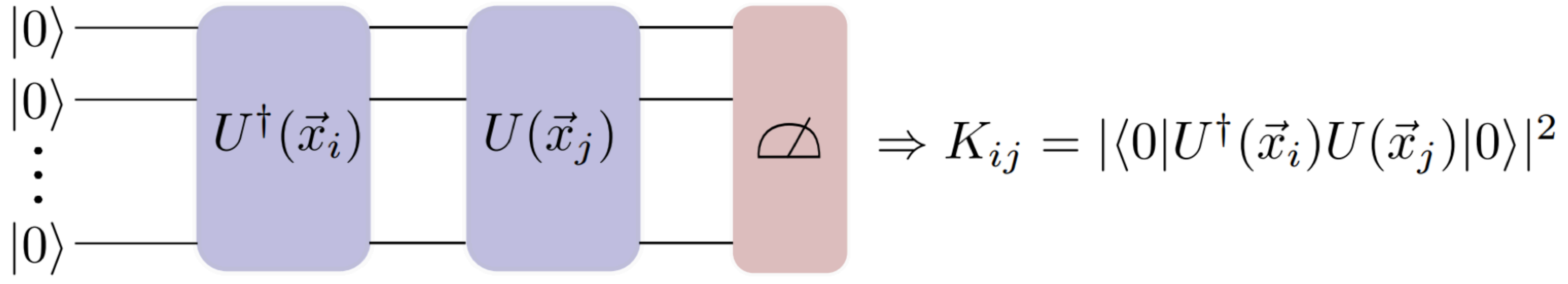
Classification of $t\bar{t}H(b\bar{b})$ versus the dominant $t\bar{t}b\bar{b}$ background

https://arxiv.org/pdf/2104.07692.pdf



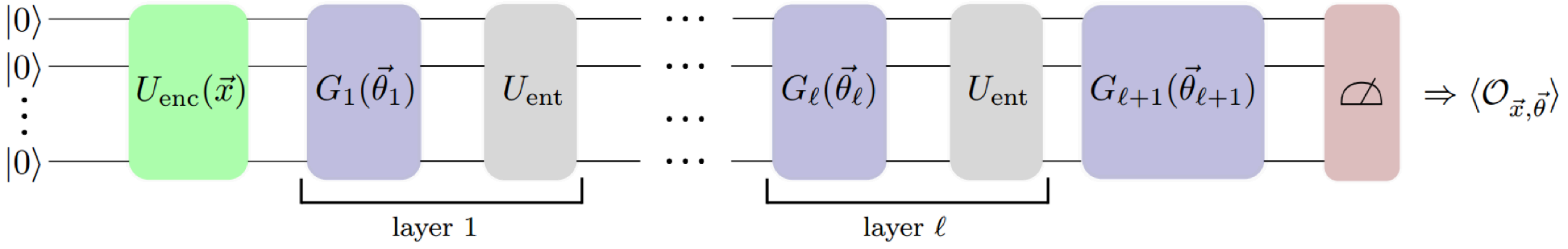
- Data from simulation with CMS Delphes
- 67 input features are reduced to 12 (8 in latent space) with a classical neural network Auto-encoder
- Two approaches are used for the QML classification: Quantum Support Vector Machine, and Variational Quantum Circuit

Quantum Support Vector Machine

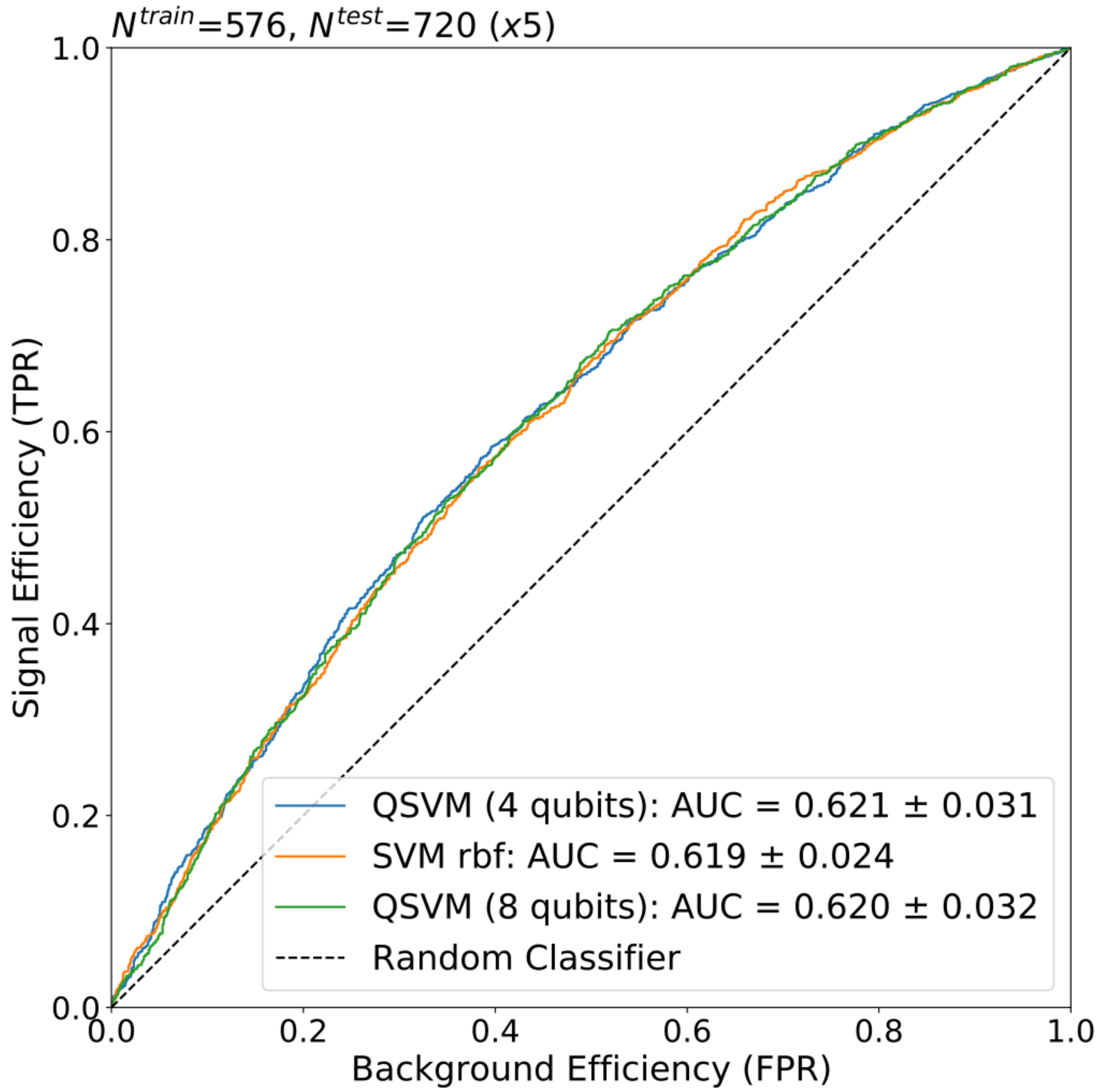


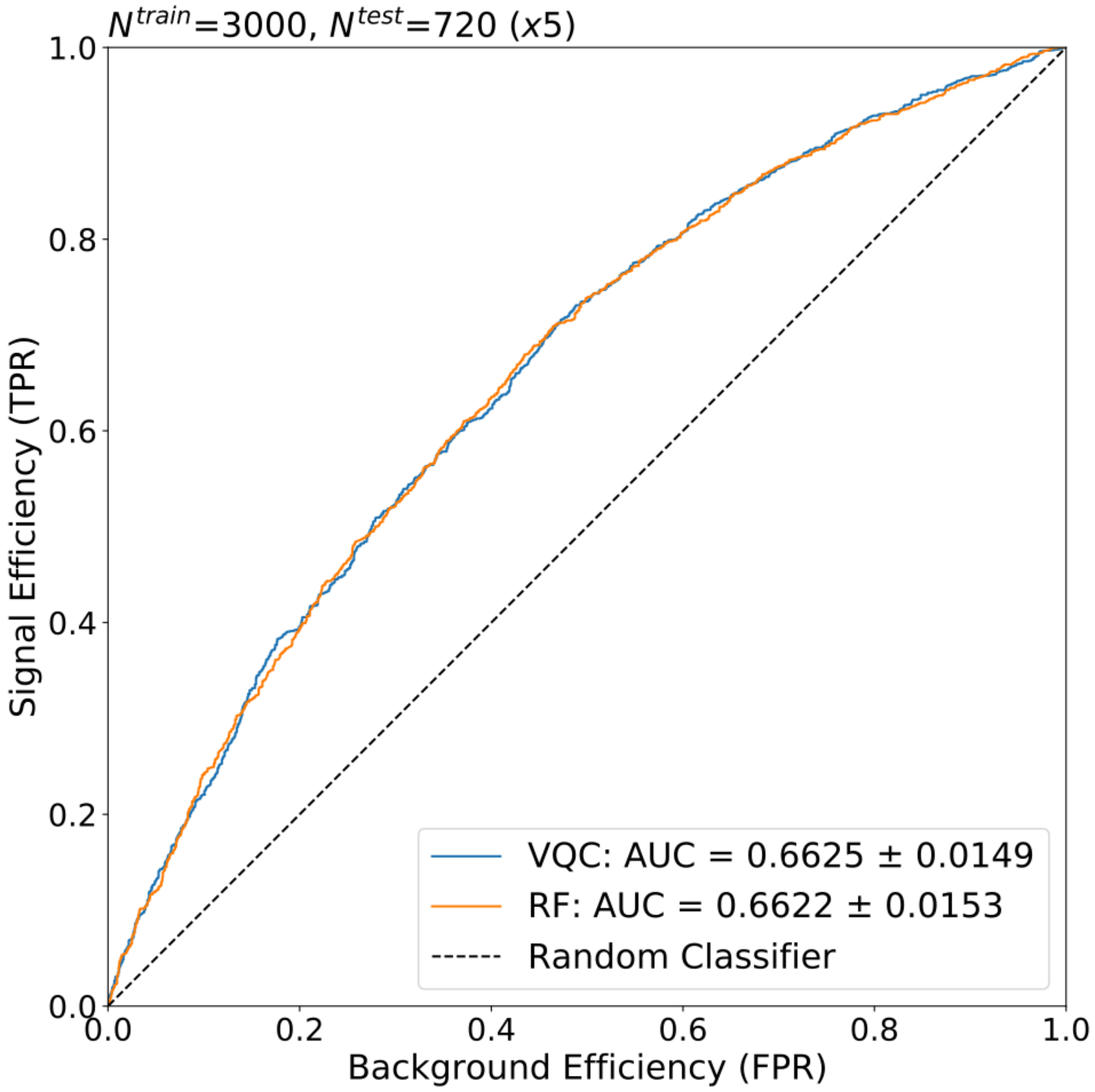
Kernel: internal product of the Hilbert space, obtained as measurement

Variational Quantum Circuit with L layers



QML: Higgs classification

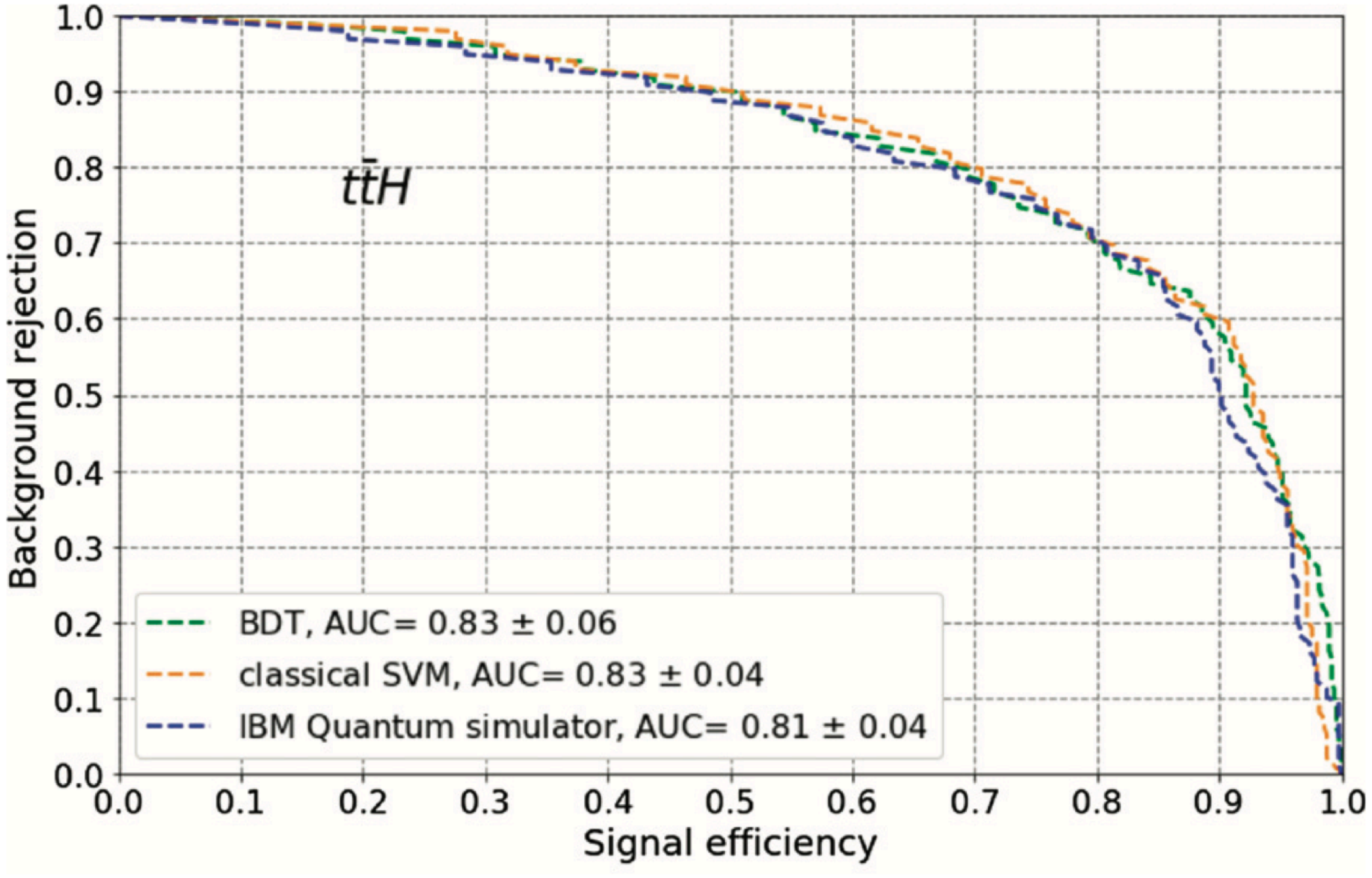


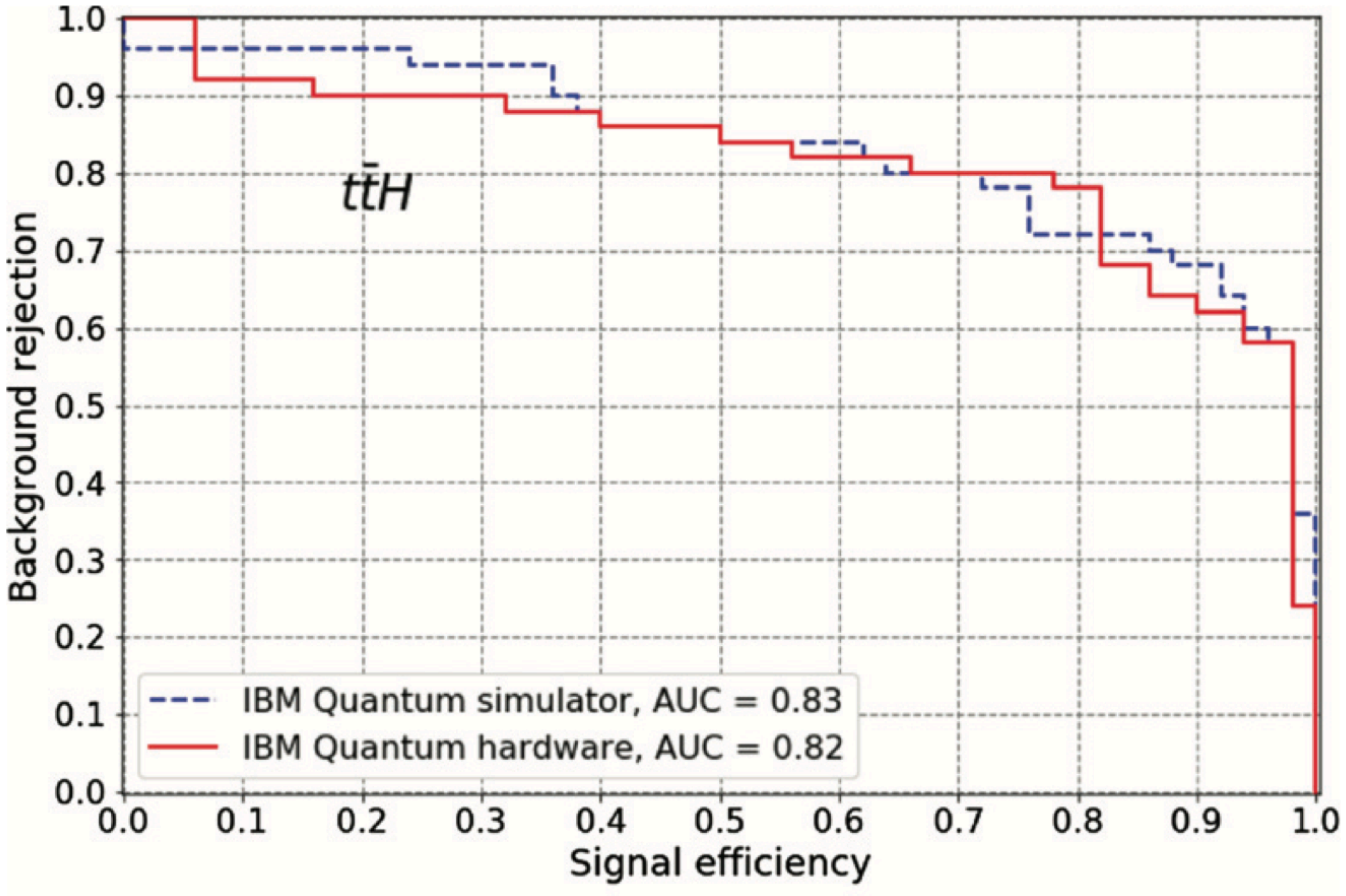


QML: Higgs classification

Higgs classification on IBM quantum simulator and quantum hardware (10 qubit)

https://iopscience.iop.org/article/10.1088/1361-6471/ac1391/pdf





Trained and evaluated in hardware. Simulator and hardware have a similar performance

QML: Higgs classification

Classification of H → γγ versus diphoton background by using a **programmable quantum annealer**

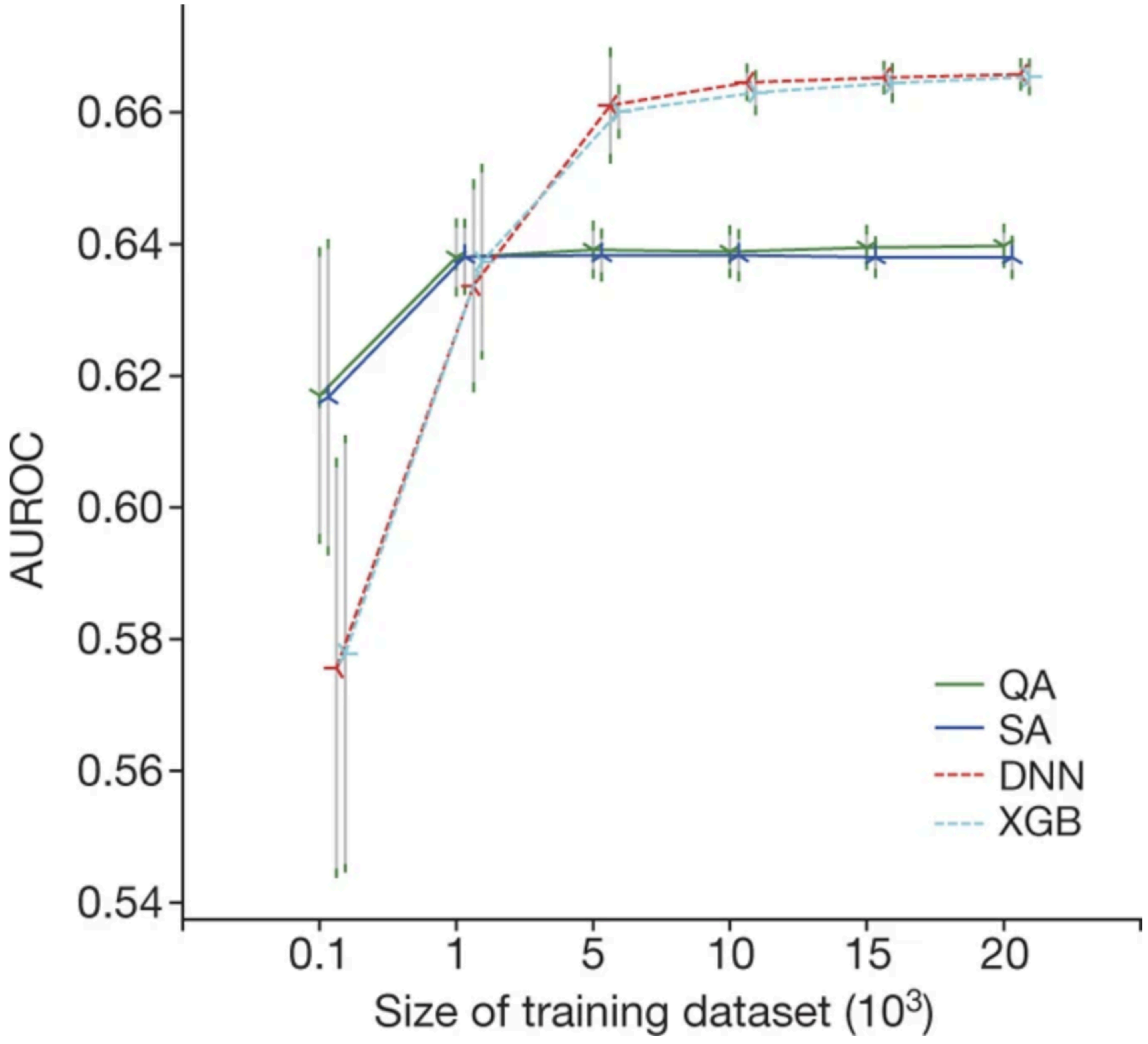
Quantum annealing (QA)

$$H = \sum_{i,j} J_{ij} s_i s_j + \sum_i h_i s_i$$

i and j are event indexes, J_{ij} and h_i are constructed from dataset and true labels

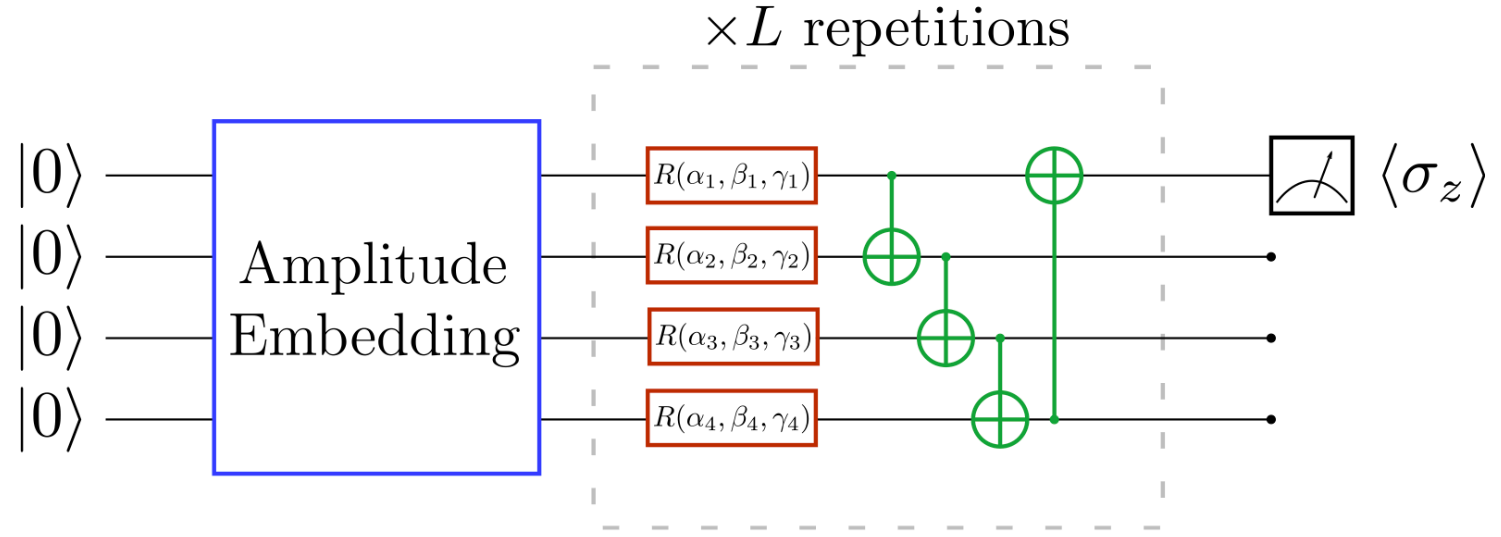
- DNN performs better than QA for large datasets (but still comparable)
- QA achieve the asymptotic performance with a smaller dataset than DNN

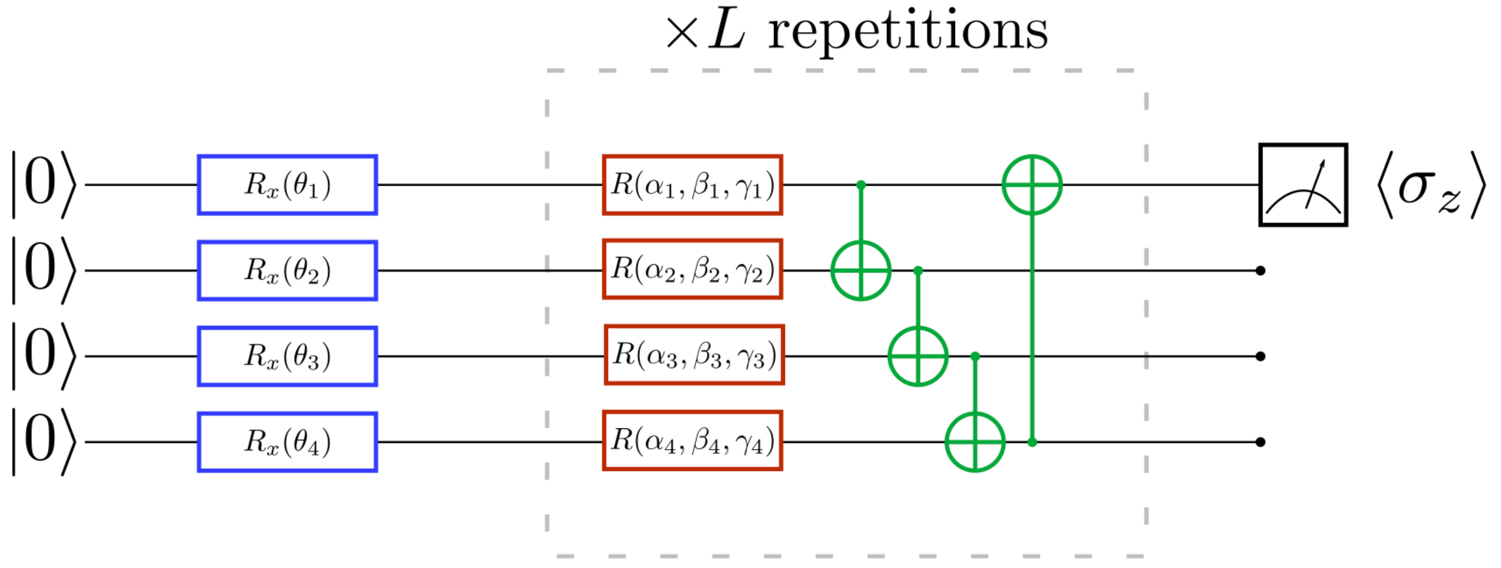
Nature 550 (2017) 7676, 375-379



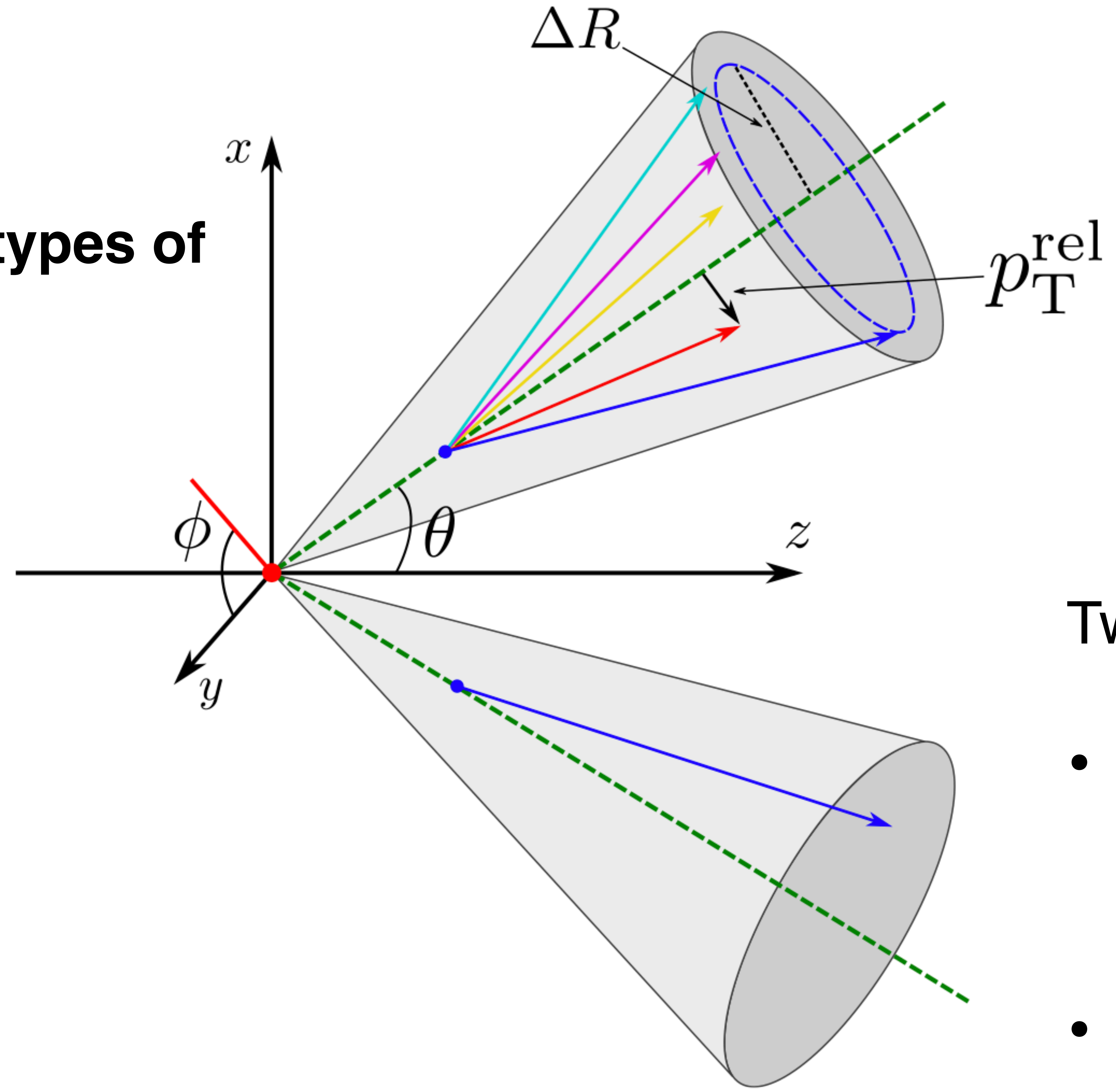
QML: b-jet tagging at LHCb

- Study performed with official LHCb full simulation
- Classification of b and $\bar{b}$ jets
- Variational Quantum Circuits with **different types of data embedding** are tested





A total of 16 features related to the jet substructure are considered

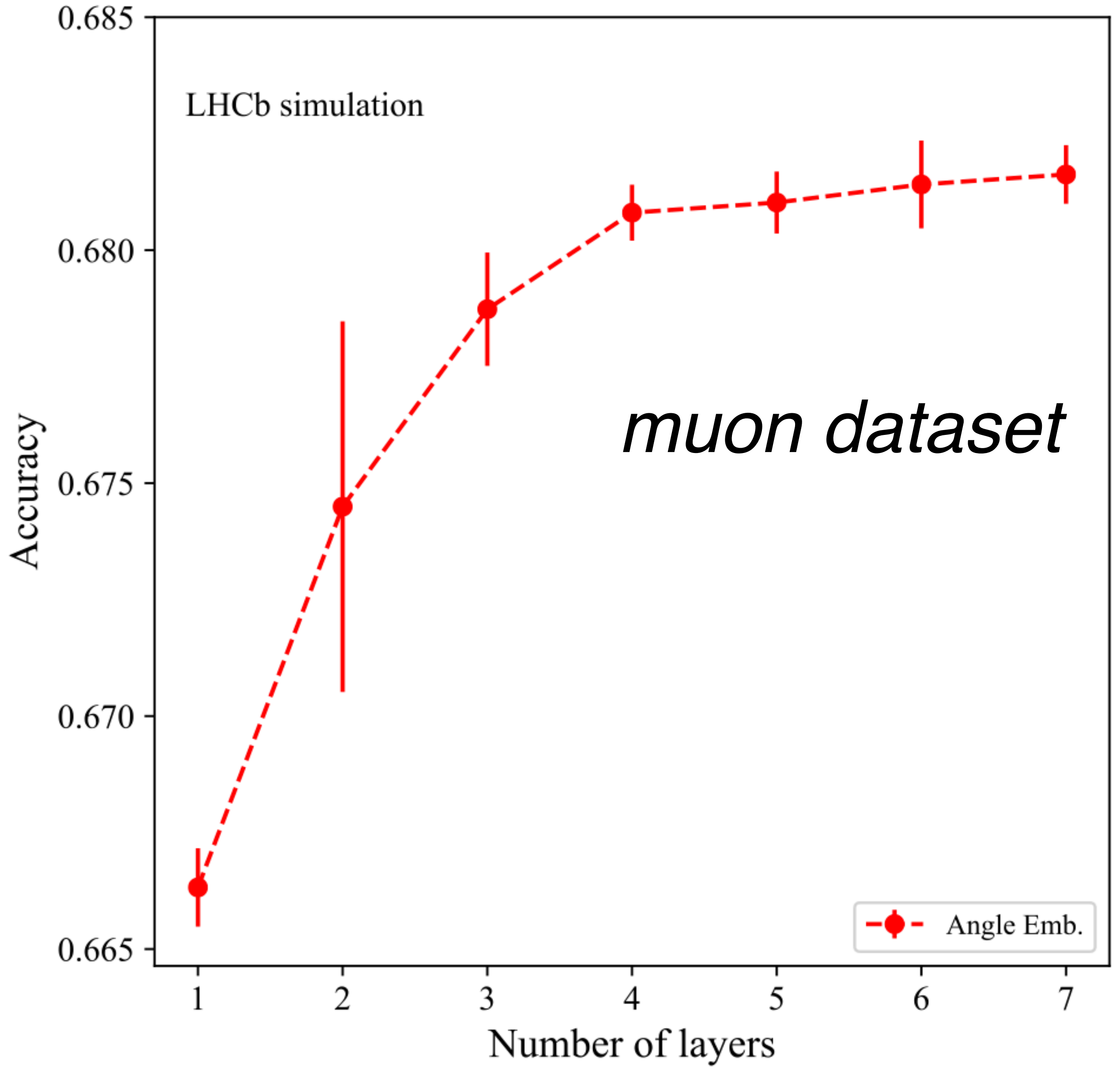


JHEP08(2022)014

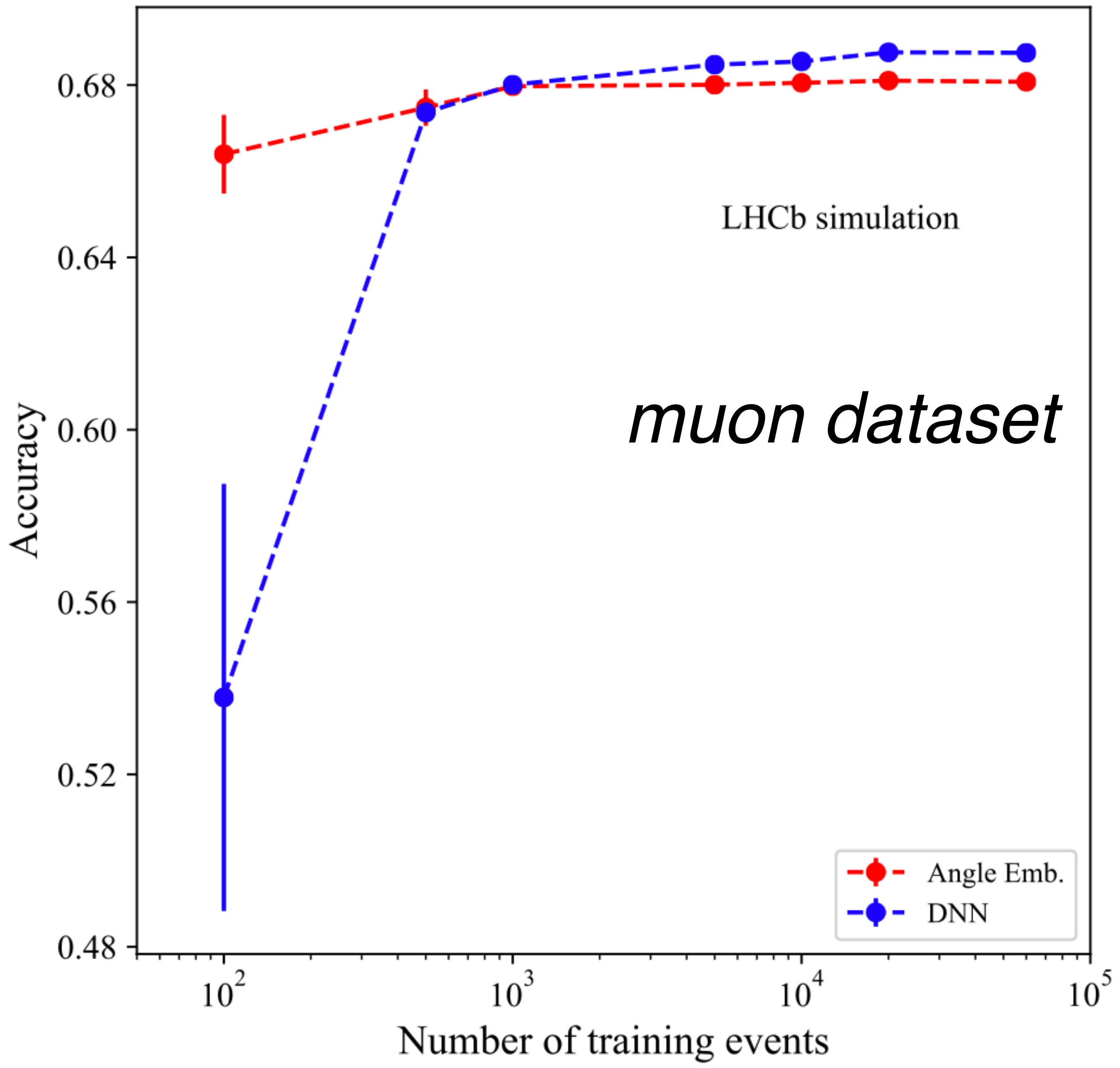
Two datasets/set of features:

- **Muon dataset**: jets with at least one muon, 3 muon features+jet charge
- **Complete dataset**: all jets, 15 particle features+jet charge

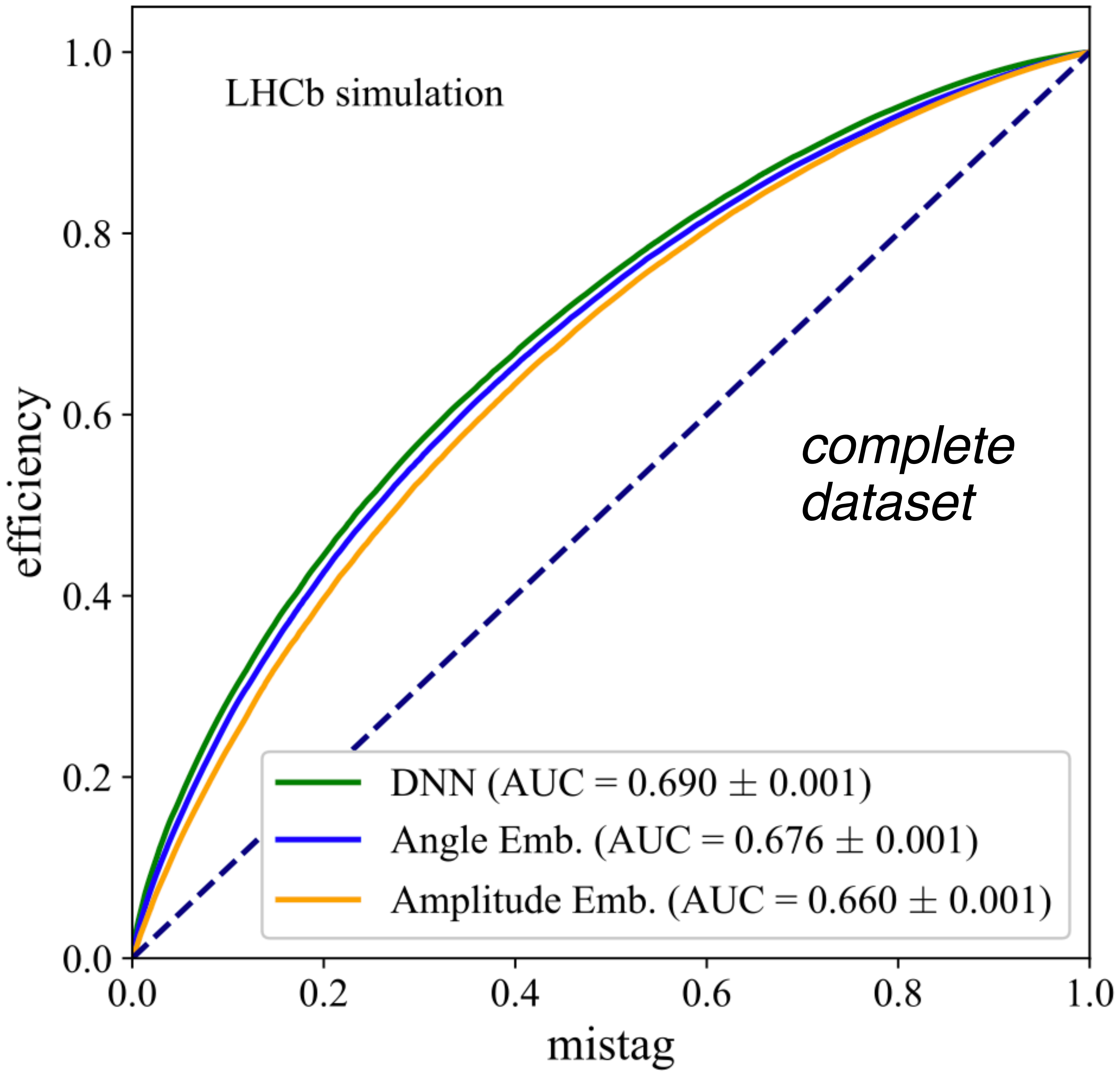
QML: b-jet tagging at LHCb



Different number of rotational layers tested: **the accuracy saturates after few layers**



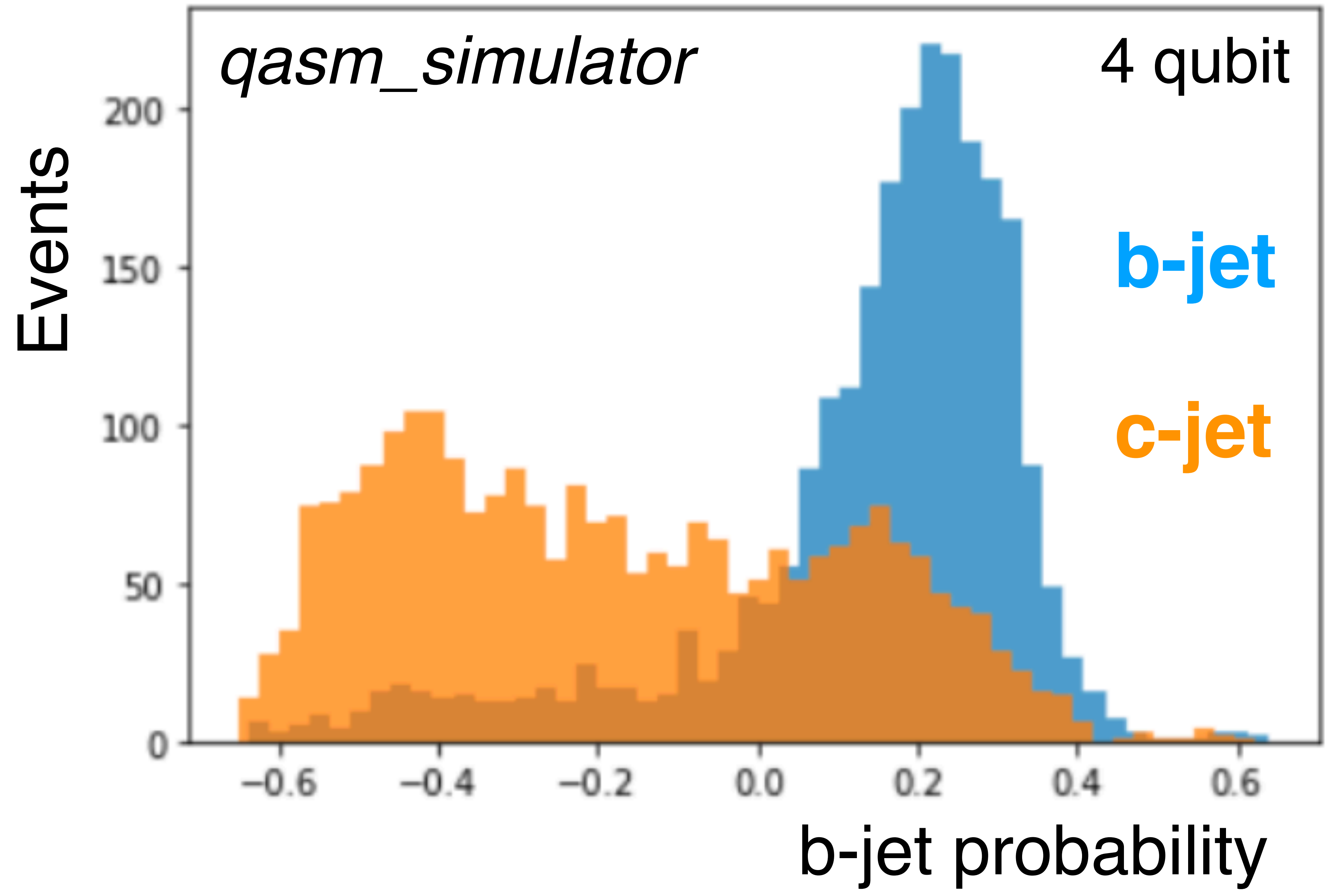
Compared to a classical DNN, **the quantum classifier requires less training events to achieve the same accuracy**

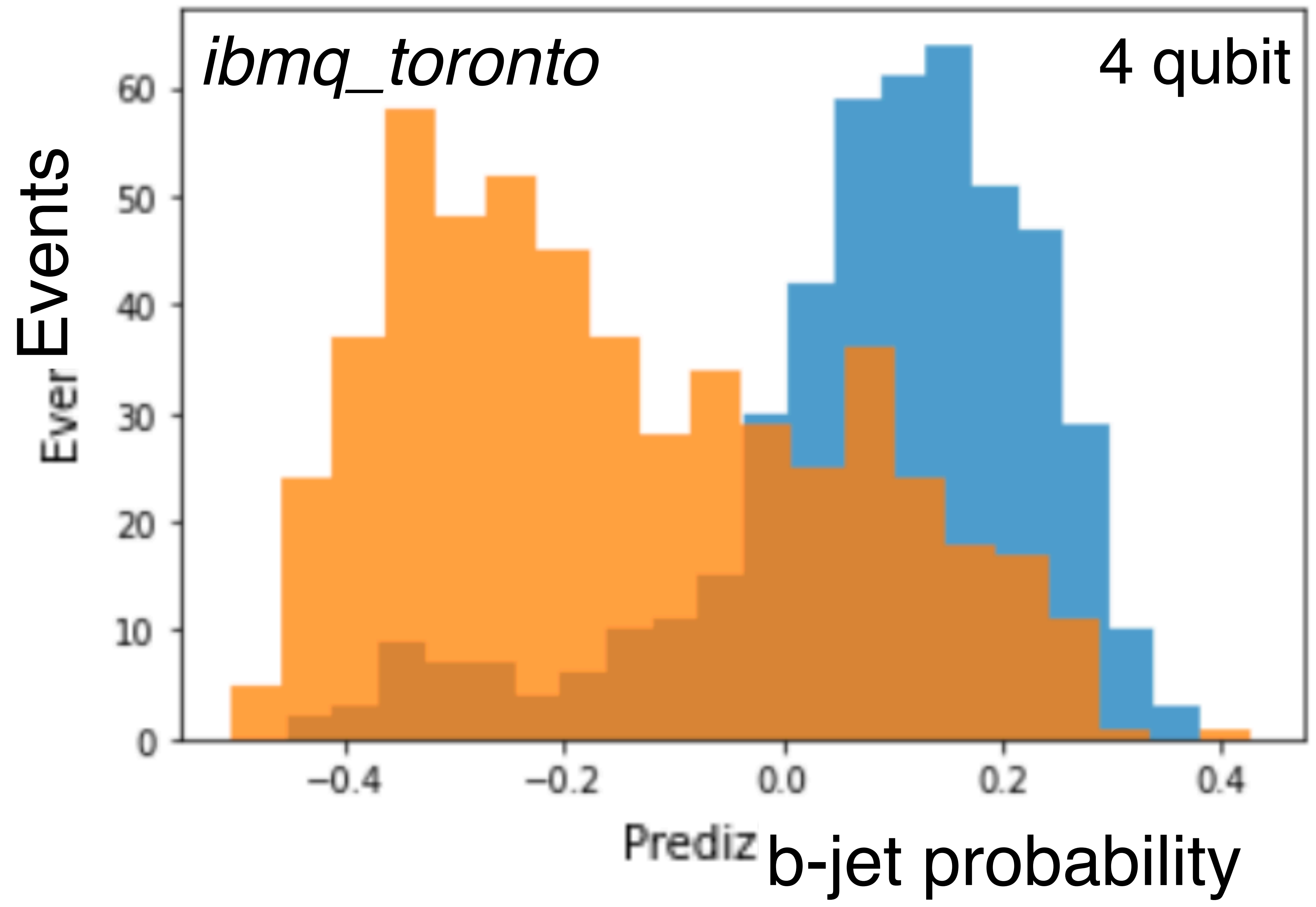


The DNN and the quantum circuits show similar ROC areas

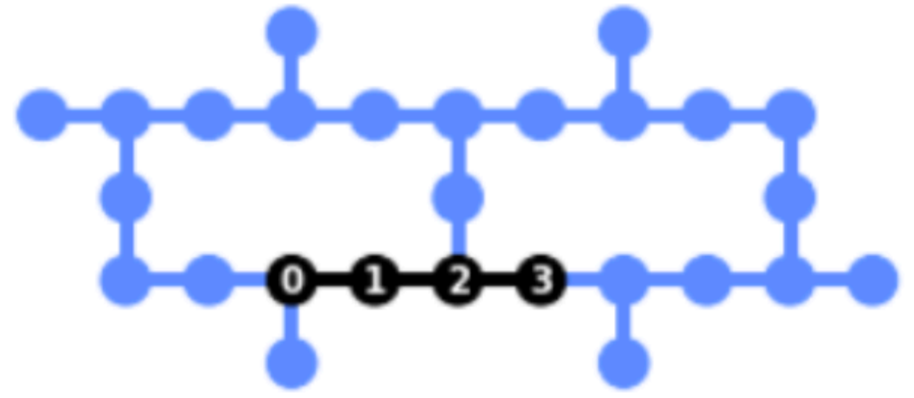
QML: b-jet tagging at LHCb

- The evaluation of the pre-trained quantum circuit for b vs c has been performed on **IBM hardware**
- b-jet probability: probability to obtain 0 by measuring the output qubit (1000 shots per event)
- For this task the circuit has been implemented using the **Qiskit** library, (angle embedding is considered)
- **The probability distributions show some differences, but the discriminating power is similar**



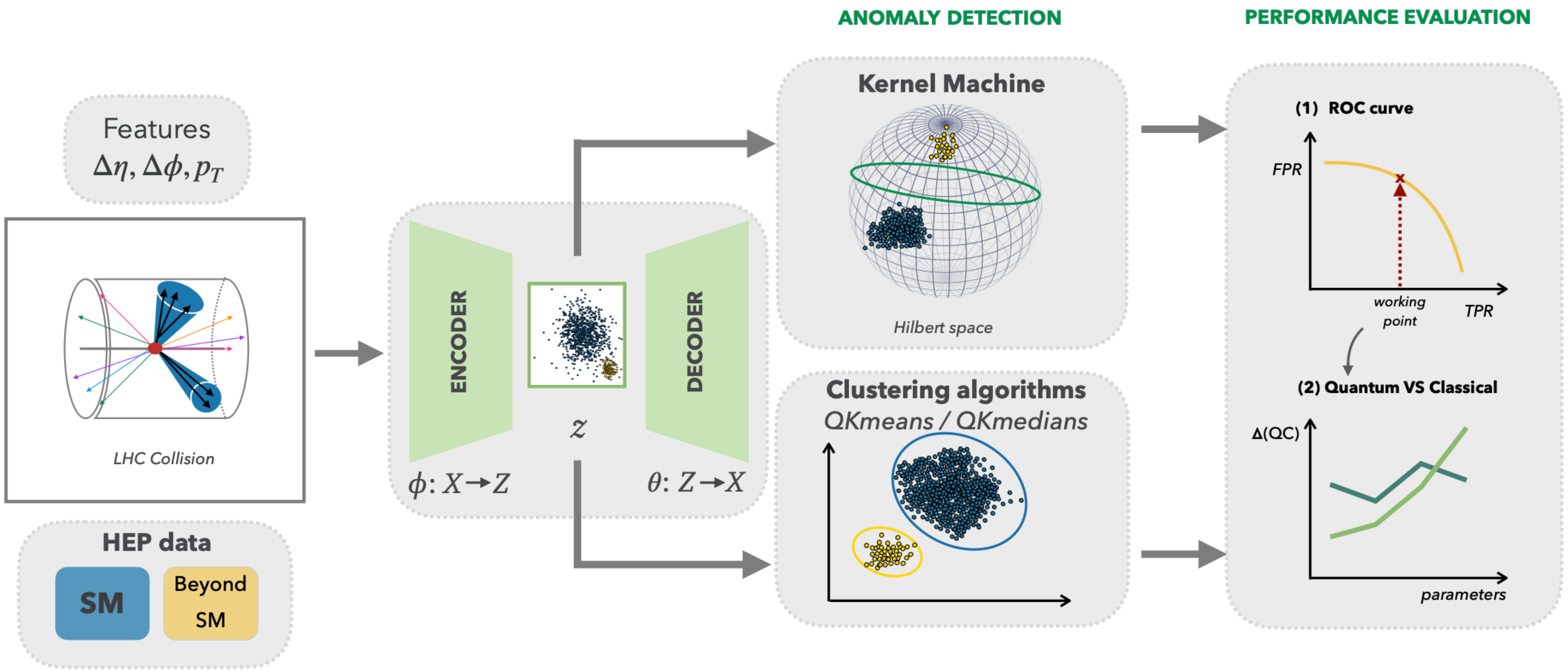


ibmq_toronto 27 qubits



QML: anomaly detection

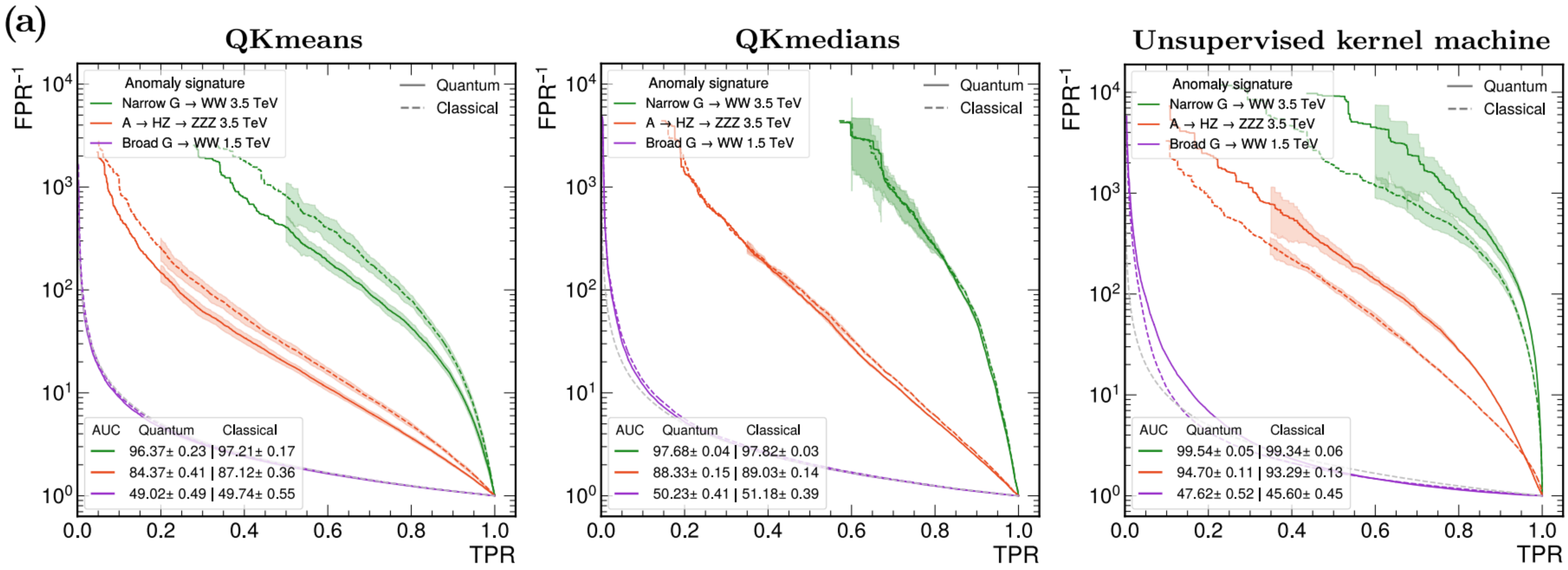
- **Example of unsupervised QML**: new physics is searched as deviation from the Standard Model prediction
- Anomaly detection in **dijet events**, dataset from CMS Delphes simulation



https://arxiv.org/abs/2301.10780

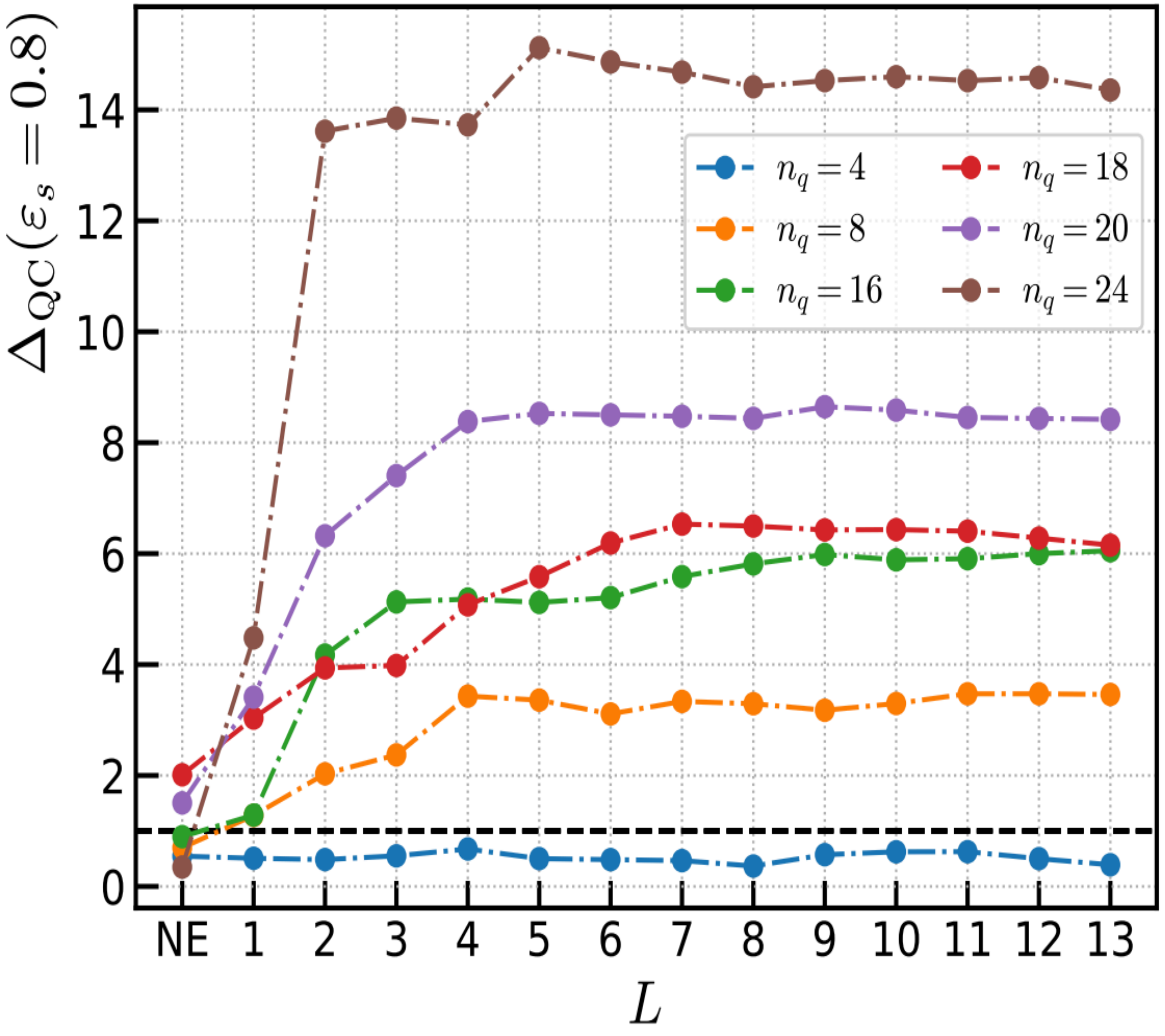
QML: anomaly detection

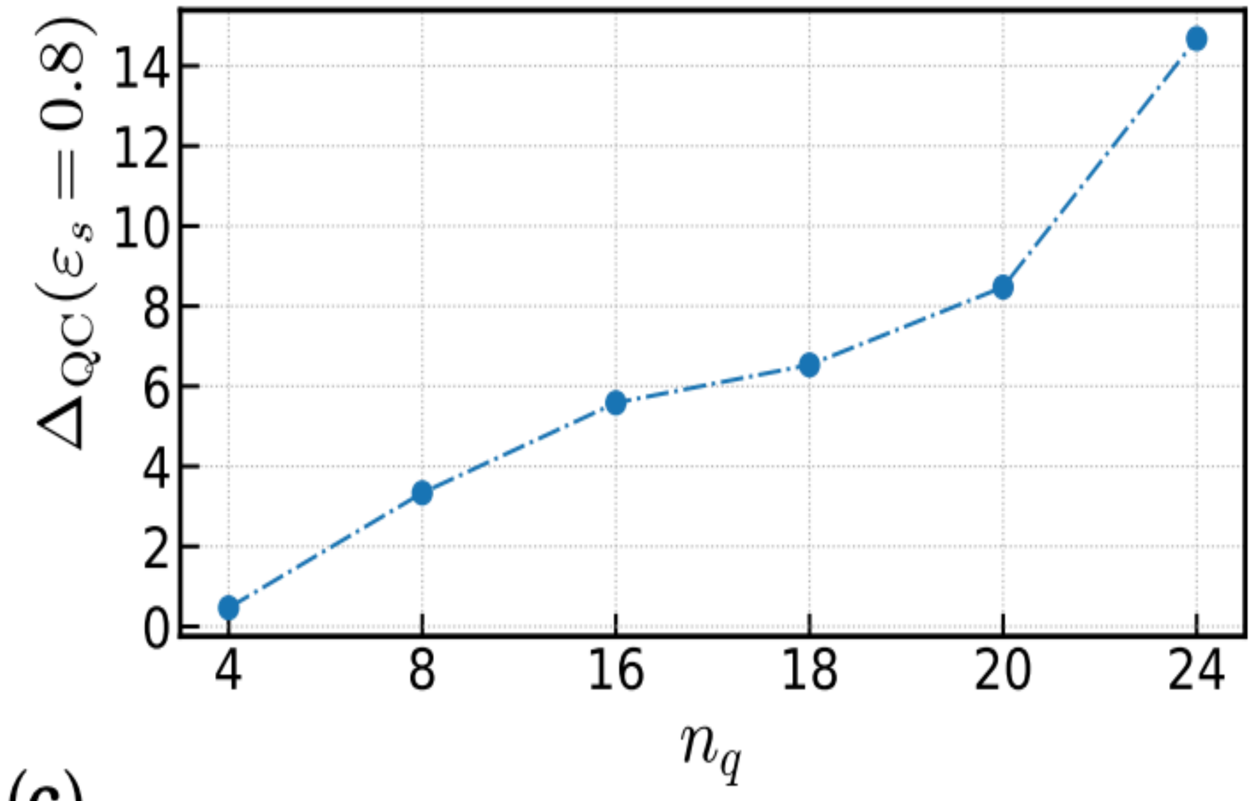
One of the first examples of quantum advantage in HEP!



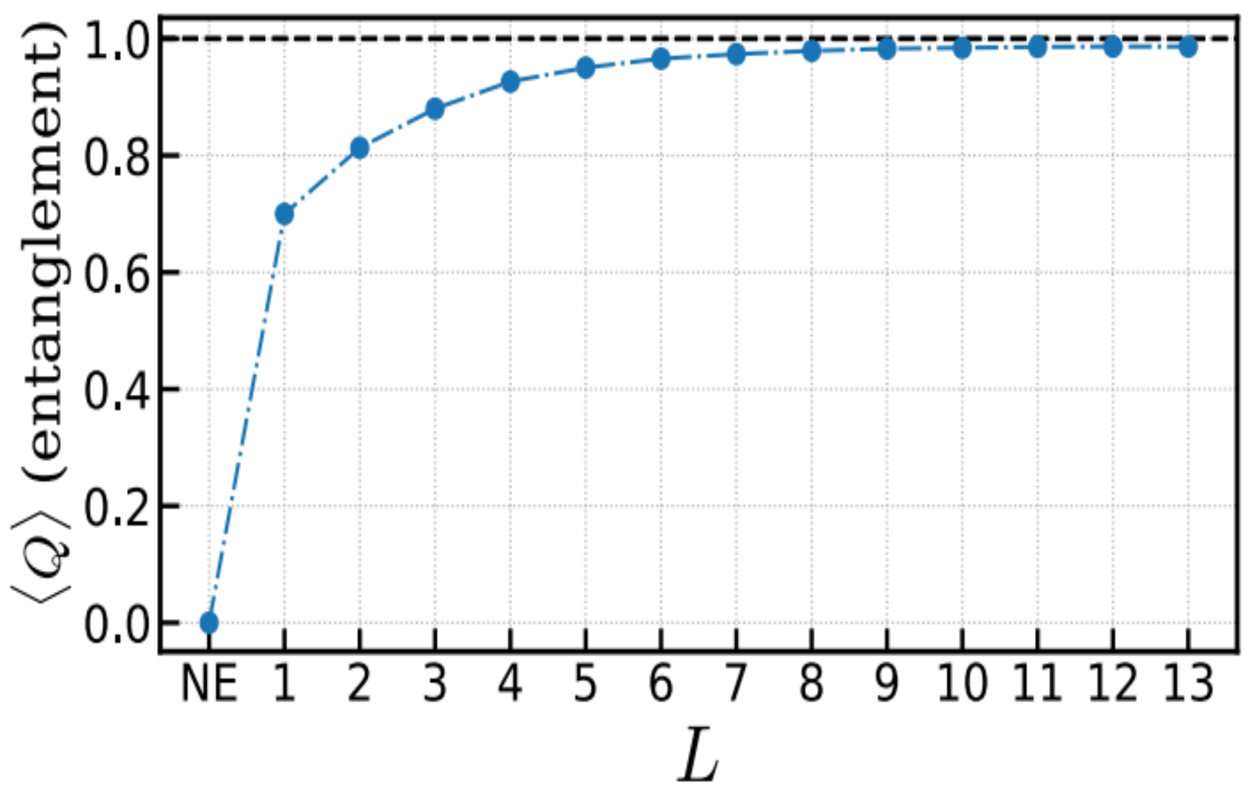
QML: anomaly detection

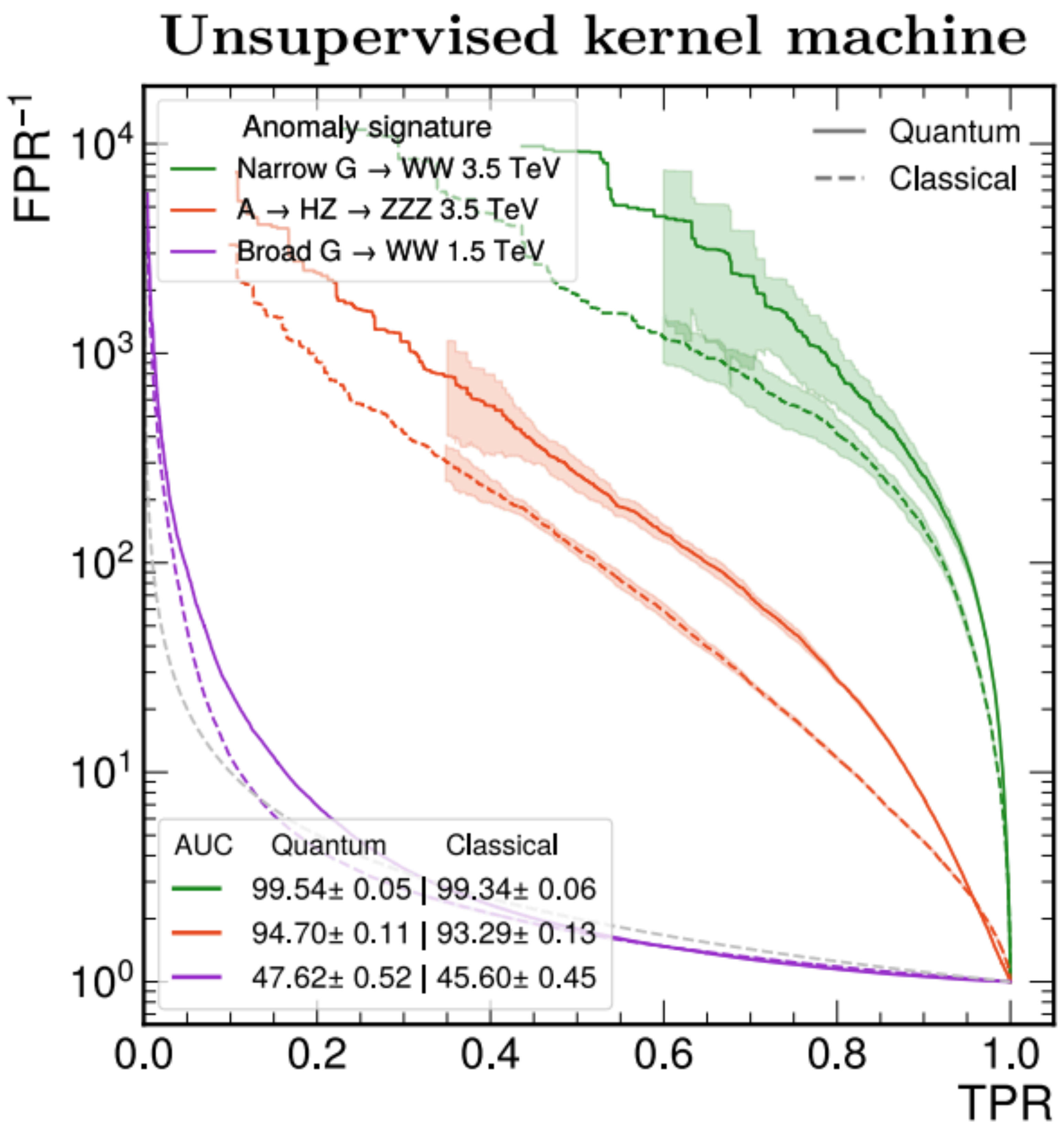
One of the first examples of quantum advantage in HEP!





(c)





Generative QML

Generative QML: Quantum Born Machines

- **Quantum Circuit Born Machines (QCBM)** make use of the stochastic nature of quantum measurements, no classical analogs

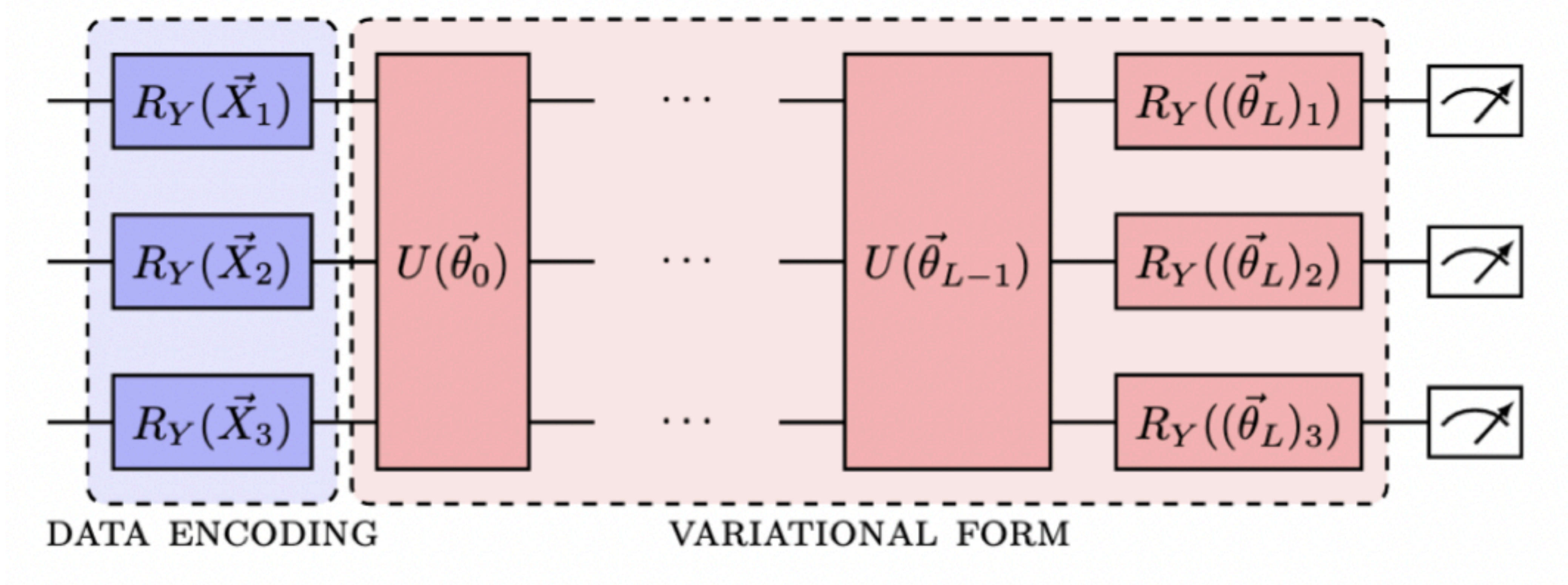
- Each base element of the quantum space is mapped to a specific configuration of the system we want to simulate

$$p_\theta(x) = |\langle x|\psi(\theta)\rangle|^2$$

- As an example if we have N qubits we can simulate a distribution in 2^N bins

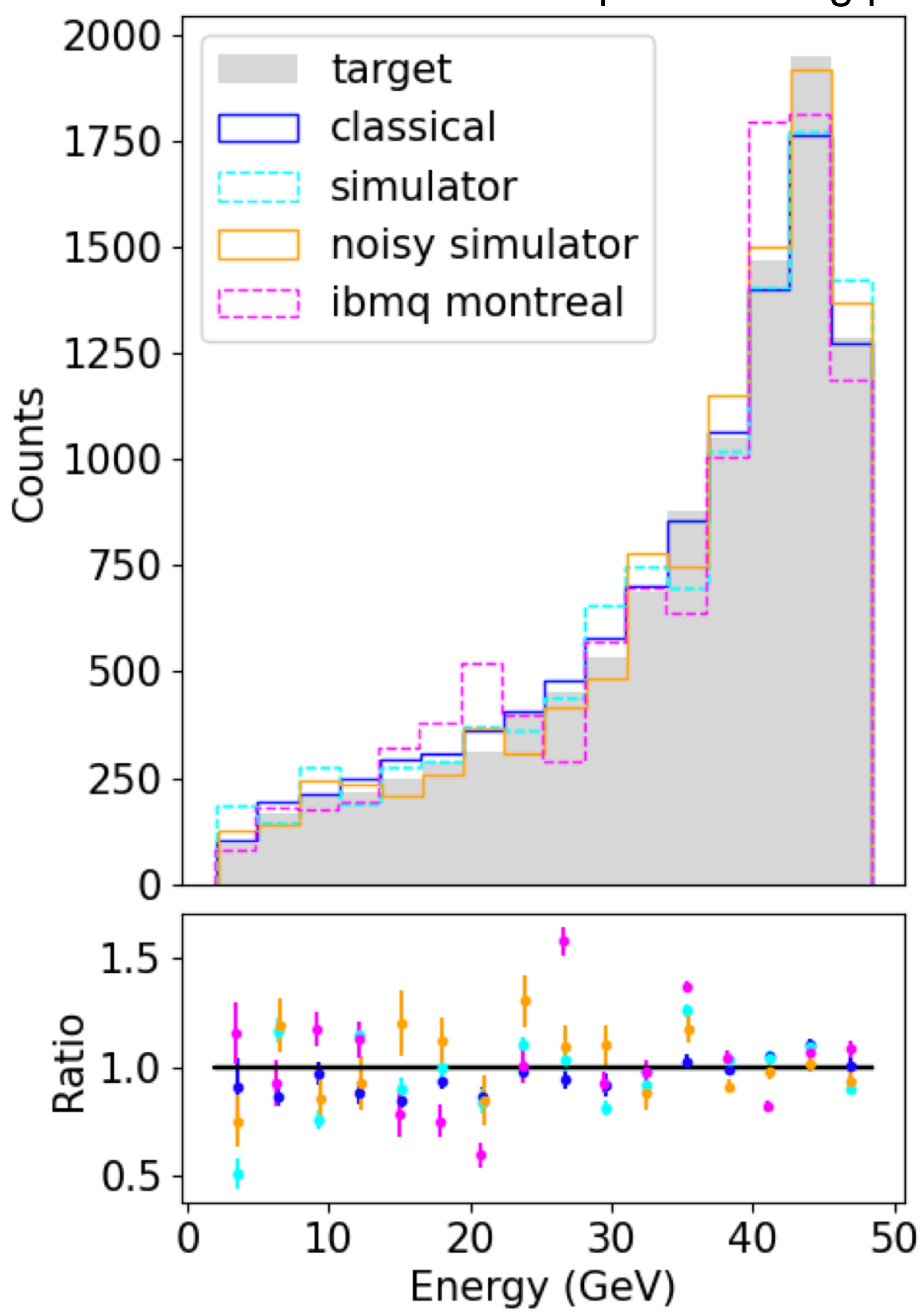
- Variational Quantum Circuits are trained to obtain the best compatibility with respect to the original dataset. **The initial state has a negligible impact**.

Conditional Born Machines: conditions are given in input to the circuit



$$p(x|y)$$

https://arxiv.org/pdf/2205.07674.pdf

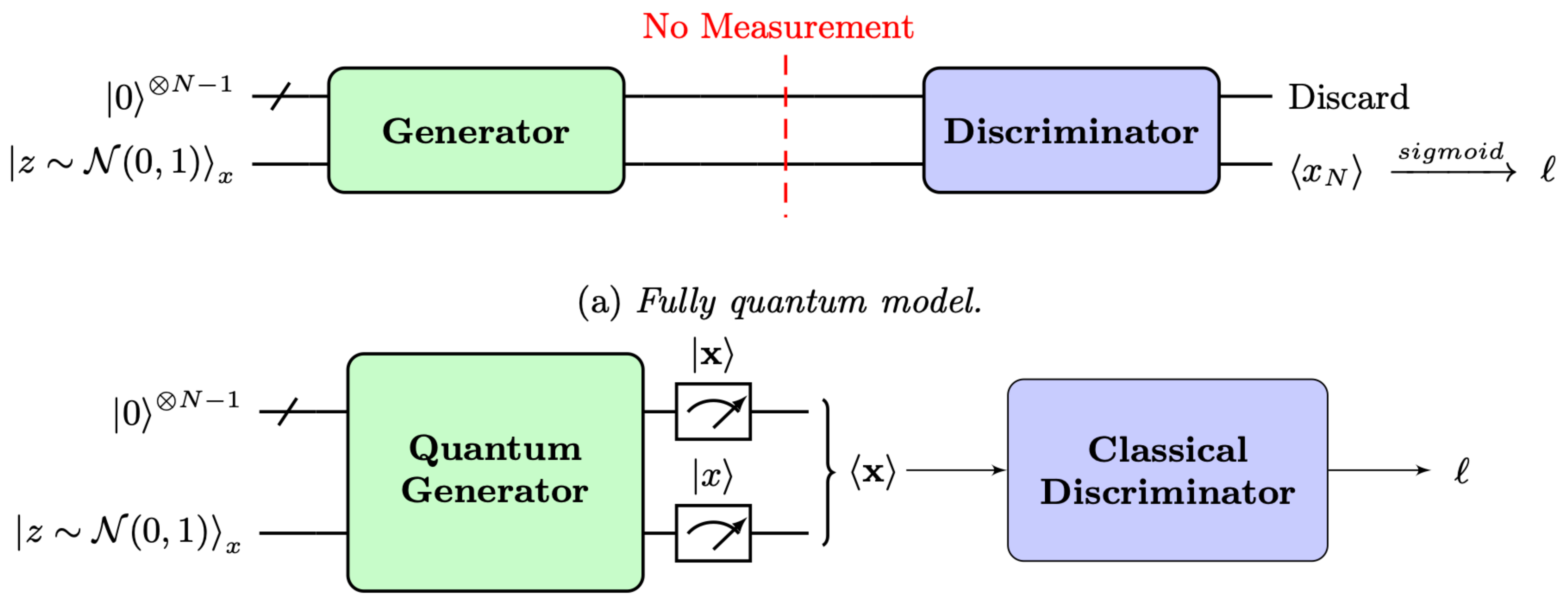


Example: Muonic Force Carriers energy distribution

QCBM are pretty stable and reliable, but many qubits are needed for multi-dimensional simulations

Generative QML: qGAN

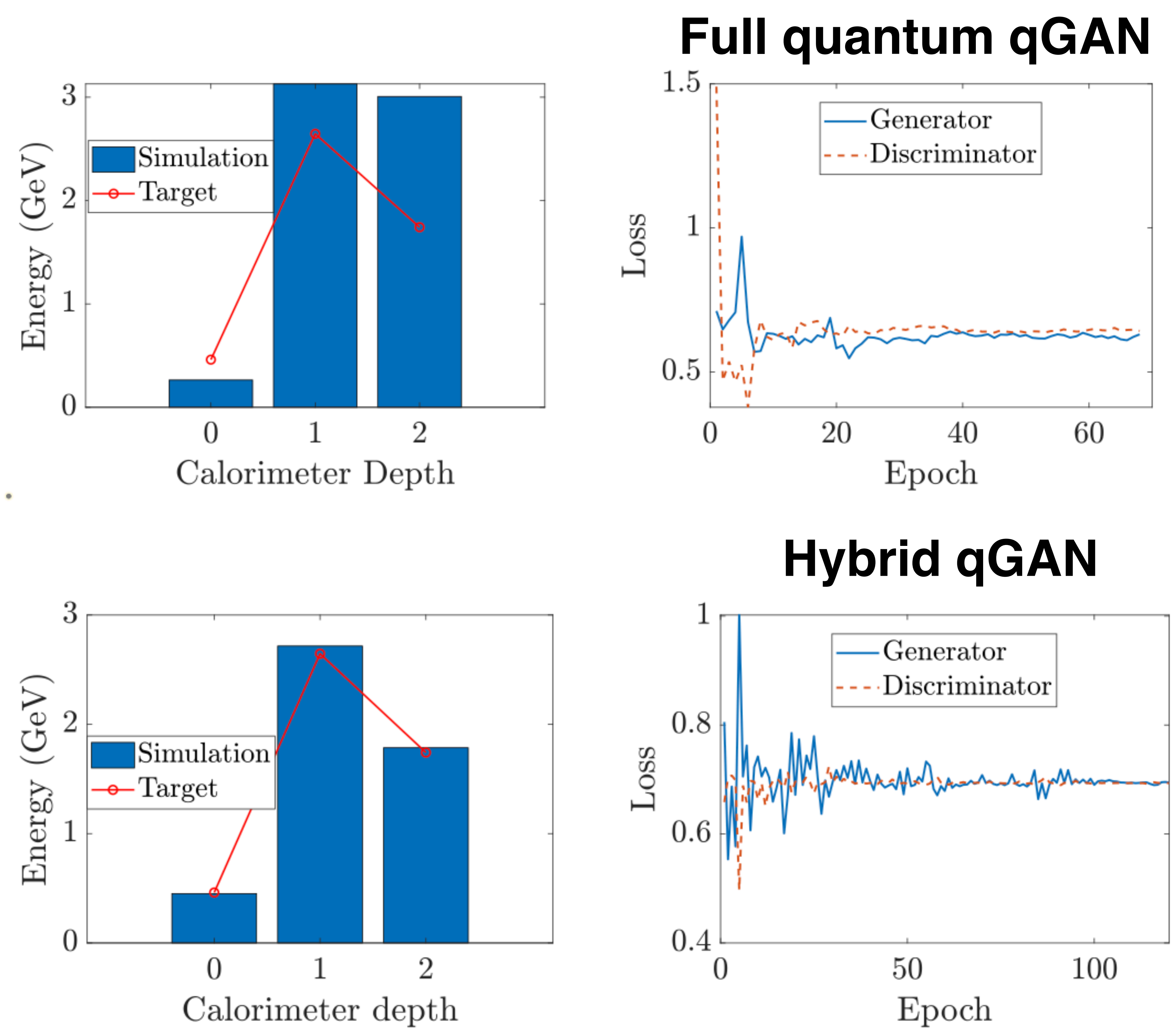
- **Quantum Generative Adversarial Networks**: a quantum generator is trained against a discriminator (classical or quantum)
- In general, GAN (not only qGAN) could replace time-consuming program as Geant4
- With qGAN, N qubits can be used to simulate 2^N features (NOT 2^N configurations as in Born Machines)
- The problem is the stability and convergence: it is useful to increase the latent space dimension, e.g. adding ancillary qubits



(a) *Fully quantum model.*

(b) *Hybrid model.*

https://arxiv.org/pdf/2101.11132.pdf



Possible future prospects

Prospects: entanglement and correlations

- **Quantum circuits could give us more information on data than classical machine learning** by measuring **entanglement entropy**

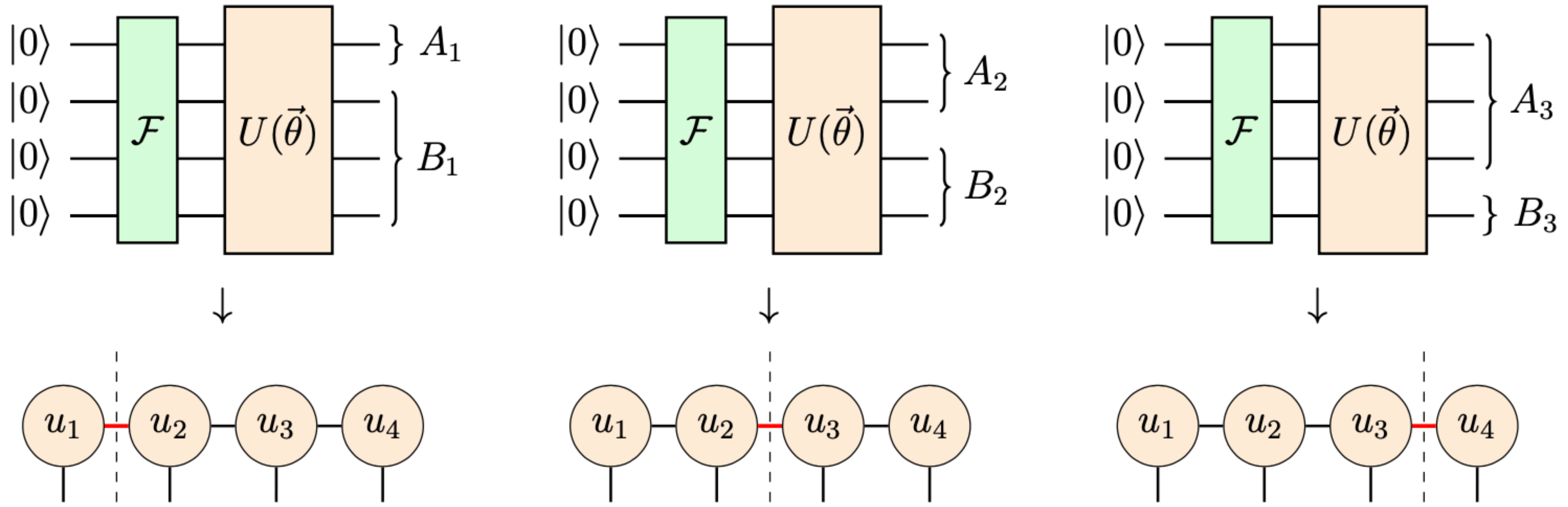
- **Benchmarking**: the entropy is correlated with its expressibility and can be used to **optimize the circuit**: choice of circuit design, embedding scheme, cost function and data preprocessing

- **Entanglement-based models**: the circuit can be trained to obtain characteristic wave-functions of the two categories. **Measurement of entanglement entropy can be used to determine meaningful quantities**, **like feature importance and correlations**

Von Neumann entropy between quantum bipartitions A and B. ρ_A is the reduced density matrix of A, obtained by tracing out the degrees of freedom of B

$$S(\rho_A) = -\mathrm{Tr}(\rho_A \log(\rho_A))$$

Definition of bipartition in a 4-qubit circuit



from S. Monaco master thesis

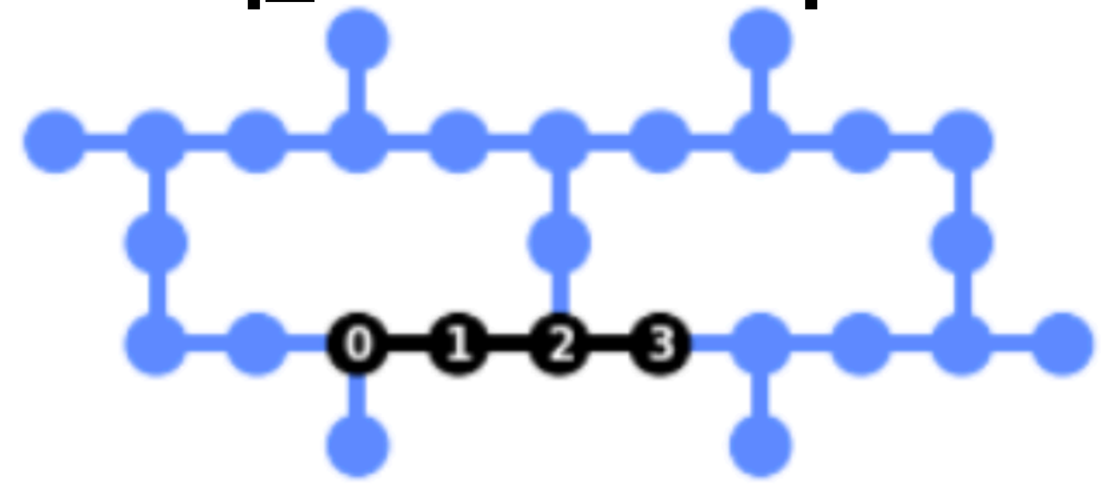
Prospects: circuit optimization

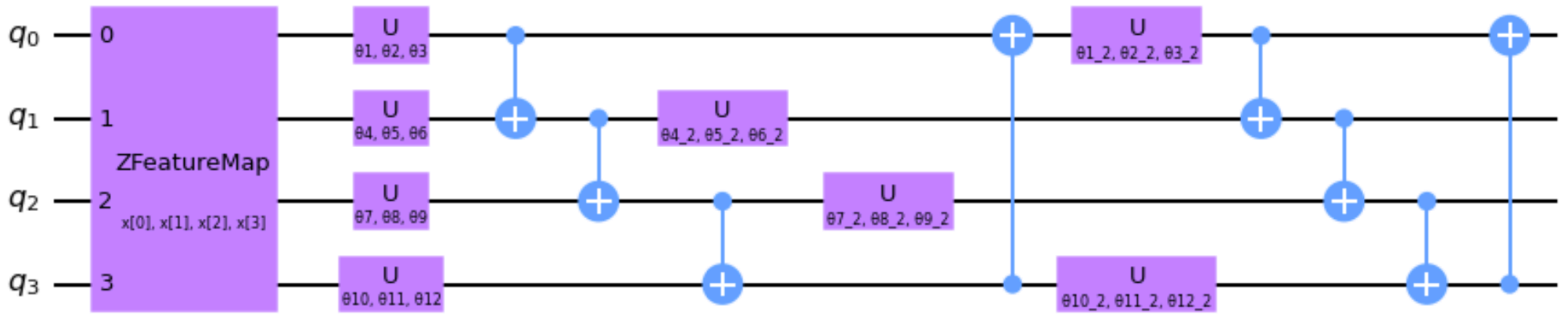
- **When circuits are ported to the hardware, they look very different from the original design**: the implementation depends on the qubit connections, geometry and native gates

- The optimization is done with the **transpiler**

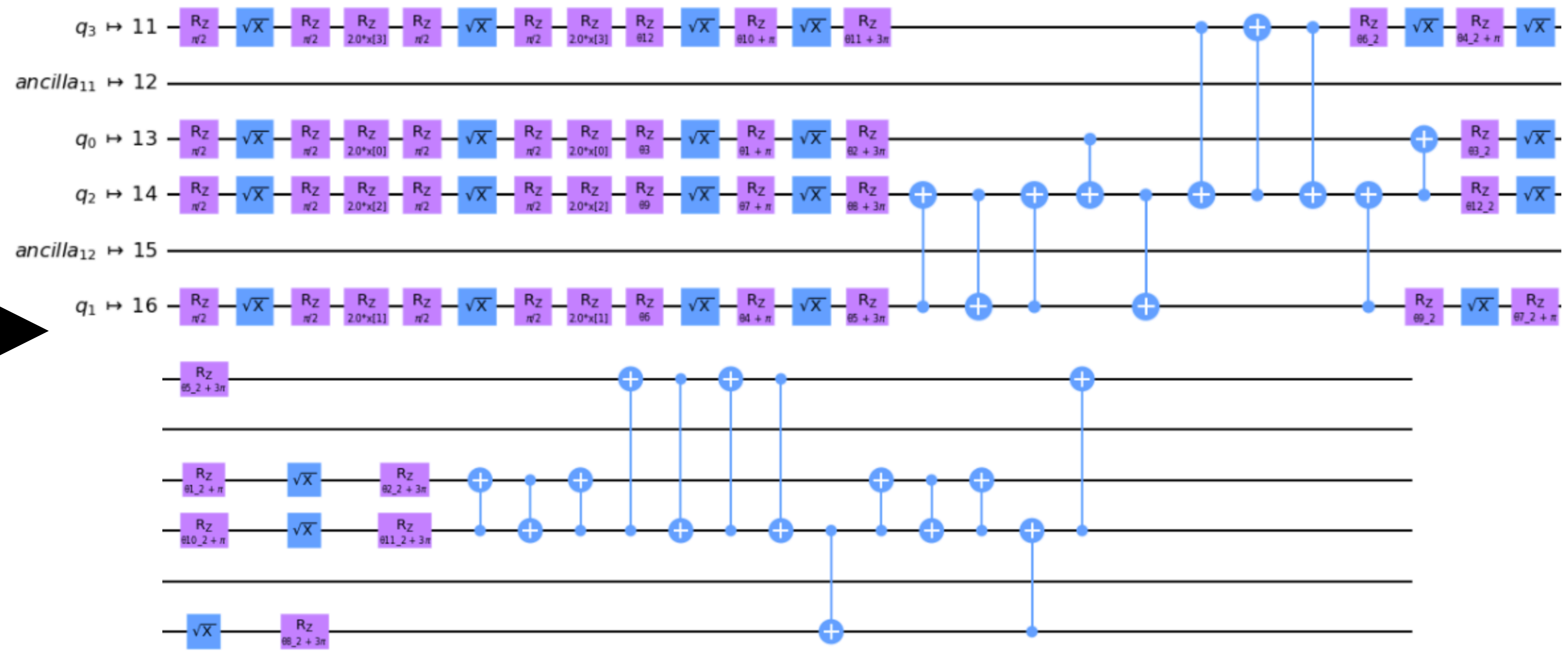
- However we should try to perform an accurate circuit design to improve the **timing performance, impact of the noise** etc.

ibmq_toronto 27 qubits





4-qubit angle embedding circuit



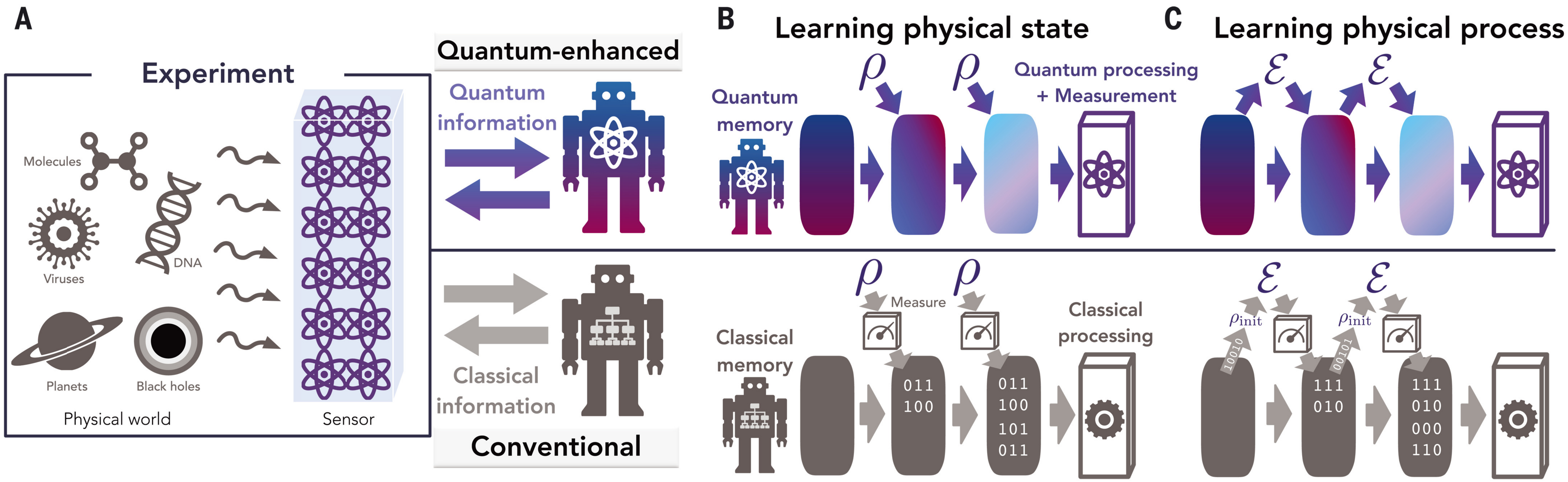
Same circuit on the ibmq_toronto hardware

Prospects: quantum data

- Treatment of classical data with QML is not yet clear

- Analyze **quantum data** with QML could lead to a real advantage (e.g. quantum sensors in the long term)

Science VOL. 376, NO. 6598



Conclusions

- The number of QC and QML applications in HEP is rapidly increasing
- **A real quantum advantage over classical algorithm is not yet established**
- We are at in the R&D phase, but **performance comparable to classical algorithms are already achievable**
- The availability of quantum computers, the number of qubits are currently limitation factors, simulators are not efficient with a high number of qubits
- The prospects on quantum hardware from the industries look promising
- **Many research directions**: data embedding, entropy, circuit optimization etc.

Thanks for your attention!