PROCEEDINGS OF SCIENCE

Feasibility study of QNN-based latent space generation for fast calorimeter simulation in ATLAS

Federico Andrea Guillaume Corchia* **and Matteo Franchini**

Department of Physics and Astronomy "Augusto Righi", Alma Mater Studiorum University of Bologna, Viale Carlo Berti Pichat 6/2, Bologna, Italy

INFN Bologna, Viale Carlo Berti Pichat 6/2, Bologna, Italy

E-mail: corchia@bo.infn.it, matteo.franchini@bo.infn.it

The increasing demands on simulation statistics for HL-LHC analyses challenge the scalability of traditional calorimeter simulation within all LHC collaborations. Fast simulation techniques based on machine learning have proven effective, yet further improvement may arise from quantum-inspired models. In this study we investigate the feasibility of integrating Quantum Neural Network (QNN) components into the ATLAS fast calorimeter simulation framework, to evaluate potential gain in generative performance and generalisation.

We focus on a hybrid pipeline where a quantum Generative Adversarial Network (qGAN) is employed, whose generator is composed of a Parametrised Quantum Circuit (PQC) followed by a classical neural network, while the discriminator is kept classical. We report on implementation details, benchmarking against established fast simulation components, and performance metrics. Preliminary results show that qGAN models are technically feasible within ATLAS software and are promising for benefit with respect to the classical ATLAS fast simulation system.

International Symposium on Grids and Clouds (ISGC2026)
15-20, March, 2026
Academia Sinica, Taipei, Taiwan

*Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0) All rights for text and data mining, AI training, and similar technologies for commercial purposes, are reserved. ISSN 1824-8039 . Published by SISSA Medialab.

PoS(ISGC2026)020

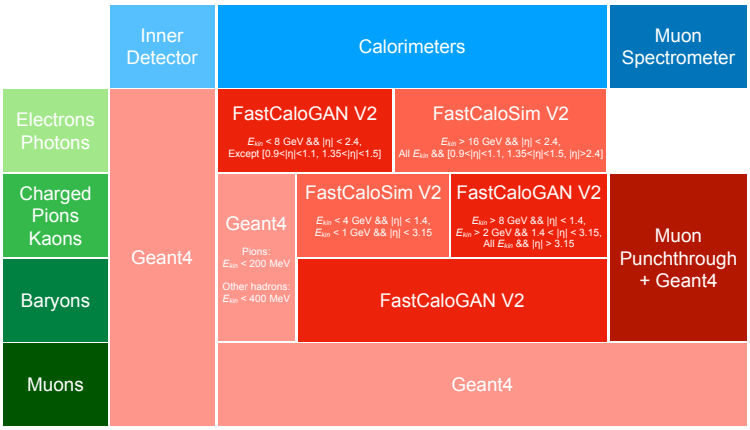

Figure 1: Configuration of the subsystems of AtlFast3, as used for Run 3, depending on detector region (AtlFast3 handles all ATLAS components, not just calorimeters), particle type and particle energy. Geant4 is still used to run simulation in specific cases. Muon Punchthrough (the spray of particles into the ATLAS Muon Spectrometer resulting from late interacting high-energy hadrons) is modelled with a tool based on Deep Neural Networks (DNNs) [1].

1. Introduction

1.1 Fast Simulation and AtlFast3

A major computing challenge at LHC experiments is simulation of Monte Carlo (MC) events in the detector, which takes about 40% of the total load on the computing resources [2] of the ATLAS experiment [3]. A large part (about 80% of CPU consumption in the case of $t\bar{t}$ processes) is taken by the detailed simulation of electromagnetic and hadronic showers in the ATLAS calorimeter systems [4]. In addition, the computing requirements of the ATLAS Collaboration are expected to increase significantly in the coming years with the latest data taking run of the LHC (Run 3) and the next one approaching (Run 4), which is also the first one of the High Luminosity LHC (HL-LHC) [5].

To address these demanding and increasing requirements, significant research and development are required and one of the solutions developed in the Collaboration is the usage of *fast simulation* tools. These programs simulate detector response faster than the standard full process simulation tool Geant4 [6–8], but keeping the loss of accuracy to a minimum.

The ATLAS Collaboration has developed AtlFast3 as its fast simulation tool and put it into production for Run 3 [1, 4]. AtlFast3 replaces the slow propagation and interactions of particles inside the calorimeter volume with the direct generation of energy deposits, by means of an underlying parametrisation. AtlFast3 is composed of two main subsystems, corresponding to two different approaches to fast simulation: FastCaloSimV2, based on longitudinal and lateral parametrisation of showers, and FastCaloGANV2, a machine learning approach based on Generative Adversarial Networks (GANs). In AtlFast3 fast simulation runs through one of the two approaches, as shown in Fig. 1; the choice of the simulator was carefully tuned through extensive validation and is optimised to give the best physics performance.

AtlFast3 is from 3 to 15 times faster than the Geant4 simulation of the ATLAS Run 3 detector [9] and only needs on average 20% of CPU time with respect to Geant4 [1]. For what concerns

PoS(ISGC2026)020

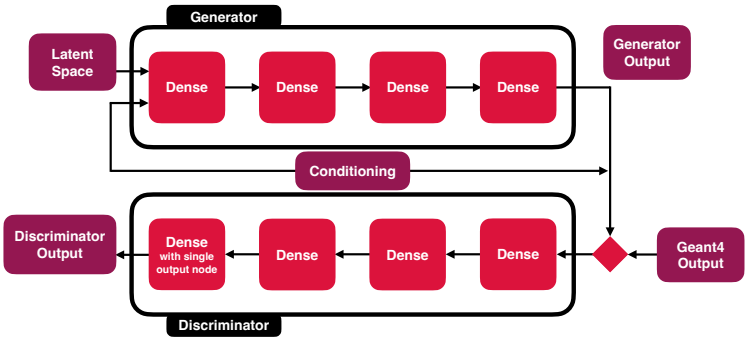

Figure 2: Architecture of FastCaloGANV2.

physics performance, AtlFast3 and Geant4 agree within a few percent for most observables used in physics analyses [1].

1.2 FastCaloGANV2

FastCaloGANV2 is one of the AtlFast3 subsystems, based on GANs [10]. This architecture, first introduced in [11], features the simultaneous training of two neural networks, one called a *generator* and the other one a *discriminator*. The generator aims to generate samples as similar as possible to the showers generated by Geant4, while the discriminator is input samples from both Geant4 and the generator and aims to distinguish actual Geant4 samples from those produced by the generator. During training both neural networks try improving their abilities and, when a Nash equilibrium is reached between the two, the generator is ready to simulate calorimeter showers, running much faster than Geant4 but keeping at the same time good accuracy.

In particular, FastCaloGANV2 employs Wasserstein GANs with Gradient Penalty (WGAN-GP) [12], an evolution of the original GAN architecture that makes training more stable and performant. The FastCaloGANV2 architecture is shown in Fig. 2. FastCaloGANV2 was trained for electrons, photons, charged pions and protons as particles initiating the shower, and with conditioning (the GAN being supplied additional information acting as class labels) on truth momentum; calorimeter hits are grouped into three-dimensional bins (*voxels*). The generator input is a latent space, set to random values, and the generator transforms the latent space into an energy deposition "image" showing the energy deposited into each voxel. Different latent space values create different shower energy deposition events.

2. Quantum FastCaloGAN

2.1 General Idea

Quantum Computing is of great interest nowadays, as it has the potential to have relevant influence in Physics studies and beyond. Examples of Quantum Computing works in High Energy Physics are shown in [13–16]. Question arises whether it can be leveraged to run fast simulation and further improve FastCaloGANV2 and AtlFast3. In the following a Proof of Concept for a quantum version of FastCaloGANV2 (Quantum FastCaloGAN) is presented. With respect to other works on quantum fast calorimeter simulation [14–16], this study features all of the following:

PoS(ISGC2026)020

the model is a quantum GAN, a quantum version is made of a fast simulation tool of the ATLAS Collaboration actually in production and energy deposition images (as illustrated next) are produced of hundreds of voxels, not of a more limited count such as of the order of ten.

The general idea is to place a Parametrised Quantum Circuit (PQC) in the GAN generator after the latent space and before the classical neural network layers, in order to move part of the computational load from the neural network to the quantum addition. This idea is implemented into different models about to be presented, that play with the complexity of the PQC architecture and the one of the neural network to find a combination that provides a benefit of any kind compared to the fully classical case. All presented models have their PQC simulated classically, not run on an actual quantum computer at the moment: this is kept in mind when discussing the possible benefit.

2.2 Hardware, Software and Training Dataset

Quantum FastCaloGAN has its PQC coded in PennyLane [17], while the overall hybrid quantum-classical generator neural network of which the PQC is part is coded in Keras (which supports implementing PennyLane PQCs into its neural networks) with the TensorFlow backend [18]. The code derives from the *CaloChallenge* version of FastCaloGANV2. The CaloChallenge was a challenge where participants were tasked with training their favourite generative model on the provided ATLAS public datasets of shower energy deposition events in the calorimeter systems, with comparison and discussion of the results, that ultimately led to the paper in [19]. This version of the code is used as it is publicly accessible, allowing more flexibility in using computing resources and personnel external to CERN.

The training dataset is also one of those provided with the CaloChallenge, always for the same considerations as for the code. It contains 120,230 shower energy deposition events in the ATLAS calorimeter systems as simulated by Geant4, produced by pions of 15 different values of incident energy in the pseudorapidity range $0.20 \leq |\eta| \leq 0.25$. Each event contains 533 (i.e., the number of voxels n_v into which the calorimeter volume is divided) floats, representing the energy deposition in each voxel. From this dataset, two training datasets are created. Both datasets have a limited number of events; this is appropriate considering that the quantum models are simulated, with the associated processing time and computational cost, and not run on an actual quantum computer at the moment. Training times would be unreasonable otherwise. The two training datasets differ as follows:

- *single-energy* dataset: containing 1000 events produced by an incident pion of energy 65 GeV. Models are tested first with this dataset, intended to be the easy training task compared to the second training dataset;
- *three-energy* dataset: containing 1500 events, produced by an incident pion of energy 4 GeV, 65 GeV or 1 TeV (500 events per incident energy value). This dataset introduces an additional difficulty, that is to be able to choose the energy of the incident pion and generate energy deposition events for each incident energy value. This is indeed what classical FastCaloGANV2 is able to do, even though it can for 15 different incident energy values, while for this Proof of Concept we limit at the moment to three values to have an increase in difficulty, with respect to the single-energy dataset case, that is not excessive. The models performing best with the single-energy dataset are also tested with this more difficult one.

PoS(ISGC2026)020

The batch size is set to 10 for all models. Trainings run on one NVIDIA H100 GPU. Models are given 24 hours to train, instead of being given a defined number of epochs for which to train: this is because the PQC, as it is simulated, places considerable overhead on training (as shown in the continuation), so the models are let train for the highest number of epochs over a reasonable time frame. The specific architectures of the tested models are described in the next sections.

2.3 PQC Template

All Quantum FastCaloGAN generator models discussed here feature an architecture following the general template below. Each component (a neural network layer, a layer of quantum gates in the PQC, a measurement operation...) receives input from the component above and plugs its output into the component below:

1. **Input layer**: this is the latent space, set to random values and of size 10 for all models;
2. **Angle-encoding layer**: the latent space is encoded onto 10 qubits by means of R_x gates;
3. R_y **gates** with trainable angle, one per qubit;
4. R_z **gates** with trainable angle (only in some models), one per qubit;
5. **CNOT gates** between qubit i and qubit $i + 1$ (referred to in the continuation as *CNOT+1*);
6. **Additional CNOT gates** (only in some models, defined immediately in the continuation);
7. **Z measurement** on all qubits;
8. **Classical neural network hidden Dense layer(s)**;
9. **Classical neural network output Dense layer** of size n_v, as the output of the generator is the energy deposition in each voxel.

The block made of components 3, 4 and 5 is repeated twice or three times in some models. The models presented here do not run out all the possible combinations that can be created from this template, as this study is initial and in progress. The "additional CNOT gates" can be CNOT gates between qubit i and qubit $i + 2$ (referred to in the continuation as *CNOT+2* for clarity and conciseness), between qubit i and qubit $i + 3$ (referred to as *CNOT+3*) or between qubit i and qubit $i + 4$ (referred to as *CNOT+4*). For all CNOT gates, if the index of the controlled qubit would exceed 9 ($i \in [0, 9]$), counting restarts from 0 (e.g., a *CNOT+3* gate controlled by qubit $i = 8$, that would control a non-existent qubit $i = 8 + 3 = 11$, controls qubit $i = 8 + 3 - 10 = 1$).

2.4 Performance Metrics

The best training epoch is not necessarily the last one because of training fluctuations, so it is essential to have an *evaluation* step after training. For a given incident energy value, events from the training dataset (i.e., Geant4 events) are used to calculate the voxel total energy distribution. During training, every 1000 epochs a training checkpoint (cp.), that stores the model weights as they are at that point in training, is saved. At the evaluation step, the GAN generator generates for each training checkpoint the same number of events as those taken from the training dataset (for fair

PoS(ISGC2026)020

comparison between simulations as done by GEANT4 and the GAN generator) and computes their voxel total energy distribution. The two distributions are then compared computing a reduced χ^2 (referred to in the continuation as $\tilde{\chi}^2$, being the χ^2 divided by the number of degrees of freedom) between them. The total energy distribution is used as it is simple to define but difficult to simulate. The best training checkpoint is the one with the lowest $\tilde{\chi}^2$ and the final generator has its weights set as in this checkpoint.

The following section discusses the performance of various models created from the template in Section 2.3. Models are evaluated depending not only on the $\tilde{\chi}^2$ value they can achieve, but also on how fast they converge. The quantum models are also compared to a *reference classical model*, being a neural network composed of three hidden Dense layers of size 50, 100 and 200 neurons respectively, followed by the output layer of size n_v neurons; this is to assess whether the quantum models are able to obtain a benefit over it. This reference model is indeed the CaloChallenge version of FASTCALOGANV2, with the only variation being a different setting of a *multiplication hyperparameter*. This hyperparameter multiplies the size of the hidden layers of the generator and in the CaloChallenge FASTCALOGANV2 model it is set to 4, yielding a neural network with three hidden layers of size 200, 400 and 800 neurons respectively. As the CaloChallenge version handles 15 different incident energy values, while the quantum models just one or three depending on the training dataset, the multiplication hyperparameter is set to one here as the simulation task is simpler. The very good result achieved by this reference model, shown in the continuation, confirms the validity of this modification of the hyperparameter. The discriminator for all quantum models is kept the same as in the CaloChallenge version: a classical neural network takes as input an energy deposition image of size n_v and is composed of three hidden Dense layers of size 800, 400 and 200 neurons respectively. The final output layer is of size one neuron as the output is just one value, telling whether the energy deposition image is an actual GEANT4 image or one produced by the GAN generator.

3. Models and Results

3.1 Single-Energy Models

The first investigated models have the classical neural network in a very simple form, so the PQC plugs into a neural network with just one hidden layer of size 100 neurons (i.e., removing from the CaloChallenge model the preceding layer of size 50 neurons and the following one of size 200 neurons). Results refer to the starting single-energy dataset. Table 1 and those shown in the continuation describe the PQC circuit and the neural network configuration, and are to be read with reference to the circuit template shown in Section 2.3. For example, as shown in Fig. 3, model #1.4 has a PQC composed of R_y gates, one per qubit, followed by R_z gates (always one per qubit), by *CNOT+1* gates and then by *CNOT+2* gates; then it plugs into the neural network with the single hidden layer of size 100 neurons and the output layer of size n_v. The angle-enconding layer and the Z measurement layer are always present implicitly. Model #1.1 does not have the R_z gates instead and after the R_y gates there are the *CNOT+1* gates only. Also tested is the classical neural network alone (*NN Only 1* in Table 1) to better assess the impact of the addition of the PQC. The following metrics are also defined:

PoS(ISGC2026)020

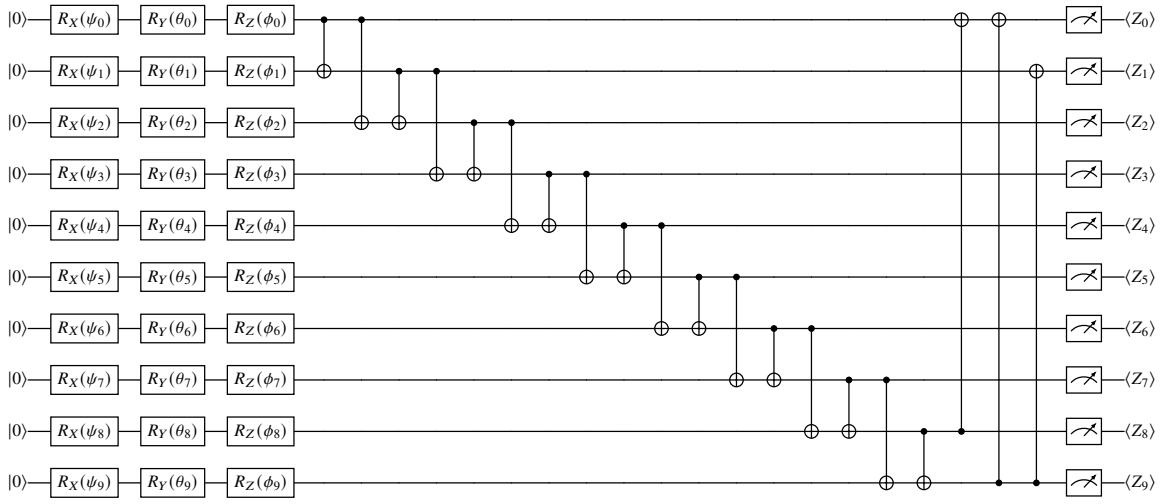

Figure 3: Circuit diagram of the PQC of model #1.4.

- $\langle\tilde{\chi}^2_q\rangle$: the mean $\tilde{\chi}^2$ of the given quantum model in a contour around its best checkpoint (the one yielding the lowest $\tilde{\chi}^2$) of ±30 checkpoints. This average is to exclude the effect of particularly positive fluctuations during training;

- $\langle\tilde{\chi}^2_c\rangle$: the mean $\tilde{\chi}^2$ of the classical neural network alone in a contour of ±30 checkpoints around the best checkpoint **of the given quantum model**. $\langle\tilde{\chi}^2_c\rangle$ is computed in the same range as $\langle\tilde{\chi}^2_q\rangle$ to see what the training performance of the sole neural network is at the point in training where the quantum model is yielding its best result. If $\langle\tilde{\chi}^2_q\rangle$ is lower than $\langle\tilde{\chi}^2_c\rangle$, that means the quantum model is converging faster than the classical neural network alone;

- $\langle\tilde{\chi}^2_{cc}\rangle$: the mean $\tilde{\chi}^2$ of the CaloChallenge reference model in a contour of ±30 checkpoints around the best checkpoint **of the given quantum model**. The same considerations of $\langle\tilde{\chi}^2_c\rangle$ apply. If $\langle\tilde{\chi}^2_q\rangle$ is lower than $\langle\tilde{\chi}^2_{cc}\rangle$, that means the quantum model is converging faster than the CaloChallenge reference model.

A first observation from Table 1 is the fact that the PQC (which is, as said, simulated on classical hardware) adds considerable overhead to training times, as the neural network-only model trains for many more epochs than the quantum ones. For these first quantum models, performance is poor, as the $\tilde{\chi}^2$ is much higher than the neural network-only model and $\langle\tilde{\chi}^2_q\rangle$ is also higher than $\langle\tilde{\chi}^2_c\rangle$. The quantum addition is actually hindering training at the moment. When adding *CNOT+2* gates, the $\tilde{\chi}^2$ of the quantum models starts decreasing, with an additional decrease when also adding the R_z gates, even though performance is still limited (see Fig. 4). However, it is also clear that the classical neural network must be made more powerful and take a more appropriate load of computation. No comparison is proposed here with the CaloChallenge model, as the poor performance of the quantum models with respect to the neural network-only model (simpler than the CaloChallenge model) makes it premature.

In models #2.1, #2.2 and #2.3, presented in Tab. 2, the hidden layer of size 100 neurons of the neural network is replaced by two hidden layers of size respectively 50 and 200 neurons. In this

PoS(ISGC2026)020

Table 1: Performance of various quantum models, made of a PQC configured in different ways plugging into a neural network with one hidden layer of size 100 neurons and one output layer of size n_v, on the single-energy dataset. The quantum models are compared to a classical one only made of the described neural network. The PQC configurations are with reference to the template in Section 2.3.

Model	PQC Configuration	NN Layers	$\tilde{\chi}^2$	$\tilde{\chi}^2$ cp.	Max cp.	$\langle\tilde{\chi}^2_q\rangle$	$\langle\tilde{\chi}^2_c\rangle$
NN Only 1	No PQC	100, n_v	16.8	418	419	-	22.2
#1.1	R_y, *CNOT+1*	100, n_v	45.9	270	280	52.1	30.0
#1.2	R_y, *CNOT+1*, *CNOT+2*	100, n_v	34.0	207	231	42.6	32.2
#1.3	R_y, R_z, *CNOT+1*	100, n_v	54.1	262	271	58.9	30.2
#1.4	R_y, R_z, *CNOT+1*, *CNOT+2*	100, n_v	25.6	229	275	36.4	31.0

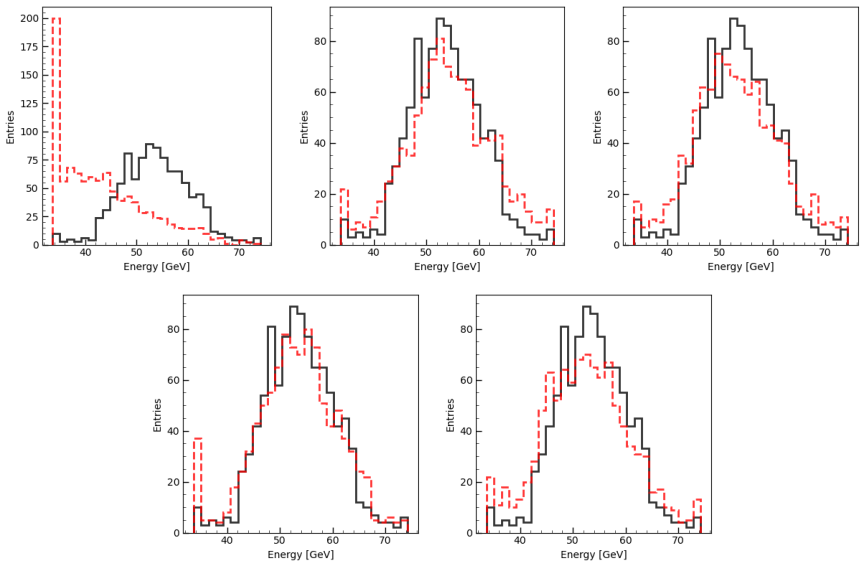

Figure 4: Voxel total energy distribution for the single-energy dataset as simulated by GEANT4 (black solid line) and GAN models #1.4 (top left plot), #2.2 (top central), #2.3 (top right), "2x" (bottom left) and "3x" (bottom right) at their checkpoint with the lowest $\tilde{\chi}^2$ (red dashed line).

way, there is a fivefold increase in size from the PQC output of size 10 to the neural network layer of size 50 neurons and a comparable one (fourfold) from the layer of size 50 neurons to the one of size 200 neurons. This is also equivalent to removing the central hidden layer from the CaloChallenge model keeping the other two hidden ones. It is observed that quantum models #2.2 and #2.3 (see Fig. 4) yield best $\tilde{\chi}^2$ comparable to the neural network-only model and their $\langle\tilde{\chi}^2_q\rangle$ are lower than their corresponding $\langle\tilde{\chi}^2_c\rangle$, confirming that their best $\tilde{\chi}^2$ are no fluctuation, but these models actually converge faster than the neural network-only model. The PQC is not hindering the training process anymore. However, their $\langle\tilde{\chi}^2_q\rangle$ are still both higher than their $\langle\tilde{\chi}^2_{cc}\rangle$, showing that the PQC must

PoS(ISGC2026)020

Table 2: Performance of various quantum models, made of a PQC configured in different ways (repeated multiple times for models "2x" and "3x") plugging into a neural network with two hidden layers of size 50 and 200 neurons respectively and one output layer of size n_v, on the single-energy dataset. The quantum models are compared to a classical one only made of the described neural network and with the CaloChallenge (C.C.) model. The PQC configurations are with reference to the template in Section 2.3. A PQC with configuration [R_y, R_z, *CNOT+1*] (not repeated multiple times) has not been tested as its performance was particularly poor in the case shown in Tab. 1.

Model	PQC Configuration	NN Layers	$\tilde{\chi}^2$	$\tilde{\chi}^2$ cp.	Max cp.	$\langle\tilde{\chi}^2_q\rangle$	$\langle\tilde{\chi}^2_c\rangle$	$\langle\tilde{\chi}^2_{cc}\rangle$
NN Only 2	No PQC	50, 200, n_v	2.13	401	561	-	5.03	3.14
#2.1	R_y, R_z, *CNOT+1*, *CNOT+2*	50, 200, n_v	2.71	222	266	10.5	6.51	2.48
#2.2	R_y, R_z, *CNOT+1*, *CNOT+2*, *CNOT+3*	50, 200, n_v	2.22	155	165	5.39	8.21	3.07
#2.3	R_y, R_z, *CNOT+1*, *CNOT+2*, *CNOT+3*, *CNOT+4*	50, 200, n_v	2.12	126	163	5.91	9.48	3.26
2x	[R_y, R_z, *CNOT+1*] repeated 2x	50, 200, n_v	1.76	124	125	3.10	9.72	3.27
3x	[R_y, R_z, *CNOT+1*] repeated 3x	50, 200, n_v	2.40	66	104	7.92	21.1	5.66
C.C.	No PQC	50, 100, 200, n_v	1.25	196	556	-	6.98	2.97

still be made more powerful to achieve a benefit, which is simplifying the CaloChallenge neural network by offloading part of its computation load to the quantum part. However, this is already a promising hint.

Models "2x" and "3x" presented in Tab. 2 and at the bottom of Fig. 4 have their PQC, composed of R_y, R_z and *CNOT+1* gates, repeated multiple times. Their $\langle\tilde{\chi}^2_q\rangle$ are lower than their $\langle\tilde{\chi}^2_c\rangle$ and for model "2x" (whose PQC circuit is shown in Fig. 5) $\langle\tilde{\chi}^2_q\rangle$ is comparable to, slightly better than $\langle\tilde{\chi}^2_{cc}\rangle$. This means that this model has performance equivalent, slightly better, to the CaloChallenge classical reference and that part of the computation originally handled by the classical neural network can effectively be offloaded to the quantum PQC.

3.2 Conditioned Quantum FastCaloGAN

Up to now, Quantum FastCaloGAN has only been tested on events with incident energy of 65 GeV. However, it should be able to generate energy deposition events starting from particles of energy of one's choice, as classical FastCaloGANV2 does. Therefore, models "2x" and "3x" presented in the previous section, which were shown to yield the best results for the single-energy dataset, are tested here with the three-energy dataset. The only modification is the implementation of the energy conditioning itself (missing for the single-energy dataset case as useless), that is done

PoS(ISGC2026)020

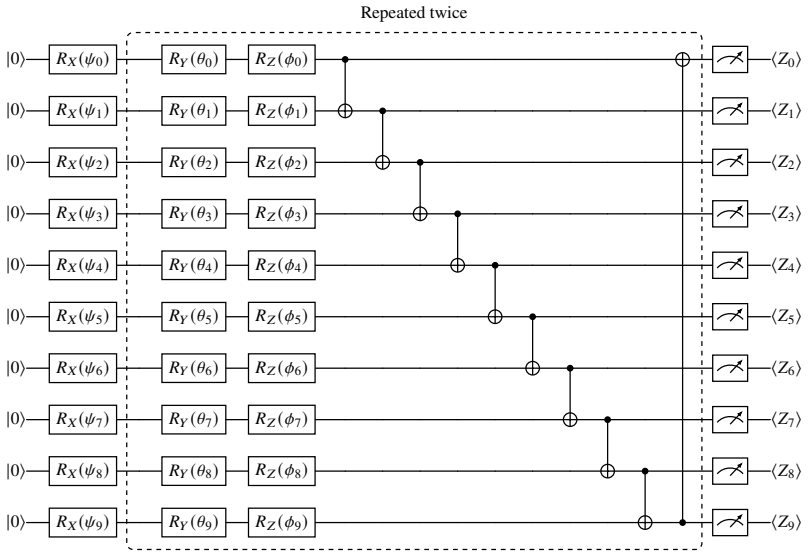

Figure 5: Circuit diagram of the PQC of model "2x" for the single-energy dataset case.

Table 3: Performance of various quantum models, made of a PQC repeated different times plugging into a neural network with two hidden layers of size 50 and 200 neurons respectively and one output layer of size n_v, on the three-energy dataset. The quantum models are compared with the CaloChallenge (C.C.) one. Performance is evaluated both globally (i.e., for all incident energy values) and for each incident energy value. The PQC configurations are with reference to the template in Section 2.3.

Model	PQC Configuration	NN Layers	$\tilde{\chi}^2$	$\tilde{\chi}^2$ cp.	Max cp.	$\langle\tilde{\chi}^2_q\rangle$	$\langle\tilde{\chi}^2_{cc}\rangle$	$\tilde{\chi}^2$ for energy in GeV		
								4	65	10^3
2x	[R_y, R_z, *CNOT+1*] repeated 2x	50, 200, n_v	10.6	78	85	15.8	5.17	6.97	14.3	10.0
3x	[R_y, R_z, *CNOT+1*] repeated 3x	50, 200, n_v	15.8	26	91	21.7	17.2	6.34	20.4	19.7
C.C.	No PQC	50, 100, 200, n_v	1.32	517	564	-	2.30	1.87	1.03	1.10

by adding an additional qubit in the PQC, which is now of size eleven qubits (ten angle-encoding the latent space and the eleventh one angle-encoding, always with a R_x gate, the incident energy).

As shown in Tab. 3 and in Fig. 6, both models have rather high global $\tilde{\chi}^2$. When computing it for each incident energy value, the agreement between GEANT4 and the GAN varies, with better results observed for 4 GeV and worse results for 65 GeV. This shows that the models are starting learning how to handle multiple incident energy values with a single hybrid network, but there is anyway the need of a more powerful model and/or more training epochs (as these quantum models only managed to train for a limited number of them).

PoS(ISGC2026)020

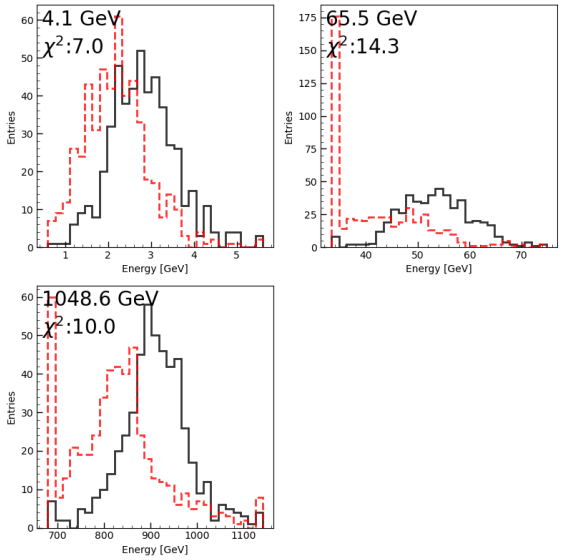

Figure 6: Voxel total energy distribution for the three-energy dataset as simulated by GEANT4 (black solid line in all plots) and quantum GAN model "2x" at the checkpoint with the lowest global $\tilde{\chi}^2$ (red dashed line), for the case of incident energy 4 GeV (top left plot), 65 GeV (top right plot) and 1 TeV (bottom left plot).

4. Conclusions and Outlook

This work presented QUANTUM FASTCALOGAN, a Proof of Concept for a quantum version of FASTCALOGANV2, a fast simulation tool developed by the ATLAS Collaboration at the LHC. The aim of this study was to investigate whether parts of the classical neural network composing the FASTCALOGANV2 GAN generator could be replaced by a PQC obtaining a benefit. It was observed that the repetition of a PQC composed of R_y gates, R_z gates and *CNOT* gates with qubit i acting on qubit $i+1$, combined with a Deep Neural Network made of two hidden layers of size 50 and 200 neurons respectively, hints for a benefit over the classical reference. This hint is currently only observed for the case where the generator simulates shower energy deposition events produced by pions of a single incident energy (65 GeV), not yet for the three-energy dataset case.

This study is initial and in progress, as there are still many more combinations and ideas to test. Ideas to investigate include more connections inside the PQC, tests on a hybrid quantum-classical discriminator, more complex Quantum Neural Networks also including architectures alternative to GANs (e.g. Quantum Invertible Neural Networks, as their classical counterpart was shown to yield very good results in the CaloChallenge study [19], and Quantum Reservoir models), hyperparameter tuning and tests on more incident energy values and events. Of great interest is also the possibility to run on an actual quantum computer, as all results presented here were produced on simulated hardware.

References

[1] ATLAS Collaboration, *Software and computing for Run 3 of the ATLAS experiment at the LHC*, *Eur. Phys. J. C* **85** (2025) 234.

PoS(ISGC2026)020

[2] ATLAS Collaboration, *ATLAS HL-LHC Computing Conceptual Design Report*, *CERN-LHCC-2020-015* (2020).

[3] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, *JINST* **3** (2008) S08003.

[4] ATLAS Collaboration, *AtlFast3: The Next Generation of Fast Simulation in ATLAS*, *Comput Softw Big Sci* **6** (2022) 7.

[5] ATLAS Collaboration, *ATLAS Software and Computing HL-LHC Roadmap*, *CERN-LHCC-2022-005* (2022).

[6] J. Allison et al., *Recent Developments in Geant4*, *Nucl. Instrum. Meth. A* **835** (2016) 186-225.

[7] J. Allison et al., *Geant4 Developments and Applications*, *IEEE Trans. Nucl. Sci.* **53** (2006) 270-278.

[8] S. Agostinelli et al., *Geant4 - A Simulation Toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250-303.

[9] ATLAS Collaboration, *ATLAS-SIM-2023-005* (2023).

[10] ATLAS Collaboration, *Fast simulation of the ATLAS calorimeter system with Generative Adversarial Networks*, *ATL-SOFT-PUB-2020-006* (2020).

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative Adversarial Nets*, *Advances in Neural Information Processing Systems* **3** (2014) 11.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, *Improved Training of Wasserstein GANs*, arXiv:1704.00028 (2017).

[13] A. Delgado et al., *Quantum Computing for Data Analysis in High-energy Physics*, arXiv:2203.08805 (2022).

[14] F. Rehm, S. Vallecorsa, M. Grossi, K. Borras, D. Krücker, *A Full Quantum Generative Adversarial Network Model for High Energy Physics Simulations*, arXiv:2305.07284 (2024).

[15] S. Hoque, H. Jia, A. Abhishek et al., *CaloQVAE: Simulating high-energy particle-calorimeter interactions using hybrid quantum-classical generative models*, *Eur. Phys. J. C* **84** (2024) 1244.

[16] J. Quetzalcóatl Toledo-Marín, S. Gonzalez, H. Jia et al., *Conditioned quantum-assisted deep generative surrogate for particle-calorimeter interactions*, *NPJ Quantum Inf* **11** (2025) 114.

[17] V. Bergholm et al., *PennyLane: Automatic differentiation of hybrid quantum-classical computations*, arXiv:1811.04968 (2018).

[18] F. Chollet et al., *Keras*, keras.io (2015).

[19] C. Krause et al., *CaloChallenge 2022: A Community Challenge for Fast Calorimeter Simulation*, *Rep. Prog. Phys.* **88** (2025) 116201.

PoS(ISGC2026)020