

QUANTUM MACHINE LEARNING FOR HEP DETECTOR SIMULATIONS

F. Rehm^{1,2,a}, S. Vallecorsa¹, K. Borras^{2,3}, D. Krücker³

¹ *CERN, Esplanade des Particules 1, Geneva, Switzerland*

² *RWTH Aachen University, Templergraben 55, Aachen, Germany*

³ *DESY, Notkestraße 85, Hamburg, Germany*

E-mail: ^a florian.matthias.rehm@cern.ch

Quantum Machine Learning (qML) is one of the most promising and very intuitive applications on near-term quantum devices which possess the potential to combat computing resource challenges faster than traditional computers. Classical Machine Learning (ML) is taking up a significant role in particle physics to speed up detector simulations. Generative Adversarial Networks (GANs) have proven to achieve a similar level of accuracy compared to Monte Carlo-based simulations while decreasing the computation time by orders of magnitude. In this research we are moving on and apply quantum computing to GAN-based detector simulations.

Given the limitations of current quantum hardware in terms of number of qubits, connectivity, and noise, we perform initial tests with a simplified GAN model running on quantum simulators. The model is a classical-quantum hybrid ansatz. It consists of a quantum generator, defined as a parameterised circuit based on single and two qubit gates, combined with a classical discriminator. Our initial qGAN prototype focuses on a one-dimensional toy-distribution, representing the energy deposited in a detector by a single particle. It employs three qubits and achieves high physics accuracy thanks to hyper-parameter optimisation. Furthermore, we study the influence of real hardware noise for the qML GAN training. A second qGAN is developed to simulate 2D images with a 64-pixel resolution, representing the energy deposition patterns in the detector. Different quantum ansatzes are studied. We obtained the best results using a tree-tensor-network architecture with six qubits. Additionally, we discuss challenges and potential benefits of quantum computing as well as our plans for future developments.

Keywords: Quantum Machine Learning, Hybrid Classical-Quantum GAN, Noisy Quantum Simulations

Florian Rehm, Sofia Vallecorsa, Kerstin Borras, Dirk Krücker

Copyright © 2021 for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Calorimeter Training Data

Calorimeters represent one key component of High Energy Physics experiments by measuring the particles energy [1]. Particles enter the calorimeter and deposit their energies by producing secondary particles which, in turn, create other particles by the same mechanisms. This chain process leads to the generation of particle showers. One example shower image is shown in Figure 1 (left) for a primary particle with 500 GeV energy. The training data is generated using the Geant4 toolkit [2]. The training data, including the classical ML GAN studies for calorimeters, are further explained in Ref. [3].

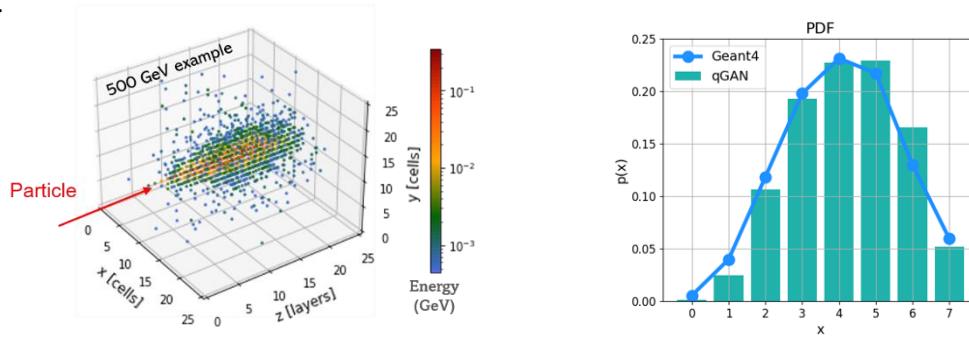


Figure 1. Shows (left) an example 3D shower image of an electromagnetic calorimeter and (right) the generated probability density function (PDF) of the trained 1D qGAN model in green as well as the 1D training data in blue.

2. Hybrid Quantum GAN

Quantum circuits are limited in size and depth, because today's quantum computers are still very noisy and have only a small number of qubits. However, the resilience of ML algorithms against noise as demonstrated with classical algorithms makes qML attractive for noisy intermediate-scale quantum (NISQ) hardware [4]. The quantum simulations are often run with a hybrid quantum-classical approach. In this research we modified and optimised a hybrid quantum-classical qGAN approach, presented in Figure 2, initially developed by IBM [5].

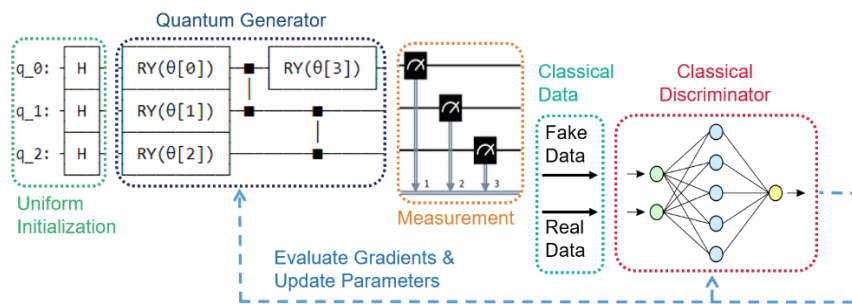


Figure 2. Hybrid quantum-classical GAN model

The model utilizes a parameterised quantum generator and a classical discriminator neural network. The classical data is represented in quantum states using the amplitude encoding approach and the quantum generator output state is measured in classical data format. At each forward generation step one quantum output state is calculated which corresponds to one classical pixel value. The complete shower images are generated by operating the quantum generator for multiple repetitions (so-called shots). Therefore, the generated images are rather probabilistic distributions instead of real images. The quantum generator circuit is illustrated in Figure 3. It consists of three qubits to represent the 8 pixels ($2^3 = 8$) of the simplified shower image as shown in Figure 1 (right). The qubits are initialized in basis state zero $|0\rangle$ followed by Hadamard gates to initiate superposition.

The generator consists of two layers as indicated in Figure 3. Parameterised Y-rotational gates provide the trainable part: each gate possesses its own trainable parameter θ which corresponds to the rotation angle. Entanglement is created by controlled Z-gates in a linear entanglement topology.

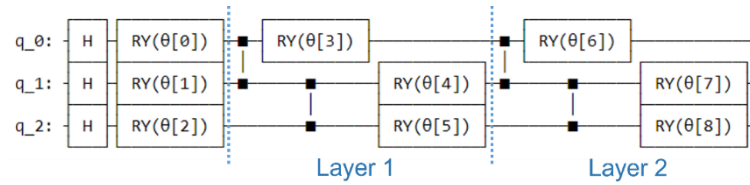


Figure 3. 1D quantum generator circuit

The generated images represent the fake data which are, together with the training data (real data), input into the classical discriminator neural network. The discriminator consists of one input feature, two hidden dense layers with 512 nodes (and the LeakyReLU activation function) and one output layer with dimension 256 (with the Sigmoid activation function). A single output neuron provides the true-fake probability. Following the standard GAN approach a binary cross-entropy loss function is used, the gradients are classically computed, and the parameters of the quantum generator and the classical discriminator updated.

3. Hyperparameter Optimization

Initially, we run the qGAN training process using the QASM quantum simulator [6] without including any noise, on an Intel(R) Xeon(R) Gold 6130 CPU (with 2.10 GHz cores). Convergence was achieved after 4000 training epochs, requiring extremely long simulations (over one day). Multiple hyperparameter searches, using the Optuna [7] tool, managed to speed up the qGAN training by a factor of 10x. In the first place, we observed that higher learning rates increase the training speed but, on the other side, decrease the accuracy. To overcome the decreasing accuracy, we implemented an exponential learning rate decay. This allows in the initial training iterations fast learning with a high learning rate and in the latter one's accurate results with a moderate learning rate. Furthermore, the learning rate decay stabilises the training during later epochs with fewer oscillations. Training classical GAN models, we observed that separate generator and discriminator learning rates improve the training quality. Therefore, we implemented this approach to the hybrid qGAN model and achieved an improved accuracy and faster training. As a last study, we modified the number of discriminator training iterations with respect to generator training iterations within each epoch. We found, that in case the discriminator is trained ten times more often than the generator, the training converges much faster in terms of number of epochs. With all previously mentioned adoptions and with a generator learning rate of 0.008, a discriminator learning rate of 0.001 and an exponential learning rate decay of 0.004 we were able to decrease the training time from 4000 down to only 300 epochs. The probability density function (PDF) of the best training is close to the training data is shown in Figure 1 (right).

4. qGAN Noise Studies

In the following section we study the influence of quantum hardware noise. Initially, we apply exclusively readout noise to the measurements of the generator output. Additionally, the training converged rapidly to a low relative entropy and kept this level stable without oscillations. To perform the study as realistic as possible we apply the noise model measured for the IBMq Belem quantum computer [8]. The readout noise level of the Belem quantum computer at the time we carried out the tests are documented in Table 1. The IBMq Belem quantum computer possesses five qubits, but as we require three qubits for the circuit, we neglect the residual two qubits. At the time we performed the tests the noise level was relatively high, with 3.6%, 4.7% and 9.6% respectively. However, when we look at the PDF plot in Figure 4 (left) we can see that the readout noise does not result in a lower

accuracy. Additionally, the training converged more rapidly to a low relative entropy value and kept this level stable without oscillations.

Table 1. Readout noise levels for the single qubits of the IBMq Belem quantum computer [8].

Qubit Number:	0	1	2
Readout Error:	3.6%	4.7%	9.6%

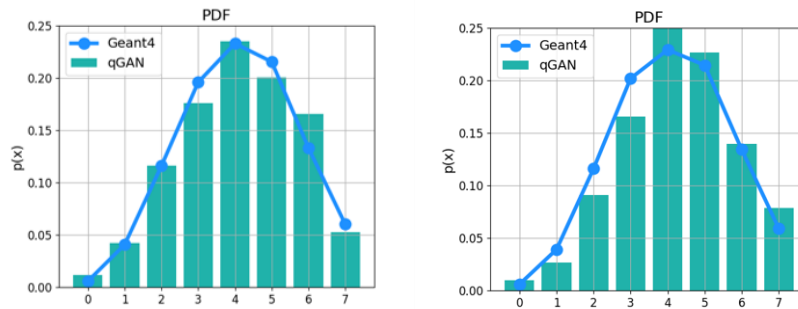


Figure 4. Shows (left) the PDF of the qGAN model trained with only readout noise and (right) the model trained with the full noise model (readout noise + gate level noise)

In the subsequent step we apply the full noise model to the qGAN training. The full noise model comprises gate level noise additionally to the readout noise. Gate level noise is usually determined by the noise of the cx -gate, because the cx -gate is the gate which has typically the highest noise level. At the time we loaded the noise model of the IBMq Belem quantum computer the gate level noise was on average 4.32% for the three considered qubits. The readout noise is summarised in Table 1.

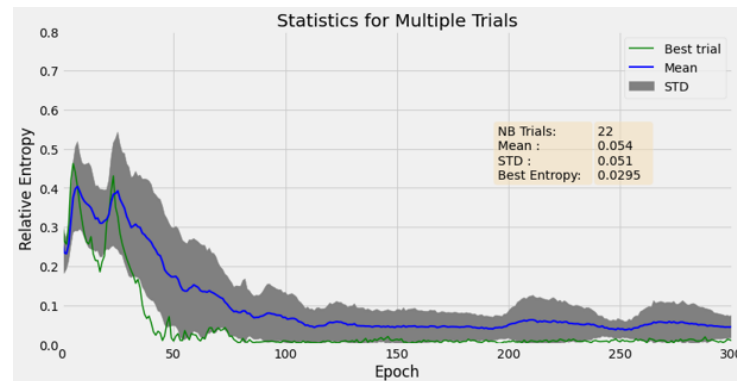


Figure 5. Statistics plot for the accuracy of the full noise qGAN model for multiple trials

The PDF of the trained model with the full noise model is shown on the in Figure 4 (right). One can see that the generator output is farther off from the Geant4 distribution and performs slightly worse. We trained the full noise model multiple times with the same hyperparameters and evaluated the training statistics in Figure 5. The accuracy metric to determine the best trial is the relative entropy which is a measure for the difference between two probability distributions. The training was run for 23 trials with the same hyperparameter set. The relative entropy of the best trial is shown in green, the mean value of all trials in blue and the grey band measures one standard deviation. On average the model converges within the first 100 epochs and the relative entropy remains stable for higher epochs. There are some fluctuations between the various trials (indicated by the broad grey band), however, this effect is expected and similar for the classical ML GAN model due to statistical effects.

The results of the statistical evaluation of the noiseless model and the full noise model are summarized in Table 2 for comparison. One can see that the performance of the model with noise is slightly worse in terms of the mean relative entropy and the relative entropy best trial. However, the

model with noise has a lower standard deviation indicating that the values of the relative entropy of the trained models are closer to each other. This result can be interpreted as an indication that the qGAN models learns the hardware noise behaviour and is moderately resilient against it.

Before running the training on a real quantum device, we plan to perform additional tests using different noise models to better understand the qGAN behaviour. Additionally, a relevant question we examine is what the impact on the performance can be seen if we adapt the hyperparameters of the qGAN model for training with noise influence. For the noise studies presented in this paper, we ran the training with the hyperparameters which performed best in the noiseless case.

Table 2. Shows the accuracy of the statistical evaluation of the noiseless qGAN model and the model with full noise

	Noiseless Model	Full Noise Model
Mean	0.046	0.054
STD	0.064	0.510
Best	0.0077	0.0125

5. 2D qGAN

Because the one-dimensional qGAN model reached a high accuracy, we increased the complexity of the model to tackle a more realistic detector simulation. We increased the model dimension to a total $8 \times 8 = 64$ pixels (8 times as much as for the 1D qGAN model) in order to reproduce a two-dimensional image of the energy pattern in the detector. For simplicity we stacked the pixels of the image into a one-dimensional vector.

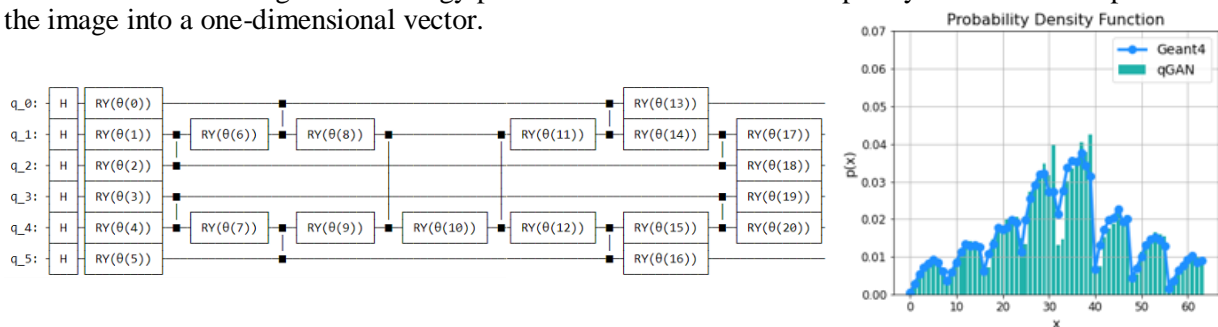


Figure 6. (left) The 2D quantum generator circuit with 6 qubits and (right) the PDF of the best trial

We tested different quantum generator circuit architectures. The most promising results were achieved with a Tree-Tensor-Network (TTN) architecture published in Ref. [9]. In this case, the generator circuit consists of six qubits ($2^6 = 64$) and it is shown in Figure 6 (left). We have also increased the discriminator size, including two additional dense layers with 512 nodes. The generated PDF for the 2D case is shown in Figure 6 (right). One can see that the qGAN output, in green, is close to the training data, in blue, except for a few slightly off pixels. However, the training process turned out to be unstable with convergence in rare cases reached. Additionally, the training lasts for over 6 000 epochs which results in training times of more than five days on the quantum simulator.

6. Conclusion and Future Work

In this paper we implemented a one-dimensional qGAN model to correctly generate single particle energy distributions as measured in a calorimeter. By optimizing the training hyperparameters, we achieved a high accuracy, and were able to accelerate the training process by a factor of $10 \times$. We performed noise studies to understand the influence of quantum hardware noise on the qGAN training process. We measured that a realistic noise model, such as the IBMq Belem's, does not influence the

accuracy if only readout noise is applied. Adding gate-level noise, we experienced a slight decrease of accuracy. Before running the qGAN training and inference on a real quantum device, we plan to perform further noise tests to better understand its influence on the final accuracy.

In addition, we created a more realistic qGAN model which is capable of generating two-dimensional energy distributions with eight times more pixels than the initial one-dimensional model. By using a Tree-Tensor-Network architecture for the quantum generator circuit, we were able to reproduce the 2D shower image correctly. However, further optimization of the training hyperparameters is needed in order to reach faster, more stable, training and the desired level of accuracy.

7. Acknowledgements

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research.

References

- [1] R. M. Brown and D. J. A. Cockerill, "Electromagnetic Calorimetry," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, pp. 47-79, 2012.
- [2] S. Agostinelli, GEANT4--a simulation toolkit, *Nucl. Instrum. Meth. A*, 2003.
- [3] F. Rehm, S. Vallecorsa, K. Borras and D. Krücker, "Validation of Deep Convolutional Generative Adversarial Networks for High Energy Physics Calorimeter Simulations," in *AAAI 2021 - Association for the Advancement of Artificial Intelligence*, 2021.
- [4] Braccia, Paolo, Caruso and Filippo, "How to enhance quantum generative adversarial learning of noisy information," *New Journal of Physics*, May 2021.
- [5] "qGANs for Loading Random Distributions," [Online]. Available: https://qiskit.org/documentation/machine-learning/tutorials/04_qgans_for_loading_random_distributions.html. [Accessed March 2021].
- [6] H. Abrahm, AduOffei and Rochisha, "Qiskit," 2019. [Online].
- [7] T. Akiba, S. Shotaro and Y. Toshihiko, "Optuna: A Next-generation Hyperparameter Optimization Framework," 2019. [Online].
- [8] "IBM Quantum Services," July 2021. [Online]. Available: https://quantum-computing.ibm.com/services?services=systems&system=ibmq_belem.
- [9] E. Grant, M. Benedetti, S. Cao and A. Hallam, "Hierarchical quantum classifiers," *npj Quantum Information*, 2018.