The Bologna Quantum Group
E. Apicella, E. Ballabene, L. Bellagamba, G. Bruni,
F.A.G. Corchia, M. Franchini, L. Rinaldi, F. Semeria

Feasibility Study of QNN-Based Latent Space Generation for Fast Calorimeter Simulation in ATLAS

Federico A.G. Corchia
University of Bologna
and INFN Bologna

International Symposium on Grids and Clouds
(ISGC) 2026
Taipei, 18 March

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

INFN
BOLOGNA

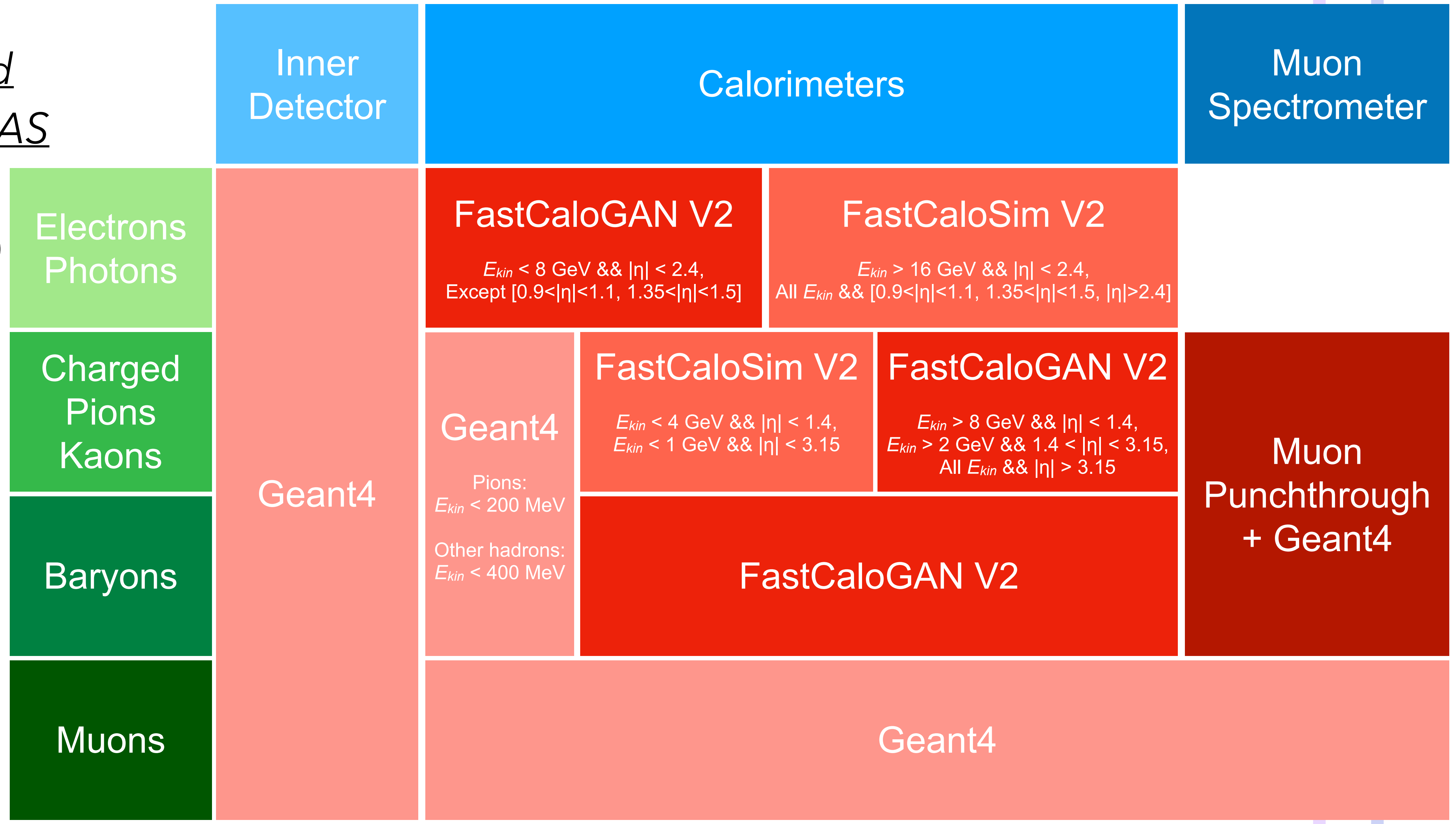
CERN

ATLAS
EXPERIMENT

Introduction

Fast Simulation and AtlFast3

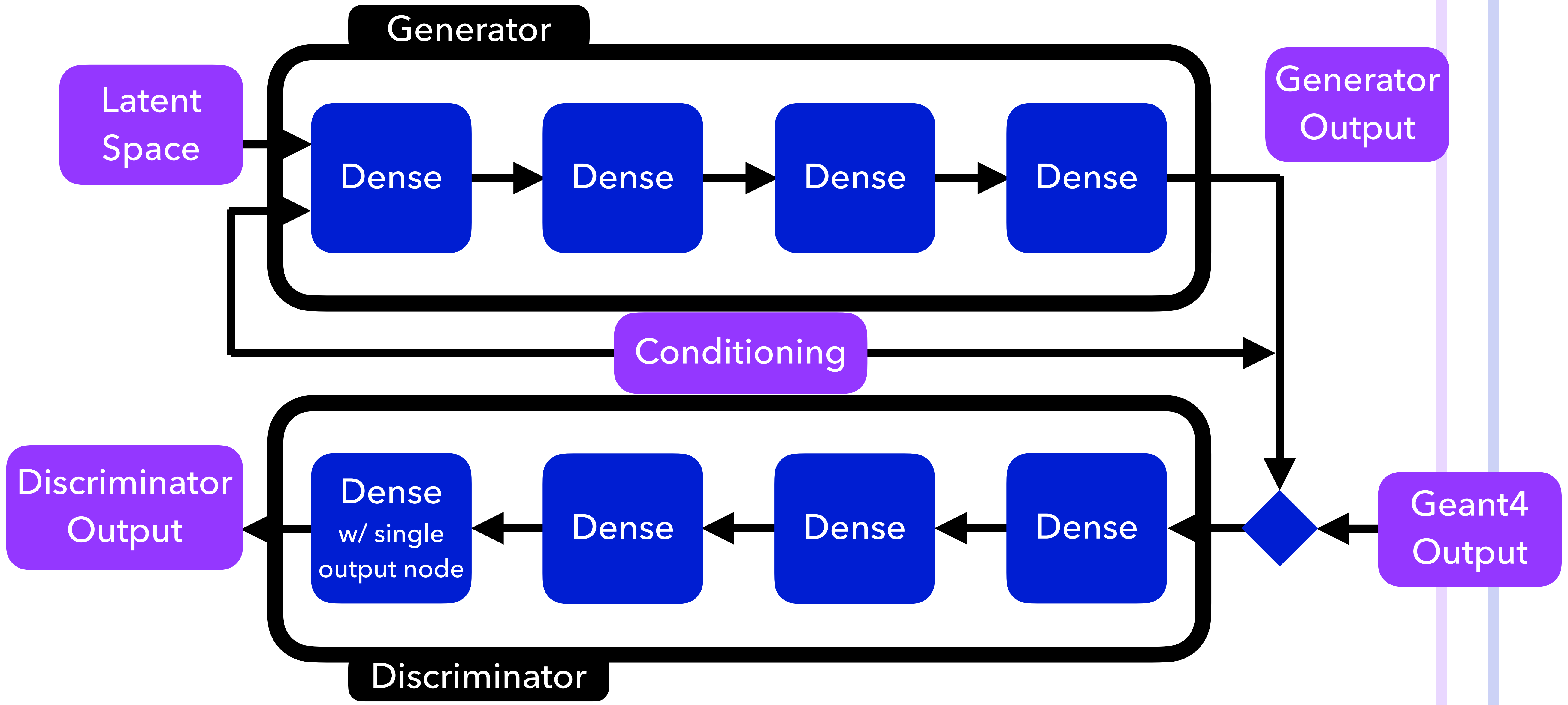
The ATLAS Coll., *Software and computing for Run 3 of the ATLAS experiment at the LHC*, Eur. Phys. J. C **85**, 234 (2025)



- About **25%** of ATLAS CPU-hours **taken just by simulation of calorimeter response**! Needs expected to **increase** during the coming years towards Run 4 → solutions needed!
  - **Fast simulation tools**: simulating calorimeter response faster than the ("traditional") full process simulation tool Geant4 but keeping high accuracy.
- **AtlFast3**: ATLAS fast simulation tool. Employs two fast simulation approaches: FastCaloSim (parametric approach for shower development) and **FastCaloGAN** (based on **Generative Adversarial Networks**, GANs).
- AtlFast3 runs fast simulation through one of the two approaches, depending on which one **returns the best simulation** (i.e. the most accurate with respect to Geant4) for the type of the particle initiating the shower and its energy.
- AtlFast3 is **3** to **15 times faster** than Geant4 and only needs on average **20% of CPU time** with respect to Geant4! For most observables used in Physics analyses, AtlFast3 and Geant4 **agree within a few %.**

Introduction

FastCaloGAN



- Based on **Wasserstein GANs** with gradient penalty
  I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, *Improved Training of Wasserstein GANs*, arXiv:1704.00028 [cs.LG] (2017)

- **Simultaneous training of two neural networks** (i.e. the typical GAN structure):

  - **Generator**: aims to generate sample the most similar to Geant4 datasets;

  - **Discriminator**: aims to distinguish Geant4 samples from the ones produced by the generator.

- When reached an **equilibrium** between the two, the generator is ready to simulate!

- Trained on e^-, γ, p, $\pi^\pm$ as incident particles. Calorimeter hits grouped into ***voxels*** (= 3D bins).

- The **input** of the generator is a **latent space**, set to random values, its **output** an array of size #voxels, containing the **energy deposition in each voxel**. The generator transforms the latent space into an energy deposition "image". Different latent space values create different shower energy deposition events.

Quantum FastCaloGAN

Basic Idea

- **Quantum computing** is of great interest nowadays, as it has the potential to have a relevant influence in Physics studies and beyond.
- **Can it be used to do fast simulation**? Can some of the load on the neural network be given to a **Parametrised Quantum Circuit** (PQC)?
- In the following a **Proof of Concept for a quantum version** of FastCaloGAN is presented.
- General idea: **place a PQC** in the GAN generator after the latent space and before the classical Neural Network (NN) layers. We'll play with the complexity of the PQC architecture and the one of the NN architecture.
- What we'd like to observe is a **benefit** of some kind from adding the PQC (in the limits of the PQC being simulated, **currently** not run on a quantum computer).

Quantum FastCaloGAN

Hardware, Software and Dataset

- Coded in PennyLane + Keras with Tensorflow backend.
- Trained on CERN LXBATCH with 1 NVIDIA H100 GPU (currently no quantum hardware). Models are given 24 h to train.
- Started from the CaloChallenge version of FastCaloGAN.
  - A public version, based on ATLAS Open Data, developed for the *CaloChallenge* paper Claudius Krause, Federico A.G. Corchia *et al.*, *CaloChallenge 2022: a community challenge for fast calorimeter simulation*, Rep. Prog. Phys. **88**, 116201 (2025).
  - More flexibility in using resources (and collaborators!) external to CERN.
- 2 datasets created from the CaloChallenge dataset. Both contain shower energy deposition events in the calorimeter as simulated by Geant4 (the "traditional" simulator). Each event contains 533 (= #voxels) floats = energy deposition in each voxel. Batch size = 10.
  - Single-energy dataset: 1000 events, produced by an incident π of energy 65 GeV. We're starting with this dataset;
  - 3-energy dataset: 1500 ev., produced by an incident π of energy 4 GeV, 65 GeV or 1 TeV (500 ev. per energy value). This dataset introduces an extra difficulty: be able to choose the energy of the incident π and generate energy deposition events for each incident energy value. The models performing best with the single energy dataset will then be tested with this more difficult one.

Quantum FastCaloGAN

Circuit Template

Latent space input, set to random values, of size 10

R_x gate angle-encoding the latent space on 10 qubits

R_y gate with trainable angle

R_z gate with trainable angle (only in some models)

CNOT gate between qubit *i* and qubit *i*+1

Additional CNOT gates (only in some models, see next slide)

Z measurement

NN **hidden Dense** layer(s)

NN **output Dense** of size #voxels (output is the energy deposition in each voxel)

PQC

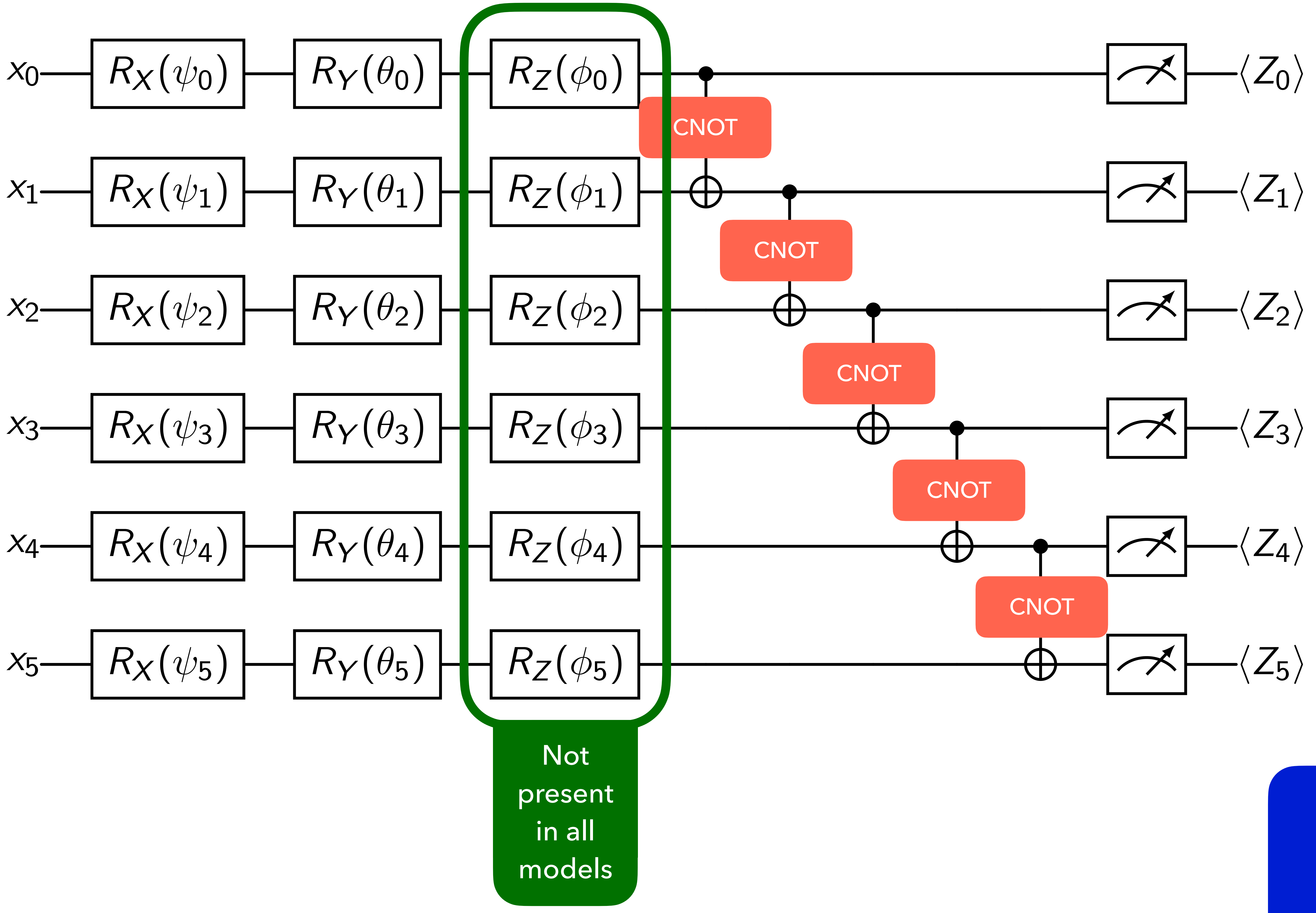
This part is repeated 2x or 3x in some models

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.

Quantum FastCaloGAN

Circuit Template - CNOT Naming Scheme

CNOT +2: qubit *i* acting on qubit *i*+2
CNOT +3: qubit *i* acting on qubit *i*+3
CNOT +4: qubit *i* acting on qubit *i*+4

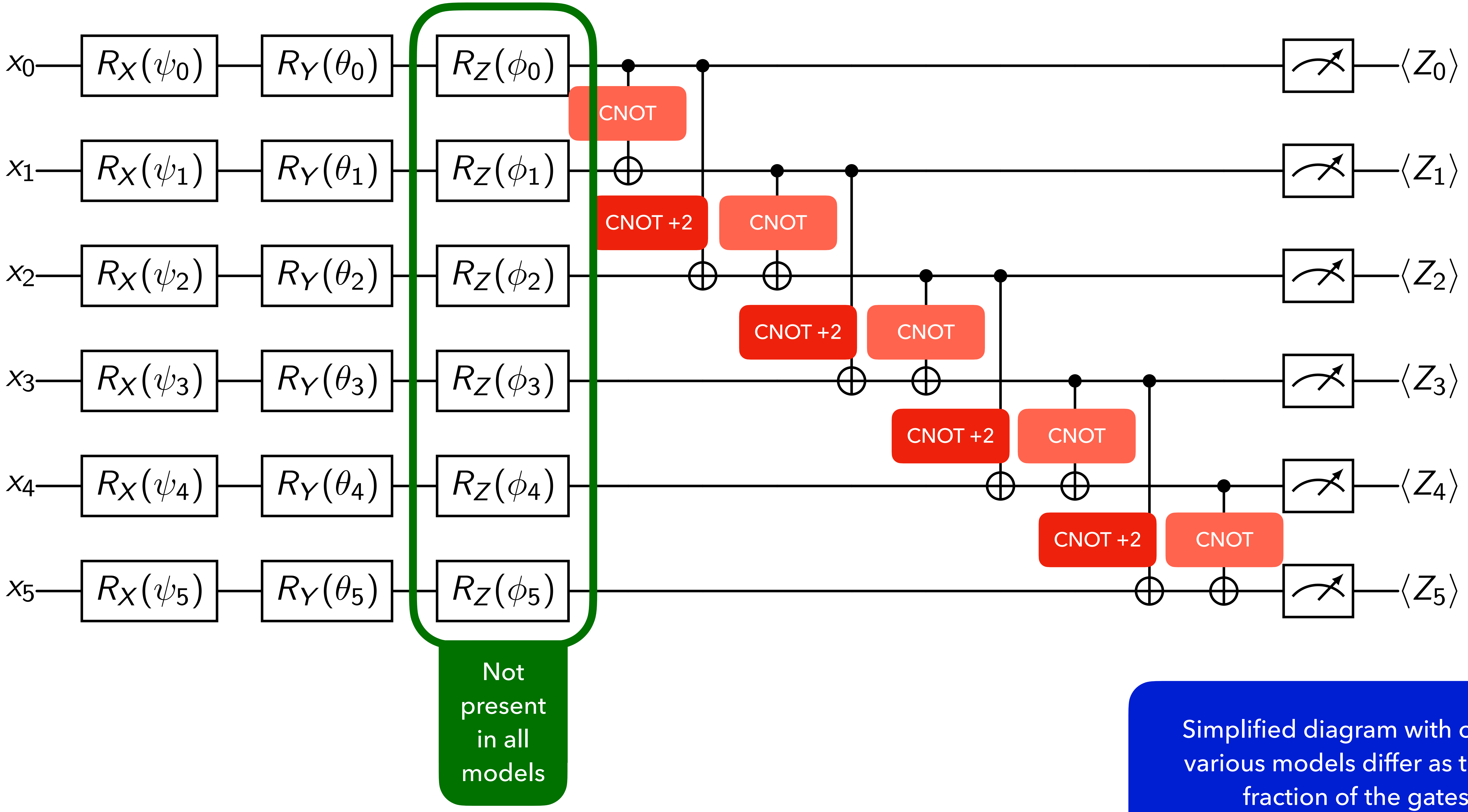


Simplified diagram with only six qubits. The various models differ as they only contain a fraction of the gates shown here.

Quantum FastCaloGAN

Circuit Template - CNOT Naming Scheme

CNOT +2: qubit *i* acting on qubit *i*+2
CNOT +3: qubit *i* acting on qubit *i*+3
CNOT +4: qubit *i* acting on qubit *i*+4

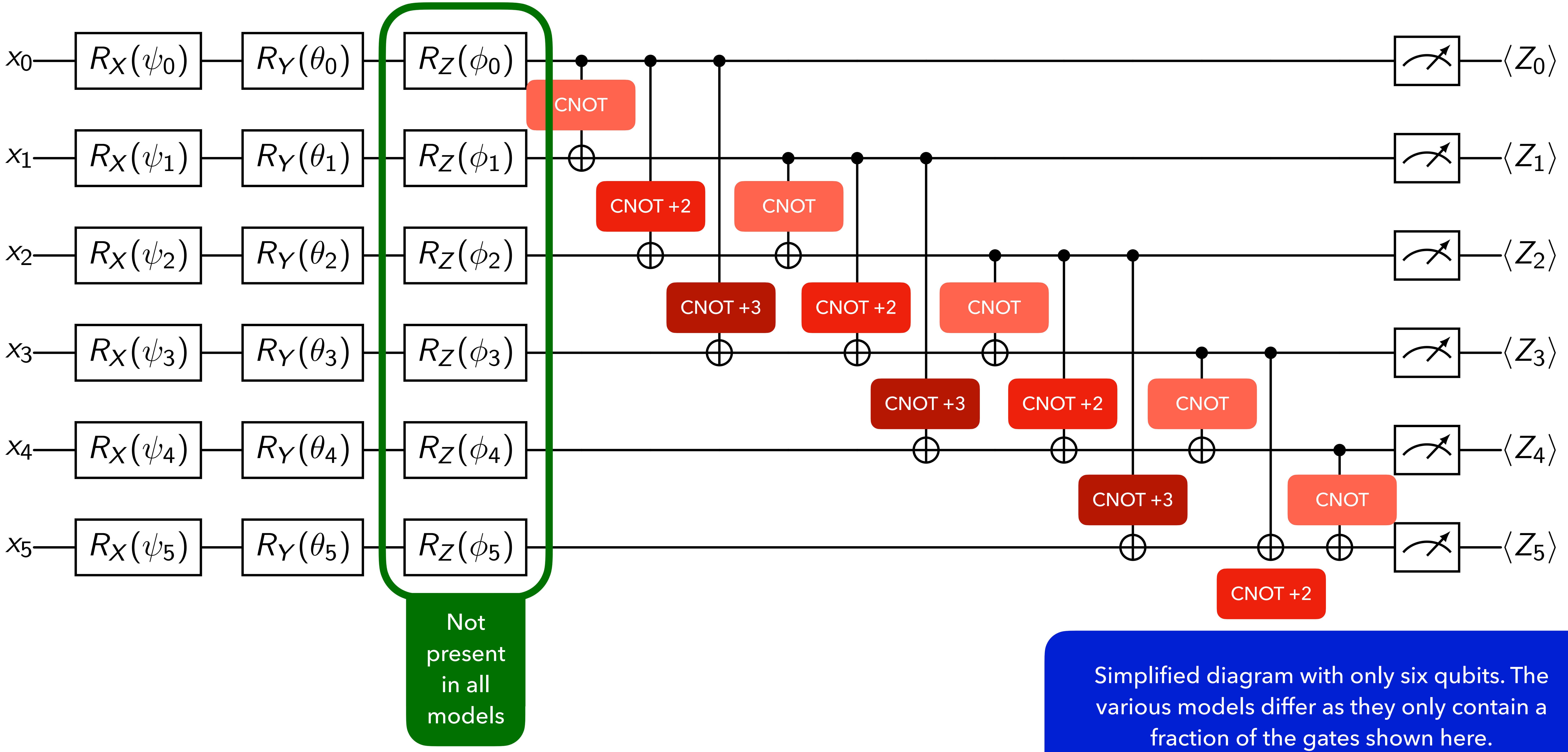


Simplified diagram with only six qubits. The various models differ as they only contain a fraction of the gates shown here.

Quantum FastCaloGAN

Circuit Template - CNOT Naming Scheme

CNOT +2: qubit *i* acting on qubit *i+2*
CNOT +3: qubit *i* acting on qubit *i+3*
CNOT +4: qubit *i* acting on qubit *i+4*

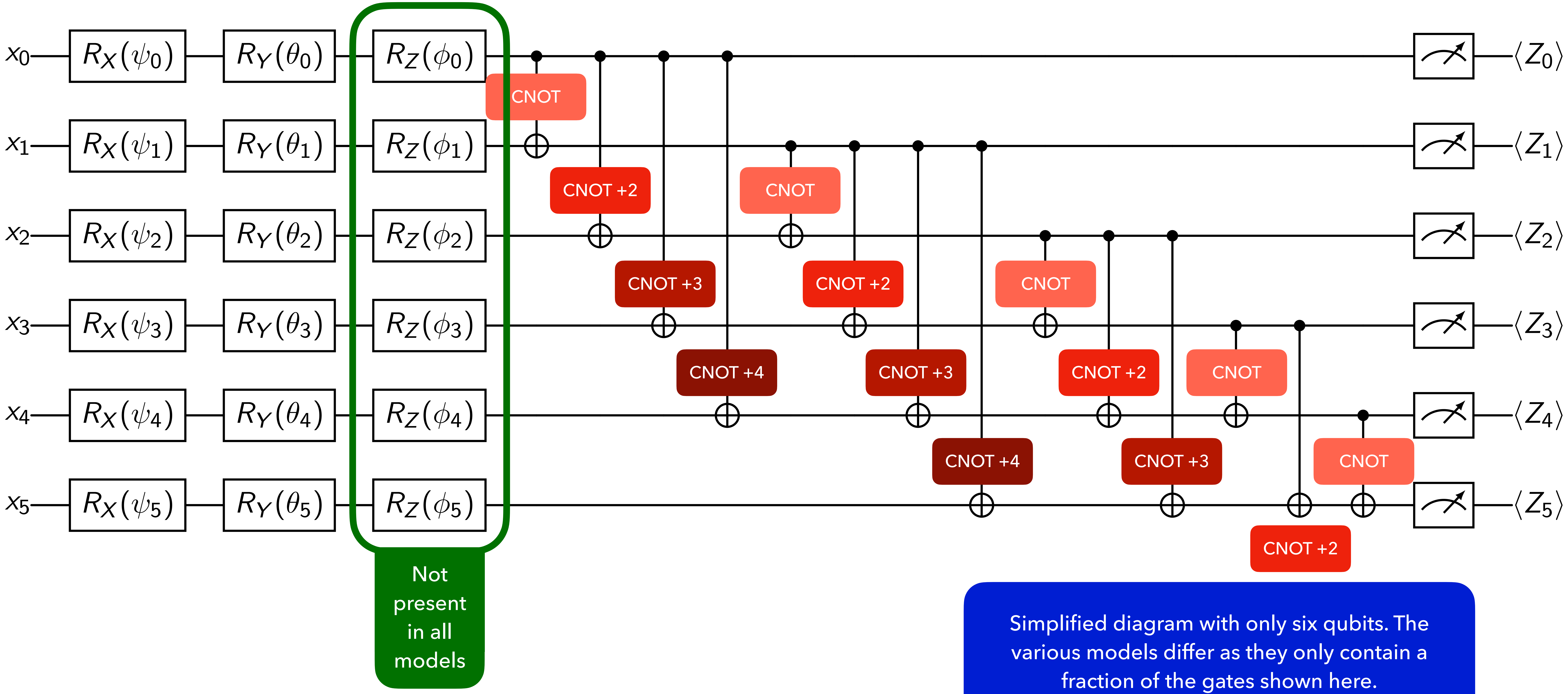


Simplified diagram with only six qubits. The various models differ as they only contain a fraction of the gates shown here.

Quantum FastCaloGAN

Circuit Template - CNOT Naming Scheme

CNOT +2: qubit *i* acting on qubit *i*+2
CNOT +3: qubit *i* acting on qubit *i*+3
CNOT +4: qubit *i* acting on qubit *i*+4



Simplified diagram with only six qubits. The various models differ as they only contain a fraction of the gates shown here.

Quantum FastCaloGAN

Performance Metrics

- In GANs, the best training iteration is not necessarily the last one → evaluation step.

- For a given incident energy value, events from the training dataset (= Geant4 events) are used to calculate the voxel total energy distribution. The GAN generator generates for each training iteration (for which a training checkpoint, storing the weights as they were at that point in training, has been saved) the same number of events and computes their total energy distribution. The two distributions are compared computing a reduced chisquare between them. This metric is used as it is simple to define but difficult to simulate.

  - This is done for all iterations. The best one is the one with the lowest reduced chisquare.

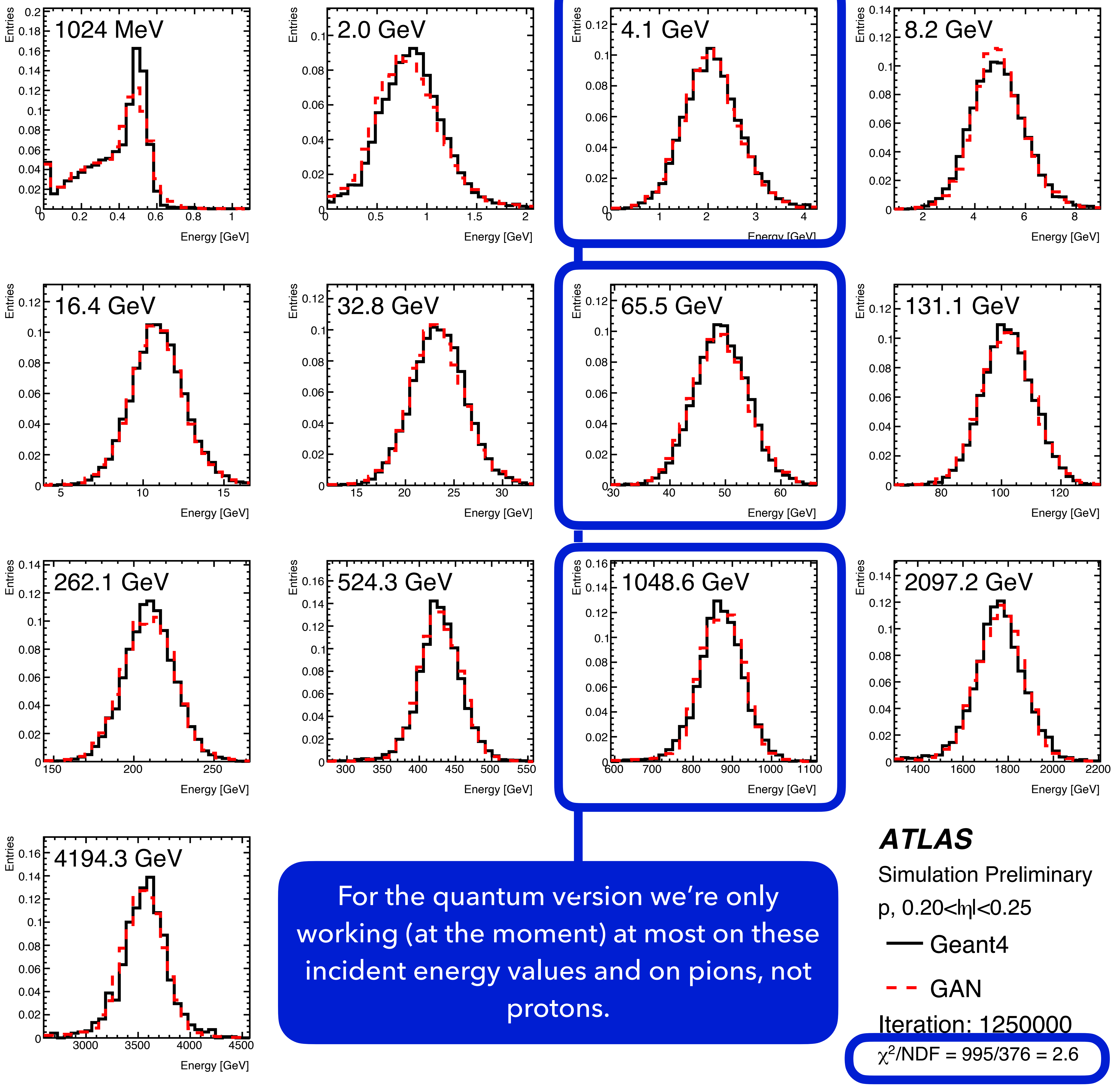
- Models are evaluated depending on the reduced chisquare value they can achieve and how early convergence is.

Quantum FastCaloGAN

Performance Metrics (cont'd)

- Non-quantum FastCaloGAN (red line) compared to Geant4 (black line) via a reduced chisquare (shown at the bottom right).
- Classical case plot shown as an example to clarify definition of the performance metrics.
- Classical FastCaloGAN can work on many more incident energy values.

Federico A.G. Corchia *et al.* on behalf of the ATLAS Computing Activity, *ATLFAST3: Fast Simulation in ATLAS for LHC Run 3 and Beyond*, EPJ Web Conf. **337**, 01355 (2025)



For the quantum version we're only working (at the moment) at most on these incident energy values and on pions, not protons.

ATLAS EXPERIMENT CERN INFN BOLOGNA ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Results

PQC+NN Models (1/4)

- We start with the PQC plugging into a simple NN with just **one hidden Dense layer of size 100** (halfway-ish between latent space of size 10 and output of size #voxels). We're working on the **starting single-energy dataset**.

- The (simulated) quantum layer adds considerable **overhead** on training times (indeed, the fully classical model trains for many more epochs than the quantum ones).

$<\chi^2>_q$: Mean χ^2 of Quantum model around its best checkpoint (±30 iterations)

$<\chi^2>_c$: Mean χ^2 of Classical model around the best checkpoint of the quantum model (±30 iterations)

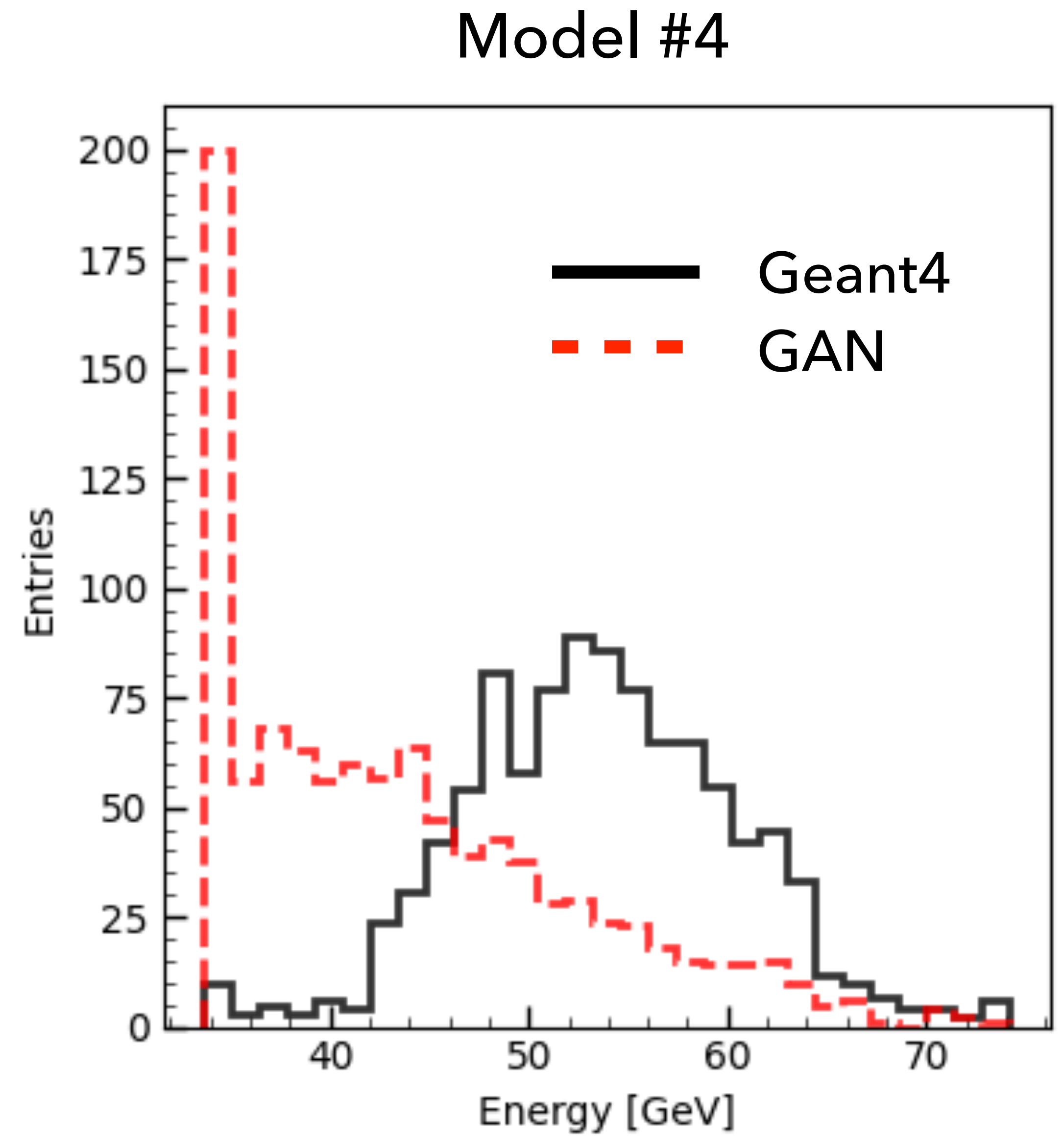
Model no.	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$<\chi^2>_q$	$<\chi^2>_c$
Classical	❌	❌	❌	❌	100, #voxels	16.8	418	419	-	22.2
1	✅	❌	✅	❌	100, #voxels	45.9	270	280	52.1	30.0
2	✅	❌	✅	✅	100, #voxels	34.0	207	231	42.6	32.2
3	✅	✅	✅	❌	100, #voxels	54.1	262	271	58.9	30.2
4	✅	✅	✅	✅	100, #voxels	25.6	229	275	36.4	31.0

Results

PQC+NN Models (2/4)

- When adding **CNOT +2**, the reduced χ^2 of the quantum models **starts decreasing**. There's even more decrease when also adding the **R_z gate**. The models are **starting learning**, but there's still a large gap with the fully classical model (the quantum addition is indeed **hindering** training at the moment!).

- Model #4 has its best iteration at iteration no. 229. Its $\langle\chi^2\rangle_q$ is 36.4, the $\langle\chi^2\rangle_c$ (N.B. computed in the same range of the quantum model to see how the classical model training was going when the quantum one was giving its best result) is 31.0.



Model no.	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$\langle\chi^2\rangle_q$	$\langle\chi^2\rangle_c$
Classical	❌	❌	❌	❌	100, #voxels	16.8	418	419	-	22.2
1	✅	❌	✅	❌	100, #voxels	45.9	270	280	52.1	30.0
2	✅	❌	✅	✅	100, #voxels	34.0	207	231	42.6	32.2
3	✅	✅	✅	❌	100, #voxels	54.1	262	271	58.9	30.2
4	✅	✅	✅	✅	100, #voxels	25.6	229	275	36.4	31.0

Results

PQC+NN Models (3/4)

- Adding **another classical DNN layer**, obtaining a DNN with layers of size [50, 200, #voxels] (10 ➝ 50 = 5x increase; 50 ➝ 200 = 4x increase; 200 ➝ #voxels = 2.665x increase).

- Quantum models #3 and #4 return **reduced chisquare comparable** to the classical model.

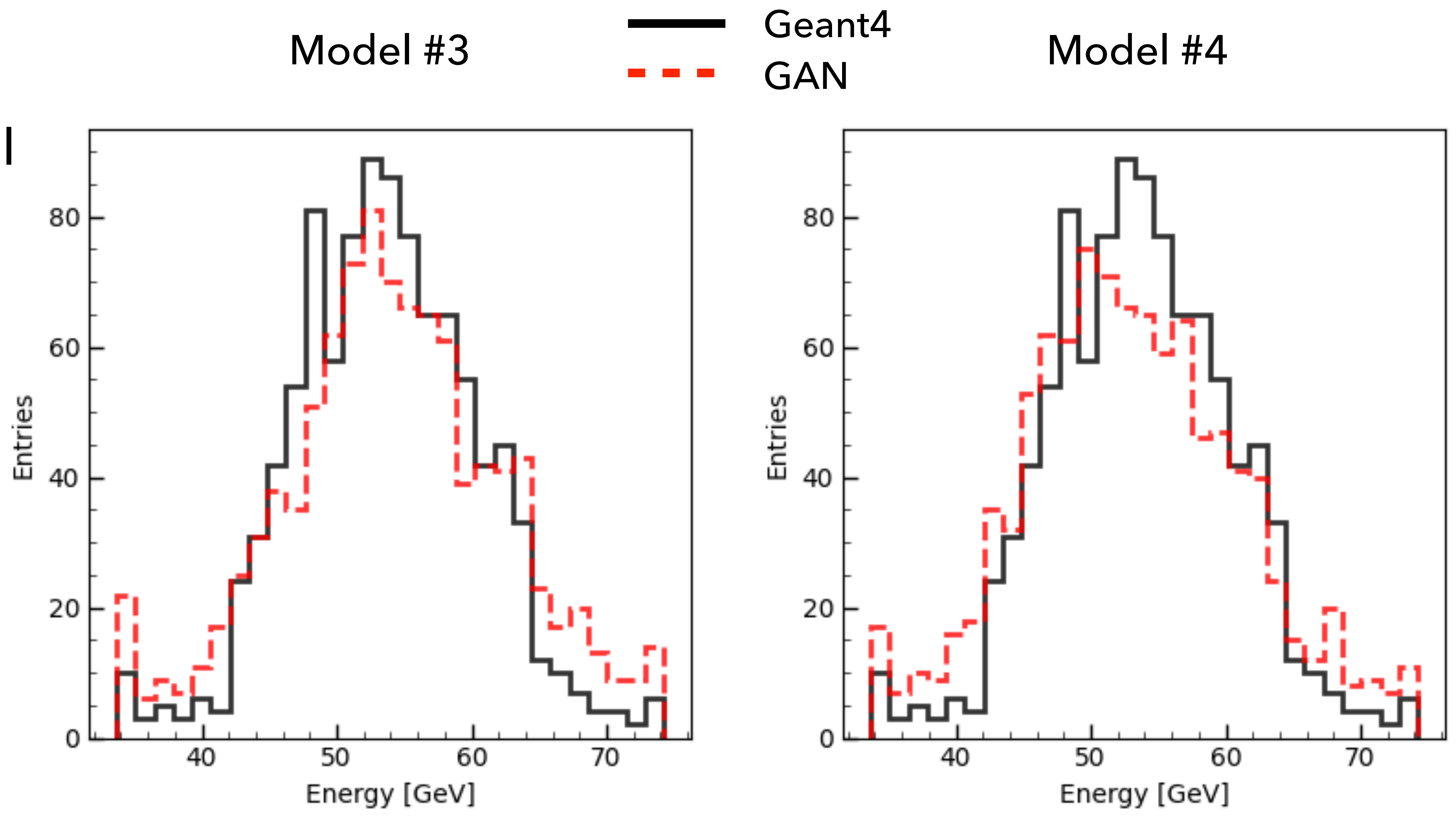
- The **average reduced chisquare in a range of ±30 checkpoints around the best one** ($<\chi^2>_q$) for model #3 (range = [125:165]) is 5.39 and for model #4 (range [96:156]) is 5.91. In both those ranges, the average reduced chisquare of the classical model ($<\chi^2>_c$) is higher. This means that it's no "lucky" fluctuation, but those quantum models actually **converge faster** than the classical one.

Model no.	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$<\chi^2>_q$	$<\chi^2>_c$	$<\chi^2>_c'$
Classical	❌	❌	❌	❌	❌	❌	50, 200, #voxels	2.13	401	561	-	5.03	3.14
1	✅	✅	✅	❌	❌	❌	50, 200, #voxels	Skipped as not good for [100, #voxels]					
2	✅	✅	✅	✅	❌	❌	50, 200, #voxels	2.71	222	266	10.5	6.51	2.48
3	✅	✅	✅	✅	✅	❌	50, 200, #voxels	2.22	155	165	5.39	8.21	3.07
4	✅	✅	✅	✅	✅	✅	50, 200, #voxels	2.12	126	163	5.91	9.48	3.26
Cl. reference	❌	❌	❌	❌	❌	❌	50, 100, 200, #voxels	1.25	196	556	-	6.98	2.97

Results

PQC+NN Models (4/4)

- Now that we've seen that adding the PQC is now beneficial to training, we want it to replace parts of the classical NN and be offloaded part of the latter's computation.
- We're comparing the quantum models to a reference model being a NN of architecture [50, 100, 200, #voxels].
- $\langle\chi^2\rangle_c'$ (being $\langle\chi^2\rangle_c$ but computed for the reference classical model) is still lower, so the PQC must still be made more powerful to make it possible to simplify the classical NN. However, this is already a promising hint.



Model no.	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$\langle\chi^2\rangle_q$	$\langle\chi^2\rangle_c$	$\langle\chi^2\rangle_c'$
Classical	❌	❌	❌	❌	❌	❌	50, 200, #voxels	2.13	401	561	-	5.03	3.14
1	✅	✅	✅	❌	❌	❌	50, 200, #voxels	Skipped as not good for [100, #voxels]					
2	✅	✅	✅	✅	❌	❌	50, 200, #voxels	2.71	222	266	10.5	6.51	2.48
3	✅	✅	✅	✅	✅	❌	50, 200, #voxels	2.22	155	165	5.39	8.21	3.07
4	✅	✅	✅	✅	✅	✅	50, 200, #voxels	2.12	126	163	5.91	9.48	3.26
Cl. reference	❌	❌	❌	❌	❌	❌	50, 100, 200, #voxels	1.25	196	556	-	6.98	2.97

Results

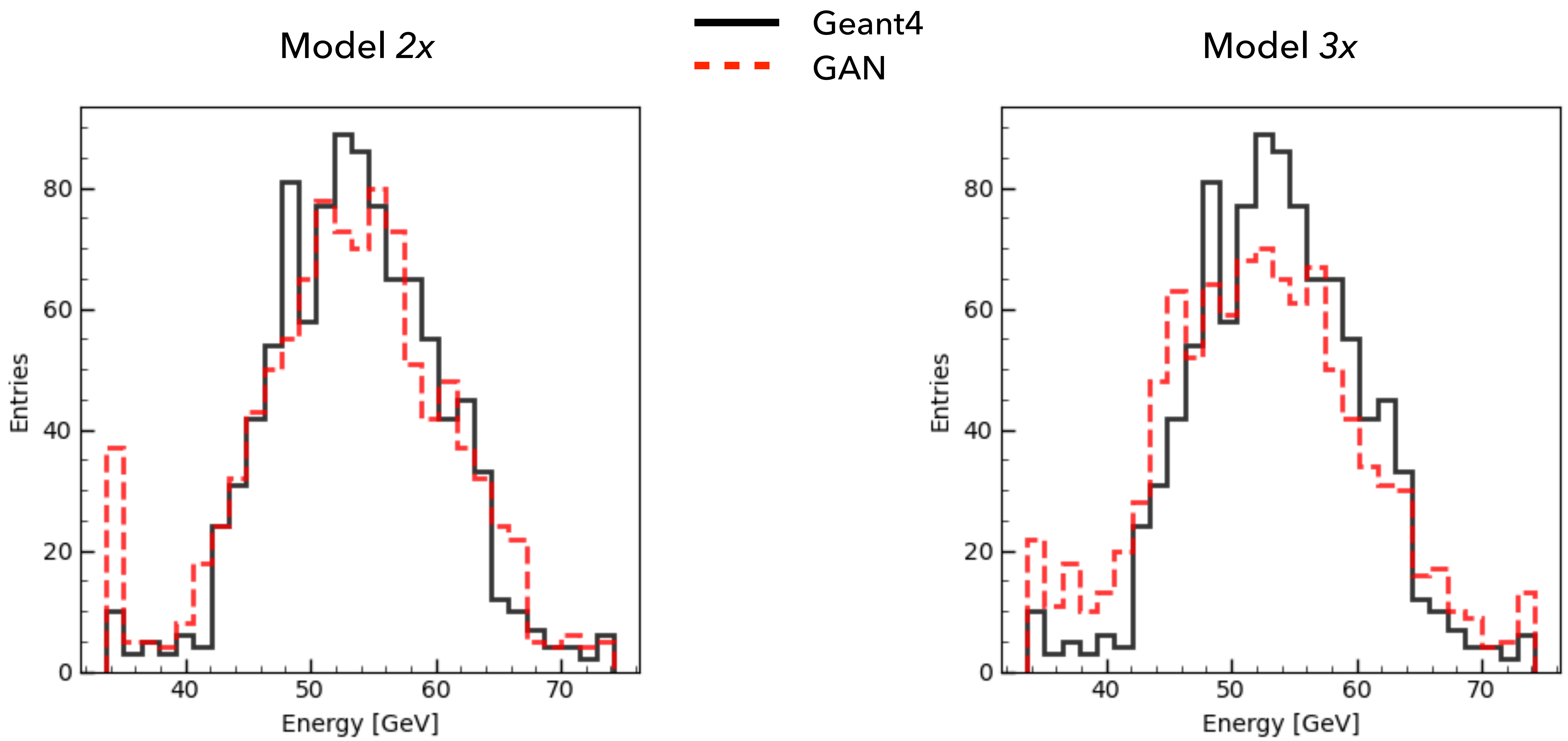
Repeated PQC+NN Models (1/2)

- Let's now try **repeating our PQC multiple times.**
  - Repeated block $R_y(\theta) \rightarrow R_z(\varphi) \rightarrow$ "CNOT +1", for 2 and 3 times.
- Reduced χ^2 is low and $<\chi^2>_q$ is lower than $<\chi^2>_c$ of the classical [50, 200, #voxels] NN.
- $<\chi^2>_q$ for model *2x* is **comparable, slightly better** than $<\chi^2>_c'$ (i.e. for the reference classical [50, 100, 200, #voxels] NN) → quantum model *2x* has performance equivalent, slightly better, to the classical reference → part of the computation **can effectively be offloaded to the quantum part!**

Model	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	#repetitions	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$<\chi^2>_q$	$<\chi^2>_c$	$<\chi^2>_c'$
Classical	❌	❌	❌	❌	❌	❌	-	50, 200, #voxels	2.13	401	561	-	5.03	3.14
2x	✅	✅	✅	❌	❌	❌	2	50, 200, #voxels	1.76	124	125	3.10	9.72	3.27
3x	✅	✅	✅	❌	❌	❌	3	50, 200, #voxels	2.40	66	104	7.92	21.1	5.66
Cl. reference	❌	❌	❌	❌	❌	❌	-	50, 100, 200, #voxels	1.25	196	556	-	6.98	2.97

Results

Repeated PQC+NN Models (2/2)



Model	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	#repetitions	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$<\chi^2>_q$	$<\chi^2>_c$	$<\chi^2>_c'$
Classical	❌	❌	❌	❌	❌	❌	-	50, 200, #voxels	2.13	401	561	-	5.03	3.14
2x	✅	✅	✅	❌	❌	❌	2	50, 200, #voxels	1.76	124	125	3.10	9.72	3.27
3x	✅	✅	✅	❌	❌	❌	3	50, 200, #voxels	2.40	66	104	7.92	21.1	5.66
Cl. reference	❌	❌	❌	❌	❌	❌	-	50, 100, 200, #voxels	1.25	196	556	-	6.98	2.97

Results

Conditioned Quantum FastCaloGAN (1/2)

- Up to now, the quantum GAN has been tested on events with incident energy 65 GeV. Our quantum GAN should be able to generate (as the classical one does) energy deposition events starting from particles of energy of our choice → let's move to the **3-energy dataset** and see how the models that yielded the best results for the single-energy dataset behave with this one.

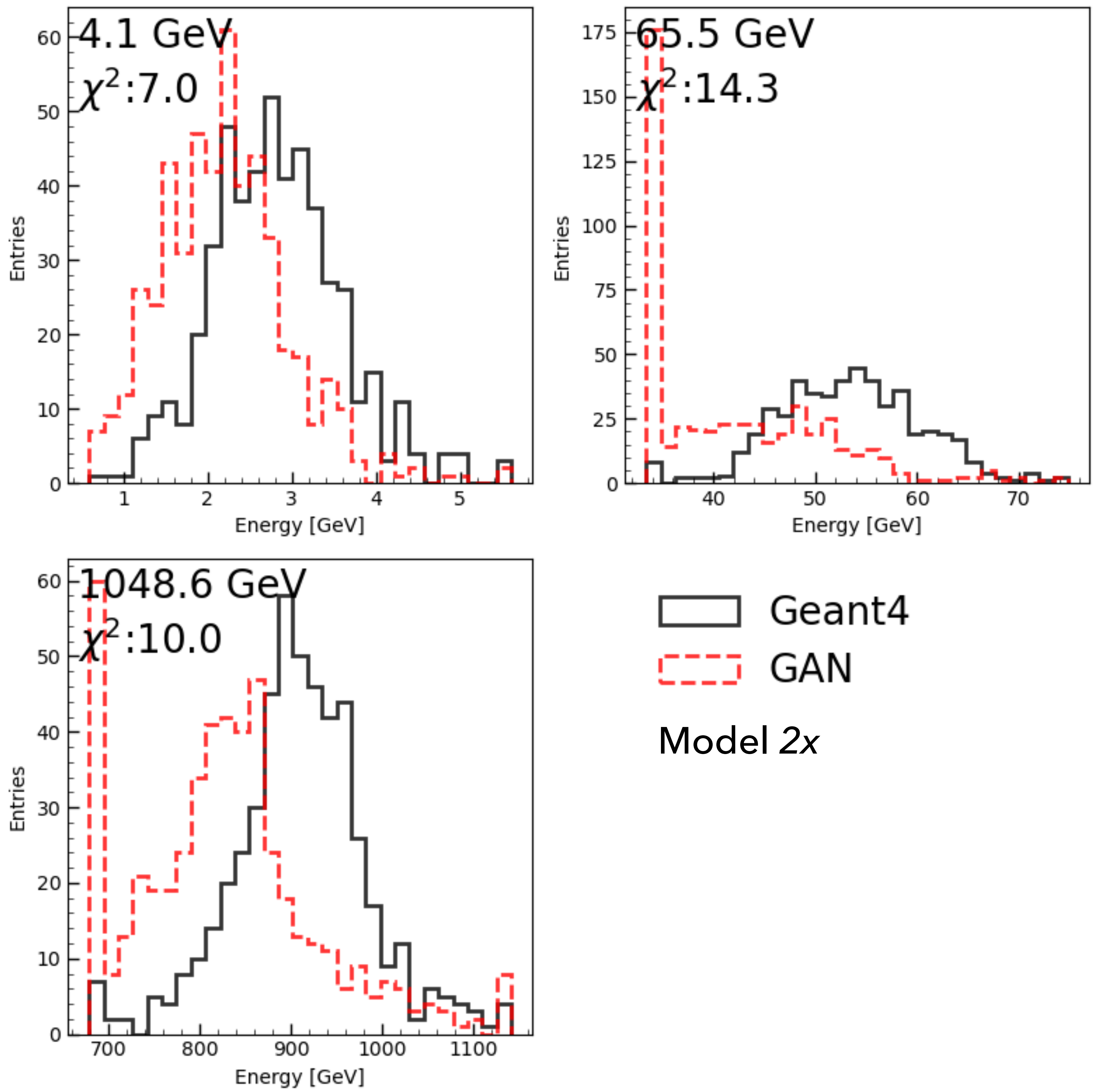
    - Only modification: energy conditioning implemented by an **additional qubit** in the generator (→ 10 latent space qubits + 1 energy conditioning qubit = 11 qubits in total).

Model	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	#repetitions	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$<\chi^2>_q$	$<\chi^2>_c$'	χ^2/NDOF per inc. energy 4 GeV	65 GeV	1 TeV
2x	✅	✅	✅	❌	❌	❌	2	50, 200, #voxels	10.6	78	85	15.8	5.17	7.0	14.3	10.0
3x	✅	✅	✅	❌	❌	❌	3	50, 200, #voxels	15.8	26	91	21.7	17.2	6.3	20.4	19.7
Classical reference	❌	❌	❌	❌	❌	❌	-	50, 100, 200, #voxels	1.32	517	564	-	2.30	1.9	1.0	1.1

Results

Conditioned Quantum FastCaloGAN (2/2)

- Both models have **rather high** total reduced χ^2. When computing it for each incident energy value, Geant4-GAN agreement **varies**.
- The quantum model is **starting learning** how to handle multiple incident energy values, but we need a **more powerful model and/or more epochs** (these quantum models only managed to run for a limited number of them!)



Model	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	#repetitions	NN Layers	χ^2/NDOF	At #iteration	Max iter.	$\langle\chi^2\rangle_q$	$\langle\chi^2\rangle_{c'}$	χ^2/NDOF per inc. energy		
														4 GeV	65 GeV	1 TeV
2x	✅	✅	✅	❌	❌	❌	2	50, 200, #voxels	10.6	78	85	15.8	5.17	7.0	14.3	10.0
3x	✅	✅	✅	❌	❌	❌	3	50, 200, #voxels	15.8	26	91	21.7	17.2	6.3	20.4	19.7
Classical reference	❌	❌	❌	❌	❌	❌	-	50, 100, 200, #voxels	1.32	517	564	-	2.30	1.9	1.0	1.1

Results

Quantum Discriminator

- Up to now, we've studied quantum generators. What if we work instead on a **quantum discriminator**?
- Generator: generative task. Discriminator: **classification** task (telling if a given event is true Geant4 or fake one by the generator, so it outputs just one value telling which case it is). Must be kept in mind when designing the quantum discriminator, but the considerations arisen from the work on the generator **can anyway yield reasonable ansätze** to start with.
- **Very preliminar try** using a **"reverse" version** of the best generator models (models *2x* and *3x*): images **input to a classical NN** (to both **scale down** from #voxels being the input size to #qubits=10 and provide the **learning power** for which DNNs are anyway needed) of architecture [200, 50, 10], which is then **followed by the PQC with output of size 1** (Z measurement performed on the first qubit). Generator: reference [50, 100, 200, #voxels] classical one.
- These quantum discriminators yield very poor results, so they're **currently incapable** of "guiding" the training of the generator.
    - However, training only possible for a **limited number of epochs** (*D/G ratio* = 5 → discriminator invoked **5 times** per generator invocation! Hyperparameter left at the moment as in original FastCaloGAN as it is normally beneficial for GAN training).
    - **More study is needed**.

Model	Discriminator initial NN	$R_y(\theta)$	$R_z(\varphi)$	CNOT +1	CNOT +2	CNOT +3	CNOT +4	#repetitions	χ^2/NDOF	At #iteration	Max iter.
Classical	800, 400, 200, 1	❌	❌	❌	❌	❌	❌	-	1.25	196	556
2x	200, 50, 10	✅	✅	✅	❌	❌	❌	2	70.6	Same χ^2 all epochs	31
3x	200, 50, 10	✅	✅	✅	❌	❌	❌	3	70.6	Same χ^2 all epochs	4 (then NaN loss)

Conclusions

- A simple quantum version of ATLAS FastCaloGAN is presented.
- The **repetition** of block R_y+R_z+CNOT in the PQC and two hidden layers in the DNN **seem to be able to provide benefit** over the classical reference.
    - Benefit at the moment **observed for the generator single-energy case**, not yet for the 3-energy one.
- This study is initial and in progress, there are still many more combinations and ideas to test!
- **Promising results to investigate**: more PQC connections, more discriminator ansätze, more complex QNN/QINN (Quantum Invertible Neural Network), hyperparameter tuning, tests on more energies/events and shower shapes.