Quantum Machine Learning in High Energy Physics
Outline

- Introduction
- The CERN Quantum Technology Initiative
- Qubits and circuits
- Quantum Machine Learning
- Applications in High Energy Physics
- Examples from CERN
- Summary
QC use cases in different sectors: the situation in 2019 with the estimated medium (2025) and long (2035) term impact.
Potential Applications

Quantum effects improve and accelerate complex algorithms

- Efficient sampling, searches and optimization
- Linear algebra, matrices and machine learning
- Algorithms/methods for cryptography and communication

Challenge is re-thinking algorithms design and define fair benchmarking and comparison to classical algorithms

Many potential applications in High Energy Physics:

- Monte Carlo and Event Generation
- Quantum simulation
- Pattern Recognition
- QML

Ex.: Exponential data compression with a Quantum Associative memory

CERN QTI and its Roadmap

CERN established the QTI in 2020

• Roadmap in 2021
• Publicly available on Zenodo
  • Accessed more than 6000 times
  https://doi.org/10.5281/zenodo.5553774

T1 - Scientific and Technical Development and Capacity Building

T2 - Co-development

T3 - Community Building

T4 - Integration with national and international initiatives and programmes

CERN meets quantum technology

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 AqgS-LT antinuclide trap stack. CERN's AqgS 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)
Scientific Objectives

Computing & Algorithms
- Assess the areas of potential quantum advantage in HEP (QML, classification, anomaly detection, tracking)
- Develop common libraries of algorithms, methods, tools; benchmark as technology evolves
- Collaborate to the development of shared, hybrid classic-quantum infrastructures

Simulation & Theory
- Identify and develop techniques for quantum simulation in collider physics, QCD, cosmology within and beyond the SM
- Co-develop quantum computing and sensing approaches by providing theoretical foundations to the identifications of the areas of interest

Sensing, Metrology & Materials
- Develop and promote expertise in quantum sensing in low- and high-energy physics applications
- Develop quantum sensing approaches with emphasis on low-energy particle physics measurements
- Assess novel technologies and materials for HEP applications

Communications & Networks
- Co-develop CERN technologies relevant to quantum infrastructures (time synch, frequency distribution, lasers)
- Contribute to the deployment and validation of quantum infrastructures
- Assess requirements and impact of quantum communication on computing applications (security, privacy)
Quantum Computing at CERN

• QC is one of the four research areas in the CERN QTI

• Understand which applications can profit from quantum algorithms
  • Choose representative use cases
  • Understand challenges and limitations (on NISQ and fault tolerant hardware)
  • Optimize quantum algorithms

• Quantum Machine Learning algorithms are a primary candidate for investigation
  • Increasing use of ML in many computing and data analysis flows
  • Can be built as hybrid models where quantum computers act as accelerators
  • Efficient data handling is a challenge
Quantum Computing Intro

An Introduction to Quantum Computing, E. Combarro, https://indico.cern.ch/event/970905
Qubit: Quantum bit

• Classical bits are binary “0 or 1”
• Quantum Mechanics predicts superposition states “simultaneously 0 and 1”
• Superposition can lead to highly parallel computations (exponential speedup)
• State of the “output qubit” has to be measured (stochastic nature of the result)
  • Qubit state collapses
• No-cloning theorem
Creating qubit: superconducting rings

- Current oscillates in resistance-free circuit loop
- Injected microwave signal excites the current into superposition states
  Ex. Google, IBM, …

Electron potential energy

Quantization of energy

Anharmonic d.o.f in spin system

No level here

| 1 ⟩
| 0 ⟩

Energy

Anharmonic oscillator

| 1 ⟩
| 0 ⟩

Magnetic flux

Z. Minev, Qiskit Global Summer School 2020
Different qubits

SuperConducting loops

Spin States

Polarization States

PHOTONS:

Trapped Atoms and Ions

Time qubits

Path Qubits:

Neutral atom arrays

• Configurable arrays of **single neutral atoms**
• 2 energy levels represent the qubit states
• Use **lasers** to control position and the state of the atom
  • assemble and read-out registers made of **hundreds of qubits**
  • fully programmable quantum processing
• High connectivity
• Specific computation cycle because the **register is not permanently built**
  • register preparation
  • quantum processing
  • register readout

Photonic based quantum computers

• Quantum superposition of different number of photons in a resonator generated by laser pulses (squeezed states)

• Set of quantum gates is implemented in a interferometer network (phase shifters and beam splitters)

• Photons are detected during the readout stage by superconducting counters

• Naturally represent continuous variables
Qubit representation

- **Dirac notation** is used to describe quantum states.

  Given a basis of orthogonal vectors

  \[ |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

  And a 2-dimensional vector in complex space

  \[ \alpha, \beta \in \mathbb{C}^2, \quad |\alpha|^2 + |\beta|^2 = 1 \]

  A quantum state is represented as

  \[ |\psi\rangle = \alpha |0\rangle + \beta |1\rangle \]
Quantum Gates

• Evolution of isolated quantum states follow Schrodinger equation
• Operations on qubits are unitary matrices describing state evolution
  • Reversible operations
  • Input and output states have the same dimension
  • Some classical gates (or, and, nand, xor...) cannot be implemented directly
  • Can simulate any classical computation with small overhead

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

\[
UU^\dagger = U^\dagger U = I
\]

\[
|\psi\rangle = \alpha |0\rangle + \beta |1\rangle
\]

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

\[
| (a\alpha + b\beta)^2 + (c\alpha + d\beta)^2 = 1
\]

04.08.22
Example gates

The $H$ or Hadamard gate

- The $H$ or Hadamard gate is defined by the (unitary) matrix
  \[
  \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
  \]
- Its action is
  \[
  |0\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}
  \]
- We usually denote
  \[
  |+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}}
  \]
  \[
  |\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}
  \]

The $Z$ gate

- The $Z$ gate is defined by the (unitary) matrix
  \[
  \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
  \]
- Its action is
  \[
  |0\rangle \rightarrow |0\rangle
  \]
  \[
  |1\rangle \rightarrow -|1\rangle
  \]

The $X$ or NOT gate

- The $X$ gate is defined by the (unitary) matrix
  \[
  \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
  \]
- Its action (in quantum circuit notation) is
  \[
  |0\rangle \rightarrow X |1\rangle
  \]
  \[
  |1\rangle \rightarrow X |0\rangle
  \]
  that is, it acts like the classical NOT gate.
- On a general qubit its action is
  \[
  \alpha |0\rangle + \beta |1\rangle \rightarrow X (\alpha |0\rangle + \beta |1\rangle) = \alpha |1\rangle - \beta |0\rangle + \alpha |1\rangle
  \]

Other important gates

- $Y$ gate
  \[
  \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}
  \]
- $S$ gate
  \[
  \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}
  \]
- $T$ gate
  \[
  \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}
  \]

The gates $X$, $Y$ and $Z$ are also called, together with the identity, the Pauli gates. An alternative notation is $\sigma_X$, $\sigma_Y$, $\sigma_Z$. 

E. Combarro, https://indico.cern.ch/event/970905
Quantum entanglement

- **Quantum entanglement** creates correlation between qubit that, classically, would be independent
- Example: Bell state
Quantum circuits

Classical circuits combine **logical operations** (and, or, not, nand, and xor). Quantum circuits use reversible gates that change the quantum states of one, two, or more qubits.
Quantum Algorithms

A collection on http://quantumalgorithmzoo.org

- Multiple algorithms have been studied
  - Shor algorithm for **prime factorization**
  - Grover algorithm for unsorted DB **searches**
  - Quantum **Fourier Transform**
  - …

- Quantum-inspired algorithms (emulate quantum effects on classical hardware)
- Quantum Machine Learning
- Challenge is re-thinking **algorithms design** and define fair **benchmarking** and **comparison** to classical algorithms
Grover algorithm for pattern recognition

Quantum Associative Memory: Reconstruct particle trajectory by designing a DB of expected patterns and use the *generalised Grover algorithm* to match them to the detector output.
Quantum Annealing

- Annealing for optimization problems
  - PDF as a mountain landscape
  - Smoothly evolve probability of being at any given coordinate with time.
  - Probability increases around the coordinates of deep valleys
- Quantum systems based on superconducting qubits
- D-Wave Advantage: 5436 qubits - 15 connection (Pegasus)
  - Quantum superposition: scan simultaneously multiple coordinates
  - Quantum tunneling: reduces risk of local minima (tunnel through hills)
  - Quantum entanglement: discover correlations between the coordinates that lead to deep valleys.

\[ H(t) = A(t)H_0 + B(t)H_p \]

Training a classifier with QA

- Map the problem to a Ising model (spin lattice as qubit graph)
- Define Hamiltonian and train by minimizing energy
- First QC application to High Energy Physics

\[ H_{\text{Ising}} = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z \]

Adjacent qubits

Initial features

weak classifiers = Ising spins

Today’s challenges

• **Noisy Intermediate-Scale Quantum devices**
  • Limitations in terms of **stability** and **connectivity**
  • **De-coherence**, measurement errors or gate level errors (**noise**)
    • Specific **error mitigation techniques**
    • **Circuit optimisation**
    • Prefer algorithms **robust against noise**
• Quantum computers initially integrated in **hybrid quantum-classical infrastructure**
  • Engineering, cooling, I/O
  • Hybrid algorithms, QPU as accelerators

### Development Roadmap

<table>
<thead>
<tr>
<th>Year</th>
<th>Goal</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>Run quantum circuits on the IBM cloud</td>
<td>✔️</td>
</tr>
<tr>
<td>2020</td>
<td>Demonstrate and prototype quantum algorithms and applications</td>
<td>✔️</td>
</tr>
<tr>
<td>2021</td>
<td>Run quantum programs 100x faster with Qiskit Runtime</td>
<td>✔️</td>
</tr>
<tr>
<td>2022</td>
<td>Bring dynamic circuits to Qiskit Runtime to unlock more computations</td>
<td>✔️</td>
</tr>
<tr>
<td>2023</td>
<td>Enhancing applications with elastic computing and parallelization of Qiskit Runtime</td>
<td>✔️</td>
</tr>
<tr>
<td>2024</td>
<td>Improve accuracy of Qiskit Runtime with scalable error mitigation</td>
<td>✔️</td>
</tr>
<tr>
<td>2025</td>
<td>Scale quantum applications with circuit knitting toolbox, controlling Qiskit Runtime</td>
<td>✔️</td>
</tr>
<tr>
<td>Beyond 2026</td>
<td>Increase accuracy and speed of quantum workflows with integration of error correction into Qiskit Runtime</td>
<td>✔️</td>
</tr>
</tbody>
</table>

**Model Developers**

- Quantum algorithm and application modules
  - Machine learning | Natural science | Optimization

**Algorithm Developers**

- Quantum Serverless
  - Intelligent orchestration | Circuit Knitting Toolbox | Circuit libraries

**Kernel Developers**

- **Circuits**
  - Falcon: 27 qubits
  - Hummingbird: 65 qubits
  - Eagle: 127 qubits
  - Osprey: 433 qubits
  - Condor: 1,121 qubits
  - Flamingo: 1,386+ qubits
  - Kookaburra: 4,158+ qubits

- **Qiskit Runtime**
  - Dynamic circuits
  - Threaded primitives
  - Error suppression and mitigation
  - Error correction

- **System Modularity**
  - Heron: 133 qubits x p
  - Crossbill: 408 qubits
Quantum Machine Learning

QML tutorials and resources https://pennylane.ai

Supervised Learning with Quantum Computers
Maria Schuld
Francesco Petruccione
Quantum Machine Learning

Use **Quantum Computing** to accelerate **ML/DL**.

Quantum circuits are **differentiable** and can be trained **minimizing a cost function** dependent on training data:

1. **Feature extraction** and **data encoding**
   - How to represent classical data in quantum states?

2. **Model definition** (kernel based or variational)
   - Design wrt data

3. **Optimisation** and **convergence in Hilbert space**
   - **Convergence vs expressivity**
   - Barren plateau and vanishing gradients
   - Gradient-free or gradient-based optimisers
   - …
Dimensionality reduction and feature extraction

- Reduce size of classical data
- Optimize input (PCA, Auto-Encoders..)
- Pre-trained or co-trained in hybrid setup


Patrick Odagiu, 2021: End-to-end Sinkhorn autoencoder with a classifier NN (green). Sinkhorn part consists of an encoder (blue), decoder (red) and noise generator (orange).
Quantum embedding

Data embedding in quantum states: 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

\[ n_{\text{qubit}} \propto O(\log(N)) \]

Polynomial number of gates

\[ n_{\text{gate}} \propto O(\text{poly}(N)) \]

Figure 5. ROC distributions and AUC score for DNN (green), Angle Embedding (blue) and Amplitude Embedding circuits (yellow) for the muon dataset (a) and the complete dataset (b). The dashed line represents a random classifier.
Model definition

Kernel methods

Feature maps as quantum kernels

Use quantum computers to create classically intractable features $|\phi(x)\rangle$

- Build inner product of feature vectors $\rightarrow O(N_{data}^2)$
- Use classical kernel-based training
  - Convex losses, global minimum
- Identify classes of kernels that relate to specific data structures
- Given a variational circuit of the form $U(x, \theta) = V_{\theta}U_{\phi}(x)$, can define a quantum kernel method with better accuracy: $|\phi(x)\rangle = U_{\phi}(x)|0\rangle$
- Classically: not all machine learning models can be described by kernel methods.


Quantum Support Vector Machine

SVM are **kernel methods**:
- Trained to find the optimal separating plane

**Quantum SVM** use feature maps as kernels
- Feature maps enable SVM to design non-linear decision boundaries
- Feature maps in high dimensionality space improve separation power

\[ K_{ij} = \langle 0 | U^\dagger(\vec{x}_i) U(\vec{x}_j) | 0 \rangle^2 \]

**NB:**
- Quantum kernels sampled on quantum device
- Minimisation step is classical
Model definition

Variational algorithms

Define a parametric quantum circuit with trainable parameters $\vartheta$

$$U(x, \vartheta)$$

Given an observable $O$, build a model

$$y(x, \vartheta) = \langle 0|U^{\dagger}(x, \vartheta)OU(x, \vartheta)|0 \rangle$$

- Trained using gradient-free or gradient-based optimization in a classical loop
  - Backpropagation and auto-differentiation
- Data Embedding $V_{\phi}(x)$ can be learned
- Improve performance by designing architectures to leverage data symmetries
- There are quantum circuits that hard to simulate classically

Defining quantum Advantage for QML

Different possible definitions
- Runtime speedup
- Sample complexity
- Representational power

Classical Intractability: a quantum algorithm that cannot be efficiently simulated classically
- No established recipe for classical data
- Need to use the whole exponential advantage in Hilbert space, but will it converge?
  (Algorithm expressivity vs convergence and generalization)
Practical advantage

Practical implementation vs asymptotic complexity
- Data embedding
- NISQ vs ideal quantum devices
- Realistic applications

Performance metrics and fair comparison to classical models

HEP data is classical, but originally produced by quantum processes. It is these intrinsically quantum correlations we are trying to identify.

A change of paradigm could reflect in interesting insights
- What are natural building blocks for QML algorithms?
- How can we construct useful bridges between QC and learning theory?
- How can we make quantum software ready for ML applications?


Quantum Machine Learning examples
QML in High Energy Physics


Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy C Y Li, and et al. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lh on ibm quantum computer simulator and hardware with 10 qubits. Journal of Physics G: Nuclear and Particle Physics, 48(12):125003, Oct 2021


Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy C Y Li, and et al. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lh on ibm quantum computer simulator and hardware with 10 qubits. Journal of Physics G: Nuclear and Particle Physics, 48(12):125003, Oct 2021


Sau Lan Wu, Jay Chan, Wen Guan, Shaojun Sun, Alex Wang, Chen Zhou, Miron Livny, Federico Carminati, Alberto Di Meglio, Andy C Y Li, and et al. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lh on ibm quantum computer simulator and hardware with 10 qubits. Journal of Physics G: Nuclear and Particle Physics, 48(12):125003, Oct 2021


**QML at CERN**


**Chang S.Y. et al., Running the Dual-PQC GAN on Noisy Simulators and Real Quantum Hardware, QTML2021, ACAT21**

**M. Shenk, V. Kain, Quantum Reinforcement Learning, BQIT 2021, 2022 CERN openlab Tech Workshop**

**O. Kiss, Quantum Born Machine for event generation, ACAT2021**


**Kinga Wozniak, Unsupervised clustering for a Randall–Sundrum Graviton at 3.5TeV narrow resonance, 5th IML workshop, May 2022**
Charged particle tracking

Graph Neural Networks for particle trajectory reconstruction

Data as a graph of connected hits
Connect hits using geometric constraints
Embedding requires large graphs (~$10^5$ nodes)
GNN for particle tracking

https://exatrkx.github.io/

arxv:2007.00149
Quantum models

Replace Edge and Node networks with **hybrid classifiers**
Quantum circuit systematics

(a) Axis of angle embedding comparison.

(b) Number of layers ($N_L$) comparison.

(c) Number of iterations ($N_I$) comparison.

(d) Hidden dimension size ($N_D = N_Q$) comparison.
Quantum SVM for Higgs classification

Classical models trained on 67 features
Test several dimensionality reduction strategies (PCA, AutoEncoder, Kmeans..)

N(train): 1827808, N(test) 456952

(b) Models trained on the original input features (67), discarding the 3 least informative ones (64).

(a) Models trained on the AE latent space features (16).
Hybrid setup for anomaly detection

Di-jet events ($\Delta \phi$, $\Delta \eta$,$p_T$). Train AE on **QCD sidebands**. Train classifiers on **signal region**.
Agent interacts with environment
• Receives reward after every action
• Learns through trial-and-error
• Training sample: \((s_t, a_t, r_t, s_{t+1}, d_t)\)

Decision making
• Agent follows policy \(\pi: S \rightarrow A\)
• Goal: find optimal policy \(\pi^*\)
• Optimal \(\Leftrightarrow\) maximizing return: \(G_t = \sum_k \gamma^k R_{t+k}\)

Expected return can be estimated through value function \(Q(s, a)\)
• Helps answering: “Best action to take in given state?”
• Not a priori known, but can be learned iteratively

Reinforcement Learning

RL book: Sutton & Barto

https://www.youtube.com/watch?v=SsJ-AustniU
https://www.youtube.com/watch?v=Lu56xVlZ40M
https://www.youtube.com/watch?v=imOt8ST4Ey
Quantum Reinforcement Learning

**Q-learning**: learn $Q(s,a)$ using function approximator

- **DQN**: Deep Q-learning (*feed-forward neural network*)
- **FERL**: Free energy based RL (*quantum Boltzmann machine*)

**Free Energy RL: clamped Quantum Boltzman Machine**

- Network of coupled, stochastic, binary units (spin up / down)
- $\hat{Q}(s,a) \approx$ negative free energy of classical spin configurations $c$
- **Sampling** $c$ using (simulated) quantum annealing
- **Clamped**: visible nodes not part of QBM; accounted for as biases
- Using 16 qubits of D-Wave Chimera graph
- Discrete, binary-encoded state and action spaces

$$\hat{Q}(s,a) \approx -F(v) = -\langle H^\text{eff}_v \rangle - \frac{1}{\beta} \sum_c P(c|v) \log P(c|v)$$
Beam optimisation in linear accelerator

- **Action**: deflection angle
- **State**: BPM position
- **Reward**: integrated beam intensity on target
- **Optimality**: what fraction of possible states does agent take the right decision

- **Training efficiency**: FERL massively outperforms classical Q-learning (8±2 vs. 320±40 steps)
- **Descriptive power**: QBM can reach high performance with much fewer weights than DQN (52 vs. ~70k)
Getting real..

- AWAKE electron beam line (10BPM)  
  [https://gitlab.cern.ch/be-op-ml-optimization/envs/awake](https://gitlab.cern.ch/be-op-ml-optimization/envs/awake)

Actor-critic Q-learning training on simulated annealing.

Successful evaluation the real beam-line (but one BPM was broken)

Stay tuned for the new result!
Quantum Circuit Born Machine

Sample from a variational wavefunction $|\psi(\theta)\rangle$ with probability given by the Born rule:

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

- Only able to generate **discrete PDFs** (continuous in the limit $\#\text{qubits} \to \infty$)
- Train using **Maximum Mean Discrepancy**:

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

with $K$ a gaussian kernel

- **Pros**: relatively easy to optimize, **Cons**: empirically less efficient than an adversarial approach

**Muon Force Carriers** predicted by several theoretical models:

- Could be detected by muon fixed-target experiments (FASER) or muon interactions in calorimeters (ATLAS)\(^1\).

Generate \(E, p_t, \eta\) of outgoing muon and MFC

---

Quantum Generative Adversarial Networks

Density estimation by comparison

- Sample-based comparison between estimated $q(x)$ and true distribution $p(x)$
- Multiple implementations, mostly classical-quantum hybrid
- Used for
  - Data generation
  - PDF loading on quantum systems
  - Anomaly detection

![Diagram of Quantum Generative Adversarial Networks](image_url)
qGAN as a data loader

Cross section integration using Quantum Amplitude Estimation
Focus on electroweak process

Data encoding in quantum states affects quality of integration
Test QGAN for data embedding and compare to direct loading

Test on \(1 + x^2\) distribution:
- 10k events, 3 qubits, circular entanglement

\[
G(\phi) |\psi_{in}\rangle = |g(\phi)\rangle = \sum_{i=0}^{N-1} \sqrt{p_g(\phi)} |i\rangle
\]

<table>
<thead>
<tr>
<th>Loading</th>
<th>Difference per bin [%] Min.</th>
<th>Max.</th>
<th>Average</th>
<th>(\sigma_x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>+0.207</td>
<td>-1.88</td>
<td>1.35</td>
<td>(1.80 \times 10^{-3})</td>
</tr>
<tr>
<td>qGAN default</td>
<td>+2.36</td>
<td>-21.1</td>
<td>8.51</td>
<td>0.0118</td>
</tr>
<tr>
<td>qGAN optimised</td>
<td>-0.995</td>
<td>-12.4</td>
<td>4.65</td>
<td>(7.00 \times 10^{-3})</td>
</tr>
</tbody>
</table>

**qGAN for event generation**

Generate Mandelstam \((s, t) + y\) variables for \(t\)-\(t\) production

Introduce a style-based approach

---

IBM Q Santiago

---

**Qubits** | \(pp \rightarrow t\bar{t}\) LHC events
---|---

| \(D_{\text{latent}}\) | 3 |
| Layers | 5 |
| Epochs | \(3 \times 10^4\) |
| Training set | \(10^4\) |
| Batch size | 128 |
| Parameters | 62 |
| \(U_{\text{ent}}\) | 2 sequential \(CR_y\) gates

---

Quantum simulator

---

Increasing generated dimensionality

Energy Profiles in Calorimeters

- Calorimeter simulation is one of the main use cases for classical GAN in HEP
- Represented as a 3D regular grid
- Reduce to:
  - 1D distribution along the calorimeter depth (8 pixel)
  - 2D distribution on the y-z plane (64 pixel)


Noise effect on ML training

- Hybrid GAN model reproducing particle energy profiles in detectors
- Training is up to ~5% readout noise tolerant
- Effect on training hyperparameters

---

Readout $p_0=1\%$

- Worst (Highest $D_{KL}$)
- Best (Lowest $D_{KL}$)
- Mean

Training Epochs vs. Relative Entropy

---

Readout $p_0=5\%$

- Worst (Highest $D_{KL}$)
- Best (Lowest $D_{KL}$)
- Mean

Training Epochs vs. Relative Entropy

---

Readout $p_0=10\%$

- Worst (Highest $D_{KL}$)
- Best (Lowest $D_{KL}$)
- Mean

Training Epochs vs. Relative Entropy

---

Importance wrt Objective Function

- $l_{rg}$
- $l_{rd}$
- $\gamma$

- $p = 0.01$
- $p = 0.05$
- $p = 0.1$

---

PDF (Inference on ibmq_manila)

- No noise
- No mitigation
- IBF mitigation
- BF mitigation

---

qGAN Benchmarks on hardware

Train models using **noisy simulator** and test the inference of the model on the superconducting (IBMQ) and trapped-ion (IONQ) quantum hardware

- For IBMQ machines, choose the qubits with the lowest CNOT gate error

<table>
<thead>
<tr>
<th>Device</th>
<th>Readout error</th>
<th>CX error</th>
<th>$D_{KL}/D_{KL,\text{ind}} (\times 10^{-2})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibmq_jakarta</td>
<td>0.028</td>
<td>1.367 · 10^{-2}</td>
<td>0.14 ± 0.14</td>
</tr>
<tr>
<td>ibm_lagos</td>
<td>0.01</td>
<td>5.582 · 10^{-3}</td>
<td>0.26 ± 0.11</td>
</tr>
<tr>
<td>ibmq_casablanca</td>
<td>0.026</td>
<td>4.58 · 10^{-2}</td>
<td>4.03 ± 1.08</td>
</tr>
<tr>
<td>IONQ</td>
<td>NULL</td>
<td>1.59 · 10^{-2}</td>
<td>1.24 ± 0.74</td>
</tr>
</tbody>
</table>

Figure 4: Mean (a,c) and individual images (b,d) obtained by inference test on ibmq_jakarta (a,b) and IONQ (c,d).
Research on QML applications in High Energy Physics is producing a large number of prototypes

- So far focus on different steps of data processing in «controlled environment»
- Some preliminary hints of advantage in terms of input feature size and representational power
- Mostly we do «as good as classical methods»
- Need more robust studies to relate quantum model architecture and performance to data sets
- Identify use cases where quantum approach could be more effective than classical machine/deep learning
- Studying QML algorithms today can build links between QC and learning theory
Thanks!
Sofia.Vallecorsa@cern.ch

https://quantum.cern/
Equivalent interpretations?

Characterize the behaviour of different models, similarity and links among them and link to data properties.

Ex:

- Data Re-Uploading circuits: alternating data encoding and variational layers.
  - Represented as explicit linear models (variational) in larger feature space
  → can be reformulated as implicit models (kernel)
- Representer theorem: implicit models achieve better accuracy
  - Explicit models exhibit better generalization performance

See M. Grossi summary at the 2022 CERN Openlab Technical Workshop: https://indico.cern.ch/event/1100904/contributions/4775169/
Given the size of the Hilbert space a compromise between expressivity, convergence and generalization performance is needed.

Classical gradients vanish exponentially with the number of layers (J. McClean et al., arXiv:1803.11173)

- Convergence still possible if gradients consistent between batches.

Quantum gradient decay exponentially in the number of qubits

- Random circuit initialization
- Loss function locality in shallow circuits (M. Cerezo et al., arXiv:2001.00550)
- Noise induced barren plateau (Wang, S et al., Nat Commun 12, 6961 (2021))

TTN for MNIST classification (8 qubits), Zhang et al., arXiv:2011.06258