Quantum Machine Learning

Introduction and examples from HEP

QUANTUM
TECHNOLOGY
INITIATIVE

Michele Grossi (CERN QTI) , Carla Rieger (CERN QTI)

Sofia Vallecorsa (CERN QTI Coordinator)

Agenda

- Part 1: A brief introduction to Quantum Computing
- Part 2: Quantum Computing for Machine Learning
- Part 3: Quantum Machine Learning for HEP

Quantum Computing

[…] "Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical…" [1]

[1] Richard P. Feynman, Department of Physics, California Institute of Technology, International Journal of Theoretical Physics, Vol 21, Nos. 6/7, 1982

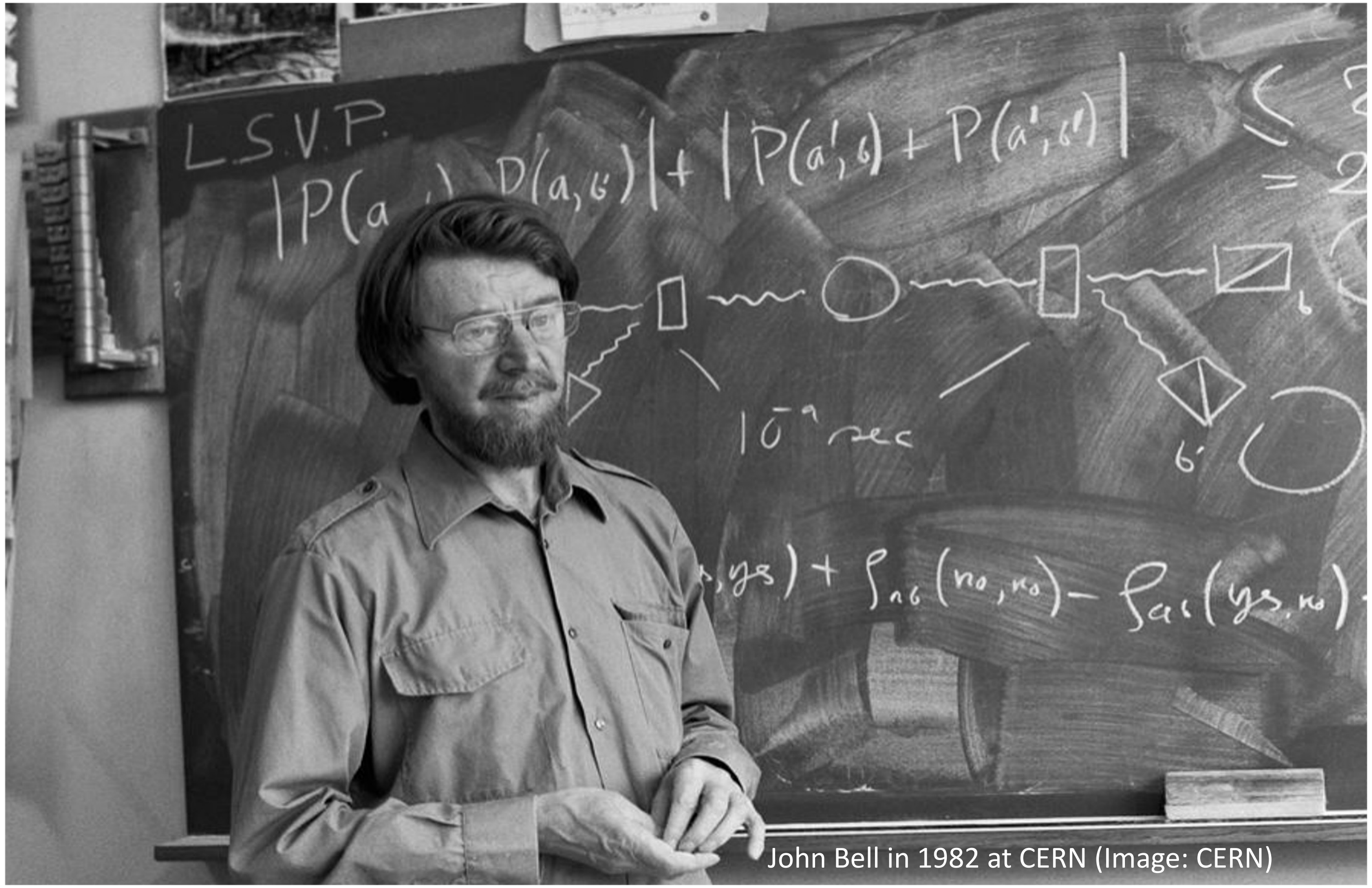
Classical Computation

- **Based on classical binary logic**
- Reached incredibly peaks since late 40s
  - Many problems still can not be addressed adequately

Quantum Computation

- **New frontier of computation**
- Started in early 80s
- First prototypal QC available since 2010s
  - Still in NISQ (Noisy Intermediate Scale Quantum) era

The Second Quantum Revolution and Quantum Computing



John Bell in 1982 at CERN (Image: CERN)

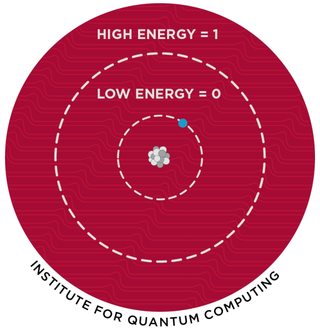
- **Quantum Mechanic principles** are exploited to develop new technology
- Create **"artificial" quantum states** for a range of applications (single photons, trapped ions, superconductors, etc.)

- 1964: **Bell inequalities** prove that no theory based on **local hidden variables** (realism) can reproduce QM results
- Major step confirming the possibility of using distant entangled photons as a quantum information resource

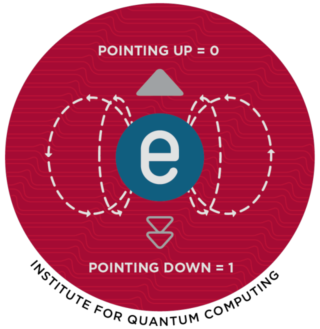
→ **Quantum Computing**

Multiple technologies

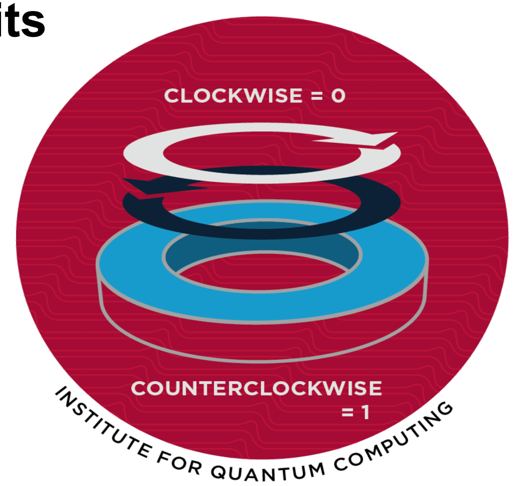
Trapped Ion and Atoms



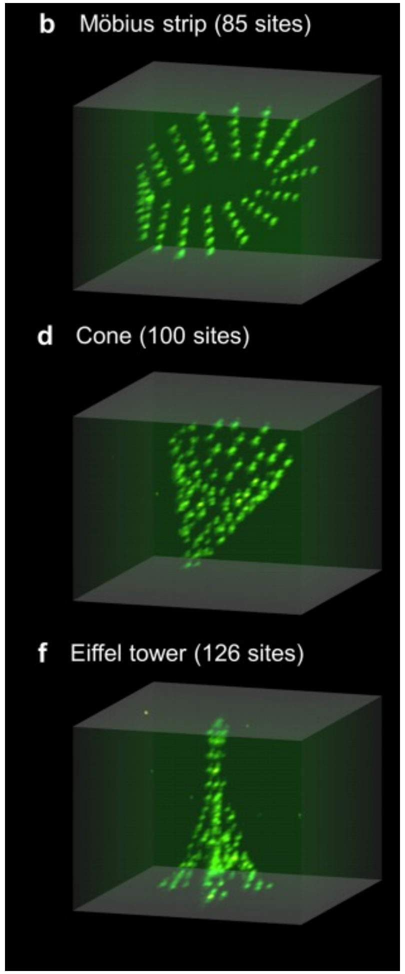
Spin Qubit



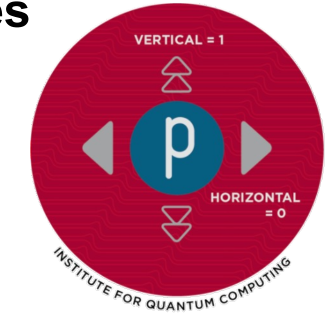
Superconducting Qubits



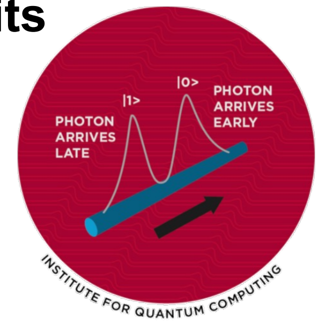
PASQAL



Polarization States



Time Qubits



Path Qubits

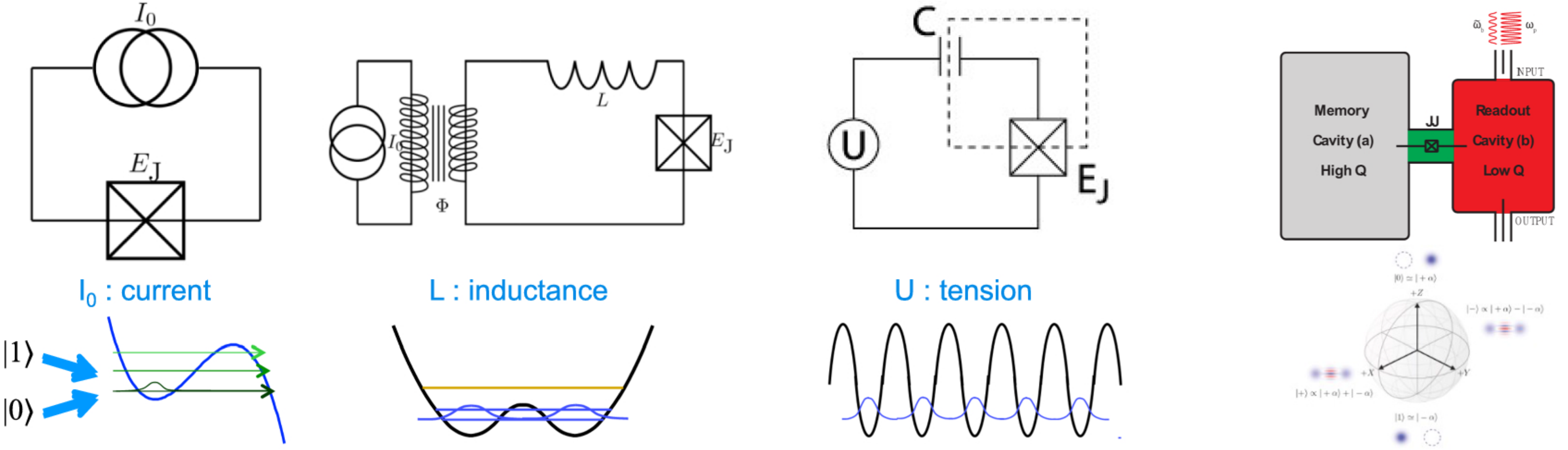
PHOTONS

See Institute of Quantum Computing, U. of Waterloo, https://uwaterloo.ca/institute-for-quantum-computing/quantum-101/quantum-information-science-and-technology/what-qubit

Superconducting Qubits

Ezratty, O. **Perspective on superconducting qubit quantum computing**. Eur. Phys. J. A 59, 94 (2023). https://doi.org/10.1140/epja/s10050-023-01006-7

phase qubit	flux qubit	charge qubit - transmon	cat-qubits



	phase qubit	flux qubit	charge qubit - transmon	cat-qubits
	Josephson junctions handle the qubit degree of liberty			Josephson junctions prepare, couple and correct the cat-qubits
|0⟩ and |1⟩ qubits	two energy levels in a potential well	two superconducting current directions	two levels of charge of Cooper pairs	pairs of entangled microwave photons in a cavity
quantum gates	micro-waves	magnetic field	micro-waves	micro-waves
qubits readout	resonator and micro-waves	magnetometer (SQUID)	resonator and micro-waves	resonator and micro-waves
commercial vendors	**abandonned**	D-Wave The Quantum Computing Company, rigetti, Qilimanjaro Quantum Tech, Alibaba, bleximo, Atlantic Quantum	IBM, Google, IQM, OQC	ALICE & BOB, qci, amazon

Development Roadmap

IBM **Quantum**

	2016–2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2033+
	Run quantum circuits on the IBM Quantum Platform	Release multi-dimensional roadmap publicly with initial aim focused on scaling	Enhancing quantum execution speed by 100x with Qiskit Runtime	Bring dynamic circuits to unlock more computations	Enhancing quantum execution speed by 5x with quantum serverless and Execution modes	Improving quantum circuit quality and speed to allow 5K gates with parametric circuits	Enhancing quantum execution speed and parallelization with partitioning and quantum modularity	Improving quantum circuit quality to allow 7.5K gates	Improving quantum circuit quality to allow 10K gates	Improving quantum circuit quality to allow 15K gates	Improving quantum circuit quality to allow 100M gates	Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing
Data Scientist						Platform						
						Code assistant	Functions	Mapping Collection	Specific Libraries			General purpose QC libraries
Researchers					Middleware							
					Quantum Serverless	Transpiler Service	Resource Management	Circuit Knitting x P	Intelligent Orchestration			Circuit libraries
Quantum Physicist			Qiskit Runtime									
	IBM Quantum Experience		QASM3	Dynamic circuits	Execution Modes	Heron (5K) Error Mitigation 5k gates 133 qubits Classical modular 133x3 = 399 qubits	Flamingo (5K) Error Mitigation 5k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (7.5K) Error Mitigation 7.5k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (10K) Error Mitigation 10k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Flamingo (15K) Error Mitigation 15k gates 156 qubits Quantum modular 156x7 = 1092 qubits	Starling (100M) Error correction 100M gates 200 qubits Error corrected modularity	Blue Jay (1B) Error correction 1B gates 2000 qubits Error corrected modularity
	Early Canary 5 qubits Albatross 16 qubits Penguin 20 qubits Prototype 53 qubits	Falcon Benchmarking 27 qubits		Eagle Benchmarking 127 qubits								

Innovation Roadmap

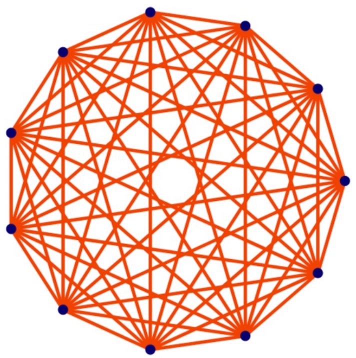
	2016–2019		2020	2021	2022	2023	2024	2025	2026	2027	2028
Software Innovation	IBM Quantum Experience	Qiskit Circuit and operator API with compilation to multiple targets	Application modules Modules for domain specific application and algorithm workflows	Qiskit Runtime Performance and abstract through Primitives	Serverless Demonstrate concepts of quantum centric-supercomputing	AI enhanced quantum Prototype demonstrations of AI enhanced circuit transpilation	Resource management System partitioning to enable parallel execution	Scalable circuit knitting Circuit partitioning with classical reconstruction at HPC scale	Error correction decoder Demonstration of a quantum system with real-time error correction decoder		
Hardware Innovation	Early Canary 5 qubits Penguin 20 qubits Albatross 16 qubits Prototype 53 qubits	Falcon Demonstrate scaling with I/O routing with Bump bonds	Hummingbird Demonstrate scaling with multiplexing readout	Eagle Demonstrate scaling with MLW and TSV	Osprey Enabling scaling with high density signal delivery	Condor Single system scaling and fridge capacity	Flamingo Demonstrate scaling with modular connectors	Kookaburra Demonstrate scaling with nonlocal c-coupler	Demonstrate path to improved quality with logical memory	Cockatoo Demonstrate path to improved quality with logical communication	Starling Demonstrate path to improved quality with logical gates
						Heron Architecture based on tunable-couplers	Crossbill m- coupler				

Executed by IBM

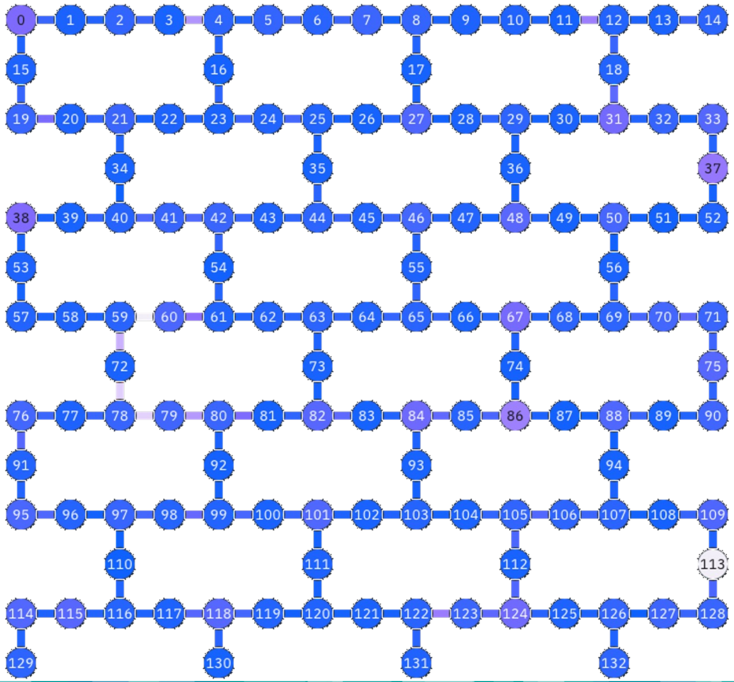
On target

Noisy Intermediate-Scale Quantum devices

Trapped ion technology: *ionQ*
with all-to-all connectivity



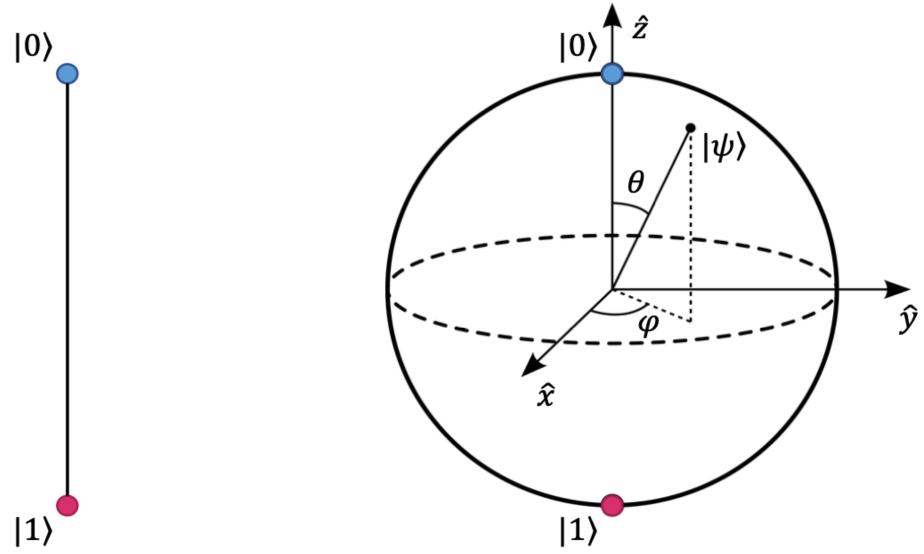
- Limitations in terms of **stability** and **connectivity**
  - **Circuit optimisation**
- **De-coherence**, measurement errors or gate level errors (**noise**)
  - Specific **error mitigation techniques**
  - Prefer algorithms **robust against noise**
- **Problem size**
- Initially integrated in **hybrid quantum-classical infrastructure (HPC + QC)**
  - **Quantum Processing Units** as new "hardware accelerators"



Superconducting qubits:
IBM Torino – 133 qubits

Quantum Information Theory

Unit of information



bit **qubit**

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right)e^{i\gamma}$$

where $\alpha, \beta \in \mathbb{C}$ and $\theta, \phi, \gamma \in \mathbb{R}$

Quantum logic gates

- Single qubit operations
  - **Hadamard gate:** creation of superposition $\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
  - **Pauli gates:** π rotations along main axes

$$\boldsymbol{\sigma_x} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \boldsymbol{\sigma_y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \boldsymbol{\sigma_z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Two-qubit operations

$$\boldsymbol{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\boldsymbol{C - \varphi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{-i\varphi} \end{pmatrix}$$

creation of entanglement

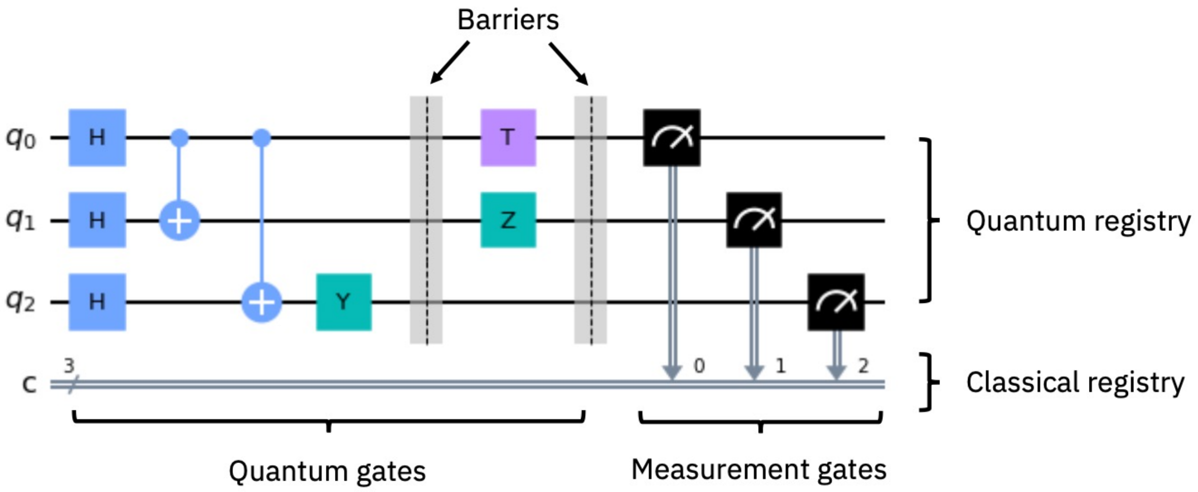
- Generic multi-qubit operations: decomposed in single-qubit and two-qubit gates

- **Universal gate sets**

Quantum Information Theory

Composing quantum gates: quantum circuits

- Set of actions to be performed to the selected qubits
  - qubits initialization
  - single-qubit gates, multi-qubit gates
  - measurements



Principles of quantum computation

- Quantum algorithm: set of quantum circuits performing certain task
  - Purely quantum, e.g. *Shor*
  - Hybrid classical-quantum, e.g. *VQE*
- Quantum Simulation: simulation of time evolution of quantum system
  - Analog Simulator
  - Digital Simulator: quantum logic gates, more flexible

Quantum Computing

→ Do **classically intractable** computations **efficiently** on a Quantum Computer leveraging Quantum Effects

Problems for a quantum computer

Modeling nature at the quantum level

Mathematics

Quantum chemistry,
Material science and
High energy physics

Algebraic algorithms with provable speed-ups

- Shor's Algorithm
- Grover's Algorithm

Empirical heuristic algorithms

- Optimization
- Machine learning

Quantum potential… and computer science

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

Can enable speed-up though **highly parallel** computations

Quantum Superposition State

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Also, **non-classical correlations** may speed-up computations

Quantum Entanglement
(here: Bell state)

Operations (gates) are unitary transformations → **reversible computing?**

Output is the result of a measurement according to Born rule → **stochastic computation ?**

No-cloning theorem → information security

Quantum state coherence and isolation → computation stability and errors

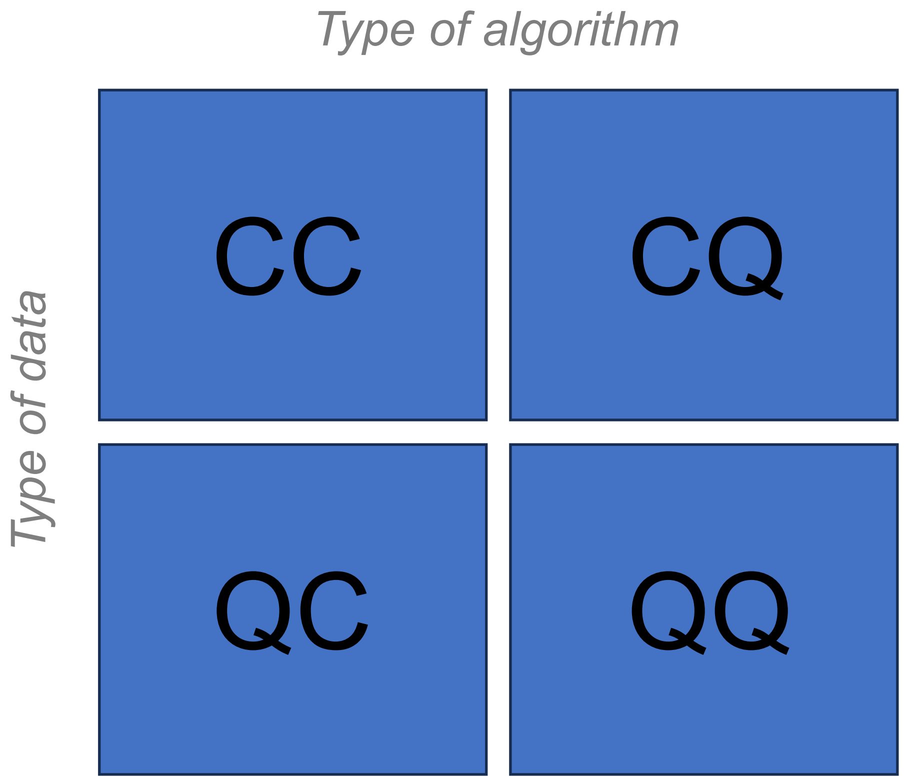
Qubit state collapses → reproducibility?

Agenda

- Part 1: A brief introduction to Quantum Computing
- Part 2: Quantum Computing for Machine Learning
- Part 3: Quantum Machine Learning for HEP

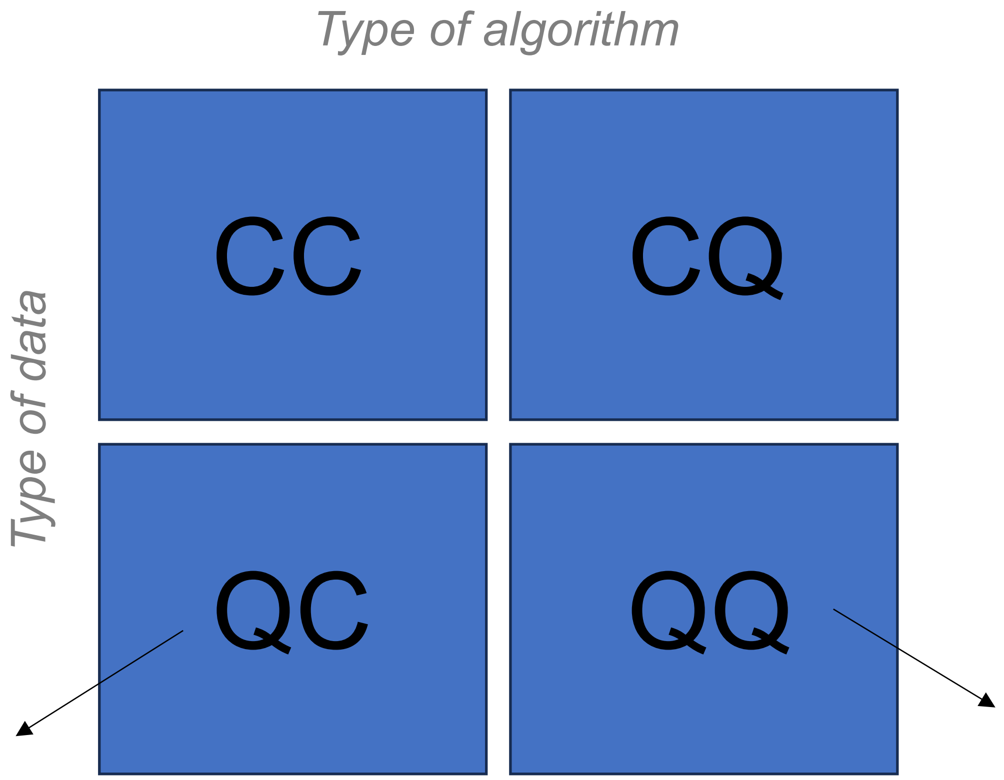
What is Quantum Machine Learning?

Fields in Quantum Computing



Source: Qiskit Textbook

Fields in Quantum Computing



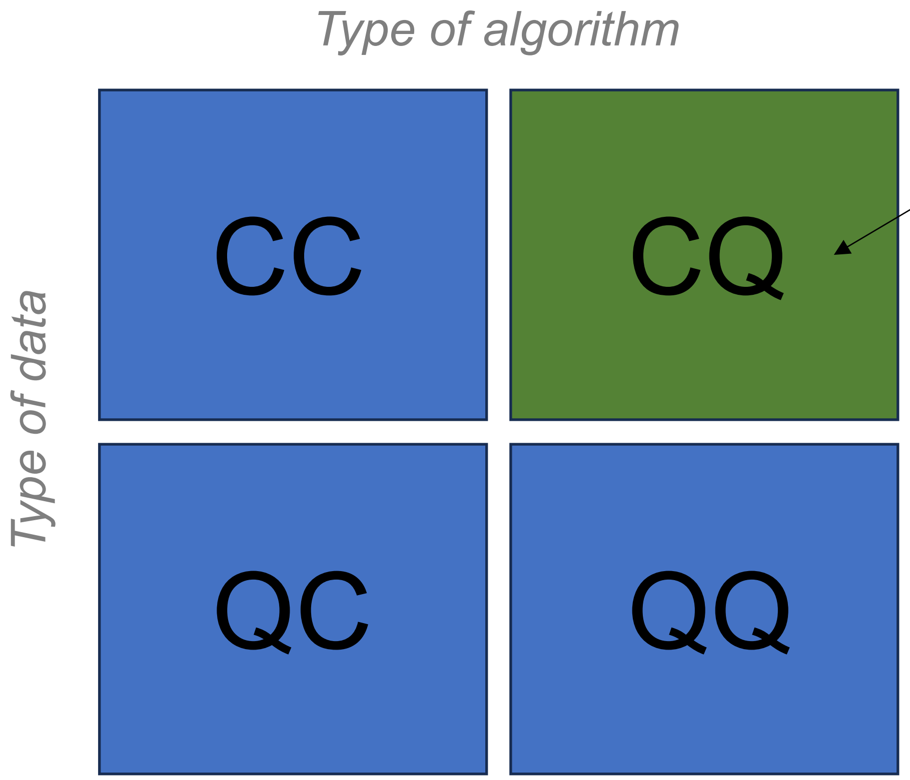
Simulation of Quantum Systems using classical ML

Simulation of Quantum Systems using a Quantum Computer

Source: Qiskit Textbook

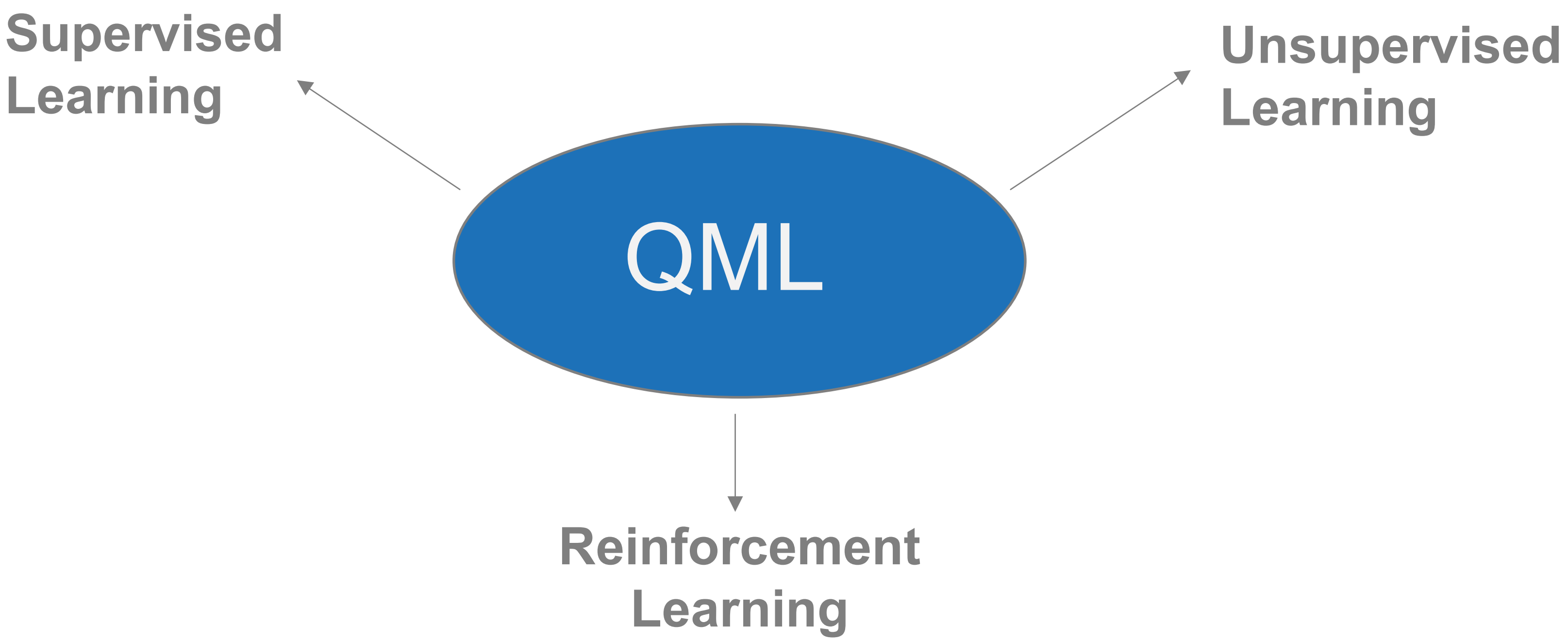
Fields in Quantum Computing

Here: **focus on quantum algorithm with classical input data**



Source: Qiskit Textbook

Fields in Quantum Machine Learning (QML)



Source: Qiskit Textbook

Fields in Quantum Machine Learning (QML)

QML

Supervised Learning

e.g., **Quantum Classifier**: aims to learn *input-output relation* of labeled dataset $f: x_{in} \mapsto x_{out}$ by optimizing quantum network

Unsupervised Learning

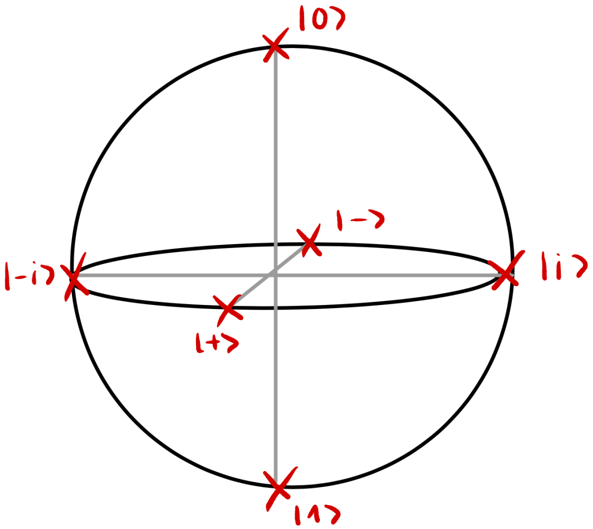
e.g., **Quantum Generative Models**: aims to *learn the underlying probability distribution* $\pi(y)$ of a given data set and *generates samples* from it using quantum network

Reinforcement Learning

e.g., **Quantum Reinforcement Learning**: find *policy* for agent that *maximizes reward* (expected reward computed using QC)

Source: Qiskit Textbook

What is Quantum Advantage in QML?



Bloch sphere: only the marked points are produced by the Clifford operators acting on a computational basis state

Multiple considerations:

1. Runtime speed-up
2. Sample complexity
3. Representational power

This includes considerations regarding **classical intractability**:

Focus on Quantum Circuits that are **not efficiently simulable classically**

Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." (2002).
Gottesman, Daniel. "The Heisenberg representation of quantum computers." *arXiv preprint quant-ph/9807006* (1998).
See also: - Kübler, Jonas, Simon Buchholz, and Bernhard Schölkopf. "The inductive bias of quantum kernels." *Advances in Neural Information Processing Systems* 34 (2021): 12661-12673.
- Huang, HY., Broughton, M., Mohseni, M. *et al.* Power of data in quantum machine learning. *Nat Commun* **12**, 2631 (2021). https://doi.org/10.1038/s41467-021-22539-9

Quantum Machine Learning Lyfecycle

The quantum advantage of many known QML algorithms is impeded by an input or output bottleneck

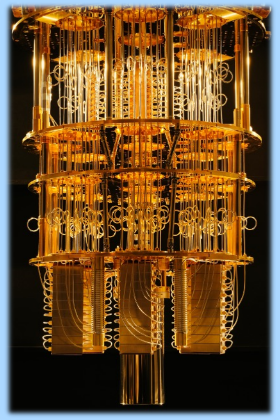
Data Preparation

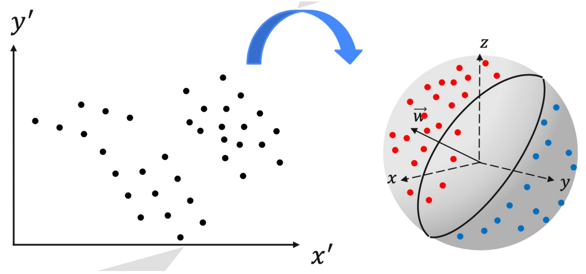
Model Definition

Model Training

Model Testing

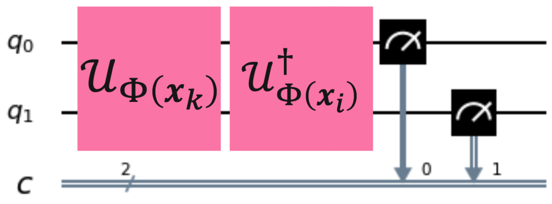
Model Interpretation



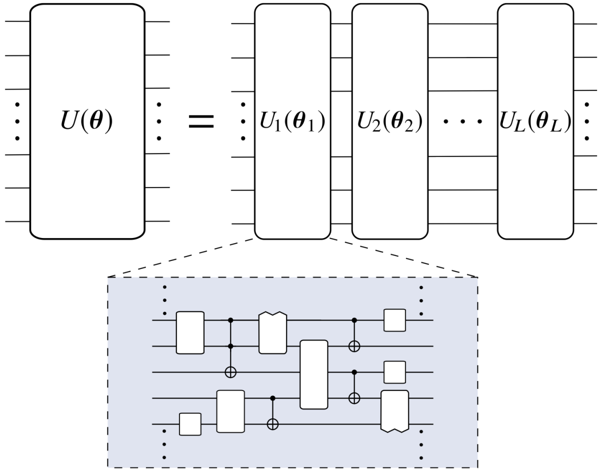


Data Reduction
Data Encoding [1,2,3]

Read Out



Trainability (BP…)



[1] Robust data encodings for quantum classifiers, Ryan LaRose and Brian Coyle, Phys. Rev. A 102, 032420

[2] Quantum convolutional neural network for classical data classification, https://arxiv.org/pdf/2108.00661.pdf

[3] Quantum Support Vector Machines for Continuum Suppression in B Meson Decays, https://arxiv.org/abs/2103.12257

Models

Gradient-free or **gradient-based** optimization

Data Embedding can be **learned**

Ansatz design can leverage data symmetries[1]

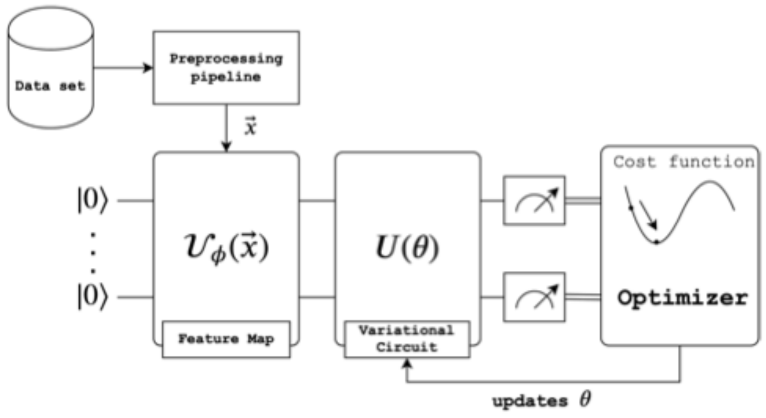


Image credit SwissQuantumHub

Representer theorem:

Implicit models achieve **better accuracy**[3]

Explicit models exhibit **better generalization** performance

Feature maps as **quantum kernels**

Classical **kernel-based training** (**convex** losses)

Identify classes of kernels that relate to specific data **structures**[2]

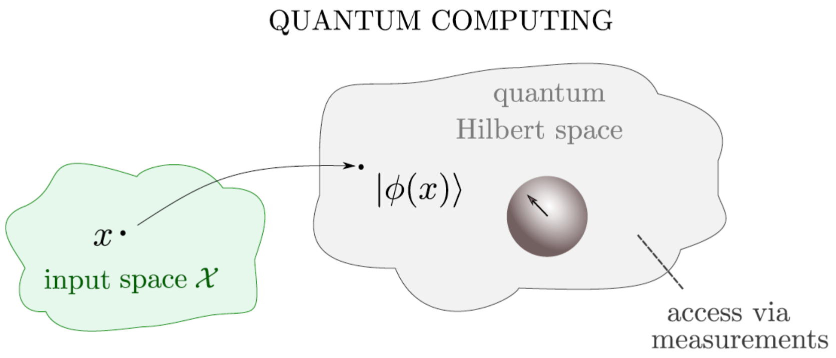


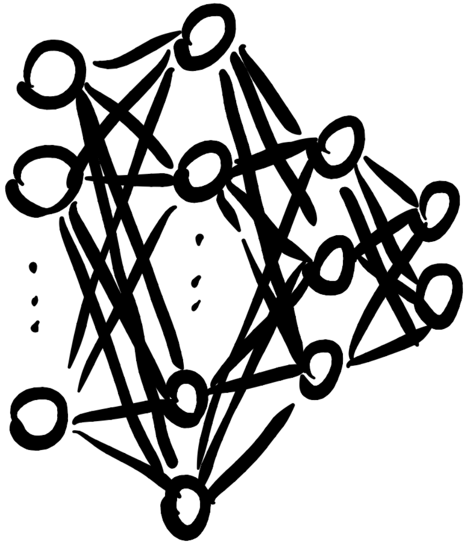
Image credit M. Schuld

Build network of stochastic binary units and optimise their energy.
QBM has quadratic energy function that follows the Boltzman distribution (Ising Hamiltonian)

[1] Bogatskiy, Alexander, et al. "**Lorentz group equivariant neural network for particle physics**." PMLR, 2020.
[2] Glick, Jennifer R., et al. "**Covariant quantum kernels for data with group structure**." *arXiv:2105.03406* (2021).
[3]Jerbi, Sofiene, et al. "**Quantum machine learning beyond kernel methods**." *arXiv:2110.13162* (2021).

Supervised Learning in Quantum Computing: Quantum Classifiers

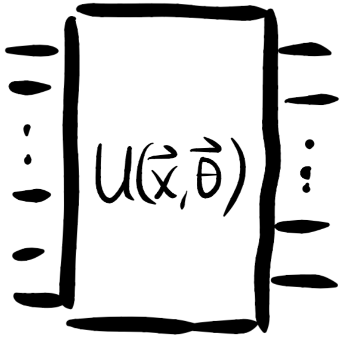
→ Goal: learn input-output relation of labeled data



$$\Psi(\underbrace{\vec{x}}_{\text{input}},\underbrace{\vec{\theta}}_{\text{trainable weights}}): \mathbb{R}^{n_{in}} \rightarrow \mathbb{R}^{n_{out}}$$

Classical Neural Network

$|0\rangle^{\otimes n_{in}}$

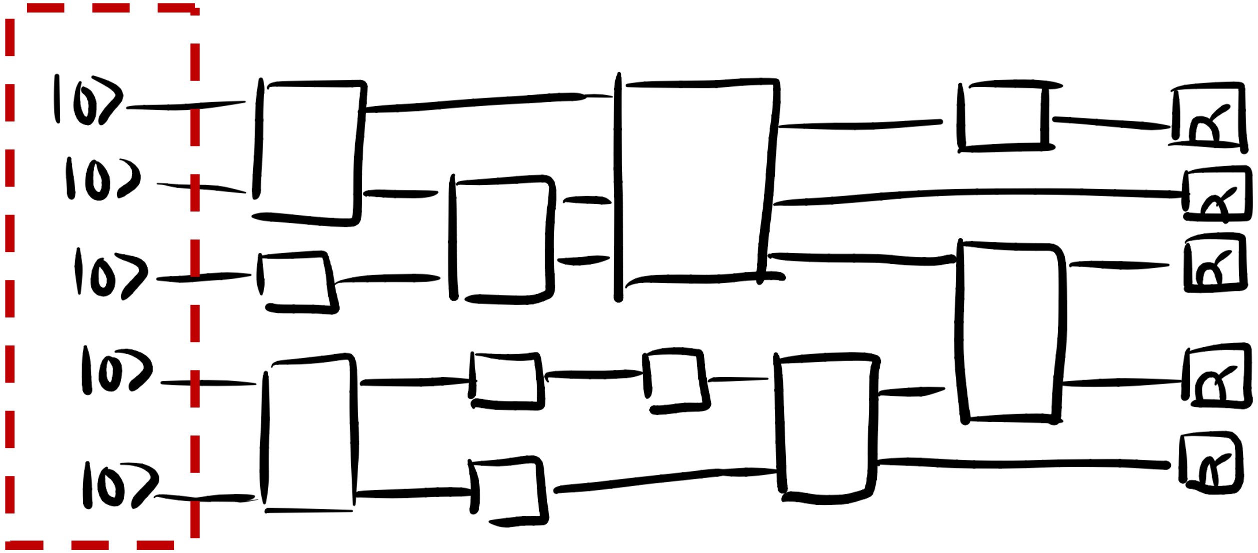


$$|\Psi_{out}(\vec{x},\vec{\theta})\rangle = U(\underbrace{\vec{x}}_{\text{input}},\underbrace{\vec{\theta}}_{\text{trainable weights}})|0\rangle^{\otimes n_{in}}$$

$$y(\vec{x},\vec{\theta}) = \langle \Psi_{out}(\vec{x},\vec{\theta})|\,\hat{O}\,|\Psi_{out}(\vec{x},\vec{\theta})\rangle$$

Parametrized Quantum Circuit

Quantum Circuits and *the Born rule*

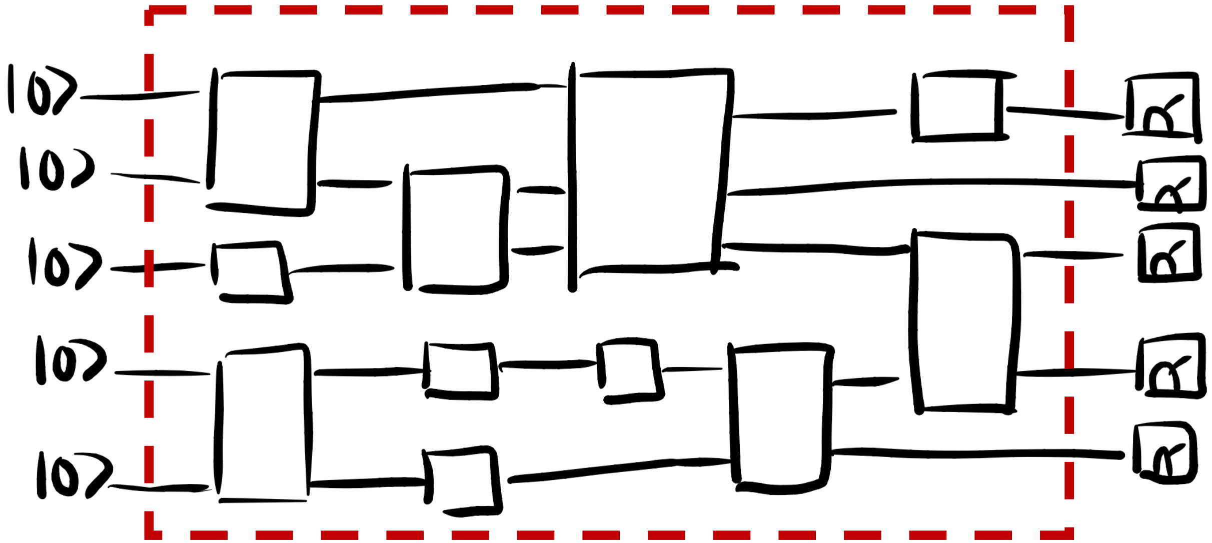


An arbitrary quantum circuit generating the state $|\Psi\rangle$

Initialization:

→initialize qubits in computational basis state

Quantum Circuits and *the Born rule*

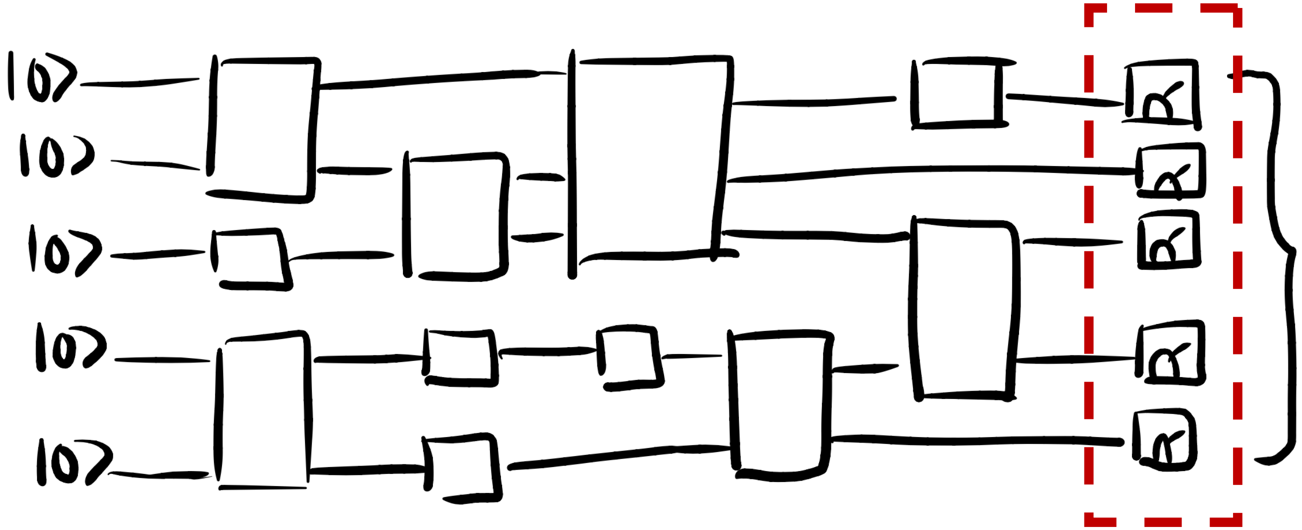


An arbitrary quantum circuit generating the state $|\Psi\rangle$

Evolve initial state:

→Apply set of **unitary** gates that may **encode classical input data x** and include **parametrized gates**

Quantum Circuits and *the Born rule*



An arbitrary quantum circuit generating the state $|\Psi\rangle$

Quantum Measurement

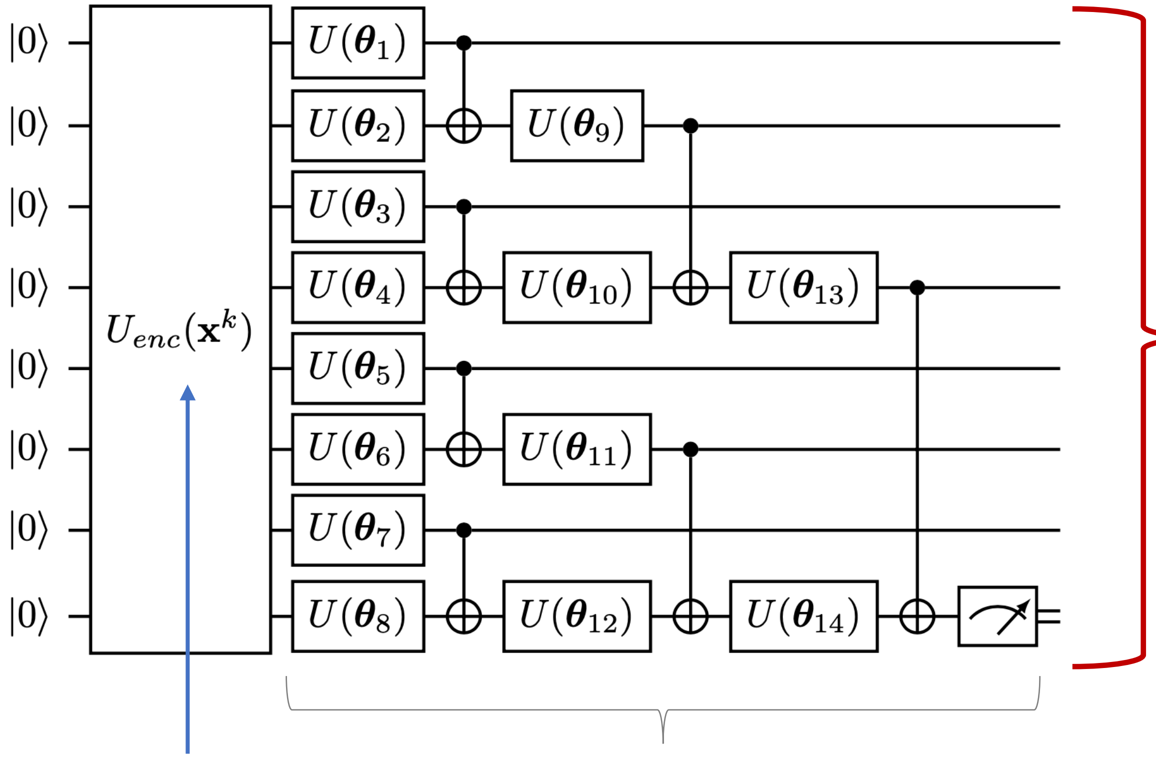
$\longrightarrow$ retrieve a classical output distribution $|\langle x|\Psi\rangle|^2$ of classical output states

(with $x \in \{0,1\}^n$) according to Born rule

Quantum Classifier example: Quantum Tree Tensor Network

Quantum Tree Tensor Network with generic single-qubit unitary gates $U(\theta, \phi, \lambda)$

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{pmatrix}$$



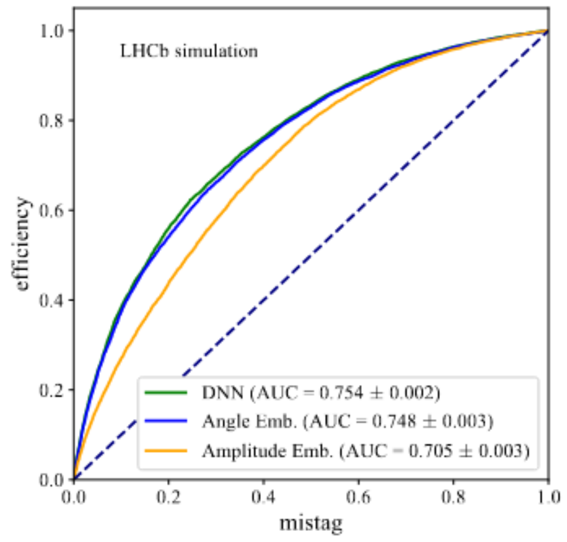
Apply **QTTN as binary classifier**: measure one qubit

We encode our classical input features here

See: Grant, Edward, et al. "Hierarchical quantum classifiers." *npj Quantum Information* 4.1 (2018): 65.

Quantum embedding for classical data

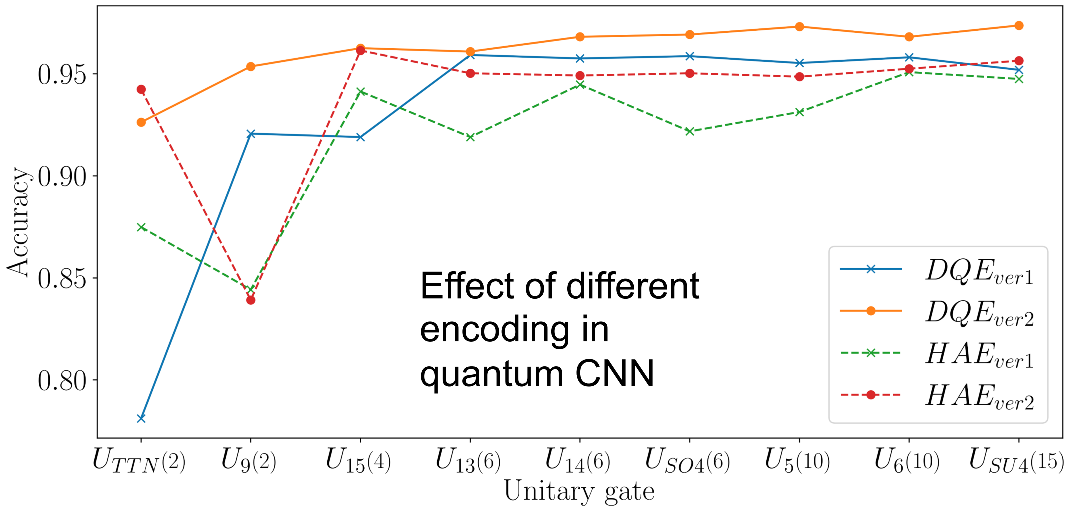
Gianelle, A., Koppenburg, P., Lucchesi, D. *et al.* **Quantum Machine Learning for *b*-jet charge identification**. *J. High Energ. Phys.* **2022,** 14 (2022). https://doi.org/10.1007/JHEP08(2022)014



Compromise between **exponential compression and circuit depth**

Ex: **Amplitude Encoding**

$$|\phi(x)\rangle = \frac{1}{\|x\|}\sum_{i=0}^{N} x_i|i\rangle$$



- Exponential compression
  $n_{qubit} \propto O(\log(N))$
- Polynomial number of gates
  $n_{gate} \propto O(\text{poly}(N))$

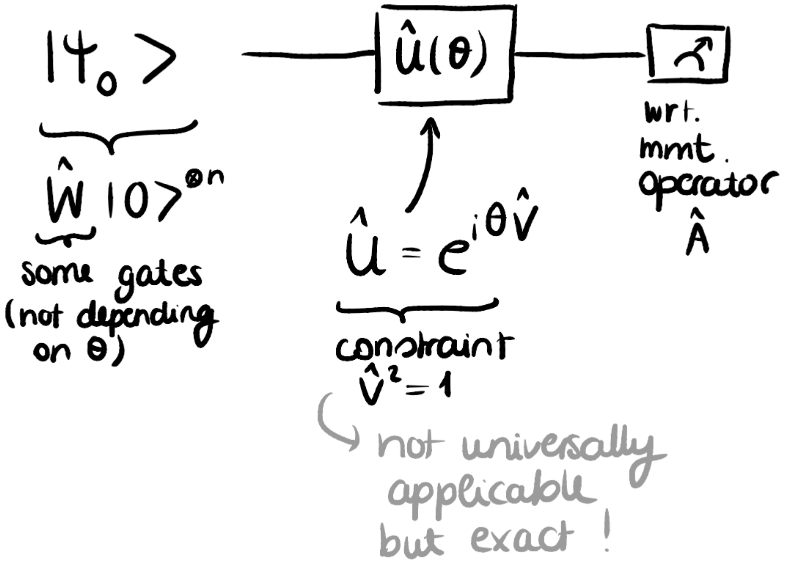
Parameter optimization

$$\theta \to \theta - \eta \nabla_\theta f$$

$$\langle \hat{A}(\theta) \rangle$$

The parameter-shift rule (gradient-based)

→ Compute **partial derivative** of variational circuit parameter θ, alternative to analytical gradient computation and classical finite difference rule (numerical errors and resource cost considerations)



$$\Rightarrow \nabla_\theta \langle \hat{A} \rangle = u\left[\langle \hat{A}(\theta + \tfrac{\pi}{4u}) \rangle - \langle \hat{A}(\theta - \tfrac{\pi}{4u}) \rangle\right]$$

→ Evaluate Quantum Circuit twice at shifted parameters to compute gradient

Source:https://pennylane.ai/qml/demos/tutorial_stochastic_parameter_shift/

Parameter optimization

$$\theta_{k+1} \leftarrow \theta_k - a_k \underbrace{\hat{g}_k(\hat{\theta}_k)}_{\substack{\text{stochastic} \\ \text{estimate} \\ \text{of } \nabla_\theta f}}$$

Simultaneous Perturbation Stochastic Approximation (SPSA)
(gradient-free)

→ If gradient computation not possible, too resource-intensive, or noise-robustness required (slower convergence but fewer function evaluations)
→ Gradient is approximated by two sampling steps and parameters are perturbed in all directions simultaneously

$$\text{"} \; y(\theta) = f(\theta) + \varepsilon \; \text{"}$$

random output perturbation

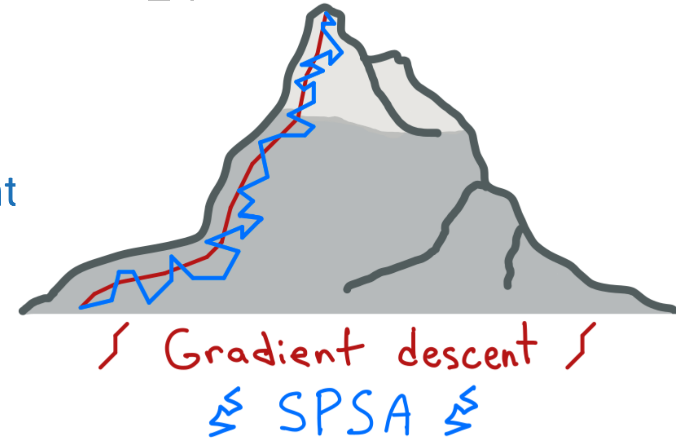
$$\hat{g}_{ki}(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k \Delta_k) - y(\hat{\theta}_k - c_k \Delta_k)}{2 c_k \Delta_{ki}}$$

$$c_k \geq 0, \quad \Delta_k = (\Delta_{k_1}, \Delta_{k_2}, \dots, \Delta_{k_p})^T$$ perturbation vector

(~ randomly sampled from zero-mean distr.)

Iterative update rule comparable to classical stochastic gradient descent

https://pennylane.ai/qml/demos/tutorial_spsa



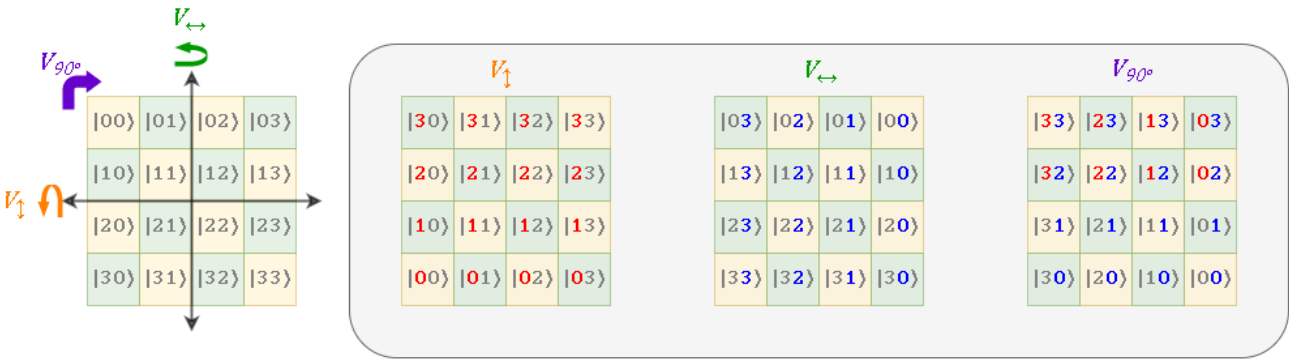
Challenges when using Parametrized Quantum Circuits

- Efficient **data handling** and data **embedding**
- Find balance: **Generalization** and **representational power** vs. **Convergence**
  - Problem of barren plateaus and vanishing gradients in optimization landscape
  - How well can we survey the Hilbert space (expressibility)?

- Current hardware limitations
  - Limited number of qubits and connectivity
  - **Quantum Noise Effects** (decoherence, measurement errors or gate-level errors)
  - Efficient interplay between classical and quantum computer
- ....

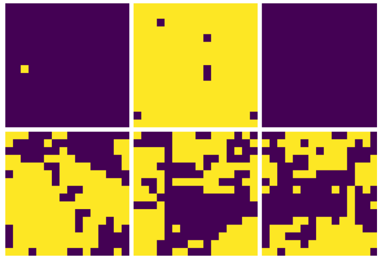
Equivariant Quantum CNN

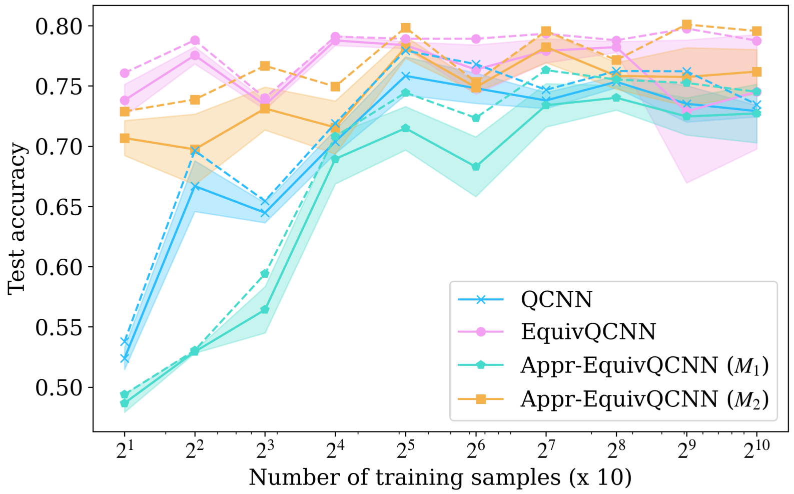
- Construct **equivariant** quantum CNN **under rotational & reflectional symmetry**
- Improved generalization power



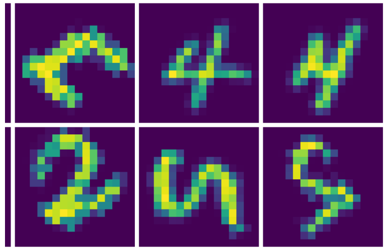
$$\mathcal{H} = -J\sum_{\langle ij\rangle}\sigma_i\sigma_j$$

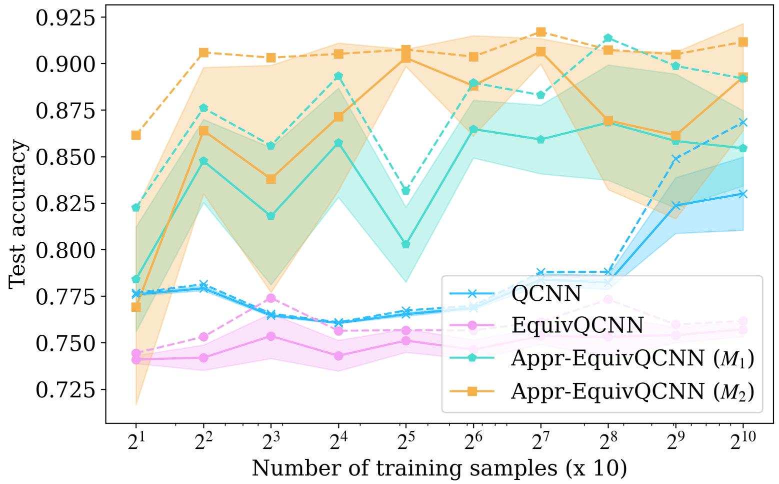
Ising spins phase classification :





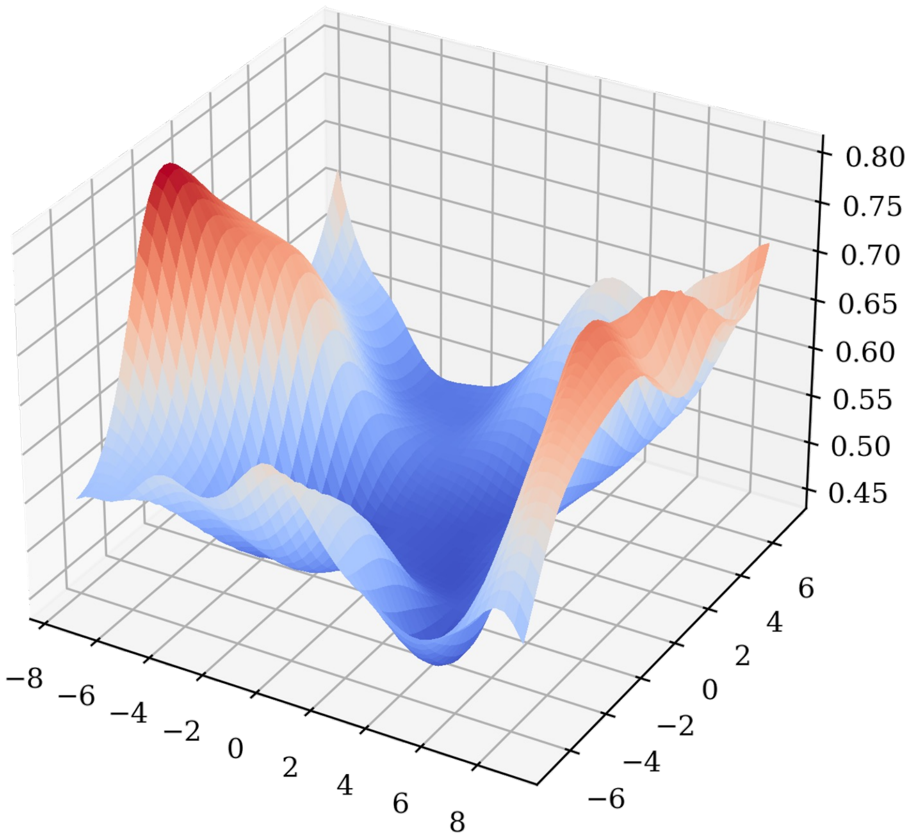
Extended MNIST Image classification: (digits 4,5)



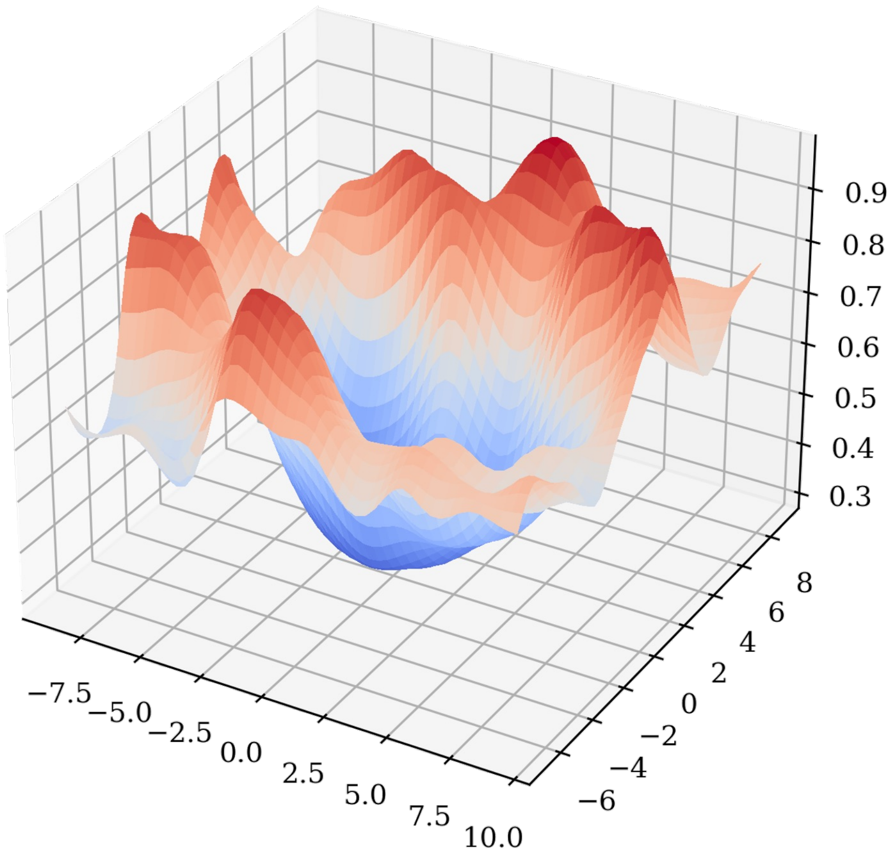


Non-convexity of loss landscape

Loss landscape plotted with orqviz



Non-equivariant QCNN

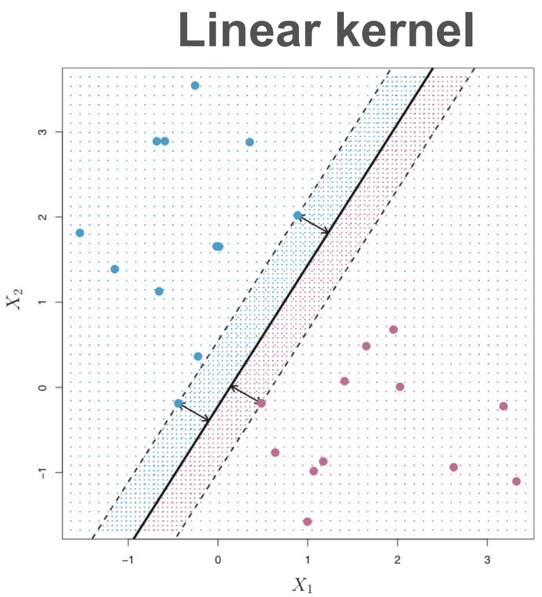


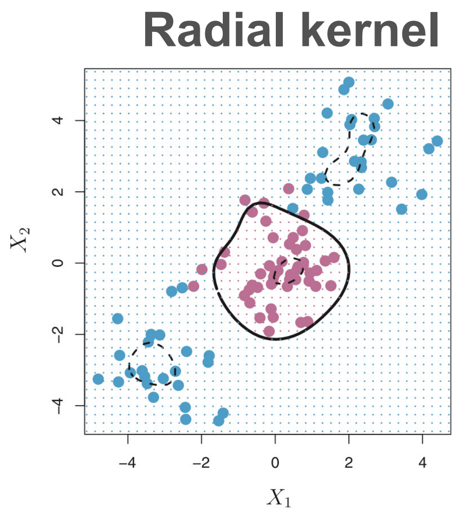
ApprEquivQCNN

Support Vector Machine

Classification problem: find the **hyperplane that better divides data classes**,

defining the prediction as an **inner product** and trying to **maximize the margins**.





$$K(x_i, x_i') = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

- **Polynomial scaling** in training the model $\mathcal{O}(n^3)$ (where n is the number of training data) → improve scalability
- Crucial to **select the right kernel**, but we have a limited set of well studied kernels → help in finding useful kernel functions.

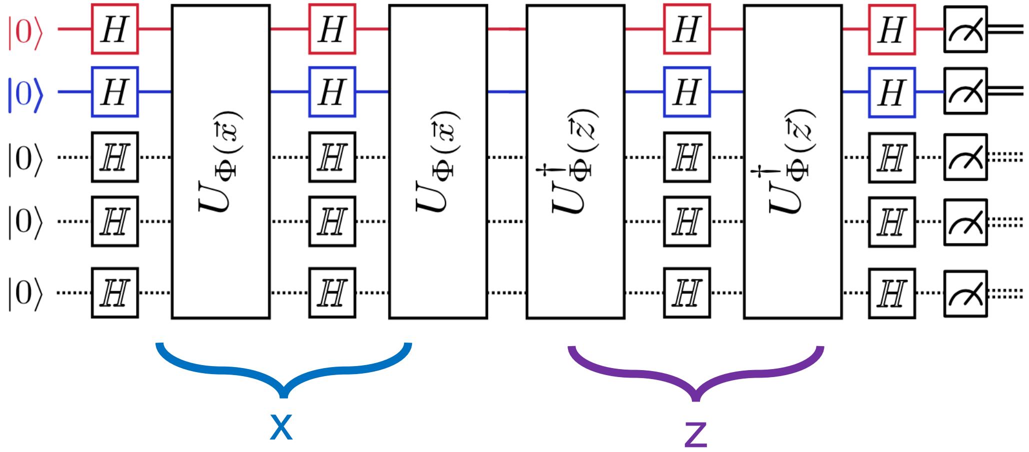
G.James, D.Witten, T.Hastie, R.Tibshirani: An introduction to statistical learning

Quantum Kernel Estimator

Use quantum computer to:

- encode the data;
- estimate the kernel as the ***fidelity*** between pairs of feature vectors;
- plug $\boldsymbol{K}_{i,j}$ into the Dual $L_D(\alpha)$ and get α_i
- **Classical computer are then used to do the SVM according to:**

$$label(s) = sign\left(\sum_{i \in T} \alpha_i y_i \boldsymbol{K}(x_i, s) + b \right)$$



$$|\langle \Phi(\vec{x}) | \Phi(\vec{z}) \rangle|^2 = |\langle 0^n | \mathcal{U}^\dagger_{\Phi(\vec{x})} \mathcal{U}_{\Phi(\vec{z})} | 0^n \rangle|^2$$

V.Havlicek et al, Nature 567, 209 (2019)

Quantum SVM

QSVM replaces the kernel of classical SVM with a quantum kernel (inner product of quantum state)

SVM Execution Flow

- Prepare training dataset
- Calculate kernel matrix $K(\boldsymbol{x}_i, \boldsymbol{x}_k)$ ← Make it quantum
- Determine parameters which maximize margin M
- Determine the classification boundary
- Classify the data for prediction

Classical

Polynomial Kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_k) = \left(\boldsymbol{x}_i^T \boldsymbol{x}_k + c\right)^d$

- Correspond to Feature vectors where all of monomials of order d and below are in each element

Gaussian Kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_k) = \exp\left(-\frac{|\boldsymbol{x}_i - \boldsymbol{x}_k|^2}{2\sigma^2}\right)$

- Equivalent to using an infinite dimensional feature vector ϕ
- Expected to contain essential feature

Quantum

Quantum Kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_k) = |\langle \Phi(\boldsymbol{x}_i) | \Phi(\boldsymbol{x}_k) \rangle|^2$

- Conjectured to difficult to simulate classically
- Constructable without deep circuit

Software

Agenda

- Part 1:intro QC
- Part 2: QC for Quantum Machine Learning
- Part 3: QML for HEP

How does CERN engage in Quantum Technologies?

QT4HEP

Can CERN stay out of quantum technologies?

- Develop **technologies, capabilities** required by CERN scientific programmes
- Allow CERN to interoperate with **future quantum infrastructures**

- **Extend and share** technologies uniquely available at CERN
- Boost development and adoption of QT beyond CERN
- Use CERN reputation to **maximise impact**

HEP4QT

How can CERN contribute to quantum technologies?

The CERN Quantum Technology Initiative

Understanding the impact of quantum technologies in HEP

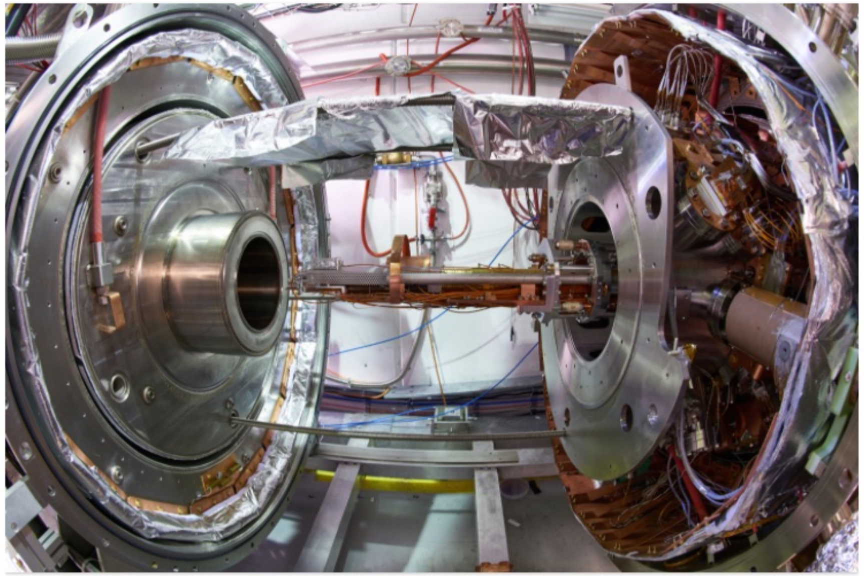
Voir en français

CERN meets quantum technology

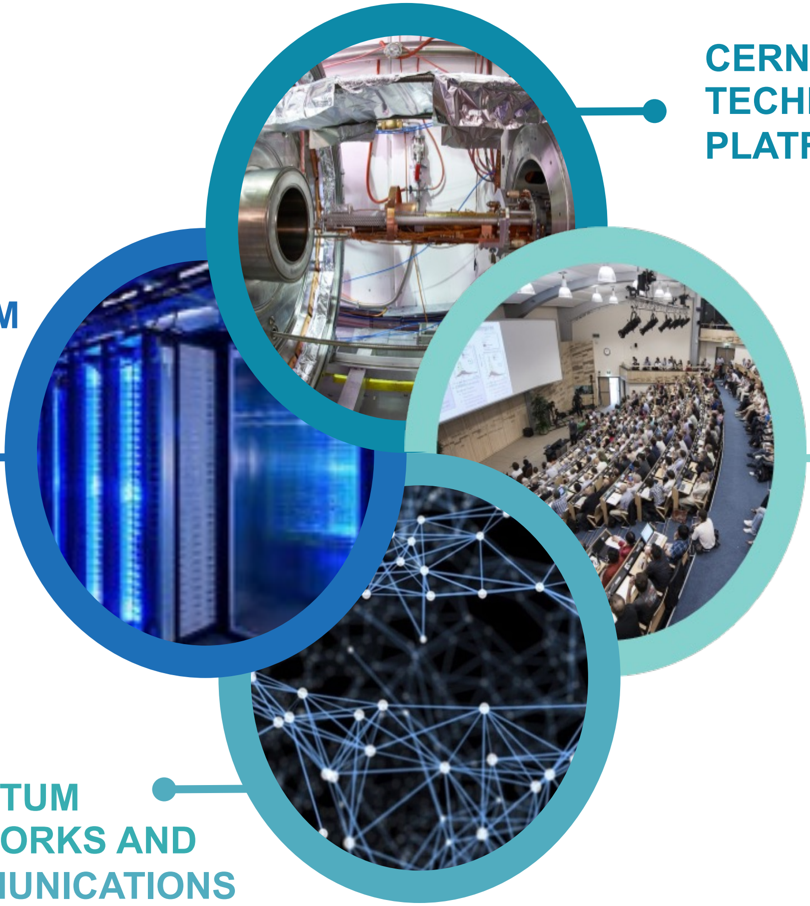
launched in 2020

The CERN Quantum Technology Initiative will explore the potential of devices harnessing perplexing quantum phenomena such as entanglement to enrich and expand its challenging research programme

30 SEPTEMBER, 2020 | By Matthew Chalmers



The AEgIS 1T antimatter trap stack. CERN's AEgIS experiment is able to explore the multi-particle entangled nature of photons from positronium annihilation, and is one of several examples of existing CERN research with relevance to quantum technologies. (Image: CERN)



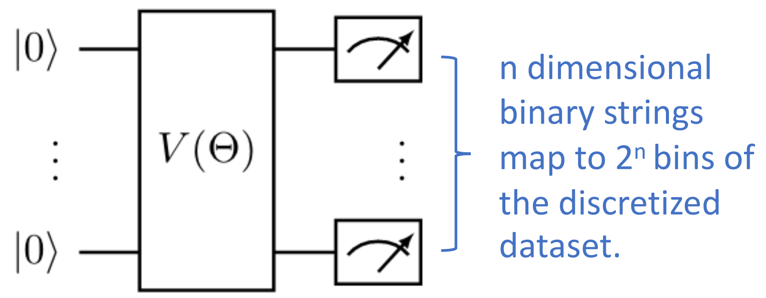
QTI Roadmap: https://doi.org/10.5281/zenodo.5553774

Quantum Generative Models

Delgado and Hamilton, arXiv:2203.03578 (2022)
Zoufal, et al., *npj Quantum Inf* **5**, 103 (2019)
Leadbeater et al., *Entropy* **2021**, *23*, 1281.
Amin, et al. *Physical Review X* 8.2 (2018): 021050.

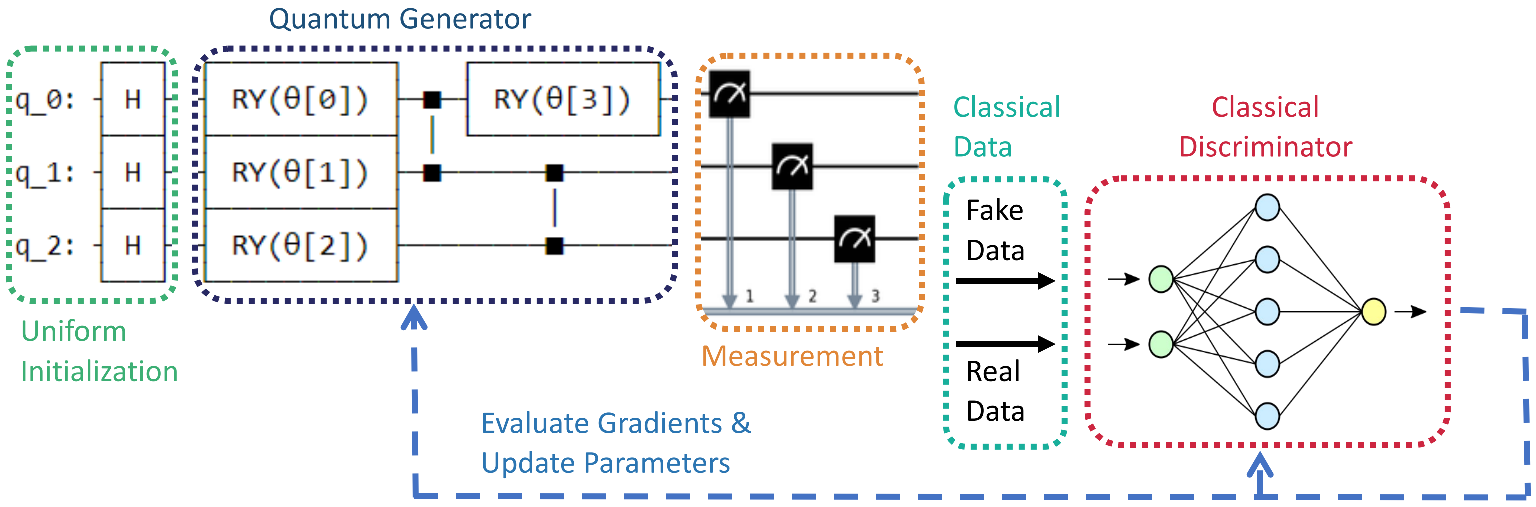
QCBM

Sample variational pure state $|\psi(\theta)\rangle$ by projective measurement through **Born rule**: $\mathbf{p}_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x}|\boldsymbol{\psi}(\boldsymbol{\theta})\rangle|^2$.



QGAN

Multiple implementations, mostly classical-quantum hybrid



QBM

Network of stochastic binary units with a quadratic energy function that follows the Boltzman distribution (Ising Hamiltonian)

$$H = -\sum_a b_a \sigma_a^z - \sum_{a,b} w_{ab} \sigma_a^z \sigma_b^z$$

Typical metrics:

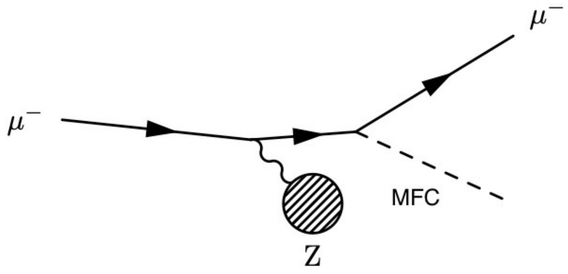
$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\,\log\left(\frac{P(i)}{Q(i)}\right)$$

$$\mathrm{MMD}(\mathbb{P}_r, \mathbb{P}_g) = \left(\mathbb{E}_{\substack{\mathbf{x}_r, \mathbf{x}_r' \sim \mathbb{P}_r, \\ \mathbf{x}_g, \mathbf{x}_g' \sim \mathbb{P}_g}}\left[k(\mathbf{x}_r, \mathbf{x}_r') - 2k(\mathbf{x}_r, \mathbf{x}_g) + k(\mathbf{x}_g, \mathbf{x}_g')\right]\right)^{\frac{1}{2}}$$

Quantum Circuit Born Machine for Event Generation

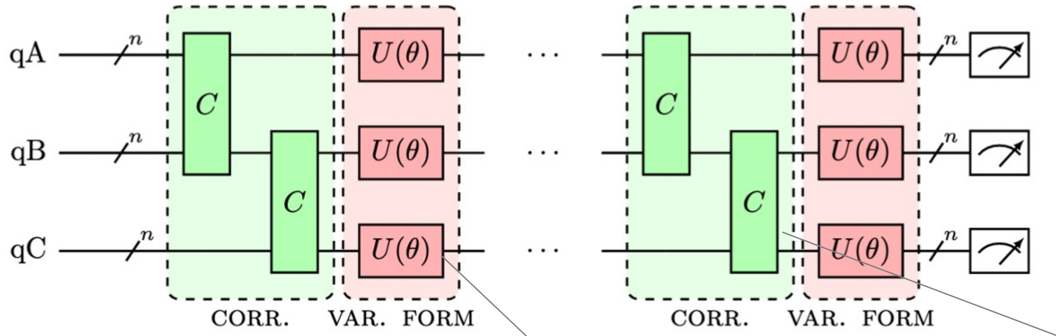
Born machine:

Produces statistics according to Born's measurement rule using parametrized quantum circuit $|\psi(\theta)\rangle$



Muon fixed target scattering experiment

$$p_\theta(x) = |\langle x|\psi(\theta)\rangle|^2, x \in \{0,1\}^{3n}$$



energy

pt

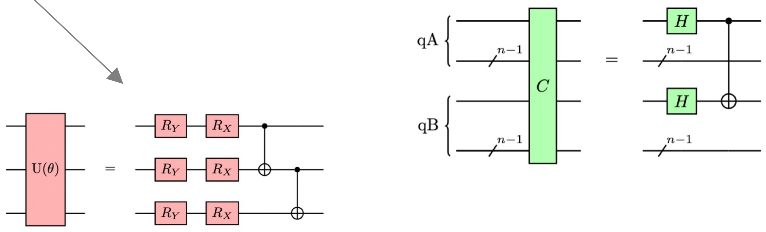
Transversal momentum

η

Pseudorapidity

Generate discrete PDFs
(continuous in the limit increasing no. of qubits)

Parametric Quantum Circuit



Kiss O., Grossi M. et all., **Conditional Born machine for Monte Carlo events generation**, *Phys. Rev. A* ***106****, 022612 (2022)*

Coyle, B., Mills, D. et al, **The Born supremacy.** In: *npj Quantum Inf 6, 60* (2020)

Quantum Circuit Born Machine for Event Generation

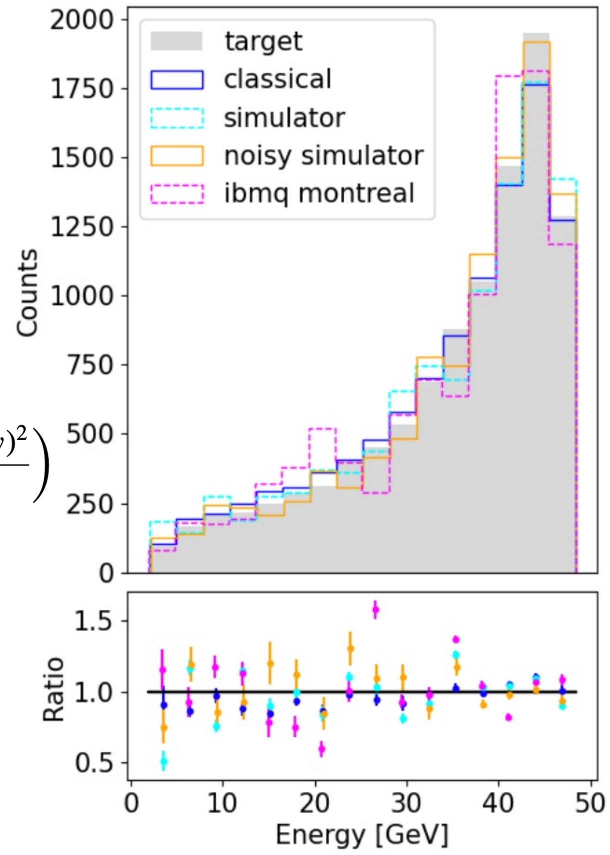
- Generate samples of discrete PDFs with Born machine
- Train using Maximum Mean Discrepancy loss function:

$$\text{MMD(P,Q)} = \mathbb{E}_{\substack{X\sim P \\ Y\sim P}}[K(X,Y)] + \mathbb{E}_{\substack{X\sim Q \\ Y\sim Q}}[K(X,Y)] - 2\mathbb{E}_{\substack{X\sim P \\ Y\sim Q}}[K(X,Y)]$$

Gaussian kernel

$$K(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma}\right)$$

→ **efficient way to generate multivariate (and conditional) distributions** with only linear connectivity, suitable for NISQ devices (suggested by numerical evidence)



Kiss O., Grossi M. et all., **Conditional Born machine for Monte Carlo events generation**, *Phys. Rev. A* ***106****, 022612 (2022)*

Coyle, B., Mills, D. et al, **The Born supremacy.** In: *npj Quantum Inf 6, 60* (2020)

Quantum Kernels for classification and anomaly detection

Analysis setup

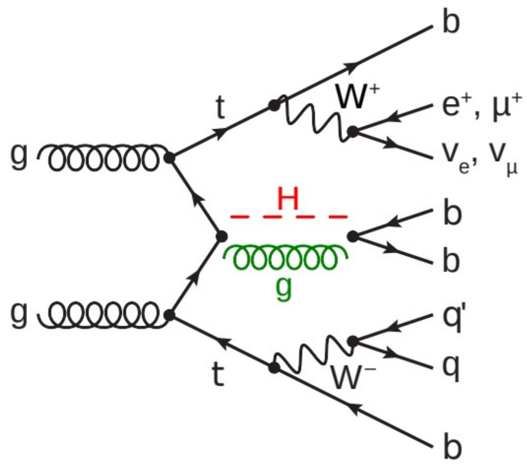
Analysis
Discrimination of the signal over the overwhelming background

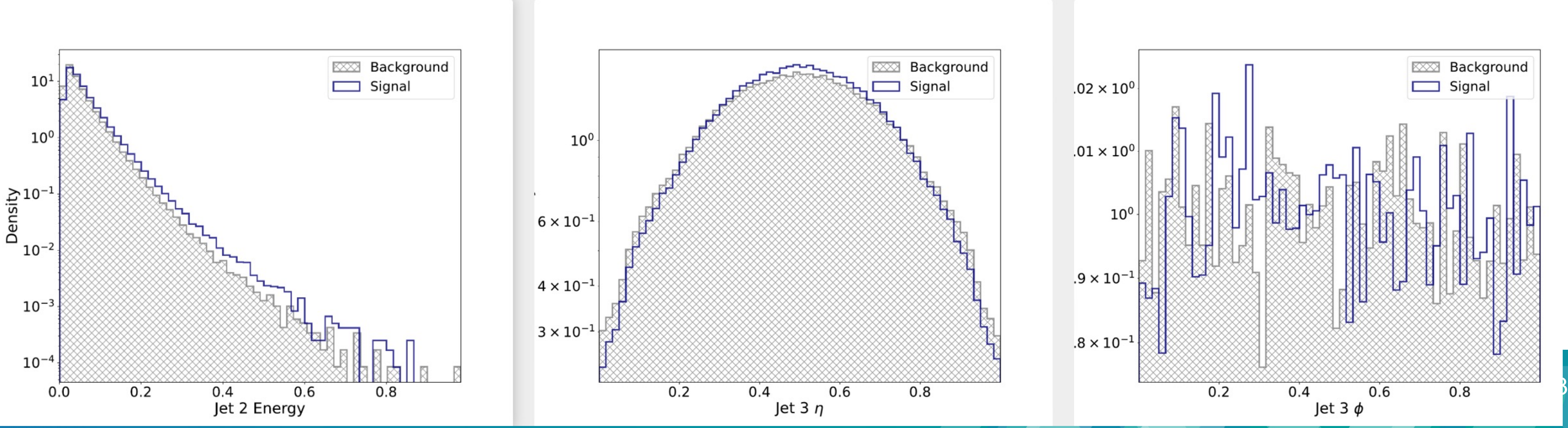
Features

- For the each jet we have 8 features: (pT,η,ϕ,E,b tag,px,py,pz)
- For MET we have 4 features: (pT,px,py,ϕ)
- For the lepton (electron or muon) we have 7

features: (pT,η,ϕ,E,px,py,pz)

#features = 8×7(jets)+7(1lepton)+4(MET) = 67



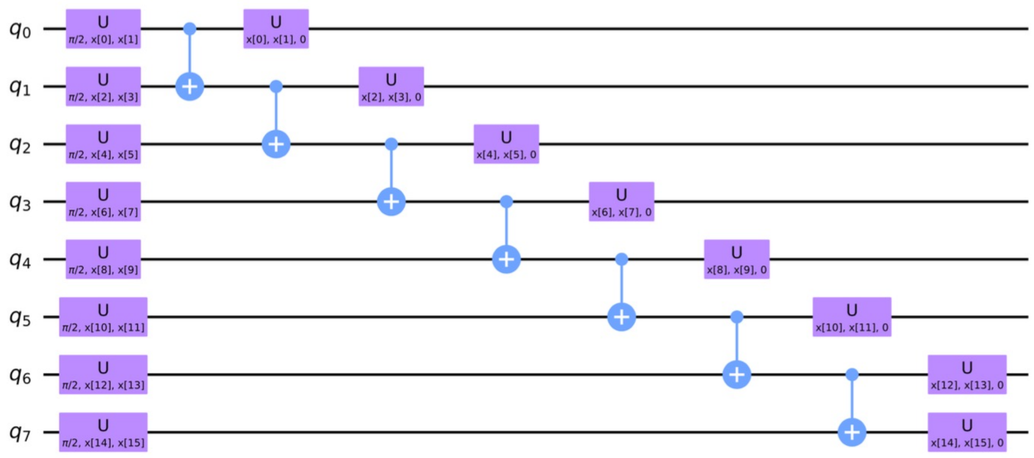


Quantum SVM for Higgs Classification

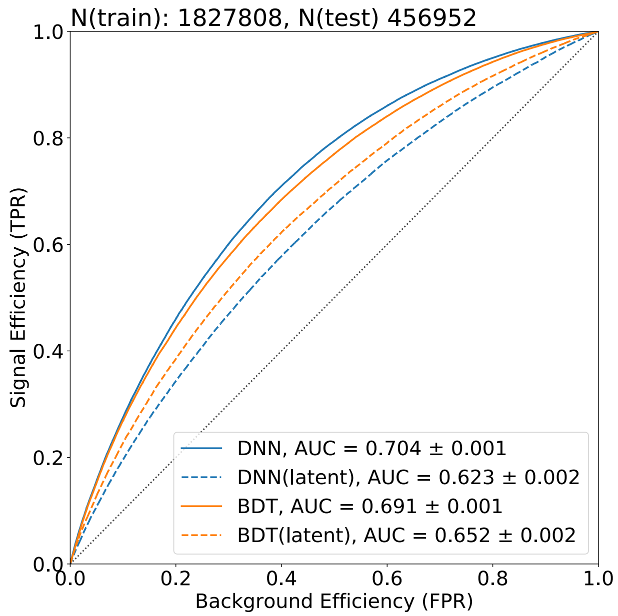
Input dimensionality reduction through an Auto-Encoder projects to a lower dimension latent space (8,16)

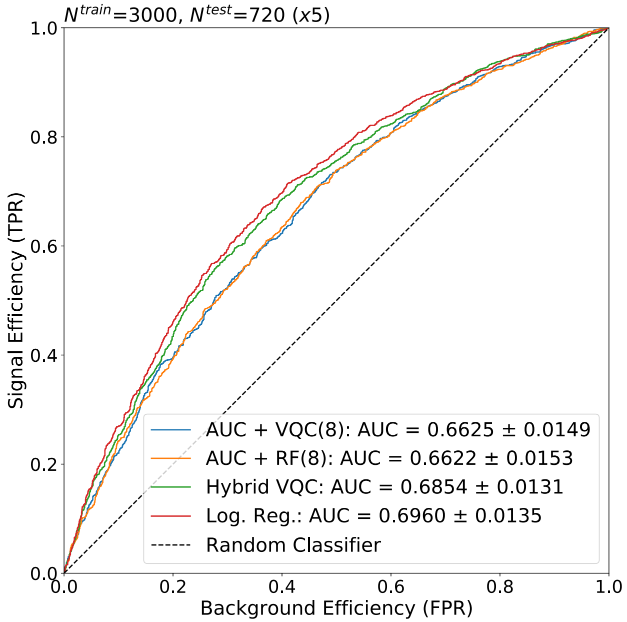
Feature selection + Model	AUC
AUC + QSVM	0.66 ± 0.01
PyTorch AE + QSVM	0.62 ± 0.03
AUC + SVM rbf	0.65 ± 0.01
PyTorch AE + SVM rbf	0.62 ± 0.02
KMeans + SVM rbf	0.61 ± 0.02

Feature selection + Model	AUC
AUC + QSVM	0.68 ± 0.02
AUC + Linear SVM	0.67 ± 0.02
Logistic Regression	0.68 ± 0.02

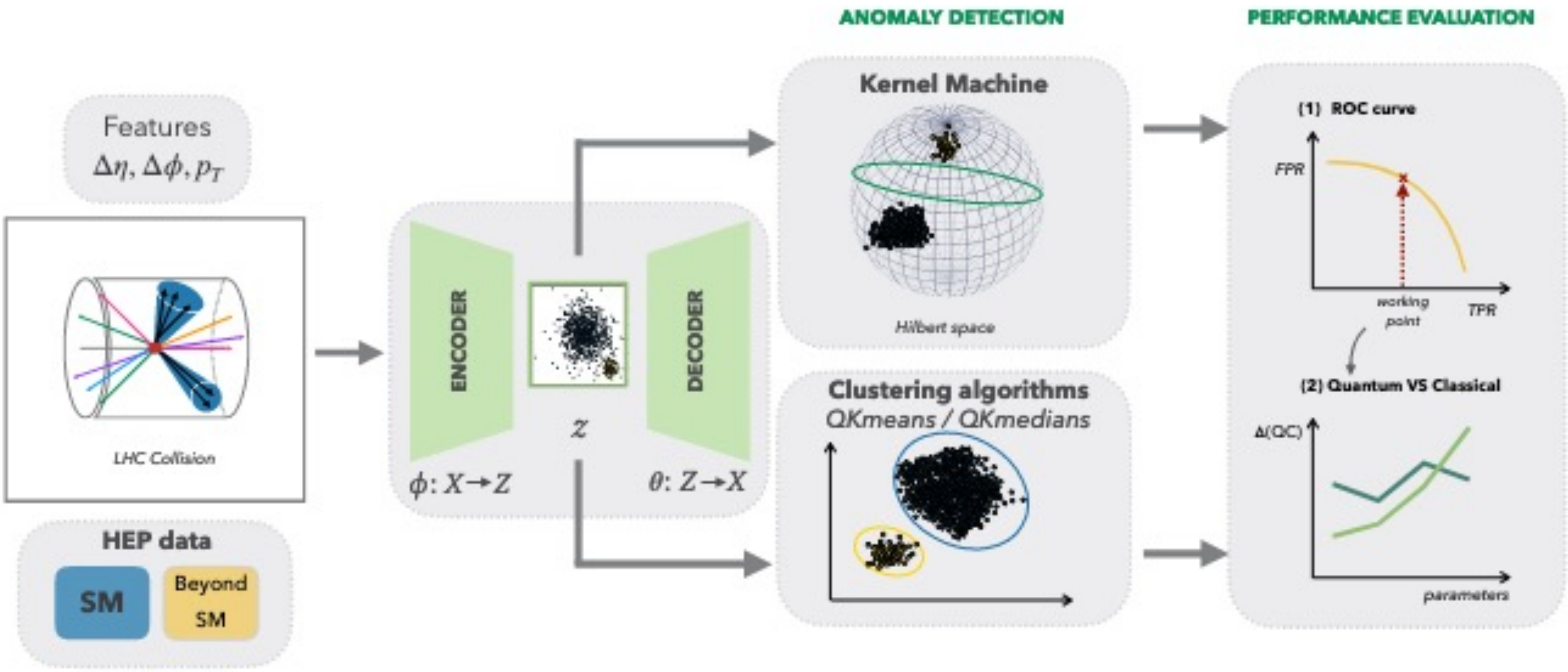


Data encoding circuit serving as feature map for the 8-qubit QSVM implementation.





Unsupervised learning for Anomaly Detection

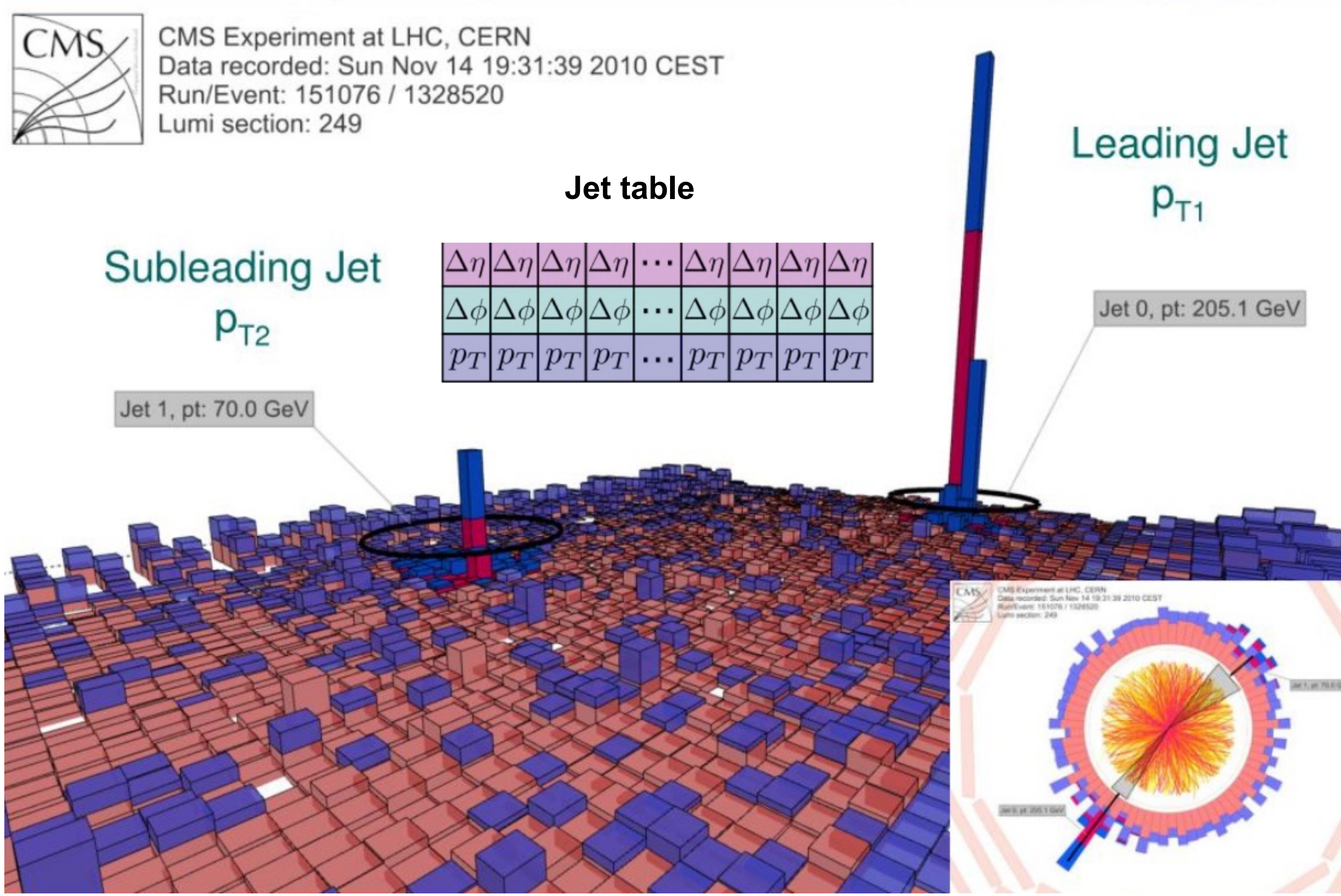


Standard Model jets

- Simulate QCD multi-jets at the LHC
- Build jet from 100 highest pt particles
- Apply realistic event selection

Convolutional AutoEncoder learns the jet **internal structure**

$\mathbb{R}^{300} \rightarrow \mathbb{R}^{\ell}$, $\ell = 4, 8, 16$

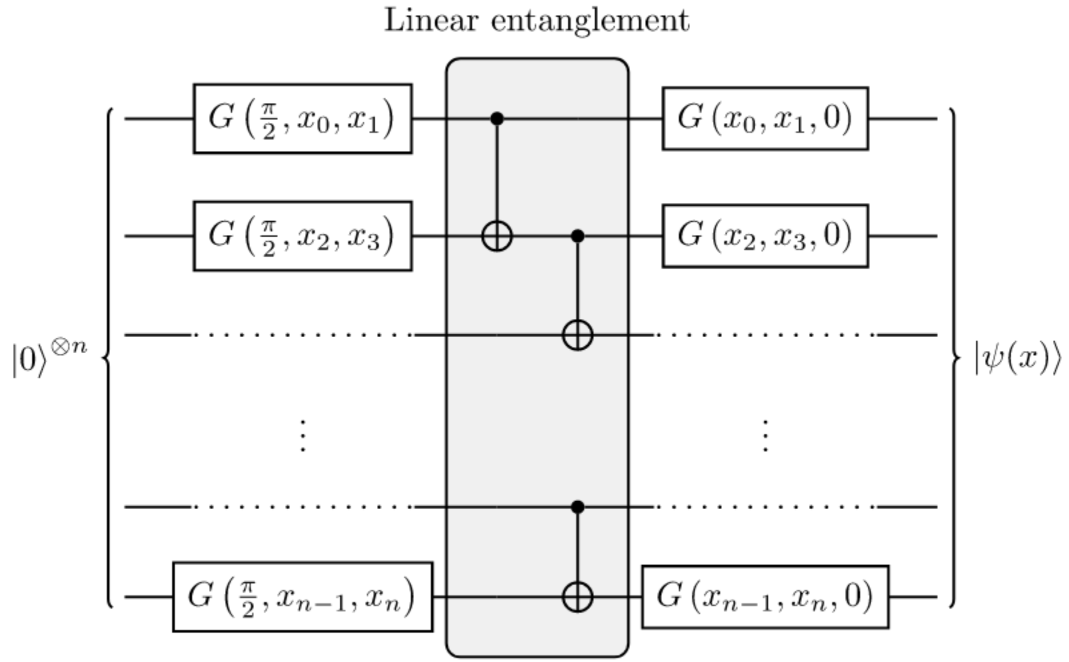


Unsupervised kernel machine

$$k(x_i, x_j) := \mathrm{tr}[\rho(x_i)\rho(x_j)] = \left|\langle 0|U^\dagger(x_i)U(x_j)|0\rangle\right|^2$$

$$\rho(x_i) := U(x_i)\,|0\rangle\,\langle 0|\,U^\dagger(x_i)$$

- Find the hyperplane that maximizes the distance of the data from the origin of the feature vector space

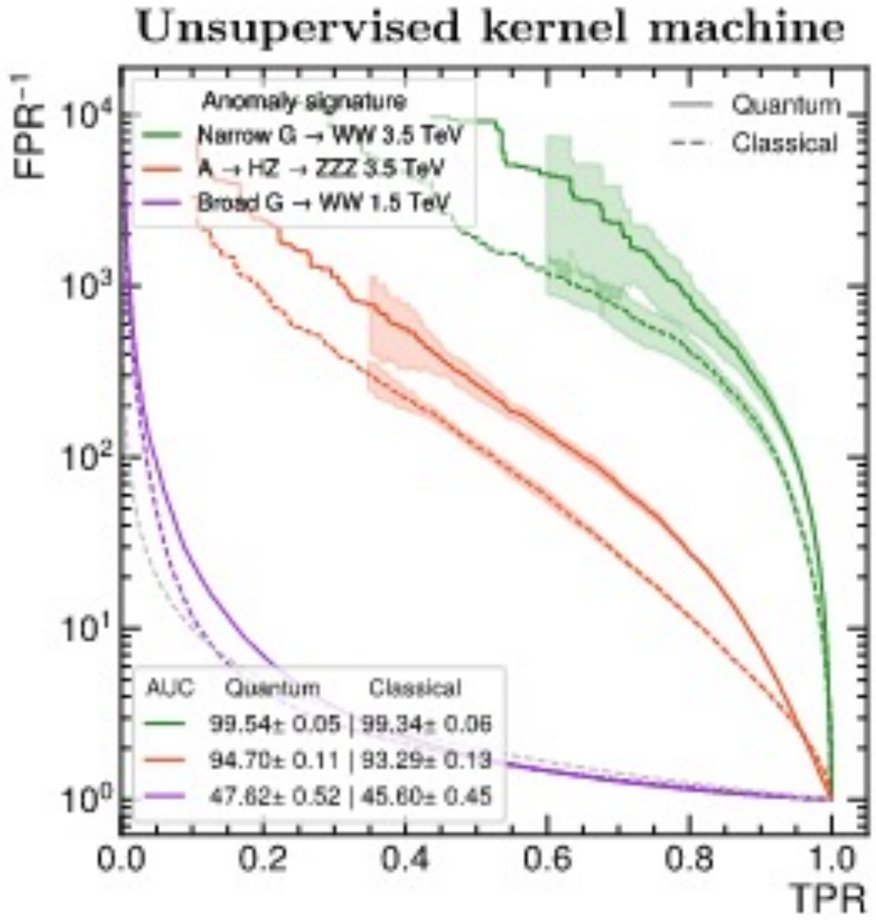


Upper bound on fraction of anomalies in training data at 0.01 (at most 1% QCD training data are falsely flagged)

$$\min_{w\in\mathcal{F},\,\xi\in\mathbb{R}^\ell,\,\rho\in\mathbb{R}} \quad \frac{1}{2}||w||^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho$$

$$\text{subject to} \quad w\cdot\Phi(x_i) \geq \rho - \xi_i,\ \xi_i \geq 0,\ \forall i. \quad \nu \in (0,1)$$

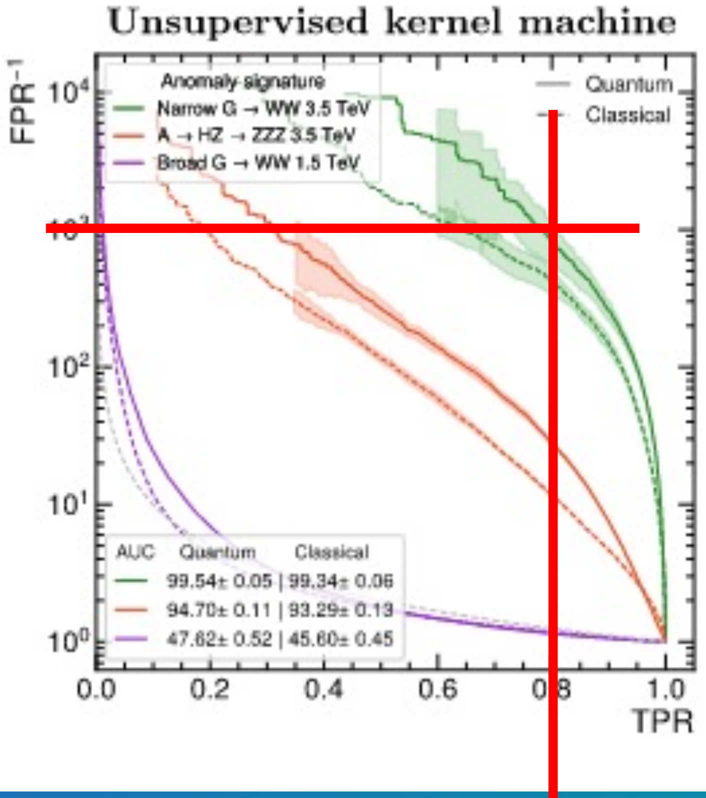
Results



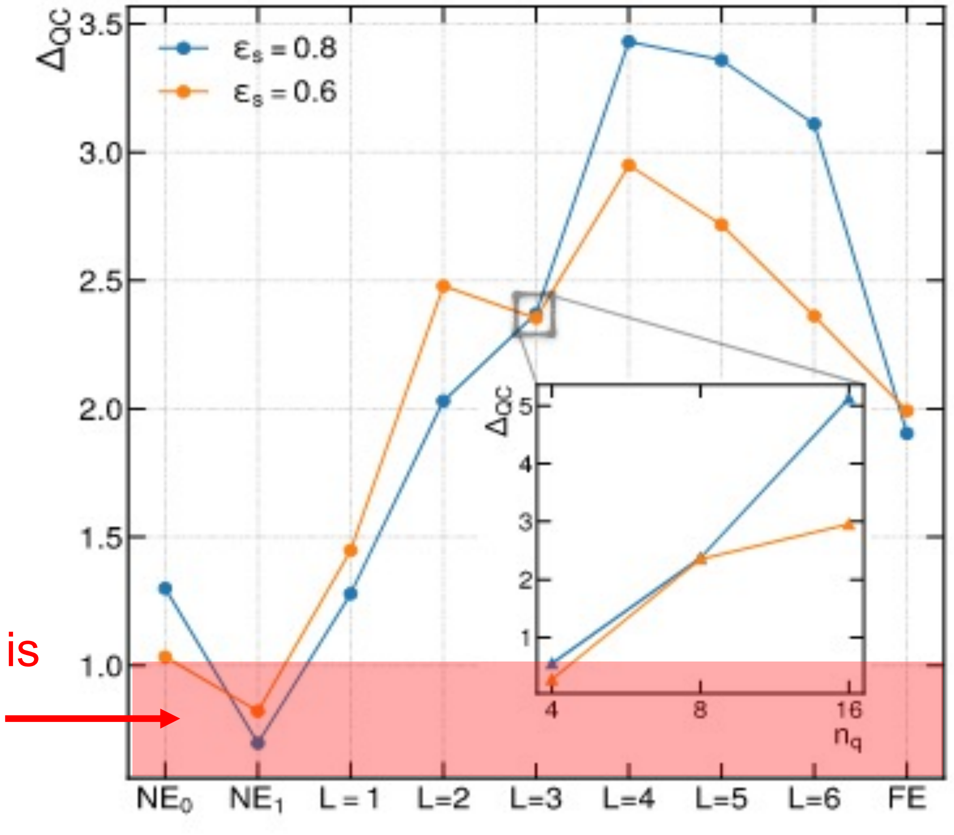
Is this an «advantage» we can use?

Quantum anomaly detection in the latent space of proton collision events at the LHC
Vasileios Belis *et al.*, *arXiv:2301.10780*.

In reality….



Classical is better



Higher is better

Increasing entanglement & expressivity

Quantum anomaly detection in the latent space of proton collision events at the LHC
Vasileios Belis *et al.*, *arXiv:2301.10780*.

Working with Quantum Kernels

A priori methodology to assess quantum advantage according to data geometry and kernels structure

- Create **classically intractable features in the Hilbert space** to reach advantage. However

Hilbert space is exponentially larger

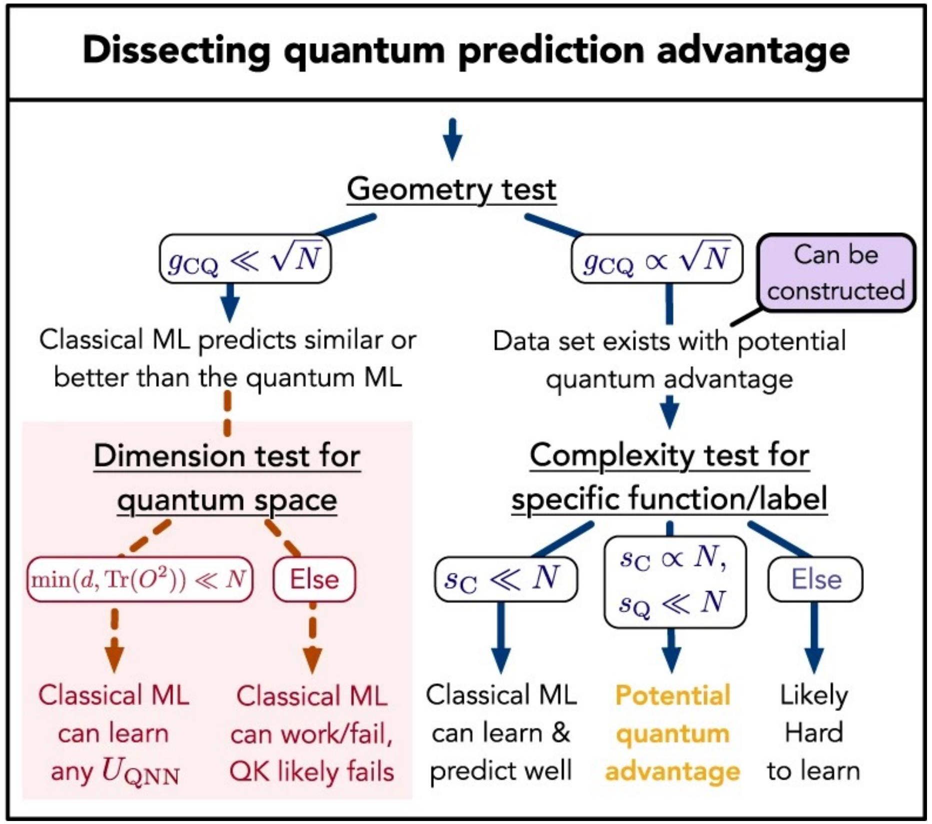
↓

Sparser data

↓

Loss of predictive power

- How do we find optimal kernel properties ?



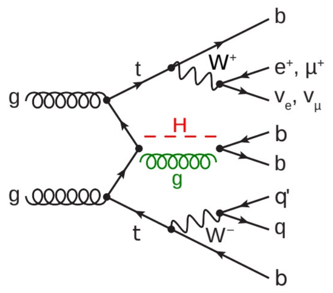
HY Huang et al, (2021), Power of Data in Quantum Machine Learning, Nature Comm

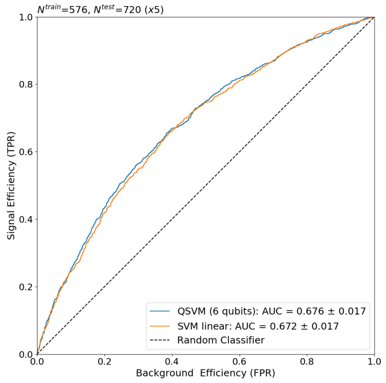
Ex. Projected Quantum Kernel

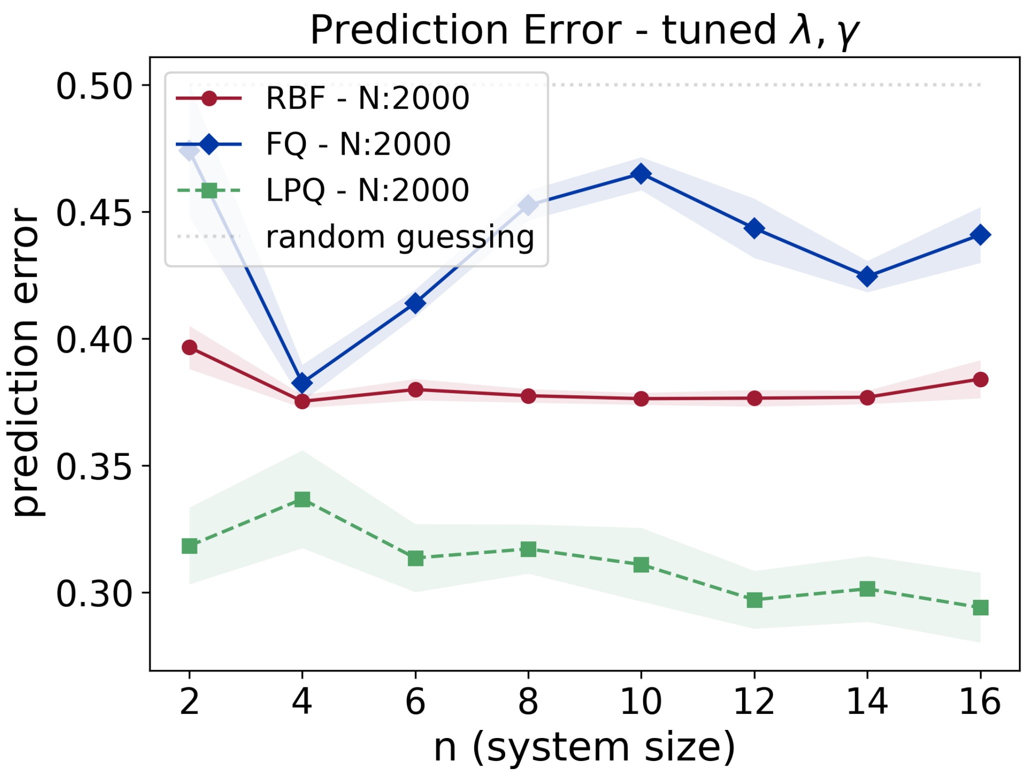
Project quantum kernels lower dimensionality of the representation (i.e. local density matrix)[1]:

- Improved generalizion while keeping features into states classically hard

- Example: ttH(bb) binary classification[2]

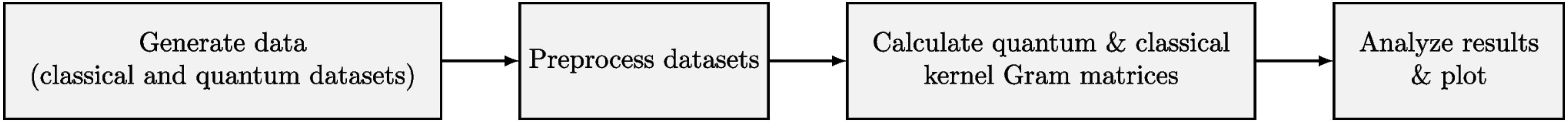






[1] Huang, Hsin-Yuan, et al. "Power of data in quantum machine learning." *Nature communications* 12.1 (2021): 2631.
[2] V Belis et al, (2021), *Higgs Analysis with Quantum Classifiers*, EPJ Web Conf

Predicting advantage with QUASK



Can we automatize this pipeline?

Di Marcantonio, F., Incudini, M., Tezza, D., and Grossi, M. **"Quantum Advantage Seeker with Kernels (QuASK): a software framework to speed up the research in quantum machine learning."**
Quantum Mach. Intell. 5, 20 (2023).
https://doi.org/10.1007/s42484-023-00107-2

Quask – Documentation and Tutorial

QUASK

https://quask.readthedocs.io/en/latest/index.html

Summary

Research on QML applications in High Energy Physics is producing a **large number of prototypes algorithms for potential future use-cases**

- Current focus on *algorithms for data processing* in a *controlled* environment for current hardware
- Preliminary hints for advantage in terms of *representational power of quantum states*
- Mostly, algorithm performance is *as good as* the classical counterpart
- Need *more robust studies* to relate architecture of quantum computational model and its performance to data sets
- *Identify use-cases* where quantum approach is provably *more efficient* than classical model
- Studying QML algorithms today *links Quantum computing and Learning Theory* and draw separation between classical and quantum learner

Open questions

- Quantum computing offers great opportunties while HEP provides challenging problems
  - **What are the most promising applications?**
  - How do we define performance and validate results on **realistic use cases**?

- Experimental data has high dimensionality
  - Can we **train Quantum Machine Learning algorithms effectively**?
  - Can **we reduce the impact of data reduction** techniques?
- Experimental data is shaped by **physics laws**
  - Can we leverage them to build better algorithms?
- CERN is committed to creating impact on QT research in the coming years

Lectures and Hands-On at CERN

- «A practical Introduction to quantum computing», Elias Combarro

https://indico.cern.ch/event/970903/

- «Introduction to quantum computing », Heather Grey

https://indico.cern.ch/event/870515/

- A set of two hands-on (introduction) sessions part of the 2023 openlab summer student lectures series

https://indico.cern.ch/event/1293871/

https://indico.cern.ch/event/1293874/


CERN Quantum Technology Initiative
Accelerating Quantum Technology Research and Applications

Thanks!

RZZGate

`qiskit.circuit.library.RZZGate(theta, label=None, *, duration=None, unit='dt')` GitHub ↗

Bases: `Gate`

A parametric 2-qubit $Z \otimes Z$ interaction (rotation about ZZ).

This gate is symmetric, and is maximally entangling at $\theta = \pi/2$.

Can be applied to a `QuantumCircuit` with the `rzz()` method.

Circuit Symbol:

```
q_0: ───■────
        │zz(θ)
q_1: ───■────
```

Matrix Representation:

$$R_{ZZ}(\theta) = \exp\left(-i\frac{\theta}{2} Z\otimes Z\right) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 & 0 & 0 \\ 0 & e^{i\frac{\theta}{2}} & 0 & 0 \\ 0 & 0 & e^{i\frac{\theta}{2}} & 0 \\ 0 & 0 & 0 & e^{-i\frac{\theta}{2}} \end{pmatrix}$$

This is a direct sum of RZ rotations, so this gate is equivalent to a uniformly controlled (multiplexed) RZ gate:

$$R_{ZZ}(\theta) = \begin{pmatrix} RZ(\theta) & 0 \\ 0 & RZ(-\theta) \end{pmatrix}$$

Examples:

$$R_{ZZ}(\theta = 0) = I$$

$$R_{ZZ}(\theta = 2\pi) = -I$$

$$R_{ZZ}(\theta = \pi) = -Z \otimes Z$$

$$R_{ZZ}\left(\theta = \frac{\pi}{2}\right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1-i & 0 & 0 & 0 \\ 0 & 1+i & 0 & 0 \\ 0 & 0 & 1+i & 0 \\ 0 & 0 & 0 & 1-i \end{pmatrix}$$

Create new RZZ gate.

Qiskit

FidelityQuantumKernel

`class FidelityQuantumKernel(*, feature_map=None, fidelity=None, enforce_psd=True, evaluate_duplicates='off_diagonal')` [source]

Bases: `BaseKernel`

An implementation of the quantum kernel interface based on the `BaseStateFidelity` algorithm.

Here, the kernel function is defined as the overlap of two quantum states defined by a parametrized quantum circuit (called feature map):

$$K(x, y) = |\langle \phi(x) | \phi(y) \rangle|^2$$

Parameters:

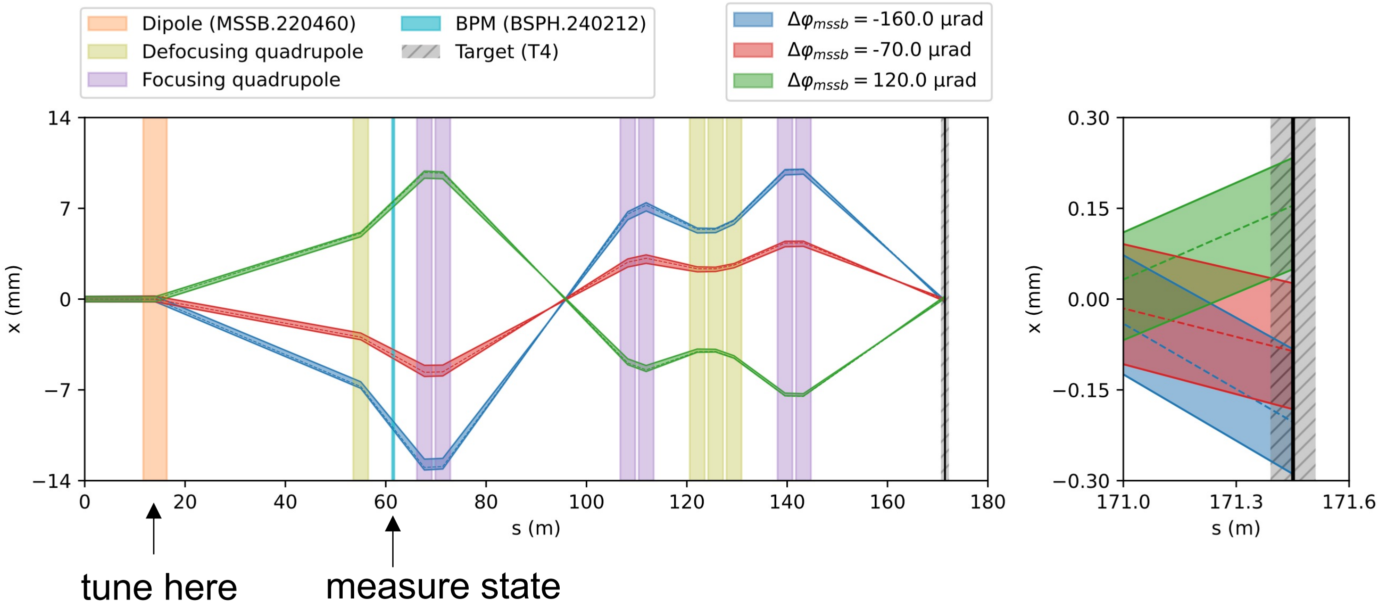
- **feature_map** (*QuantumCircuit | None*) – Parameterized circuit to be used as the feature map. If `None` is given, `ZZFeatureMap` is used with two qubits. If there's a mismatch in the number of qubits of the feature map and the number of features in the dataset, then the kernel will try to adjust the feature map to reflect the number of features.
- **fidelity** (*BaseStateFidelity | None*) – An instance of the `BaseStateFidelity` primitive to be used to compute fidelity between states. Default is `ComputeUncompute` which is created on top of the reference sampler defined by `Sampler`.
- **enforce_psd** (*bool*) – Project to the closest positive semidefinite matrix if `x = y`. Default `True`.
- **evaluate_duplicates** (*str*) –

  Defines a strategy how kernel matrix elements are evaluated if duplicate samples are found. Possible values are:

  - `all` means that all kernel matrix elements are evaluated, even the diagonal ones when training. This may introduce additional noise in the matrix.
  - `off_diagonal` when training the matrix diagonal is set to *1*, the rest elements are fully evaluated, e.g., for two identical samples in the dataset. When inferring, all elements are evaluated. This is the default value.
  - `none` when training the diagonal is set to *1* and if two identical samples are found in the dataset the corresponding matrix element is set to *1*. When inferring, matrix elements for identical samples are set to *1*.

Quantum Reinforcement Learning (RL)

Michael Schenk et al., **Hybrid actor-critic algorithm for quantum reinforcement learning at CERN beam lines.** arXiv:2209.11044



Beam Target Steering Task

Formulate as RL problem:

- Action: (discrete) deflection angle
- State: (continuous) BPM position
- Reward: integrated beam intensity on target
- Optimality: fraction of states for which the agent takes the right decision

Quantum Reinforcement Learning (RL)

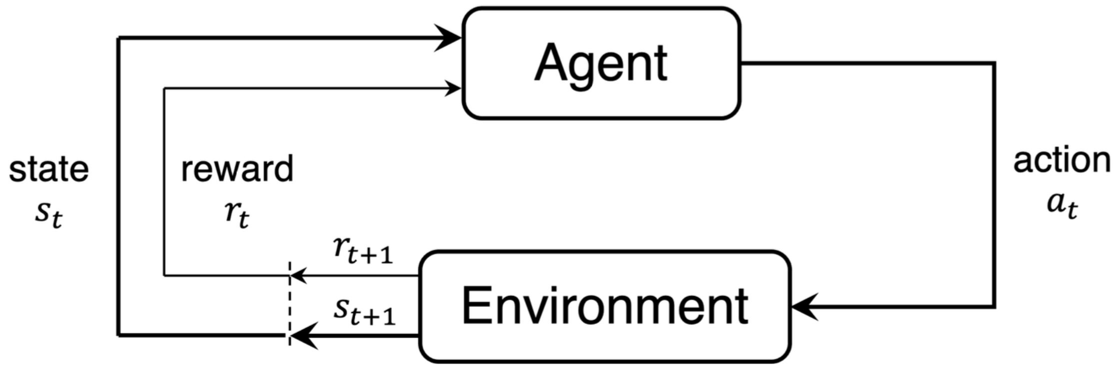
Michael Schenk et al., **Hybrid actor-critic algorithm for quantum reinforcement learning at CERN beam lines.** arXiv:2209.11044

Task: Beam optimization in linear accelerators

→ Use Reinforcement Learning (sample efficient)

Agent interacts with environment

- Follow **policy** $\boldsymbol{\pi(a_t|s_t)}$
- Goal: Find policy that **maximizes reward**



Schema of iterative Feedback-loop in RL

Expected reward is estimated by ***value function*** $\boldsymbol{Q(s,a)}$

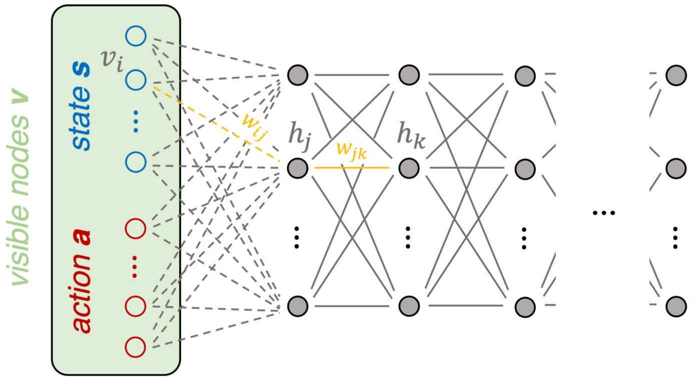
- **DQN:** Deep Q-learning *(NN-based)*
- **FERL:** Free energy-based RL *(clamped Quantum Boltzmann Machine)*

Structure of the **Quantum RL scheme**:

- Agent is **classical**
- Q-function is computed as the **energy of a qubit system**

Quantum Reinforcement Learning (RL)

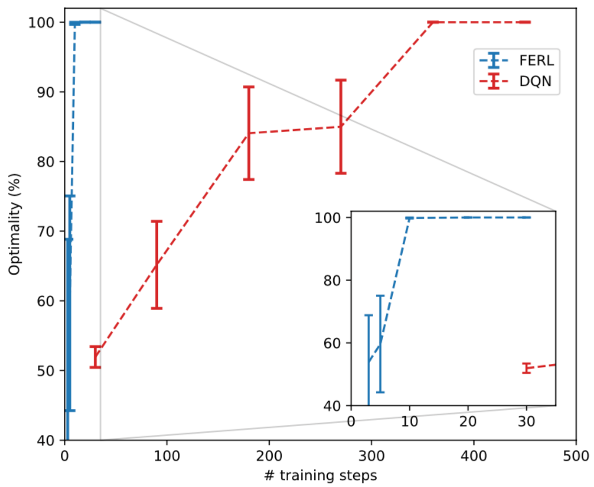
Michael Schenk et al., **Hybrid actor-critic algorithm for quantum reinforcement learning at CERN beam lines.** arXiv:2209.11044



Structure of clamped Quantum Boltzmann Machine (QBM)

→ Weights of QBM can be learned iteratively (analogous to classical Q-learning)

Quantum Annealing



Convergence Study for one-dim. beam target steering task

Transverse Field Ising model

$$\mathcal{H}(\mathbf{v}) = -\sum_{\substack{i\in V,\\ j\in H}} w_{ij} v_i \sigma^z_{h_j} - \sum_{j,k\in H} w_{jk}\sigma^z_{h_j}\sigma^z_{h_k} - \Gamma \sum_{j\in H} \sigma^x_{h_j}$$

$$\hat{Q}(s,a) \approx -F(\boldsymbol{v}) = -\langle H^{\mathrm{eff}}_{\boldsymbol{v}} \rangle - \frac{1}{\beta}\sum_c \mathbb{P}(c|\boldsymbol{v}) \log \mathbb{P}(c|\boldsymbol{v})$$

→ **Quantum RL converges much faster than** classical Q-learning *(8±2 vs. 320±40 steps with e. r.)*