DESY. QUANTUM

RWTH AACHEN UNIVERSITY

H2020 Engage Conference:

Detector Simulation and Jet Clustering for HL-LHC with Quantum Computing

Saverio Monaco

June 19, 2025

Status of the PhD

Progress Overview

Status of the PhD

Overview

Supervisor: Kerstin Borras

Title: Development of Quantum Computing Models for Calorimeter Simulations and Jet-Clustering in Particle Physics

03/2024 - 03/2027 (?)

Status of the PhD

Overview

Supervisor: Kerstin Borras

Title: Development of Quantum Computing Models for Calorimeter Simulations and Jet-Clustering in Particle Physics

03/2024 - 03/2027 (?)

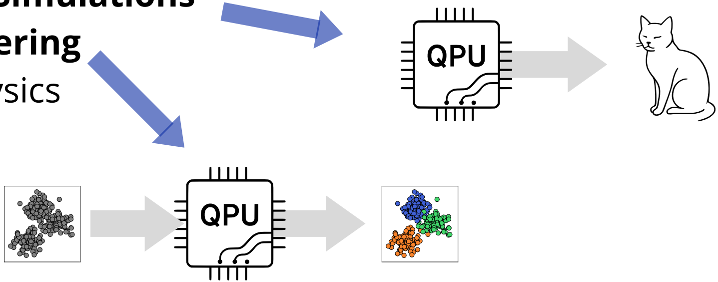
Progress:

Status of the PhD

Milestones

Title: Development of Quantum Computing Models for
Calorimeter Simulations
and **Jet-Clustering**
in Particle Physics

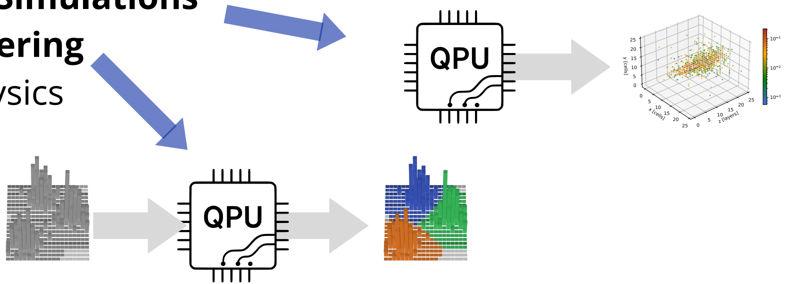
Title: Development of Quantum Computing Models for **Calorimeter Simulations** and **Jet-Clustering** in Particle Physics



Status of the PhD

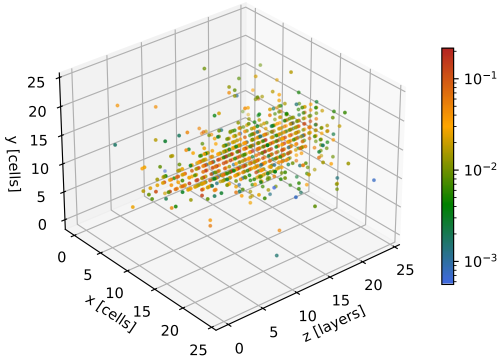
Milestones

Title: Development of Quantum Computing Models for **Calorimeter Simulations** and **Jet-Clustering** in Particle Physics



Motivation

A brief motivation for Quantum Models in High Energy Physics.



Tom’s Hardware article

IBM is building a large-scale quantum computer that 'would require the memory of more than a quindecillion of the world's most powerful supercomputers' to simulate

News

By Mark Tyson published 3 days ago

That's
1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000

Comments (20)

Motivation

Taming the Quantum Hype

Tom’s Hardware article

IBM is building a large-scale quantum computer that 'would require the memory of more than a quindecillion of the world's most powerful supercomputers' to simulate

News By Mark Tyson published 3 days ago

That's 1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000

Comments (20)

$= 10^{48}$

Motivation

Taming the Quantum Hype

Tom's Hardware article

GUARANTEE
TECHNICALLY
TRUE

IBM is building a large-scale quantum computer that 'would require the memory of more than a quindecillion of the world's most powerful supercomputers' to simulate

News By Mark Tyson published 3 days ago

That's

1,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000

$= 10^{48}$

Comments (20)

Motivation

Taming the Quantum Hype

Representation of a quantum state

- A quantum state of n qubits is represented by a complex vector of 2^n amplitudes.
- On a classical machine, each amplitude (a complex number) needs:

  32 bytes for the real part + 32 bytes for the imaginary part = 64 bytes

- Hence to represent a state of n qubits, we need 2^n amplitudes, which amounts to $64 \cdot 2^n$ bytes

Motivation

Taming the Quantum Hype

IBM Starling (2029) - 200 (logical) qubits

- A quantum state of 200 qubits is represented by a complex vector of 2^{200} amplitudes.
- On a classical machine, each amplitude (a complex number) needs:

  32 bytes for the real part + 32 bytes for the imaginary part = 64 bytes

- Hence to represent a state of 200 qubits, we need 2^{200} amplitudes, which amounts to $64 \cdot 2^{200}$ bytes $\sim 10^{43}$ EB

Motivation

Taming the Quantum Hype

IBM Starling (2029) - 200 (logical) qubits

- A quantum state of 200 qubits is represented by a complex vector of 2^{200} amplitudes.
- On a classical machine, each amplitude (a complex number) needs:

  32 bytes for the real part + 32 bytes for the imaginary part = 64 bytes

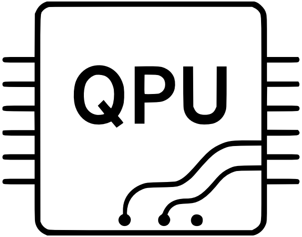
- Hence to represent a state of 200 qubits, we need 2^{200} amplitudes, which amounts to $64 \cdot 2^{200}$ bytes $\sim 10^{43}$ EB

El Captain (Current best supercomputer): $\sim 10^{-2}$ EB

$$\frac{\text{"Memory" IBM Starling}}{\text{"Memory" El Captain}} = \frac{10^{43}\text{ EB}}{10^{-2}\text{ EB}} = 10^{45}$$

Motivation

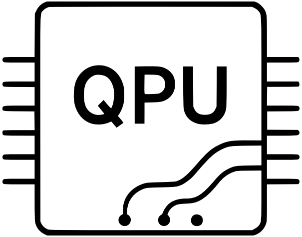
Quantum Computer Bottleneck



$$\underbrace{|0\rangle^{\otimes n}}_{\text{dummy state}} \rightarrow |\psi\rangle \rightarrow \{0,1\}^n$$

Motivation

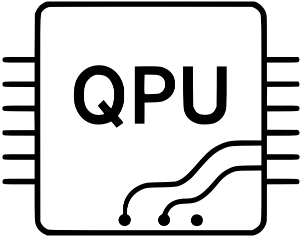
Quantum Computer Bottleneck



$$|0\rangle^{\otimes n} \rightarrow \underbrace{|\psi\rangle}_{\text{wavefunction: impossible to describe classically for large } n} \rightarrow \{0,1\}^n$$

Motivation

Quantum Computer Bottleneck

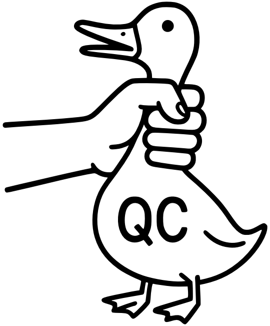


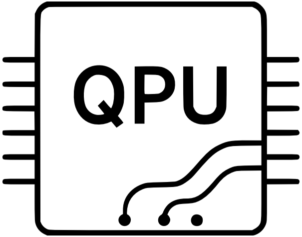
$$|0\rangle^{\otimes n} \to |\psi\rangle \to \underbrace{\{0, 1\}^n}_{\substack{\text{output / measurement} \\ \text{bitstrings of } n \text{ qubits}}}$$

Motivation

Quantum Computer Bottleneck

Measurement bottleneck





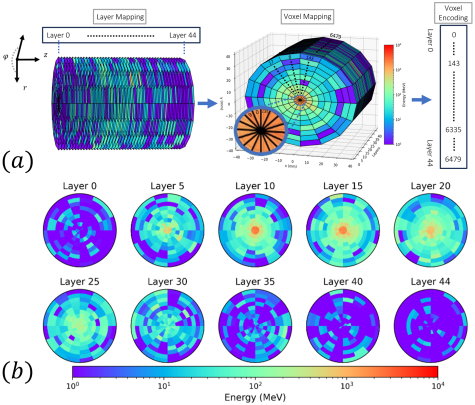
$$|\psi\rangle \rightarrow \{0,1\}^n$$

collapse of the wavefunction

Motivation

Quantum Computing in HEP

Argument #1: Complexity of HEP datasets

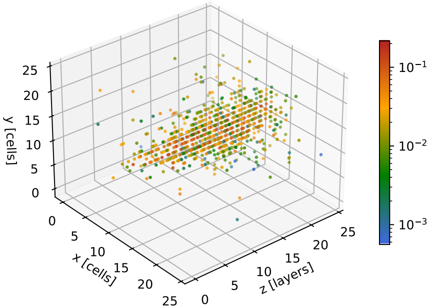


[5] Toledo-Marin et al.

Potential Advantages:

- Variational models
  - Faster solution times
  - Improved generalization capabilities [2]
  - Reduced number of required parameters
- Standard Quantum Algorithms
  - Access to a family of different algorithms with different scalings:
    ex: Shor: Exponential $\rightarrow$ *Polynomial*

Argument #2: HEP datasets are *quantum* (in a way)

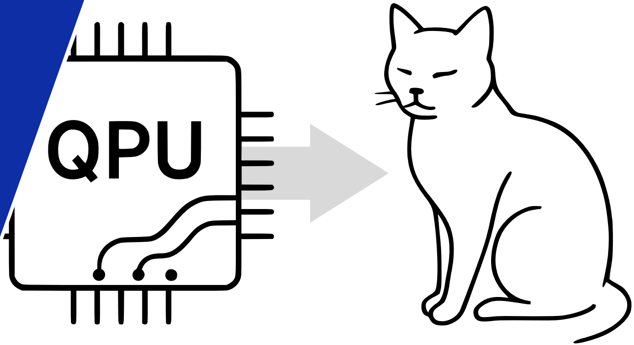


Example: Detector Dataset

A particle shower can be seen as the collapse of a complex wave-function which might be best described through a quantum state.

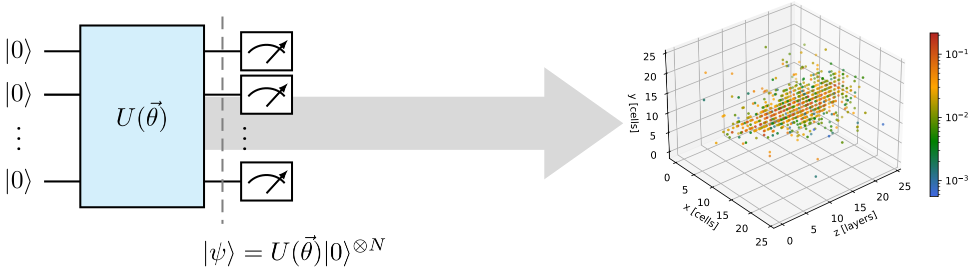
Detector Simulation

How to generate images from quantumness



Detector Simulation

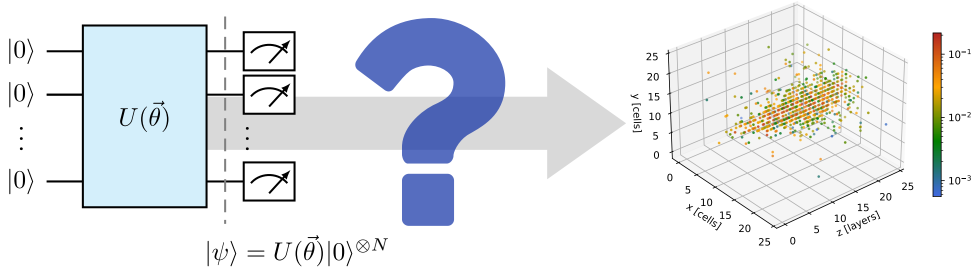
Mapping of the generative task



Detector Simulation

Mapping of the generative task

Not a trivial task!

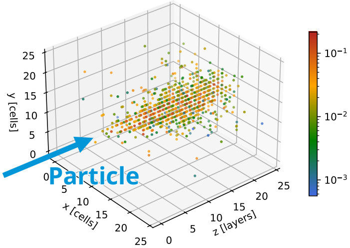


Detector Simulation

Downsampling

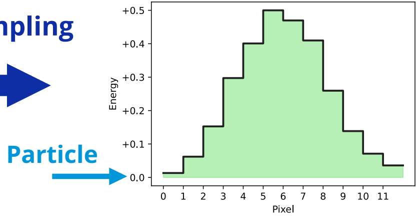
- Preliminary study using simplified models
- Understand advantages and challenges

$(25 \times 25 \times 51)$



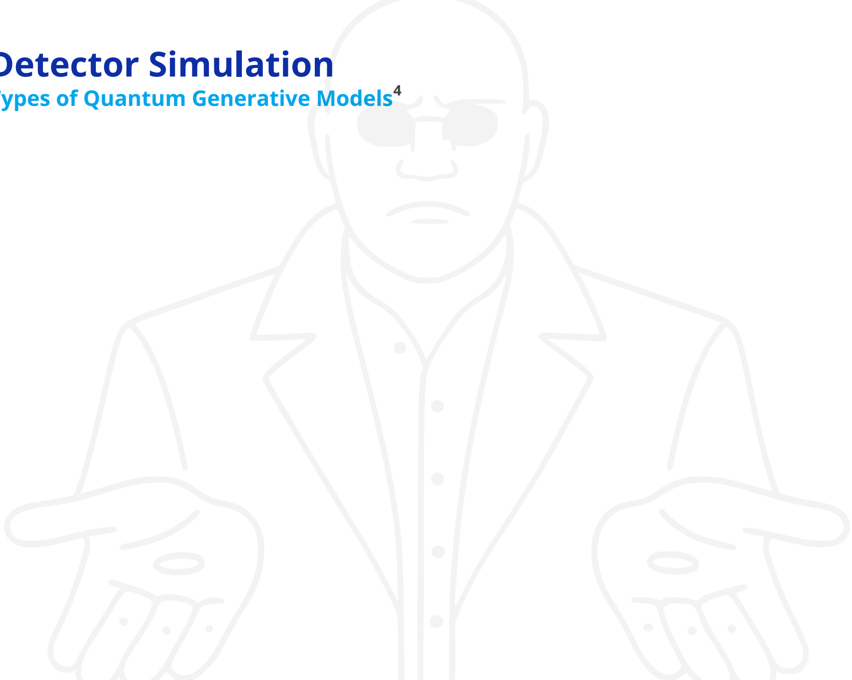
Downsampling

$(1 \times 1 \times 12)$



Detector Simulation

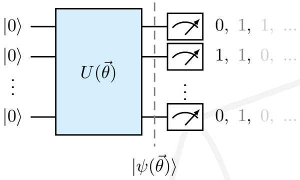
Types of Quantum Generative Models[4]



Detector Simulation

Types of Quantum Generative Models[4]

Discrete Architectures



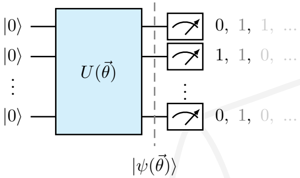
Output: Single measurement of the wavefunction

- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

Detector Simulation

Types of Quantum Generative Models[4]

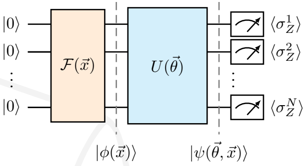
Discrete Architectures



Output: Single measurement of the wavefunction

- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

Continuous Architectures



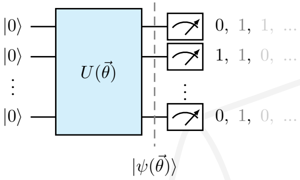
Output: Multiple measurements of the wavefunction

- ✗ Slow generations
- ✗ Generally classically simulable[1]
- ✓ Robust to noise
- ✓ *Grayscale* images

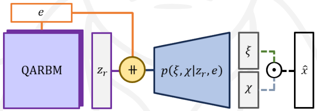
Detector Simulation

Types of Quantum Generative Models[4]

Discrete Architectures

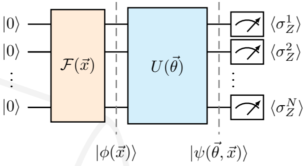


Hybrid Architectures



[5] Toledo-Marin et al.

Continuous Architectures



Output: Single measurement of the wavefunction

- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

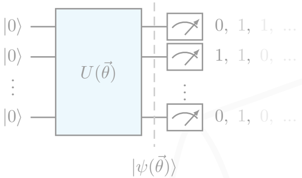
Output: Multiple measurements of the wavefunction

- ✗ Slow generations
- ✗ Generally classically simulable[1]
- ✓ Robust to noise
- ✓ *Grayscale* images

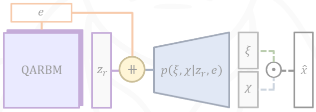
Detector Simulation

Types of Quantum Generative Models[4]

Discrete Architectures

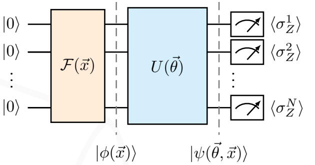


Hybrid Architectures



[5] Toledo-Marin et al.

Continuous Architectures



Output: Single measurement of the wavefunction

- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

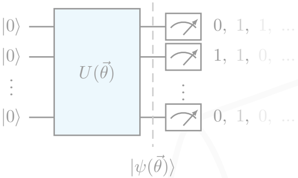
Output: Multiple measurements of the wavefunction

- ✗ Slow generations
- ✗ Generally classically simulable[1]
- ✓ Robust to noise
- ✓ *Grayscale* images

Detector Simulation

Types of Quantum Generative Models[4]

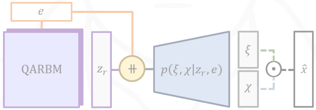
Discrete Architectures



Output: Single measurement of the wavefunction

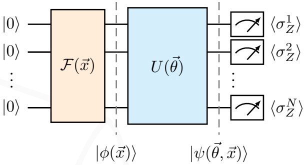
- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

Hybrid Architectures



[5] Toledo-Marin et al.

Continuous Architectures



Output: Multiple measurements of the wavefunction

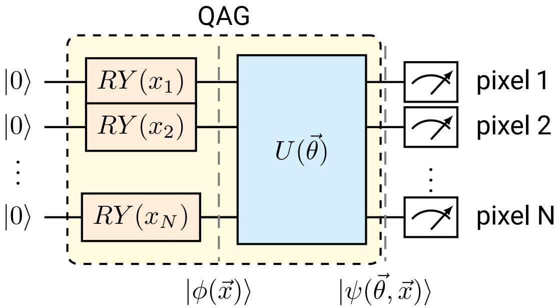
- ✗ Slow generations
- ✗ Generally classically simulable[1]
- ✓ Robust to noise
- ✓ *Grayscale* images

Calorimeter images ≡

Detector Simulation

Quantum Angle Generator

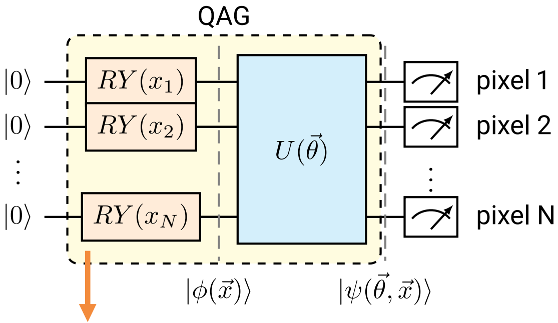
- Generation of N pixels requires N qubits



Detector Simulation

Quantum Angle Generator

- Generation of N pixels requires N qubits

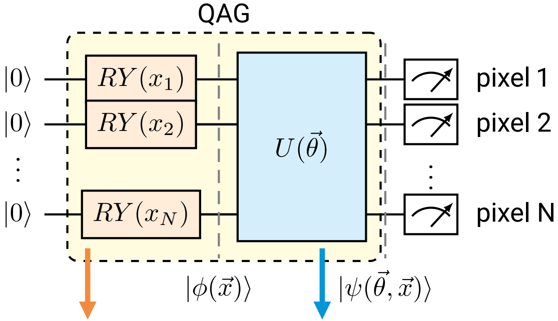


Quantum State Preparation

- Implement randomness using RY rotations

Detector Simulation
Quantum Angle Generator

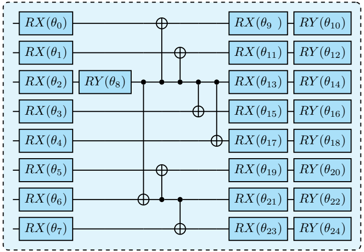
- Generation of N pixels requires N qubits



Quantum State Preparation

- Implement randomness using RY rotations

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



Detector Simulation

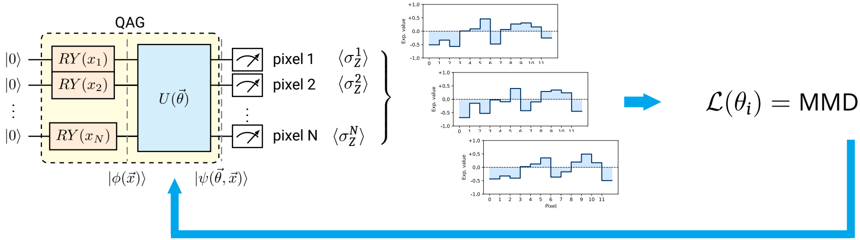
Quantum Angle Generator

Choose a random initial set of parameters $\vec{\theta}_0$

For each epoch i

1. Generate M images
2. Evaluate the *MMD* loss
3. Update the parameters $\theta_i \to \theta_{i+1}$

M images (epoch 0)



$$\mathcal{L}(\theta_i) = \text{MMD}^2(\text{True Data}, \text{Gen. Data}(\theta_i))$$

$\theta_i \to \theta_{i+1}$

Detector Simulation

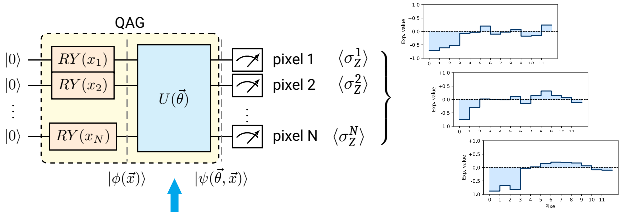
Quantum Angle Generator

Choose a random initial set of parameters $\vec{\theta}_0$

For each epoch i

1. Generate M images
2. Evaluate the *MMD* loss
3. Update the parameters $\theta_i \to \theta_{i+1}$

M images (epoch 20)



$$\mathcal{L}(\theta_i) = \text{MMD}^2(\text{True Data}, \text{Gen. Data}(\theta_i))$$

$$\theta_i \to \theta_{i+1}$$

Detector Simulation

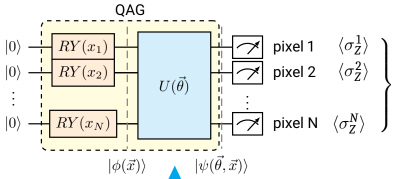
Quantum Angle Generator

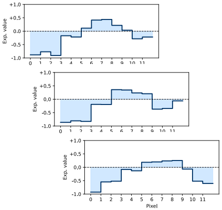
Choose a random initial set of parameters $\vec{\theta}_0$

For each epoch i

1. Generate M images
2. Evaluate the *MMD* loss
3. Update the parameters $\theta_i \to \theta_{i+1}$

M images (epoch 40)





$$\mathcal{L}(\theta_i) = \text{MMD}^2(\text{True Data}, \text{Gen. Data}(\theta_i))$$

$\theta_i \to \theta_{i+1}$

Detector Simulation

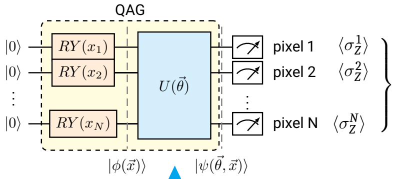
Quantum Angle Generator

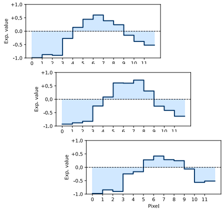
Choose a random initial set of parameters $\vec{\theta}_0$

For each epoch i

1. Generate M images
2. Evaluate the *MMD* loss
3. Update the parameters $\theta_i \to \theta_{i+1}$

M images (epoch 60)





$$\mathcal{L}(\theta_i) = \text{MMD}^2(\text{True Data}, \text{Gen. Data}(\theta_i))$$

$$\theta_i \to \theta_{i+1}$$

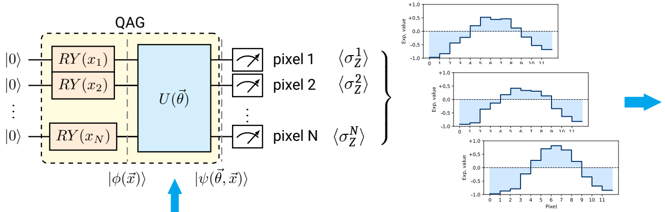
Quantum Angle Generator

Choose a random initial set of parameters $\vec{\theta}_0$

For each epoch i

1. Generate M images
2. Evaluate the *MMD* loss
3. Update the parameters $\theta_i \rightarrow \theta_{i+1}$

M images (epoch 80)

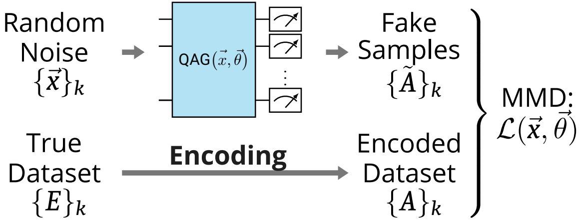


$$\mathcal{L}(\theta_i) = \mathrm{MMD}^2(\text{True Data}, \text{Gen. Data}(\theta_i))$$

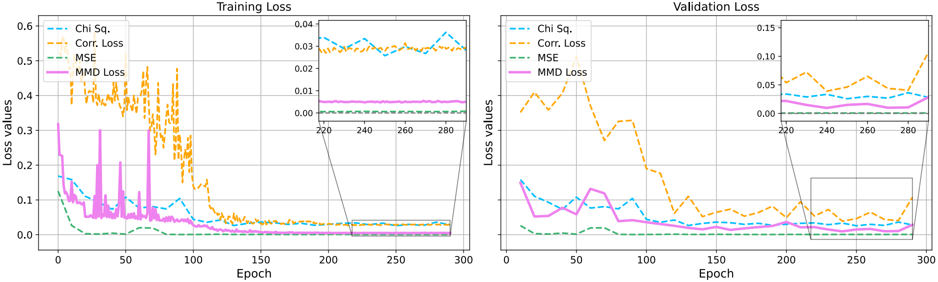
$$\theta_i \rightarrow \theta_{i+1}$$

Detector Simulation

Training



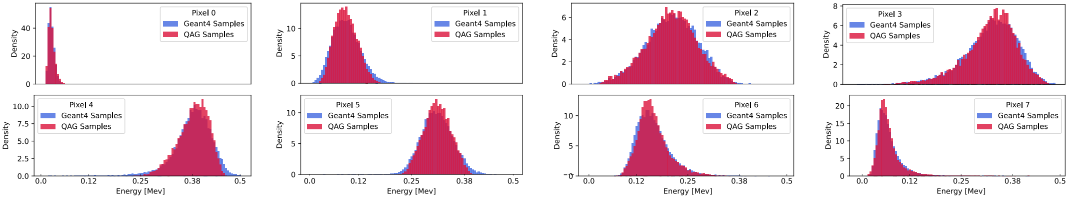
✓ MMD sufficient to learn correlations between pixels



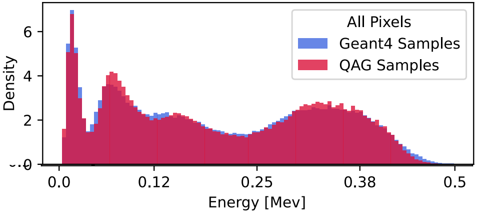
Detector Simulation

Quantum Angle Generator

Pixel-wise Energy distributions



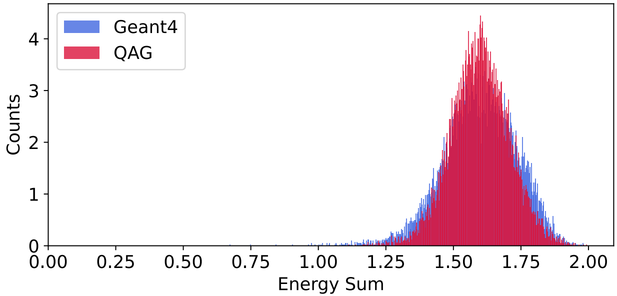
Combined Energy distribution



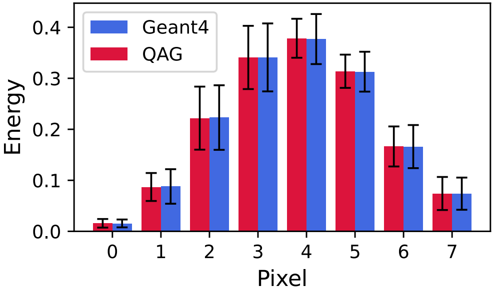
Detector Simulation

Quantum Angle Generator

Energy sum



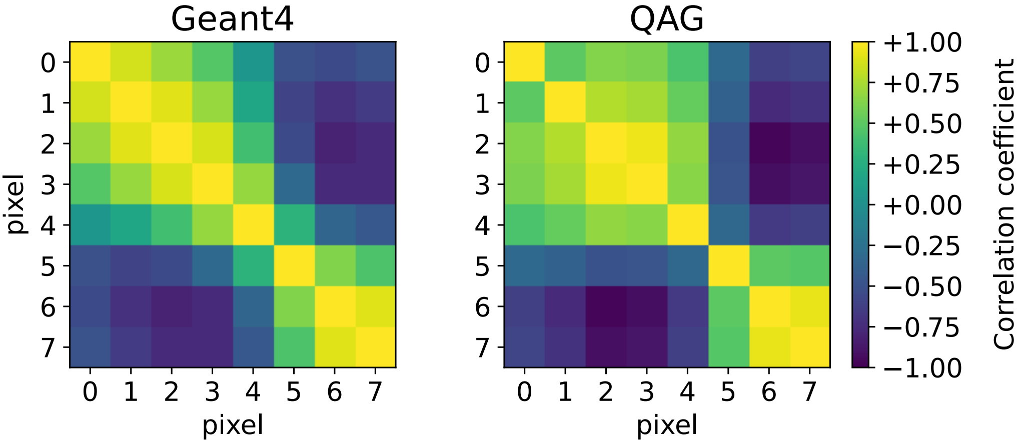
Energy average



Detector Simulation

Quantum Angle Generator

Correlations



Detector Simulation

Quantum Angle Generator

Done:

- Implementation
  - Implementation in both *Pennylane* and *Qiskit*
  - GPU support
- In depth study of Hyperparameters:
  - Number of qubits
  - Ansatz (State Preparation + Trainable)
  - Loss function
  - Data preprocessing
  - Optimizer

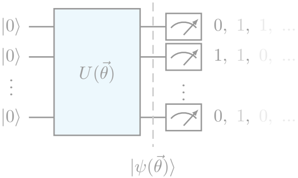
In progress:

- Study under different levels of simualted noise
- Deployment on IBM machines (superconductng)
- Deployment on IQM machines (superconducting)
- Deployment on eleQtron machines (ion-based)
- Comparison with classical counterpart

Detector Simulation

What's next

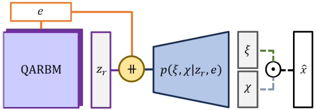
Discrete Architectures



Output: Single measurement of the wavefunction

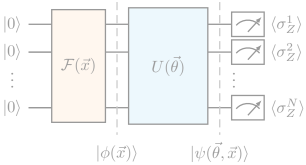
- ✓ Fast generations
- ✓ Always hard to simulate classically
- ✗ Sensitive to noise
- ✗ *Black and white* images

Hybrid Architectures



[5] Toledo-Marin et al.

Continuous Architectures

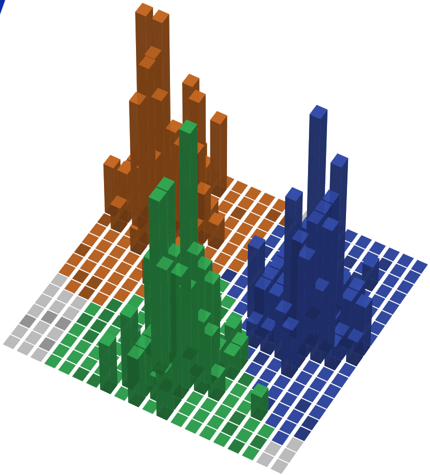


Output: Multiple measurements of the wavefunction

- ✗ Slow generations
- ✗ Generally classically simulable[1]
- ✓ Robust to noise
- ✓ *Grayscale* images

Jet Clustering

How to cluster quantumly



Jet Clustering

Overview

Classical algorithm (anti-k_T)

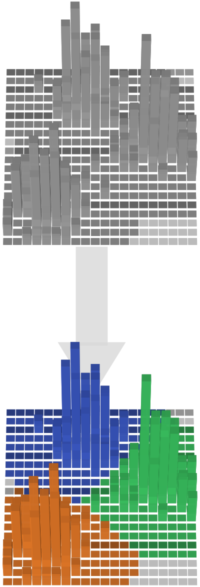
combines particles in order of decreasing transverse momentum, computing the pairwise distance measures between all particles ($\mathcal{O}(N^2)$)

Quantum algorithm[3]

Quantum subroutines

1. Computing distances between all particles
2. Seach for maxima in the unsorted list of resulting distances

(Theoretical exponential speed-up)



Bibliography I

[1] Angrisani et al.: ***Classically estimating observables of noiseless quantum circuits***, arXiv preprint (2024)

[2] Caro et al.: ***Generalization in quantum machine learning from few training data***, Nature Communications (2022)

[3] De Lejarza et al.: ***Quantum clustering and jet reconstruction at the LHC***, PhysRevD (2022)

[4] Riofrio et al.: ***A Characterization of Quantum Generative Models***, ACM Transactions on Quantum Computing (2024)

[5] Toledo-Marin et al.: ***Conditioned quantum-assisted deep generative surrogate for particle-calorimeter interactions***, arXiv preprint (2024)

Questions?

Contact

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.