Quantum Generative Models for High Energy Physics.

FhSciComp 2026

Saverio Monaco

June 17, 2026

Roadmap.

Motivation for Quantum Generative Models in HEP

Q: Is there an *advantage* in using QC in HEP?

Introduction to Quantum Computing

Q: How to Process Information on a Quantum Computer?

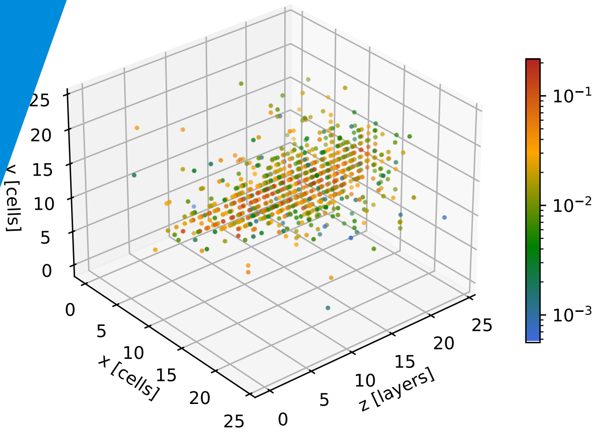
Quantum Machine Learning (+ Examples)

Q: How to make Quantum Computers learn?

Conclusions

Motivation

Quantum Computing in HEP



Use Cases of Quantum Computing.

in HEP

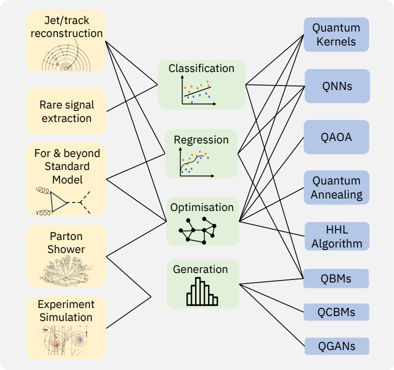


Figure: Proposed HEP experimental challenges (orange) with corresponding approaches (green) and quantum algorithms (blue) [1]

Quantum Computing offers a different frame of computations

↓

Potential useful advantage in computationally expensive tasks

Classical computers face fundamental bottlenecks in HEP, e.g.:

- Simulating complex quantum systems
- Combinatorial optimization problems

Use Cases of Quantum Computing.

Example #1: Track Reconstruction

Track Reconstruction: leveraging quantum interference to tackle combinatorial optimization.

Problem: a large number of detector hits leads to an exponentially large number of possible track combinations

Idea: encode candidate paths as quantum states, so that correct tracks interfere **constructively** while wrong combinations cancel out

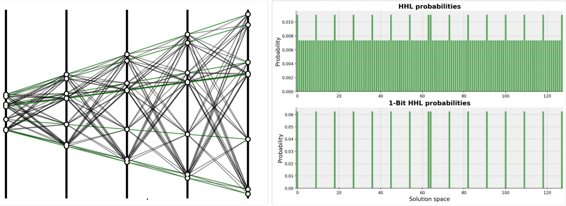


Figure: Xenofon et. al. [2]

Use Cases of Quantum Computing.

Example #2: Calorimeter simulation

Calorimeter Simulation: exploiting the Born rule to sample from learned distributions:

$$|\psi\rangle = \sum_i c_i |x_i\rangle, \qquad P(x_i) = |c_i|^2$$

Problem: calorimeter showers obey highly complex, high-dimensional probability distributions that are prohibitively expensive to simulate step-by-step

Idea: train a quantum generative model to **sample directly** by measuring the quantum state.

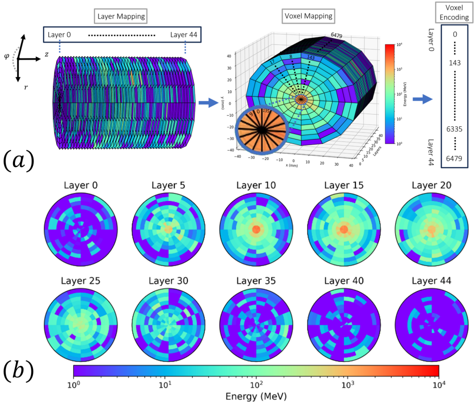


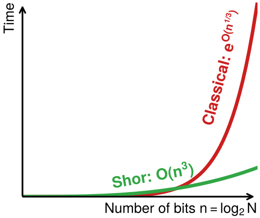
Figure: Toledo-Marín et. al. [3]

Use Cases of Quantum Computing.

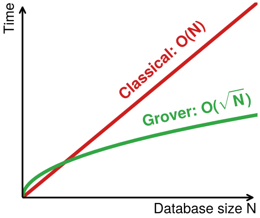
Sought advantages

Scaling: better growth **for some problems** of resource cost with problem size

Factoring: exponential speedup



Unsorted search: polynomial speedup



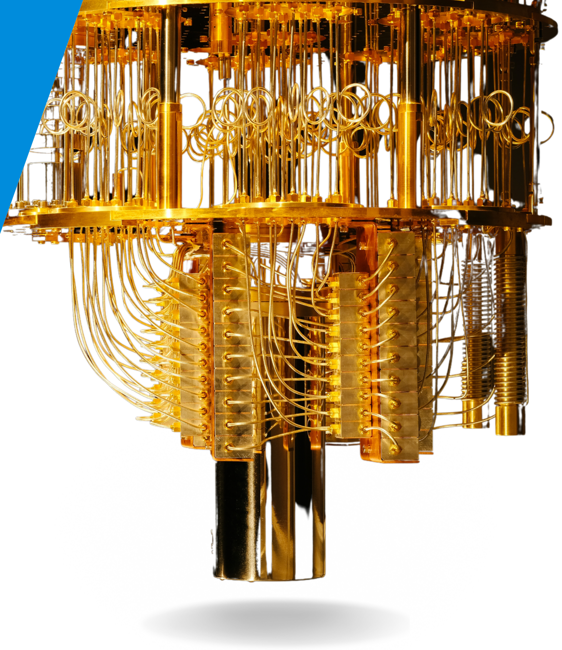
Expressivity:

a state $|\psi\rangle \in \mathcal{H}$ on n qubits lives in a space of dimension $\dim \mathcal{H} = 2^n$, exponentially larger than the n bits available classically.

Quantum Advantage: choosing a quantum implementation over a classical one in practice, given a faster algorithm, the **hardware** to run it, and **access** to that hardware.

Quantum Computing

How to Process Information on a Quantum Computer?



Quantum Computing.

Fundamentals

Definition: A **Quantum Computer** is any device capable of preparing and accurately evolving a quantum state (typically represented as a spin system).

Abstract Level: A qubit can be represented as a two-level system:

- Ground state

  $|0\rangle \rightarrow$ (Spin down)

- Excited State

  $|1\rangle \rightarrow$ (Spin up)

Actual Level: Depends on the implementation:

- Superconducting
  - Josephson junction
- Trapped Ion
  - Individual ions trapped using electromagnetic field
- Photonic
  - Polarization of photons
- ...

Quantum Computing.

Running on Simulators vs Quantum Hardware

Classical Simulator:

- ✔ Accessible
- ✖ Different levels of accuracy: noiseless, shot noise, gate noise, ...
- ✖ Up to 20 qubits

Common tools: Qiskit, Pennylane, ...

Quantum Hardware

- ✖ Requires special access agreements and queuing

Hardware:

- IQM : Superconducting;
- IBM : Superconducting;
- eleQtron : Trapped Ions;

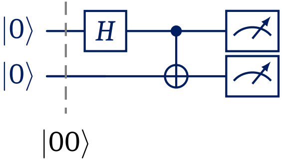
Workflow:

Test ideas intensively on simulators first, then evaluate them on quantum hardware

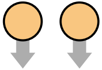
Circuit representation.

Bell state

Creation of a *Bell state* (maximally entangled two-qubit quantum state)



$$|\psi\rangle = |00\rangle \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



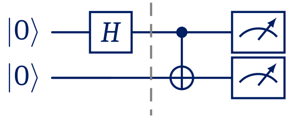
$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\boxed{H} \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

Circuit representation.

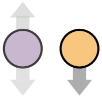
Bell state

Creation of a *Bell state* (maximally entangled two-qubit quantum state)



$$\frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right)$$

$$|\psi\rangle = (H \otimes I)|00\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}$$



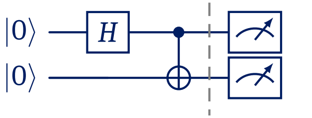
$$\equiv \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

$$\boxed{H} \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}+1 & -1\\ -1 & +1\end{pmatrix}$$

Circuit representation.

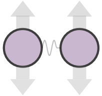
Bell state

Creation of a *Bell state* (maximally entangled two-qubit quantum state)



$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

$$|\psi\rangle = CNOT(H \otimes I)|00\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$



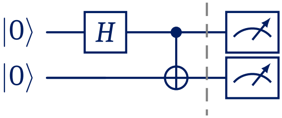
$$\equiv \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$$

$$\equiv \frac{1}{\sqrt{2}}\begin{pmatrix}+1 & -1\\-1 & +1\end{pmatrix}$$

Circuit representation.

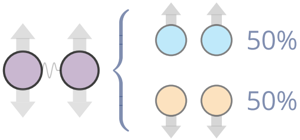
Bell state

Creation of a *Bell state* (maximally entangled two-qubit quantum state)



$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

$$|\psi\rangle = CNOT(H \otimes I)|00\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

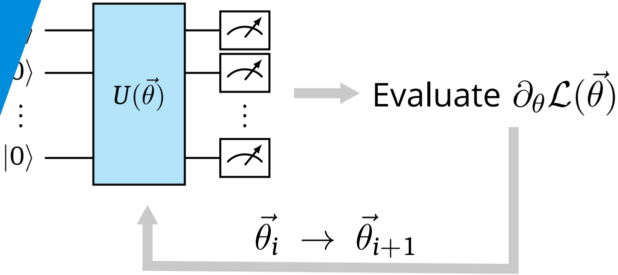


$$\equiv \begin{pmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{pmatrix}$$

$$\boxed{H} \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}+1 & -1\\-1 & +1\end{pmatrix}$$

Quantum Machine Learning

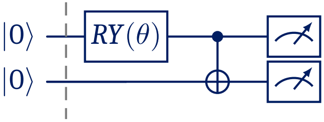
Training models on quantum circuits



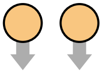
Quantum Machine Learning.

Parametrized Bell state

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions



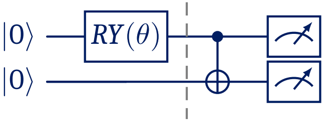
$$|\psi\rangle = |00\rangle \equiv \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



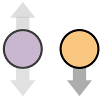
$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\boxed{RY(\theta)} \equiv \begin{pmatrix} +\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & +\cos(\frac{\theta}{2}) \end{pmatrix}$$

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions



$$|\psi\rangle = (RY(\theta) \otimes I)|00\rangle = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \end{pmatrix}$$

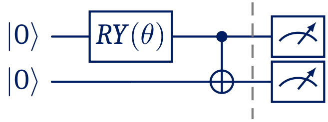


$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \boxed{RY(\theta)} \equiv \begin{pmatrix} +\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & +\cos(\frac{\theta}{2}) \end{pmatrix}$$

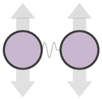
Quantum Machine Learning.

Parametrized Bell state

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions



$$|\psi\rangle = CNOT(RY(\theta) \otimes I)|00\rangle = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \end{pmatrix}$$



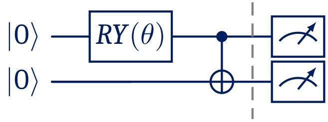
$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\boxed{RY(\theta)} \equiv \begin{pmatrix} +\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & +\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

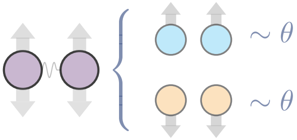
Quantum Machine Learning.

Parametrized Bell state

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions



$$|\psi\rangle = CNOT(RY(\theta) \otimes I)|00\rangle = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\theta}{2}\right) \end{pmatrix}$$

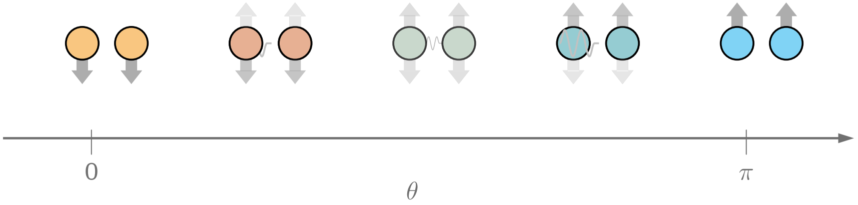


$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad -\boxed{RY(\theta)}- \equiv \begin{pmatrix} +\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & +\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions

Superposition of $|00\rangle$ and $|11\rangle$ depending on θ

$$|\psi(\theta)\rangle = \cos\frac{\theta}{2}\,|00\rangle + \sin\frac{\theta}{2}\,|11\rangle$$



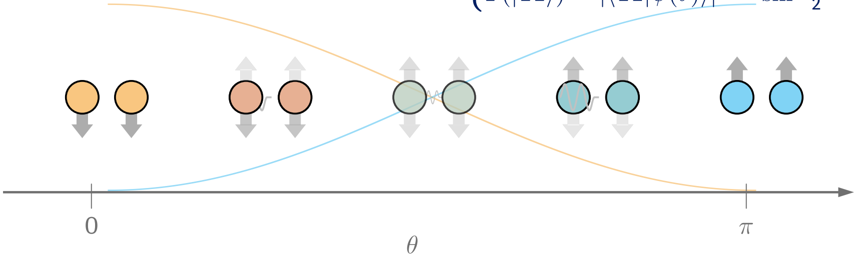
Quantum Machine Learning.

Parametrized Bell state

Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions

Superposition of $|00\rangle$ and $|11\rangle$ depending on θ

$$|\psi(\theta)\rangle = \cos\frac{\theta}{2}\,|00\rangle + \sin\frac{\theta}{2}\,|11\rangle \qquad \begin{cases} \mathbb{P}(|00\rangle) = |\langle 00|\psi(\theta)\rangle|^2 = \cos^2\frac{\theta}{2} \\ \mathbb{P}(|11\rangle) = |\langle 11|\psi(\theta)\rangle|^2 = \sin^2\frac{\theta}{2} \end{cases}$$

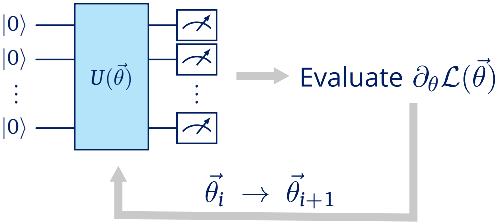


Idea: combine parametrized single-qubit gates RX(θ), RY(θ), RZ(θ) with multi-qubit entangling gates, so the circuit can represent complex, correlated functions

Goal: find the set of parameters $\vec{\theta}$ for the *parametrized unitary* $U(\vec{\theta})$ that minimizes a loss function: $\mathrm{argmin}_{\vec{\theta}}\left[\mathcal{L}(U(\vec{\theta}))\right]$.

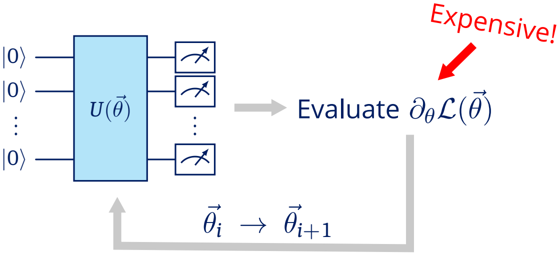
Pipeline:

1. Select loss function and ansatz unitary $U(\cdot)$
2. Training loop:
   - 2.1 Evolve circuit under $U(\vec{\theta})$
   - 2.2 Perform measurements
   - 2.3 Evaluate gradients*
   - 2.4 Update parameters via optimizer



Quantum Machine Learning.

Obstacle #1: No Backpropagation



Gradients cannot be computed efficiently.

Efficient gradient estimation exists only for **a limited class of quantum circuits** [4].

Parameter-Shift rule: gradient estimation technique restricted to expectation values of observables, with poor scalability (circuit executions: $2 \times N \times N_{\text{param}} \times N_{\text{shots}}$).
Estimated time for a single training epoch of gradient descent on MNIST [5]: **38 years**.

Barren Plateau: exponential vanishing of gradients as the number of qubits N grows, rendering parameter updates increasingly uninformative [6, 7].

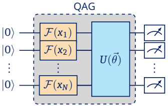
Barren Plateau $\sim \mathcal{O}(b^{-N})$

Quantum Machine Learning.

Examples from the group

Goal: Generating calorimeter shower events using quantum circuits.

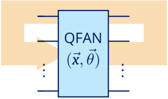
Quantum Angle Generator



Fully quantum generative model, limited scalability

Paper Arxiv

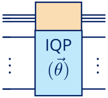
Quantum Feature Amplification



Hybrid autoregressive generative model, high scalability

Paper Arxiv
Code

Compiled IQP Circuit



Classically trainable sampling-type generative model for continuous variable distributions

Paper Arxiv
Code

...

Quantum Machine Learning.

Compiled IQP Circuit: Slim et al. "An IQP Born Machine for Calorimeter Image Generation at 64 Qubits with Compiled-IQP Deployment"

Idea: Making a sampling-type quantum generative model for calorimeter dataset that can entirely be trained on classical computers, while keeping the output (samples) *quantum-hard*.

Tradeoff:

- ✔ Encoding of continuous variables
- ✔ Correlation between variables can be trained
- ✖ Loss function more expensive

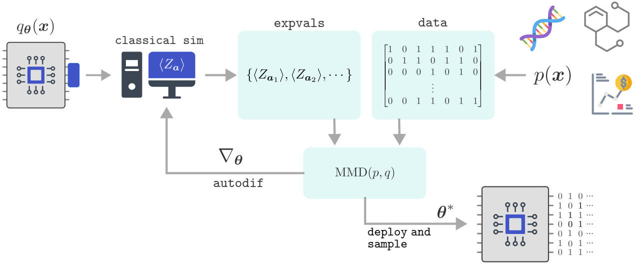


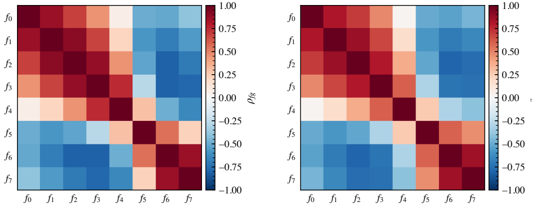
Figure: Recio-Armengol et al.[5]

Quantum Machine Learning.

Compiled IQP Circuit: Slim et al. "An IQP Born Machine for Calorimeter Image Generation at 64 Qubits with Compiled-IQP Deployment"

Results:

- Trained a quantum generative model on real calorimeter shower data, largest such demonstration to date (64 qubits)
- New training method (PSCK) better captures correlations between detector layers than previous approaches
- Result lands within $\sim$0.02 of the best possible reconstruction given our data encoding



(a) Ground truth ρ_{fg}^{data} (b) PSCK–MMD ρ_{fg}^{PSCK}

Figure: Slim et al. [8]

Before this, correlation were notoriously hard to learn on a IQP model!

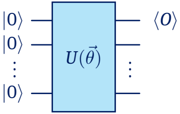
Symbolic Pauli Propagation: Monaco et al. "Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits"

Symbolic Pauli Propagation enables efficient classical approximation of expectation values for large quantum circuits.

Single quantum state collapse = hard!

Expectation values = classically tractable!

Outline in QML:

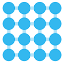


Propagate → Train → Deploy (*ibm_berlin*)

$\langle O\rangle(\boldsymbol{\theta}) = \sum_x c_x \prod_{i\in S_x} f_{x,i}(\theta_i)$

$\mathcal{L}(\theta) = g\left(\langle O\rangle(\boldsymbol{\theta})\right)$

ADAM: $\vec{\theta}_t \to \vec{\theta}_{t+1}$



Step 1: Define the ansatz and express the loss function as expectation values of Pauli observables.

Step 2: Apply Pauli Propagation to obtain an explicit expression for the expectation value in terms of the circuit parameters.

Step 3: Apply ADAM to optimize the circuit parameters classically.

Step 4: Deploy the optimized circuit on a quantum device.

Symbolic Pauli Propagation: Monaco et al. "Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits"

Application # 1: Variational Quantum Eigensolver on a 64-qubits 2D Ising model

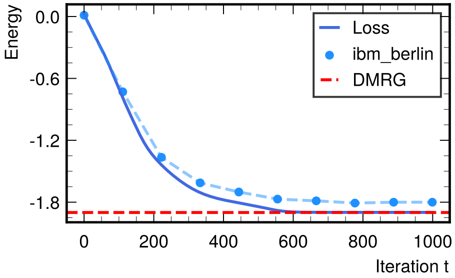
Objective:

Find the unitary $U(\theta)$ that evolves the state $|0\rangle$ to the ground state of the Hamiltonian H.

Loss:

$$\mathcal{L}(\theta) = -\sum_{\langle i,j\rangle} J\langle Z(i)Z(i+1)\rangle + \sum_i h\langle X(i)\rangle$$

Result:



Application # 2: Sampling Type Generative Training on 8×8 binarized MNIST digits images

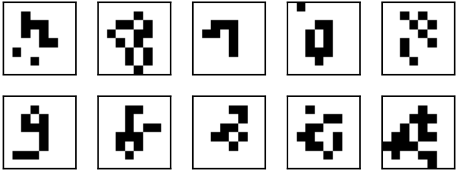
Objective:

Train the unitary $U(\theta)$ such that $U(\theta)\,|0\rangle$ is a superposition of possible images.

Loss:

$$\mathcal{L}_{\text{MMD}}(\vec{\theta}) = \frac{1}{|\mathcal{A}|}\sum_j \left(\langle Z_{a_j}\rangle_{\text{data}} - \langle Z_{a_j}(\vec{\theta})\rangle\right)^2$$

Result:



Limited hardware + Complex task: evolving the state into a superposition of all possible images!

Conclusion

Final Remarks

Conclusion.

State of QML for HEP

	Non-QML (HHL, QAOA)	**QML**
Proven Advantage	Yes (theoretical, some tasks)	No
Hardware barrier	Yes (noise, connectivity)	Partially
Group’s approach	—	Hybrid / Classically Trainable

HEP provides a natural testbed for quantum implementations: its datasets are highly-dimensional and quantum-mechanical in origin, and its tasks are mainly combinatorial or distribution-learning problems can benifit from quantum-speedup or expressivity.

Conclusion.

Open challenges & Outlook

Open Challenges:

- **Hardware:** fault-tolerant devices with sufficient connectivity and coherence needed before non-QML algorithms reach HEP scale
- **Gradient estimation:** no efficient backpropagation on quantum hardware
- **Barren plateaus:** no general mitigation strategy for deep, wide ansätze
- **Scalability:** moving beyond 64 qubits toward full calorimeter granularity and conditional event generation

Resources:

- Group's GitHub

Our works:

- Precise Image Generation on Current Noisy Quantum Computing Devices
- Quantum Feature Amplification Network (QFAN) as An Autoregressive Quantum Generative Model
- An IQP Born Machine for Calorimeter Image Generation at 64 Qubits with Compiled-IQP Deployment
- Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits

Thank you!

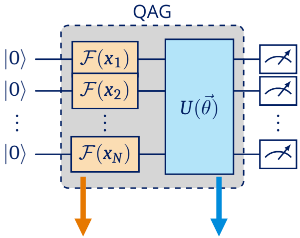
Saverio Monaco
saverio.monaco@desy.de

ENGAGE has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement No. 101034267.

Quantum Angle Generator.

The model

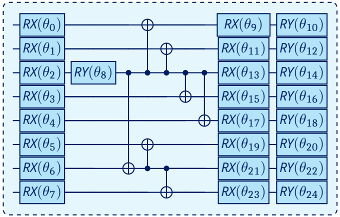
- Generation of N pixels requires N qubits



Quantum State Preparation

- Implement randomness using RY rotations

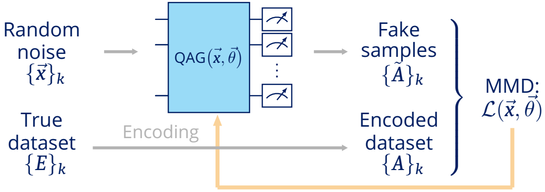
Parametrized Unitary $U(\vec{\theta})$

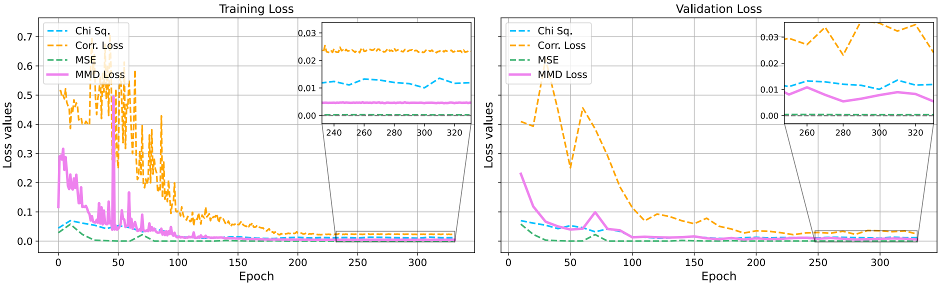


Quantum Angle Generator.

Training

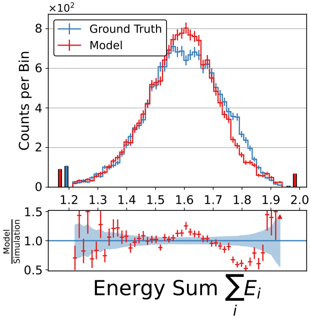
✔ MMD loss is sufficient to learn correlations between pixels

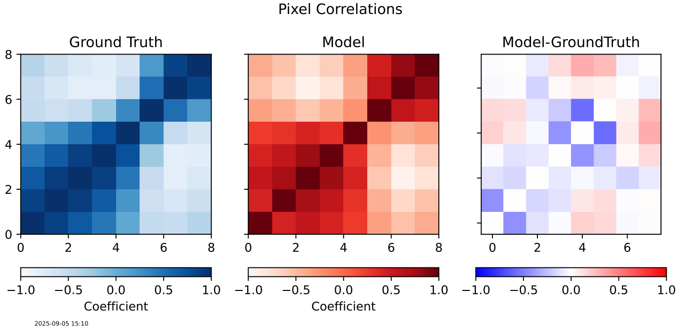




Quantum Angle Generator.

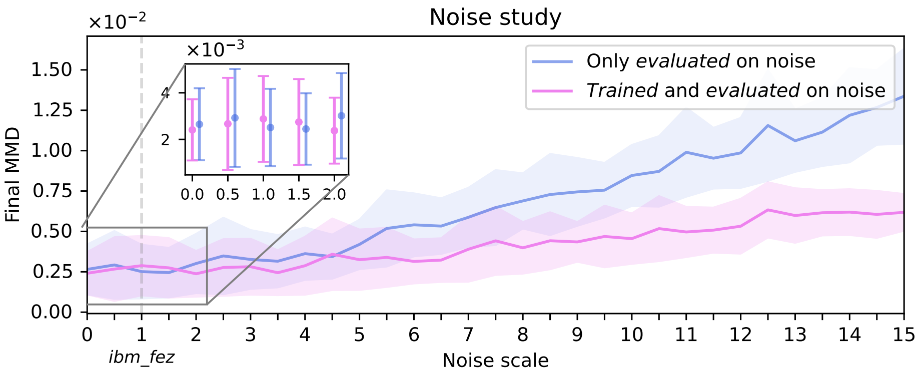
Energy sum & Correlation





Quantum Angle Generator.

Noise study



The model **adapts to the underlying noise**.

Current noise levels on real QPUs do not degrade performance.

Parameter Shift.

Performing a single epoch of gradient descent on MNIST

1. Precision Requirement: To estimate a loss bounded in $[0, 1]$ to a precision of $\epsilon = 0.001$, we need

$$\epsilon^{-2} = (0.001)^{-2} = 10^6 \quad \text{samples per evaluation.}$$

2. Parameter Shift Evaluations: Each parameter requires two circuit evaluations ($\theta + s$ and $\theta - s$). For $p = 10\,000$ parameters:

$$10\,000 \times 2 \times 10^6 = 2 \times 10^{10} \text{ samples per gradient evaluation.}$$

3. Scaling to the MNIST Dataset: With $\approx 60\,000$ training images, one epoch requires:

$$2 \times 10^{10} \times 60\,000 \approx 1.2 \times 10^{15} \text{ samples.}$$

Time Estimate: At a sample rate of 1 MHz (10^6 samples/second):

$$\frac{1.2 \times 10^{15}}{10^6} = 1.2 \times 10^9 \text{ seconds} \approx \frac{1.2 \times 10^9}{31.5 \times 10^6} \approx 38 \text{ years.}$$

Classical Simulation.

Simulation Methods

Simulation methods enable the classical simulation of restricted regions of the Hilbert space.

If something is classically simulable
$\longrightarrow$ no quantum advantage

Key insight in QML: simulate the loss function classically while maintaining a *quantum-hard* circuit output [5]

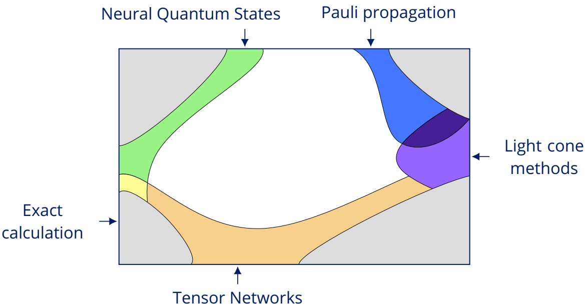


Figure: Recreated after a slide presented by Z. Holmes, NORDITA, 2026.

Remarks.

Are hard distributions *useful*?

Do we need to draw from distributions that are **provably hard** to sample classically?

Establishing how hard calorimeter images are to generate classically is a necessary prerequisite for claiming quantum advantage in calorimeter simulation.

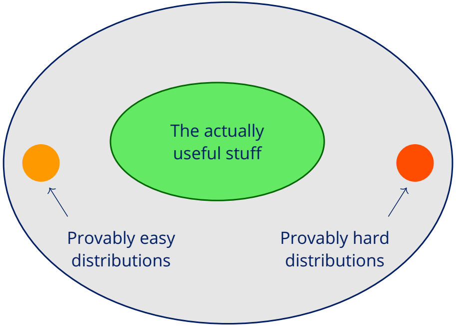


Figure: Adapted after a slide presented by J. Bowles, NORDITA, 2026.

References I.

[1] Alberto Di Meglio, Karl Jansen, Ivano Tavernelli, Constantia Alexandrou, Srinivasan Arunachalam, Christian W Bauer, Kerstin Borras, Stefano Carrazza, Arianna Crippa, Vincent Croft, et al.
Quantum computing for high-energy physics: State of the art and challenges.
Prx quantum, 5(3):037001, 2024.

[2] Xenofon Chiotopoulos, Miriam Lucio Martinez, Davide Nicotra, Jacco A. de Vries, and K.
Trackhhl: A quantum computing algorithm for track reconstruction at the lhcb, 2025.

References II.

[3] J Quetzalcoatl Toledo-Marin, Sebastian Gonzalez, Hao Jia, Ian Lu, Deniz Sogutlu, Abhishek Abhishek, Colin Gay, Eric Paquet, Roger Melko, Geoffrey C Fox, et al.
Conditioned quantum-assisted deep generative surrogate for particle-calorimeter interactions.
arXiv preprint arXiv:2410.22870, 2024.

[4] Joseph Bowles, David Wierichs, and Chae-Yeun Park.
Backpropagation scaling in parameterised quantum circuits, 2025.

[5] Erik Recio-Armengol, Shahnawaz Ahmed, and Joseph Bowles.
Train on classical, deploy on quantum: scaling generative quantum machine learning to a thousand qubits.
arXiv preprint arXiv:2503.02934, 2025.

References III.

[6] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, and Lukasz.
Barren plateaus in variational quantum computing, 2025.

[7] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven.
Barren plateaus in quantum neural network training landscapes, 2018.

[8] Jamal Slim, Saverio Monaco, Florian Rehm, Dirk Krücker, and Kerstin Borras.
An iqp born machine for calorimeter image generation at 64 qubits with compiled-iqp deployment, 2026.

[9] Quantum@DESY.
DESY-QML: Code repositories of the CMS DESY quantum group.
GitHub organization, `https://github.com/desyqml`, 2026.
Accessed: 2026-06-12.

References IV.

[10] Saverio Monaco.
DESY-SINTEF: A Beamer theme for DESY presentations.
https://github.com/SaverioMonaco/DESY-SINTEF, 2025.
LaTeX/Beamer template, GPL-3.0. Adapted from the UIC Beamer template by U. Muneeb, https://github.com/usamamuneeb/uic-beamer-template.