Thesis

Machine Learning Techniques for Fast Shower Simulation at the ATLAS Experiment

SALAMANI, Dalila

Abstract

The simulation of the passage of particles through the detectors of the Large Hadron Collider (LHC) is a core component of any physics analysis. However, a detailed and accurate simulation of the detector response using the Geant4 toolkit is a time and CPU consuming process. This is especially intensified with the large number of simulated events, typical physics analysis need. This thesis documents Machine Learning (ML) based alternatives for a faster simulation of the showering of particles in the ATLAS calorimeter. An ML approach that extends current parametrized simulation is also proposed. The work presented in this thesis follows three main stages: data preprocessing, ML model design, validation and integration into the ATLAS simulation framework. For data preprocessing, the calorimeter cell information is used to derive a suitable data structure. A finer granularity of voxels is then used to better capture the structure of the shower and extend the range of energy and the calorimeter regions. In the preprocessing stage, an innovative ML technique is introduced to automatically learn the optimal structure of the [...]

Reference

SALAMANI, Dalila. *Machine Learning Techniques for Fast Shower Simulation at the ATLAS Experiment*. Thèse de doctorat : Univ. Genève, 2021, no. Sc. 5626

DOI : 10.13097/archive-ouverte/unige:158540
URN : urn:nbn:ch:unige-1585406

UNIVERSITÉ DE GENÈVE
Département de physique nucléaire et corpusculaire
Département d'informatique

FACULTÉ DES SCIENCES
Professeur Tobias Golling
Professeur Svyatoslav Voloshynovskyy

Machine Learning Techniques for Fast Shower Simulation at the ATLAS Experiment

THÈSE

Présentée à la Faculté des sciences de l'Université de Genève pour obtenir le grade de Docteur ès sciences, mention interdisciplinaire.

par

Dalila Salamani

d'Algérie

Thèse 5626

Genève
Atelier d'impression ReproMail
2022

UNIVERSITÉ DE GENÈVE

FACULTÉ DES SCIENCES

DOCTORAT ÈS SCIENCES, MENTION INTERDISCIPLINAIRE

Thèse de Madame Dalila SALAMANI

intitulée :

«Machine Learning Techniques for Fast Shower Simulation at the ATLAS Experiment»

La Faculté des sciences, sur le préavis de Monsieur T. GOLLING, professeur associé et directeur de thèse (Département de physique nucléaire et corpusculaire), Monsieur S. VOLOSHYNOVSKIY, professeur ordinaire et codirecteur de thèse (Département d'informatique), Monsieur S. SCHRAMM, professeur (Département de physique nucléaire et corpusculaire), Monsieur G. A. STEWART, docteur (HEP Software Foundation Coordinator (EP-SFT), CERN, Genève), autorise l'impression de la présente thèse, sans exprimer d'opinion sur les propositions qui y sont énoncées.

Genève, le 4 janvier 2022

Thèse - 5626 -

Le Doyen

N.B. - La thèse doit porter la déclaration précédente et remplir les conditions énumérées dans les "Informations relatives aux thèses de doctorat à l'Université de Genève".

À ma mère, à mon père ..

Abstract

The simulation of the passage of particles through the detectors of the Large Hadron Collider (LHC) is a core component of any physics analysis. However, a detailed and accurate simulation of the detector response using the Geant4 toolkit is a time and CPU consuming process. This is especially intensified with the large number of simulated events, typical physics analysis need.

This thesis documents Machine Learning (ML) based alternatives for a faster simulation of the showering of particles in the ATLAS calorimeter. An ML approach that extends current parametrized simulation is also proposed. The work presented in this thesis follows three main stages: data preprocessing, ML model design, validation and integration into the ATLAS simulation framework. For data preprocessing, the calorimeter cell information is used to derive a suitable data structure. A finer granularity of voxels is then used to better capture the structure of the shower and extend the range of energy and the calorimeter regions. In the preprocessing stage, an innovative ML technique is introduced to automatically learn the optimal structure of the data. The resulting structure is general enough to be compatible with any particle energy and detector region.

Once the data is preprocessed and an adapted structure is defined, a Variational AutoEncoder (VAE) termed FastCaloVSim learns to reproduce the showering of particles in the ATLAS calorimeter. A new, physics inspired, loss function is proposed to accurately map the shower energy, the total energy deposited per calorimeter layer and the total energy deposited in all the layers. Furthermore, the VAE is designed as a conditional model, i.e. the learning is conditioned on the pseudorapidity region of the calorimeter as well as the energy of the particle. The model performance is evaluated both as a standalone algorithm and as part of the ATLAS simulation framework. The last stage of this work describes the integration of FastCaloVSim into Athena, the ATLAS simulation framework, allowing further validation of the overall simulation pipeline.

Résumé

La simulation du passage des particules dans les détecteurs du Grand collisionneur de hadrons (LHC) est un élément crucial de toute analyse dans la physique des hautes énergies. Cependant, une simulation détaillée et précise de la réponse du détecteur à l'aide de l'outil Geant4 est un processus lent et consommant de mémoire (CPU).

Cette thèse documente des alternatives basées sur l'apprentissage automatique (Machine Learning : ML) pour une simulation plus rapide des gerbes crées par des particules primaires dans le calorimètre de l'expérience ATLAS. Une extension de la simulation paramétrée, actuellement utilisée dans ATLAS, basée sur le ML et est également proposée. Le travail présenté dans cette thèse suit trois étapes principales : le prétraitement des données, la conception et la validation du modèle ML et l'intégration dans l'outil de simulation de l'expérience ATLAS. Pour le prétraitement des données, les informations des cellules du calorimètre sont utilisées pour définir une représentation des gerbes. Une granularité plus fine est alors utilisée pour mieux capturer la structure des gerbes ainsi que pour étendre le spectre d'énergie et de direction des particules primaires. À cette étape de prétraitement, une technique ML innovante est introduite pour apprendre automatiquement la structure optimale de ces données. Cette structure est suffisamment générale pour être compatible avec toute énergie ou direction de la particule primaire.

Une fois les données traitées et une structure adaptée est définie, un modèle appelé FastCaloVSim (basé sur le Variational AutoEncoder) est défini pour apprendre à reproduire les gerbes de particules dans le calorimètre d'ATLAS. Une nouvelle fonction d'apprentissage inspirée de la physique est proposée pour un apprentissage plus précis. De plus, ce modèle est conçu sur un principe conditionnel, c'est-à-dire que l'apprentissage est conditionné sur l'énergie ainsi que la direction de la particule primaire. Les performances du modèle sont évaluées sur deux étapes : la première est basée sur la comparaison des données d'apprentissage aux données générées par le modèle, tandis que la seconde utilise cette comparaison en intégrant le modèle dans l'outil de simulation d'ATLAS.

Contents

Introduction

In Large Hadron Collider experiments, such as ATLAS, the calorimeter is a key detector technology to measure the energy of particles, leading to their identification. The particles emerging during collisions interact electromagnetically and hadronically with the material of the calorimeter, creating cascades of secondary particles or showers. Describing the showering process relies on simulation methods that precisely describe all the particle interactions with matter. Constrained by the need for precision, the full particle simulation is inherently slow and constitutes a bottleneck for analysis. Furthermore, the High Luminosity upgrade of the LHC in 2025 will result in more complex events that heavily relies on the simulation.

Simulation techniques combine randomness with computational algorithms in order to approximate a function as complex as the one describing the showering of a particle. Specifically, Monte Carlo (MC) methods are a collection of computational algorithms used to estimate probability distributions of an uncertain event. These methods were firstly used in the late 1940s when scientists were facing intractable problems such as models of neutron diffusion and where analytical solutions were not available.

MC methods are applied to a large range of problems in different domains. In mathematics, for example, they are used to evaluate multidimensional definite integrals. In business, they are largely used in the calculation of risks. In physics, and more precisely in experimental particle physics, MC methods are used for designing detectors, simulating their response to particle interactions and comparing experimental data to theory.

The main focus in this thesis is the design of a fast simulation model based on Machine Learning (ML) techniques that can learn to generate showers in the ATLAS calorimeter. This model is termed FastCaloVSim and is completely independent of existing physics based simulation techniques. State-of-the-art fast simulation techniques in the ATLAS experiment (referred to as FastCaloSim or FCS) rely on parameter estimation that does not necessarily model all subsequent calorimeter interactions. These interactions are however crucial in the faithful reconstruction of physics objects. Because of the nature of deep neural networks, correlations between objects (such as correlation between pixels) can be well modeled and fully incorporated in the generation process. In this thesis, the correlation modeling capacity of generative models is combined with FCS techniques, resulting in a novel model termed CoVAE.

A graphical summary is presented in Figure 1 with the two major components of the thesis. Different data representations are studied and implemented in the thesis, with cells and voxels derived using a physics based approach and the centroids are derived using an ML based approach.

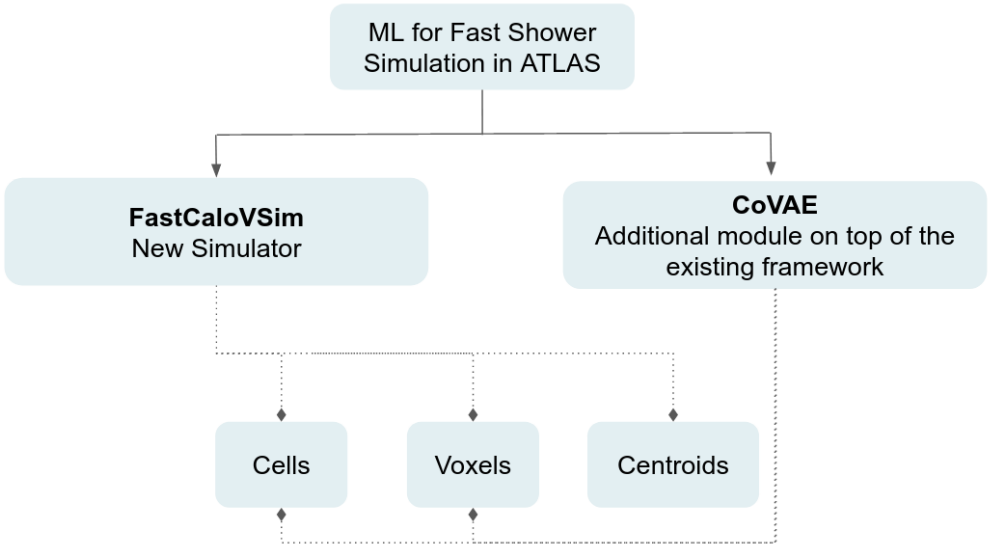


Figure 1: Overview of the two major components of the thesis: FastCaloVSim and CoVAE.

Chapter 1 introduces the Standard Model of particle physics. The ATLAS detector layout is described in Chapter 2 and Chapters 3 and 4 are dedicated to the calorimeter part and physics objects. The ATLAS detector simulation in Chapter 5 introduces the detailed simulation and the fast approach FCS. Chapter 6 gives an overview of ML models and concepts cited in this thesis. All the details of implementations, strategies, tests, discussions, and comparisons are detailed in Chapters 7, 8, 9, 10 and 11. Chapter 12 shows how an ML model can be applied on top of the classical parametrization to learn the correlated fluctuations.

1 The Standard Model of Particle Physics and Beyond

According to Aristotle, the quest for the understanding of the universe and its fundamental components is driven by curiosity; *"human beings, by nature, desire to know"*. The central question in particle physics "what is matter?" is perhaps the oldest and still ongoing research journey. The concept of fundamental particles started about 2,000 years ago with first speculations of Greek philosophers that particles are hard and indivisible. Recent decades have led to significant progress in the understanding of these fundamental structures and culminated with the formulation of the Standard Model (SM) of particle physics. The SM is a theory based on mathematical descriptions of matter components and the interactions governing them at the smallest scales. It unifies and explains elementary particles and their interactions.

1.1 The Standard Model of Particle Physics

The SM was initially formulated in the 1970s to combine knowledge about particles and forces, resulting in a quantum theory of matter [1]. The discovery of the Higgs boson in 2012 added the missing piece in the SM theory. As a result, the SM is considered as the most complete description of elementary particles and their interactions down to the 10^{-16}cm scale [2].

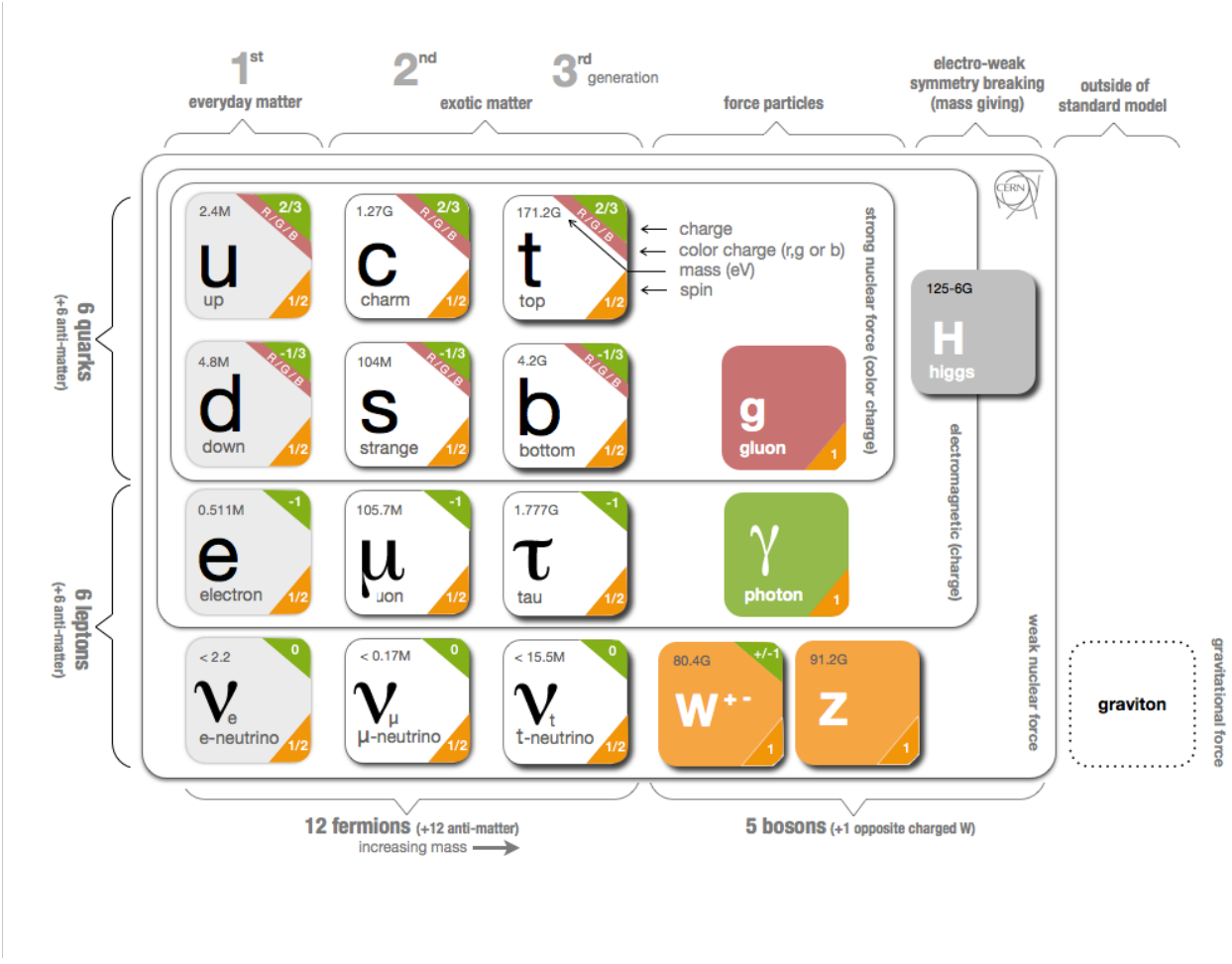


Figure 2: Schematic representation of the particle content of the Standard Model and the interactions among them. Values of charge, mass, color charge and spin of the different particles are reported in this representation [5].

1.2 Elementary and Composite Particles

Elementary particles, or fundamental particles, constitute the set of subatomic particles with no substructure [4]. The classification of elementary particles is shown in Figure 2. The first level classification divides them into fermions and bosons. **Fermions**, the building blocks of matter, are particles with half-integer spin: a quantum quantity to parameterize a particle's intrinsic angular momentum. Fermions are divided into two families: leptons and quarks. **Leptons** are particles that can exist independently. They have electroweak charge, i.e., +1 or -1 but no color charge. The term color is a quantum property which describes the relative charges of particles

as red, green and blue and anti-particles are given the quantum property of the anti-color. This property allows us to define a unique color and unique quantum state when particles are combined together. In the lepton family there are unit electric charge particles such as electrons e , muons μ and taus τ and zero charge particles representing the corresponding neutrinos; electron neutrino v_e, muon neutrino v_μ and tau neutrinos v_τ. **Quarks** are particles that cannot exist freely due to the fact that the energy required to free one quark is large enough to create new particles and therefore will cause the quark to be re-confined into a colorless state. The idea of confinement states that the force and energy to free a quark increases exponentially with the distance to other quarks. Therefore, separating quarks would require infinite energy. For this reason, quarks can not be separated, and they bind together into doublets and triplets. The SM defines six types of quarks up u, charm c, top t down d, strange s and bottom b. Quarks doublets are called mesons and triplets are called baryons. Protons, p, are baryons composed of two up quarks and one down quark. Hadrons is the name given to refer to bound quark states. Fermions are arranged into three generations of increasing mass and from ordinary to exotic, as shown in Figure 2. Each of the generations consists of two leptons and two quarks. Antiparticles with identical properties and opposite charges exist for each particle in each generation. **Bosons** are particles with integer spin. Gluons g, photons γ, W, Z and Higgs H are all bosons which mediate interactions between elementary particles. Gluons are particles that carry the strong nuclear force, γ carries the electromagnetic force, W and Z are the carrier of the weak nuclear force. The spin-0 **Higgs boson** in the SM carries mass to elementary particles via the Higgs mechanism. The last component shown in Figure 2 is the **graviton**, which is not part of the SM. It is the proposed name for the mediator of the gravitational force if a quantum description of gravity is known and thus a force carrying particle is included.

1.3 Forces and Interactions

The SM defines not only elementary particles but also their interactions. All physical phenomena can be described in terms of the three fundamental interactions [3]. The strong interaction provides the binding force between particles. The electromagnetic interaction, interrelated with electric and magnetic fields, determines atomic structures, chemical reactions and all electromagnetic phenomena. The weak interaction on the other hand, is the mechanism responsible for radioactive decays. Each interaction is defined by its strength or coupling constant, and its specific conservation laws. The coupling constant is part of the matrix element for the physics process, where its square defines the decay probability or cross section. In the electromagnetic interaction for example, the coupling constant is $\frac{e^2}{\hbar c} \approx \frac{1}{137}$, where e is the elementary charge, c the speed of light, $\hbar$ represents the quantum of angular momentum known as the reduced Planck constant which is defined as $\frac{h}{2\pi}$ with h being the Planck constant. The cross section for Compton scattering of a photon by an electron, as an example is then of order of $(\hbar/mc)^2(e^2/\hbar c)^2$, where the $\hbar/mc$ represents the wavelength of the electromagnetic wave when the energy of the photon is equivalent to the rest energy of the electron. This quantum mechanical property is the Compton wavelength. Based on observations of cross sections, the strong coupling is larger that the electromagnetic one. If, for example, the incident energy equals 1 GeV, the pion-nucleon cross section in total is about 10^{-26} cm^2 larger than 10^{-29} cm^2 for the electromagnetic production of pion-photon.

All matter particles take part in the weak interaction. Unlike the strong interaction, which only applies to quarks. Quarks and charged leptons on the other hand interact electromagnetically. Each force is moderated by force-carrying particle exchange. The weak force, for example, is carried by $W\pm$ and Z bosons. Electromagnetic and strong forces are carried by photons and gluons respectively.

In the language of group theory, the SM is a composition of quantum chromodynamics (QCD) theory and the electroweak (EW) theory. The SM obeys a symmetry as defined in Equation 1 with the strong $SU(3)$ interaction arising from the QCD theory and $SU(2)$ and $U(1)$ electroweak interactions arising from the EW theory.

$$G_{SM} = SU(3) \times SU(2) \times U(1) \tag{1}$$

QCD [9] describes the interaction among the quark fields and interposed by gluons in addition to the dynamics of the gluon fields. Quarks are 1/2 spin particles as reported in Figure 2 and they obey Fermi-Dirac statistics. To meet the anti-symmetric property of the total wave function, a new quantum number has to be introduced. It manifested as the color in the $SU(3)$ symmetry. $SU(3)$ is described by the triplet red, blue, and green color-charge states and quarks can figure in one of the three states.

Quantum electrodynamics (QED) is the theory of mutually interacting electrically charged particles and the quantized electromagnetic field and their interaction with an external electromagnetic field and external currents [6]. The EW theory combines the QED theory and the weak theory. It is based on $SU(2)\times U(1)$ gauge symmetry. The $SU(2)$ symmetry describes the weak interactions and it is known as weak isospin. On the other hand, the $U(1)$ symmetry is related to the weak hypercharge associated itself to the electric charge Q and the weak isospin. The purpose of the EW theory is to describe a unified fundamental force. A relatively old assumption

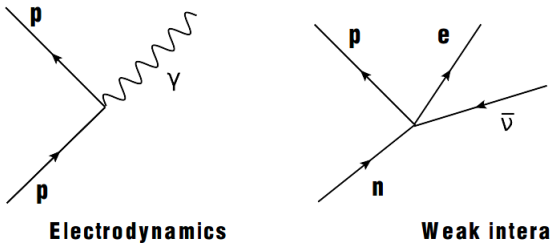


Figure 3: Fermi's analogy of β decay with electromagnetic interaction [13].

concerns the conservation of parity in all interactions, which means that an interaction can be replaced with its mirror image. This parity was experimentally contradicted for weak interactions. The latter interactions started with the discovery of radioactivity. β decays of nuclei is known to be caused by the weak interaction. Fermi's theory makes it possible to accurately describe β decay. Similarly to the electromagnetic interaction, Fermi schematically represented the weak interaction of the β decay of the neutron n. Figure 3 illustrates Fermi's analogy to an electromagnetic interaction. The decay's product is an electron, anti-neutrino $(e, \overline{v})$ pair. In this decay, the neutron converts itself into a proton p as well.

The Higgs Mechanism

Experimental results have shown an attributed mass to all particles, excluding photons and gluons. Adding a mass term into the Lagrangian formulation of the SM would lead to a loss of the invariance of the theory. Therefore, this was overcome with the EW symmetry breaking mechanism, referred to as Higgs mechanism [7]. The idea behind it stands on defining a Higgs field which quarks, leptons W and Z bosons interact with and as a result acquire mass. Photons and gluons do not interact with the Higgs field and therefore are massless particles. This symmetry breaking requires adding an isospin doublet of complex scalar fields Φ in order to preserve the symmetry of $SU(2) \times U(1)$. Φ a 2D representation, is defined as

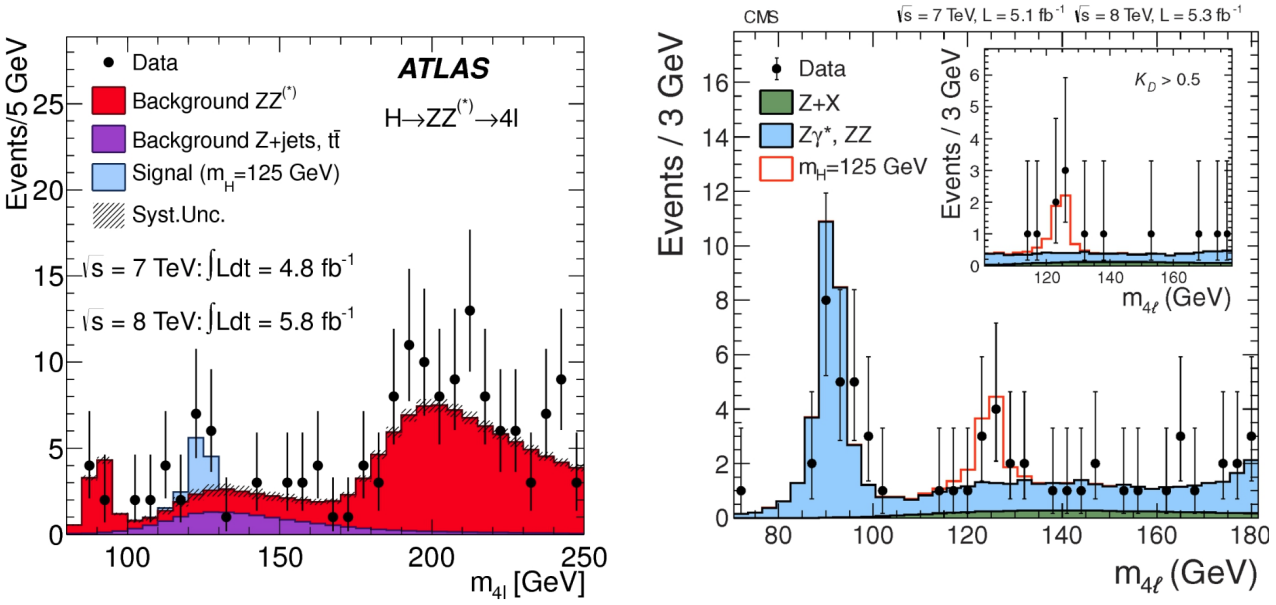
$$\Phi = \begin{pmatrix} \Phi^+ \\ \Phi^0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \Phi_1 + i\Phi_2 \\ \Phi_2 + i\Phi_3 \end{pmatrix},$$

where Φ_i are scalar real fields.

Studying and searching at the scale of EW symmetry breaking is one of the core goals of experimental physics. ATLAS and CMS collaborations of the LHC announced in 2012 [14, 15] the discovery of a new boson with properties agreeing with the H boson predicted by the SM. Figure 4 shows, for both experiments, the distribution of the four-lepton invariant mass $m4l$ for the selected candidates compared to the background expectation. The range of mass is between 80 and 250 GeV. Data is combined from a total center-of-mass energy of the colliding particle $\sqrt{s} = 7$ TeV and $\sqrt{s} = 8$ TeV data. The expected signal for an SM Higgs with $H = 125$ GeV is also shown.

1.4 Limitation of the Standard Model and Beyond

The SM is one of the most powerful and successful scientific theories. It accurately modeled our understanding of particles and their interactions. Figure 5 shows the SM production cross-section over a wide range of experimental measurements probing all three of the fundamental forces applicable to particles at current energy scales. Despite the success of the SM over the years, certain phenomena cannot be interpreted within the current theoretical formulation. One of these phenomena is the asymmetry of matter and antimatter. At the Big Bang, equal quantities of matter and antimatter were produced. However, now there is an imbalance in the universe. The SM does not explain this observed asymmetry. The combination of charge conjunction symmetry and parity symmetry is known as the CP-symmetry. The C-symmetry states that the physics laws should remain the same if a particle is interchanged with its antiparticle. On the other hand, the P-symmetry is the reflection effect through the origin of the space coordinates where spatial coordinates of the antiparticle are inverted. Decays of neutral kaons, for example, violate this CP-symmetry. Introducing the CP violation in the SM was an attempt to explain the asymmetry of matter and antimatter phenomena, but experimental measurements confirm that it is too small to be the only reason for this asymmetry. Another phenomena not explained by the SM is dark matter and dark energy. In fact, 95% of the entire mass-energy content of the observable universe is dark matter and dark energy. These two components are not part of the SM. The studies show about 1% of dark matter is caused by neutrinos [26].



(a) (b)

Figure 4: Distributions of the invariant mass of selected events with two pairs of electrons or muons measured by the ATLAS [14] (a) and CMS [15] (b) experiments at the LHC. The inset plot shows the result of applying a tight selection on a kinematic discriminant constructed to separate signal and background.

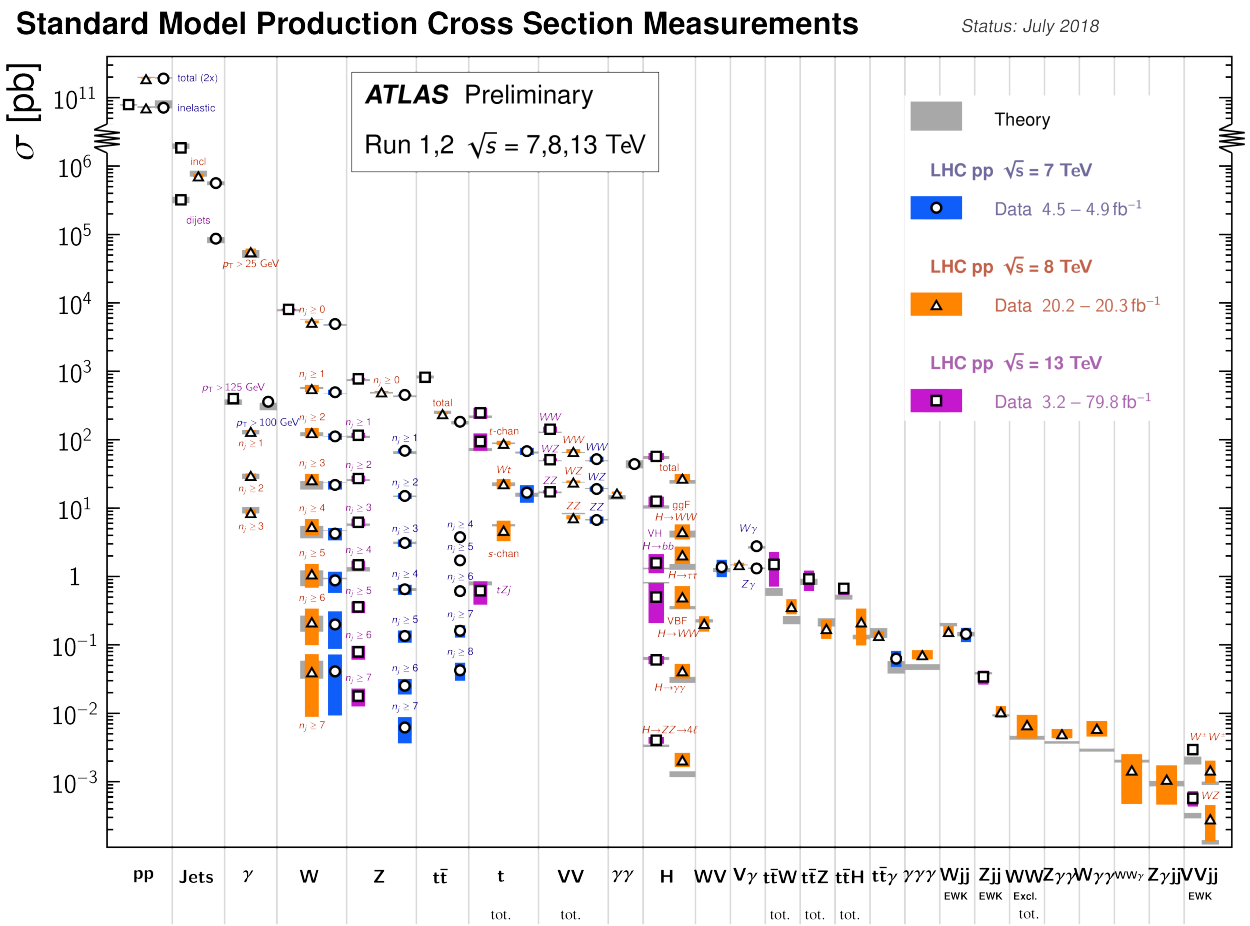


Figure 5: Summary of SM production cross-section measurements at $\sqrt{s} = 7, 8, 13$TeV, corrected for leptonic branching fractions, compared to the corresponding theoretical expectations. The luminosity used for each measurement is indicated close to the data point in the legend [16].

The SM is formulated as product of $SU(3) \times SU(2) \times U(1)$ with different gauge couplings. A particle's chirality is determined if either it is transformed into a right or left-handed representation of the Poincaré group. The SM does not explain why only $U(1)$ is parity and chiral violating. In addition, the SM has no explanation for charge quantization. Furthermore, in the SM formulation, only three out of four interactions are described. Currently, no theory including the gravity as a component has evidence to support it. SM with general relativity describes accurately natural processes. However, there are events where gravity impacts are non-negligible. At the particle level, the SM does not explain certain phenomena, such as the existence of three fermion generations and the spread of their masses over orders of magnitude. For the Higgs boson, the size of the mass is at the core of theoretical and experimental studies and searches. In the SM, none of the symmetry principles is dedicated to preventing quantum corrections for the Higgs boson mass. These corrections lead to sensitivity of the mass to particles and forces at high energies, therefore creating an unnatural hierarchy. This hierarchy indicating the difference of the SM in both electroweak and Plank scales with ($O(10^2)$ GeV) and (10^{19} GeV) respectively.

Beyond the Standard Model (BSM) groups theories and models that have been studied and developed to address the limitations of the SM. BSM theories include extensions of the SM such as the Minimal Supersymmetric Standard Model (MSSM) and new theories such as string theory. Supersymmetry (SUSY) extends the SM by predicting that each fermion or boson in the SM has a superpartner with a different spin of half a unit. For example, a superpartner of an electron is called a selectron and Higgsino for the Higgs boson.

The gauge problem, considered as a limitation for the SM, suggests a grand theory in which the SM gauge group $SU(3) \times SU(1) \times U(1)$ is incorporated in a simple group G with leptons and quarks combined in the same multiplets. If the unification of these three interactions is possible, it would refer to a grand unification epoch in the early moments of the universe in which they were not distinct. Experimentally, at high energies, the weak and the electromagnetic interactions were unified under the EW interaction. The prediction of Grand Unified Theories (GUTs) models states that the electroweak and the strong interactions will unify into an electronuclear interaction. Unification of electronuclear interaction and gravity would result in more comprehensive theory of everything (with a high energy out of reach of the LHC) rather a GUT.

2 The ATLAS Experiment at the Large Hadron Collider

A scientific approach starts with theoretical hypotheses detailed in rigorous formulas that produce predictions. These predictions are then compared with experimental results. The previous chapter explored theory (and predictions) in particle physics. In this chapter, these theories are validated against experimental data to compare the degree of agreement. These experiments are conducted using a particle accelerator: A machine that uses electromagnetic fields to accelerate hadrons to very high energies, focusing them in narrow paths to form beams [30]. Circular colliders accelerate opposite direction beams in a circular way before colliding them at specific locations along the ring. Characteristics of the beams, such as number of protons, collision energy and width of the beam, considerably impact the nature of the collisions. The number of relevant interactions, referred to as number of events, is an important feature in a collision. This is more apparent when rare events with small production cross section, σ_P, are investigated. The luminosity quantifies a particle accelerator's ability to produce the needed number of interactions. The relation between the luminosity $\mathcal{L}$, the cross section σ_P and the number of events per second dR/dt is given by $\frac{dR}{dt} = \mathcal{L}.\sigma_P$. For a given physics process, increasing the luminosity leads to increasing the number of events. Another important feature of a collision is known as the center of mass energy, E_{CM}. It represents the available energy for that collision. The kinematics of a particle can be defined with its energy E and momentum $\vec{p}$ forming $p = (E, \vec{p})$ with $p^2 = E^2 - \vec{p}^2 = m^2$, where m is the rest mass of the particle. Therefore, for two colliding particles, the total center of mass energy is defined as $E_{CM} = \sqrt{s} = \sqrt{(p_1 + p_2)^2}$.

The Large Hadron Collider (LHC) is a particle accelerator which has been designed to produce the experimental data needed to advance our understanding of the universe. The LHC collides high energy proton beams, that is pp collisions at nearly the speed of light. This allows us to reproduce the conditions nano seconds after the Big Bang, where all particles and forces emerged. LHC collisions are captured by massive devices called "detectors". Depending on the design and material of the detectors, different functionalities and physics targets are defined. Each detector serves a specific physics experiment run by a scientific collaboration. ATLAS, one of the two major LHC experiments, is a general purpose detector that investigates a wide range of physics.

This chapter describes the LHC, the architecture and various functions of the ATLAS detector and the World-wide LHC Computing Grid (WLCG).

2.1 The Large Hadron Collider

The LHC is the most powerful particle accelerator ever built. It is located in the European particle physics laboratory, CERN, beneath the Swiss-French boarder near Geneva. It is a 27 km circular ring of superconducting magnets, 100 meters underground, with a total weight of 38,000 tonnes. It accelerates hadrons, such as protons or heavy ions, from two opposite direction beams and brings them to collide at four points around the ring as illustrated in Figure 6. The circular ring of the LHC uses accelerating components to increase the energy of particles. The process of acceleration occurs via Radio Frequency (RF) cavities with a field gradient of 16 MV/m. There are eight cavities, each supplying 2 MV. With this technique, a beam cannot be continuous, but is rather grouped into bunches of protons. The number of bunches in the LHC ring is 2808 with each bunch containing 1.15×10^{11} particles. The bunches frequency is 40 MHz which translates to a collision every 25 ns. About two thousand superconducting dipole magnets provide an 8.4 T magnetic field to steer and focus the beams. These magnets are cooled with superfluid helium to a temperature of 1.9 K in order to maintain their state of super conduction. In order to measure interesting physics events, the LHC needs to operate at very high luminosity. As mentioned above, the luminosity describes the number of interactions per unit area and time. It can also be formulated as

$$\mathcal{L} = f\frac{N_c n_1 n_2}{A},$$

where N_c is the number of bunches per beam, n_1 and n_2 represent the number of particles per bunch, A is the overlapping area of the colliding bunches and f is the resolution frequency. The integrated luminosity, $L = \int \mathcal{L}dt = \frac{N}{\sigma_P}$, relates N, the expected number of collisions for a physics process to the cross section σ_P.

In the life cycle of an LHC collision, electrons are first stripped off Hydrogen atoms and the remaining protons (or inons) are gradually accelerated until reaching the LHC ring. Protons are first accelerated by a linear accelerator, which increases their energy to 50 MeV. They are then circulated in the Proton Synchrotron (PS), a 25 m circular accelerator, until their energy reaches 25 GeV. In the third stage, the protons are accelerated in the Super Proton Synchrotron (SPS) with a radius 7 km to an energy of 450 GeV. Protons are then injected into the LHC are accelerated to reach 13 TeV during the LHC Run2. The beams of protons travel at a speed of

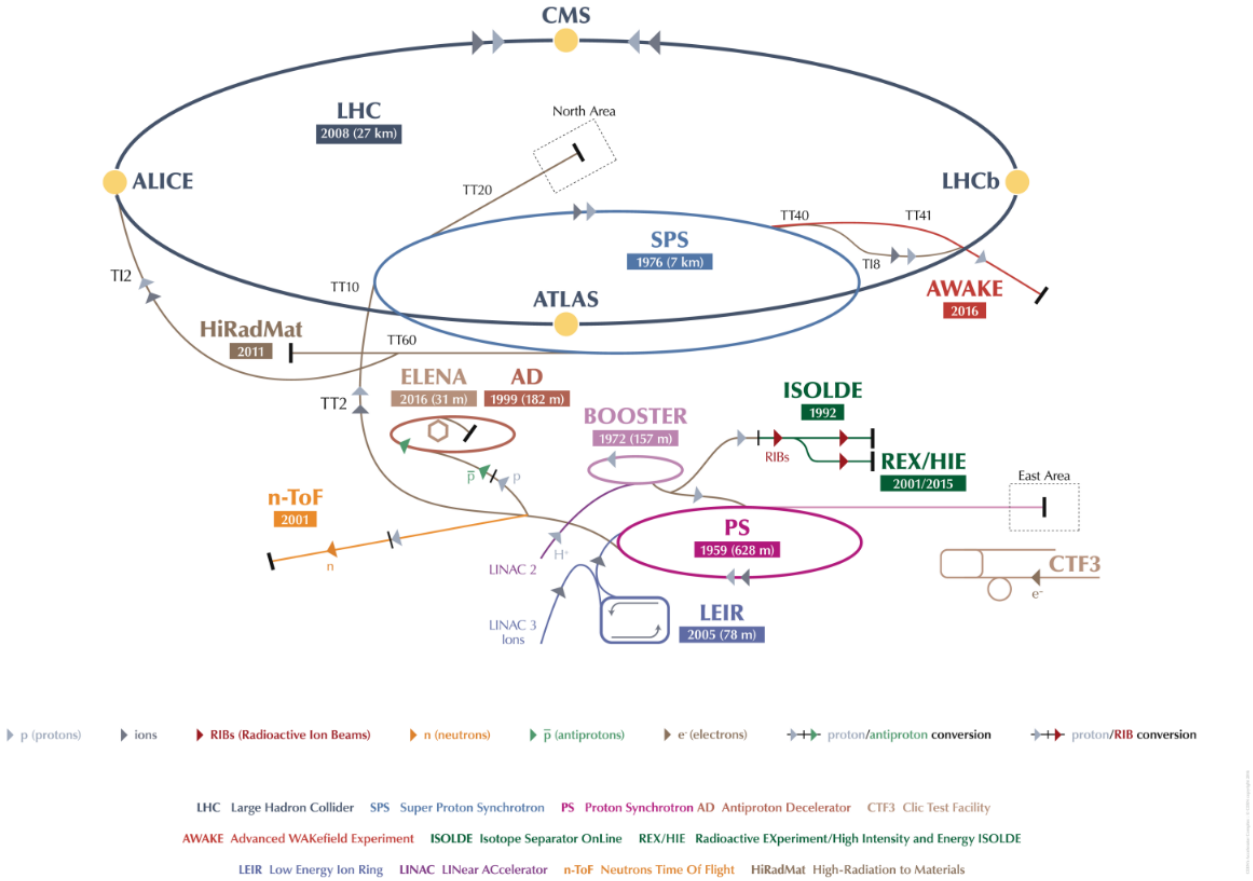


Figure 6: The LHC accelerator complex at CERN, including the various pre-accelerators. Shown are the linear accelerator (LINAC 2), the Proton Synchrotron Booster, the Proton Synchrotron (PS), Super Proton Synchrotron (SPS), and the Large Hadron Collider (LHC). The four main experiments on the LHC ring are also shown, although their relative positions on the ring are not to scale [32].

0.999999991 times the speed of light. The protons are brought to collision in four different locations, shown in yellow markers in Figure 6. These points represent the four main LHC experiments. ATLAS (A Toroidal LHC ApparatuS) [102] and CMS (the Compact Muon Solenoid) [34] are general-purpose detectors that investigate a wide range of physics. LHCb (Large Hadron Collider beauty) [35] is a dedicated experiment for precise measurements of CP violation and rare decays of b-hadrons. ALICE (A Large Ion Collider Experiment) [36] is dedicated to study quark-gluon plasma created in heavy ion collisions. The LHC has also three smaller experiments: TOTEM [37], LHCf [38] and MoEDAL [39]. TOTEM and LHCf search for forward particles, and they are close to ATLAS and CMS. MoEDAL, installed near LHCb, searches for magnetic monopoles and related exotic particles.

When the two bunches collide, the number of proton collisions that happen simultaneously is called in-time pile-up. These collisions produce hundreds of particles dispersing away from the interaction point.

The first LHC collisions started in 2009 with $\sqrt{s} = 900$ GeV. This was increased to 7 TeV in the following two years. In 2012, the energy was ramped up to $\sqrt{s} = 8$ TeV. This phase is identified as Run1. The second run of the LHC (Run2), ran from 2015 until 2018 with $\sqrt{s} = 13$ TeV and protons colliding with a bunch spacing of 25 ns (in Run1 it was 50 ns). Figure 7 shows the cumulative luminosity versus time delivered to ATLAS during stable beams for high-energy pp processes, shown for the various years through Run1 and Run2 data-taking. Larger luminosity is provided with a larger number of bunches using different strategies of bunch schemes. Figure 8 on the other hand, summarizes the schedule for LHC operations to from 2019 to 2036 where there is a cycle of periods of data taking and long shutdowns which allow maintenance and upgrade operations.

2.2 The ATLAS Detector

When the proton beams collide, ATLAS records the collision result in a range of 4π steradians as the interaction happens in the various detector components. Figure 9 shows a cut-away view of the ATLAS detector. Its cylindrical and symmetric shape around the beam line has a length of 44 m and a diameter of 25 m. Its total weight is 7000 tonnes. The collision point is located in the center of the detector. Therefore, the result of

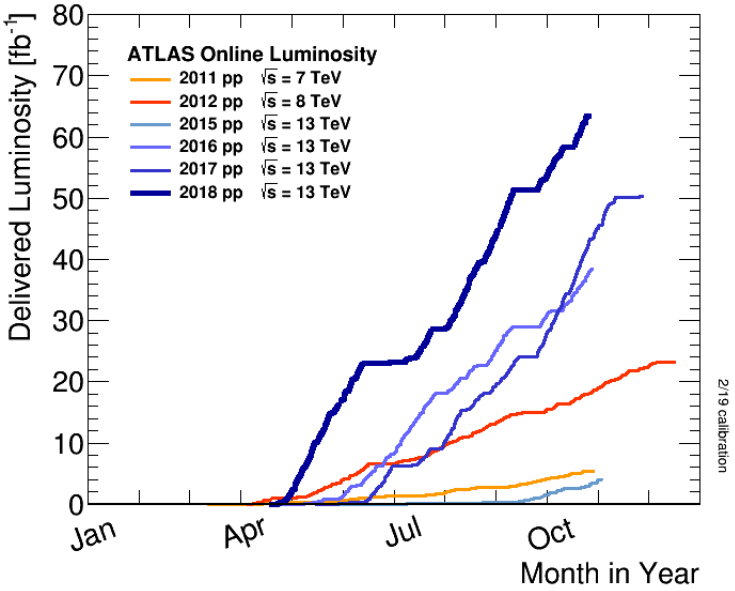


Figure 7: Cumulative luminosity versus day delivered to ATLAS during stable beams and for high energy *pp* collisions [43].

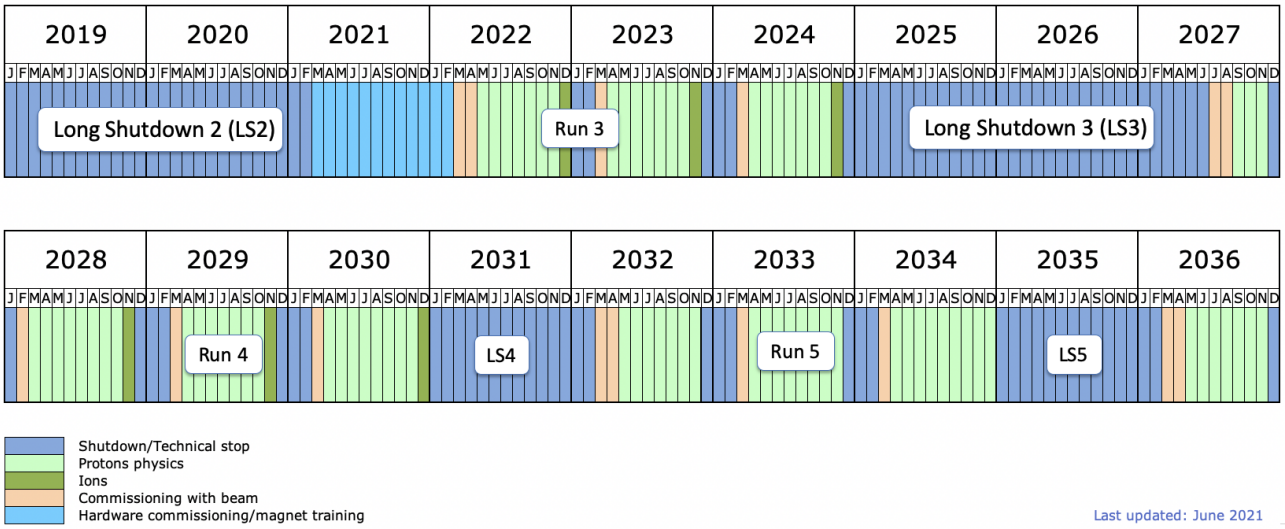
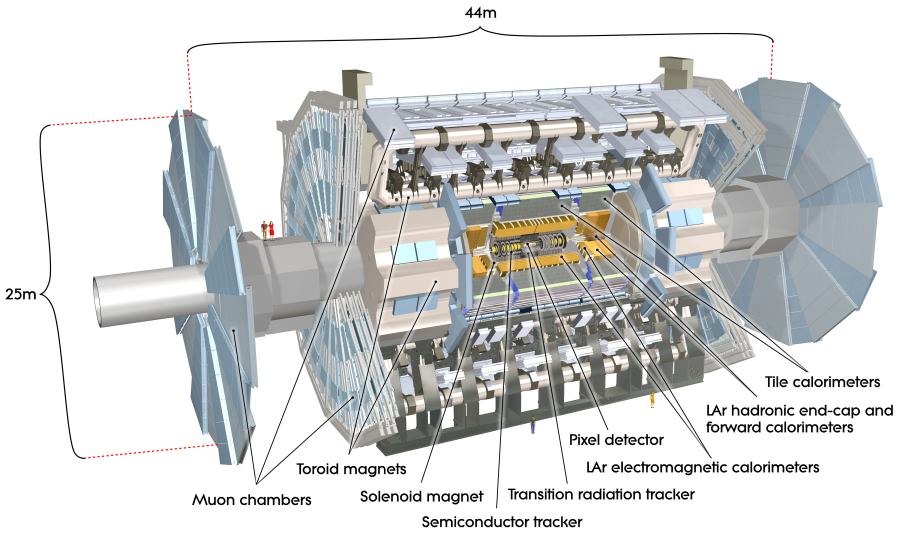
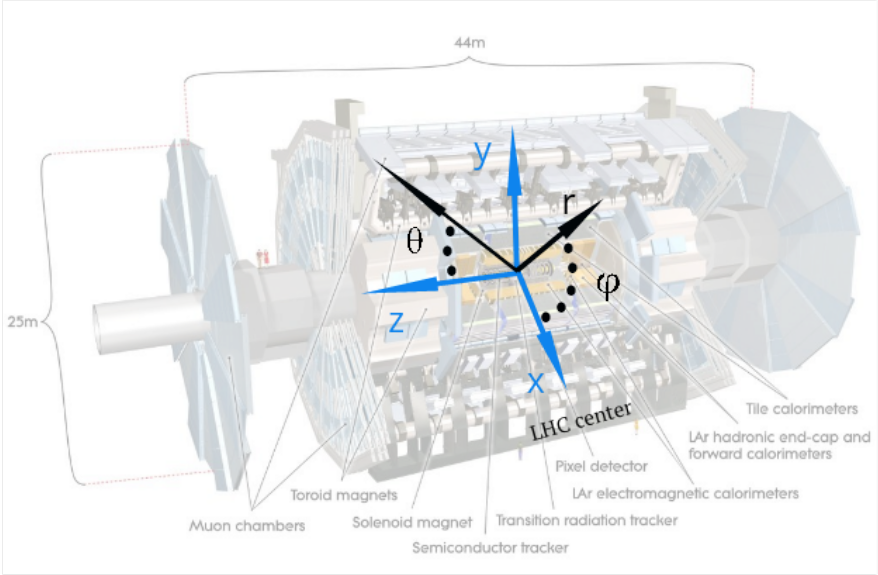


Figure 8: Schedule for LHC operations from 2019 to 2036.



(a)



(b)

Figure 9: Cut-away view of the ATLAS detector (a) and the ATLAS coordinate system (b). (x, y, z) form the right-handed Cartesian coordinate system and (r, ϕ, z) the cylindrical coordinates.

the collision will travel from the inside to the outside of the detector. This passage involves an interaction with the different sub-detectors, each with a particular measurement task. The innermost sub-detector is used to reconstruct trajectories of charged particles produced in collisions (Section 2.3). Particles then enter the calorimeter, where their energies are measured (detailed in Chapter 3). The muon spectrometer (Section 2.4) was designed to improve the identification of muon particles and precisely estimate their momentum.

As shown in Figure 9, the inner detector consists of the Pixel detector, the SCT tracker and the TRT tracker. A

solenoid magnet generates an almost uniform magnetic field with a strength of 2 T throughout all inner detector components. The calorimeter system consists of the liquid argon (LAr) and the scintillator-tile calorimeter systems. The muon system, together with the superconducting toroid magnets, forms the outermost part of the ATLAS detector [42]. Figure 9 also shows the magnet system of the ATLAS detector. This system provides an almost orthogonal field to a particle's trajectory. It consists of four large superconducting magnets, one central solenoid and three air-filled toroids. This hybridization allows an extension of the pseudo-rapidity coverage ($|\eta| < 3$). The toroidal part creates a magnetic field to bend muon particles in the muon spectrometer. It is composed of a central toroid and two end-cap toroids. Each toroid is an eight rectangular coils oriented in a radial direction from the beam axis.

Figure 9(b) illustrates the global coordinate system used in the ATLAS experiment. The interaction point, at the center of the detector, constitutes the origin of the ATLAS coordinate system [102]. For the right-handed Cartesian coordinate system (x, y, z), positive x-axis, y-axis and z-axis point to the LHC ring center, the surface and the LHC beam direction respectively. Alternatively, cylindrical coordinates (r, ϕ, z) can also be used. The azimuthal angle $\phi \in [-\pi, \pi)$ describes particle trajectories from the interaction point in the x-y plane. r and ϕ are defined as

$$r = \sqrt{x^2 + y^2} \quad ; \quad \phi = \frac{1}{tan\frac{x}{y}}.$$

Energy and transverse momentum properties allows us to characterize the rapidity of a particle [44]. Generally, experiments can only measure the angle of the detected particle with respect to the beam axis. Thus, the pseudorapidity η of a particle can be defined as the rapidity of an equivalent, but massless particle. η is defined in Equation 2 as a function of the polar coordinate $\theta \in [0, \pi)$, which is measured in the r-z-plane, representing the angle between the particle momentum and the beam axis.

$$\eta = -\ln\left[tan\left(\frac{\theta}{2}\right)\right]. \tag{2}$$

The momentum for colliding particles is unknown along the z-axis and therefore the momentum and energy are defined as the boost-invariant transverse component. Equation 3 describes them as a projection on the x-y plane.

$$p_T = \sqrt{p_x^2 + p_y^2} \quad ; \quad E_T = E \sin\theta. \tag{3}$$

Angular distances in $\eta - \phi$ space are generally measured in units of ΔR, where

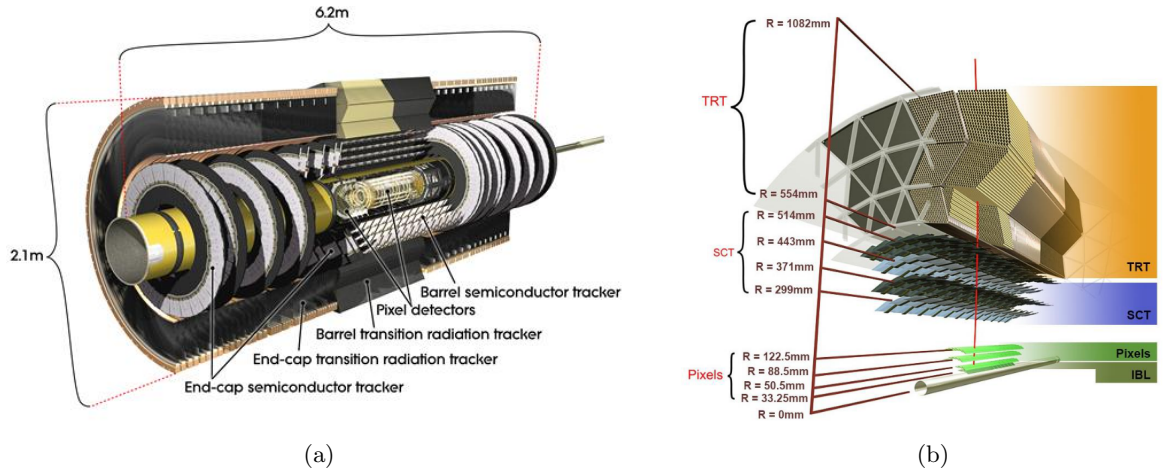
$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}.$$

2.3 The Inner Detector

Particles generated from the LHC collisions start their detection journey in the Inner Detector (ID). The ID is the innermost component of ATLAS, and therefore the closet to the collision point. The ID measures hits (charge deposition in a silicon sensor) where a particle passes. These hits serve to reconstruct the particles tracks, their momentum as well as the primary and secondary vertices corresponding to the interaction vertex and decay vertex respectively. The tracking reconstruction of charged particles takes place within the pseudo-rapidity range $|\eta| < 2.5$. The ID is embedded in a superconducting solenoidal magnet, creating a magnetic field of 2 T. This magnetic field bends the trajectory of charged particles in order to compute their momentum. The ID is composed of three sub-detectors each with a different material technology as shown in Figure 10: the Silicon Pixel Detector, the Semi Conductor Tracker and the Transistion Radiation Tracker.

The Silicon Pixel Detector is the innermost cylinder of four layers around the beam axis. It provides precise measurements of the hits left by charged particles. The pixel detector is made of silicon sensors that provide a high granularity and a good position resolution. Each layer of the pixel sub-detector is composed of 250 μm thick silicon modules. In each module, pixels provide information about the position and value of the free charge induced by particles traversing them. Figure 10 (b) shows the Insertable B-Layer (IBL) that was introduced during the Long Shutdown 1 to increase the precision of measurements of the tracker and to cope with the increased luminosity. The beam pipe was replaced with a smaller radius pipe to allow the insertion of the IBL layer.

The Semiconductor Tracker (SCT) differs from the pixel detector in having binary readouts and strips of sensitive silicon material. In total, there are four SCT layers. Each layer has two silicon strip layers in which



(a) (b)

Figure 10: Schematic representation of the ATLAS Inner Detector and its three sub-detectors [40]: (a) Cut-away view of the ATLAS Inner Detector, (b) representation of the sensors and structural elements of the ATLAS ID in the barrel.

readout signals are created from the passage of a charged particle. This provides an accurate measurement of track position of the particle at the crossing point of the two strips. The SCT contributes to the momentum, the resolution of the impact parameter and to the vertex position. Both the SCT and the pixel detectors are impacted by the high radiation due to their closeness to the beam axis, leading to performance degradation.

The Transition Radiation Tracker (TRT) contains about 351 000 straw drift tubes of 4 mm diameter filled with a gas mixture and an anode wire along the center. Particles traversing the TRT ionize the gas and the free charge drifts to the anode wire, allowing measurement of the signal and derivation of the passage radius of the particle. These measurements also contribute to computing particle momenta and in reconstructing the tracks at this detector. In the TRT, the number of activated sensors is larger than the pixel and SCT detectors, due to the fact that the TRT straws are larger.

2.4 The Muon Spectrometer

Muons are particles that lose only a small fraction of their energy when traversing the calorimeter. Their properties are therefore measured in the Muon Spectrometer (MS). The MS contains eight superconducting coils generating a toroid-shaped magnetic field to measure the charge over momentum ratio of a particle. The MS is a precise particle tracking system which combines this measurement with the one reconstructed in the ID. It also performs track reconstruction as a stand-alone detector, a key feature for event triggering. Figure 11 shows the ATLAS MS with its four subsystems.

Monitored Drift Tubes (MDT) of 29.97 mm diameter and a length between 1 and 6 m are filled with gas. MDT measure muons tracking information in the region $|\eta| < 2$ of the MS. MDT covers the largest area of the spectrometer. **Cathode Strip Chambers (CSC)** are radially oriented multi-wire detectors allowing tracking measurements in the MS region $2 < |\eta| < 2.7$. **Resistive Plate Chambers (RPC)** covering the $|\eta| < 1.05$ region of the MS, provide a fast output signal when a particle is traversing the MS. RPC are composed of two parallel plates with a 2 mm gap filled with gas mixture. The plates readouts use metallic strip technology. For reconstructing particles tracks, RPC measure orthogonal properties to the MDT. Note that, RPC are part of the trigger system. **Thin Gap Chambers (TGC)** are also part of the trigger system. TGC cover $1.05 < |\eta| < 2.7$ of the MS. They provide measurements to reconstruct the trajectory of muon particles with good time resolution and high rate capability.

2.5 The Trigger System

Every 25 ns one LHC pp collision happens inside the ATLAS detector, resulting in approximately 40 million pp events produced per second. Each event is a few megabytes in size and therefore the total amount of generated data exceeds the storage capacity. The Trigger and Data Acquisition (TDAQ) system allows us to select, in real time, the relevant events from the collisions. The relevance refers to having interesting physics objects for analysis investigation, such as high missing transverse energy in the calorimeter system. The ATLAS trigger system contains two different levels to reduce the event rate from 40 MHz to 1 kHz during LHC Run2. **The Level-1 trigger (L1)** is a hardware component. A subset of detector information used for L1 comes from the calorimeters (coarse granularity) and the MS. This information is used to accept or reject the event within

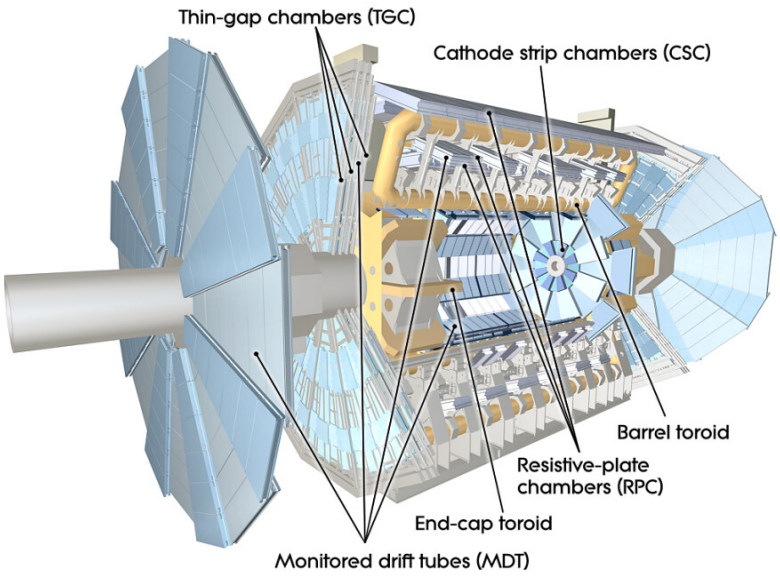


Figure 11: Schematic representation of The ATLAS muon system with the eight superconducting toroid magnets and its various sub-systems[40].

a latency of 2.5 μs. If the L1 accepts an event, the corresponding information from the detectors is stored in buffers. Information from the ID is not used due to the long readout time caused by the large number of channels. **High Level Trigger (HLT)** is the software component in which regions of interest are investigated and events are generally fully reconstructed using offline algorithms. The average processing time per event in the HLT is 0.2 s.

2.6 Towards High Luminosity LHC Upgrade

After the Higgs Boson discovery in 2012, the LHC research program focuses on widening our knowledge of particle physics with a list of upgrades such as moving from 8 to 13 TeV center of mass energy and an increase from 1380 to 2808 in the number of bunches per beam. Simultaneously, experiments like ATLAS also upgraded specific detector components, such as the ID in which the IBL layer was inserted. The High-Luminosity Large Hadron Collider (HL-LHC) upgrade aims to boost the LHC's discoveries potential starting from 2027. This upgrade concerns an increase in the luminosity by a factor of 10 beyond the designed LHC value. The HL-LHC will allow experiments to collect more data to study known physics processes in more detail, and also to observe rare events and new physics. Figure 12 shows the LHC upgrade program from 2011 to 2037 with the increase of the centre-of-mass energy and integrated luminosity. Currently, the ATLAS detector is undergoing the phase I experimental upgrade.

ATLAS Phase-I Upgrade [46] targeted the trigger system in the MS and calorimeters. The New Small Wheels are one of the most vital components currently being installed in the ATLAS detector [48] for more selective cuts of muon particles. They provide the required technology to cope with the HL-LHC in terms of high pile-up and high background rates.

ATLAS Phase II upgrade [49], constitutes the final addition towards the HL-LHC, during which tracking and trigger systems will undergo important modifications. The readout electronics will be replaced with a new Level0-Level1 (L0-L1) trigger with 200 kHz acceptance rate for L0 and an improved accuracy for L1. The tracker will also be replaced with a full silicon based architecture of pixel layers. The bandwidth of the tracker will be increased to provide timing information to the L1 trigger.

2.7 The Worldwide LHC Computing Grid

Measuring particles and interaction properties from known and new processes is based on physics analysis. Experiments run their analyses workflows on a computing infrastructure located around the world. This infrastructure is known as the Worldwide LHC Computing Grid (WLCG). It provides computing resources for storage, simulation, and analysis of data generated by the LHC. WLCG is built in a hierarchical structure of three levels. The levels vary according to the type of tasks to perform. **Tier-0** represents the core of the WLCG. A collision event in an LHC experiment Data Acquisition System will be sent to Tier 0. At this stage, a first

Figure 12: LHC upgrade program towards high luminosity (last update of the plan January 2021) [45] .

processing of the raw data is applied using experiment specific reconstruction algorithms. The reconstructed data is then distributed to the next level in the hierarchy. The CERN computing center is a Tier-0 site which stores over 200 petabytes of data on tape where on average 1 petabyte is processed every day [50]. **Tier-1** sites are generally large national computer centers. They perform tasks of reprocessing and centralized analysis. They represent the main provider of datasets from reconstructed detector data to simulation. The output of derivation tasks from these sites is copied over to the next level computing centers. **Tier-2** sites are dedicated to simulation jobs and physics analyses. Various Tier-2 facilities are hosted by institutes collaborating in different LHC experiments. These facilities are used by the institutes locally for data processing and storage operations.

3 ATLAS Calorimetry

Calorimetry, in nuclear and particle physics, is a detection method that measures properties of charged and neutral particles. The measurement takes place via energy absorption by the material of the detector [52]. As a result, the accurate modeling of the interaction between a particle and the detector material is a fundamental part of a simulation tool. This chapter details the calorimetry principles and the structure of the ATLAS calorimeter detector.

3.1 Calorimetry Principles

A calorimeter is a key component in experimental physics detectors to provide energy measurements and identification of photons, electrons, jets and inference of missing transverse energy (E_T^{miss}) [53]. These measurements are possible due to a showering processes as illustrated in Figure 13: when an incoming particle $p_{(t,e)}$ of type (t) and energy (e) hits the calorimeter surface, it creates a cascade of secondary particles called a shower. Particles produced in this shower deposit energy and produce further particles until the energy is completely absorbed in the calorimeter.

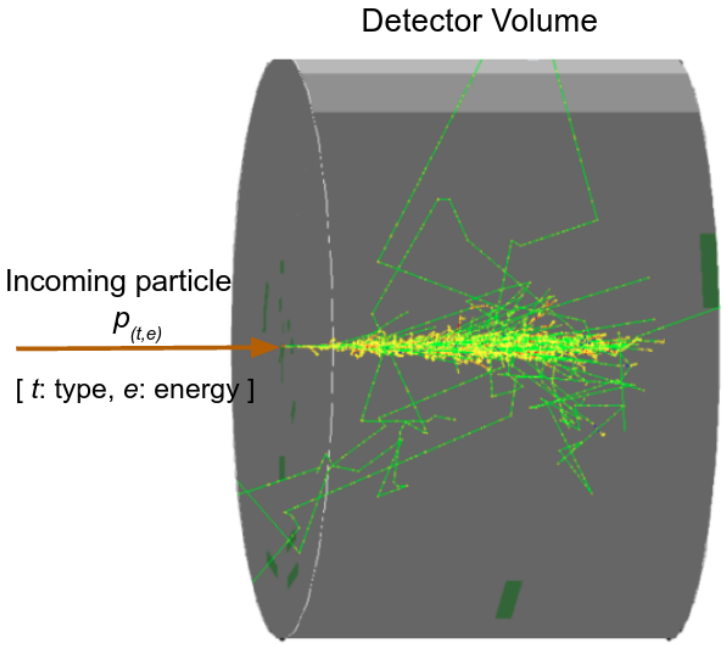


Figure 13: Example of a particle shower in a detector volume: an incoming particle of type t, energy e and a perpendicular direction to the calorimeter surface, creates a cascade of secondary particles in the detector volume.

In the absorption process, the energy is transferred as a mixture of heat[1], ionization, excitation of atoms, Čerenkov light and nuclear interactions. As a result, the choice of the material composing the calorimeter depends on the targeted effect. Two types of calorimeters exist: homogeneous and sampling. The material of a homogeneous calorimeter is at the same time the absorbing material and the detector. On the other hand, a sampling calorimeter is composed of alternating layers of a passive absorber of dense material used to reduce the energy of the incoming particle, and active detectors for signal generation. The choice of the absorber and detector material can vary according to the application.

The energy resolution of a calorimeter is defined as the ratio between the Gaussian width of the energy response and the value of the input energy. Maximizing the energy resolution and minimizing the particle shower leakage (particles escaping the calorimeter), are the main figures of merit to assess the quality of a designed calorimeter. The energy resolution is parametrized by the energy detector response (E), the noise (N) resulting from pile-up, readout electronics and the event-by-event fluctuations in the shower development (S) [57]. It is defined as,

$$\frac{\sigma(E)}{E} = \frac{\sigma(S)}{\sqrt{E}} \bigoplus \frac{N}{E} \bigoplus C, \tag{4}$$

[1]The origin of the term calorimetry comes from thermodynamics and calore in Italian means heat: referring to the energy converted into heat during the absorption process[52]

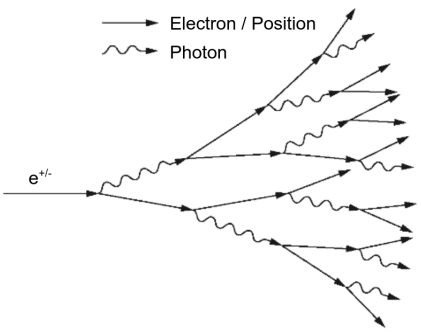


Figure 14: Schematic representation of an electromagnetic shower development of an $e^{\pm}$. [55]

where, $\bigoplus$ is a quadratic sum and C is a constant term to represent the systematic effects from inactive material and mis-calibration of the detector.

Given a statistical process, a particle shower would generate on average M secondary particles, where M is proportional to the energy of the incoming particle. Therefore, the energy resolution is dominated by statistical fluctuations of M. On the other hand, the shape of a shower and its composition depend only on the type and the energy of the particle. In the case of a photon, an electron or a positron, the resulting shower is "eletromagnetic". However, a hadron (pion or proton) being subject to strong nuclear interactions create a "hadronic" shower.

Electromagnetic Showers

Electromagnetic calorimeters (ECALs) measure the energy deposited by electromagnetic showers developed from light electroweak particles such as photons, electrons, and positrons. An example is illustrated in Figure 14 where an incoming electron or positron emits a photon through Bremsstrahlung, which in turn produces electron and positron pairs. The material components of ECALs allows us to derive the radiation length X_0, which represents the average distance travelled by an electron before its energy E is reduced by a factor of $1/e$.

Frequently, at lower energies, material ionization or atomic/molecular excitations may happen, resulting in a scintillation signal upon de-excitation. The average energy loss per unit distance $< dE/dx >$, known as ionization density, is well modeled by the Bethe formula [52]. Minimum ionizing particles (MIPs), are defined as unit charge particles with an energy equal to the position of the minimum in the $< dE/dx >$ curve. Low energy photons dissipate energy through the photoelectric effect and Compton scattering. Coulomb and Rayleigh scattering also contribute to changing the trajectory of incoming particles. For particles traveling faster than the local speed of light, Čerenkov light is also emitted. The radiative process is the major cause of energy loss for high energy electrons and positrons with mass m_e. These interactions produce an exponentially developing shower of secondary particles with respect to the direction of the initial particle, with low energies. On the other hand, highly energetic particles, via a multiplication mechanism based on pair production and the Bremsstrahlung effect, produce more secondary particles and on average penetrate deeper in the calorimeter. Thus, as a function of the calorimeter depth, the initial energy deposited per unit length $< dE/dx >$ increases with the multiplication process to reach the maximum $< dE/dx >$ of the shower. At that point, most shower particles will have low "sufficient" energy and the loss of the energy would be related to non multiplying processes.

Hadronic Showers

Hadronic calorimeters (HCALs) are usually sampling calorimeters that measure the energy of charged and neutral hadrons, such as pions and protons. The shower development is similar to ECALs. The difference lies in the strong hadronic interactions that lead to more complex showers, i.e., with a variety of development processes. Hadronic showers are expected to reach larger longitudinal and lateral displacements than electromagnetic ones. Figure 15 illustrates the different hadronic interactions. In the first cascade (intra-nuclear), the components of the nucleus get sufficient energy to interact with each other and produce hadrons (such as pions). The second cascade (inter-nuclear) shows the particles escaping the nucleus and hitting another nucleus. Due to ionization, the shower particles lose their energy and the end of a shower is reached when the energy is insufficient for interactions with the nuclei.

Similarly to electromagnetic showers, hadronic showers start with a multiplicative process leading to an increase in the number of secondary particles up to a maximum limit, upon which an absorption process starts. A number of particles within a hadronic shower are likely to decay electromagnetically. As an example, a high

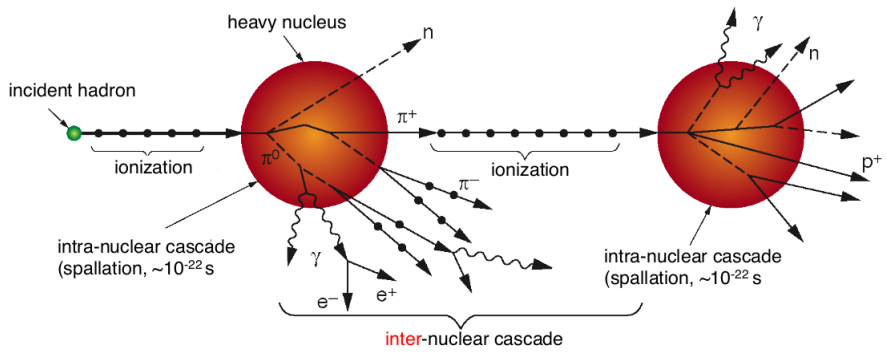


Figure 15: Schematic representation of hadronic interactions [55].

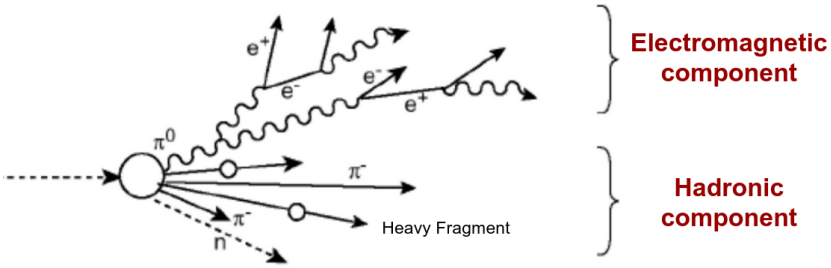


Figure 16: Schematic representation of a neutral meson decay [55].

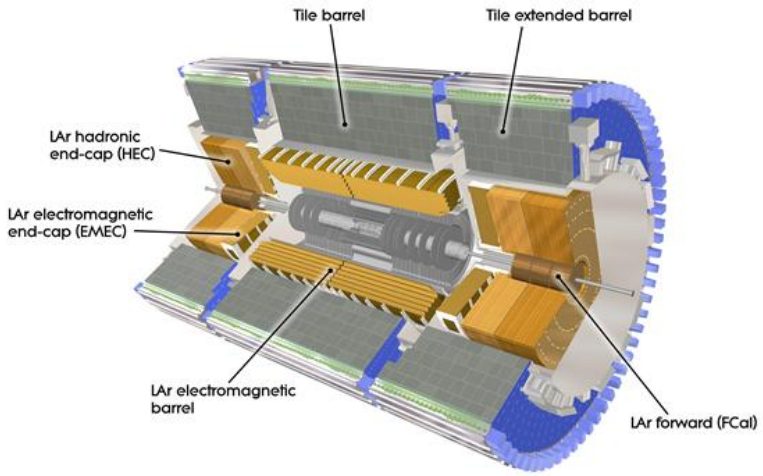


Figure 17: Cut-away view of the ATLAS calorimeter [63].

energy hadron, such as a neutral meson, decays into photons and therefore initiates an electromagnetic shower within the hadronic shower. This process is shown in Figure 16. The electromagnetic fraction includes π^0, representing approximately a third of the particles produced during the first interaction in the cascade of Figure 16.

3.2 The ATLAS Calorimeter

The ATLAS calorimeter is composed of electromagnetic and hadronic calorimeters, with full ϕ symmetry and coverage around the beam axis up to $|\eta|$ <5 [63]. Figure 17 shows a representation of the ATLAS calorimeter where three main parts are identified: barrel, end-cap and forward. The barrel part forms the cylindrical shape of the detector: LAr eletromagnetic barrel and Tile hadronic barrel. The electromagnetic end-cap calorimeter (EMEC) and the hadronic end-cap calorimeter (HEC) define the base of the cylinder. The forward region (FCAL) surrounds the closest region to the beam pipe.

The ATLAS calorimeter was designed to provide at least 25 electromagnetic radiation lengths up to $|\eta|$ <3.2 and

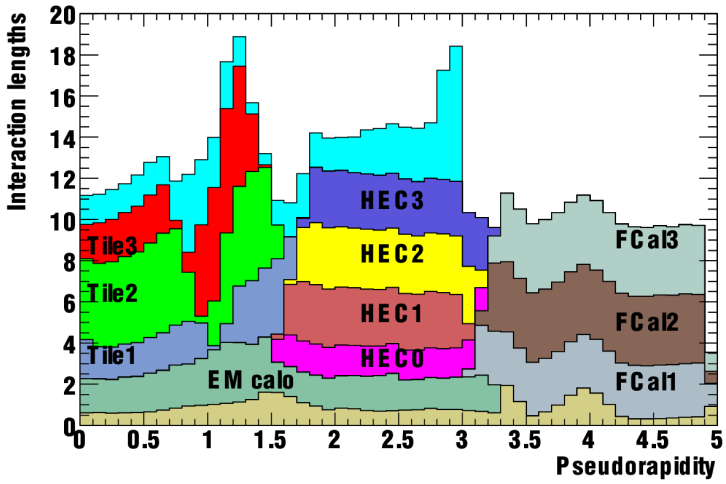


Figure 18: Cumulative amount of material in units of hadronic interaction length λ vs the pseudorapidity $|\eta|$, in front of the ATLAS ECAL (first region preceding the EM calo), in the ECALs, HCALs and the total amount at the end of the active calorimeter. The total amount of material in front of the first active layer of the muon spectrometer (up to $|\eta| < 3.0$) is also shown in cyan [53].

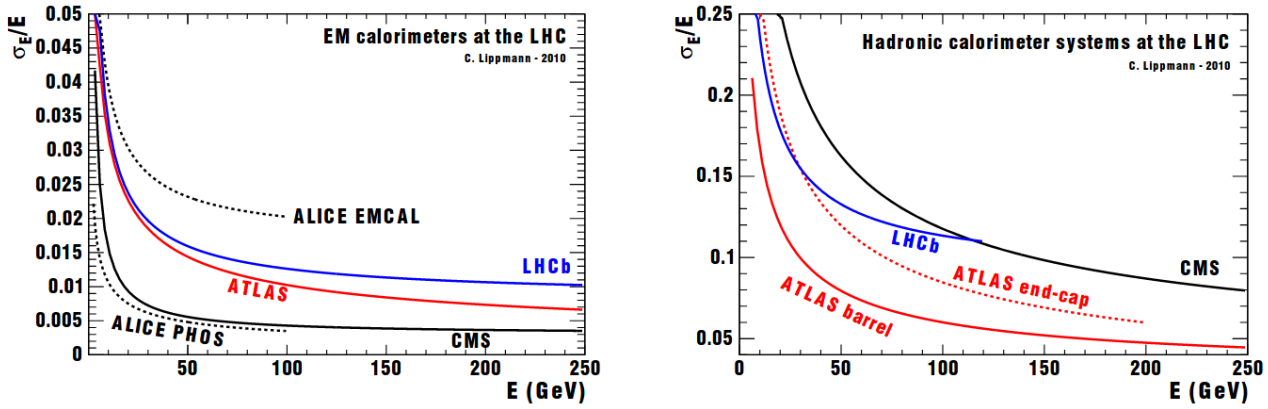


Figure 19: Comparison of the relative energy resolutions: eletromagnetic calorimeters (left) and hadronic calorimeters (right) at the LHC experiments [60].

at least 10 hadronic interaction lengths up to $|\eta|$ <4.9. Figure 18 illustrates the profile of the absorption length as function of the pseudorapidity for electromagnetic and hadronic calorimeters. Furthermore, the ATLAS calorimeter was designed to achieve an energy resolution (Equation 4) of 10% and 50 % for S, 0.1% and 1% for N using C values of 0.7% and 3% for the electromagnetic and hadronic calorimeters respectively [58, 59]. The resolution curves of the ATLAS electromagnetic and hadronic calorimeter compared to other LHC experiments are reported in Figure 19.

3.2.1 The ATLAS Electromagnetic Calorimeter

The ATLAS ECAL [53] is segmented longitudinally, with a sampling structure of alternating active liquid argon (LAr) and passive material of steel and lead. It is divided into a barrel (LAr electromagnetic barrel EMB) and an endcap (LAr electromagnetic end-cap EMEC) regions. The former covers a range in pseudorapidity of $|\eta| <$ 1.475, while for the later covers the range of $1.375 < |\eta| < 3.2$. The ATLAS ECAL has an accordion geometry shape in the ϕ direction as shown in Figure 20. The design of the accordion shape provides a more precise electromagnetic measurement by optimizing the cable length and the dead region compared to the standard parallel-plate geometry [64] in addition to optimizing the time of signal readout and transfer.

The EMB region in the ATLAS ECAL is composed of two identical half-barrels with a gap of few millimeters [53] and an accordion shape in which the width of each accordion increases with the depth to maintain a constant angular distance $\Delta\phi$. Each half-barrel is equipped with readout boards containing cells pointing in the η and ϕ directions.

In the ATLAS ECAL design, the first layer is referred to as a presampler layer. It is used to correct for the energy lost in front of the calorimeter. This loss is due to the material of the detector with the distribution of

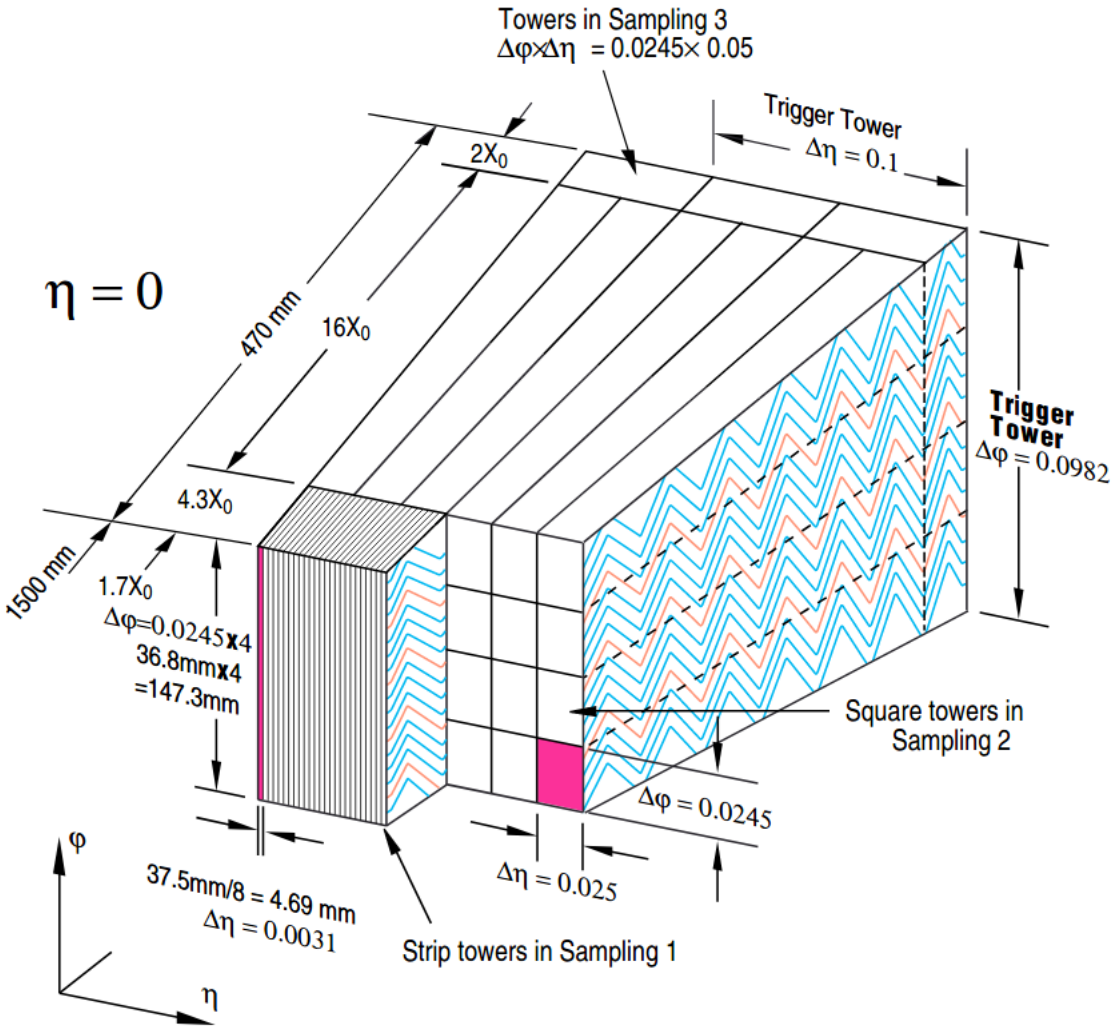


Figure 20: Schematic representation of a cut-away view of the ATLAS electromagnetic barrel calorimeter at $\eta = 0$ [63]. The trigger towers, analog signal sums of super-cell cluster (grouping of detector cells) are shown, and three layers are represented with their cells granularities in $\Delta\eta \times \Delta\phi$.

Coverage	$0< \|\eta\| <1.4$	$1.4< \|\eta\| <1.8$	$1.8< \|\eta\| <2$	$2< \|\eta\| <2.5$	$2.5< \|\eta\| <3.2$
Presampler	0.025×0.1	0.025×0.1			
Layer1	0.003×0.1	0.003×0.1	0.004×0.1	0.006×0.1	0.1×0.1
Layer2	0.025×0.025	0.025×0.025	0.025×0.025	0.025×0.025	0.1×0.1
Layer3	0.050×0.025	0.050×0.025	0.050×0.025	0.050×0.025	

Table 1: Granularity in $\Delta\eta \times \Delta\phi$ of the ATLAS EMCAL sampling layers [53].

dead material, such as the solenoid magnet in the front of the ECAL. The barrel presampler and the endcap presampler have, respectively, 1 cm and 5 mm of liquid argon active layer with electrodes perpendicular and parallel to the beam axis. A critical region for the presampler remains the transition between the EMB and the EMEC. A scintillator layer is mainly used for jet energy measurement correction. Beyond $\eta > 1.8$, the amount of dead material is limited and the presampler layer is no longer necessary. The next three consecutive electromagnetic layers EMB1, EMB2, EMB3 for barrel region and EME1, EME2 and EME3 in the endcaps are aligned along the longitudinal direction. Each layer is designed with a different granularity in the η and ϕ directions. The sampling layer 1 is finely granular, segmented into 4 mm wide strips to provide fine resolution in η. This fine resolution allows a sensitive shower shape measurement and is used, for example, for photon and electron ECAL shower identification. Moreover, it allows for π^0 rejection for $E_T \geq 50$ GeV. Sampling layer 2, is segmented into cells of square shape of 0.025 in $\Delta\eta \times \Delta\phi$ for $|\eta| < 2.5$ and $0.1\times$ 0.1 beyond η of 2.5. The third layer is the last layer in the ECAL [53]. The different granularities of the ECAL layers are summarized in Table 1.

The depth of the different ECAL layers was optimized based on the π^0 rejection criterion. The sampling layer 1 is 6 X_0 deep. The second layer has a depth of is 24 X_0 and the last layer depth varies between 2 and 12 X_0. Additionally, for $\eta <0.6$, the depth of the second sampling layer is 22 X_0 and therefore 2 X_0 would be the depth of the next layer. Designing the calorimeter was based on a detailed simulation of its response, mainly using high energetic particles to define total radiation thickness up to the end of the calorimeter. For the barrel region, this value must be at least 24 X_0 and 26 X_0 in the end-caps [53].

3.2.2 The ATLAS Hadronic Calorimeter

The ATLAS HCAL [65, 67] represents the outermost part of the ATLAS calorimeter. It is a sampling calorimeter with alternating design of active (plastic scintillating tiles) and passive (steel) materials. The former material allows the conversion of the ionization signal into photons, and the latter for enhancing the showering process. The thickness of the HCAL layers is a function of the nuclear interaction length $\lambda = 20.7$ cm. The HCAL is segmented into a tile (TileCAL), an endcap (HEC) and a forward (FCAL) regions. The TileCAL, the central segment of the ATLAS HCAL, is designed with three consecutive parts along the beam line: a tile/central barrel region (with $|\eta| < 1$ coverage) between two extended barrel regions ($0.8 < |\eta| < 1.7$). The TileCal represents a key component in the jet and missing-energy measurements, jet sub-structure, electron isolation, and triggering [62]. The tile barrel region is segmented into three layers, as shown in Figure 21. A, BC and D represent the cells in the layers. The granularity of the cells in $\Delta\eta \times \Delta\phi$ in the tile calorimeter is 0.1×0.1 for the two first layers of both regions (TileBar0, TileBar1, TileEx0, TileEx1 containing the cells A, BC) and 0.2×0.1 for the last layer (TileBar2, TileEx2 containing the D cells).

The HEC ($1.5 < |\eta| < 3.2$) is composed of two independent wheels. The first wheel has two longitudinal segments of 8 and 16 layers in depth, while the second has one segment of 16 layers in depth. The granularity remains the same for both wheels, with 0.1×0.1 for a pseudorapidity up to $|\eta|= 2.5$ and 0.2×0.2 when the pseudorapidity is up to $|\eta|= 3.1$. Figure 22, shows the HEC calorimeter using projective lines of η for illustration purposes.

The FCAL ($3.1 < |\eta| < 4.9$), is a LAr calorimeter, relatively fine segmented with cooper as an absorber material in the first layer and tungsten for the two next layers. The FCAL, located at approximately 4.7 m from the interaction point, is designed to cover the very forward region of the ATLAS detector. This design is adapted to manage dense conditions such as radiation damage or high level noise coming from high rate of pile-up events. The FCAL contains an electromagnetic module (FCAL1) and two hadronic modules (FCAL2 and FCCAL3). The corresponding granularities in $\Delta x \times \Delta y$ in centimeters are summarized in Table 2.

Table 3 summarizes all the region of the ATLAS calorimeter with their η coverage, the layer ID and the sampling module names.

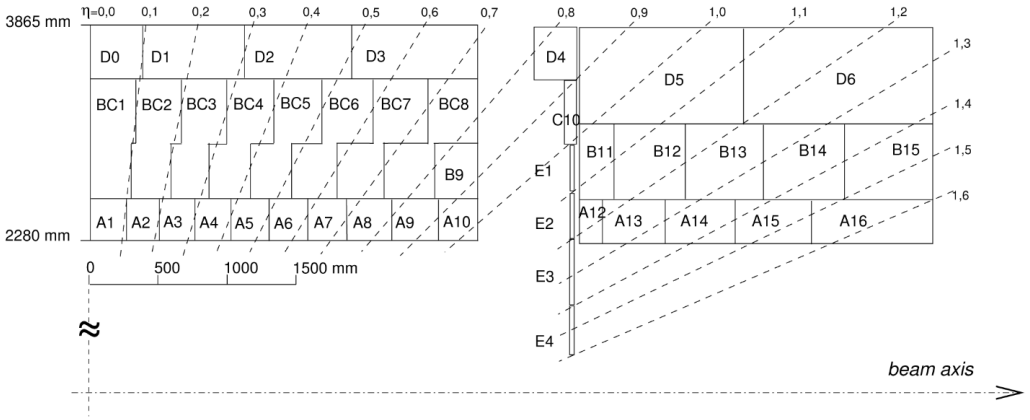


Figure 21: Tile cells in depth and η of the tile calorimeter in the central region (left) and extended region (right), showing tower structure. (E1, E2) represent the "gap" and (E3,E4) the "crack" scintillators [62].

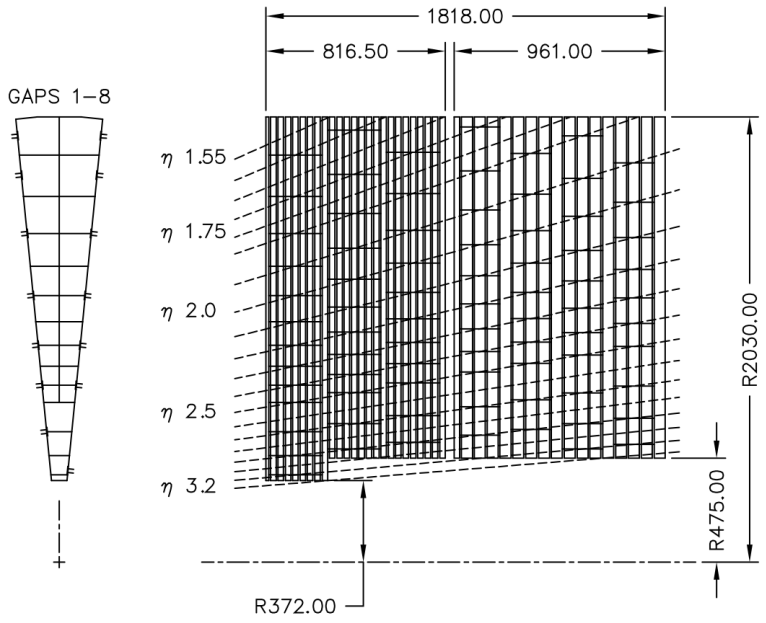


Figure 22: Schematic representation of the ATLAS hadronic end-cap calorimeter: R-ϕ view (left) and R-z view (right) with dimensions in mm. The dashed lines indicate the semi-pointing layout of the readout electrodes [53].

FCAL layer	Coverage	Granularity (cm)
FCal1	$3.15< \|\eta\| <4.3$	3×2.6
	$3.1< \|\eta\| <3.15$, $4.3< \|\eta\| <4.9$	About 4 times finer
FCal2	$3.24< \|\eta\| <4.50$	3.3×4.2
	$3.2< \|\eta\| <3.24$, $4.5< \|\eta\| <4.9$	About 4 times finer
FCal3	$3.52< \|\eta\| <4.6$	5.4×4.7
	$3.29< \|\eta\| <3.32$, $4.6< \|\eta\| <4.9$	About 4 times finer

Table 2: Granularity in $\Delta x \times \Delta y$ of the ATLAS FCAL sampling layers [53].

	η-coverage	Layer ID	Sampling module
LAr barrel	$0< \|\eta\| <1.5$	0	PresamplerB
		1	EMB1
		2	EMB2
		3	EMB3
LAr EM endcap	$1.5< \|\eta\| <3.2$	4	PreSamplerE
		5	EME1
		6	EME2
		7	EME3
Hadronic endcap	$1.5< \|\eta\| <3.2$	8	HEC0
		9	HEC1
		10	HEC2
		11	HEC3
Tile barrel	$0< \|\eta\| <1$	12	TileBar0
		13	TileBar1
		14	TileBar2
Tile gap	$1< \|\eta\| <1.6$	15	TileGap1
		16	TileGap2
		17	TileGap3
Tile extended barrel	$0.8< \|\eta\| <1.7$	18	TileExt0
		19	TileExt1
		20	TileExt2
Forward endcap	$3.1< \|\eta\| <4.9$	21	FCal0
		22	FCal1
		23	FCal2

Table 3: ATLAS calorimeter regions, layers, η coverage and sampling module names.

4 Physics Objects

One of the key tasks of a detector is to identify the type of the physics objects created during a collision event. Some of these objects leave distinct and unique traces in the detector. These traces are then used in object identification algorithms.

This chapter summarizes the main physics objects found in the ATLAS detector.

4.1 Hits to Object Reconstruction

The nature of the physics processes is determined by the type and energy of a particle interacting with the detector volume. In order to characterize this process, ATLAS uses reconstruction and identification techniques on the recorded events from collisions. The reconstruction allows us to process the recorded signals by converting the readouts into physics objects to extract their corresponding features. This reconstruction is usually a multi-level application: signals are converted into raw detector hits, which are then converted into intermediate objects and then into features. Identifying particles is a classification process that uses these reconstructed features. To assess the performance of both reconstruction and identification ATLAS uses, amongst others, the energy resolution, the selection efficiency and its false positive rate as figures of merit.

Reconstruction techniques are applied in the different ATLAS detector layers. At first, in the tracking step, the reconstruction targets charged particle tracks that are built from the hit positions in the ATLAS inner detector volume. Then, the reconstruction of vertices is performed using previously obtained tracking information. At the calorimeter level, the reconstruction uses the deposited energy. The ATLAS calorimeter segmentation into cells is used for energy deposition measurement, where the measured signal is converted into energy using a known calibration (from a test beam for example). Neighboring cells with the same incoming particle trace (or signature) are clustered together using a "sliding-window" clustering algorithm [68] based on a fixed-size rectangles in the grouping process. Another clustering method, known as the "topological" algorithm, groups neighboring cells where the signal is significant compared to noise. The algorithm consists of grouping spatially close energies with a high probability of belonging to the same shower. These calorimeter cells with a close-by topology are grouped into clusters referred to as topo-clusters [71]. Technically, the cell signal significance variable ζ_{cell}^{EM} is computed in the cluster formation process. ζ_{cell}^{EM} is defined as $E_{cell}^{EM}/\sigma_{noise,cell}^{EM}$, where the nominator is the cell signal and the denominator is the expected average noise in this cell. Both quantities are measured in the EM energy scale, which correctly reconstructs the deposited energy from photons and electrons. The grouped clusters may expand across the calorimeter layers, if the neighboring cells are defined as cells with at least partial overlap in $\eta \times \phi$. In the clusters, we might find negative energies in the cells, coming principally from noise and residual radiation fluctuations. The complexity of hadronic showers may lead to having sub-showers far from the primary particle where most of the energy is deposited, therefore the shower spreads over multiple topo-clusters.

4.2 Physics objects

The interaction of the different particles with the detector creates traces or signatures that are used in object identification algorithms. Figure 23 illustrates a cross-section view of the ATLAS detector with signatures of different particles traversing the detector.

Photons

Photons are neutral particles identified by their electromagnetic shower energy deposition signature in the calorimeter. This signature is mainly derived using the information of their shower shape properties. Depending on the η region, 20 to 50% of the photons become an electron-positron pair in the inner detector [72]. This results in a vertex with at least one track in the direction of the calorimeter cluster. The rest of the photons are only reconstructed from the energy of the clusters within the ECAL. The particle candidates form clusters with vertices and tracks pointing to the calorimeter. Photon identification is based on rectangular cuts and tuned on Monte Carlo identification techniques using shower shape variables describing the electromagnetic showers as listed in Table 4. These variables describe lateral and longitudinal shower development in the electromagnetic region and the leakage fraction in the hadronic part [73]. Photons produced promptly in a collision are likely to have narrow energy deposition in the ECAL and smaller leakage to the HCAL compared to photons from jets (background candidates) [73]. Two types of selection cuts, "loose" and "tight", are defined for the identification process in the Run2 data (2015 and 2016). The loose selection relies on the shower shapes in EMB2 and the energy deposition in the HCAL. The tight selection additionally improves sensitive shower shape measurements due to the fine segmentation in η in the strip layer.

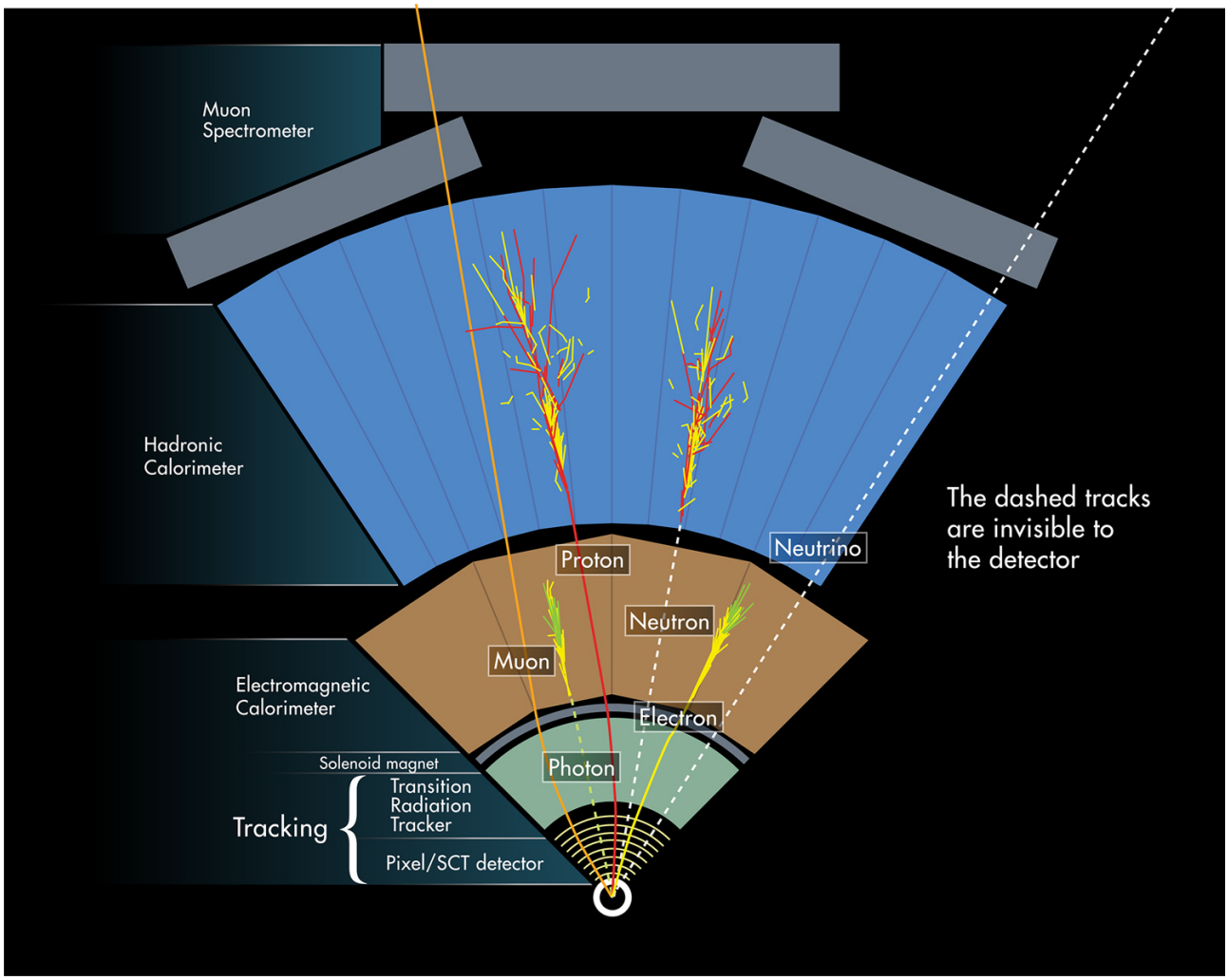


Figure 23: A diagram representing the ATLAS detector with particle signatures [70].

Category	Name	Description	Loose	Tight
Acceptance		$\|\eta\|$ <2.37, with 1.37<$\|\eta\|$ <1.52 excluded	✓	✓
ECAL middle layer	R_η	Ratio of the energy in 3×7 $\eta \times \phi$ cells over the energy in 7×7 cells centered around the photon cluster position	✓	✓
	$w_{\eta 2}$	Lateral shower width: $\sqrt{((\sum E_i \eta_i^2)/(\sum E_i) - ((\sum E_i \eta_i)/(\sum E_i))^2)}$, where E_i is the energy, η_i is the pseudorapidity of the cell i, the sum is calculated within a window of 3×5 cells	✓	✓
	R_ϕ	Ratio of the energy in 3×3 $\eta \times \phi$ cells over the energy in 3×7 cells centered around the photon cluster position		✓
ECAL strip layer	w_{s3}	Lateral shower width: $\sqrt{((\sum E_i (i - i_{max})^2)/(\sum E_i))}$, i runs over all strips in a window of 3×2 $\eta \times \phi$, i_{max} is the index of the highest-energy strip calculated from three strips around the strip with maximum energy deposit		✓
	w_{stot}	Total lateral shower width: $\sqrt{((\sum E_i (i - i_{max})^2)/(\sum E_i))}$, i runs over all strips in a window of 20×2 $\eta \times \phi$, i_{max} is the index of the highest-energy strip measured in the strip layer		✓
	f_{side}	Energy outside the core of the three central strips but within seven strips divided by energy within the three central strips		✓
	ΔE_s	Difference between the energy associated with the second maximum in the strip layer and the energy reconstructed in the strip with the minimum value found between the first and second maxima		✓
	E_{ratio}	Ratio of the energy difference between the maximum energy deposit and the energy deposit in the secondary maximum in the cluster to the sum of these energies		✓
	f_1	Ratio of the energy in the first layer to the total energy of the EM cluster		✓
Hadronic leakage	R_{had1}	Ratio of E_T in the first sampling layer of the hadronic calorimeter to E_T of the EM cluster (used over the range $\|\eta\|$ <0.8 or $\|\eta\|$ >1.52)	✓	✓
	R_{had}	Ratio of E_T in the hadronic calorimeter to E_T of the EM cluster (used over the range 0.8 < $\|\eta\|$ <1.37)	✓	✓

Table 4: Discriminating variables used for loose and tight photon identification [73].

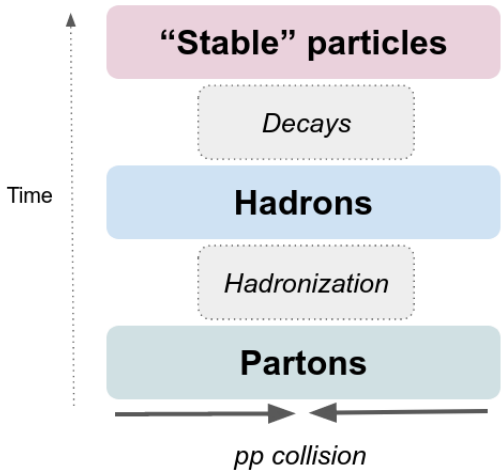


Figure 24: A diagram representing a jet evolution over time.

Electrons

Two properties are available for the detection of an electron signature: the charged track and the electromagnetic shower. Track reconstruction is crucial to obtain the resolution of track parameters. These parameters are used to match the corresponding tracks to the ECAL clusters through an extrapolation to the central layer of the ECAL. Additionally, electron-like signatures can also be identified in the HCAL when measuring the tails of the shower distribution. The shower shape variables such as the lateral width of the shower or the energy fraction per layer are discriminating features for electron identification. ATLAS electron identification in Run1 and Run2 relied on an online cut based selection technique and an offline likelihood identification [75].

Jets

Gluons and quarks, referred to as "partons", are not directly observed as individual particles in the detector. Under the strong interaction, partons are subject to fragmentation and hadronisation resulting in additional radiated partons and final state hadrons. The set of charged and neutral hadrons form a spray of particles called a "jet". The partons interact mostly via parton scattering, creating a cascade of showers of particles. The algorithm of energy measurement from hadrons and their decays defines the nature of the jet. Jets represent a key element in the data analysis of the LHC experiments to improve our understanding of hard-scattering processes and to determine the properties of the originating quarks and gluons.

The "particle level" jets, also called "stable" particles in Figure 24, can be observed in a HEP detector as : photons, electrons, protons, neutrons, pions, muons or other particles with a longer lifetime. Therefore, the properties of a jet depend on these particles. A jet algorithm is used to derive track measurements at the inner detector level or energy cluster measurements at the calorimeter level [76]. The "particle flow" approach, for example, combines both measurements for particle identification by matching tracks with calorimeter clusters. Jet reconstruction is then applied to an ensemble of particle flow objects consisting of the calorimeter energy and tracks which are matched to jets [59].

A charged hadron can ionize the calorimeter material and under the strong interaction with an atomic nucleus (as shown previously in Figure 15) is likely to produce additional hadrons propagating in the calorimeter, interacting and producing several hadrons until their energy is too small for new strong interactions. Many of these hadrons are likely to be pions. In the ECAL and HCAL, jets are reconstructed from a connected cell topology of energy depositions. The ratio between EM energy deposits through π^0 and hadronic energy deposits is a major contributor to the jet energy resolution.

Muons

Muons are key components in the electro-weak processes. They deposit very little energy in the calorimeter and if their energy is sufficient they are detected in the muon spectrometer. Reconstruction and identification of muon candidates is based on combined information from the inner detector and the muon spectrometer. Due to the spectrometer gap around $|\eta| = 0$, measurements from the calorimeter are here used instead.

5 ATLAS Detector Simulation

Building software that predicts the physics processes happening at the detector level is known as "simulation". Detector simulation allows us to improve the quality of physics measurements and detector design. It enables the study of physics models by matching theoretical predictions to experimental measurements. Figure 25 shows the agreement between simulation and real data for a recent and first observation of two W bosons produced from the scattering of two photons [78].

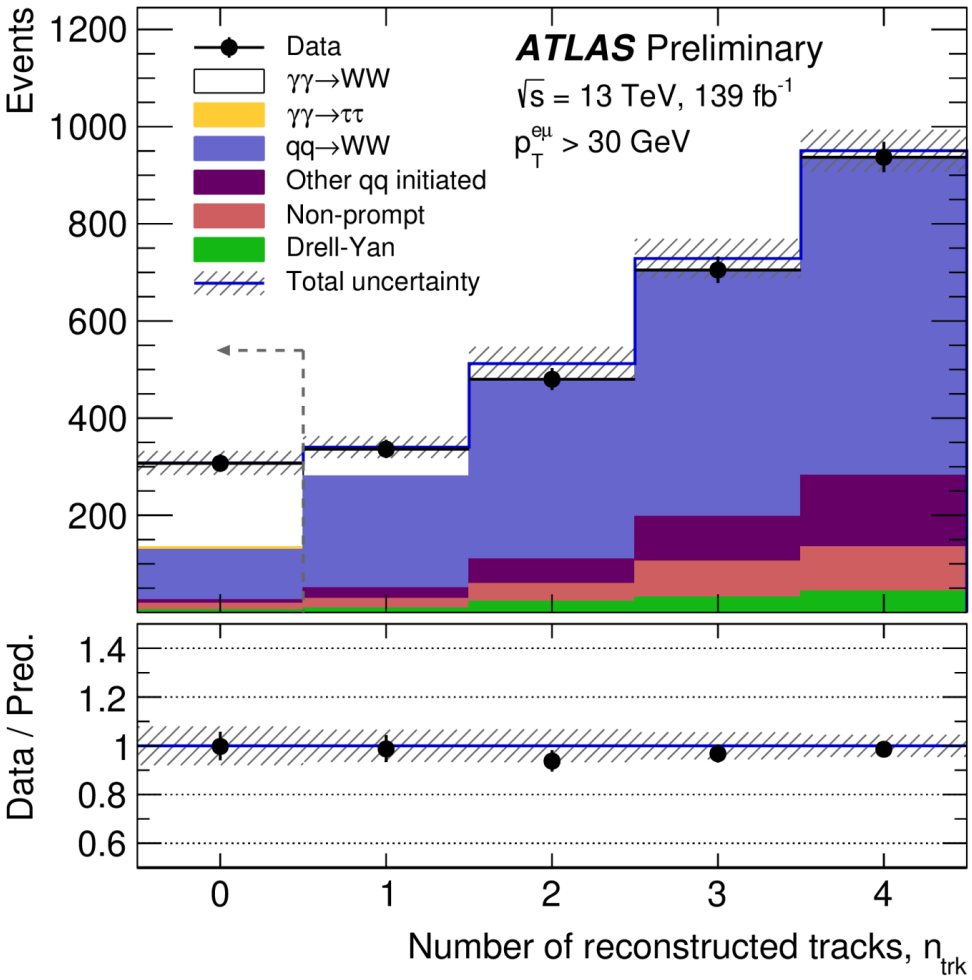


Figure 25: Number of reconstructed tracks distribution. The blue distribution represents the simulation of the W-boson-pair production from proton constituents, and the black points are the observed data. The white area represents the accumulation at low particle multiplicities of the photon-induced W-boson pair production [78].

At the ATLAS experiment, the simulations of particle showers produced in the calorimeters (detailed in Chapter 3) are performed with an advanced software framework known as Athena. Athena uses the Geant4 software that is based on Monte Carlo (MC) methods to accurately simulate the physics processes as they occur in the detector. This modeling process is commonly referred to as "full simulation" or "particle tracking simulation". Since the full simulation models every aspect of the interaction, it is inherently slow and a faster version of it has been developed under the name "Fast Simulation". In the calorimeter subdetector, for example, the current Fast Simulation relies on parametrization in both longitudinal and lateral directions of energy depositions to model the outcome of te detailled particle interaction.

5.1 Geant4 and Athena

In order to understand and label physics processes such as those shown in Figure 25, it is necessary to accurately simulate the detector response. This response is simulated iteratively, i.e., the physics at a small scale helps understand the physics at a larger scale. Each one of these physics scale iteration processes is stochastic and intractable. Monte Carlo (MC) methods are powerful tools to model an intractable probability distribution

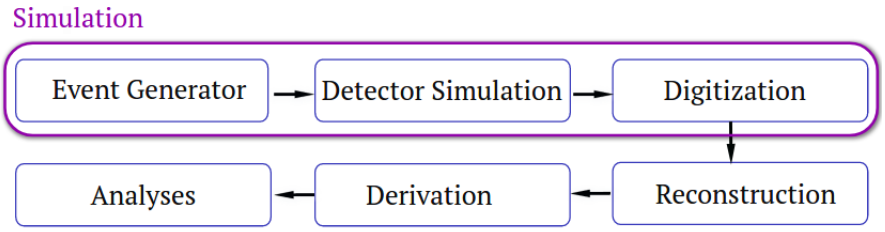


Figure 26: The ATLAS simulation data flow. The blue boxes represent the different algorithms. The simulation is a set of three algorithms: generate primary particles from LHC-like collisions, simulate the detector response when these particles interact with the detector, and finally compute the detector response as digits. The reconstruction transforms the detector response into physics objects. The derivation consists of filtering relevant data from the reconstruction output. This filtering creates outputs for physics analysis

based on random sampling. As a consequence, these techniques are relied on in many frameworks to simulate physics processes.

Geant4 is a simulation toolkit [83, 84] written in C++. It simulates particle interaction with matter using MC techniques. It is used by large scale experiments and projects such as nuclear medicine, astrophysics, and high energy physics. NASA, for example, uses Geant4 for radiation dose estimation for astronauts and electronic components. In high energy physics, it provides a toolkit to model HEP experiment detectors. Geant4 was designed and implemented by an international collaboration of experts in MC simulation of physics detectors and processes. Geant4 components include, among others, geometry and tracking descriptions, detector response modeling, event management and a user interface. The geometry feature covers all the physics aspects of an experiment, including the detector material. The tracking component allows the modelling of the particle trajectory through matter, its potential interactions and decays.

5.2 ATLAS Simulation Chain

Figure 26 summarizes the ATLAS simulation-reconstruction chain, a collection of interdependent components to produce relevant data for physics analysis [82]. The simulation steps consist of: event generation, detector simulation and digitization.

Event generation allows us to define the primary particle content of a collision according to the Standard Model. The output is a very large number of quasi-stable particles such as muons, pions, electrons and photons and an even larger amount of unstable particles such as mesons.

Pythia [93] and Sherpa [94] are the most used event generators in ATLAS [86]. PythiaB [99] refers to the ATLAS version of Pythia for B-physics event generation to produce $b\bar{b}$ pairs. The Pythia implementation is based on hard and soft scattering processes in a single event. We can find two models of QCD radiation in Pythia: showering and hadronization models. The first model is implemented to match the QCD shower description from a theoretical point of view. The second model, allows us to combine quarks and gluons into hadrons at the end of the showering model. Figure 27 represents a hadron-collider generated event. The final state is produced and develops by including QCD Bremsstrahlung effects in both initial and final state, hadronization and unstable hadrons decays into stable ones.

Event recording is an intermediate step that allows us to store the full event produced by the generator. It builds a connected tree of the event history of the decay chain and retains information about interacting partons and the secondaries from the interactions in the detector. The next step is the simulation, which models the response of the detector. It is detailed in Section 5.3.

At the end of the simulation workflow, digitization allows us to convert the simulated hits into digits that represent the detector response. These digits translate the voltage on a readout channel within the predefined conditions of threshold cuts and timing. Moreover, specific digitization algorithms can model the properties of each subdetector such as electronic noise, cross-talk between neighboring cells and dead cells.

Simulating a realistic detector response includes all the pp interactions from a bunch crossing, the inelastic interactions from hard scattering, beam gas effects and the response to long-lived particles.

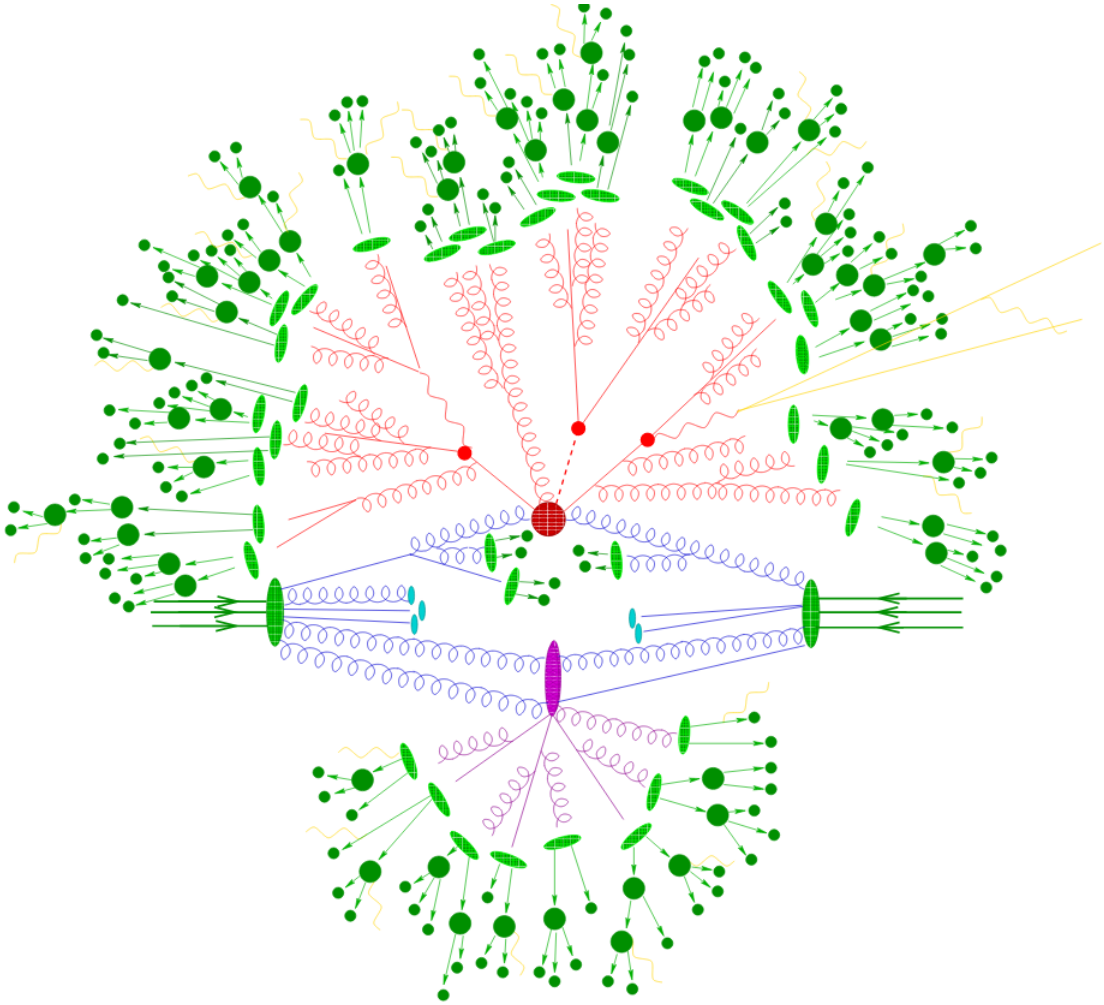


Figure 27: Representation of a $t\bar{t}h$ event produced by an event generator. The large red sphere represents a hard interaction, and the smaller ones represent the decay of top quarks and the Higgs boson. Additional produced hard QCD radiation are shown in red and a secondary interaction in purple happens before the final-state partons hadronize in light and dark green for hadrons decay. The yellow segments illustrate the photon radiation happening at multiple stages [100].

5.3 Full and Fast Detector Simulation

The full simulation based on the Geant4 toolkit is the most detailed description of the detector response. However, it is heavily resource consuming. Fast techniques are being developed by the ATLAS collaboration, such as Fast ATLAS Tracking Simulation (FATRAS) to simulate the inner tracker and the muon spectrometer response and the Fast Calorimeter Simulation (FastCaloSim or FCS) to emulate the calorimeter response. Combinations of both full and fast simulations have been the motivation for developing the ATLAS Integrated Simulation Framework (ISF). They are known as ATLFASTII or AF2 which combines Geant4 and FastCaloSim and ATLFASTIIF which combines FATRAS and FastCaloSim.

5.3.1 Full Simulation

The first stage of the full simulation pipeline consists of constructing a detector geometry to define the detector volume. This construction describes every detector component and its composing material. The ATLAS detector description consists over five million volumes. Moreover, about 400 different materials are used such as argon, aerogel and Kapton cable. These descriptions are technical translations of how all the ATLAS components are set up together. The full detector simulation then models the passage of particles while interacting with every detector component. For each of the particles, this simulation includes the physics models to describe processes such as ionization, decays and nuclear interactions. In practice, simulating all these processes at the highest fidelity is not feasible due to the tremendous amount of time required. Instead, any effects that have a small impact on the physics and the detector are turned off. In addition, specific Geant4 parameters are set, such as the parameters to control the creation of secondary photons or electrons during ionization and Bremsstrahlung processes. This detector response modeling includes as well a detailed map of the magnetic field used to bend charged particles and effects of detector misalignment.

Photons and pions are key physics objects that require an accurate simulation of their secondaries. The full simulation of energy deposition of electromagnetic and hadronic showers in the calorimeter is the slowest part in the simulation because it requires the modeling of interactions of all the secondary particles with matter at the microscopic level.

Although processes can be fully simulated, only the relevant ones are stored for further analysis. During the simulation, a large number of secondary tracks is produced and the definition of relevance, in this case, relies on a strategy at the level of the inner tracker and the calorimeter detectors. For example, the strategy at the inner detector level concerns the storage of ionization vertices if the primary particle has an energy above 500 MeV and the energy of the generated electron is above 100 MeV. At the calorimeter level, muon bremsstrahlung vertices are stored if they have energies above 1 GeV and 500 MeV for the primary muon and the generated photon, respectively. The parts of the detector that are of relevance are known as sensitive detectors in which the simulation creates a snapshot of the physical interactions known as hits. These hits are uniquely labeled with an ID. In the tracker, a hit for every single step of every single track is stored. It contains information on the position, time, deposited energy and the corresponding track identifier. At the calorimeter level, a hit is created for every cell. The calorimeter hit contains the cell ID and its accumulated energy deposition. Geant4 stores also the truth information which describes what actually happened in the simulation.

As a consequence, the detailed and step-by-step modeling at distance scales as small as 10^{-20} m using the Geant4 toolkit is CPU and memory intensive. Moreover, the full simulation depends entirely on the detector design, which means if parts of the detector are upgraded, the simulation has to adapt. Many directions are explored in order to reduce the CPU time and disk space of the full simulation. One of the optimization targets is the physics list by reducing or removing the number of generated neutrons in a hard scattering event after the primary interaction. Another optimization is the Dynamic linking of large address spaces libraries instead of using static Geant4 to speed up the simulation process [87].

5.3.2 Fast Simulation

The FastCaloSim or FCS (and also referred to as FCS V2) simulation is designed to provide fast simulation of the ATLAS calorimeter based on two parametrizations of the longitudinal and lateral energy profile of eletromagnetic and hadronic showers. The longitudinal component models the energy propagation from the interaction point through each layer of the calorimeter. The lateral component parametrizes the shower shape within each layer. These parametrizations replace the slowest part of the full simulation of a shower development in the calorimeter. Figure 28 details the different components of the fast simulation as well as their interactions.

The FCS parametrizations are derived using the Geant4 simulations of photons, electrons and charged pions. As described in Chapter 3, the ATLAS calorimeter has a complex structure which changes along η. To simplify the modeling of this structure, FCS uses a detector segmentation of the full η range, forming 100 equidistant bins of size 0.05. Thus, this segmentation provides a uniform structure. The energies of the particles considered

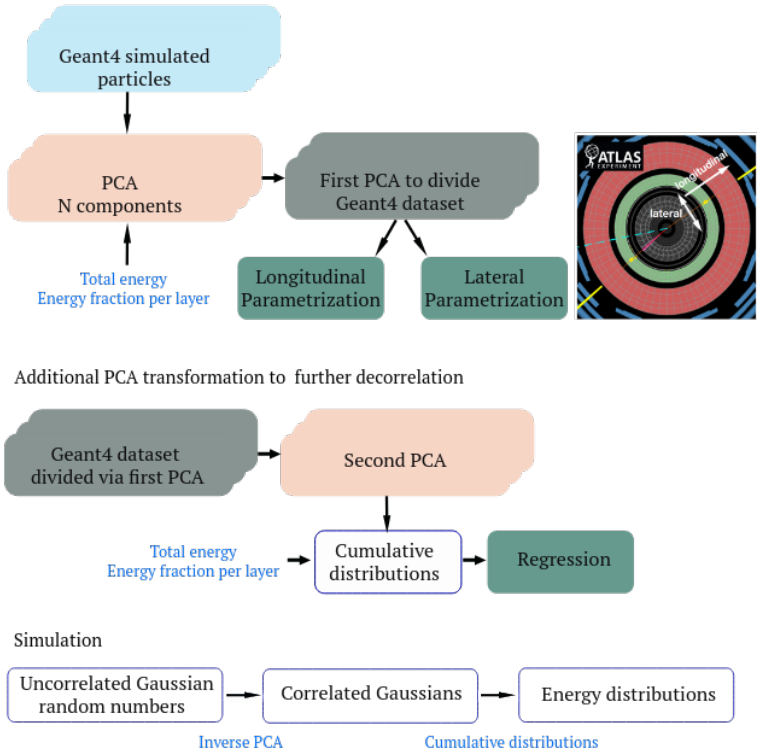


Figure 28: FCS baseline to derive the longitudinal and lateral parametrizations using PCA decomposition. Two PCA steps are applied. All the inputs for each component and the set of operations are shown in blue. The Simulation step using FCS is also illustrated.

range from 1 GeV to 4 TeV [104]. Incident particles, known as truth particles, are simulated starting from the calorimeter surface only (ignoring prior interactions)[104]. Around 10k events are generated for each sample without the beam spread in the interaction region.

The FCS approach uses the reconstructed geometry of the calorimeter, segmented into cuboids in η, ϕ in the barrel and x, y, z in the forward region [103]. This geometry is referred to as the "simplified ATLAS geometry". Geant4 version 10.1.3 with MC16 Run2 simplified ATLAS geometry is used to simulate the detector response of the truth particles [104]. The electronic noise, cross talk between neighboring cells and dead cells are turned off in the digitization step. FCS parametrizes only the "relevant layers" of a calorimeter shower. A layer is "relevant" when the average energy fraction with respect to the total energy is above 0.1 % per shower in this layer.

Longitudinal parametrization

The longitudinal energy paramaterization describes the deposited energy in each layer of the calorimeter. These energies across layers are strongly correlated, making the process of energy response modeling difficult. FCS uses a chain of principal component analyses (PCA) [106] to decorrelate these energies. The PCA allows a conversion of these correlations into a set of linearly uncorrelated energies using an orthogonal transformation of the coordinate system. The input parameters to the PCA are the total energy of the event over all layers and the fractions of energy deposition per layer. The first step in the PCA chain transforms the energy inputs into Gaussian distributions using the inverse error function and the cumulative distributions. These Gaussian distributions allow us to build the PCA matrix by projecting them onto the eigenvectors of the corresponding covariance matrix. The result of the PCA matrix represents the decorrelated features. The new coordinate system is defined by the normalized eigenvectors to unity. The first vector is defined in a way that the data shows the largest variance if mapped to the new axis. The second vector is orthogonal to the first one, with the second-largest variance in the data. The next vector v is orthogonal to the $v-1$ axes. This first transformation step is referred to as "first PCA". Figure 29 shows the correlation after Gaussian transformation. The correlation factors are calculated from the 2D histograms where 0 means no correlation, 1 for maximal correlation and -1 for maximal anti-correlation. In this figure, we can observe important correlations.

A PCA rotation is applied to get the leading principal component (PC) containing the largest variance in the input data. The showers are divided into quantiles referred to as "PCA bins" where each bin carries an equal distribution of events. Figure 30 shows that correlation factors are almost zero between the output components

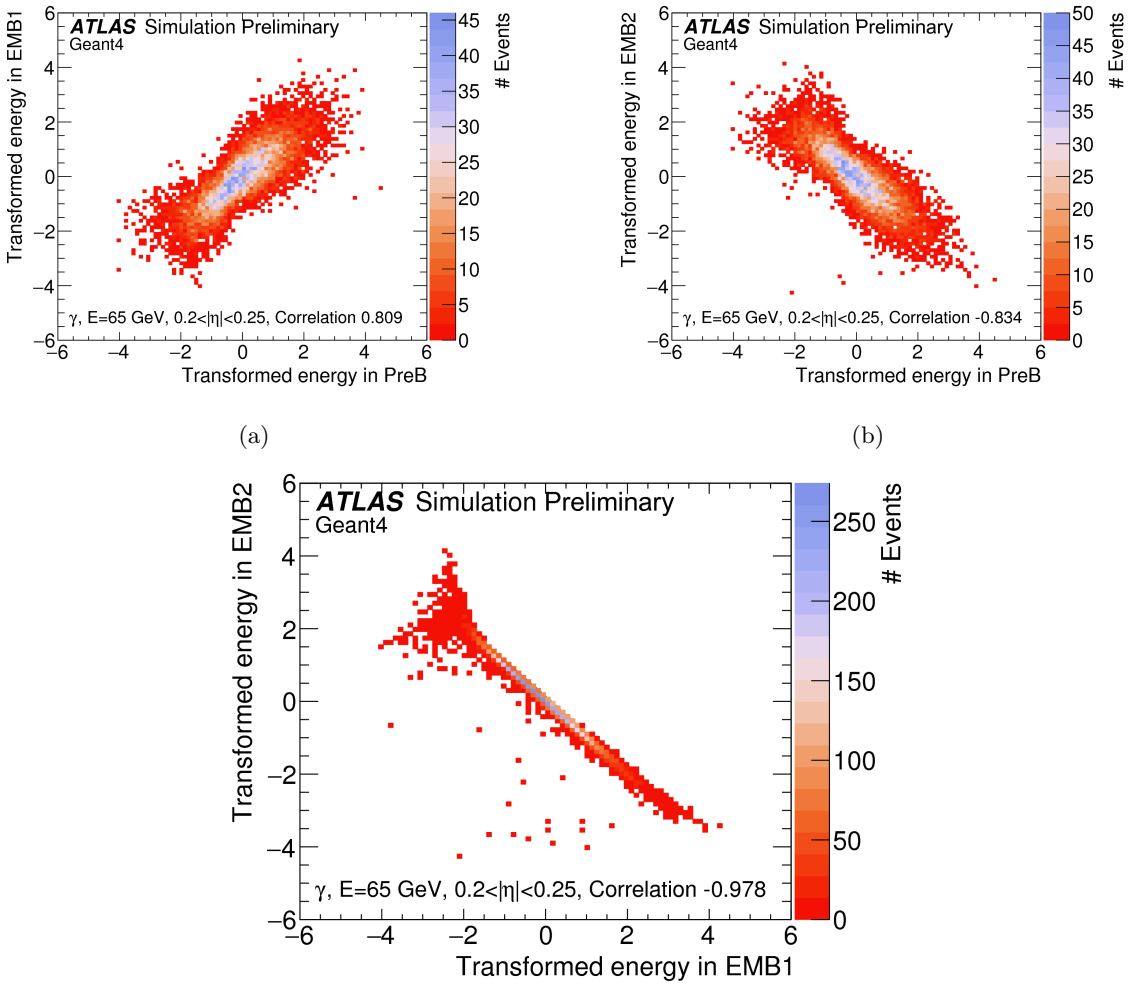
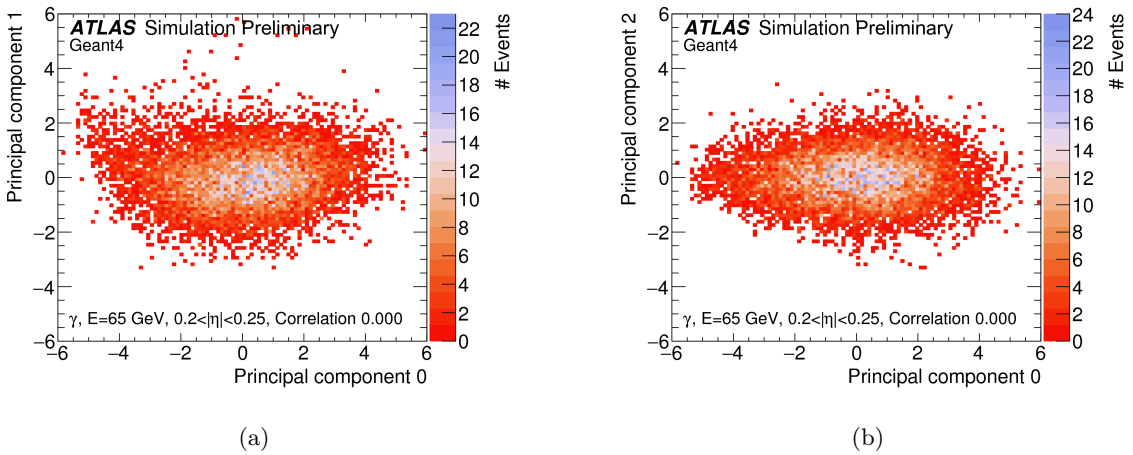
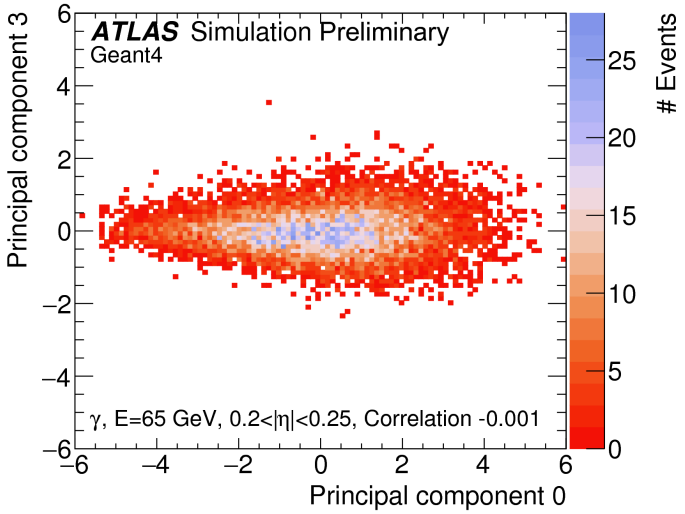


Figure 29: Correlations between the transformed energies (before PCA rotation) of 65 GeV energy photons in $0.2 < |\eta| < 0.25$. (a) pre-sampler barrel vs. EM barrel 1, (b) pre-sampler vs. EM barrel 2 and (c) EM barrel 1 vs. EM barrel 2. The energies were transformed into Gaussian distributions and the correlation factors from these 2D histograms are displayed 0.809, -0.834 and -0.978 from left to right [104].





(c)

Figure 30: Correlations between the components after PCA rotation. The plots show that individual components are approximately Gaussian distributed. After PCA transformation, the correlation factors are almost 0 [104].

of the first PCA after rotation.

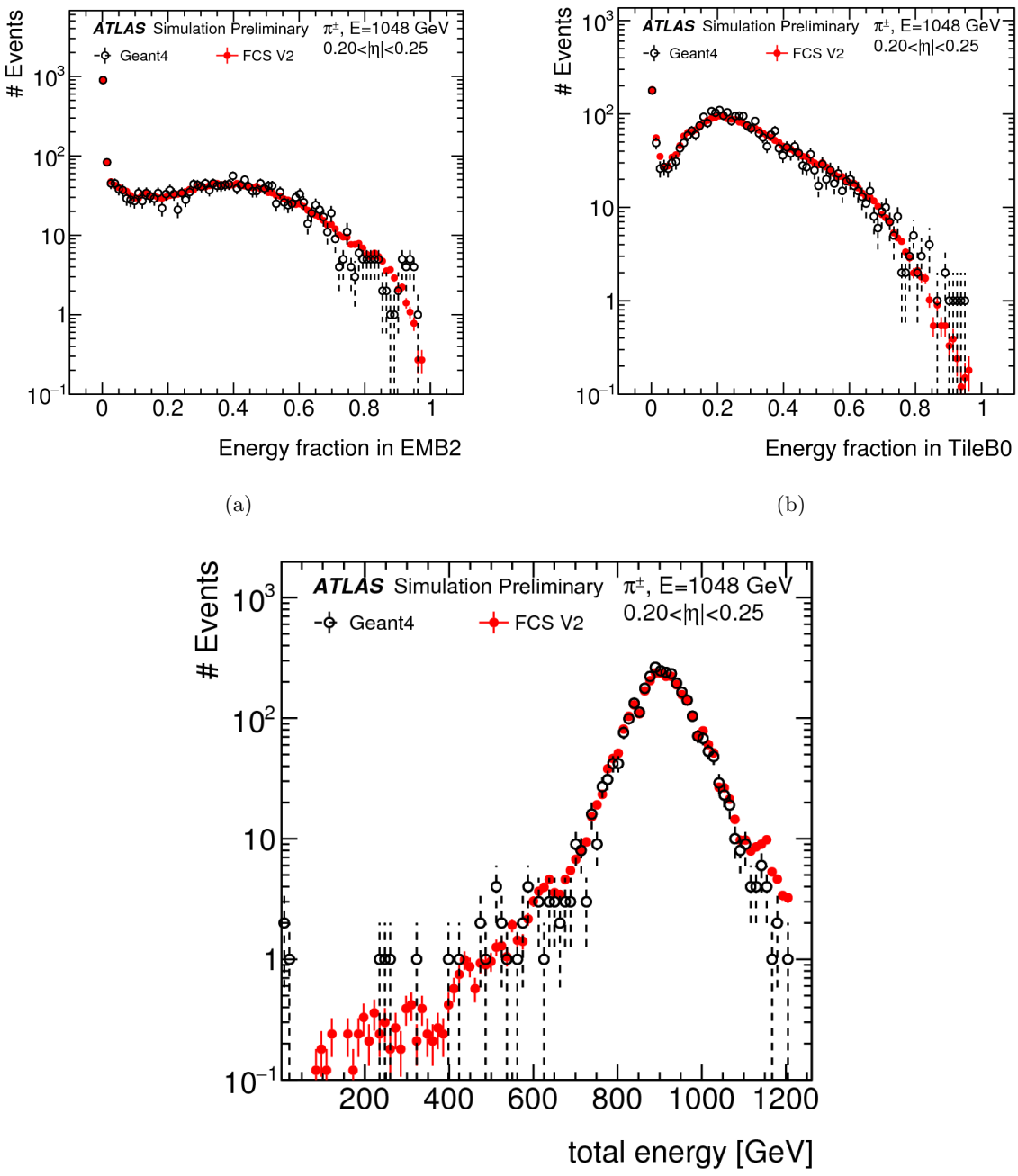
To refine the energy decorrelation, the second step in the PCA chain consists of applying a PCA on each bin called "second PCA". At the end of the process, the parametrization file will contain the cumulative energy distributions, the PCA matrices from the second PCA transformation, the mean, and RMS of the Gaussian distributions after the PCA rotation [104].

Figure 31 presents the longitudinal energy parametrization for pions of 1 TeV energy in $0.2 < |\eta| < 0.25$ for an energy in the electromagnetic region and hadronic region along with the total energy.

Lateral parametrization

A basic parametrization of lateral energy distribution is a symmetric radial function around the impact point of a particle in one layer of the calorimeter. Therefore, the shower development is parametrized in polar coordinates (r, α). In FCS, the lateral shower parametrization is defined per relevant layer for each PCA bin from the longitudinal parametrization. The shower develops in millimeter (mm) units, lateral shower coordinates calculated with respect to the extrapolated position of the particle are also transformed to mm units as

$$\begin{aligned}
\Delta\eta &= \eta^{hit} - \eta^{extr}, \\
\Delta\phi &= \phi^{hit} - \phi^{extr}, \\
\Delta\eta_{mm} &= \Delta\eta \times \eta_{Jacobi,hit} \times \sqrt{r_{cell}^2 + z_{cell}^2}, \\
\Delta\phi_{mm} &= \Delta\eta \times r_{cell},
\end{aligned}$$

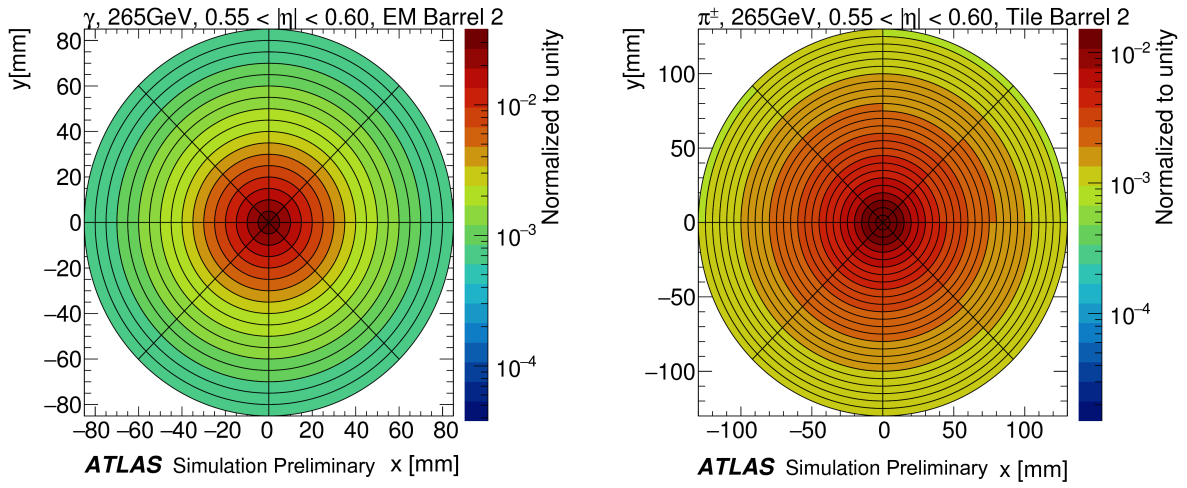


(a)

(b)

(c)

Figure 31: Longitudinal energy parametrization using a toy simulation for 1 TeV pions in $0.2 < |\eta| < 0.25$ for EMB2 (a), TileBar0 (b) and the total energy (c) [104].



(a) (b)

Figure 32: The lateral shower development of (a) photons and (b) pions of energy 265 GeV in $0.55 < |\eta| < 0.60$ parametrized in the second layer of EM barrel and Tile barrel respectively [104].

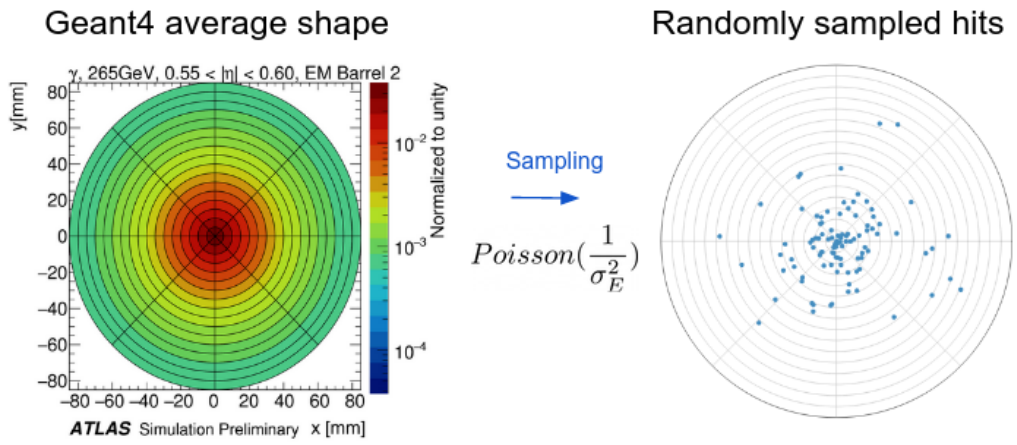


Figure 33: Shower shapes in FCS. The number of hits per layer is drawn using a Poisson distribution where its parameter σ_E^2 represents the sampling uncertainty per layer. For EMB2 for example, is approximately equals to $\frac{10\%}{\sqrt{E}}$.

where $\eta_{Jacobi,hit} = |2 \times exp(-\eta_{cell})/(1 + exp(-2 \times \eta_{cell})|$, hit represents the energy distribution of a calorimeter cell. The showers are symmetric around the center in binning of (r, α) with

$$r_{mm} = \sqrt{(\Delta\eta^{mm})^2 + (\Delta\phi^{mm})^2},$$
$$\alpha = arctan2(\Delta\phi^{mm}, \Delta\eta^{mm}).$$

The parametrized shower development in (r, α) is stored in a 2D histogram. Figure 32 shows the (r, α) representation of a photon and a pion parametrized showers in EMB2 and TileBar2 respectively. Since the showers are symmetric in ϕ, this helps reduce the memory storage of these 2D histograms by only saving the ϕ coordinates for $0 \leq \alpha \leq \pi$.

To simulate a particle's shower, the coordinates of the hits are randomly sampled from the 2D histograms, as shown in Figure 33. These hits are then assigned to cells of the simplified cuboid geometry. This geometry can lead to an incorrect cell energy deposition due to the accordion structure of the ECAL. A correction for the accordion geometry function is implemented to displace the wrong hit to cell assignment, known as the wiggle correction.

Fast simulation implementation and validation

The input to the calorimeter simulation is a particle (t, e, η) from the inner detector, where t, e and η

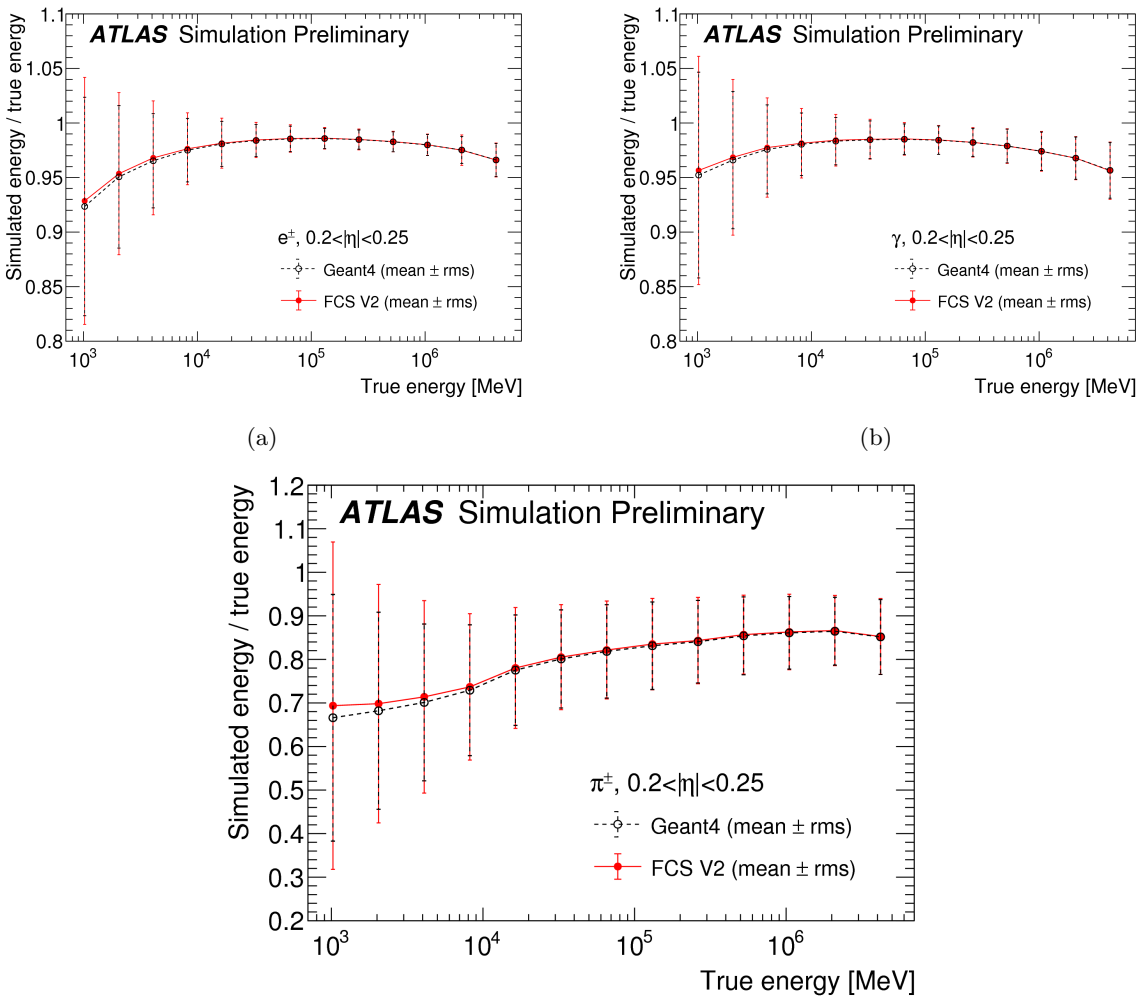


Figure 34: The total energy response for (a) electrons, (b) photons and (c) pions for truth energies from 1 GeV to 4 TeV for $0.2 < |\eta| < 0.25$. The response is defined as the ratio of measured energy to truth energy. The points represent the mean and the error bars the RMS comparing the total energy distributions for FCS V2 and Geant4 [104].

represent respectively the particle type, energy and pseudorapidity. To generate a shower, the parametrization file corresponding to a specific (t, e, η) triplet is selected.

To evaluate the simulation performance of FCS, figures of merit comparing FCS output to the Geant4 full simulation are based on the goodness of the agreement in reproducing the shower shape variables including fluctuations and correlations and key features of reconstructed object properties [103]. Figure 34 shows the good agreement of the total energy response as function of the truth energy for electrons, photons, and pions for the different energy points. Some energy response disagreements are caused by the detector geometry transitions in η bins, the cell granularity and the detector material difference. The performance is also evaluated on cluster variables. An example is shown in Figure 35. Figure 36 shows the CPU time for FCS, Geant4 and AF2 of single photons in the $0.2 < |\eta| < 0.25$ with 8 GeV, 65 GeV or 256 GeV energies generated on the calorimeter surface. FCS and AF2 are equivalent while showing important speed-up compared to Geant4.

5.4 Summary and Discussion

The ATLAS physics program relies on high precision MC simulation to test hypotheses about the underlying distribution of real data. This simulation process is CPU and time consuming and thus presents a bottleneck in the overall simulation pipeline. To alleviate the resource problem, fast simulations techniques have been studied and developed. In Run3 and beyond, as shown in Figure 37, most of the events will be modeled using fast simulations. In Run4, with the increase in luminosity, Geant4 simulation is still expected to remain the main CPU bottleneck as shown in Figure 38.

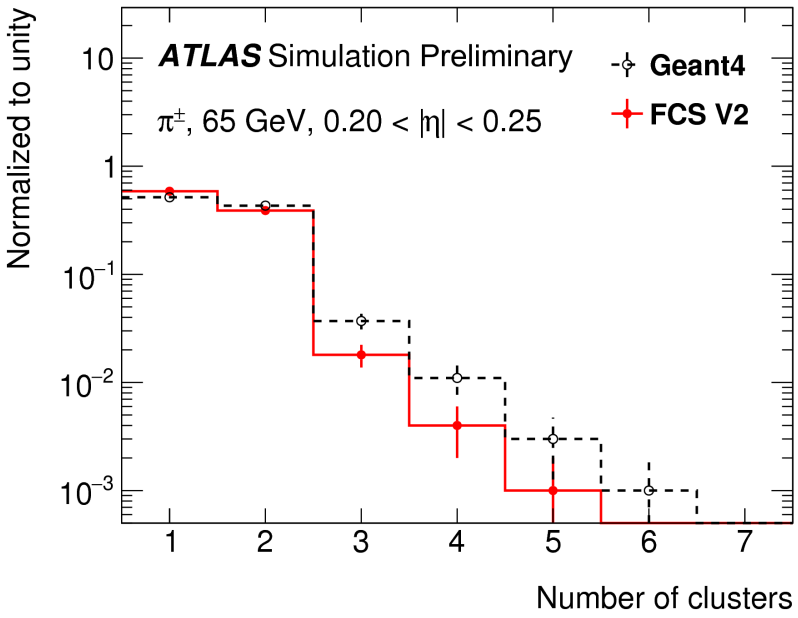


Figure 35: Number of clusters for single charged pion shower with energy 65 GeV in $0.2 < |\eta| < 0.25$ [104].

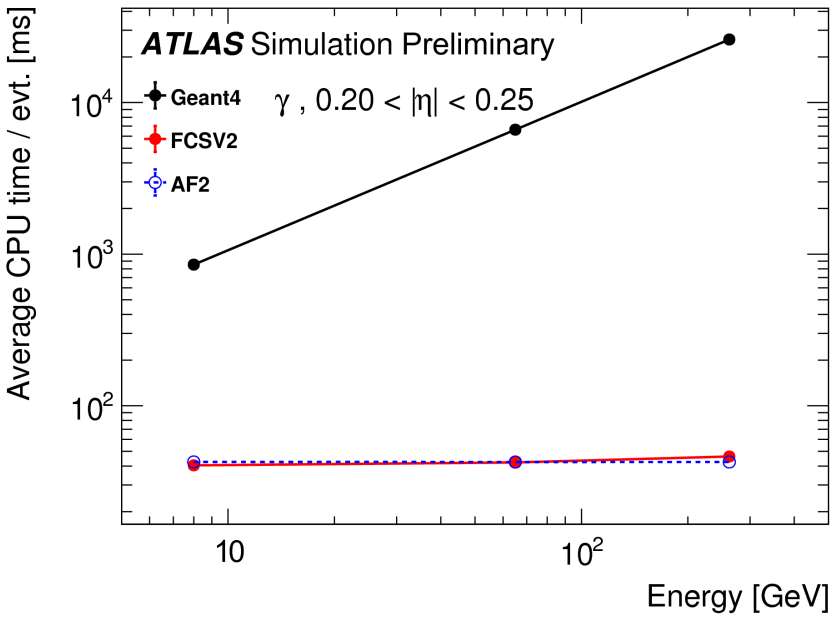


Figure 36: CPU time to simulate photons of 8 GeV, 65 GeV and 256 GeV in the range $0.2 < |\eta| < 0.25$ using Geant4 (black), FCS V2 (red) and AF2 (blue). The average time is calculated by generating 100 events for each simulation type using Athena with release 21.0.73 [104].

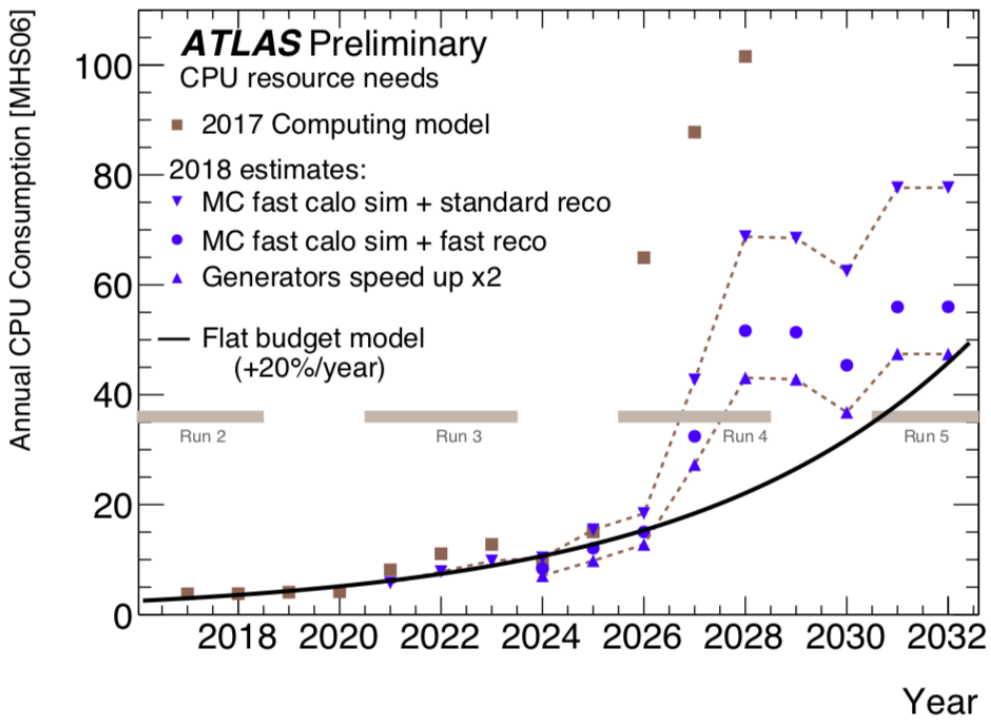


Figure 37: Estimated CPU resources (in MHS06) needed from 2018 to 2032 for both data and simulation. The plot updates the projection made in 2017 based on LHC Run-2, with updated running conditions and future computing models. The brown points represent estimates from 2017. The blue points illustrate possible improvements in three different scenarios (1) top curve with fast calo sim used for 75% of the MC simulation, (2) middle curve using in addition a faster version of reconstruction, (3) bottom curve, where the event generation time reduced by a half (software improvements or by re-using a number of the events). The solid line shows, for a flat funding scenario, the expected resources assuming a 20% performance gain per year, based on trends from current technology [88].

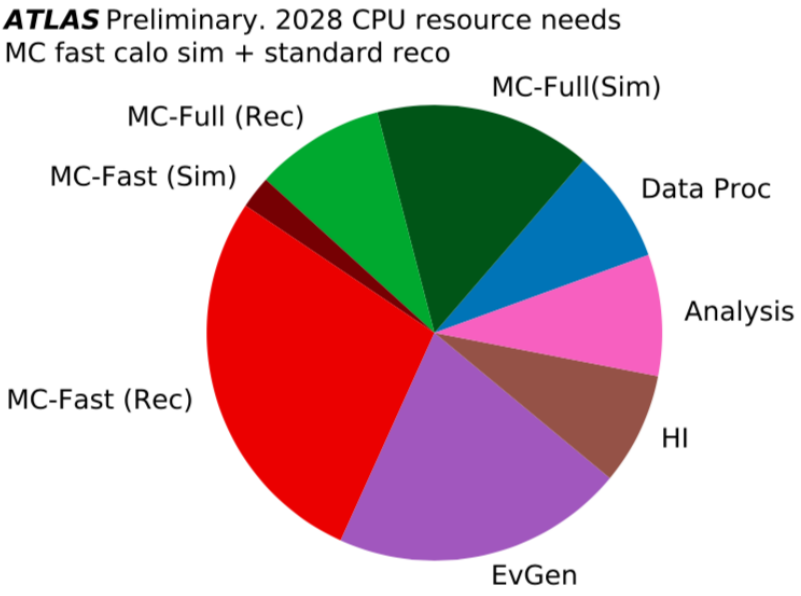


Figure 38: Fraction of CPU resources needed in 2028 at the end of Run-4. The MC-Full section in green represents the fraction of time spent on the full ATLAS Geant4 simulation. It is divided into Geant4 simulation part (Sim) and a reconstruction part (Rec), time to reconstruct the events. Similarly, the MC-Fast section in red shows this distribution for the time to run the FastCaloSim. This chart uses FastCaloSim for 75% of the MC simulation and standard reconstruction [88].

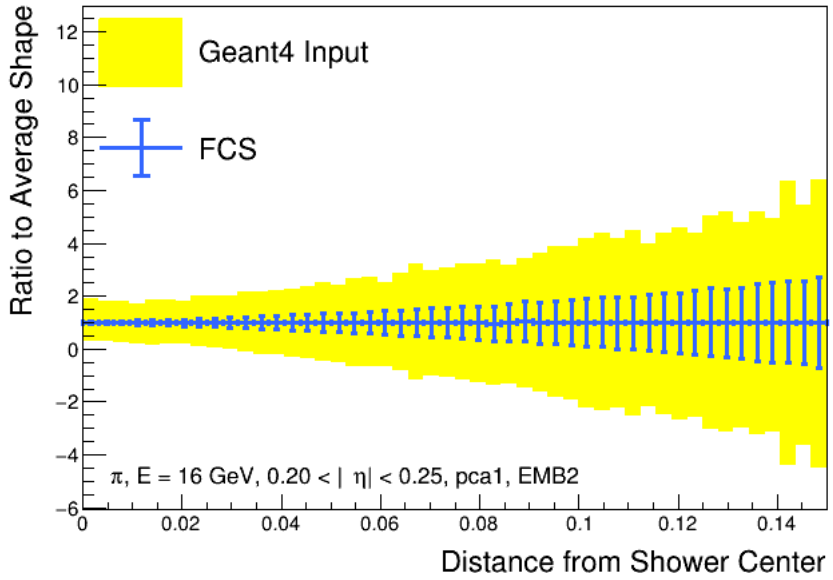


Figure 39: RMS fluctuation about the average shape as a function of distance from the shower center for 16 GeV pions, $0.2 < |\eta| < 0.25$ in EMB2.

FCS is an important improvement of the shower simulation in the ATLAS calorimeter. It is based on parametrizing the detector response of single particles instead of simulating their interactions when travelling through the detector. Although FCS considerably improves the simulation time compared to Geant4, it presents few drawbacks that impede the overall performance:

Complex parametrization chain: FCS separates the two directions of the energy into lateral and longitudinal, resulting in a loss of the correlation information between cells. Moreover, the simulation parametrization are defined for every particle type, η and energy slices resulting in 5100 configuration files. Every output is stored including the cumulative energy fractions, mean, and RMS of Gaussians and PCA matrix. This leads to an inefficient storage, especially since a fraction of these parametrization files are incorrect and the process of identifying and discarding those remains manual.

Limitation to model the correlated fluctuations: during simulation as shown in Figure 33, FCS draws randomly N_{hits} from the 2D shape histograms with the energy of a hit equivalent to $E_{hit} = E_{layer}/N_{hits}^{layer} \times w$, where the weight w depends on its radial position. Modeling shower shapes with purely random fluctuation neglects correlations. For pions, these correlations are more visible. For FCS, modeling these correlations correctly is the first challenge towards the accurate modeling of the substructure. Figure 39 shows the RMS fluctuation of the average shape as a function of distance from the shower center for 16 GeV pions, $0.2 < |\eta| < 0.25$ in EMB2. A PCA bin is chosen which has showers with significant energy deposited in EMB2. With the FCS lateral parametrization, the RMS is visibly much smaller than in Geant4.

Limitation of the PCA definition: FCS can be described as linear combination of the unitary PCA bin probability vectors. The PCA definition, by design, is not homogeneous over the energies and η regions. Taking the property of an early shower, for example, for $0 < |\eta| < 0.6$ the last PCA bin contains the earliest showers and for $\eta > 0.6$ the first PCA bin contains the earliest showers. Therefore, using a correction function to the energy or the correlation would require a manual scanning of the PCA per energy and η.

In this thesis, our goal is to address most if not all the points listed above through the use of an ML based approach. Chapters 7, 8, 9 and 10 describe the ML based solution for the ATLAS calorimeter simulation. Chapter 12 describes the ML approach to model the correlated fluctuations on top of FCS.

6 Learning to Encode and Decode with Deep Neural Networks

In the past half century, human minds became increasingly concerned with the concept of "intelligence". In several applications, this intelligence can now be modelled, applied and packaged to tackle complex and abstract tasks. The process of creating or incorporating intelligence into the world is commonly referred to as Artificial Intelligence (AI).

6.1 Probability Learning Theory

In AI applications, probability theory plays a major role in designing algorithms to approximate decision-making functions. If probabilistic modeling is used with a value indicating an absolute certainty that, for example, particle collisions will occur at a specific region in space in the detector, this is a degree of belief representation. Frequentist probability quantifies the events occurrence rate in many trials, whereas Bayesian probability explains the concept of probability as a reasonable expectation [107] of knowledge or prior information. Technically, probabilistic modeling is a mathematical description of learning from data using random variables and probability distributions. The output of this model is not a single outcome as a deterministic model, but rather a distribution. Given x an observed input vector $x = \{ x_1,\ldots,x_m \}$ of dimension m, randomly sampled from an underlying process with an unknown true distribution $p(x)$, building an approximator to this data distribution can be formulated as a probabilistic model to find the optimal values of parameters θ that best approximate $p(x)$, i.e., finding $p_\theta(x)$ such as $p_\theta(x) \approx p(x)$. For high dimensional dataset, learning a probabilistic model can then be formulated as the task of learning the joint distribution over all variables (continuous, discrete or both). In domain applications, we are often interested in conditional learning. One common example illustrating this learning is the generation of handwritten digits (MNIST) [108] to approximate the distribution $\mathrm{p}_\theta(\mathrm{x}|\mathrm{c})$, where x represents an image and the condition c represents the class/label (0 to 9).

6.2 Information Theory and Information Bottleneck

Information theory, based on probability theory and statistics, was proposed by Shannon in 1948 in his paper "A Mathematical Theory of Communication" to study quantification and transmission of information. Information is quantified with the entropy for a single random variable and the mutual information between two random variables. The entropy is an uncertainty measure of the data source s in units of bits. It is formulated as

$$H(s) = -\sum_s p(s) \log_2 p(s),$$

where $p(s)$ is the probability occurrence of the source symbol s.

Information bottleneck learning, or representation learning, is an information theory technique to derive (infer) intrinsic structure from the data. Over the years, it was used as a data compression tool for object detection and speech recognition [109]. Moreover, it was used in natural language processing to learn a distributed representation for each word, referred to as word embedding [110]. Learning word embeddings can be combined with learning image representations in a way that allows to associate text and images. This approach has been used successfully to build Google's image search, exploiting huge quantities of data to map images and queries in the same space [111]. Among the first methods in representation learning is principal component analysis proposed by Pearson in 1901 [112], a linear projection of the base feature set to a new feature space where the new features are uncorrelated. Fisher, in 1936 [113], proposed the linear discriminant analysis to project a dataset onto a lower-dimensional space with a class-separability (distance between the mean of different classes) in order to avoid overfitting. The introduction of meaningful representation with a variational principle of the input data appeared first in 1999 [114]. Extracting relevance from the data was presented as finding a compressed version of an input x that preserves the information about x using a set of bottleneck code words. In 2014, authors in References [115, 116] presented the stochastic variational algorithm for inferring and learning from a continuous unobserved or latent space in the presence of intractable posterior distributions.

6.3 Statistical Inference

Let D=$\{ x_1,\ldots, x_N \}$ denote a dataset of N data points. Given D, an observed dataset, the goal is to infer unobserved or latent variables from D, referred to as z.

Descriptive statistics explores the observed dataset by describing basic features like the average tendency or the measure of spread. However, in research we are also interested in deriving unobserved features. Inference, in this context, refers to using the observed data to retrieve properties of the underlying process. Generally, the

observation is a sample at a time, repeating the same process leads to a variation in the observation. This means a presence of an uncertainty in the observed samples, called measurement uncertainty. Then, the goal of using statistical inference is to estimate the variation over the samples or the uncertainty. The level of uncertainty is conditioned on the value of other variables, known as the conditional probability distribution. In a physics context, deriving unobserved features, such as the effect of the beam width on the collision rate, is based on using *pp* collisions as observed data. The uncertainty in this process is related to measurements and the nature of the physics process, which is intrinsically stochastic.

Bayes rule is the basis for inference and learning z given the input x from D. It is stated mathematically as

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)},$$

where $p(z|x)$, $p(x|z)$ and $p(z)$ are the posterior, the likelihood and the prior distributions respectively. When using Bayesian probabilistic inference of the latent variables to approximate the posterior, the major assumption is the prior distribution $p(z)$ over the latent variables. The likelihood definition relates the drawn variables from the prior to the observations x from D. Inference relies on conditioning on x and computing the posterior $p(z|x)$. In fact, approximating probability densities remains one of the core challenges of Bayesian statistics, for which the posterior is computationally complex. Markov Chain Monte Carlo (MCMC) [117, 118] is a method for inference approximation, based on building an ergodic Markov chain on the unknown parameters whose stationary distribution is the posterior $p(z|x)$. From the later distribution comes the sampling process from the chain, and then the approximation of this posterior with an empirical estimate built from the samples. MCMC algorithms are studied, developed and applied, however a limitation on having a fast response arises when the dataset is large or the model is very complex. An alternative method to compute the posterior distribution is Variational Bayesian inference referred to as Variational Bayes or Variational inference (VI), that proves to be faster than MCMC sampling [119]. VI approximates probability densities based on the optimization of the Kullback-Leibler (KL) divergence [120]. VI uses a family of densities and find the closest member of that family to the target density using the KL divergence. The KL divergence is a fundamental quantity in information theory to measure the difference between two probability distributions. If a probability distribution $q(x)$ is used to approximate $p(x)$ (x is a discrete random variable) then the KL divergence, defined in Equation 5, measures the loss in information using the approximation.

$$KL(p(x), q(x)) = \sum_x p(x) \ln \frac{p(x)}{q(x)}. \tag{5}$$

Note that $KL(p(x)||q(x))$ is non-negative and non-symmetric.

For VI, ζ is a family of approximate densities over the latent variables, and the goal is to find the member of that family minimizing the KL divergence to the exact posterior as

$$q^*(z) = \arg\min_{q(z)\in\zeta} KL(q(z), p(z|x)).$$

Therefore, the posterior is approximated with the optimized member q^*. A key concept in VI is choosing ζ to be simple for efficient optimization and flexible to capture the target density.

6.4 Maximum Likelihood

Given an observation x from D=$\{ x_1,\ldots, x_N \}$, a hypothesis h, the likelihood $\mathcal{L}(x|h)$ is defined as the probability of the data under the hypothesis and referred to as the probability density function (PDF). The observation x can be a single or multiple discrete values or a continuous distribution. One of the common approaches for probabilistic models is maximum log-likelihood. The idea is to build a parametric (θ parameter) model that maximizes the sum, or equivalently the average, of the log-probabilities of the data approximated by $p_\theta(x)$. For the observed dataset, D the likelihood is expressed as

$$\mathcal{L}(\theta; D) = \prod_{x\in D} p_\theta(x).$$

The goal of this optimization is to maximize the likelihood $\mathcal{L}(\theta; D)$. For stability computation, the log-likelihood is rather minimized. This can be expressed as

$$\theta = \arg\min_{\theta}(-\sum_{x \in D} ln(p_\theta(x))).$$

The gradients of this objective can be derived using a chain rule for derivative computation to iteratively achieve the local optimum of the maximum likelihood objective. The optimization process of this objective function, referred to as the loss function $J(\theta_i)$ (with the assumption that this function is differentiable) relies on different types of optimization algorithms used according to the complexity of the loss function and the derivative computation. A well known algorithm is gradient descent. It consists of multiple iterations to search for an optimal solution of the function using its gradients information to indicate the direction of the fastest local optimal value. At each iteration i, the iterative update of the parameters θ is computed as

$$\theta_{i+1} = \theta_i - \lambda_i \nabla_\theta J(\theta_i), \tag{6}$$

where λ_i represents the step size also known as the learning rate.

Batch gradient descent is the process of computing the gradients using all N data points of D, however derivative computation is an expensive operation since it scales linearly with the dataset size N and it can be intractable for datasets that do not fit in memory. A more efficient method for optimization is Stochastic Gradient Descent(SGD), which uses randomly drawn minibatches of data $M \in$ D of size N_M. With such minibatches we can form an unbiased estimator of M the "maximum likelihood criterion".

6.5 Machine Learning and Deep Learning

Machine learning (ML) is a subdomain of AI that provides systems with the ability to automatically learn and improve from experience without being explicitly programmed. It relies on an underlying hypothesis of creating a model and trying to improve it by fitting more data to the model over time [121]. In many ways, this process advances our understanding of the data and highlights the patterns used for accurate modeling.

6.6 Machine Learning tasks

With the availability of massive amounts of data that are hard to process into knowledge, ML is often used for finding underlying structures. As a result, the knowledge extracted can be used for complex predictions and as an aid to crucial decision-making processes. ML tasks are usually described as example processing. An example is a set of features describing quantitative measures from the event of interest to model, such as pixels in an image. The learning task would belong to one or more of the following ML categories: supervised, unsupervised, semi-supervised and reinforcement learning.

Supervised learning is a learning task of mapping inputs to outputs using labeled data which represents the information about the input property. If the label is discrete, the task is known as classification, to learn how to deterministically assign inputs to their correct class by penalizing the objective if the decision function is misclassifying. In the case of continuous labels, the task of learning is called regression. The goal is to approximate one or more real valued targets, optimizing generally the mean absolute error or the mean squared error between the input and the model prediction. An example of a supervised task is the classification of particles by type, such as distinguishing photons from electrons. Decision tree learning is based on building a predictive model as a tree in which the branches represent the item features and the leaves represent the target values. Building the tree is performed by taking the features and splitting the data recursively using these features. When the target variable is discrete, the model is called a classification tree and therefore the leaves are the class labels. In the case of continuous target variable it is called a regression tree. Boosting is a method which combines many trees (called weak learners) into a strong classifier. An illustrative example of Boosted Decision Trees (BDTs) application in high energy physics is tau identification in Reference [122].

In **Unsupervised learning**, the task is to learn how to cluster the input data by finding similar structures and the objective function is fully parametrized by the unlabeled data. In clustering algorithms, K-means is often presented as the most popular and simple, yet powerful, technique. A clustering algorithm tries to find meaningful groups in the dataset using only the information relative to the data points. Data points in the same *cluster* share similar properties, or at least share more properties, than with any other data point in a different cluster. Most clustering algorithms assume that the dataset structure is well described by the data point features and use straightforward distance definitions, such as the Euclidean distance, to compute

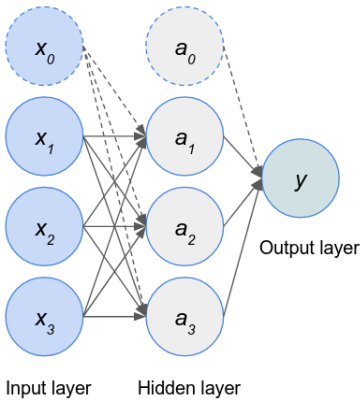


Figure 40: Basic NN representation with an input layer, a hidden layer and am output layer.

similarity. The K-means algorithm relies on the user to provide the number of clusters in the dataset: K clusters. It then performs the following actions:

1. Randomly select K data points in the dataset. These points are called centroids.
2. Compute the Euclidean distance between all the points in the dataset and the K centroids.
3. Assign each point to its closest centroid, thus forming K clusters.
4. Update the centroid value by computing the average value of each of the K clusters.
5. Repeat the steps 2,3,4 until the formed clusters remain unchanged.

There are different techniques to select the initial centroids as it often impacts the final clustering. This means that two different initialization procedures will produce different clustering. To prevent the algorithm from converging to a local minimum, the steps described above are run many times (different iterations). If we denote a data point as x, the number of elements in a cluster at any given point as n and the corresponding centroid μ, then the K-means algorithm converges when $\sum_{i=0}^{n} ||x_i - \mu||$ is minimal. This means that the sum of squares within a given cluster is minimized. This stopping criteria is referred to as *inertia*.

The unsupervised learning type can be also applied for probability distribution modeling, data representation tasks and anomaly detection. One example of an unsupervised technique with neural networks is autoencoders, where the goal is to learn a reconstruction of the input with the idea of learning a lower dimensional latent representation of that input.

For **Semi-supervised learning**, as its name indicates, it uses in its learning process a small amount of labelled data with a much larger amount of unlabeled data. It considers the problem of classification when only a small subset of the observations have corresponding class labels. Such problems are of immense practical interest in a wide range of applications, including high energy physics, such as anomaly detection to search for new physics [124].

Reinforcement Learning (RL) uses the principle of a reward to learn. It is formulated as an environment with a defined set of states, actions, and rewards. An agent interacts with the environment via actions, changing its state and receiving positive/negative reward for every action. A popular example of RL are the agents outperforming humans in various games such as GO [125]. More recently, there has been a growing interest of RL in HEP. Among the first applications is the jet grooming with RL [126].

6.7 Deep Learning Models

Inspired from neuroscience and initially designed to imitate the functioning of the human brain, Neural Networks (NNs) are among the most powerful models used in ML. They form a set of interconnected computational neurons, called also nodes or units, grouped in a chain of layers where each layer depends on the preceding one. The basic NN architecture, as shown in Figure 40, contains 3 layers: input, hidden and output layers. When the number of hidden layers is greater than one, the model is called a Deep Neural Network or simply a Deep Learning (DL) model.

Let $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ denote the input vector, the goal is to learn the function $f : X \rightarrow y$. In the input layer x_1,

x_2 and x_3 represent the input features of X. These features are connected to three neurons in the next layer, referred to as hidden neurons. They are called hidden because their ground truth value is not known, but instead learned. When all inputs are connected to all neurons in the next layer, this layer is then fully connected. The learning procedure of the network consists of automatically determining the hidden neuron's value to predict the output function f. This procedure is based on learning from (X, y) examples called observations or training examples. The observations X are assumed to be independent samples from the same underlying distribution, i.e. independently and identically distributed (i.i.d.). The output of a node given a set of inputs is called activation. In Figure 40, $a_n = g(\theta_X^T X + x_0)$ represents the activation of the neuron $n = \{1, 2, 3\}$ in the hidden layer. The final output prediction y is computed as $g(\theta_A^T A + a_0)$, with $A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, the hidden layer vector. Example of non-linear activation functions include Rectified Linear Unit (ReLu), Sigmoid and Tanh shown in Equations 7, 8 and 9 respectively.

$$g(z) = max(z, 0), \tag{7}$$

$$g(z) = \frac{1}{1 + e^{-z}}, \tag{8}$$

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{9}$$

x_0 and a_0 are scalar values called bias, they allow adjusting of the activation output. θ_X^T and θ_A^T are called parameters of the network. They represent the weights which characterize the importance of each neuron connections.

Building a DL model involves defining a model's architecture together with a learning procedure to optimize a loss function using training examples. These examples are preprocessed before the learning procedure. The prepossessing step consists of data cleaning, feature engineering and scaling. Data cleaning allows removing, or correcting inconsistent data caused by an error during the collection process of the data, a duplication issue or missing values.

The model's architecture defines the structure of the network in terms of depth (number of layers), width (number of units per layer) and the connection between the layers. The design of the model is an active area of research, and the process consists of trial and error from training to performance evaluation on validation sets [127].

The learning procedure of a mapping function from inputs to outputs with DL is achieved through a training process. The learning is based on using the gradient to descend the loss function. Let $J(\theta)$ be the loss function and θ represents a set of weights of the different neurons. When NNs are trained using maximum likelihood, the negative log-likelihood can describe the loss function, which can be the cross-entropy between the input data distribution and the learned distribution. The cross-entropy measures the difference between the two distributions. In the information theory (Section (6.2)) context, the cross-entropy computes the average number of bits required to represent data from a distribution p when another distribution q is used instead. At the beginning of the training, the weights are randomly initialized. A first forward pass from the input layer to the output layer allows the computation of the value of the output y and the loss function. The next step of the training consists of updating the weights as shown in Equation 6. This update is an iterative process defined by the number of iterations during which the model learns to produce the output by propagating the information forward from the input to the hidden layers and then to the output, this is called forward propagation. This propagation produces the value of the loss function, then used to back-propagate (the back-propagation algorithm [128] is the propagation of errors back through the network) in the network to compute the gradients to know in which direction it is better to minimize the loss function. To perform the gradient descent algorithm, one needs to compute the gradient of the loss function J by summing up the values of each sample in the training.

The evaluation of the model's performance: evaluating a DL model is a crucial step in the development pipeline. It consists of testing how well the model is fitting on test data or unseen data to assess the performance and tune the model's parameters. A good model is expected to learn to generalize well on unseen data. This property allows accurate future predictions on data that has not been used in the training. Underfitting and overfitting represent the two main characteristics of generalizing performance. The former refers to a model

that poorly fits both training data and test data. The latter fits very well the training data and poorly the test data. It occurs when the model learns details of the training data, such as noise, representing random fluctuations, which are then learned as key features of the model. On unseen data, these features are not present, which therefore impacts the model's capacity to generalize. Increasing the size or number of parameters of the model can remedy the problem of underfitting. On the other side, regularization techniques are used to avoid the model overfitting. Example of these techniques are: dropout [129], batch normalization [130], and early stopping [131]. The dropout technique consists of randomly dropping out neurons during the training phase to prevent neurons from co-adapting too much. This means temporarily ignoring a set of random neurons with all their connections. The effect of random fluctuations on hidden layers is known as internal covariate shift, which results from readjustment of the layer to new distributions of the previous layer when the weights are updated. The batch normalization technique applies a re-centering and a re-scaling of the set of neurons of a layer. Since every problem has different challenges, a tuning of the model is required to find the best configuration parameters. It requires trials, errors and metric definitions to converge on an optimized set of hyperparameters.

The model in Figure 40 is fully connected also known as a feedforward neural networks or multilayer perceptrons. There are other types of DL models such as the Convolutional Neural Networks (CNNs) adapted to image classification and object recognition problems. Unlike the full connection, each hidden neuron is only connected to pixel positions within a local area referred to as the receptive field. The weights of a neuron are shared for all positions with a convolved filter over the image. Such layers are known as a convolutional layers. The output of a convolution is called a feature map. The idea behind using the convolution is to extract the high-level features of the image, such as shapes. CNN architectures are generally multi-convolutional layers with pooling layers. These later layers apply a down-sampling of the feature map, where the size of this map is reduced in order to lower computation. Fully connected layers can also be part of a CNN architecture. They allow the learning of correlations in the features.

Recurrent Neural Networks (RNNs), also known as Auto Associative or Feedback Networks, are another class of neural networks suited to processing sequential data, such as natural language and time-series data. Unlike a feedforward network, an RNN has at least one feedback loop in order to allow information to persist. If an RNN model has a single layer of neurons, each neuron will connect its output value back to the inputs of all the other neurons. One of the most popular RNN architectures is the Long short-term memory (LSTM) capable of learning long-term dependencies. A LSTM neuron is composed of an input gate, a cell, an output gate and a forget gate. The cell is the memory unit and the three gates allow it to regulate the flow of this information.

6.8 Generative models

One of the most active fields in machine learning is generative modeling. It combines DL with statistical inference and probabilistic modeling. Generative models are well known for density estimation and data simulation. They are build of deep architectures, which are the most promising techniques for handling rich and non-linear dependencies of data spaces. Variational Auto-Encoders [115], Generative Adversarial Networks [132], Adversarial Auto-Encoders [133] are the most well known ones. These algorithms have been tested in many domain applications, with a focus on image data processing such as, forecasting from static images the prediction of a moving object in a scene [134] and facial expression editing [135]. The research on extensions of these models includes CNNs [136], RNNs [137] and hierarchical generative models [138].

6.8.1 Generative Adversarial Network

Generative Adversarial Networks (GANs) are deep generative learning models in which two non-cooperative networks define its architecture: a generator and a discriminator, as shown in Figure 41. The generator is trained to produce samples to confuse the discriminator in distinguishing fake and real samples (drawn from the training data distribution). The discriminator tries to correctly identify the original from the generated (fake) sample. When the two networks converge, the GAN is able to generate data that the trained classifier cannot recognize anymore.

Recent versions of GANs aim at improving the training procedure by varying architectures such as: conditional GANs (CGAN) [139] Deep Convolutional GAN (DCGAN) [140], Wasserstein GAN (WGAN) [141], Laplacian Pyramid GAN (LapGAN) [142] and Information Maximizing GAN (InfoGAN) [143].

6.8.2 Variational Autoencoders

An autoencoder is a neural network trained to reconstruct its input from a learned hidden representation of this input. If the representation has a lower dimension than the input, the model can be used for dimensionality

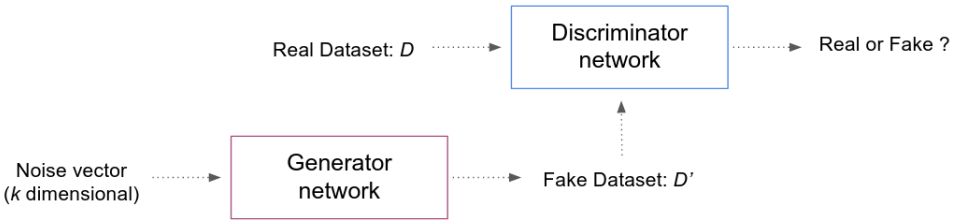


Figure 41: Composite GAN components

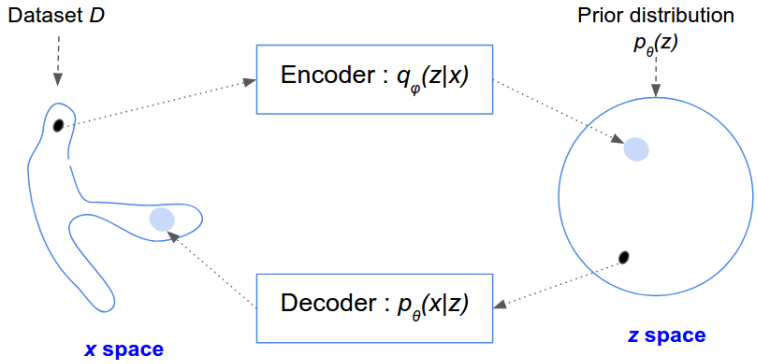


Figure 42: Stochastic mappings between observed space and latent space learned with VAEs

reduction and feature learning. In this case, it is called an **undercomplete autoencoder**. The autoencoder concept has evolved over the years with neural networks [144, 145]. The motivation behind autoencoders is related to learning low dimensional representations [146]. One of the examples of its efficient usage is to derive/retrieve information in a query database. Autoencoders as a semantic hashing approach can be used in the way they learn a reduced, binary representation, then all database entries can be stored in a hash table that maps representation vectors to entries. This hash table allows us to perform information retrieval by returning all database entries that have the same binary code as the query [127]. When the representation has a greater dimension than the input, the model is called **overcomplete autoencoder**. It learns to copy the input without learning useful features about the data distribution. Using a regularizer in this case avoids limiting the model's capacity. This consists of using a loss function to learn other features, such as the sparsity of the representation, rather than only the copy function. Autoencoders are also algorithms that learn a manifold of the data or the structure of the manifold. A manifold is a region where the data is represented as connected points associated within a neighborhood.

In the area of unsupervised deep learning, combing the idea of representation learning with latent variable models results in having the autoencoder act as a generative model. Variational Autoencoders (VAEs) [115, 116] are autoencoders designed with a prior on the representation space. A VAE learns stochastic mappings between the observed (x-space) and a latent space (z-space) as shown in Figure 42.

The VAE architecture is similar to an AE architecture shown in Figure 43. It is composed of two stacked neural networks acting as encoder and decoder. The encoder learns a mapping from the input space x to a latent space z in which a meaningful representation of the data is learned. The decoder learns the inverse mapping. Once the model is trained to reconstruct the input, the decoder can be used independently as a generator of new data by sampling from an inferred model. This property makes the VAE one of the fastest generative models [147]. The encoder and decoder models are deep neural networks. Such networks can be: MLPs, CNNs, RNNs, Attention Networks [148]. A normalizing flow provides a strategy of transforming a simple distribution into a complex one using a sequence of invertible transformations. Real-valued Non-Volume Preserving (RealNVP) [149] use normalizing flows by stacking a sequence of invertible transformations which are bijective. In each transformation, the input of dimension K is divided into two parts: the first k dimensions remain unchanged same and dimensions from $k+1$ to K are scaled and shifted with parameters as functions of the k dimensions. The autoregressive property consists of modeling sequential data, in which an observation x_i is conditioned on $x_1, .., x_{i-1}$. An autoregressive flow model applies the autoregressive property on the dimensions of each flow transformation. Examples of such models are: Neural Autoregressive Distribution Estimator (NADE) [150], Masked Autoencoder for Distribution Estimation (MADE) [151], Pixel Convolutional Neural Networks (PixelCNNs) [136] and Pixel Recurrent Neural Networks (PixelRNNs) [137]. Inverse autoregressive flow (IAF) [152]

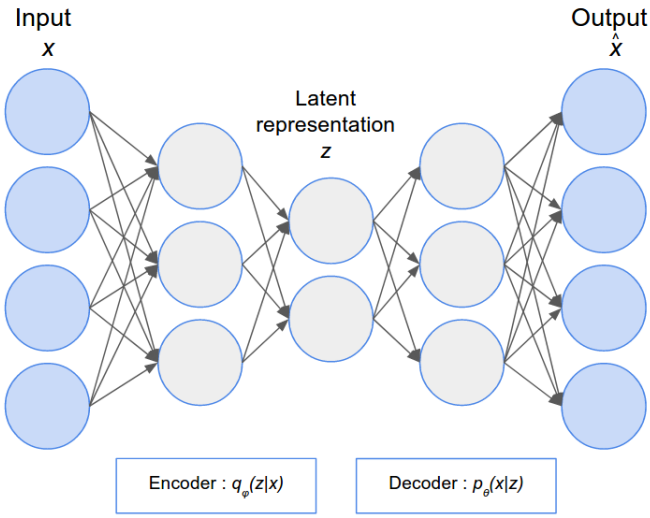


Figure 43: A fully connected autoencoder architecture

is a new type of normalizing flow which combines an autoregressive model and a reversed flow. IAF is designed to scale with high-dimensional latent spaces. VAEs with IAF achieve a significant performance improvement compared to other types of autoregressive models [152].

VAEs combine ideas from representation learning and probabilistic latent variable modelling to derive a class of deep models. Representation learning approaches have been widely used for supervised and unsupervised tasks, and in particular with deep learning architectures, with the multiple non-linear transformations yielding more abstraction and potentially more useful representations [111].

Latent variable model

Let $p_\theta(x, z)$ be the latent variable model, where x is the observed variable and z the latent variable. $p(z)$ is the prior distribution over z and the posterior inference is represented by the probability distribution $p(z|x)$.

A latent variable model introduces an unobserved random variable of m dimension for every observed data point of n dimension (where $n > m$) to explain and retrieve hidden structures. In other words, latent variables are unobserved variables, therefore not part of the dataset. VAEs are latent variable models where the latent variable z captures some structure in x. The generative process in a latent variable model of data points x consists of learning a reconstruction of x represented by $\hat{x}$ from z and can be expressed as a probability distribution $p(x|z)$ along with the prior $p(z)$. The marginal distribution over the observed variables $p_\theta(x)$, is given by

$$p_\theta(x) = \int p_\theta(x, z)dz.$$

This is also called the (single data point) marginal likelihood or the model evidence, when taking it as a function of θ representing the parameters of the model.

A deep latent variable model denotes a latent variable model whose distributions are parameterized by neural networks. We refer as well to a conditional model $p(x, z|c)$, where c is the condition. The most common models with latent variables are the ones with a factorization property:

$$p_\theta(x, z) = p(z)p_\theta(x|z).$$

The goal is to learn the generative distribution of x from z, i.e., $p(x|z)$. One assumption in this model is that the prior $p(z)$ is known. Set $p(z)$ to be a unit Gaussian. A good generative model would assign high probabilities to observed x, i.e., learning a good $p(x|z)$ is equivalent to maximizing the probability of the observed data $p(x)$. The optimization problem is then defined as

$$\max_{\theta} p_\theta(x) = \max_{\theta} \int_z p(z) p_\theta(x|z),$$

where $p(x|z)$ is parametrized by θ. This optimization involves computing the integral over z which remains intractable, i.e., non-existence of an analytical solution or an efficient estimator.

Posterior inference in a latent variable model

To overcome the tractability problem, $p(z|x)$ is approximated using the variational inference as described in Section 6.3. Variational inference models the true distribution $p(x|z)$ using a simpler parametric distribution $q_\phi(x|z)$. This modeling of $q_\phi(x|z)$ refers to the encoder part of the VAE, also called the inference model, and the parameters ϕ are called variational parameters. The distribution $q_\phi(x|z)$ can be parametrized using NNs and therefore ϕ parameters represent the weights and biases of the NNs. Applying variational inference consists of first choosing a family of distributions over the latent variables. Then the optimization procedure consists of finding the set of parameters to best approximate the posterior distribution using the KL divergence. It is formulated as

$$\min_{\phi} KL(q_\phi(z|x), p(z|x)) = \min_{\phi} \int_x q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)},$$

where $p(z|x)$ is part of this definition of the optimization problem. By decomposing the KL term

$$\begin{aligned} KL(q_\phi(z|x), p(z|x)) &= \int_x q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)} \\ &= \int_x q_\phi(z|x) \log \frac{q_\phi(z|x) p(x)}{p(x,z)} \\ &= \int_x q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(x,z)} + \int_x q_\phi(z|x) \log p(x) \\ &= -L(\phi) + \log p(x), \end{aligned}$$

where

$$L(\phi) = \int_x q_\phi(z|x) \log \frac{p(x,z)}{q_\phi(z|x)}.$$

At the end we have

$$KL(q_\phi(z|x), p(z|x)) = -L(\phi) + \log p(x) \iff L(\phi) = \log p(x) - KL(q_\phi(z|x), p(z|x)).$$

One of the properties of the KL is that it has a non-negative value. The $L(\phi)$ is a **lower bound** on the log probability of the observed data as

$$L(\phi) \leq \log p(x).$$

Therefore, the optimization objective of a VAE is this evidence lower bound (ELBO), derived through Jensen's inequality [157] stating that if $g(x)$ is a convex function on R_x, and $E[g(x)]$ and $g(E[x])$ are finite, then $g(E[x]) \leq E[g(x)]$, where $E[]$ represents the expected value. A key property of the ELBO, is the joint optimization of the encoder (ϕ) and decoder (θ) parameters using stochastic gradient descent. Note that the objective ELBO could be the sum or average of the individual data point ELBO's values.

The reparameterization trick

To perform an efficient approximation of the posterior inference and an efficient maximum likelihood estimation, authors in References [115, 116] proposed the VAE for variational inference based on a reparameterization trick. The reparameterization trick is a change of variables operation to get a differentiable ELBO (assuming a differentiable encoder and decoder and continuous latent variables). The change of variables consists of a transformation of the variable $z \sim p_\phi(z|x)$ into $z = g(\epsilon, \phi, x)$, where the random variable $\epsilon \sim p(\epsilon)$ is independent of x and ϕ. One way to get a tight bound is increasing the flexibility of the decoder. This can be seen through a connection between the ELBO and the KL divergence. Algorithm 1 defines the stochastic optimization of the ELBO.

Algorithm 1 Stochastic optimization of the ELBO. Since noise (model's noise) originates from both the minibatch sampling and sampling of $p(\epsilon)$, this is a doubly stochastic optimization procedure. This procedure is called the Auto-Encoding Variational Bayes (AEVB) algorithm [115]

Data: : D: observed dataset ;
$q_\phi(z|x)$: inference model (encoder) ;
$p_\theta(x, z)$: generative model (decoder) ;
Result: θ, ϕ learned parameters
Initialization of θ and ϕ ;
while *SGD (Stochastic Gradient Descent) not converged* **do**
  Random minibatch of data : M $\sim$ D ;
  Random noise for every datapoint: $\epsilon \sim p(\epsilon)$;
  Compute $L_{\theta,\phi}(M, \epsilon)$ and its gradients ;
  Update θ and ϕ using the SGD optimizer.
end

A common choice for the posterior distribution is the factorized Gaussian distribution:
$q_\phi(z|x) = N(z; \mu, diag(\sigma^2))$, where the encoder networks outputs the μ and σ. Using the re-parametrization trick, and choosing $\epsilon \sim N(0, I)$ z can be written as: $z = \mu + \sigma \circ \epsilon$, $\circ$ is the element wise product.

6.9 Review of Machine Learning in High Energy Physics

Machine learning has changed the way data analysis is done in all scientific fields. It has been applied to solve many problems in high energy physics in both theory to experiment, with a particular interest in developing ML based solutions for the high luminosity LHC (HL-LHC). The HL-LHC, scheduled to start taking data around 2026, will bring unprecedented amounts of data. The pileup will increase significantly, posing new challenges for the research community including the extraction of the underlying physics. Related topics where further progress is still to come include reconstruction, analysis algorithms, simulation, calibration and decreasing the data footprint.

6.9.1 Machine Learning in Theoretical High Energy Physics

In the theoretical field, ML can help in optimizing and searching for new theoretical models. One of the promising model is a neural network to model the Parton Distribution Function (PDF) [158]. The procedure starts with combining around 50 datasets from different physics processes and training a neural network per PDF. Another application of ML aims to widen our knowledge of Beyond the Standard Model (BSM) physics in order to answer the limitations of the Standard Model (SM), such as the electroweak symmetry breaking origin and the nature of dark matter. The new physical phenomenon are unknown and therefore this can be formulated as an anomaly detection search. The goal is to build a model able to derive unknown signal properties from the data. The new physics is expected to manifest as the deviations of distributions in its absence. An unsupervised learning model [159] compares if the two-dimensional input: the SM simulated event and an observed event containing a potential sign of new physics, are from the same probability distribution. The deviation measure is based on a statistical hypothesis test using a K Nearest Neighbors search to estimate the ratio of the two densities. The distribution of the test statistic is derived by a permutation test. Observing the statistical test of its tailed distribution asserts the theory of having two datasets drawn from different probability densities, indicating an anomaly. Generative models are also used for new physics search. An example using VAEs [160] is trained on SM samples from a single-lepton stream from hardware Level-1 (L1) trigger system selection. The method uses a threshold test in order to detect BSM events produced by the LHC. The threshold definition is derived from the distribution loss of the VAE when reconstructing a validation set. From this reconstruction, events are classified as potential anomalies if their loss is larger than the threshold.

6.9.2 Machine Learning in Experimental High Energy Physics

6.9.3 Online Computing

An important aspect of ML application targets the trigger level for the purpose of helping better real-time selection of interesting events to be considered for further analysis. This decision has to be made within a microsecond level and requires dedicated hardware integrated on the trigger system. Boosted decision trees (BDTs) are widely used in HEP. In decisions trees, for a classification problem, a leaf represents a decision of assigning a data sample to a class. In 2010, the CMS experiment introduced an ML based approach for level 1 trigger using BDTs [161] to approximate muon momenta, which may identify new BSM physics. The ATLAS experiment introduced a BDT at trigger level for tau identification, rejecting backgrounds from quark and gluon initiated jets [122]. A neural network based approach is used by the LHCb experiment for fast fake track and clone rejection [162].

6.9.4 Offline Computing

Particle identification: BDTs and DNNs have been popular techniques for particle identification (PID) [163, 164] and pattern recognition in different levels of detector systems. The goal is to alleviate challenges, such as the combinatorics of the tracking system [161]. For PID in Reference [164], the charged particle classes are: electron, muon, pion, kaon, proton and ghost track (fakes). The baseline PID approach, ProbNN [2], uses six binary one layer shallow neural networks [3], one for each particle type. In an ML image processing context, since detector measurements are stored digitally, creating an image from particle collisions is possible for the particle classification problem. Collision events that occur in accelerators do not produce only particles of interest, but also "noise" particles. The former represent the signal events and the later are the background. With the Higgs discovery in 2012 by the ATLAS and CMS experiments, ML was used to reproduce the discovery. This was achieved by performing a classification to choose which jets to group to reconstruct Higgs or top quark candidate mass [165]. Neutrino experiments, as well, are using ML approaches for neutrino event reconstruction and classification [166]. Another ML application is implemented to measure the associated production of top quark-antiquark pairs and Higgs bosons (ttH), with Higgs decaying to b quarks [167]. It combines multiple classifiers: a NN for the single lepton channel and a BDT for the dilepton channel.

Reconstruction: in order to reconstruct continuous quantities such as particle positions and energies, regression algorithms are used. One example relates to very high energy gamma-ray astronomy is the Cherenkov Telescope Array (CTA)[4]. A regression model in Reference [168] is based on CNNs, and it shows promising results in measuring particle features. In LHC experiments, reconstructing full tracks from a puzzle of hundreds of points consists a challenge. With the HL-LHC, this puzzle will become significantly more challenging. R&D efforts are ongoing to solve the tracking problem with ML. One of the examples is HEP.TrkX, an approach which investigates tracking using long short-term memory (LSTM) on many-core processors. Moreover, some physics phenomena happen on very small scale of time and current detector technology cannot observe them directly. The Higgs boson is one of these examples. It decays principally at the LHC collision point. A reverse approach to reconstruct the initial process is based on measuring the decay products. DNN models are used for feature extraction of particles [169] such as reconstructing images from the intensity in the calorimeters. CNNs and RNNs architectures are also used to measure electrons and photons, jets and missing energy properties.

Simulation: in the offline computing context as well, a tremendous amount of LHC computing resources are dedicated to Monte Carlo event simulation. Optimization techniques are investigated, but the speed is still a major obstacle for HL-LHC. ML solutions can alleviate the problem and provide fast simulation alternatives to model the detector response. Calorimeter simulation is a particularly time-consuming step. Early applications using generative model GAN [170] were implemented for fast electromagnetic shower simulation. Reference [123], presents the first application of GANs and VAEs for calorimeter shower simulation in the ATLAS detector. The study focuses on simulating showers of photons in central η using a range of energies from 1 GeV to 260 GeV. To assess the quality of the generative performance, shower shape variables are used, such as the energy deposited in electromagnetic calorimeter layers and calorimeter cell's relative distribution of energies. In another example in Reference [171], a GAN model, is used to simulate clusters produced by particles in the Time Projection Chamber (TPC). The model is a combination of two types of GANs: a conditional convolutional GAN and a conditional LSTM GAN. The condition refers to the initial information about the simulated particles.

ML-HEP software and open challenges beyond HEP community: multiple initiatives are undertaken in the physics community for ML software development. Toolkit for Multivariate data Analysis (TMVA) [172], integrated in ROOT [174], is a machine learning package for processing and evaluation of multivariate

[2]The network output is normalized between 0 and 1 and therefore named ProbN

[3]Usually, these networks have one hidden layer and an output layer.

[4]It consists of 19 telescopes in the Northern and 99 telescopes in the Southern Hemisphere

classification. Scikit-HEP [173] project provides a set of interfaces and Python tools for the HEP community such as root-numy, root-pandas and uproot. In the area of computing infrastructure, workflow and data management, ML solutions are also applied to optimize resource use. Application examples include predicting the popularity of a dataset from its usage to reduce disk resource saturation and to detect anomalies in network traffic to enhance security [161]. The application of ML for HEP was also made available to the data science community, such as the Higgs ML [175] and the TrackML [176] challenges. Simulated data of the Higgs decays to tau-lepton pairs were released by the ATLAS experiment in 2014, for the Higgs ML challenge on the Kaggle platform. The task was an event classification into tau-tau decay of a Higgs boson versus background. Using ML to build a fast track reconstruction particles from 3D points was the aim of the TrackML challenge, running on Kaggle and Codalab platforms. The samples were generated using an open source tracking toolkit (ACTS) [177].

As any scientific approach using ML in HEP faces multiple challenges. It requires trial and error to converge to a successful model. Optimization procedures of the model's hyperparameters using parallelization on CPUs and GPUs are also investigated [178]. Bayesian Optimization with a Gaussian Process prior and an evolutionary algorithm are two examples of such procedures. Another challenge is related to systematic uncertainties, where data augmentation and physics knowledge integration techniques are explored [179]. Furthermore, another challenge is the sculpting of variables, such as the mass distributions or the PID information, e.g., using BDTs [180].

7 FastCaloVSim: Fast Calorimeter Shower Simulation with Variational Autoencoders

The LHC experiments use a huge amount of computing resources to track and measure properties of a massive number of interacting particles. This process requires a dedicated hardware and software infrastructure. Detector simulation using the Geant4 toolkit is the main component that describes the phenomena of high energy particle interactions with the detector. The simulation output is used to test theories, both well-established, such as the Standard Model, and theories related to physics beyond the Standard Model. The simulation process for LHC experiments such as ATLAS provides a detailed description of the detector response. This description requires at least three components: the geometry of the detector, the physics modules and the primary particle definition. The first module describes the shapes, materials and the positioning of the different components in the detector. The second module describes the particles, energies and interactions. The third module represents the properties of the primary particles. The simulation process models every aspect of the interaction within the sensitive detector elements and all the remaining effects from the design choices, including dead material and cables. Therefore, it is inherently complex and slow. The resources consumed by this full simulation limit the statistics due to finite CPU and disk space availability. As an example, in order to generate Run2 statistics, the ATLAS Monte Carlo sample production used around 40 % of the total CPU resources [186].

In the ATLAS simulation chain, describing the shower development in the calorimeter is a key step and, at the same time, the most CPU intensive one. Annually, the shower simulation uses billions of CPU hours [187]. For a single event of particle showers, simulating the step-by-step shower development in the calorimeter impedes a straightforward acceleration (such as porting to GPUs). Furthermore, with high energies, the showers are created deeper in the calorimeter and therefore their simulation consumes even more resources to fully describe them. A fast approach would reduce the computational simulation footprint. This sets the stage for machine learning based techniques. The FastCaloVSim approach, detailed in the following chapters, describes a fast data-driven simulation technique based on Variational Autoencoders (VAEs).

The next Chapters 8, 9 and 10 provide the details of FastCaloVSim, among those: shower resolution, architectural design choices, physics knowledge integration and the patterns that emerge from understanding and modeling the complexity of the electromagnetic and hadronic showers in the core of the ATLAS calorimeter. This chapter provides technical details and descriptions of the common parts in the different studies in this thesis.

7.1 Geant4 samples

The Geant4 events used for training and validation of the FastCaloVSim approach are produced with the ATLAS simulation infrastructure. The datasets consist of events of single photons and pions simulated using Geant4 10.1.patch03.atlas02 with the standard MC16 Run2 ATLAS geometry tagged as ATLAS-R2-2016-01-00-01. This simulation includes the conditions tag OFLCOND-MC16-SDR-14. Simulating the physics is defined using the FTFP-BERT reference physics list [189], used as a standard inside ATLAS, offering the best tradeoffs between speed and accuracy. The FTFP refers to the FTF parton string model [191] to simulate hadron-nucleus interactions starting from 4 GeV energies, followed by a precompound Geant4 mode, applied to de-excite the nucleus. Bertini cascade [190] is used for low energy hadron–nucleus interactions.

The VAE model of the FastCaloVSim approach is designed to only learn the interactions happening in the calorimeter volume. Therefore, the truth particles of all the Geant4 samples used for different training and validation phases are generated at the boundary between the inner detector and the calorimeter in the ISF. This is known as a generation at the calorimeter surface. This generation is done by shifting the vertex of the primary particles to the surface of the calorimeter. Moreover, these samples were produced with the same settings as for Run2, i.e., without the beam spread in the ATLAS interaction region. All of the cross talk between neighboring cells and dead cells and the electronic noise are turned off in the digitization step of the simulation chain.

The Geant4 dataset used to train the model in Chapter 8 is the ATLAS production from 2018. For Chapters 9 and 10, we use the 2019 production. Both productions have single-particle samples, photons, electrons, and pions with approximately 4500 samples in total. The calorimeter is segmented into 100 η slices of equal size of 0.05 in order to homogenize the structure in each slice and simplify the definition of samples. For each particle and for each slice, around 11 (1 GeV-1 TeV) and 17 (64 MeV- 4 TeV) energy samples are produced in a logarithmic scale for 2018 and 2019 productions respectively. The particle generation process for each sample considers both sides of the detector per η slice and is uniformly distributed in ϕ. The number of events in each sample is 10000 for energies up to 256 GeV and at higher energies, such as 1 TeV (4 TeV), only 3000 (1000) events were produced due to the long simulation time. Each of the samples contains the energy depositions

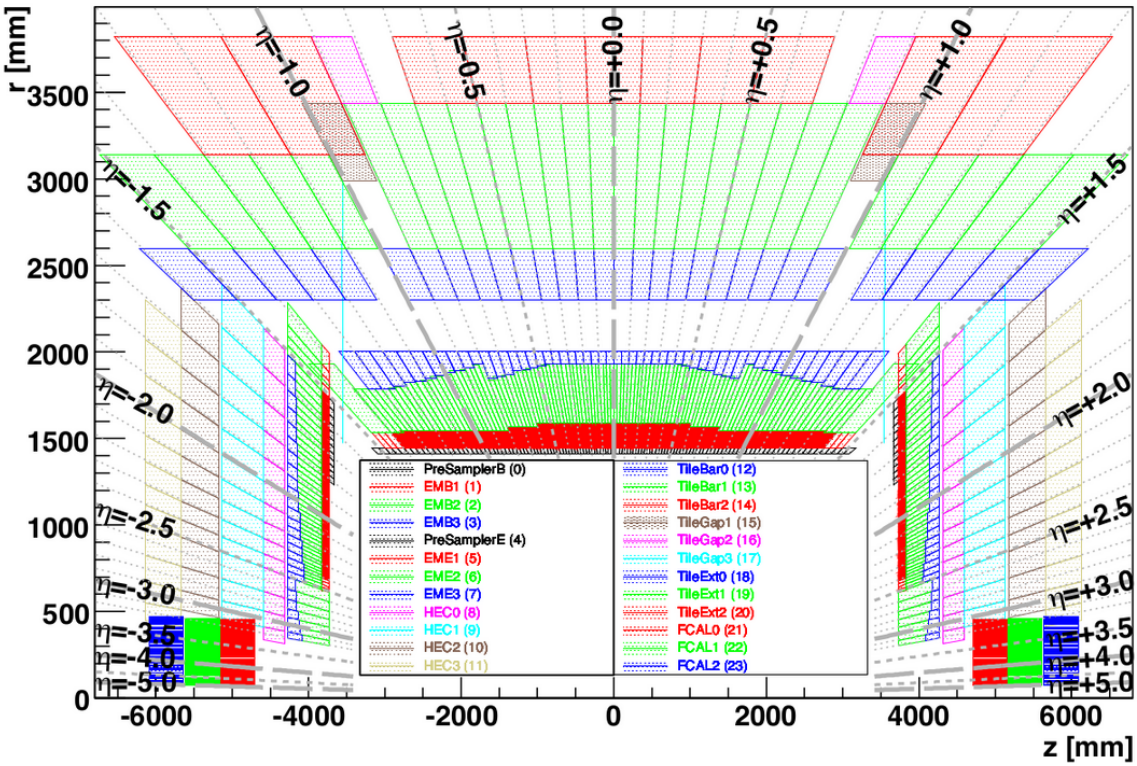


Figure 44: Visualization of ATLAS reconstruction geometry of the calorimeter in (r, z) plane.

in the ATLAS calorimeter layers. Figure 44 shows the geometry of the ATLAS calorimeter in the (r, z) plane. The colors represent the different layers of the calorimeter, with a total of 24, where each layer is only defined in a certain η region. The design of the calorimeter is based on a segmentation into three-dimensional cells, which define the granularity of the calorimeter. These cells have non-uniform sizes and shapes, in addition to some partial overlaps. Therefore, their changing sizes between layers and calorimeter regions creates a very complex structure to model. The next section introduces the different strategies to represent the showers using the Geant4 sample information of cells and hits.

7.2 Shower Representation and Granularity: From Cells to Voxels to Centroids

The ATLAS detector can be seen as a set of cameras to record pictures of collision events. At the calorimeter level, the incident particle energy is almost fully absorbed in its layers. This energy is then transformed into electronic signals. Considering calorimeter cells as cuboids, the shower development in the calorimeter can be seen as a three-dimensional image through layers as illustrated in Figure 45, where pixels represent cells. The first two dimensions are the spatial coordinates of a cell in the detector, and the third dimension represents the energy deposited by the particle in the corresponding cell. It is encoded as the intensity of the pixel. In order to train a model to learn particle showering in the calorimeter, the first step is to perform data preprocessing. This consists of converting the Geant4 data into an input structure such as an image. The model is then trained on the images of energy depositions originating from particle interaction with each calorimeter layer.

The output of the Geant4 simulation, called hits, represents the energy deposition from a shower at different space points. The number of these hits per shower is very large and their location is very sparse, due to the fact that Geant4 simulates every single interaction of the incident particle in the detector volume. Figure 46 illustrates the Geant4 hits simulated for two events in EMB2 for photons with an energy of 65 GeV in $0.2 < |\eta| < 0.25$. To overcome this sparsity problem, the hits can be aggregated into different volume spaces. Moreover, in order to train the FastCaloVSim model to learn this process of development in the calorimeter, the structure of the input has to be well-defined and identical for all events. This structure (image) is represented by the number of the volume spaces (granularity) in 2D.

The structured input is defined along three stages with increasing granularity of this structure: cell level, voxel level and centroid level. Figure 47 illustrates a representation of EMB2 layer structure. In a cell definition the structure has in total 49 volumes, for voxel definition, it has 640 and for centroid definition, the total number of volumes is 1000.

The cell level, inspired from the calorimeter cell granularity, consists of building a 2D image in $\eta \times \phi$, where

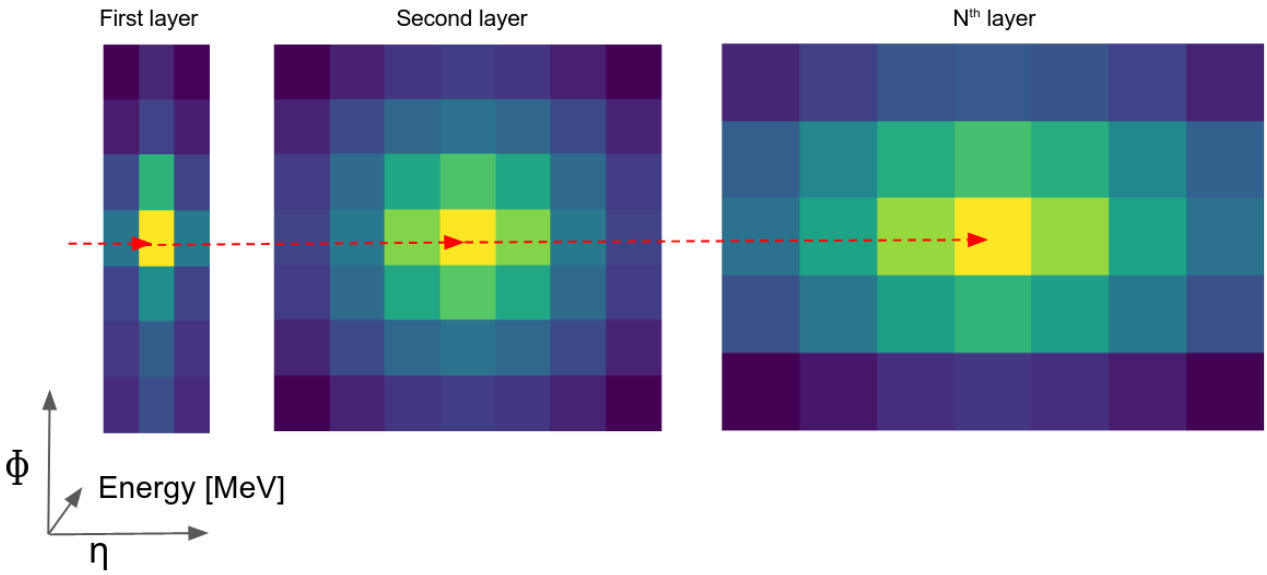
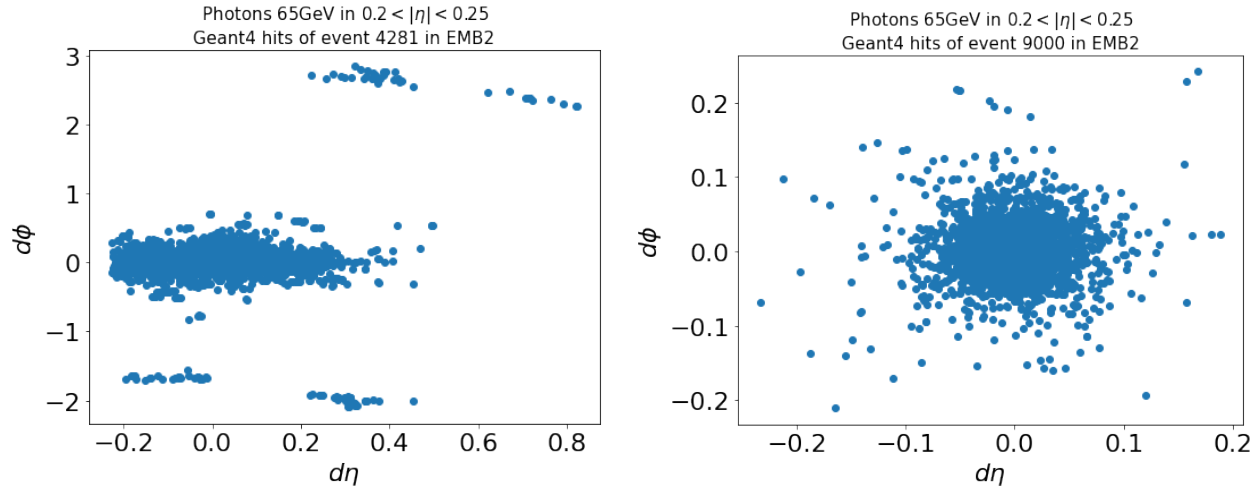


Figure 45: Representation of average energy depositions of a shower development in the calorimeter as 2D grid for each calorimeter layer. The red line presents the trajectory of the truth particle.



(a) (b)

Figure 46: Geant4 hits distribution in $d\eta \times d\phi$, centered on the primary particle's initial trajectory for (a) event number 4281 and (b) event number 9000 of photons with 65 GeV energy in $0.2 < \eta < 0.25$. In (a) the activity at the opposite side of the detector is likely caused by neutrons which produce hits later due to their slow hit time.

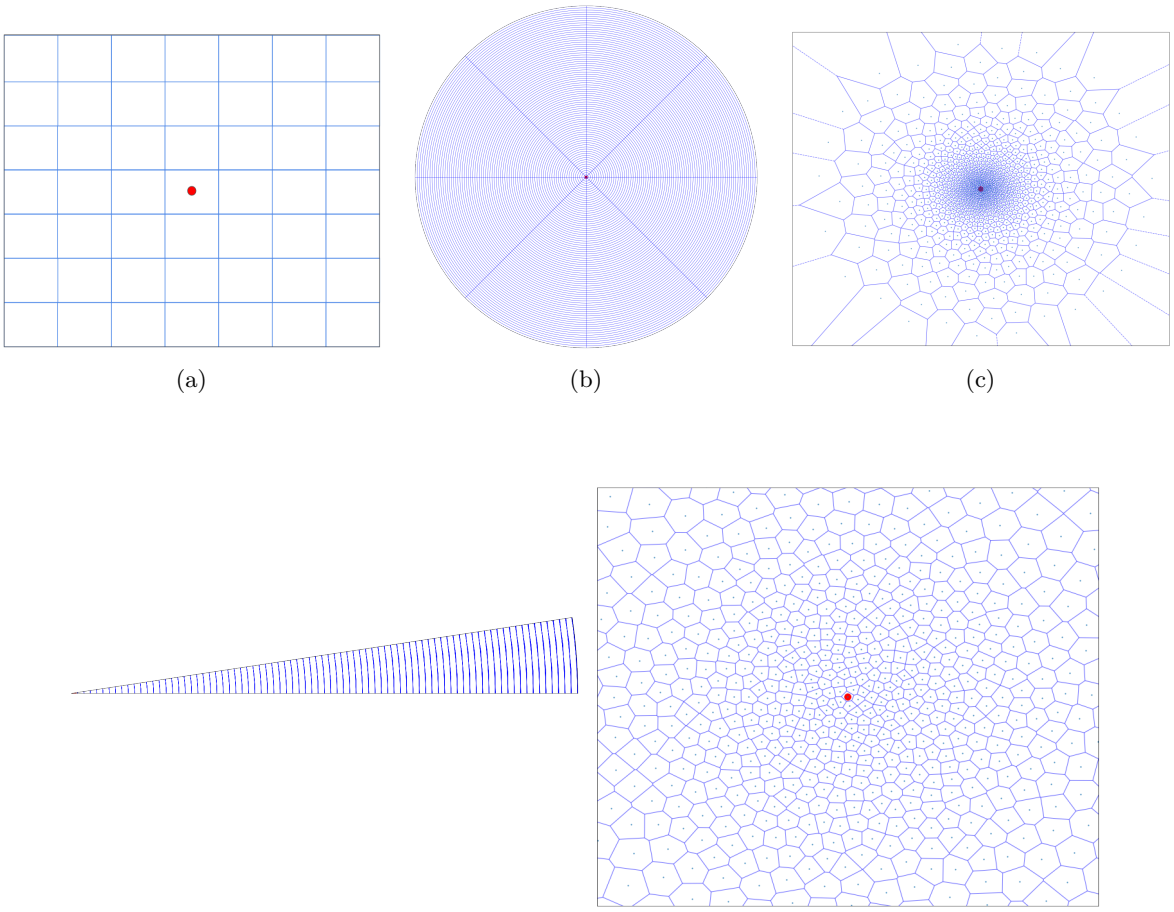


Figure 47: Image representation for (a) cells, (b) voxels and (c) centroids for photons in EMB2. (d) is a zoomed version of the voxels and (e) is a zoomed version of the centroids.

the volume space value or the pixel intensity represents the cell energy computed as the sum of the hit energies assigned to that cell. This definition of cells is intuitive and simple to derive, since Geant4 samples contain the identifier of the cells (ID). By using the cell ID, all its relevant information, such as η and ϕ, can be retrieved from the geometry file of the detector. Defining a shower representation using cells is limited because cells are too coarse to capture the intricate details of a shower. Moreover, they are completely dependent on the geometry of the calorimeter, which implies defining a representation per calorimeter region where cells have similar shapes.

In order to address the geometry dependence and to increase the granularity, a second structure, composed of 2D images called the voxel level, is defined. The Geant4 hits are then grouped into voxels in polar coordinates (r, α), inspired from the circle-like energy development process of showers.

The third approach does not rely on a physics definition, but is rather derived from an ML voxelization using a clustering algorithm. The definition is straightforward, since it relies on using hit information in a 2D space to derive centroids for each cluster of hits. These centroids are derived independently, from the truth energy. Furthermore, this definition is completely independent of the detector geometry. The ML voxelization provides more flexibility in defining the granularity of the images, referred to as the number of clusters. With highly granular images, the energy deposition of particles traversing the ATLAS detector can be modeled at a small scale. Moreover, with high granularity, the tails of the shower shape distributions are distinct and as a consequence they can be well reproduced as well as the core of the shower.

The first prototype that learned to simulate showers using the cell information as input structure is detailed in Chapter 8. It also represents the first application of generative models (VAEs and GANs) to simulate showers in the ATLAS calorimeter. The content of this chapter is adapted from the work published in References [123, 213, 214]. Chapter 9 details the pipeline to train, test and validate on voxel level inputs and Chapter 10 describes the centroid level approach of FastCaloVSim. In each of these chapters, we propose a VAE model to address the complex learning process. Each model learns features of the showers at different levels of details. Additionally, by incorporating information about the energy and η, the learning becomes global in an efficient and flexible way.

7.3 Overview of Training Strategies

Simulating showers using FastCaloVSim started with investigating and exploring the performance on a single calorimeter layer with a single energy in a central η region. The next training consisted of learning a complex structure of electromagnetic showers by considering the four electromagnetic layers of the ATLAS calorimeter in the barrel region: PresamplerB, EMB1, EMB2 and EMB3 for a single energy and η slice. This allows us to optimize the network to learn to generate showers in unevenly and spatially segmented layers. It also demonstrated the learning potential of the correlations between layers.

Concretely, a generalizable simulator should not only learn an approximation function to generate showers, but also support a conditional function based on the incident particle's features. Moving from a single energy to multiple energies in a single η slice is the purpose of designing a conditional VAE. Within the conditional model, a probability distribution of generating showers with specific energy values is learned from the training samples. In parallel, using a single energy and conditioning on a range of η slices was a proof of concept on the feasibility of conditioning on η. Tracking the network performance from all the previous steps allowed the design of a VAE model that learns to generate showers based on two conditions of energy and η using information from all the calorimeter layers.

The FastCaloVSim approach is also designed per particle type. This choice is motivated by the fact that showers originating from the different types of particles develop quantitatively and qualitatively differently. The above strategies are first applied on photons and then on pions. Figure 48 illustrates an example of a shower development of a single photon and a single pion with the same energy of 50 GeV in $0.2 < |\eta| < 0.25$ in EMB2. This shows that photons have compact shower development, i.e., a narrower and shallower shower. Hadronic showers, on the other hand, are characterized by a wider shower development, and they can penetrate deeper into the calorimeter. This means that the energy for photons and pions is not deposited in the same relevant layers, which leads to a different input structure and therefore two different models.

7.4 Shower Observables

In order to model a physics process, n events of this process are simulated. Looking at the distributions describing measured properties of these events provides an understanding of their underlying behavior. The evaluation is therefore based on shower observables, a set of variables that describe shower properties in terms of energy and directions, such as longitudinal and lateral directions. These variables are used by ATLAS for

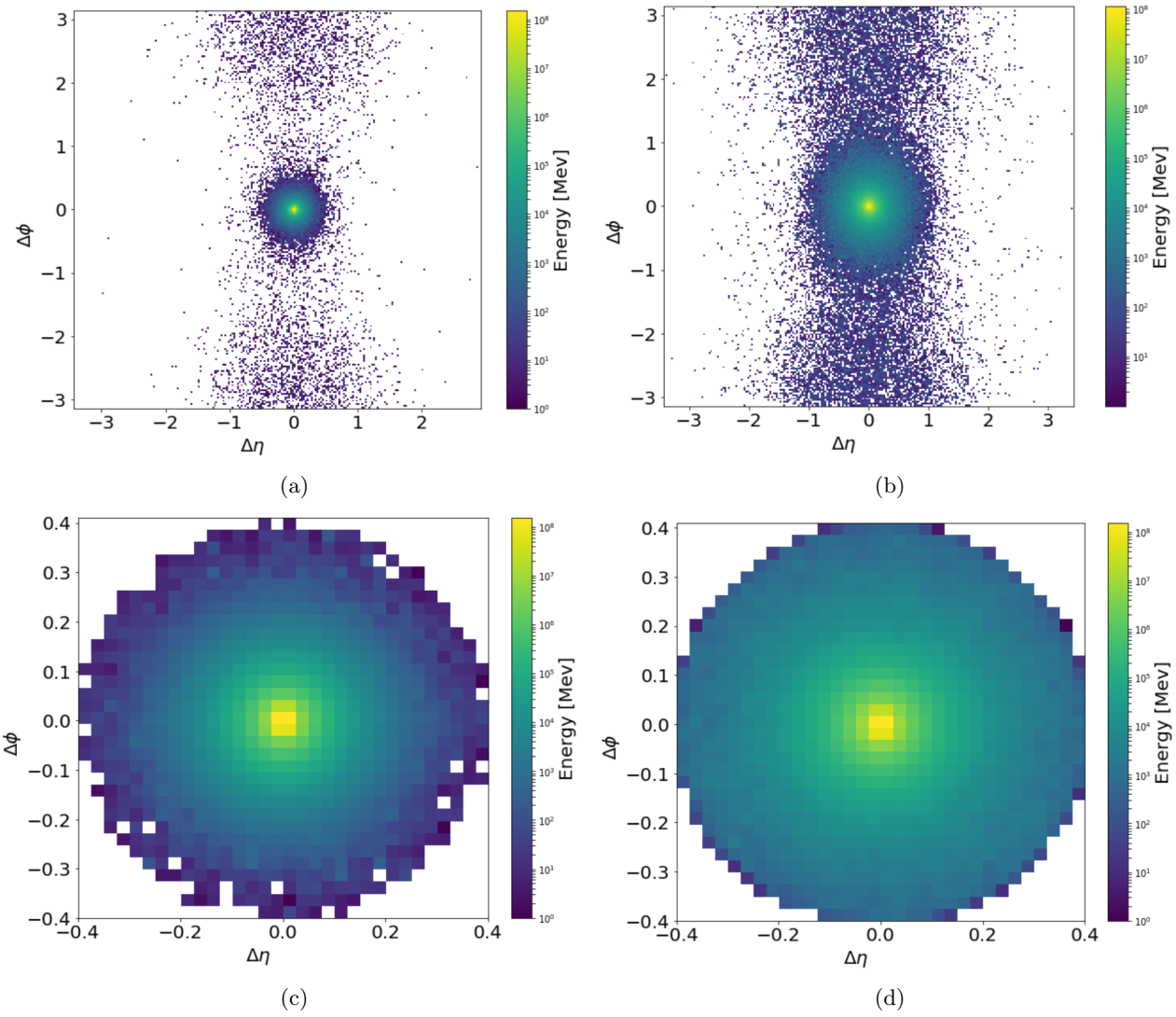


Figure 48: Shower development in EMB2 for two particles types: (a) (c) photon (b) (d) pion with an energy of 50 GeV in 0.2< $|\eta|$ <0.25. (c) and (d) are zoomed in the 0.4 region in $\Delta\eta$ and $\Delta\phi$. This development is represented as a function of the relative positions to the truth particles, and the z axis represents the deposited energy in each ($\Delta\eta$,$\Delta\phi$) bin.

Variable	Description	Formula
E_i	Energy deposited in the ^ith calorimeter layer	$\sum_{j\in pixel} E_{ij}$ (a pixel can be a cell, voxel, or centroid)
E_{Tot}	Total energy deposited in the calorimeter	$\sum_{i=0}^{l} E_i$ (l number of considered layers)
e233	Uncalibrated energy of the middle sampling in a rectangle of size 3×3 (in cell units $\eta \times \phi$)	$\sum_{j=0}^{3\times3} E_2cell_j$
e235	Uncalibrated energy of the middle sampling in a rectangle of size 3×5	$\sum_{j=0}^{3\times5} E_2cell_j$
e255	Uncalibrated energy of the middle sampling in a rectangle of size 5×5	$\sum_{j=0}^{5\times5} E_2cell_j$
e237	Uncalibrated energy of the middle sampling in a rectangle of size 3×7	$\sum_{j=0}^{3\times7} E_2cell_j$
e277	Uncalibrated energy of the middle sampling in a rectangle of size 7×7	$\sum_{j=0}^{7\times7} E_2cell_j$
ecore	Core energy in the ECAL using core cell selection	$E_0(3\times3)+E_1(15\times2)+E_2(5\times5)+E_3(3\times5)$
f1	Fraction of energy reconstructed in the first sampling	E_1/E, where E_1 is energy in all strips belonging to the cluster and E is the total energy reconstructed in the electromagnetic calorimeter cluster
f3	Fraction of energy reconstructed in the third sampling layer	E_3/E
weta or weta1	Shower width using +/-3 strips around the one with the maximal energy deposit	$\sqrt{\sum(E_i)\times(i-i_{max})^2/\sum(E_i)}$ (i: the number of the strip and i_{max} the most energetic strip number
widths1	Same as weta1 but without corrections on particle impact point inside the cell	
weta2	The lateral width calculated with a window of 3×5 cells using the energy weighted sum over all cells, which depends on the particle impact point inside the cell	$\sqrt{(\sum E_i\times\eta^2)/\sum E_i - (\sum E_i\times\eta/\sum E_i)^2}$, where E_i is the energy of the i-th cell
wtots1	Shower width determined in a window $\Delta\eta\times\Delta\phi = 0.0625\times 0.2$, corresponding typically to 20 strips in η	$\sqrt{\sum E_i\times(i-i_{max})^2/\sum E_i}$, where i is the strip number and imax the strip number of the first local maximum
emaxs1	Energy of strip with maximal energy deposit	
e2tsts1	Second maximum in strips calculated by summing 3 strips	
Eratio	Difference between the highest and second-highest energy deposit in the cells of the ith layer, divided by the sum	$emaxs1 - e2tsts1/emaxs1 + e2tsts1$
emins1	Energy reconstructed in the strip with the minimal value between the first and second maximum	
DeltaE	Difference between the second maximum in strips and the energy reconstructed in the strip, with the minimal value between the first and second maximum	$e2tsts1 - emins1$
Rη or Reta	Reconstructed η	$e237/e277$
Rϕ or Rphi	Reconstructed ϕ	$e233/e237$

Table 5: Shower observables for photons.

particle identification, e.g., the ones described in Table 4 for photons. These same features are used to validate the generation performance of the FastCaloVSim on single photons, along with the set of the *EGamma* [182] variables listed in Table 5.

The validation of single pions is based on reconstructing their showers as jets. This reconstruction is derived from the ATLAS topological cell clustering [203]. This approach allows us to reconstruct single-particle showers

Variable	Description	Formula
r_i	Radial distance to shower axis	$r_i = \|\vec{x_i} - \vec{c} \times \vec{s}\|$; with $\vec{x_i}$ $\vec{c}$ the center of gravity, $\vec{s}$ the shower axis.
λ_i	Longitudinal distance from shower center of gravity	$\lambda_i = (\vec{x_i} - \vec{c}).\vec{s}$
$< r >$	First moment in r	$\nu_{cell} = r_i$, $n = 1$
$< \lambda >$	First moment in λ	$\nu_{cell} = \lambda_i$, n=1
$< r^2 >$	Second moment in r: lateral extension of the topo-cluster	$\nu_{cell} = \lambda_i$, n=2
$< \lambda^2 >$	Second moment in λ: longitudinal extension of the topo-cluster	$\nu_{cell} = \lambda_i$, n=2
NTop	Number of topo-clusters inside anti-k_t jets formed with R=0.4	
p_T	transverse momentum: jet energy resolution	
m	Mass	
ΔR or deltaR	the distance of the cluster to the true pion	
Isolation	Measures the sampling layer energy E_s^{EM} weighted fraction of non-clustered neighbor cells on the outer perimeter of the topo-cluster	$\frac{\sum_{s \in \text{samplings with} E_s^{EM}>0} E_s^{EM} N_{cell,s}^{noclus}/N_{cell,s}^{neighbour}}{\sum_{s \in \text{samplings with} E_s^{EM}>0}}$, where E_s^{EM} is the sum of energies in a topo-cluster located in a given layer s, $N_{cell,s}^{noclus}$ is the number of calorimeter cells in s neighboring a topo-cluster but not collected into one themselves, $N_{cell,s}^{noclus}/N_{cell,s}^{neighbour}$ is the ratio to the number of all neighboring cells
Q	Charge produced by an energy deposition E	$Q = R \times E/W_{ion}$, where R is a recombination factor, W_{ion} is the average ionization energy which is equivalent to 23.6 eV for liquid argon

Table 6: Shower observables for pions.

with a higher precision in both energy and shape. This is done by retrieving the significant signal from the background, such as the electronic noise, and also from any source of fluctuations, such as pile-up. The signal extraction process is based on reconstructing three-dimensional energy groups from the secondaries in the active volume of the calorimeter. Depending on the properties of the incoming particle, the individual topo-clusters can contain a full or a partial response to a single particle, a response of different particles or a hybrid of full and partial secondaries.

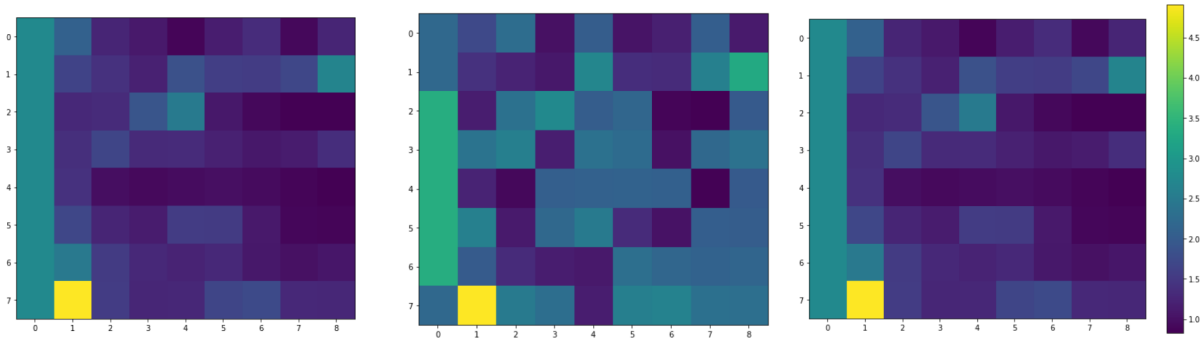
Topo-cluster moments or cluster moments define the list of reconstructed observables. Table 6 shows the main observables used to determine the location and size of the clusters. The majority of the cluster moments are defined at an order n for a calorimeter cell variable ν_{cell} as

$$< \nu_{cell}^n >= \frac{\sum_{i|E_{cell,i}^{EM}>0} \omega_{cell,i}^{geo} E_{cell,i}^{EM} \nu_{cell,i}^n}{\sum_{i|E_{cell,i}^{EM}>0} \omega_{cell,i}^{geo} E_{cell,i}^{EM}.} \tag{10}$$

The topo-cluster in a jet with the highest $p_{T,clus}^{EM}$ is called the leading cluster [203]. It is found from anti-k_T jets reconstructed with R=0.4.

7.5 Validation Performance

The strategy adopted to analyze the performance of the model is based on two validation steps. The first step compares the VAE generated output to Geant4 samples. The second step is based on a comparison of the reconstructed objects between a standard simulation (Geant4 samples) and a VAE based simulation. The reconstructed objects are obtained after running the reconstruction in the ATLAS Athena framework. This process (discussed in Section 7.5.2) involves a conversion and an integration of the FastCaloVSim approach into the Athena framework.



(a) (b) (c)

Figure 49: SSIM metric compared to the MSE of (a) 8×9 matrix of energies, (b) : (a)+noise, (c): (a)+constant.

7.5.1 Standalone Validation

The standalone validation itself is divided into two parts: reconstruction and simulation. In the former, both the encoder and decoder networks are used to assess the quality of the VAE reconstruction of the Geant4 input showers. This reconstruction allows us to understand the network behavior and is therefore used as a guide in the tuning of the hyperparameters of the model. Because the encoder is used, a shower-to-shower comparison is possible based on image quality assessment.

The definition of a single metric to select the best set of hyperparameters is not feasible due to the complexity of the process. A model is qualified as good if it can reproduce all the shower variables. On the other hand, a metric can be defined to automatically discard poorly performing models. Pixel intensities are strongly interdependent when they are spatially close. The same pattern is present for spatially close energies. A metric such as the Structure Similarity Index Metric (SSIM) [204] can be used for hyperparameter selection. It is a perception-based metric that measures the similarity between two images by taking into account the structure of the image rather than only the pixel values (such as the Mean Squared Error (MSE)). Figure 49 shows the SSIM values compared to MSE for an example of 8×9 matrix of energies in the case where a noise value is added to the matrix (b) and in the case where a constant is added (c). The MSE value remains the same for both cases, while the SSIM metric can perceive the similarity by assigning a higher value to (c) and a lower value to (b).

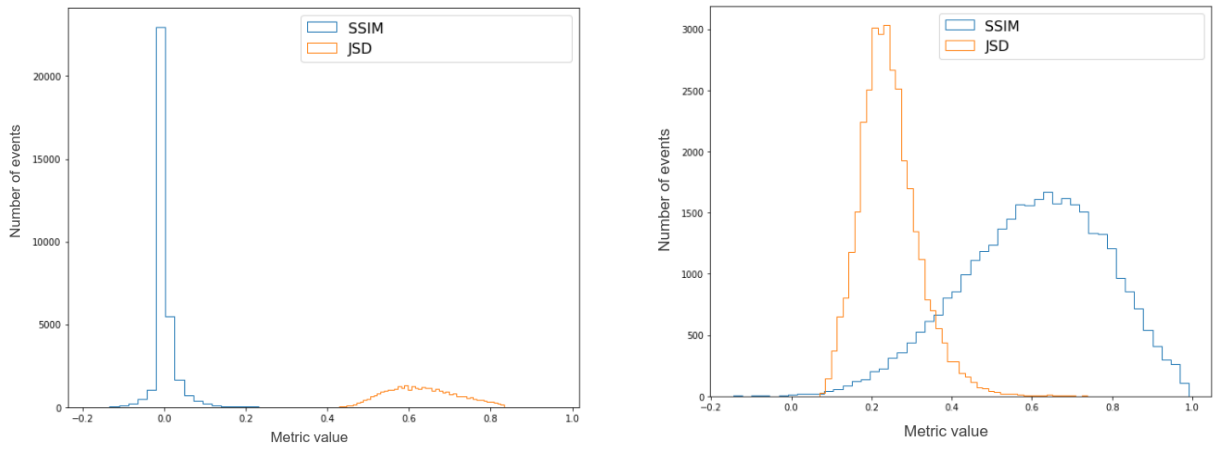
It is defined as

$$SSIM(x,\tilde{x}) = \frac{(2\mu_x\mu_{\tilde{x}} + c1)(2\sigma_{x,\tilde{x}})}{(\mu_x^2 + \mu_{\tilde{x}}^2 + c1)(\sigma_x^2 + \sigma_{\tilde{x}}^2 + c2)},$$

where x is the input shower image, $\tilde{x}$ is the reconstructed input shower image, μ the average value of the image, σ the variance of the image, $\sigma_{x,\tilde{x}}$ the covariance of x and $\tilde{x}$ and ($c1$,$c2$) two variables to ensure a non-zero denominator. Structurally similar images have an SSIM close to 1.

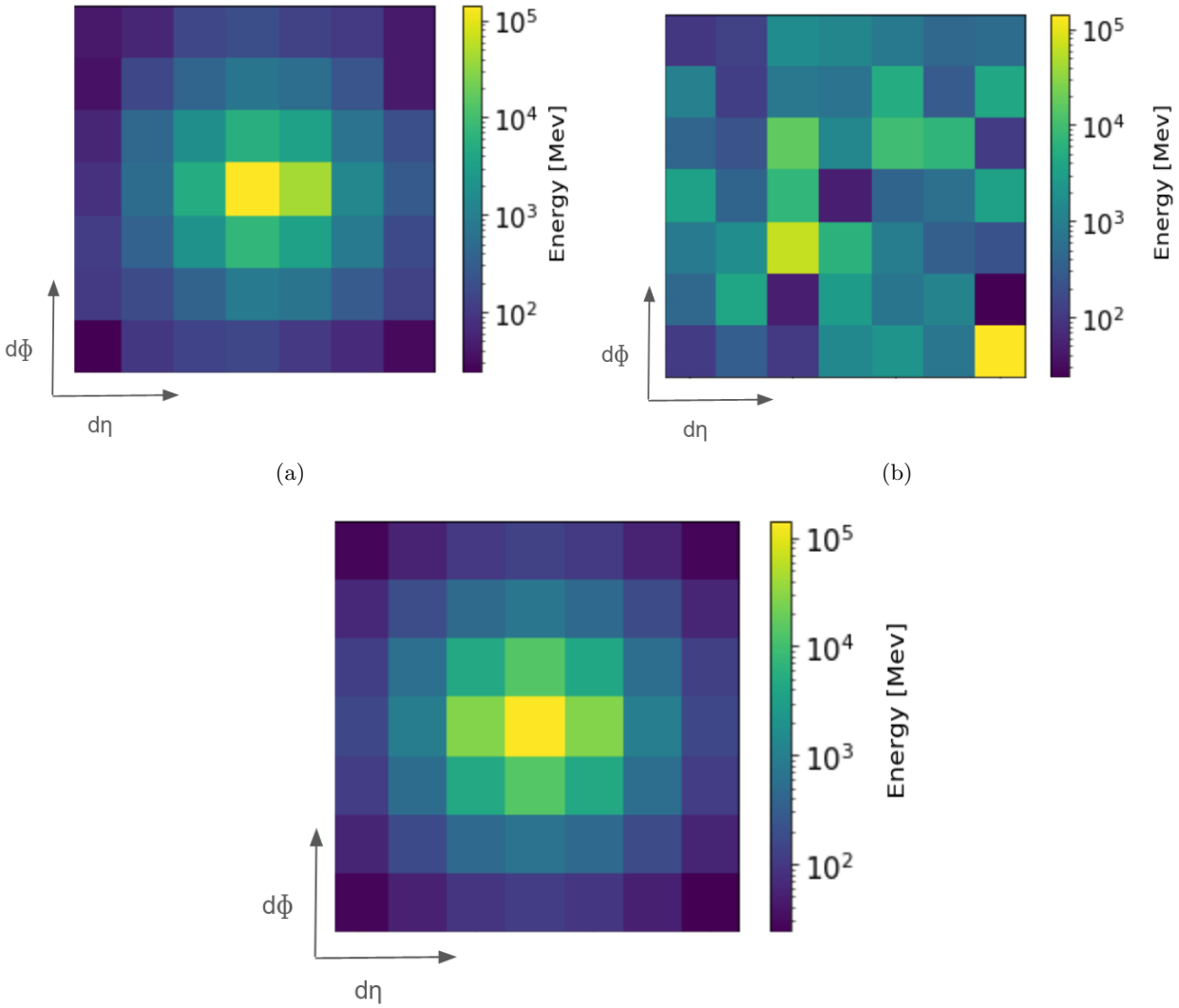
Additionally, we define a metric to measure the similarity between two probability distributions. A metric such as Jensen–Shannon Divergence (JSD) [205] can be used. It is similar to the KL divergence, but is symmetric. It is defined as $JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$, where P represents the input distribution, Q the reconstructed distribution and $M = \frac{1}{2}(P+Q)$ and $D(Q||M)$ the KL divergence. Similar distributions would have a metric value close to 0. Figure 50 illustrates the evaluation of two sets of hyper-parameters using the defined metrics JSD and SSIM. In both cases, the metrics are computed on the same number of events. This example illustrates the scenario where one set of hyper-parameters (case 1) is automatically discarded due to poor metric values and the other set (case 2) is considered for the next evaluation. Figure 51 shows an example of a Geant4 shower and its two reconstructed versions from case 1 and case 2, where the poor metric values in the later case 1 can be seen in the dissimilarity compared to the Geant4 shower.

For the simulation, only the decoder network is used as a generator by sampling from d dimensional uncorrelated Gaussians where d is the dimension of the learned latent space. In addition to d Gaussians, condition values of the energy and η of the incident particle form the inputs of the generator. The standalone validation code



(a) (b)

Figure 50: SSIM and JSD metrics for two sets of hyperparameters (a) case 1 and (b) case 2. A Good metric value for SSIM is 1 and for JSD is 0. In this case, (a) is automatically discarded from further validations.



(a) (b)

(c)

Figure 51: (a) a Geant4 shower of a photon particle with an energy of 65 GeV and 0.2< $|\eta|$ <0.25 in EMB2 (b) its VAE-Reconstructed shower from the set of hyper-parameters in case 1 in Figure 50 (c) its VAE-Reconstructed shower from the set of hyper-parameters of case 2 in Figure 50.

compares VAE-generated showers to the preprocessed Geant4 samples based on shower observables which can be computed in a standalone way, i.e., they do not need object reconstruction in Athena.

The training results of the model vary depending on the input parameters. The best set of hyperparameters is obtained using a grid search, performed in parallel on multiple GPUs. This hyperparameter scan is performed for each of the models (cells, voxels, and centroids). Since VAEs are trained to minimize a target objective of reconstructing the input data, this can be used as an optimization metric. However, this metric is relevant only if the desired task is to reconstruct the input values of energies per volume spaces or what is similarly performed in image reconstruction of pixel intensities. To strengthen this optimization, the hyperparameter values are chosen by minimizing, in addition to the reconstruction loss, the $\chi 2$ of some key physics observables. The $\chi 2$ value compares the observable distributions of generated showers to Geant4. These observables are the total energy deposited in the calorimeter and the energy per layer. As an example, in a model trained on cells considering only four electromagnetic layers in the barrel region, Figure 52 shows the $\chi 2$ values as a function of the latent space dimension, which varies from 1 to 30. The best value of this dimension is the one for which the $\chi 2$ is optimized for all the five observables. In this case a 20 dimensional latent space represents the optimal value that enables a good modeling for the distributions of the five observables.

Other optimized hyperparameters include the depth of the encoder and decoder, the number of units in each layer, the activation functions, the bias, and kernel initializers, the optimizer, its learning rate, the size of the mini-batches and the weights of the different terms in the loss function.

The cell, voxel, centroid models are implemented in Keras 2.0.8/ 2.0.8/ 2.3.1 [218] using TensorFlow 1.3.0/ 1.15.0/ 2.5.0 [217] as the backend. They trained on an NVIDIA® Titan X Pascal graphics card with a processing power of 3584 cores, each clocked at 1417 MHz. The number of training epochs depends on the two minimization options of the reconstruction loss and the $\chi 2$ values of the total energy and the energy per layer. Hence, the training time depends on the number of epochs, the number of training examples and the loading of the training data into memory.

7.5.2 Validation in the ATLAS Athena framework

In order to assess the quality of the FastCaloVSim simulation in the ATLAS Athena software framework, a new service is implemented for each of the three input levels: cells, voxels and centroids. This service allows us to generate showers from FastCaloVSim and forward them to the ATLAS Athena software. The energy of the particle and η (only used for the case of voxels and centroids) are used as conditional parameters for the inference. The output of the FastCaloVSim service is the energy generated for each of the volume spaces. For the voxels or centroids each energy value is considered as a hit and then assigned to a cell

FastCaloVSim models are developed in Python, and Athena is a C++ based environment. Therefore, these models are converted to a format ready to use in C++. This conversion is handled by the Lightweight Trained Neural Network (LWTNN) [184] library. LWTNN loads and computes the graph of the trained VAE. The decoder weights and architecture are saved into a JSON file to be used by the newly implemented FastCaloVSim service.

All simulated energy deposits are assigned to cells of the ATLAS calorimeter and processed using the ATLAS nominal reconstruction algorithms [185]. This allows the comparison of high-level variables used in physics analyses. The ATLAS geometry used for the energy assignment is the simplified geometry, an approximation of the real cell structure. This approximation does not include the accordion structure of the ATLAS ECAL layers.

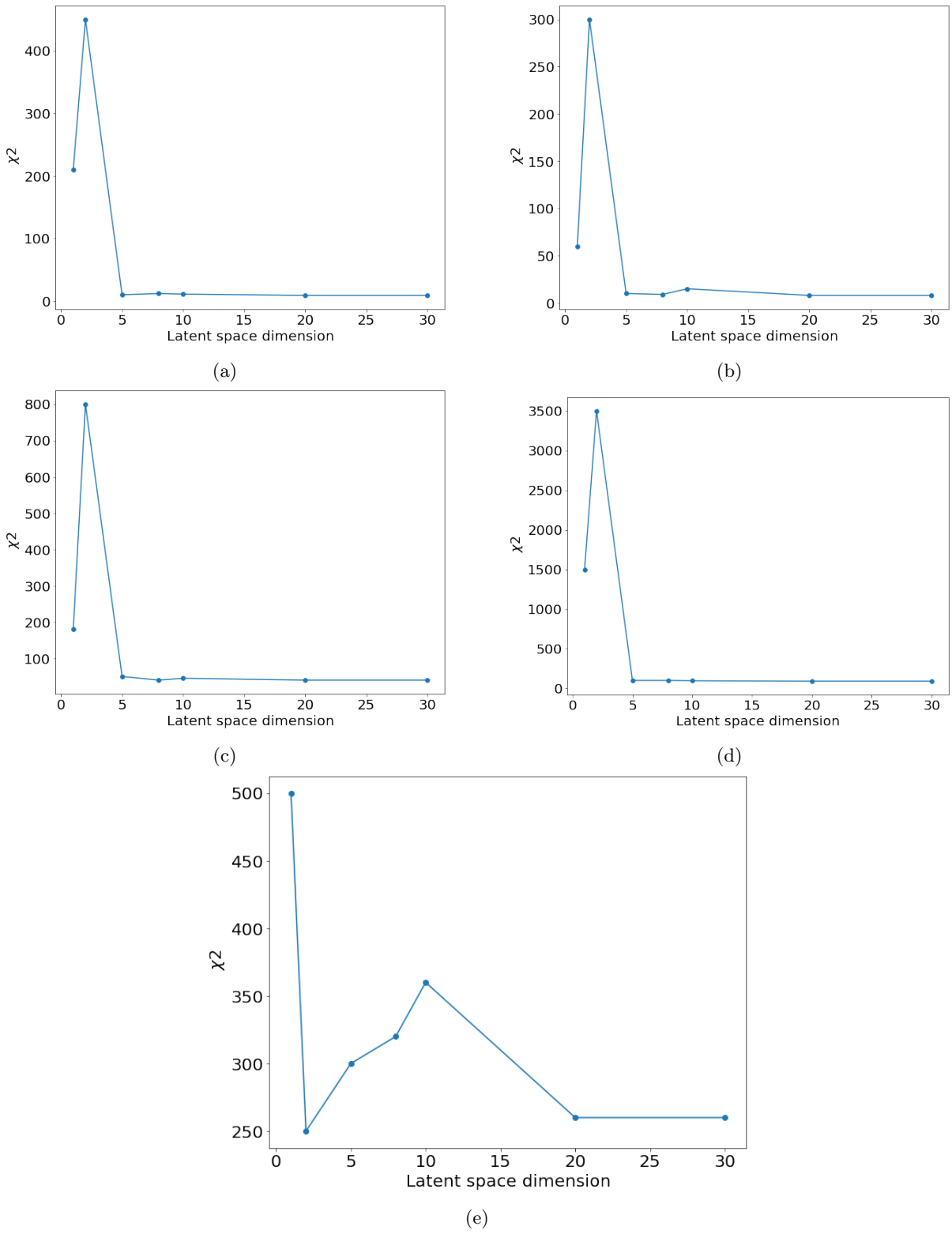


Figure 52: $\chi 2$ values as function of the latent space dimension (1, 2, 5, 8, 10, 20, 30) for shower observables distributions (a) total energy, (b) energy in the presampler layer, (c) energy in EMB1, (d) energy in EMB2, (e) energy in EMB3.

8 Cell-level FastCaloVSim

This chapter presents the details of the VAE model designed to learn the showering process of photons. The first iteration of training of this model is performed on cell energies and later evaluated on cell energy ratios. In parallel, training on ratios is augmented by incorporating physics knowledge in the loss definition. The VAE performance is systematically compared to another generative model: the Generative Adversarial Network (GAN). This comparison is the result of a research collaboration, summarized in two publications [123] and [214].

8.1 Learning to Generate Photon Showers From Cell Energies

Learning the data distributions of photon showers with a generative model is a fundamental challenge for an architectural design. This challenge is related to the complexity of the physics processes to model and the different dependencies between the abstraction levels from a cell energy to a layer energy to a total energy. The challenge is also compounded by the nature of the events themselves, an early shower starting to deposit energy in EMB1 is different from a late shower which starts to deposit energy deeper in the calorimeter. The model also has to generate energy per cell in the correct positions, a cell in the center of the shower has more energy deposited in it than a cell at the edge.

8.1.1 Data Preprocessing and Storage

The cell level photon model is trained on Geant4 samples generated for nine discrete particle energies logarithmically spaced : 1, 2, 4, 8, 16, 32, 65, 132 and 262 GeV and distributed uniformly in a single region of $0.25 < |\eta| < 0.2$. Each sample contains up to 10000 generated events. For these samples, the energy of a photon shower is entirely deposited in the ECAL and only a small amount of leakage is present in the HCAL. Therefore, only ECAL layers in the barrel region are considered: Presampler, EMB1, EMB2 and EMB3.

The goal of a preprocessing is to define a structured input that is easier for the network to learn from. A shower can be described as a 2D image per calorimeter layer, where the dimensions of each image are derived from a window selection using the information of the energy deposited by a typical shower in this layer. EMB2 is the layer in which most of the shower energy is deposited. Choosing a cut of more than 99 % of the energy deposition in this layer results in an image of size of 7×7 in $\eta \times \phi$. The image size of the other layers is defined so that the spread in $\eta \times \phi$ of EMB2 is contained. This results in images of 7×3, 56×3 and 4×7 for the Presampler, EMB1 and EMB3 respectively with a total of 266 cells as shown in Figure 53.

The cell selection procedure is based on an impact cell. The impact cell is defined as the cell in EMB2 closest to the extrapolated position of the primary photon, as illustrated in Figure 53. Cells in the other layers are selected with respect to the impact cell.

The preprocessed Geant4 events used for training and validation performance are converted to Hierarchical Data Format version 5 (HDF5) file format [188]. The structure of the file defines a hierarchy of groups, subgroups, and datasets. Two groups are defined to contain the cell and the incoming particle features: energy, η and ϕ.

8.1.2 Model Design and Training Procedure

Let $p_\phi(x|z,c)$ be the conditional approximation function the VAE learns for shower generation. The c feature is a conditional input representing the log value of the energy of the incoming particle, x is an input shower of dimension d considering l layers of the ATLAS calorimeter, z the latent representation of a shower x, $\tilde{x}$ the reconstructed x shower of dimension d, $p(z)$ the prior distribution, $q_\theta(z,c|x)$ the encoder's learned distribution. The d dimensions of one shower is represented as a concatenated vector of all cells in all l layers. The energy of each cell is normalized to the energy of the incident particle in order to scale the input values in the range [0,1].

In the loss function of the model, two additional terms are added in order to capture the total energy and the energy per layer as shown in Equation 11 and 12 respectively, with N_i the number of cells in the i-th layer of the calorimeter.

$$L_{E_{tot}}(x,\tilde{x}) = \left| \sum_{i=1}^{d} x_i - \sum_{i=1}^{d} \tilde{x}_i \right|. \tag{11}$$

$$L_{E_{L_i}}(x,\tilde{x}) = \left| \frac{\sum_{j=1}^{N_i} x_j}{\sum_{j=1}^{d} x_j} - \frac{\sum_{j=1}^{N_i} \tilde{x}_j}{\sum_{j=1}^{d} \tilde{x}_j} \right|. \tag{12}$$

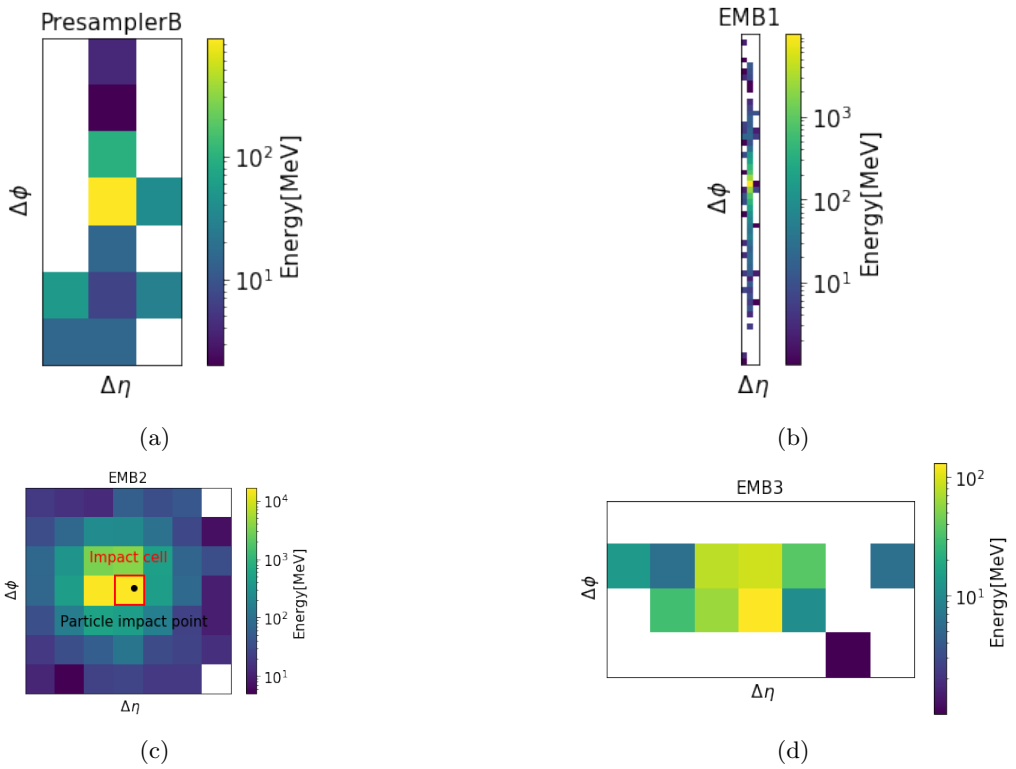


Figure 53: Two-dimensional representation of a Geant4 event per layer for a photon with an energy of 65 GeV in $0.2 < |\eta| < 0.25$: (a) PresamplerB: 7×3, (b) EMB1: 56×3, (c) EMB2: 7×7 and (d) EMB3: 4×7.

The custom loss function is then defined as

$$L_{VAE}(x,\tilde{x}) = w_{reco}L_{Reco} + w_{KL}L_{KL}(q_\theta(z|x)||p(z)) + w_{E_{Tot}}L_{E_{tot}}(x,\tilde{x}) + \sum_i^l w_{E_{L_i}}L_{E_{L_i}}(x,\tilde{x}),$$

where w represents a weight associated to each term. The weighting reflects the relative importance of each term during the optimization. The reconstruction term is defined as

$$L_{Reco} = E_{z\sim q_\theta(z|x)}[\log p_\phi(x|z)]$$

The fully differentiable function is used to train the end-to-end VAE using back-propagation. The following model parameters are a result of a grid search. For computing the update of the weight parameters, the optimizer used is RootMean Square Propagation (RMSProp) [192]. RMSProp belongs to the category of adaptive learning rate methods, where the learning rate is divided by the average root of squared gradients. In this model, we use a learning rate of 10^{-4} and a mini-batch size of 100.

Figure 54 illustrates the detailed VAE architecture. As an input, all the cells are flattened into a single input vector of 266 nodes. The four layers of the encoder learn a reduced dimensionality representation of this 1D vector by performing a non-linear transformation with the ELU activation function. Each of the layers learns abstraction details level of the input data from the previous layer. Unlike autoencoders, VAEs learn by design a continuous latent space that allows for random sampling, which can then be interpreted as generating a new variation of the input showers. This is done by learning two vectors μ and σ, each of 10 dimensions. Then the latent representation z is sampled from the Gaussian distribution parameterized by μ and σ using the reparameterization trick as explained in Chapter 6. This means that the encoder learns a distribution per shower by stochastically generating the z for the training epochs. In other words, the stochasticity means that for the same shower, μ and σ remain the same but the actual encoded value is slightly varying during the learning steps. In an intuitive way, μ represents the center for all the encoded values of a given shower and σ represents the variations from the mean value. The output of the encoder z is then fed into the decoder to reconstruct the 266 input cells. Therefore, the decoder learns the distribution that reconstructs the input

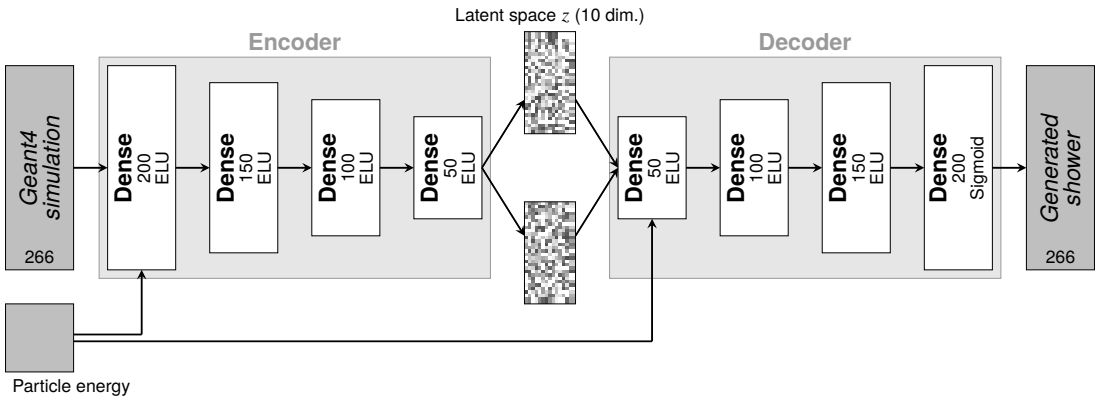


Figure 54: VAE architecture: number of units per layer and number of layers and the activation function per layer are shown for both the encoder and the decoder. The VAE is conditioned on the truth particle energy .

shower. Both networks are jointly trained to reconstruct the 266 cells based on the condition value of the incoming particle energy.

Dense (fully connected) layers are well suited to learn shower properties since they are general purpose networks without any prior assumptions on the patterns to learn. Convolutional layers, on the other hand, use fewer connections between neurons, favoring the modeling of local patterns.

8.1.3 Reconstruction and Generation Performance

Given the task of learning to reconstruct the input shower, it is vital to verify that the model learns to correctly reproduce the input. This shower-to-shower comparison is only done to assess the quality of the reconstruction, not the generation. In other words, it applies when using both the encoder and the decoder. At the generation level, on the other hand, where only the decoder is used, the information about an input shower is not available.

Validating the reconstruction performance uses the end-to-end VAE model on an unseen dataset (test set). Figure 55 illustrates the learning performance of the VAE for two randomly picked events showing cells in EMB2. The energy distribution across cells of a single event is well reproduced by the VAE. This is more visible in the energy pattern where in the first event, for example, most of the energy is shared between the middle cell and its left neighbor and this pattern is reproduced by the VAE. The energy in the center cells is accurately reproduced since they have the highest weight in terms of energy value, and therefore the penalty of the network is higher if this is wrongly reconstructed. The outermost cells contribute with a lower weight in the loss function due to their low energy values.

The event-to-event validation is coupled with another qualitative quantity of reproducing the average shower. It translates to computing an average cell representation across all showers with the same incoming particle energy. The averaging expresses an ensemble feature of an energy pattern which is not visible in the single event validation as shown in Figure 56. On average, the VAE learns to well reproduce this pattern.

Looking at the energy distribution per cell across showers originating from the same incoming particle, as shown in Figure 57, allows tracking of the model's performance on a cell by cell basis.

The previous validation plots were exclusively demonstrating the reconstruction performance of the VAE. The model is also tasked to learn an approximation of uncorrelated Gaussians in the latent space. To check the model's ability to learn this property in addition to reconstructing the input showers, Figure 58 shows the encoded distributions from photons of 65 GeV for training and unseen data compared to Gaussian distributions. Each plot corresponds to one learned dimension in the latent space. The distributions of the training data encodings are shown as a reference to see how much they deviate from unseen events. All ten distributions for both datasets approximate a Gaussian distribution. In addition to one-by-one latent distributions, Figure 59 confirms the lack of correlation between the ten dimensions of the latent space, where the correlation coefficients reported in the figure are low. Learning these uncorrelated Gaussian distributions allows us after training and validation to use the decoder as a generator of new showers. A ten dimensional z for each shower is sampled from ten uncorrelated Gaussian distributions conditioned on the truth energy c, i.e., sample from $p_{x|z,c}$. Distributions from the output of the generator are compared to the ones from the full detector simulation. This comparison is based on distributions of energies and shower shapes, which are used during event reconstruction and particle identification.

The VAE in [123] is compared to a GAN model composed of a generator and a discriminator. The detailed

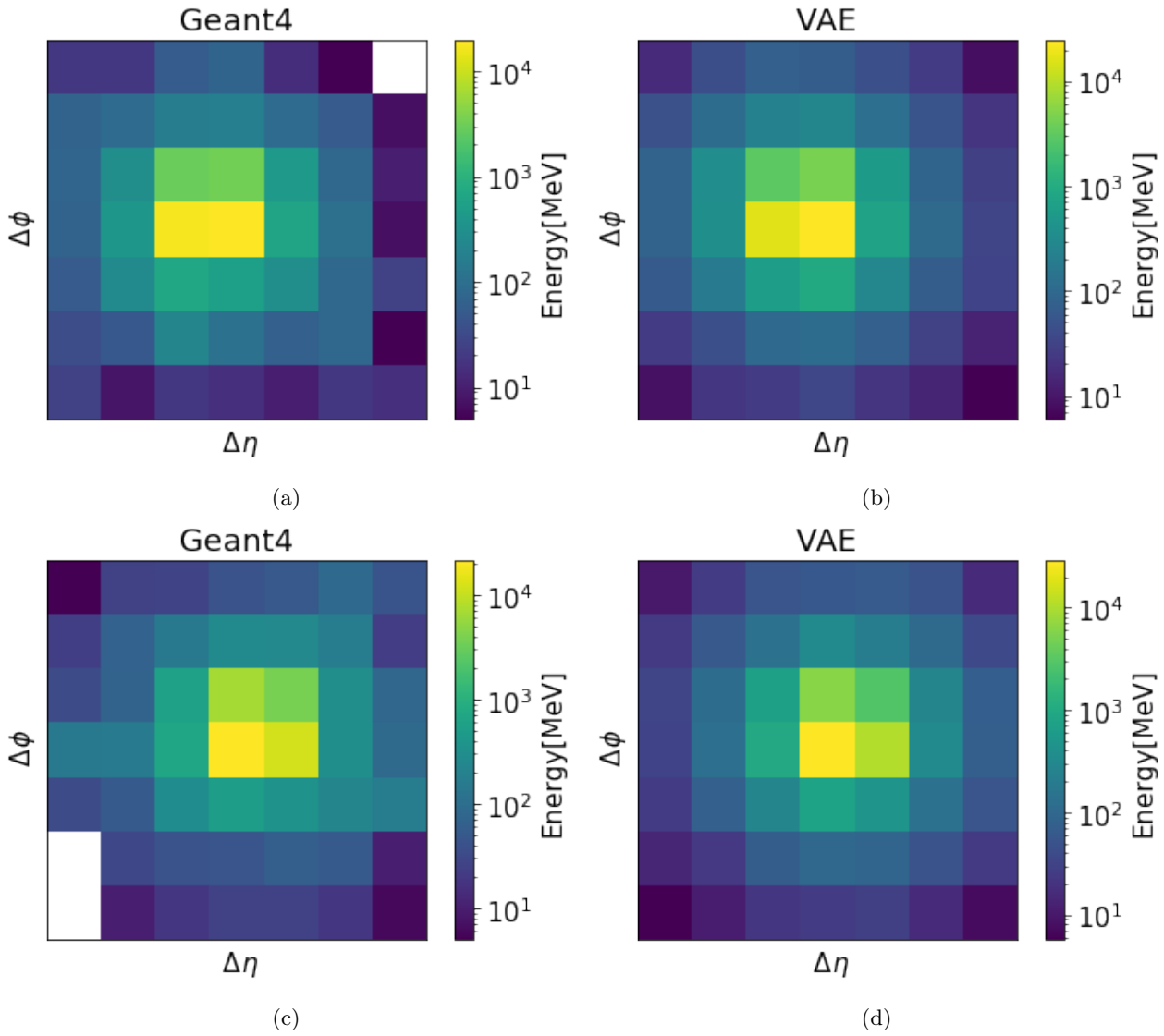


Figure 55: Reconstruction performance of Geant4 events of a photon of 65 GeV in $0.2 < |\eta| < 0.25$: (a), (c) Geant4 events and (b), (d) VAE events.

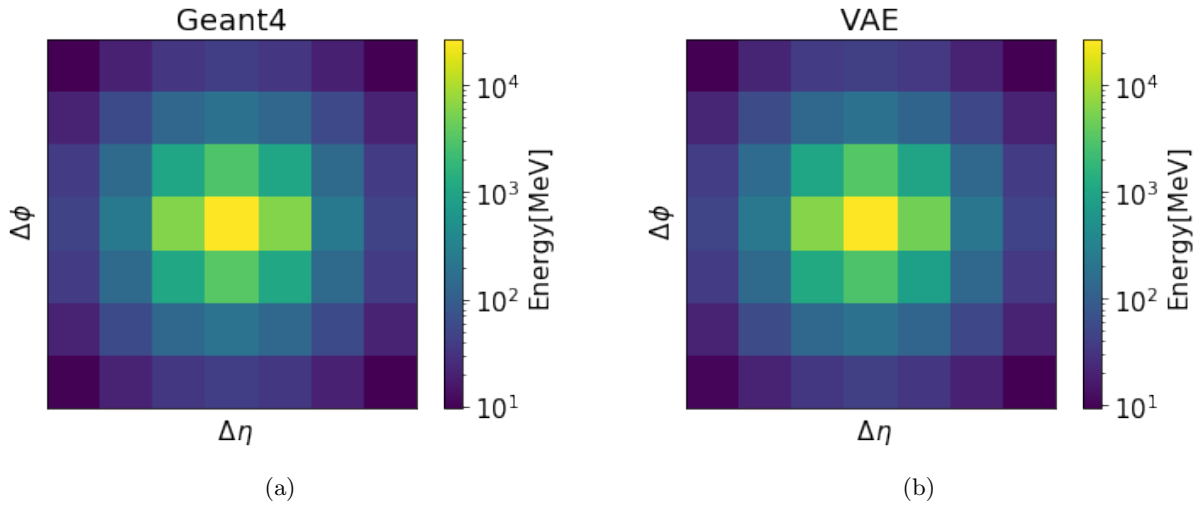


Figure 56: Reconstruction performance of averaging over all Geant4 events of photons of 65 GeV in $0.2 < |\eta| < 0.25$: (a) Geant4 and (b) VAE.

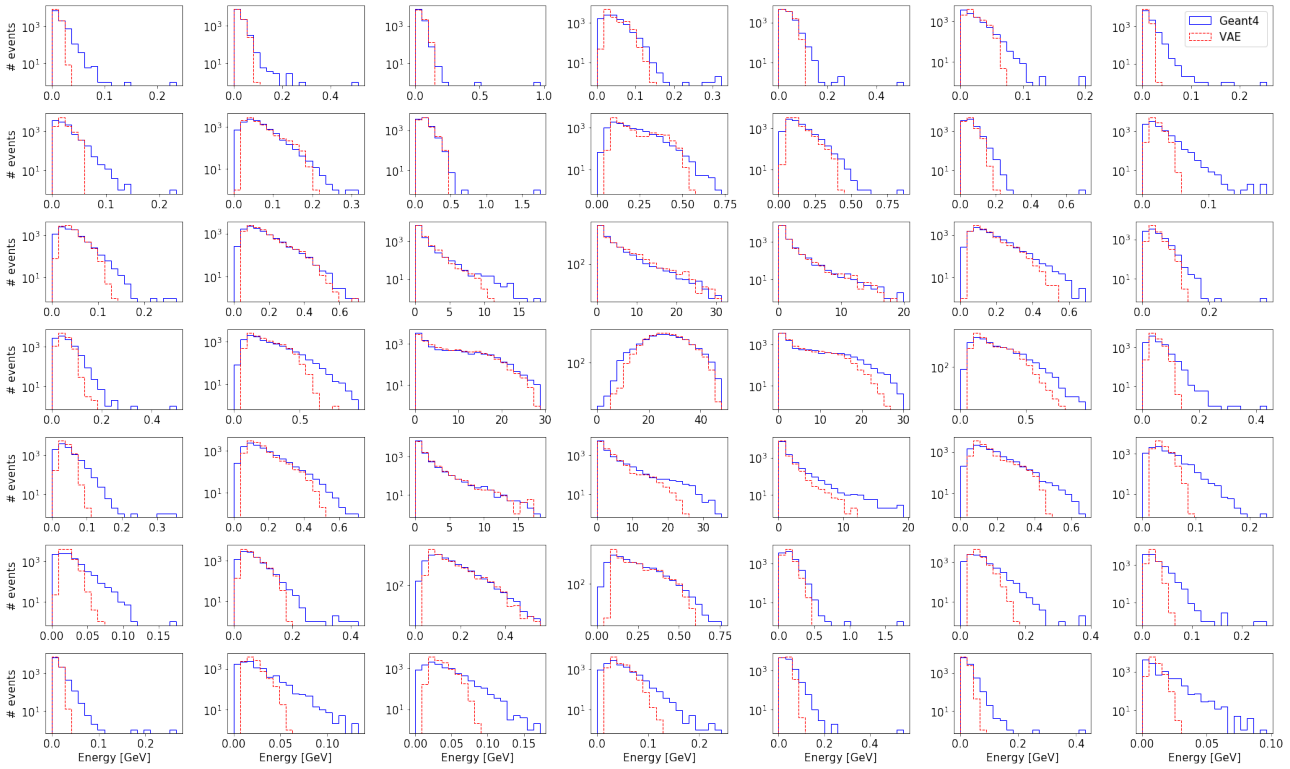


Figure 57: Generation performance on the 7×7 cells in EMB2 for randomly generated photons with an energy of 65 GeV in 0.2< $|\eta|$ <0.25. The full detector simulation (solid blue line) is compared to VAE (solid red line).

architecture of the GAN model is illustrated in Figure 60. The GAN is conditioned on the energy of the incident particle and also on the alignments of the calorimeter cells in η and ϕ. Using the cell selection procedure discussed in Section 8.1, two alignments can be seen in the back layer and four alignments for the Presampler and front layer with respect to the impact cell in the middle layer. Figure 61 illustrates these alignments which are found to impact the performance of the GAN model.

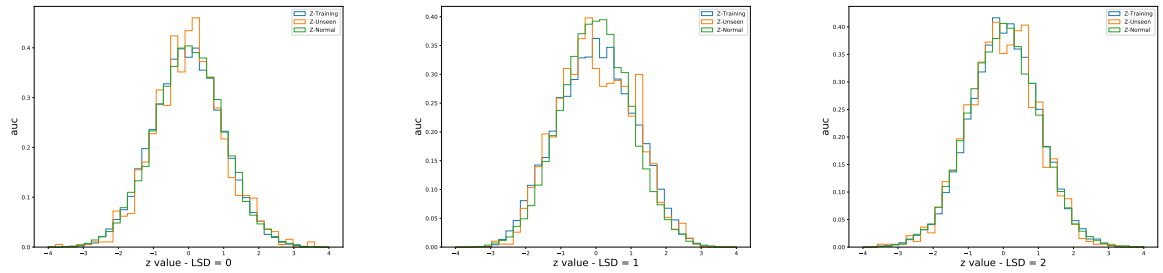
The first quality assessment of generation, consists of looking at the energy distributions per layer, where for each shower the sum of the discretized cell energy deposits per layer is computed. Figure 62 shows this quantity for 65 GeV photons in $0.20 < |\eta| < 0.25$. The VAE reproduces well the bulk of the energy per layer. This comes from the fact that the bulk contains most of the energy deposition, and therefore the attention of the network focuses on this part. The agreement on the tails on the other hand is lower, and this explains a small correlation between energy deposition per layer compared to Geant4.

The validation includes shower shape variables and correlations. Figure 63 shows the average deposited energy in the cells as function of the distance in η and ϕ from the impact point of the particles for photons with an energy of 65 GeV in $0.20 < |\eta| < 0.25$. $\Delta\eta$ and $\Delta\phi$ distances are computed from the center of the cell. These plots provide an insight into what underlying features the model is learning, knowing that it is not explicitly trained on these quantities.

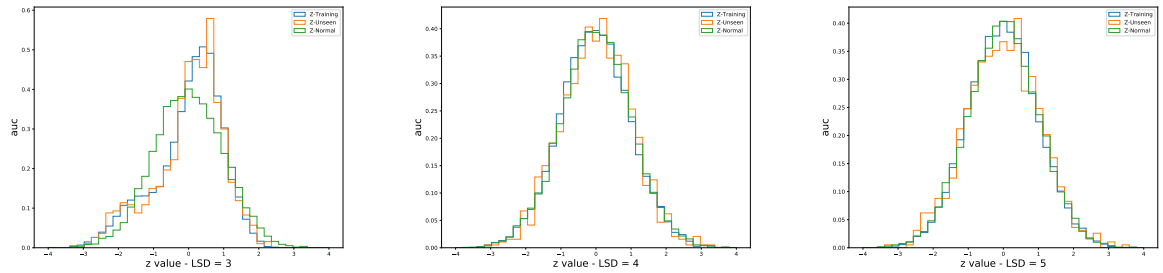
Learning the total energy deposited by a shower is an important feature. It describes the learning of the underlying distributions, such as the energy deposition per calorimeter layer. It is computed by summing up all the energy values from all the layers. Figure 64 shows the total energy of photons of 65 GeV energy in $0.20 < |\eta| < 0.25$. The VAE reproduces better the mean value in contrary to the spread. Overestimating the spread, is caused by the mismodeling of the tails per layer, shown in Figure 62. Despite using a term in the loss to encourage the model to better produce the total energy as shown in Equation 11, this quantity remains a major challenge for the learning process. Moreover, optimizing an n dimensional function is not trivial. In fact, this is known in ML as Multi-Objective Optimization (MOO), where the optimization is a trade-off between different objectives which can also be conflicting objectives. Finding a single solution which optimizes all the terms simultaneously is not as straightforward as for a single term objective function.

8.2 Learning to Generate Photon Showers From Cell Energy Ratios

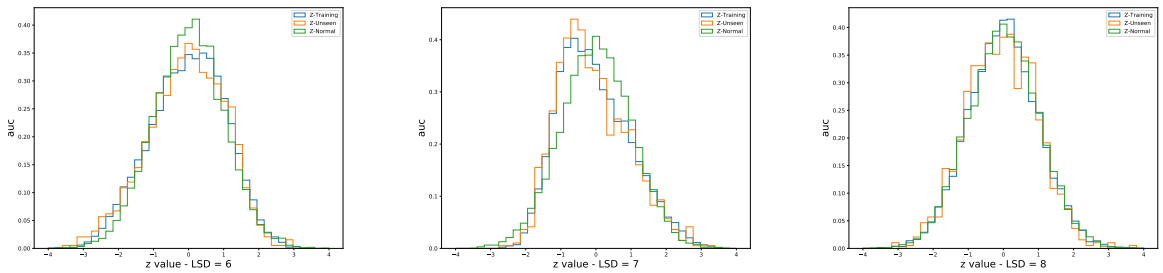
To overcome the limitation of modeling the total energy of the VAE at cell level and to further improve the quality of the generation, the idea consists of re-optimizing not only the model parameters but also the



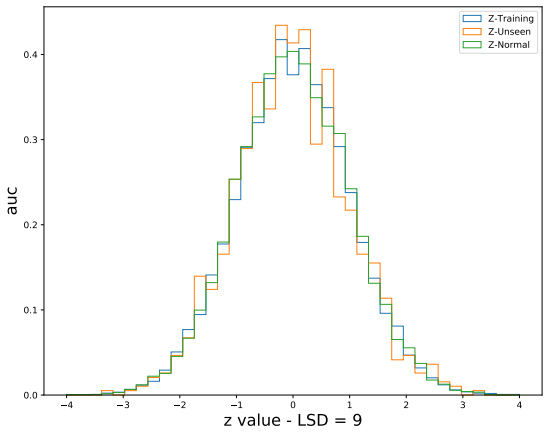
(a) LSD 0 (b) LSD 1 (c) LSD 2



(d) LSD 3 (e) LSD 4 (f) LSD 5

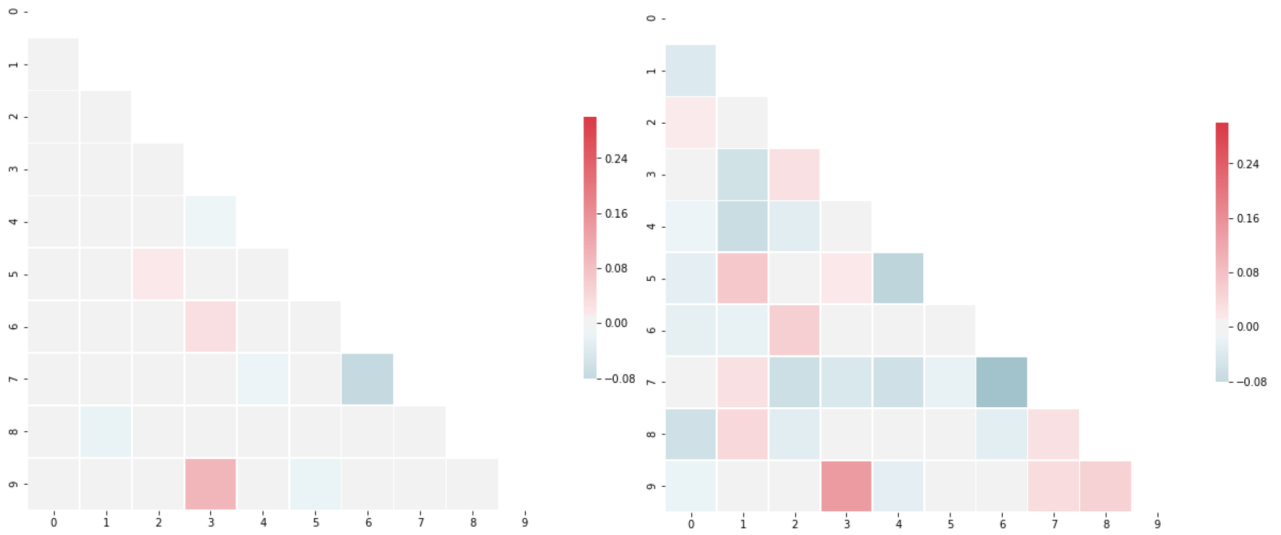


(g) LSD 6 (h) LSD 7 (i) LSD 8



(j) LSD 9

Figure 58: z distributions in each of the latent space dimensions. The distributions from the training set (blue line) are compared to the distributions from the test set (orange line) and normal distributions (green line).



(a) (b)

Figure 59: Correlation coefficients between the ten dimensions of the latent space of the (a) training set and (b) the test (unseen) set. The coefficients are color coded and their exact values are shown in the colorbar.

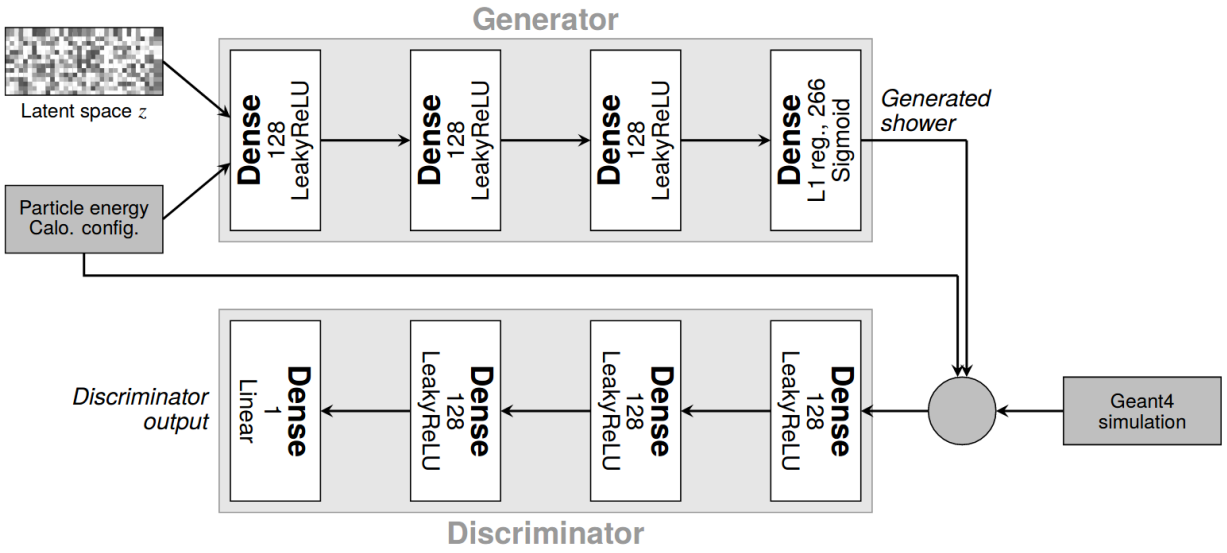


Figure 60: Schematic representation of the architecture of the GAN used in [123]. Composed of a generator and a discriminator, it takes as input 300 random numbers drawn from the latent space distribution. It is conditioned on the input particles' energy and the alignments of the calorimeter cells, represented in Figure 61. The discriminator compares the generated showers from the generator to Geant4 showers. The number of units per layer and number of layers and the activation function per layer are shown for both the generator and the discriminator.

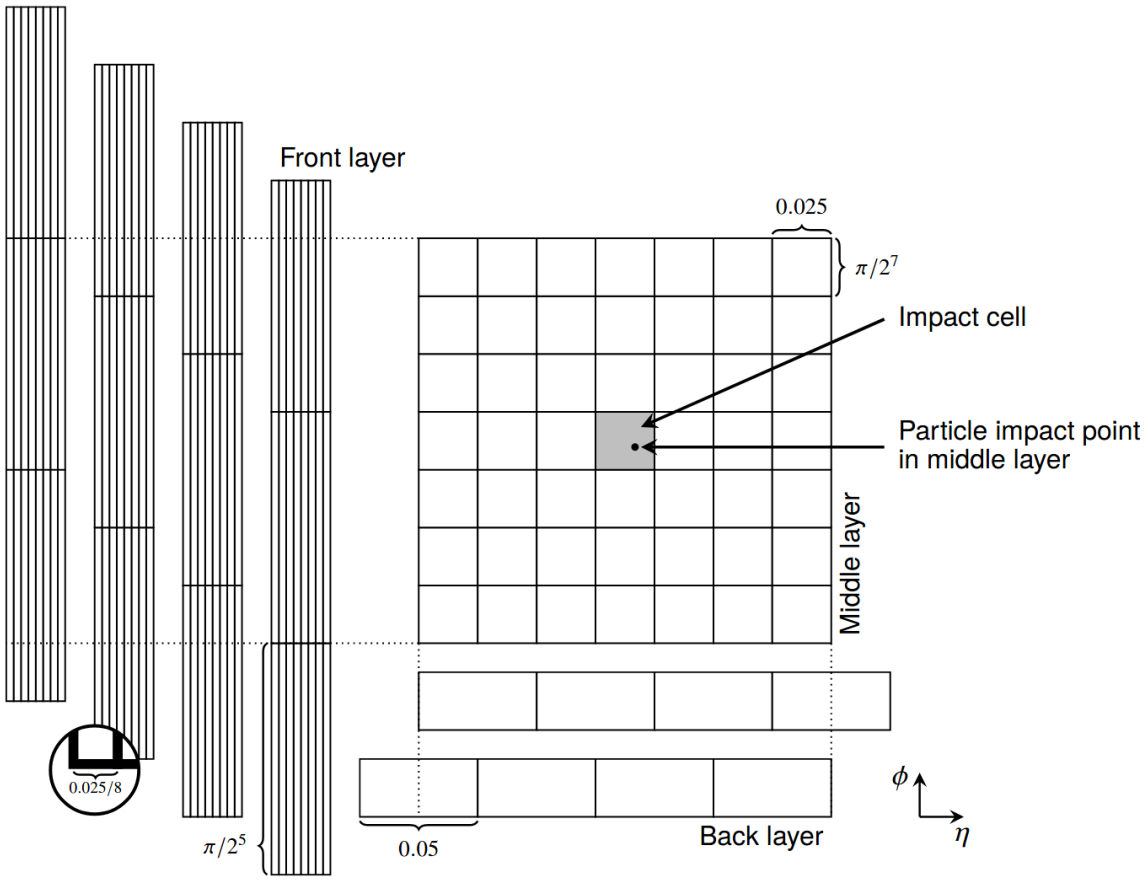


Figure 61: Illustration of possible alignments in ϕ for the front layer, (left, showing an 8×3 portion of the 56×3 cell image) and the back layer (bottom, showing a 4×1 portion of the 4×7 cell image) with respect to the middle layer(center, showing the full 7×7 image). The front (back) layer are visualized to the left (bottom) of the middle layer to illustrate the alignments in $\phi(\eta)$, but are actually one behind another in the third dimension.

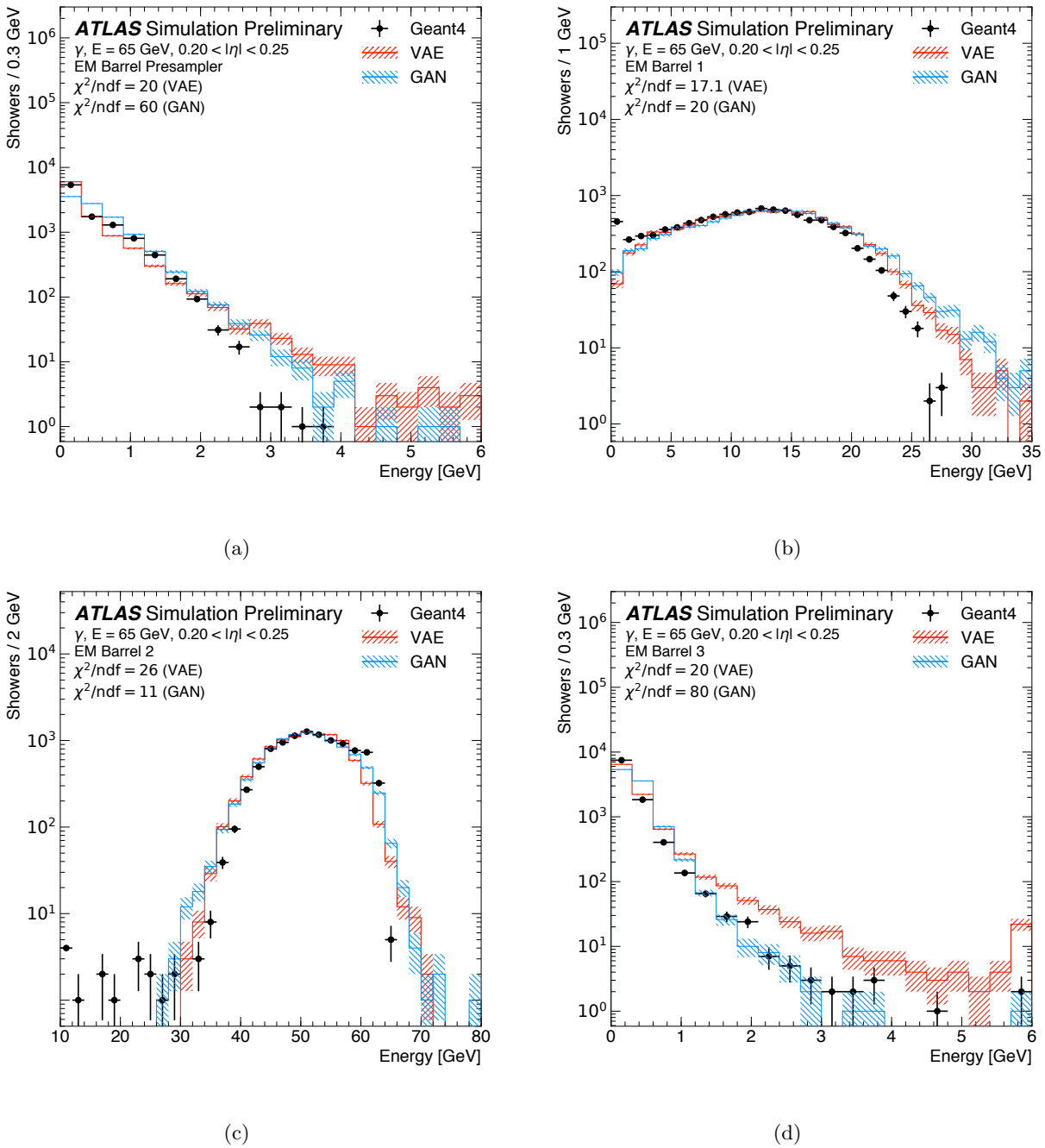
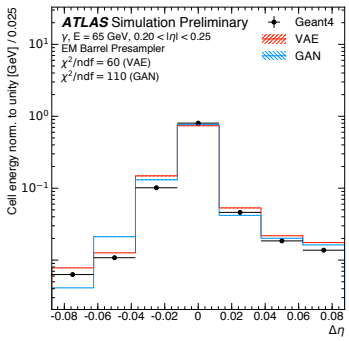
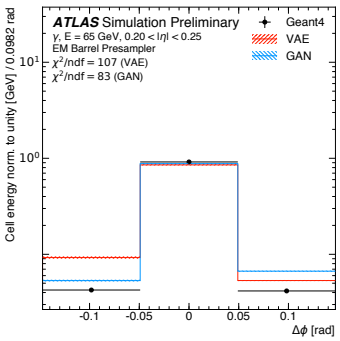


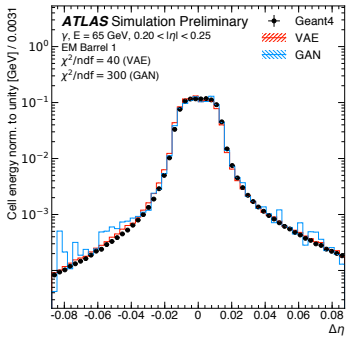
Figure 62: Energy deposited in the individual calorimeter layers (a) Presampler, (b) EMB1, (c) EMB2 and (d) EMB3 for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The error bars and the hatched bands indicate the statistical uncertainty of the reference data and the synthesized samples, respectively. The underflow and overflow are included in the first and last bin of each distribution, respectively.



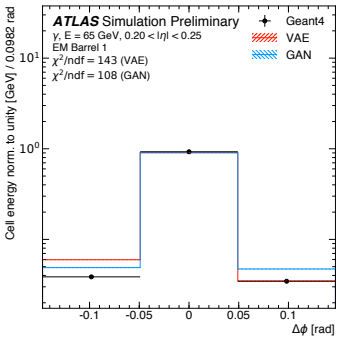
(a)



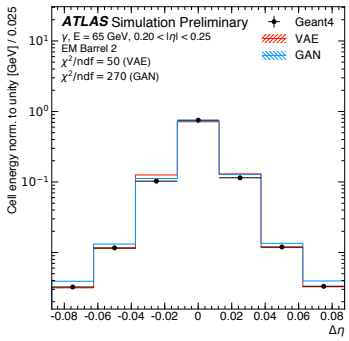
(b)



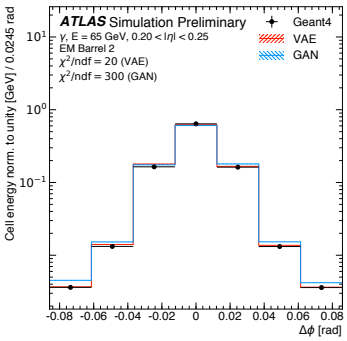
(c)



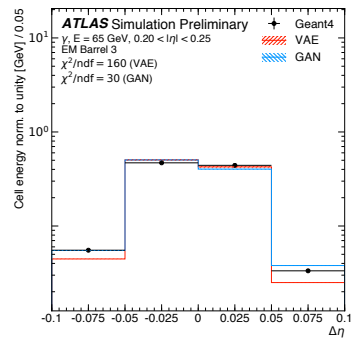
(d)



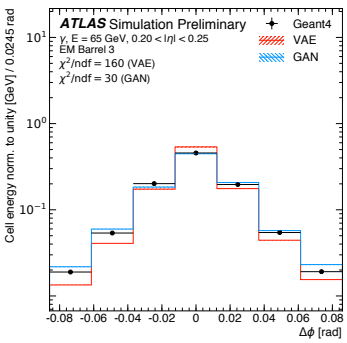
(e)



(f)



(g)



(h)

Figure 63: Average energy deposition in the cells of the individual calorimeter layers (a, b) Presampler, (c, d) EMB1, (e, f) EMB2, and (g, h) EMB3, as a function of the distance in η and ϕ from the impact point of the particles for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The chosen bin widths correspond to the cell widths in each of the layers. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars and the hatched bands indicate the statistical uncertainty of the reference data and the synthesized samples, respectively. The underflow and overflow is included in the first and last bin of each distribution, respectively.

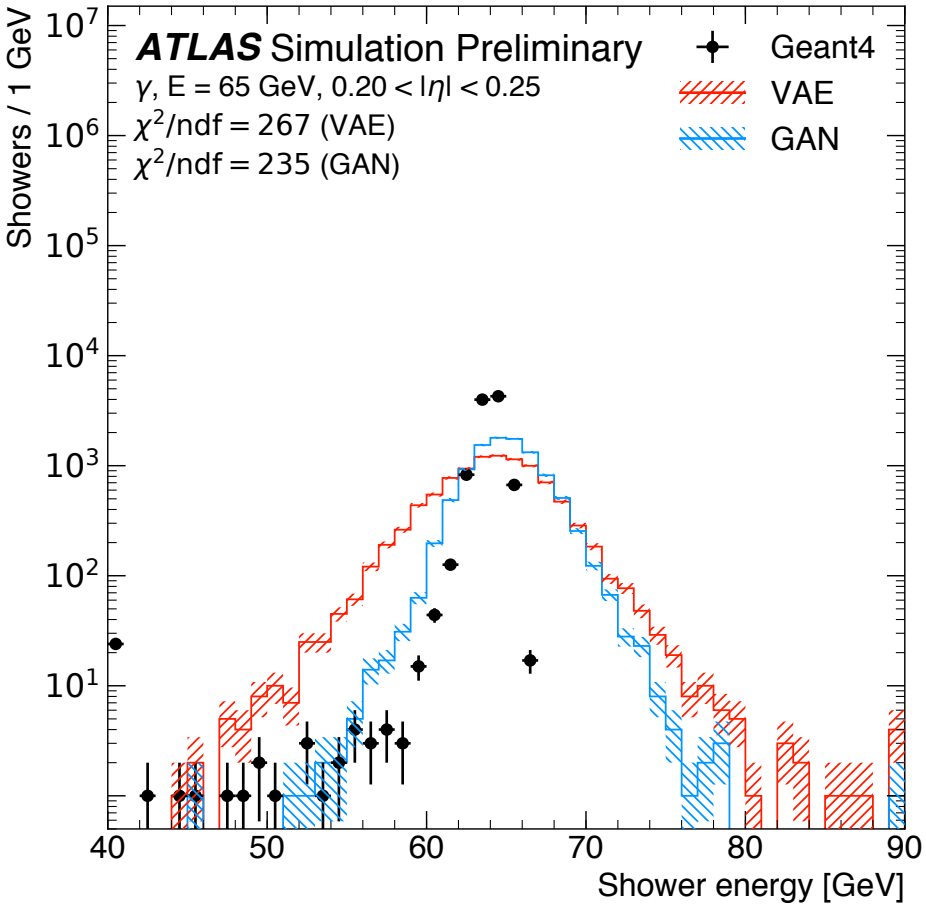


Figure 64: Total energy deposition for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars and the hatched bands indicate the statistical uncertainty of the reference data and the synthesized samples, respectively. The underflow and overflow is included in the first and last bin of each distribution, respectively.

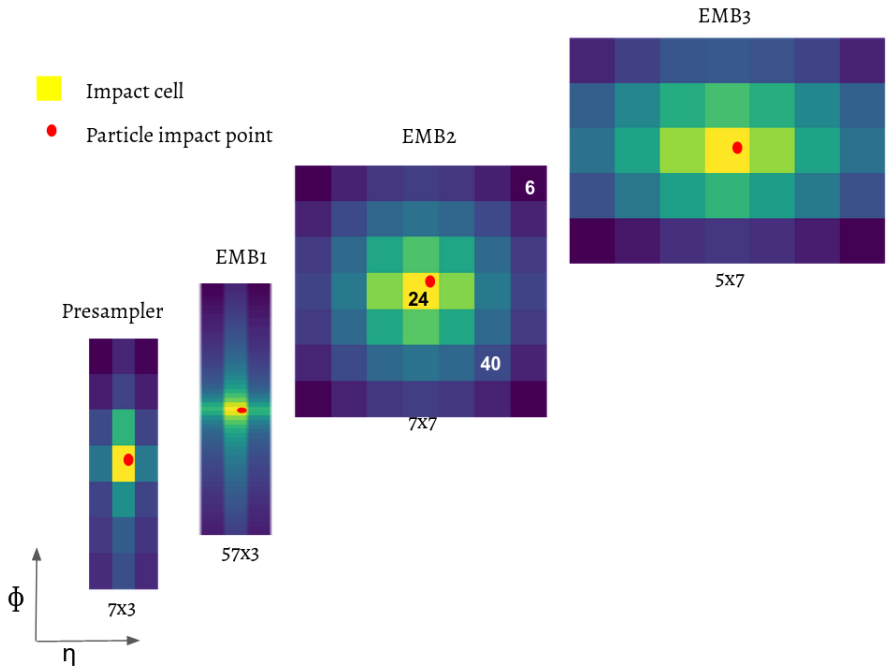


Figure 65: Illustration of two-dimensional representations of centered shower per layer (averaged over many showers) .

preprocessing of the input data. This re-optimization allows us to redefine an augmented objective function with physics knowledge. This section presents the improved VAE version with a set of experimentally motivated choices for designing an accurate model which can reproduce all the shower shape variables. The validation of this model is further evaluated within the ATLAS Athena framework. All the previous configurations of energy range, η slice, number of events and calorimeter layers remain the same.

8.2.1 Data Re-preprocessing

Unlike the preprocessing in Section 8.1.1, the centering of the cells occurs per layer, as opposed to a global impact cell in EMB2. Centering the shower per layer as shown in Figure 65 allows us to simplify the shower representation by treating each layer separately, and therefore the possible alignments of the layers with respect to one another are not taken into account. The new preprocessing is performed by taking the impact position per layer in order to define a center cell in each of the considered ECAL layers. The selected window contains in η and ϕ: 7×3, 57×4, 7×7 and 5×7 cells in the Presampler, EMB1, EMB2 and EMB3 respectively. The difference compared to Section 8.1.1 is seen in EMB1 and EMB3 with an odd number of cells to allow an exact definition of a single center cell. This results in ten additional cells, with a total number of 276 cells.

The re-preprocessing of the Geant4 input as ratio values instead of absolute values allows a better conservation of correlations of energies across layers. In other words, the reparametrization performs a normalization of each cell energy per shower and per layer with respect to the total energy deposited in that layer per event. For a cell j in a layer i, the ratio value is defined as

$$R_{Cell(i,j)} = \frac{E_{Cell(i,j)}}{E_{Layer(i)}} = \frac{E_{Cell(i,j)}}{\sum_{j=1}^{N_j} E_{Cell(i,j)}}.$$

Figure 66 (a-c) represents the distribution of the 276 cell energies for three different events when scaling on the truth energy value, while Figure 66 (d-f) shows the same quantity when applying a scaling per layer. The difference in the spread of the values is a direct consequence of the magnitude shift between truth energy and energy per layer.

8.2.2 Representing and Incorporating Prior Knowledge in the VAE Training

Training on the relative energy to the energy per layer is a way of incorporating the knowledge into the learning process. The new VAE at cell level takes advantage of the existing information of showers by applying a reparametrization while explicitly inducing relevant features to reconstruct, such as the total energy and the

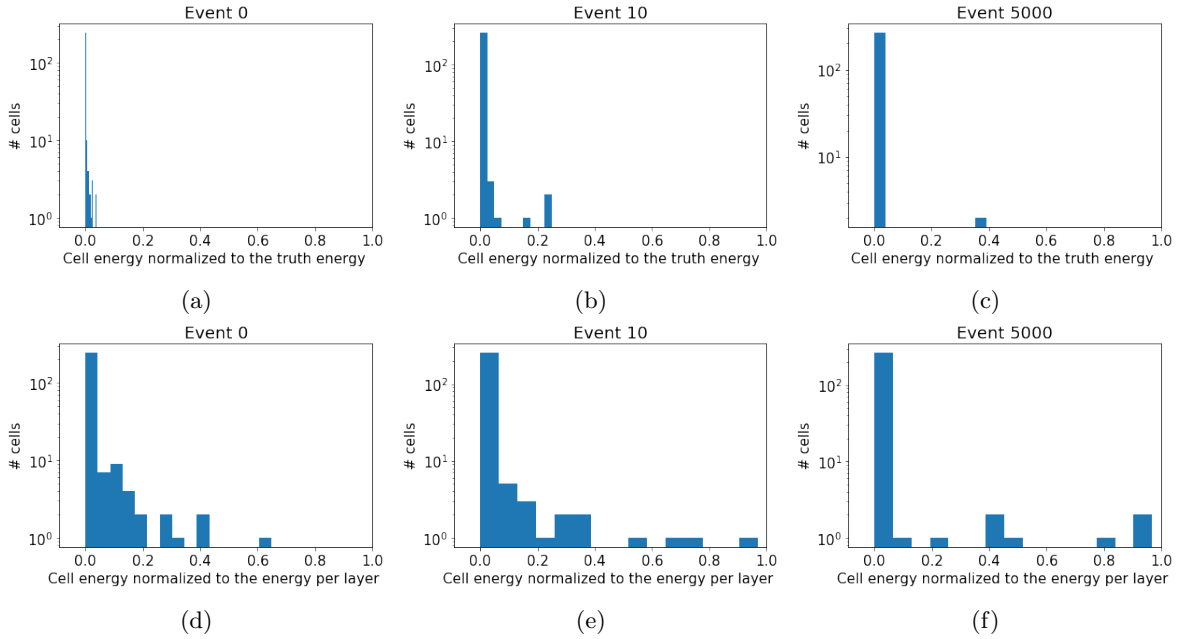


Figure 66: Energy scaling comparison of Geant4 event of photons of 65 GeV energy in $0.2 < |\eta| < 0.25$ for three different events : (a, d) event 0, (b, e) event 10 and (c, f) event 5000.

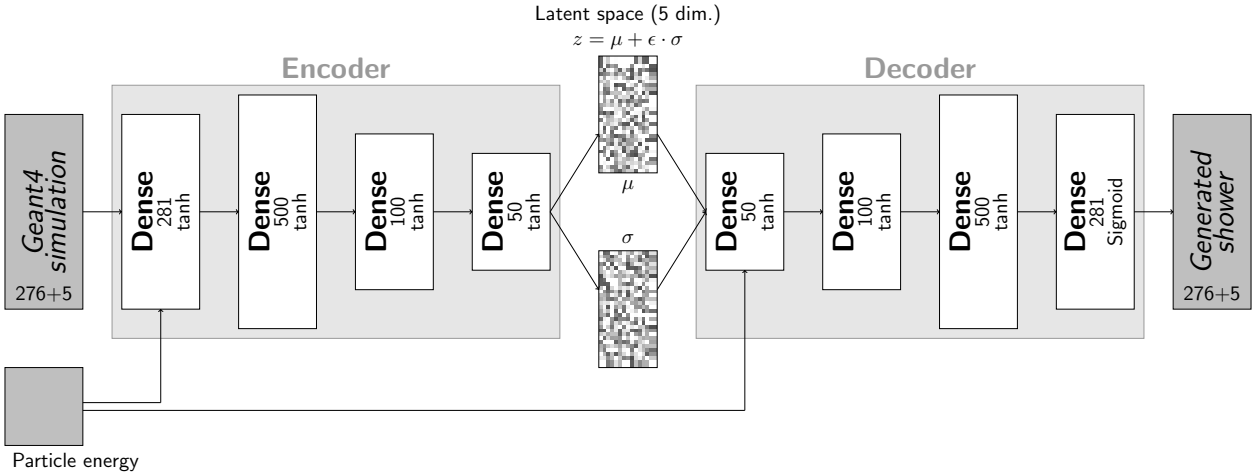
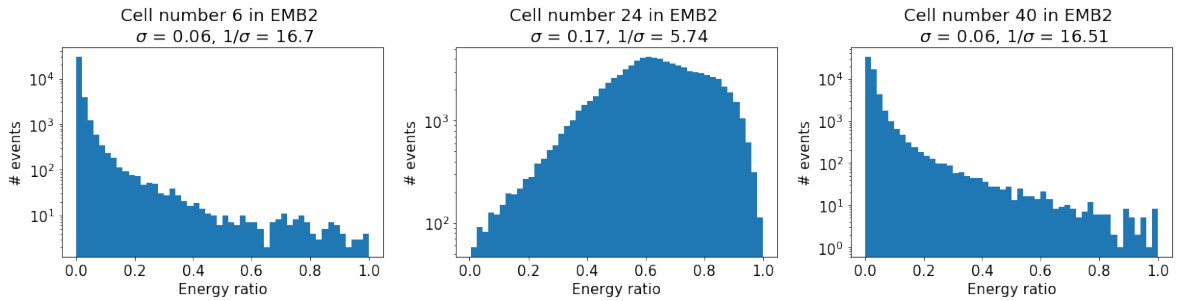


Figure 67: VAE architecture : the number of units per layer and number of layers are shown for both the encoder and the decoder. The VAE is conditioned on the truth particle energy .

energy per layer. The model has three tasks to learn: the energy distribution per layer, the total energy deposited per layer and the total energy deposited in the calorimeter. To allow the network to simultaneously and accurately learn these quantities, the idea consists of reconstructing the 276 cells energy ratios augmented with the total energy and the energy per layer. Since the 276 values are derived from normalizations, the total energy and energy per layer are also normalized to the truth energy and the total energy respectively. This results in a total of 281 values per shower. It translates into 281 nodes in the input and output layers of the model architecture. The additional five fractions allow us to re-normalize the absolute energies of each cell when generating new showers.

The VAE network is designed with four hidden layers for each of the encoder and decoder. Figure 67 shows the architecture where the conditioned model, on the truth energy, learns a mapping between the input nodes of energy ratios and a d dimensional space. The choice of the dimension d of this space is crucial for the performance of the model. A large d will scatter the relevant information producing meaningless samples and a too small d will not contain enough complexity to reproduce the different patterns. After a grid search of hyper-parameters optimization, we choose $d = 5$ as the latent space dimension. Therefore, energy ratios from the 281 sized vectors will be mapped to 5 dimensional vectors.



(a) (b) (c)

Figure 68: Energy ratio (cell energy divided by the energy of the layer) distribution of three cells in EMB2 : cell (a) 6, (b) cell 24 (core cell in EMB2) and (c) cell 40. The cell indices are shown in Figure 65.

The objective function is further optimized compared to the one in Section 8.1.2. It is a weighted sum of only two terms referring to a reconstruction loss between the input and the output layers and the KL divergence, which measures the agreement to a prior set to be a standard normal distribution and the learned latent space distribution. The ability to reconstruct can be interpreted as a quantification of the distance error between the original vector and its reconstructed version. The mean squared error (MSE) is used for this purpose.

Previously, shown in Figure 57, the model is not well reproducing the energy of the cells in the outermost region due to their low values and therefore low penalty on the reconstruction loss. In the improved VAE version, an additional component is defined as part of the optimized reconstruction loss. It is a modification of the loss in order to prioritize some cells, leading to an improved the reconstruction of the shape of the showers within each layer. This is done by deriving a physics weight for each input of the 281 nodes and incorporating it in the reconstruction loss. It is then formulated as

$$L_{Reco}(x, \tilde{x}) = \frac{1}{n} \sum_{i=1}^{n} w_i (x_i - \tilde{x}_i)^2.$$

The full loss function is then

$$L_{\text{VAE}}(x, \tilde{x}) = w_{\text{reco}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} w_i (x_i - \tilde{x}_i)^2}_{\text{Reconstruction loss}} - w_{\text{KL}} \underbrace{\text{KL}(q_\theta(z|x)||p(z))}_{\text{KL loss}}.$$

The standard deviation σ, is one of the most characterizing feature of an underlying distribution that determines its width. For each of the 281 features, the weight w_i is derived from the inverse of the width of the input distribution per feature i over all training events in order to be independent of the truth energy. This computation is based on using a 99.9 % quantile of each distribution and a normalization of the ratio values in range $(0, 1]$. Figure 68 shows an example of the energy ratio distribution for the core cell in EMB2 (cell 24) and two edge cells (cell 6 and cell 40). The three cell locations in EMB2 are annotated in Figure 65. The idea behind using the sigma inverse is to up-weight the contribution of the features with narrow distributions, such as cells 6 and 40. Figure 69 shows the derived weight values for all the 281 nodes in the ECAL with the colors encoding the different layers.

8.2.3 Standalone Generation Performance

All the above optimizations led to improving the total energy distribution as shown in Figure 70, reported as improved-VAE, compared to the model described in Section 8.1 and refereed to as ATL-SOFT-PUB-2018-001. The width of the distribution agrees better with Geant4 for low (such as 4 GeV) and high energies (such as 65 GeV).

The VAE model in [214] is compared to an improved version of the GAN model implementing the gradient penalty Wasserstein GAN (WGAN-GP) [215]. This GAN variant is known to be more stable to train than a vanilla GAN [216]. It consists of replacing the discriminator network with a critic to estimate the Wasserstein-1 distance between the real and generated probability distributions. The GAN in [214] is composed of three networks: a generator, a critic, and an energy critic. The first network takes as input random values from the latent space, the condition values of particles' energy, configurations of cell alignments and the extrapolated

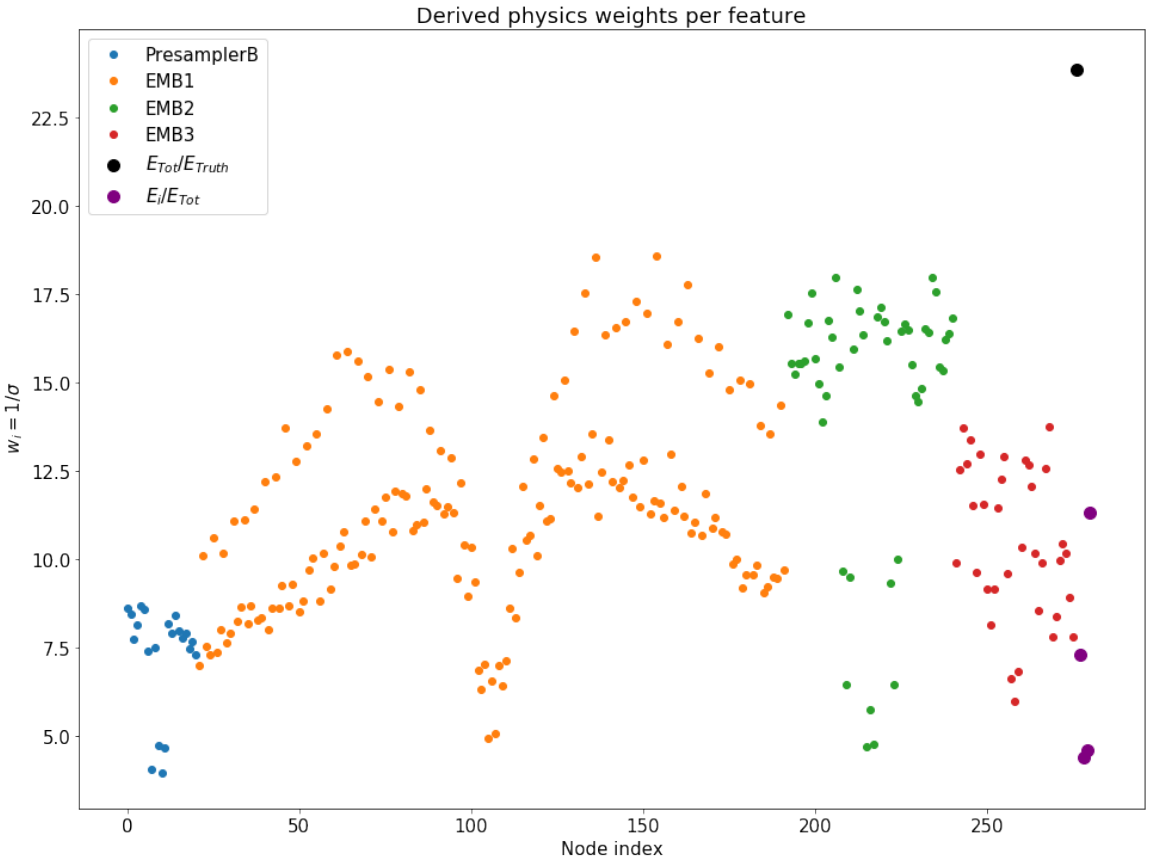
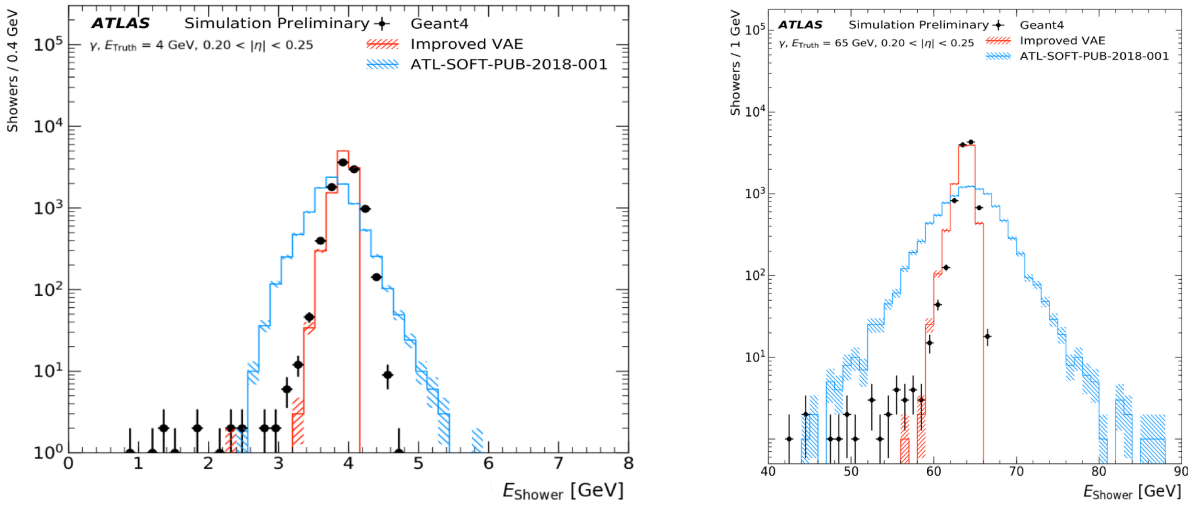


Figure 69: Distribution of the width distribution per cell for the PresamplerB, EMB1, EMB2 and EMB3. The node index represents the order number of all cells in addition to the five fractions cells in the four layers for better visualization.



(a) (b)

Figure 70: Total energy deposition for photons with an energy of (a) 4 GeV and (b) 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of the improved version of the VAE (solid red line) trained on the cell energy ratios and the VAE model in [123] trained on the energies and reported as ATL-SOFT-PUB-2018-001 (solid blue line). The shown error bars and the hatched bands indicate the statistical uncertainty of the reference data and the synthesized samples, respectively. The underflow and overflow is included in the first and last bin of each distribution, respectively.

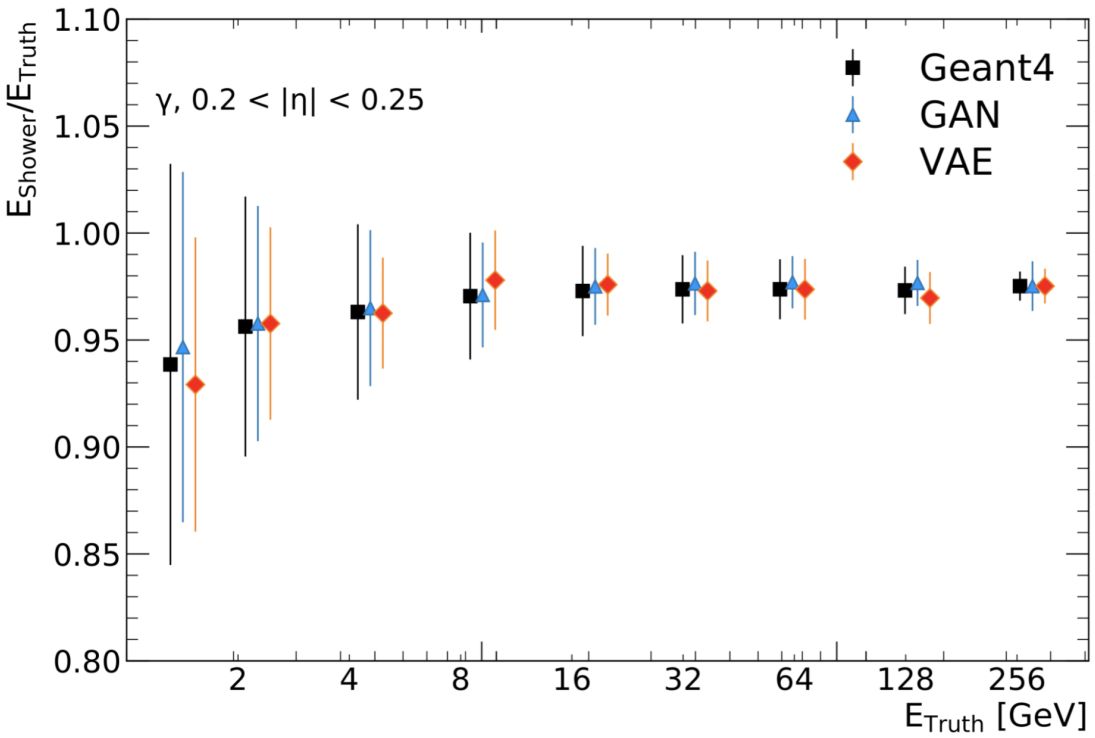


Figure 71: Energy response of the calorimeter as function of the true photon energy for particles in the range $0.20 < |\eta| < 0.25$. The calorimeter response for the full detector simulation (black markers) is shown as reference and compared to the ones of a VAE (red markers) and a GAN (blue markers). The shown error bars indicate the resolution of the simulated energy deposits.

position within the impact cell. The second network compares the showers output by the generator to Geant4 showers. The third network, the energy critic, is added to compare only the energies and therefore improve the total energy modeling.

Figure 71 shows the energy response of the calorimeter as function of the true photon energy. Both the mean and the spread are well recovered, reflecting the preservation of between energies in the layers. A slight underestimation is present for low energies. Another variable shown in Figure 72 allows us to assess the performance of preservation of the shower depth property. The VAE reproduces a good agreement with Geant4.

8.2.4 Generation Performance in the ATLAS Athena Framework

This section is dedicated to show the results of integrating the VAE (and GAN) into Athena as a simulator of the ATLAS detector response. For reference, all the variables are described in Table 5.

Among the first qualities to look at are the reconstructed energy distributions per layer and the total energy. Figure 73 shows these quantities in the four calorimeter layers considered in this study for 65 GeV photons in $0.20 < |\eta| < 0.25$. The VAE is accurately reproducing the energies per layer due to the prepossessing of the Geant4 data, in which the VAE learns the relative energy of the cell to the layer and also the additional fractions. Moreover, in addition to using the physics weights, the VAE models better the tails of the distributions and therefore the correlations between energy depositions per layer, leading to correct modeling of the total energy.

Since EMB2 is the most energetic layer when considering photons particles in $0.2 < |\eta| < 0.25$, looking at the uncalibrated energy (sum of the cells) in a rectangle of $n \times m$ in Figure 74 allows us to further probe the generation performance. The 3×3 rectangle consists of the core cells where most of the energy in EMB2 is deposited, and 7×7 matches the same rectangle selection used for Geant4 samples preprocessing (Section 8.2.1).

The second abstract level, after the core energy in a rectangular cell selection per layer, is the sum of energy in the core cells across all ECAL layers. The core cells as known in the *EGamma* set of variables [182] are 3×3, 15×2, 5×5 and 3×5 for the PresamplerB, EMB1, EMB2 and EMB3 respectively. Along with the ratio of reconstructed energy to the truth energy, these two quantities reported in Figure 75 are also well modeled.

The VAE model is trained to reconstruct the energy fraction per layer as well. Figure 76 probes the agreement with the full detector simulation, where the fraction of energy reconstructed in the EMB1 ($f1$) and EMB3 ($f3$)

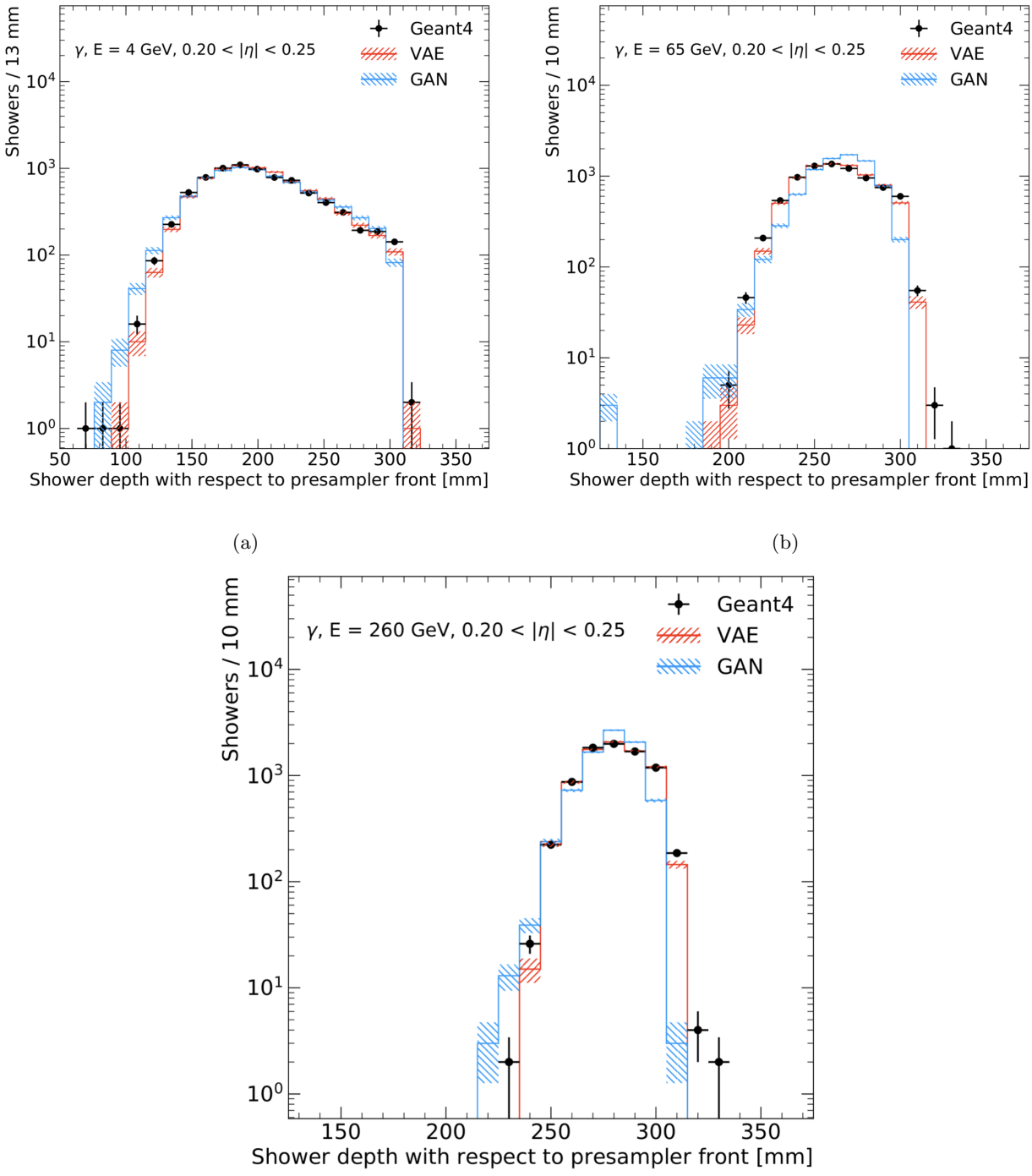


Figure 72: Shower depth with respect to the Presampler front for photons with an energy of 4 GeV (a), 65 GeV (b) and 262 GeV (c) in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars and the hatched bands indicate the statistical uncertainty of the reference data and the synthesized samples, respectively. The underflow and overflow is included in the first and last bin of each distribution, respectively.

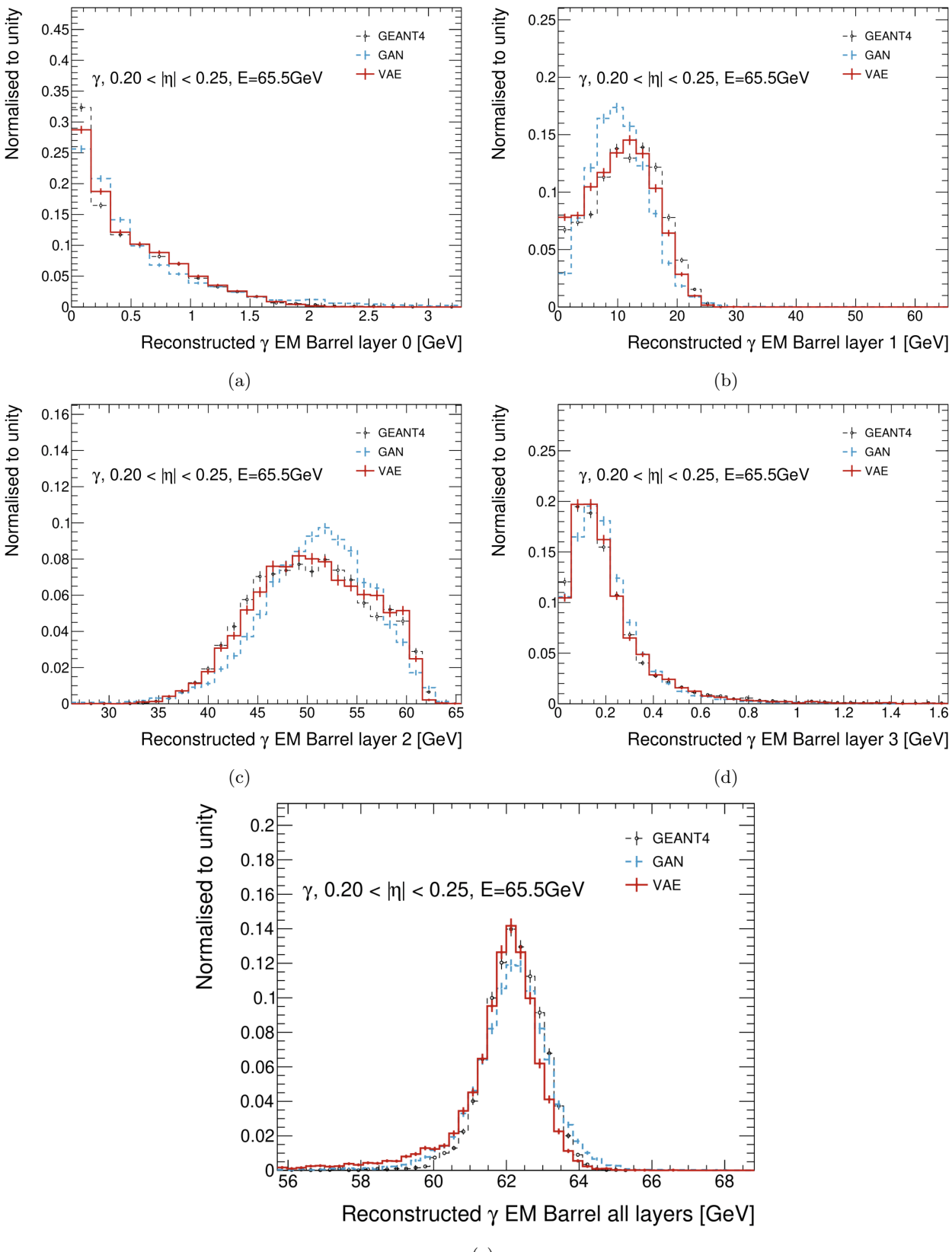
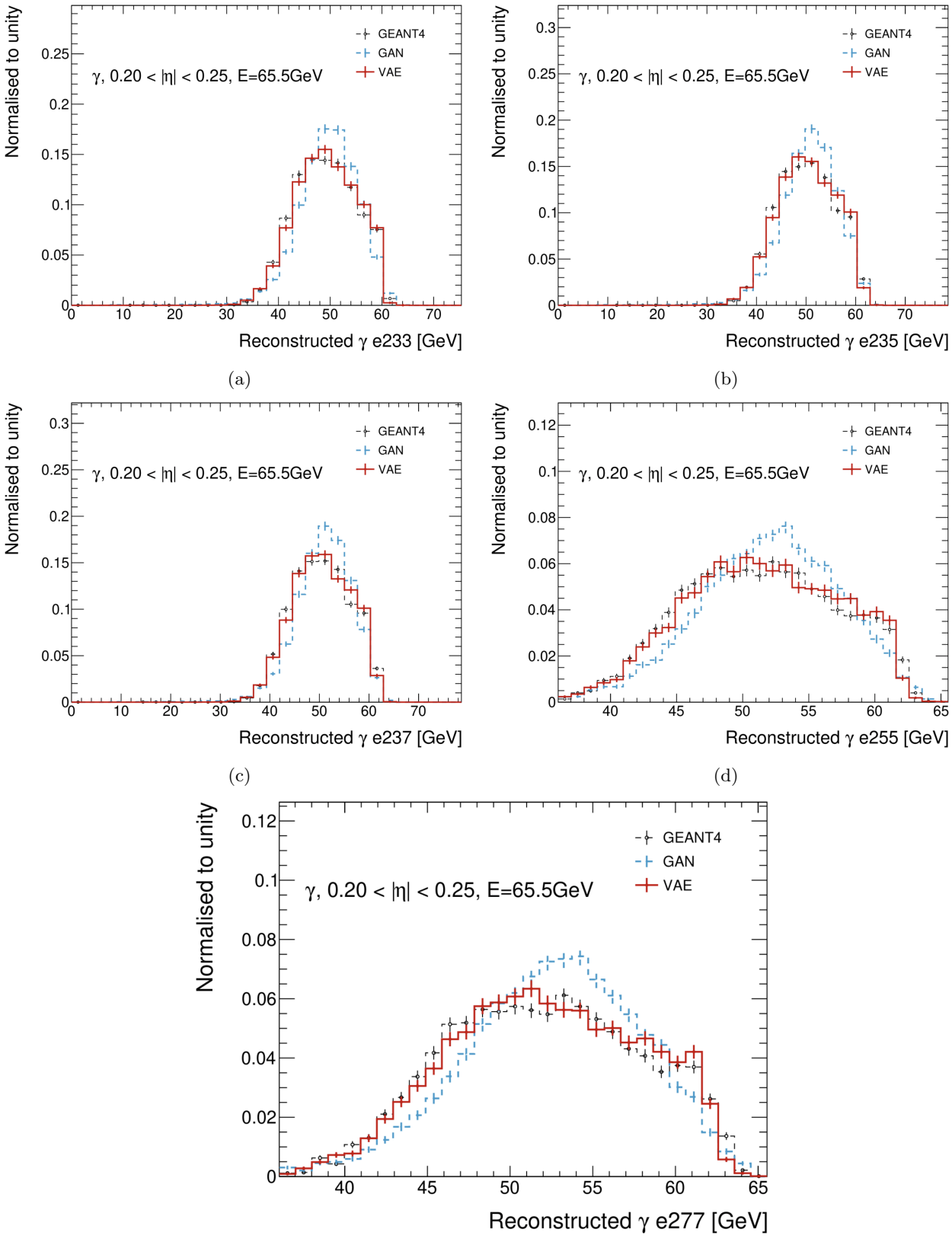
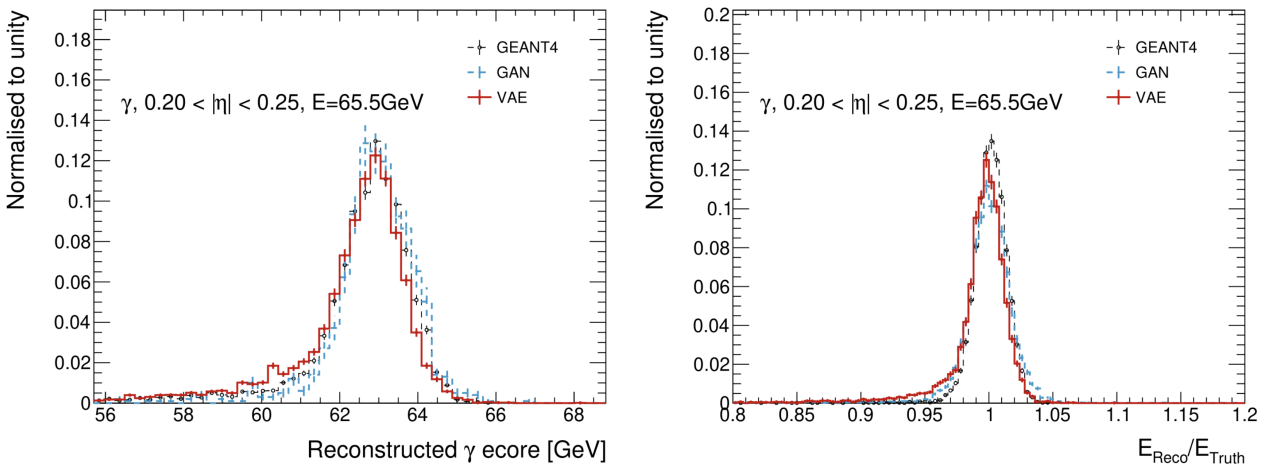


Figure 73: Energy deposited in the individual calorimeter layers Presampler (a), EMB1 (b), EMB2 (c), EMB3 (d) and the total energy (e) for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



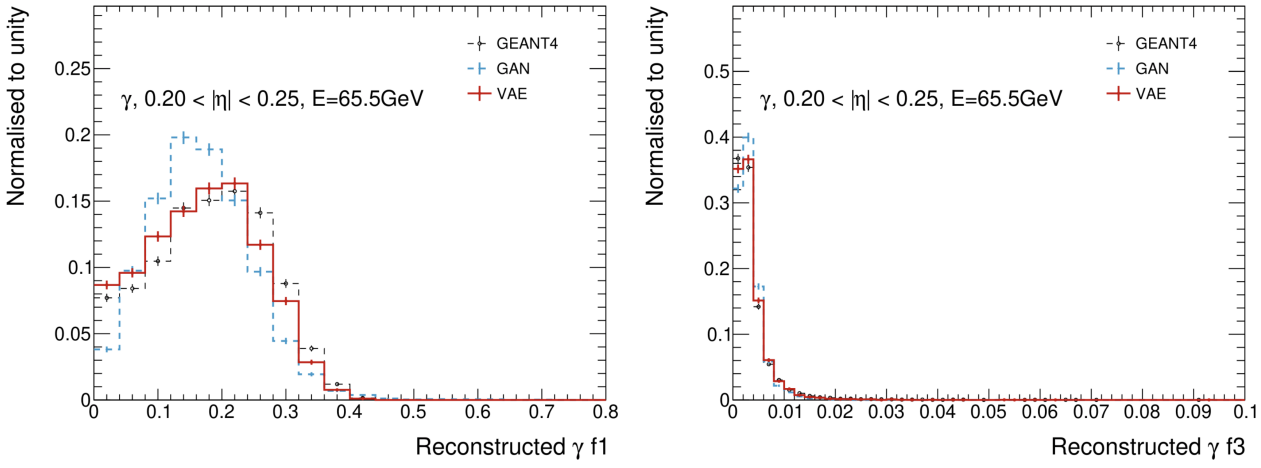
(a) (b) (c) (d) (e)

Figure 74: Energy distributions of e233 (a), e235 (b), e237(c), e255 (d), e277(e) for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



(a) (b)

Figure 75: (a) Reconstructed energy in the core (b) E_{Reco}/E_{Truth}, for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



(a) (b)

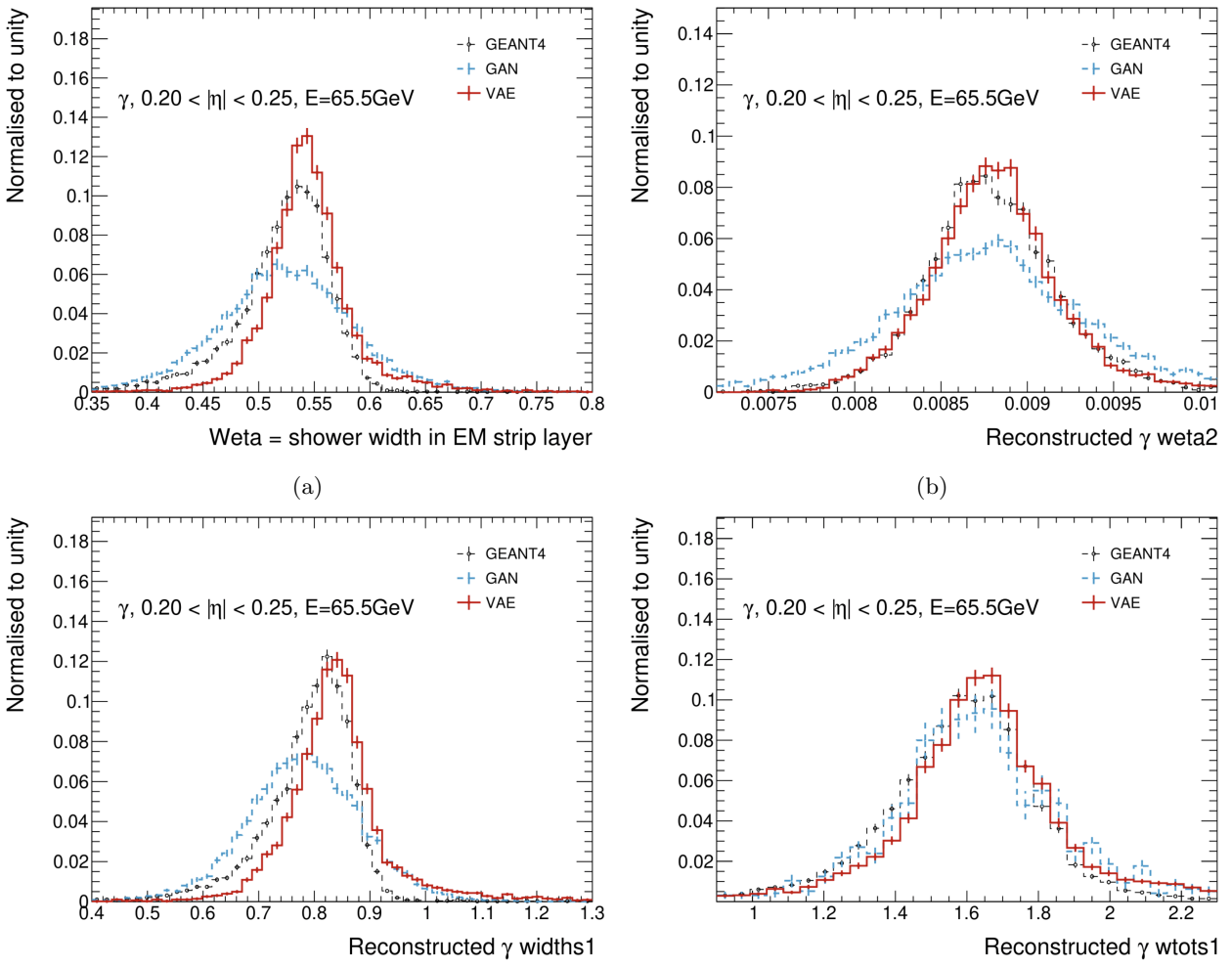
Figure 76: f1 (a) and f3 (b) for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

are shown.

The validation of the performance is not only based on general shower energies and shapes, but also on variables which describe the shower substructure. Figures 77 and 78 summarize the most relevant variables used by ATLAS for particle identification. Overall good agreement is shown for all distributions.

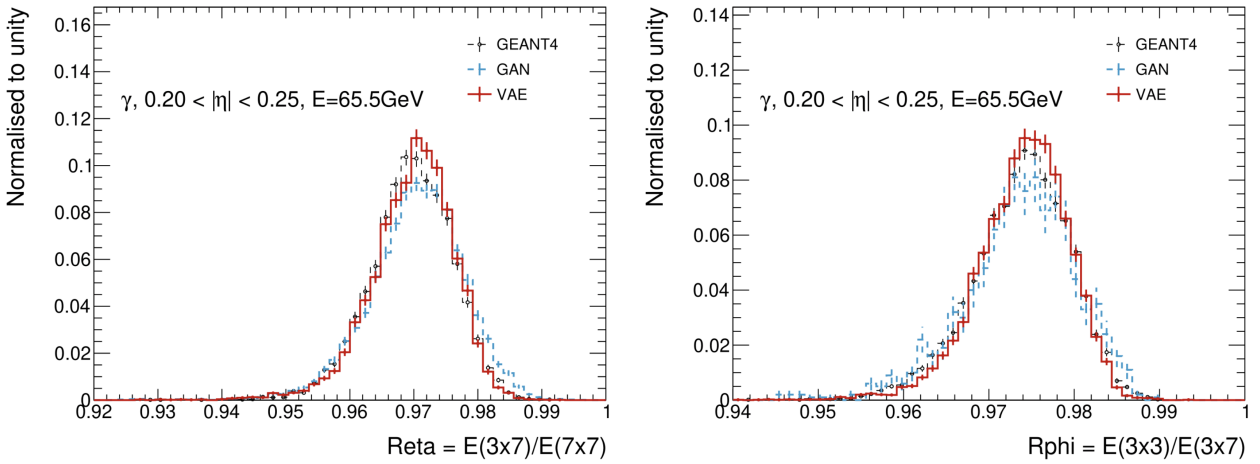
Figure 78 shows the performance on the shower shape variables *Reta* and *Rphi* which are used in *EGamma* particle identification (Section 4.2). The two quantities represent the energy ratio of the core 3×3 cells to the 3×7 cells in EMB2. The models are not explicitly optimized to learn these quantities, but the results shows a good agreement.

Figure 79 reports the *Eratio* distributions showing the good performance of the VAE for all energies from 2 GeV to 262 GeV. This variable describes the ratio of the highest and second-highest energy deposit in the cells in EMB1 to their sum.



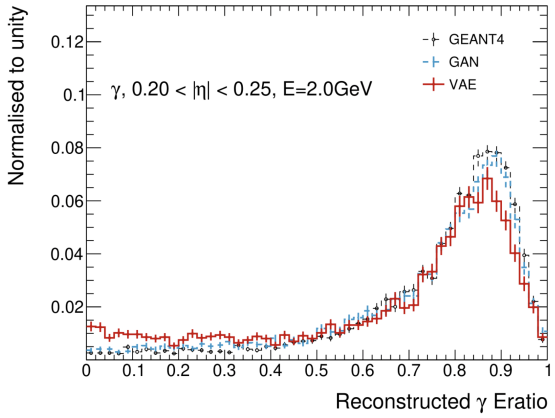
(a) (b) (c) (d)

Figure 77: *weta*1 (a), *weta*2 (b), *widths* (c) and *wtots* (d) for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

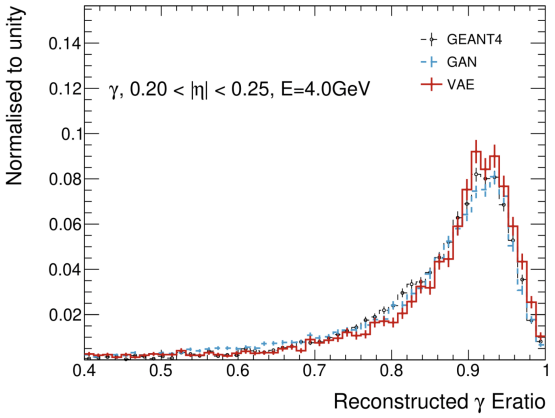


(a) (b)

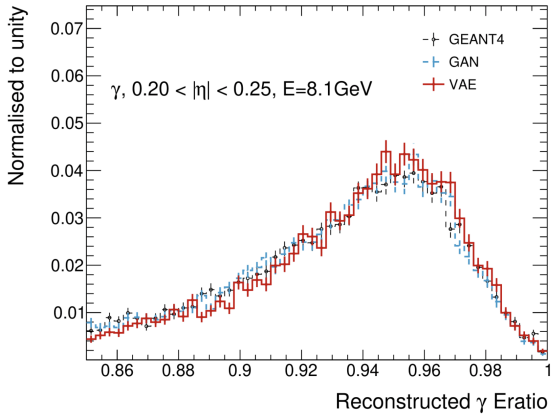
Figure 78: Reta (a) and Rphi (b) for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



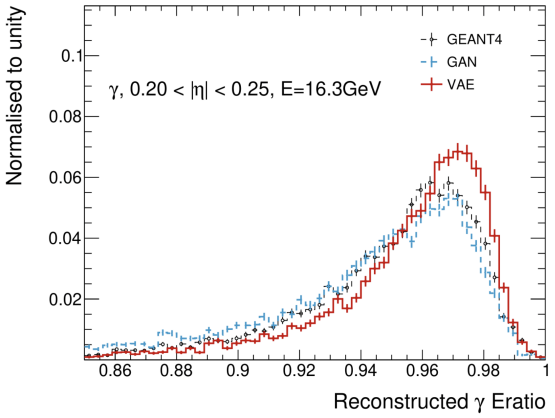
(a) 2 GeV



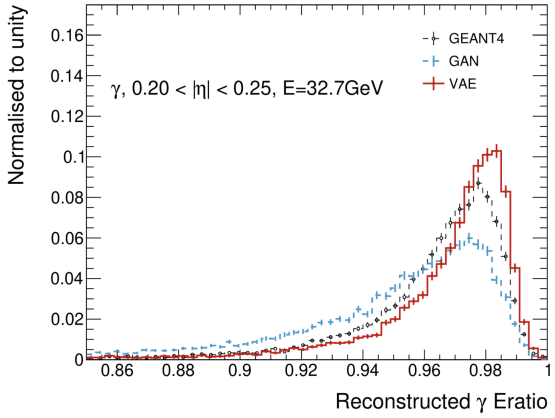
(b) 4 GeV



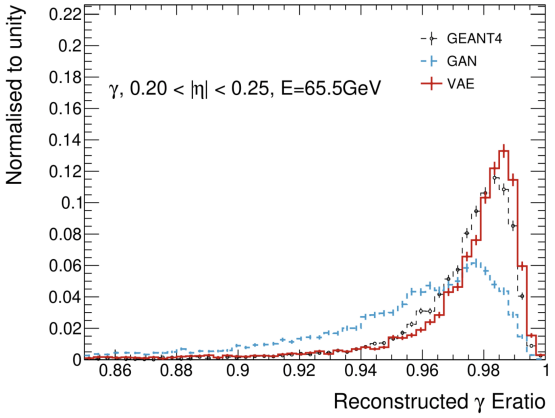
(c) 8 GeV



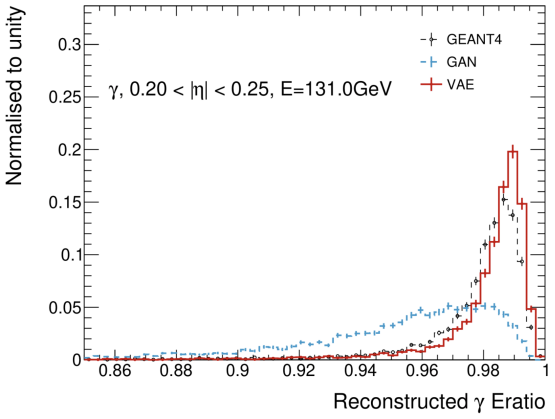
(d) 16 GeV



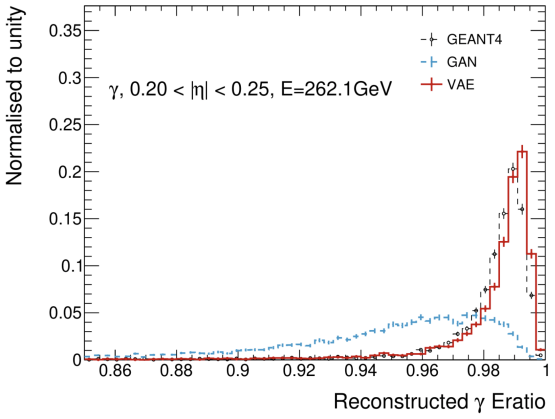
(e) 32 GeV



(f) 65 GeV

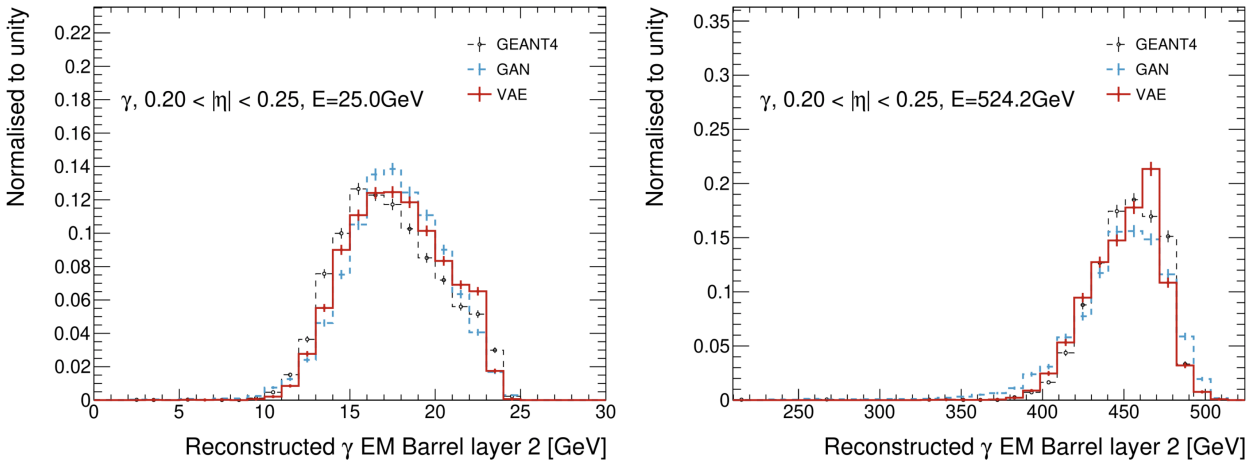


(g) 130 GeV



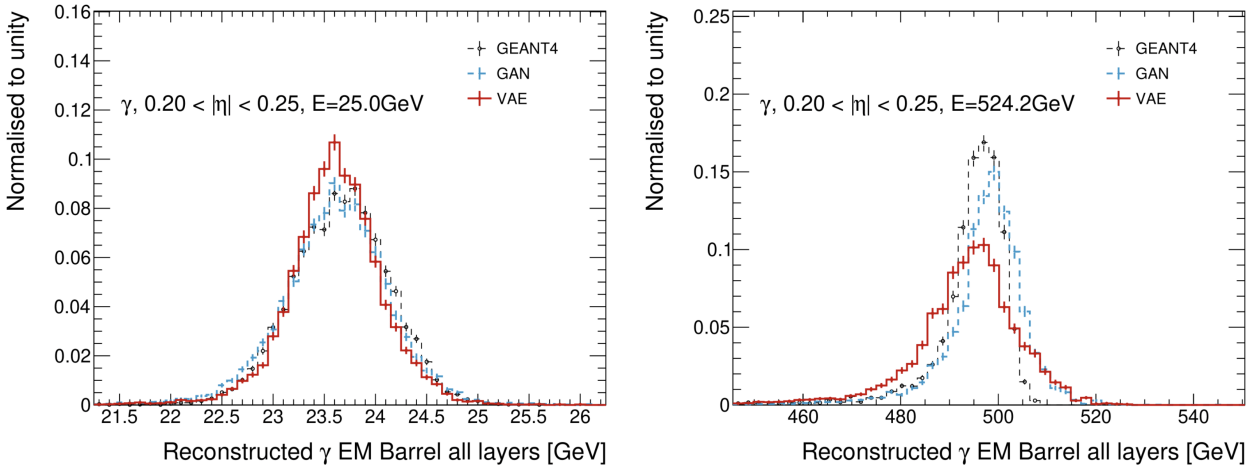
(h) 262 GeV

Figure 79: E ratio for photons with an energy of approximately energies from approximately 2 GeV to 262 GeV in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



(a) (b)

Figure 80: Energy in EMB2 for photons with an energy of 25 GeV (a) and 500 GeV (b) in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



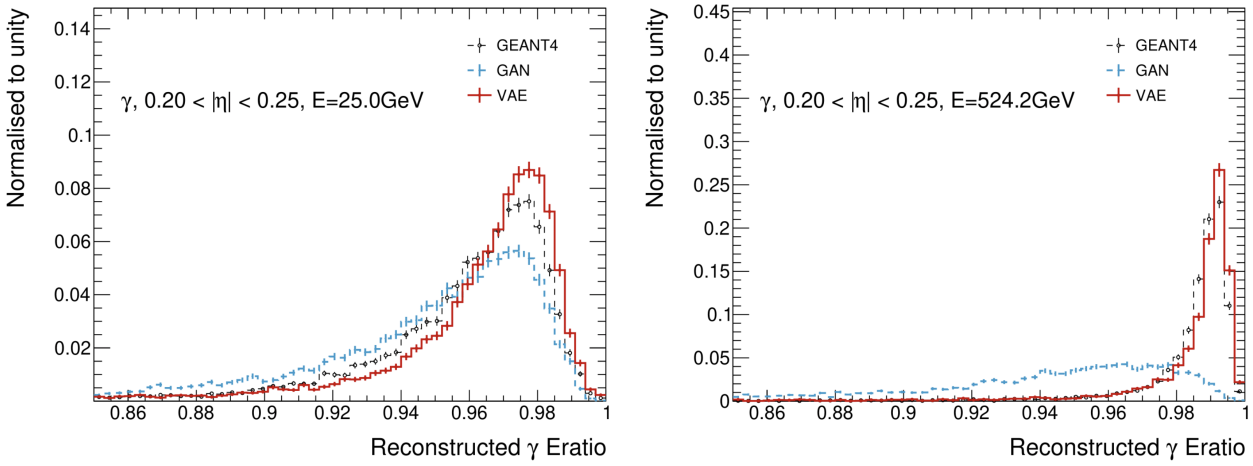
(a) (b)

Figure 81: Energy in ECAL for photons with an energy of approximately 25 GeV (a) and 524 GeV (b) in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

8.2.5 Interpolation and Extrapolation

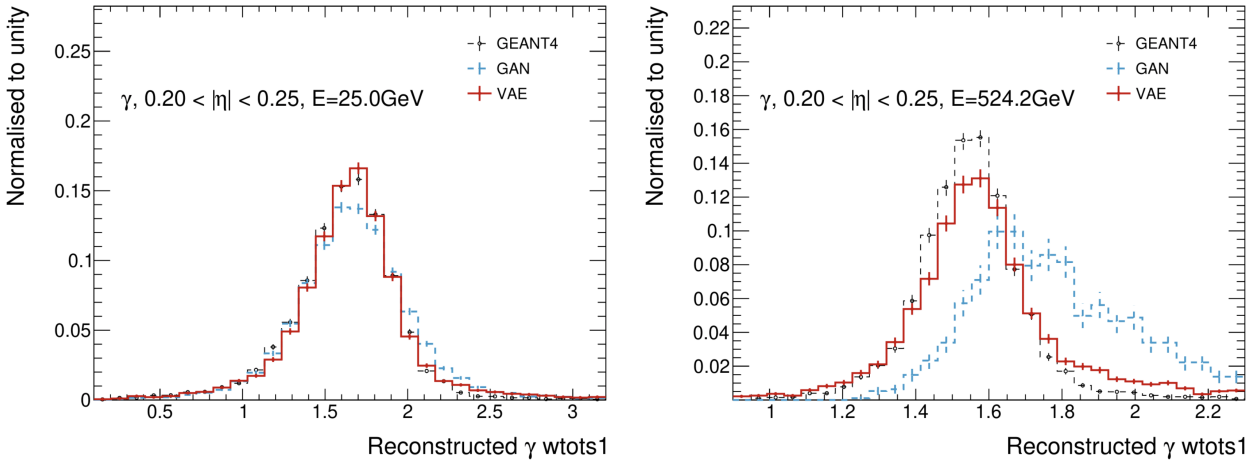
One of the key features of a generative model is the capacity to infer distributions of unseen data points during the training. Unseen data points refer to showers originating from particle energies different from the range of training. The VAE model detailed in the previous section is conditioned on nine discrete energies 1, 2, 4, 8, 16, 32, 65, 130 and 262 GeV. By conditioning on the energy, we expect to interpolate/extrapolate on other energy points. To assess the model performance of generating showers for particles with unseen energies, an interpolation, and an extrapolation results are presented in this section. Interpolating a shower development refers to simulating a shower with a particle energy within the training range, such as 25 GeV. Extrapolating, on the other hand, uses an energy point beyond the training range, such as 524 GeV.

Figures 80, 81, 83 and 83 show the distribution of the energy in EMB2, the total energy, *Eratio* and *wtots*1 respectively. The model performs better in interpolating than extrapolating. Moreover, the lack of information such as the total energy during training constrained the performance by generating a wider spread of this quantity.



(a) (b)

Figure 82: Energy ratio for photons with an energy of approximately 25 GeV (a) and 524 GeV (b) in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.



(a) (b) V

Figure 83: wtots for photons with an energy of approximately 25 GeV (a) and 524 GeV (b) in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (black markers) are shown as reference and compared to the ones of a VAE (solid red line) and a GAN (solid blue line). The shown error bars indicate the statistical uncertainty of the reference data and the synthesized samples.

8.3 Summary and Discussion

The chapter, summarizes the studies of Reference [123] which described the first application of generative models to simulate particle showers in ATLAS. A VAE and a GAN models are trained to learn the ATLAS ECAL response of photons in the range $0.20< |\eta| <0.25$ with energies between 1 and 262 GeV. The cell energy based VAE model is only conditioned on the energy of the incoming particle. Compared to the GAN model, which is in addition conditioned on the alignments of the cells, it was found that adding this condition to the VAE does not impact the performance.

To assess the performance of the VAE model, two validation steps are presented in this chapter: reconstruction and generation. The former is used to visually compare a Geant4 shower to its VAE reconstructed version. In the latter, VAE-generated showers from uncorrelated Gaussians with specific energy values are used to plot distributions describing shower energies and shapes compared to Geant4. Almost similar performance is shown for both generative models, with some quantities well described with one model compared to the other. The bulk of the energy in the calorimeter layers is better modeled than the tails by both the VAE and GAN. This indicates an underestimation of the underlying correlations. As a result, the total energy response, shown in Figure 64, is not well reproduced with a wider spread than Geant4.

It was found that the VAE training on the absolute energies of the cells tends to either model the energy per layer or the total energy of the shower. In order to overcome this limitation, in the second part of this chapter, the improved VAE model learns from a reparameterized dataset. This reparametrization consists of normalizing the energies of the cells with respect to the energy of the layer for each of the showers. This is represented as the energy ratio based model. In order to renormalize back to absolute energies after training and the generation of new showers, the VAE model is tasked to learn the energy per layer. Moreover, adding the values of the energy deposited in each layer and the total energy helps the model to learn the correlations between layers. In the reparametrization, this is encoded as five additional inputs to the model, where each of them is also a ratio value. Therefore, the energy per layer is relative to the total energy of the shower and the total energy is normalized with respect to the energy of the particle.

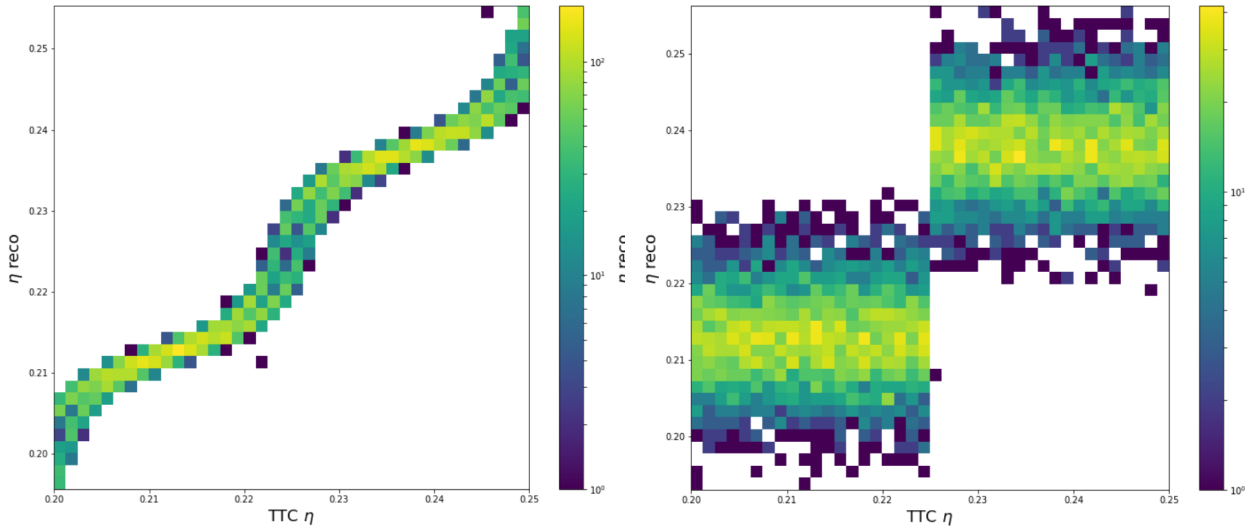
In addition, to improve the reconstruction quality of the shape of the showers, the VAE reconstruction loss function is augmented with a weighting term for each reconstructed feature. This weighting acts a penalty for the reconstruction of the feature. The weights are computed as the inverse of the standard deviation of the input distribution per feature. Cells at the edge of a shower are characterized by narrower spread of energies. The weight of these cells is higher which means a higher penalty on the reconstruction value and this means pushes the model to better learn these cells.

For the GAN model, the main component to better model the total energy is using the WGAN-GP flavor with an energy critic network. Both VAE and GAN improved the modeling of this quantity. For the energy deposited in each layer of the calorimeter, since the VAE is tasked to learn the relative energies of cells to the energy of the layer and the additional fractions for each layer, this helped to better reconstruct these distributions.

A list of shower shape distributions is presented from the integration in the Athena framework, such as *Reta* and *Rphi*. Overall, good agreement can be seen between the two generative models to Geant4.

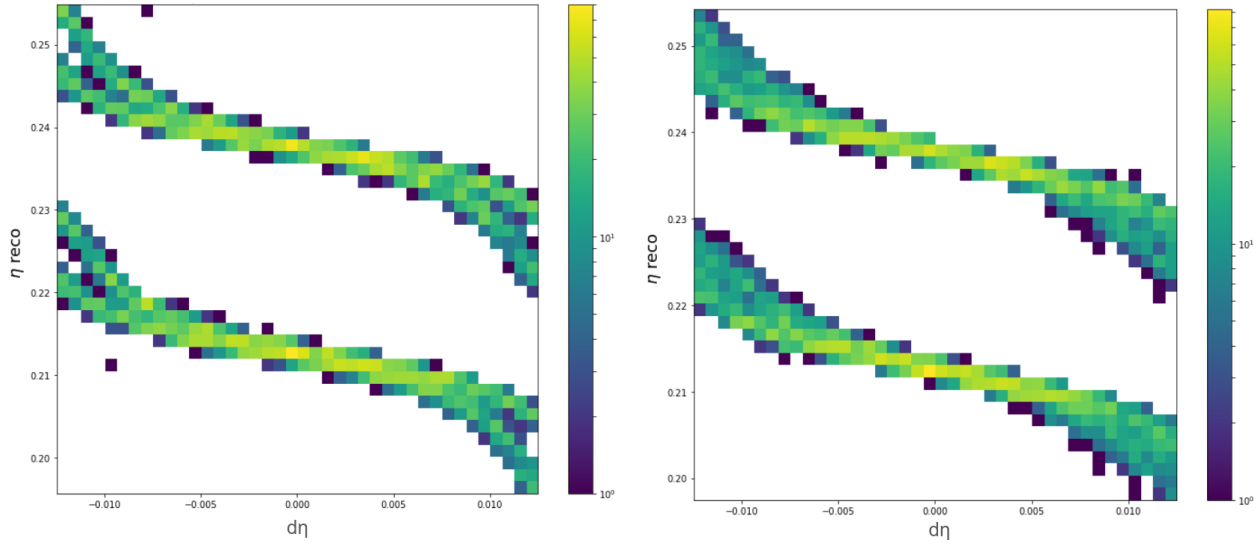
Section 8.2.5 highlighted the results of interpolating and extrapolating to other energies not seen during training. The models are trained on nine discrete particle energies. To test the interpolation (extrapolation) capacity, an energy point is chosen within (outside) the range of training of 1 GeV to 262 GeV. For interpolation, this allows us to get an insight on the generalization capacity of the model, in the way it uses patterns learned from the initial reconstruction/generation task adapted to the new energy value. The results show that similar performance is seen compared to the training energies. This means that interpolation can alleviate the problem of time consumption of training on massive datasets of showers. In other words, instead of using a continuous energy distribution of truth energies, a discrete segmentation would reduce the training time and interpolation can be used for all the non-trained energy values within the range of choice. The extrapolation, on the other hand, is a much harder task due to the fact that the model tries to deliver higher-dimensional predictions from a lower dimensional training. The behavior of the network can be interpreted as finding extrapolate-able values for cells in the ECAL layers coming from the implicit definitions about the relative energies used for training. Although the agreement to Geant4 is not well reproduced for all the shower observables, the model can predict some of the underlying shower distributions.

As described in Section 8.1, the cells are selected with respect to the impact cell in EMB2 closest to the extrapolated position of the photon. The VAE models in both sections of this chapter are not conditioned on these values. Figure 84 shows the reconstructed η as function of the true photon η (TTC η), where $\eta_{reco} = \frac{\sum E_i \times \eta_i}{\sum E_i}$, with i being the cell index in a layer (in this Figure in EMB2). The Geant4 shape is not reproduced by the VAE model. Alternatively, to visualize the impact of this condition on the performance of the VAE, a



(a) (b)

Figure 84: Reconstructed η weighted shower center as function of the impact point displacement η relative to the center of the impact cell for the (a) the full simulation (b) VAE, for photons with an energy of 65 GeV in $0.20 < |\eta| < 0.25$.



(a) (b)

Figure 85: Reconstructed η as function of relative η for photons with an energy of 65 GeV in $0.20 < |\eta| < 0.25$. The full simulation is shown in (a) and VAE in (b).

conditioning on the relative position to the center of the cell closest to the truth, ensures a modeling of this shape. Figure 85 shows the reproduced shape of Geant4.

9 Voxel-level FastCaloVSim

Extending the learning capacity of the model to generate electromagnetic showers across η regions using the cell level representation is a very challenging task. This is due to the detector geometry with its changing cell sizes per layer and per η region. These heterogeneous resolutions are hard to model with a single network. The voxel level alleviates this challenge by building 2D images of showers in polar coordinates with finer granularity than the cells. Moreover, this extension, compared to the previous chapter, includes the particle type, the subsequent relevant layers, the energy, and η ranges of the incident particle.

9.1 Voxelization Procedure

In this chapter, the VAE is trained, tested and validated on single photons and single pions with an energy range from 1 GeV to 1 TeV and $0 < |\eta| < 0.8$. The energy range covers the majority of the shower development phenomena. Moreover, beyond 1 TeV it is rare to observe particle showers of photons/pions. The η range on the other hand, represents the most homogeneous region of the calorimeter to test the extended model conditioned in addition on η.

The Geant4 hits are converted from Cartesian coordinates to cylindrical coordinates. In these coordinates (r,α,z), r represents the distance between the hit and the extrapolated direction of the truth particle in a layer, α is the angle computed as $\alpha = \arctan_2(d\phi, d\eta)$, where $d\eta$ and $d\phi$ are the relative positions of the hit to the extrapolated position of the truth particle in η and ϕ, z is the hit position of the axis pointing from the origin (0,0,0). Each of the calorimeter layers is considered independently of the others. Therefore, the z information is not used and only polar coordinates (r,α) represent a hit with its energy per layer. The (r,α) voxelization is derived per layer, where for photons it considers only the presamplerB, EMB1, EMB2 and EMB3. For pions, it adds the tile barrel layers (TileBar0, TileBar1 and TileBar2).

Table 7 summarizes the binning for both particles. The layers are binned along r with a fixed bin width for all layers except for the presampler. The fixed width is used to contain the shower shape development with minimal statistical fluctuation of energy. A variable bin width in r for the presampler is used because this layer contains the lowest fraction of the total energy. Therefore, with a progressively larger r, the shape is preserved. As a result of the bin width, the number of r bins or rings and the step size between the bins are defined per layer. The definition of the maximum r is computed as the 0.995 quantile of the dr distribution weighted by hit energies across all truth energies. Figure 86 shows this distribution for single pions in a central η slice in TileBar2. These maximum r values represent the last value per layer marked in bold text in Table 7. The binning in α on the other hand, is uniform and considered as 8 bins in all the layers, with $\alpha \in [0, 2\pi]$ and therefore it is not reported in Table 7. This number of bins represents a good compromise between granularity and energy deposition.

The binning in the ECAL layers for photons and pions remains the same, since it provides a good granularity to contain a pion shower. For the HCAL layers, by design they are coarser than the barrel layers with an order of 200/300 mm compared to 25 mm size in EMB2. This implies to have wider voxels. The granularity definition for the HCAL layers is based on using the relative size of the voxel to cell in EMB2. This relative voxel size translates to a voxel size in the HCAL layers. A voxel in EMB2 is 5 mm and using the ATLAS geometry information, the mean cell width in r is 171 mm. The relative voxel size is then 34.

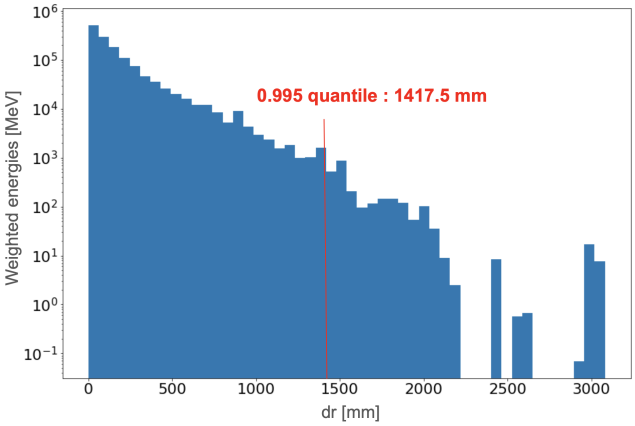


Figure 86: dr distribution weighted by the energy of the hits for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$ in the TileBar2 layer. The 0.995 quantile value is reported in red.

Layer	r edges [mm]
PresamplerB	0, 5, 10, 15, 20, 30, 50, 100, 200, 300, 400, **600**
EMB1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, **200**
EMB2	0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, 170, 175, 180, 185, 190, 195, 200, 205, 210, 215, 220, 225, 230, 235, 240, 245, 250, 255, 260, 265, 270, 275, 280, 285, 290, 295, 300, 305, 310, 315, 320, 325, 330, 335, 340, 345, 350, 355, 360, 365, 370, 375, 380, 385, 390, 395, **400**
EMB3	0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, **600**
TileBar0	0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540, 570, 600, 630, 660, 690, 720, 750, 780, 810, 840, 870, 900, 930, 960, 990, 1020, 1050, 1080, 1110, 1140, 1170, 1200, 1230, 1260, 1290, 1320, 1350, 1380, 1410, 1440, 1470, 1500, 1530, 1560, **1590**
TileBar1	0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540, 570, 600, 630, 660, 690, 720, 750, 780, 810, 840, 870, 900, 930, 960, 990, 1020, 1050, 1080, 1110, 1140, 1170, **1200**
TileBar2	0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420, 450, 480, 510, 540, 570, 600, 630, 660, 690, 720, 750, 780, 810, 840, 870, 900, 930, 960, 990, 1020, 1050, 1080, 1110, 1140, 1170, 1200, 1230, 1260, 1290, 1320, 1350, 1380, **1410**

Table 7: The r binning used for the voxelization of photons and pions in the different calorimeter layers with $0 < |\eta| < 0.8$ and energies logarithmically spaced between 1 GeV to 1 TeV.

	Cells	Voxels
PresamplerB	21	88
EMB1	171	1600
EMB2	49	640
EMB3	35	96
Total	276	2424

Table 8: Number of cells compared to the number of voxels derived from the data processing step of photon particles.

Given the different nature of particles, a model per particle is optimized. The next section is dedicated to the photon model, and the subsequent section then describes the pion model.

9.2 Voxel-level FastCaloVAE for Photons

Using the above voxelization procedure, the proposed VAE version in this section is a set of experimentally motivated choices for designing a model which can reproduce all the shower shape variables for an extended energy and η range. The number of inputs to the VAE is now a factor of 100 more as compared to the cell-based model in Chapter 8. Table 8 compares the number of volume spaces per layer and in total for cells versus voxels. EMB1 and EMB2 are the most energetic layers in the considered η regions of the calorimeter. They are 9 and 13 times more granular, with the voxel definition.

The models in this chapter are also trained on energy ratios. In order to cope with the all new extensions and to ensure a better performance of the model, the learning approach is enhanced with a set of components:

- **Softmax activation functions**: from prior knowledge on the deposited energy, both voxel values and all total energies per layer should sum up to one. This can be translated in the output layer of the VAE by applying a softmax activation function. It is commonly used for classification tasks where the network predicts N classes (N outputs). By using the softmax function, the N values are converted into probabilities that sum to one and automatically the range of values falls in [0,1]. This matches exactly the requirement of the energy ratio-based definition and therefore the output layer is augmented with two softmax functions.
- **Adapted physics weights**: using the same physics weights computation in Section 8.2.2 at the voxel level can not be straightforward due to the fact that voxels in the same r ring should be equally weighted. We propose an adapted algorithm which computes the weights by applying a successive merging of voxels in α and in r.
- **Adapted loss formulation**: in parallel with the addition of the softmax activations, an adaptive version of the loss function is designed with three weighted losses. The three terms allow the VAE to accurately model each one of the following quantities: voxel energies, total energies and energies per layer. Furthermore, since the output function has probabilities from the softmax application, it is more convenient to use the cross entropy function as a loss instead of the MSE.
- **Noise addition**: from prior knowledge on detector design, the model is augmented with an additional input of random fluctuations referred to as noise. This noise is included in the Geant4 events to compensate the non-measured energy, lost in the dead material. Since this is an uncorrelated quantity, it is injected in the learning process of the VAE. The impact of adding this noise on the performance of the VAE can be seen, for example, for the reconstructed energy per voxel as shown is Figure 87. The VAE with injected noise can better model the energy of this voxel.

9.2.1 Adapted Physics Weights

In the previous chapter, we introduced the concept of weighted reconstructions of nodes. For the voxel level, the same idea of incorporating the physics weights into the learning procedure is used in a more elaborated way. Figure 88 shows the reconstruction output of the VAE, i.e., when using the end-to-end networks by encoding and decoding on a test set of unseen showers. The figure shows the ratio of the average voxel energy of the VAE output to the average voxel energy of Geant4 inputs as function of dr values in EMB2. A single η slice of $0.2 < |\eta| < 0.25$ is shown for different energies, low, medium and high. The reconstruction shows an agreement to Geant4 translated to a ratio of 1 visible in the voxels in the core of the shower, and it degrades when dr is larger. The voxels in the outer rings represent a small amount of energy depositions.

Applying the same idea of weight computation for the cells, Algorithm 2 is defined. Figure 89 shows the derived weights using Algorithm 2. This computation is refered to as raw computation, where a weight is computed for

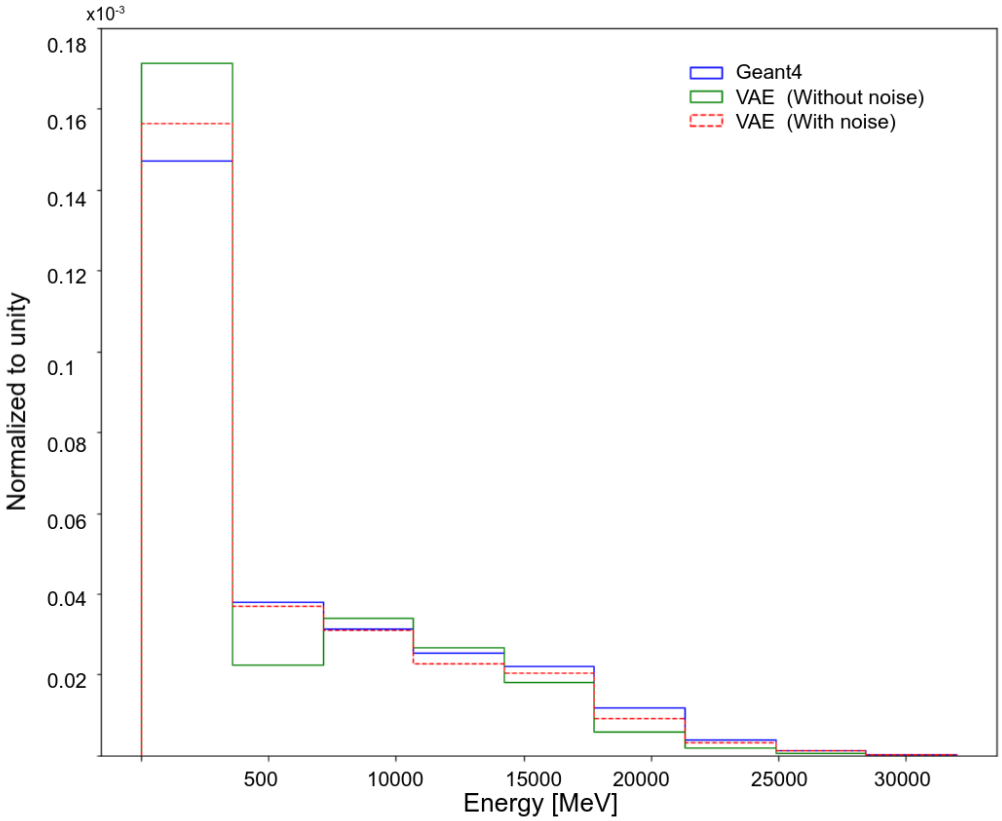


Figure 87: Energy of a center voxel in EMB2 for photons with an energy of 65 GeV (a) in the range $0.20 < |\eta| < 0.25$. The quantities from the full detector simulation (blue markers) are shown as reference and compared to the ones of a VAE without the noise (solid green line) and a VAE with the noise (dashed red line).

every voxel (and the additional fractions). All the α bins are shown in the plots as the points belonging to the same vertical line with the same dr value. This raw computation attributes different weights to the α voxels in the same r rings.

Consider now, a single event K, a single calorimeter layer l, a voxel i in a ring r. An α merging consists of computing the ratio value $R_{Ring(r)}$ per r ring by summing up the voxel energies in this ring divided by the total energy per layer for each of the events, as shows Figure 90. Using the binning defined in Table 7, the number of rings N in the ECAL layers is 11, 200, 80 and 13 for the presampler, EMB1, EMB2 and EMB3 respectively. By applying an α merging, the weights are then derived per ring. Therefore, all the voxels in the same r ring have the same weight as shown in Figure 91. The fluctuations seen in Figure 91 between the weights of the successive rings is due to the fact that the different rings have different energy depositions, resulting from the stochasticity of the showering process. Taking EMB2 as an example, a shower s_1 has only energy deposited in the three first rings and s_2 has different configuration set and has the depositions in 5 rings. Therefore, taking all showers from all energies and η slices results in having these fluctuations. To remedy these fluctuations, the weights are instead derived by grouping the rings. The grouping consists of summing up the energies deposited in the successive set of rings and then computing the energy ratios. Taking photons of 65 GeV energy in $0.2 < |\eta| < 0.25$ in Figure 92, the innermost rings, such as ring 1, have most of the energy deposited and when moving far away from the center the deposition gets smaller. Thus, the energy distributions or energy ratio distribution gets narrower. The idea of grouping rings is based on these distributions in order to determine from which ring the merging process can start and how many rings can be merged. Algorithm 3 shows the computation function of the rings merging. Figure 93 shows the output weights for each of the merged rings. These rings are 3, 12, 7 and 7 for EMB0, EMB1, EMB2 and EMB3 respectively. All voxels in the same merged ring have the same weight. Weights are also derived for the additional fractions of energy per layer and total energy. Their computation is rather straightforward using the same Algorithm 2.

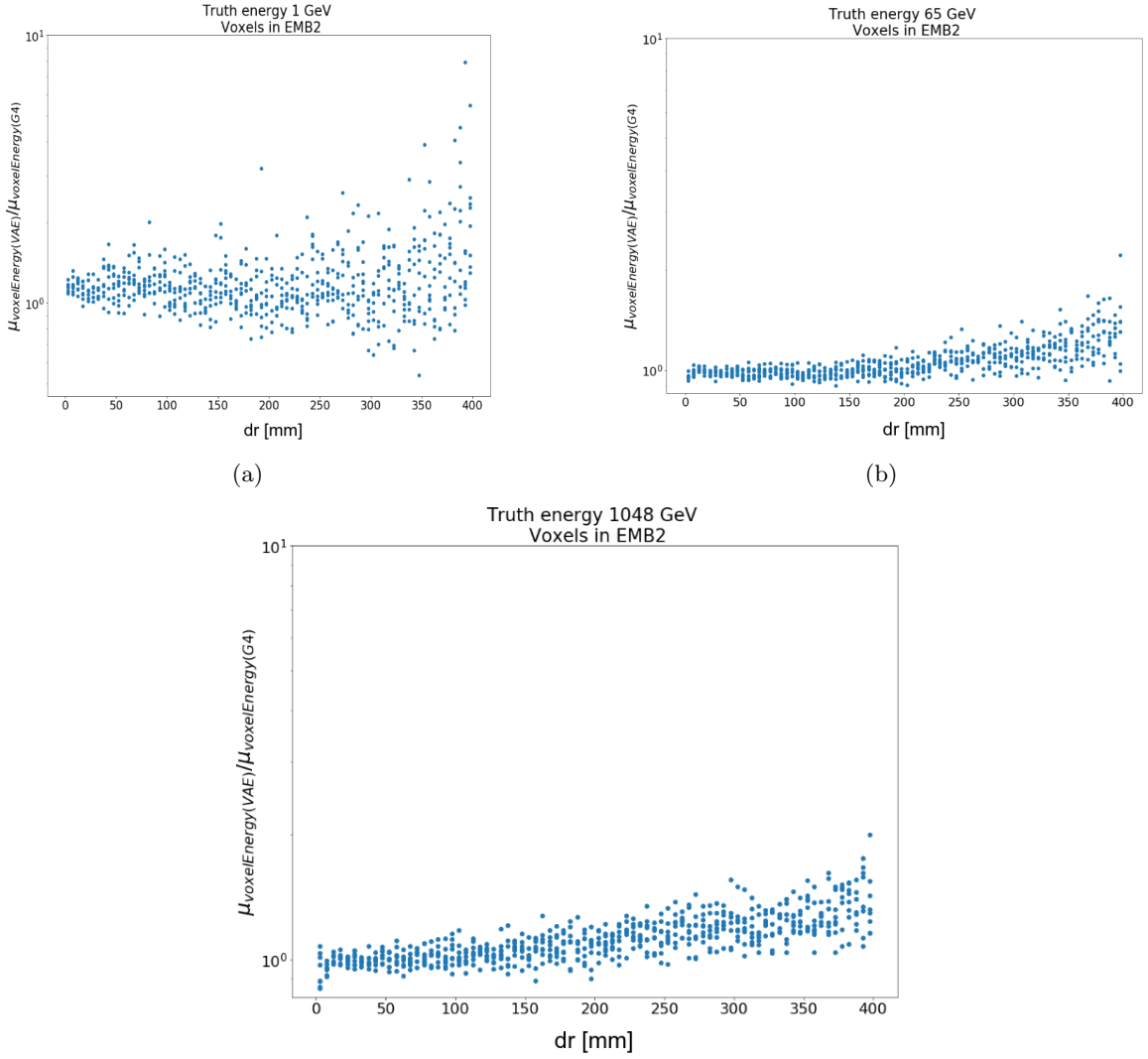


Figure 88: The ratio of the average voxel energies of the VAE output to the average voxel energies of Geant4 inputs as function of dr for photons with an energy of 1 GeV (a), 65 GeV (b) and approximately 1 TeV (c) in the range $0.20 < |\eta| < 0.25$.

Algorithm 2 Calculate a weight w_i for each voxel v_i per layer l

Result: w_i for each voxel v_i
Data: For each voxel v_i, RV_i is the voxel ratio values for all events/energies/η.
$n_{vox}(l)$ is the number of voxels in a layer l.
for i *in* $n_{vox}(l)$ **do**
  Scale RV_i in the range (0,1]
  $w_i = \frac{1}{std(RV_i)}$
end

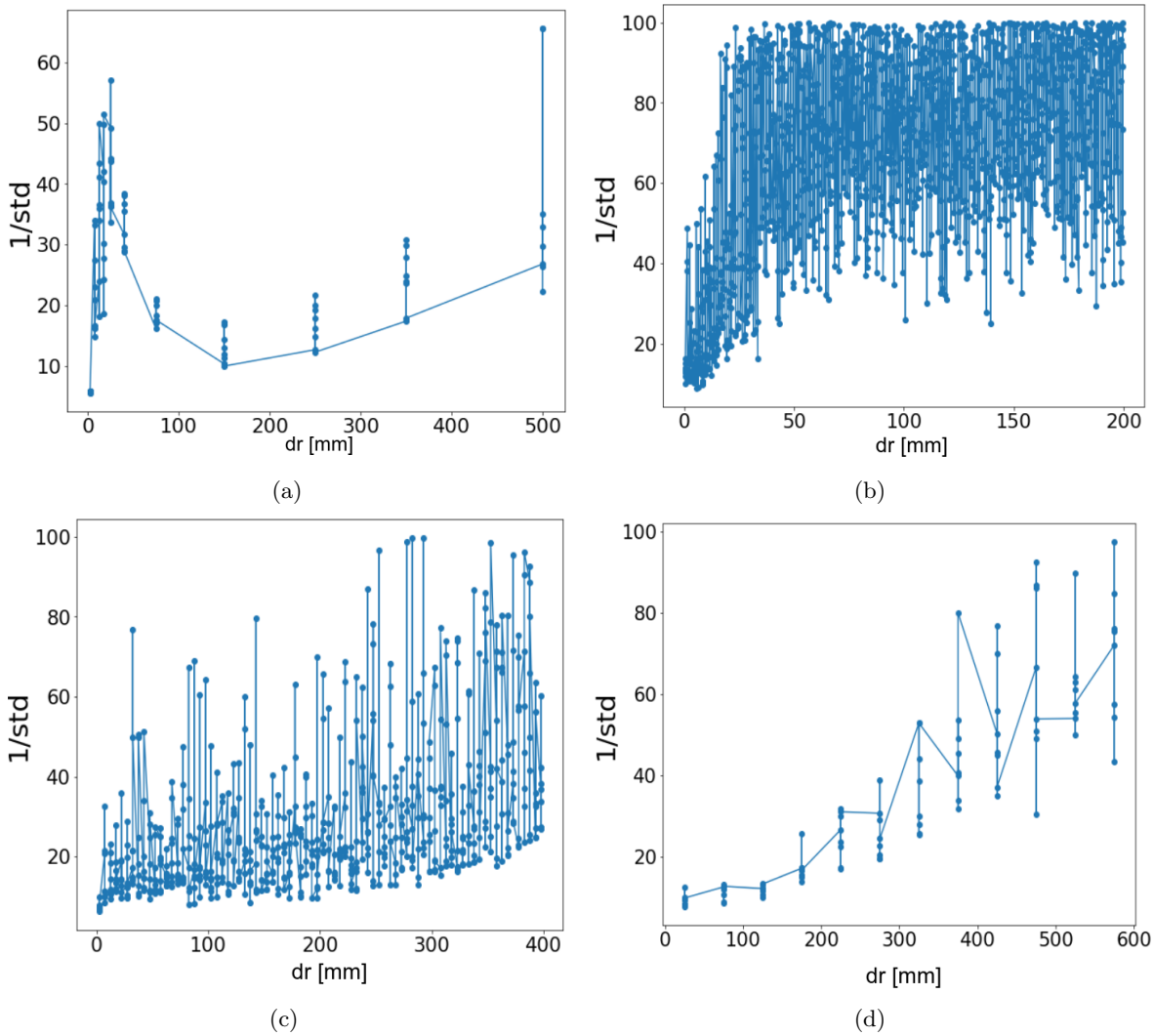


Figure 89: Derived physics weights per voxel for each of the layers (a) presampler, (b) EMB1, (c) EMB2 and (d) EMB3, using Algorithm 2.

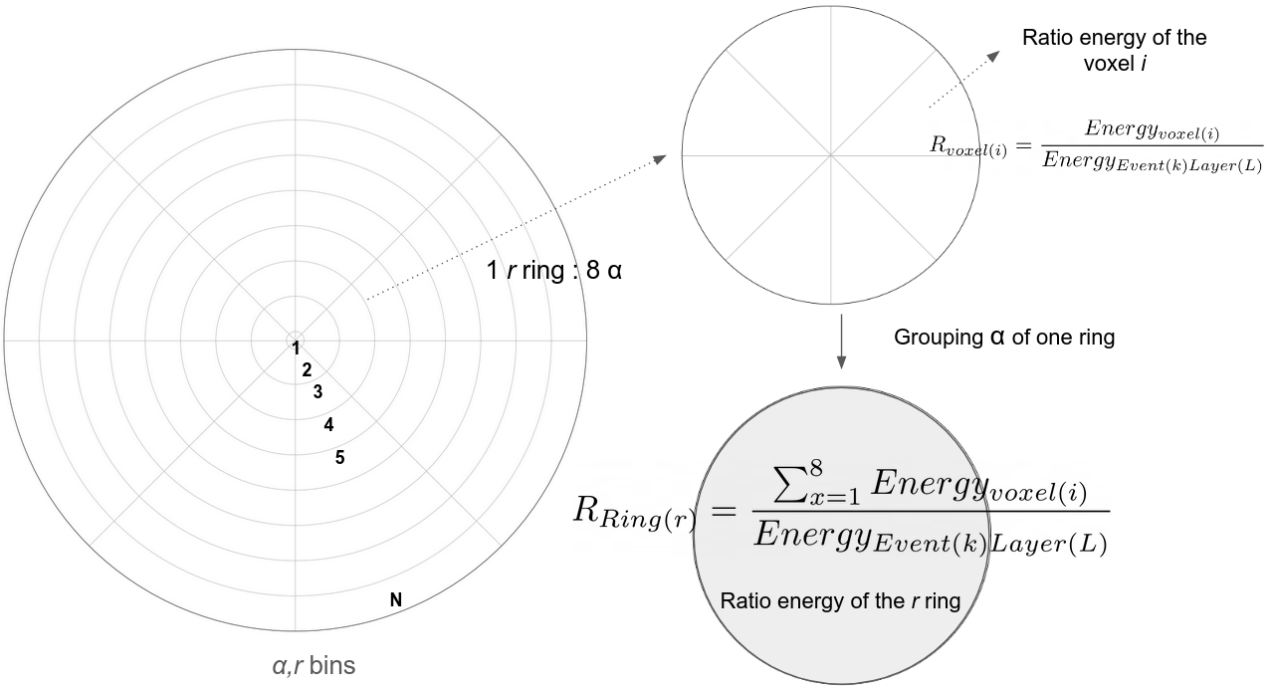


Figure 90: Representation of r, α bins with an annotation of the rings. For a single ring, the 8 α bins are shown with an energy ratio computation and a computation of a ratio per ring. .

Algorithm 3 Calculate a weight w_i for each block of rings *br* per layer

Input:

- layer l
- $n_{vox}(l)$: number of rings per layer
- start: ring from which start merge
- threshold : mean energy threshold for the grouping of the rings

ring = start

Weight values up to start ring remain the same

while $ring < n_{vox}(l)$ **do**

  Group all rings i $\geq$ ring while mean(ring i) $\approx \pm$ threshold ;

  vR = sum of ring ratios per event and per merged ring (vR array of size nbEvents);

  wV = 1/std(vR) (1 value representing the weight of the current merged ring);

  Scale RV_i in the range (0,1]

  $w_i = \frac{1}{std(RV_i)}$;

end

9.2.2 Adapted Loss Formulation and Model Design

An intuitive idea, when moving from cells to voxels, is to increase the width and depth of the model to cope with the high resolution and to allow feature learning over the layers. However, many considerations instead address optimization concerns. Figure 94 shows the architecture of the VAE model. It learns to reconstruct: 2424 nodes representing all the voxels in the four ECAL layers, one node represents the total energy and four nodes for the energy per layer. The conditional values of energy and η are an input to the encoder and decoder. The encoder learns a shower representation in ten dimensions. The fifth hidden layer of the decoder learns to recover the same number of nodes as in the input layer. The output of this layer is fed into three dense layers to add the restriction of the sum of one by using the softmax activation function on the 2424 nodes of voxel energy ratios and for the four nodes of energy per layer fractions. The activation function for the single node of total energy fraction is set to be a sigmoid in order to learn an output in [0,1]. Figure 95 shows the range of total energies starting from zero, where some of the particles do not interact in the calorimeter. The figure shows the total energies of 65 GeV photons in $0< |\eta| <0.8$. The fractions of these total energies to the truth energy (65 GeV) are in range [0, 1.04], these values are normalized to [0,1] for consistency with the sigmoid output definition. The last layer outputs a concatenation vector, an optional choice which, alternatively, can

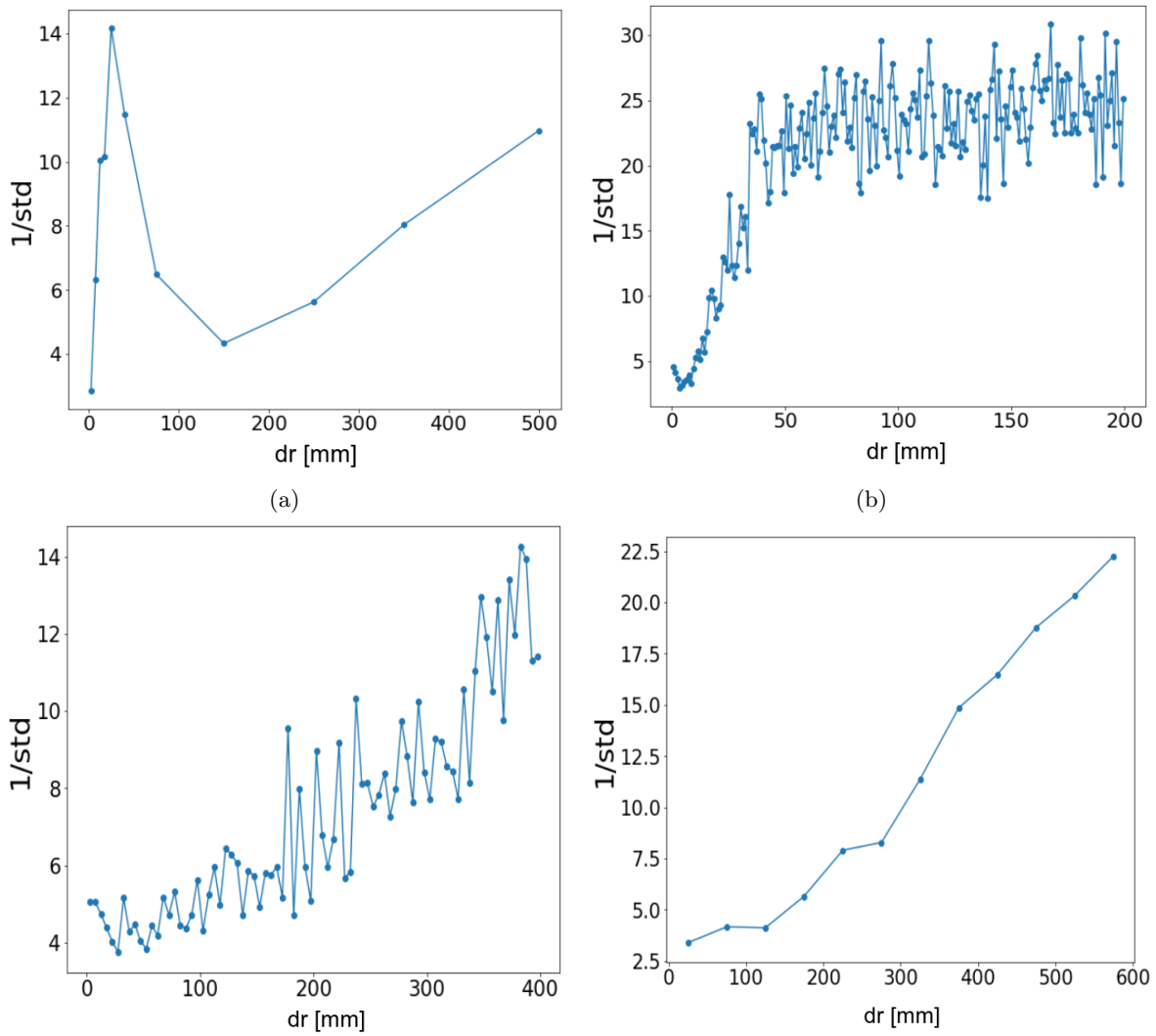


Figure 91: Derived physics weights per ring for each of the layers (a) presampler, (b) EMB1, (c) EMB2 and (d) EMB3, using Algorithm 2.

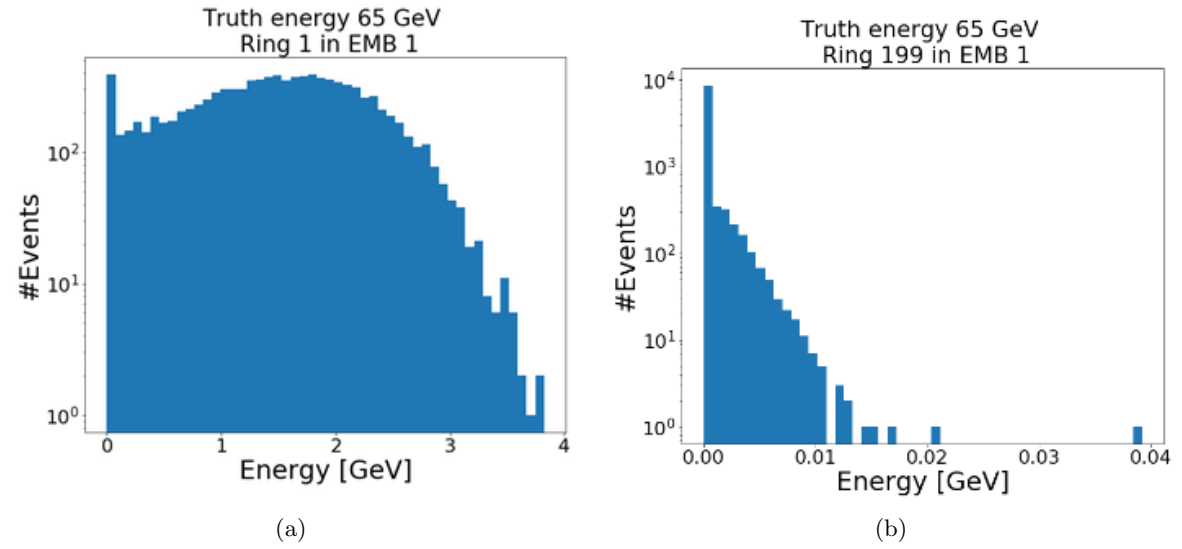
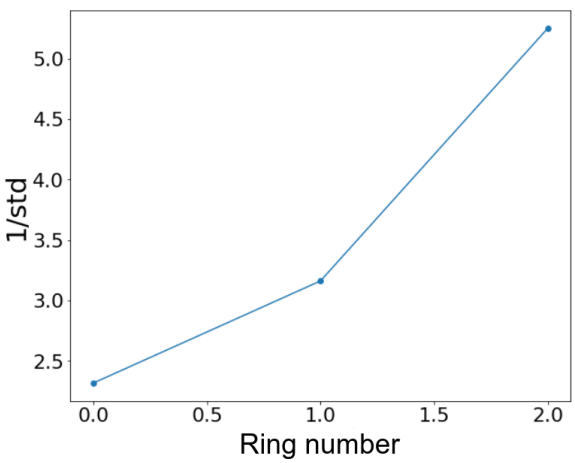
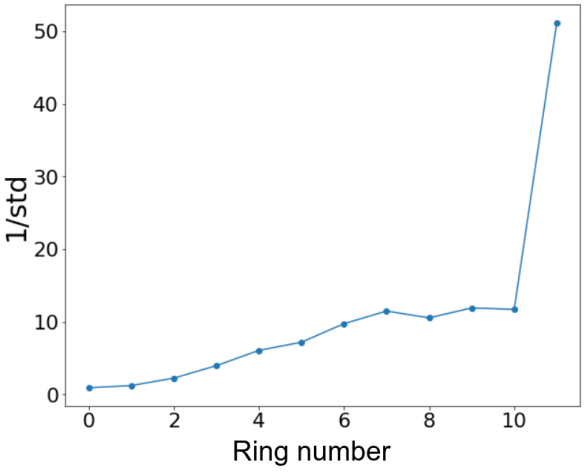


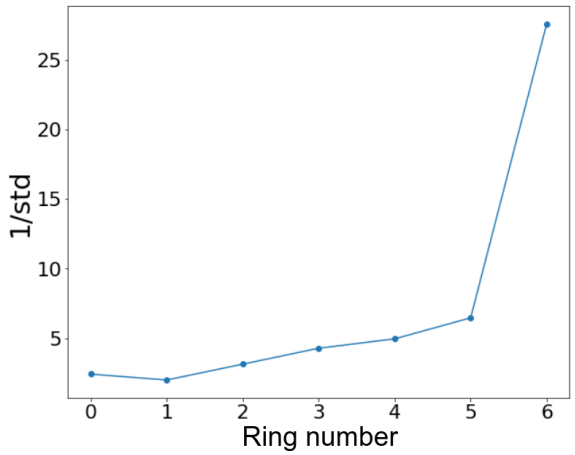
Figure 92: Energy deposition in ring 1 (a) and ring 199 (b) in EMB1 for photons of 65 GeV energy in $0.2 < |\eta| < 0.25$.



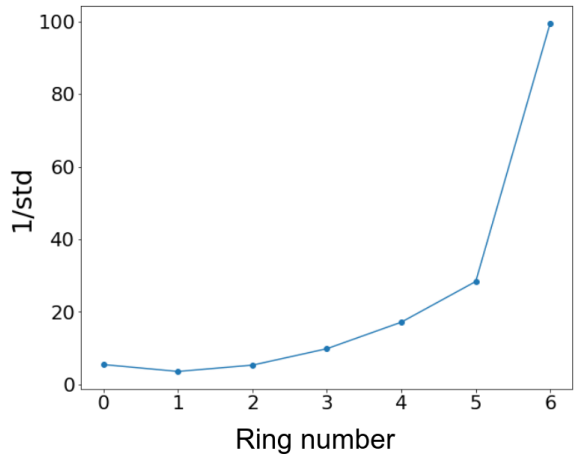
(a) Layer 0



(b) Layer 1



(c) Layer 2



(d) Layer 3

Figure 93: Derived physics weights per block ring for each of the layers (a) presampler, (b) EMB1, (c) EMB2 and (d) EMB3, using Algorithm 3.

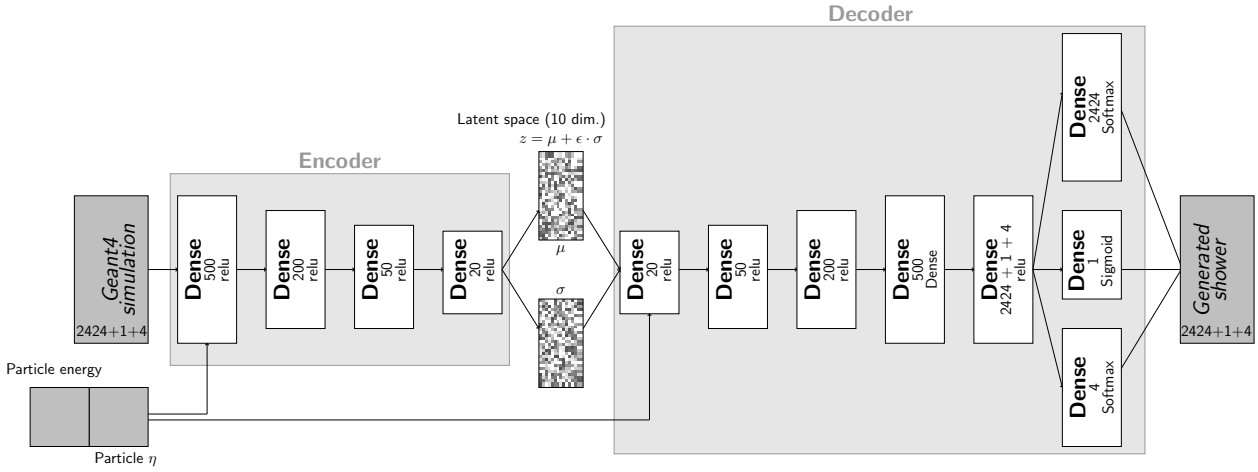


Figure 94: VAE architecture: the number of units per layer and number of layers are shown for both the encoder and the decoder. The VAE is conditioned on the truth particle energy and η .

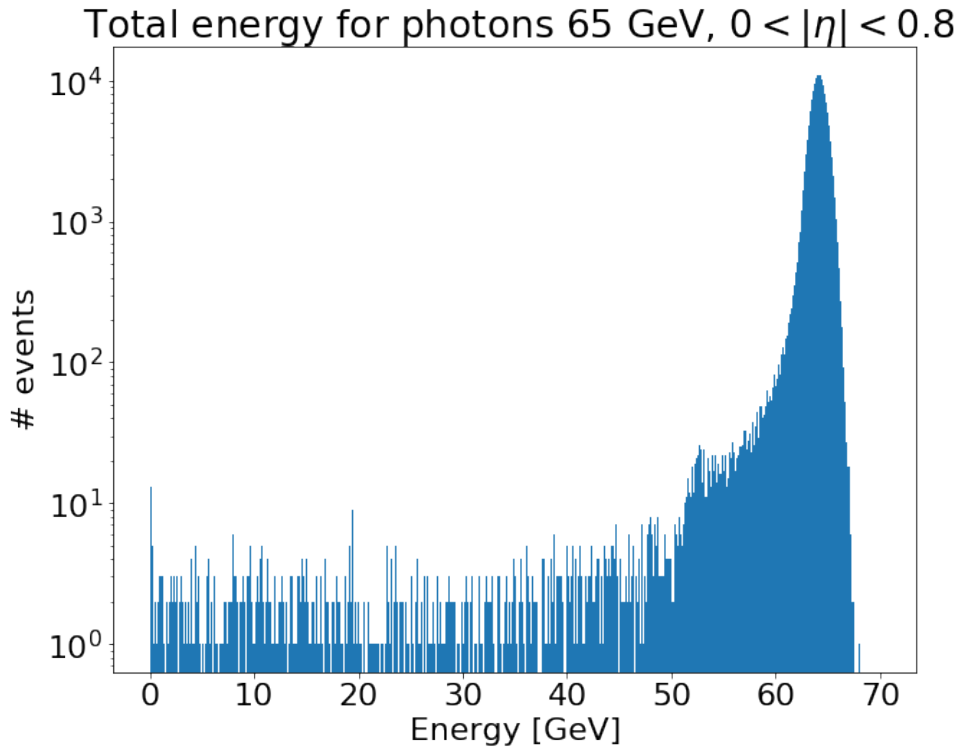


Figure 95: Total energy distribution for photons 65 GeV in $0 < |\eta| < 0.8$. 1000 events per η bin are used to plot the distribution. .

have multiple outputs (three in this case).

As mentioned above, the reconstruction loss is adapted from the cell model. Generally, when the data is a vector of binary values or a vector of probabilities, the cross entropy (Section 6.7) is optimized as a reconstruction function [194]. Furthermore, for a classification problem as an example, using the cross-entropy function instead of the sum of squares (or the mean squares) leads to an improved generalization and a faster training [195]. The VAE reconstruction loss function for the voxel level $\mathcal{L}_{Reco}$ is designed to be three embedded losses: $\mathcal{L}_{E_{vox}}$, $\mathcal{L}_{E_{Tot}}$ and $\mathcal{L}_{E_{Lay}}$ to reconstruct the 2424 voxels (softmax), the single node (sigmoid) of total energy fraction and the four nodes (softmax) of the energy per layer fractions. The three losses are of the form

$$L_{E_\lambda}(x, \tilde{x}) = -\frac{1}{n}\sum_{i=1}^{n} w_i \left[x_i \log(\tilde{x}_i) + (1 - x_i)\log(1 - \tilde{x}_i)\right],$$

where x and $\tilde{x}$ for each equation represent the same quantity of the same size n which is 2424, 1 and 4 for $\mathcal{L}_{E_{vox}}$, $\mathcal{L}_{E_{Tot}}$ and $\mathcal{L}_{E_{Lay}}$ respectively, w_i represents the physics weight of the node's reconstruction as detailed in the previous section. The reconstruction loss is then defined as the sum of the terms

$$L_{Reco}(x, \tilde{x}) = w_{vox}\mathcal{L}_{E_{vox}} + w_{Tot}\mathcal{L}_{E_{Tot}} + w_{Lay}\mathcal{L}_{E_{Lay}},$$

where w_{vox}, w_{Tot} and w_{Lay} (0.1, 0.5, 0.2) represent the weight of each loss term contribution, and they are optimized as hyper-parameters. The final loss function of the model includes the KL term to ensure the Gaussian property of the latent space. It is defined as

$$L_{VAE}(x, \tilde{x}) = L_{Reco} + w_{KL}L_{KL}(q_\theta(z|x)||p(z)).$$

9.2.3 Generation Performance

To assess the performance of the FastCaloVSim approach at the voxel level, one of the standalone validations consists of verifying the five fractional distributions that the model is trained on. Figure 96 shows these five quantities where the VAE model can reproduce them. These are crucial to learn to re-normalize the voxel ratios into energies. EMB2 is the layer where most of the photon energy is deposited in $0< |\eta| <0.8$. After the re-normalization and summing up the energies per event in EMB2, Figure 97 shows the EMB2 distribution across all the η range of the training. The plot shows that the VAE can reproduce this observable over the η range. Some discrepancies are observed in low-statistics regions.

A summary plot of the performance of the VAE trained on voxels for photons with 65 GeV in the $0< |\eta| <0.8$ range is shown in Figure 98. The plot shows the two main quantities of the total energy distribution: the mean and the RMS. The ratio between the mean of the total energy distribution of the VAE-generated events and Geant4 is shown in the y-axis. The ratio of RMS distributions is shown as the error bar. This plot shows that the VAE can reproduce all total energies over the η regions with a similar level of agreement.

The generation performance in the ATLAS Athena framework is based on simulating energy depositions in each of the voxels in the ECAL layers using the LWTNN file representing the decoder weights and architecture. After simulating the energy depositions in each of the voxels, the next step consists of assigning the voxels to the ATLAS calorimeter cells. To perform this operation, the energy of a voxel is considered as the energy of a hit. The next plots show the performance of the VAE on single photons with an energy of 65 GeV in $0.2< |\eta| <0.25$. Figure 99 shows the energy distribution per layer for each of the four considered ECAL layers. Figure 100 shows the reconstructed photon energy. Overall, the VAE is reproducing the distributions with some discrepancies compared to the full simulation.

9.3 Voxel-level FastCaloVAE for Pions

Pions are trained in the same way as photons. The VAE model learns to reconstruct energy ratios for these particles with a truth energy range from 1 GeV to 1 TeV in $0< |\eta| <0.8$. HCAL layers TileBar0, TileBar1 and TileBar2 (12,13,14) are considered in addition to the ECAL layers using the binning detailed in Table 7. The model architecture is similar to the photon architecture in Figure 94 in terms of depth of both networks and the structure of the input and output layers, with an optimized number of nodes for each of the layers. The size of the input and output layers is 6504+1+7, where 6504 represents the total number of voxels in the ECAL and HCAL layers, 1 for the fraction of the total energy to the truth energy and 7 for the fractions of the energy per layer to the total energy.

Figure 101 is a summary plot of the performance of the VAE on reproducing the mean and RMS of the total energies. The ratio between the mean of the total energy distribution of the VAE-generated events and G4 is shown in the y-axis. The ratio of RMS distributions is shown as the error bar. This plot shows that the VAE can reproduce all total energies across the truth energy range with a similar level of agreement. TileBar0 for pions is the most relevant layer. Figure 102 shows the good agreement of the energy distributions of this layer for all the truth energy range in $0.2< |\eta| <0.25$. Two performance plots from Athena are shown in Figure 103 for the leading cluster energy and η. The discrepancies illustrate the complexity of pion showers which are characterized by composite correlations difficult to learn and this leads to an incorrect energy assignment and therefore poor modeling of the energy of the cluster.

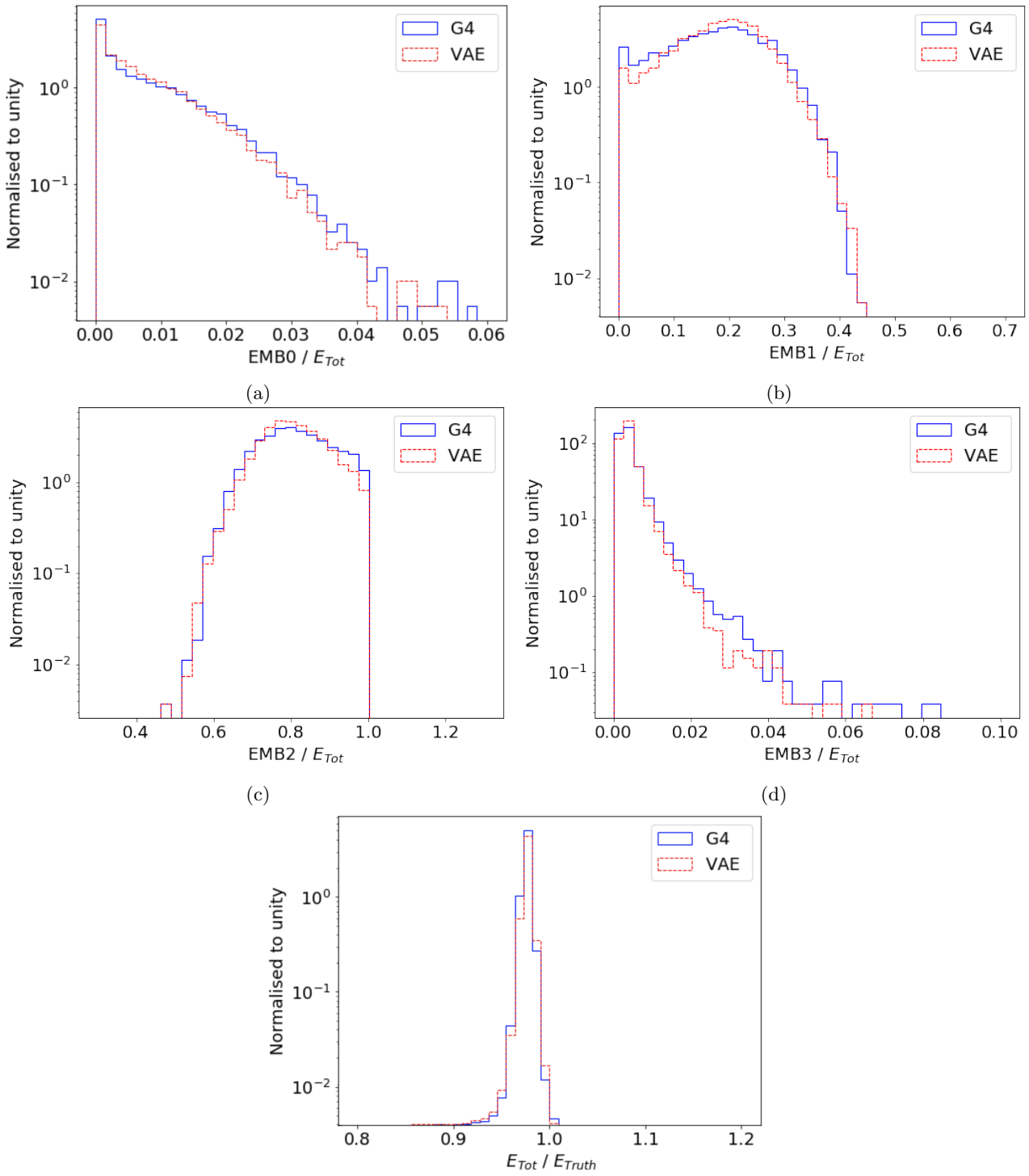
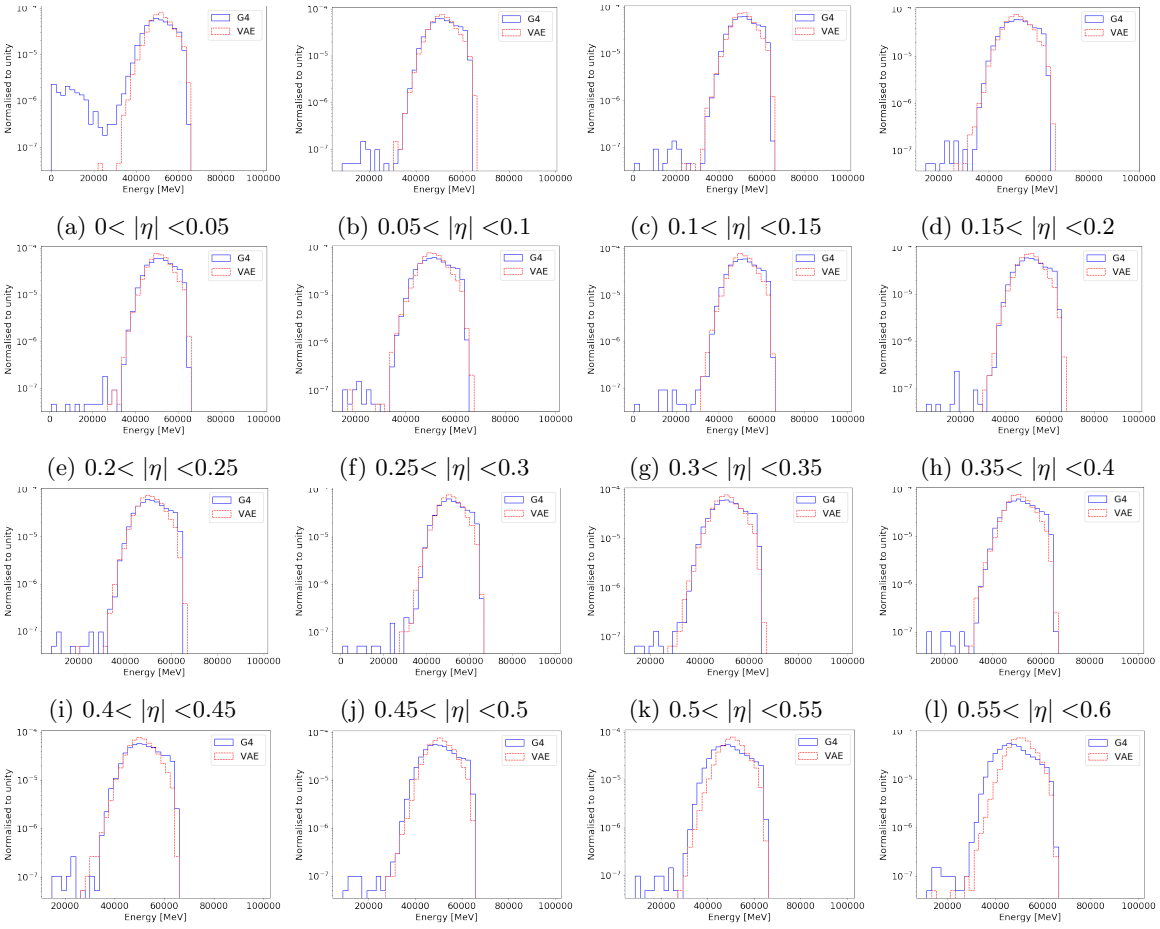


Figure 96: Fraction distributions of $EMB0/E_{Tot}$ (a), $EMB1/E_{Tot}$ (b), $EMB2/E_{Tot}$ (c), $EMB3/E_{Tot}$ and E_{Tot}/E_{Truth} (e) for photons with an energy of 65 GeV in the $0< |\eta| <0.8$ range. The full detector simulation (solid blue line) is compared to VAE (dashed red line).



(a) 0< $|\eta|$ <0.05 (b) 0.05< $|\eta|$ <0.1 (c) 0.1< $|\eta|$ <0.15 (d) 0.15< $|\eta|$ <0.2

(e) 0.2< $|\eta|$ <0.25 (f) 0.25< $|\eta|$ <0.3 (g) 0.3< $|\eta|$ <0.35 (h) 0.35< $|\eta|$ <0.4

(i) 0.4< $|\eta|$ <0.45 (j) 0.45< $|\eta|$ <0.5 (k) 0.5< $|\eta|$ <0.55 (l) 0.55< $|\eta|$ <0.6

(m) 0.6< $|\eta|$ <0.65 (n) 0.65< $|\eta|$ <0.7 (o) 0.7< $|\eta|$ <0.75 (p) 0.75< $|\eta|$ <0.8

Figure 97: Energy in EMB2 for photons with an energy of 65 GeV in the 0< $|\eta|$ <0.8 range. The full detector simulation (solid blue line) is compared to VAE (solid red line).

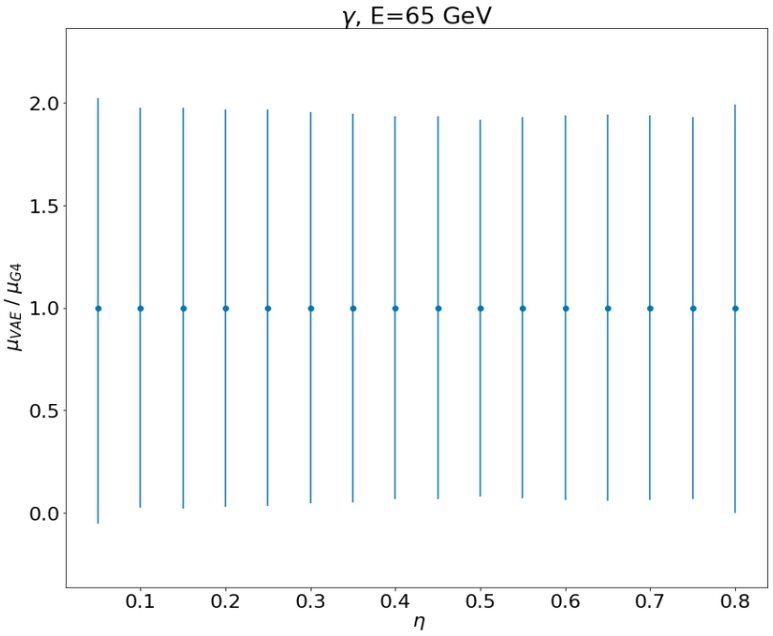


Figure 98: Ratio of the mean of the VAE total energy response to the full detector simulation for photons with an energy of 65 GeV as function of η. The error bar shows the ratio of the RMS of the total energy response.

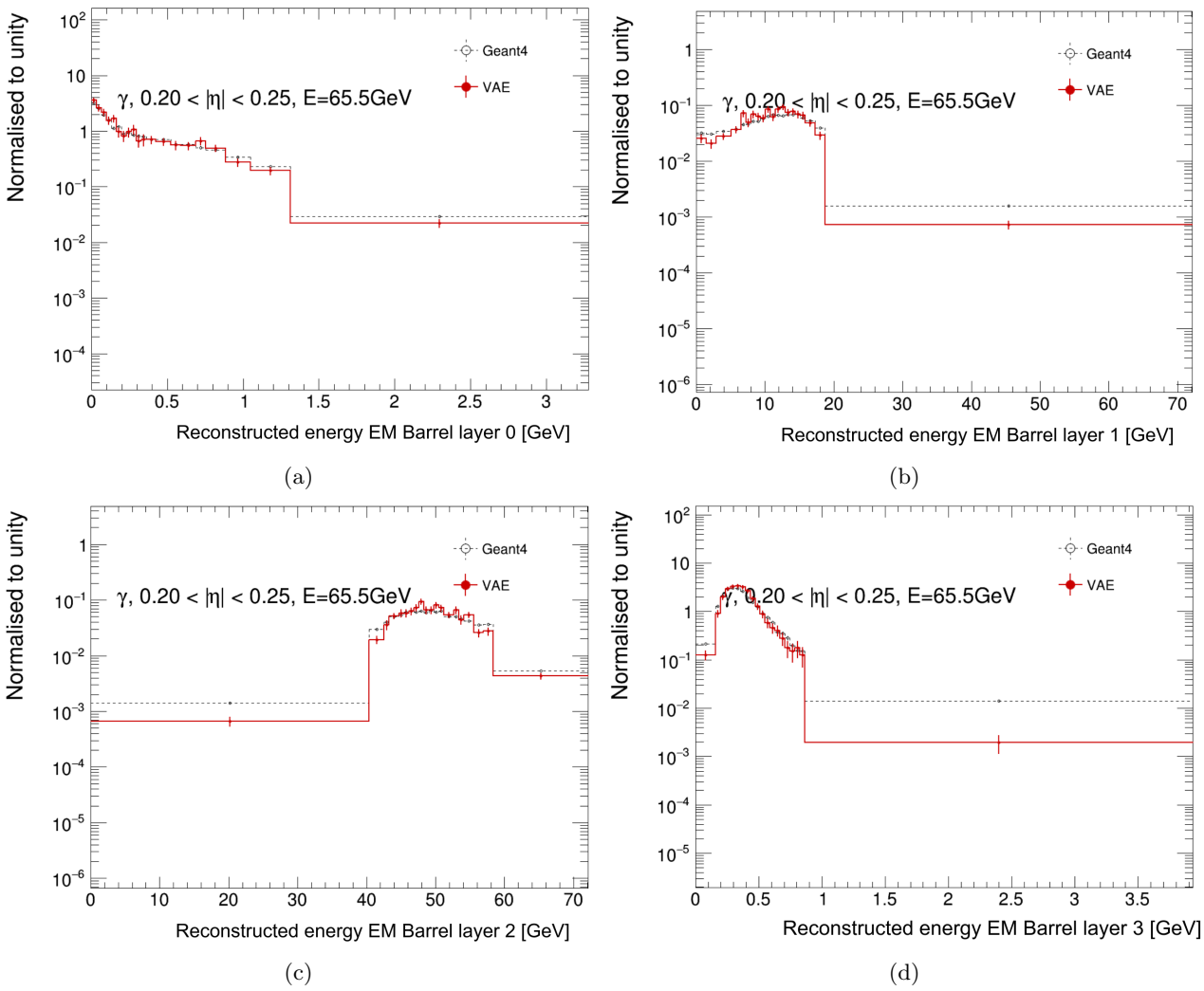


Figure 99: Reconstructed photon energy in the four ATLAS electromagnetic layers EMB0 (a), EMB1 (b), EMB2 (c), EMB3 (d) for photons with an energy of 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

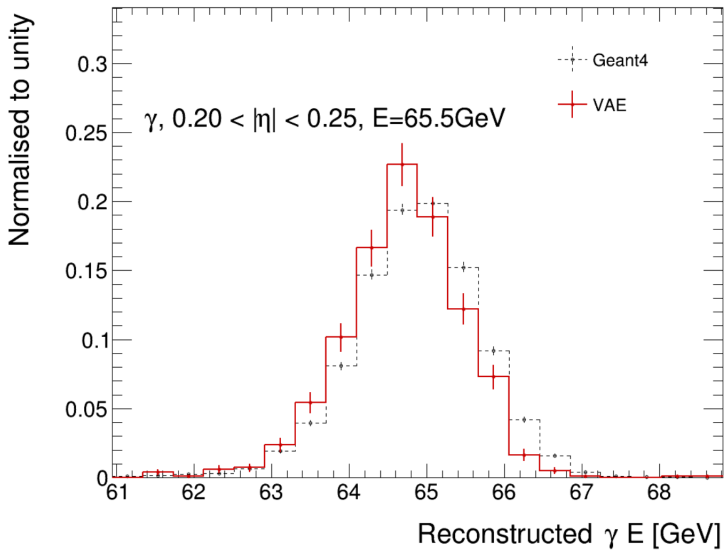


Figure 100: Reconstructed photon energy for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

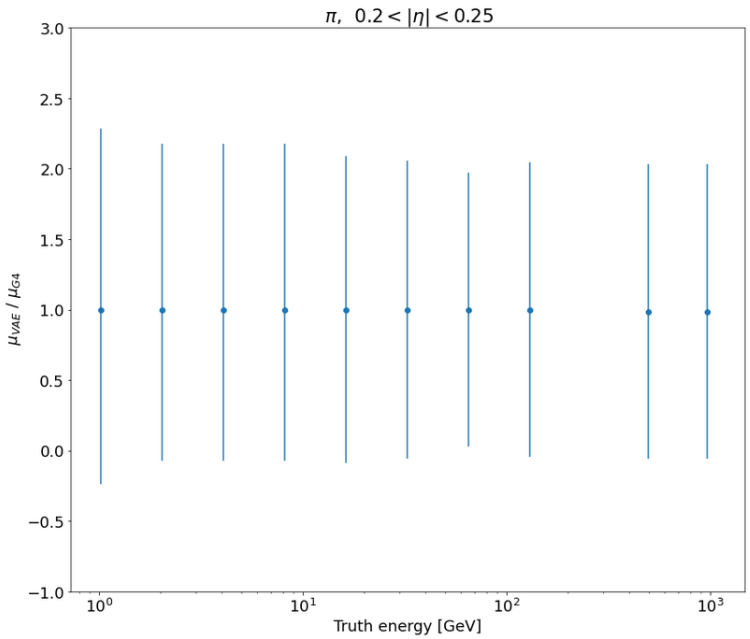


Figure 101: Ratio of the mean of the VAE total energy response to the full detector simulation for pions in the $0.2< |\eta| <0.25$ as function of the truth energy. The error bar shows the ratio of the RMS of the total energy response.

9.4 Summary and Discussion

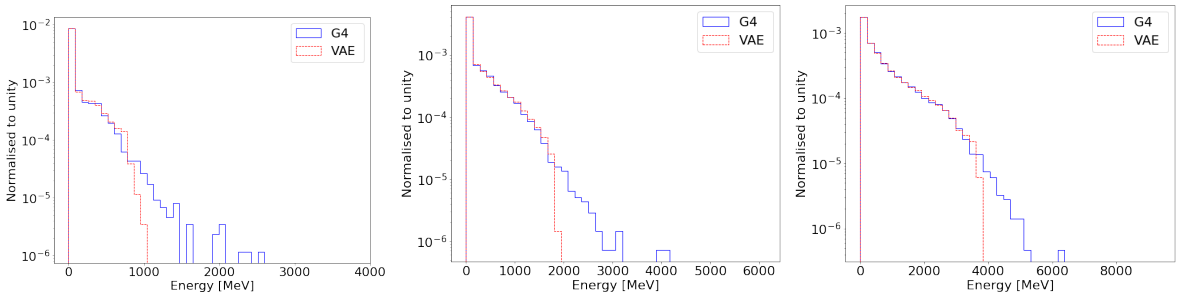
This chapter described the FastCaloVSim approach at the voxel level as an extension to the cell level based approach. This extension includes regions of the calorimeter, the energy, and the type of the particles. To accommodate the inhomogeneous cell granularities and the changing of layers, a voxelization procedure is proposed. It consists of grouping the spatial energy deposits of Geant4 hits into voxels in polar coordinates (r, α). For the particle type, a VAE model for pions is also presented in this chapter. For photons and pions the energy range of the truth particle is extended to be from 1 GeV to 1 TeV compared to 1 GeV to 262 GeV for the cell based model. The η range is extended to $0< |\eta| <0.8$, compared to a single η region in the previous chapter. Moreover, the VAE conditioning parameters were extended to include η. This conditioning on a detector parameter allows an important increase in the modeling coverage, where the VAE is now no longer constrained to a single η region.

A list of additional components are included in these VAE photon and pion models to further take advantage of available prior knowledge, such as the softmax output layer used for the energy ratios of voxels and layers. Using the softmax function, the output is mapped to probability space. In this case, the binary cross entropy loss is used instead of the mean squared error in the previous chapter.

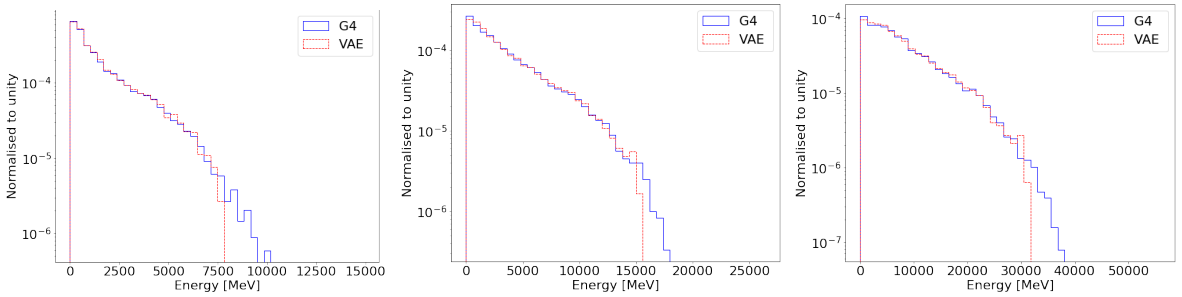
Overall, the voxel-level FastCaloVSim approach demonstrated a good modeling of Geant4 observables for photons and pions across energy and η.

The ultimate goal is to design a model that can reproduce showers in all η regions with $0< |\eta| <5$. This requires firstly deriving shower representation of energies in volume spaces. The definition of these representations influences the architecture of the model in terms of inputs, outputs, conditioning, and optimization. One way to proceed is to group into configurations the regions where we have similar layers and define one trained model, in addition to the energy and η, on these configurations, where they can be represented as a one hot encoding vector. This approach needs to first define the list of relevant layers and then group the similar regions of η into a single configuration. Table 9 summarizes an example of this grouping.

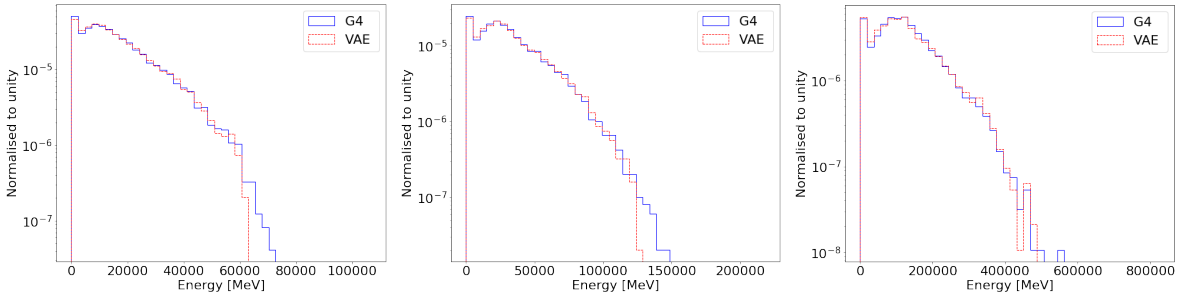
Defining a global input structure to the model that would fit the earlier definition is not straightforward. Since each η region has a variable number of layers, the resulting voxelization would have a non-uniform number of voxels. In other words, the training dataset has a variable number of features, which is a common problem in machine learning applications. In the next chapter, we address this format problem by proposing a novel method based on the K-means clustering algorithm.



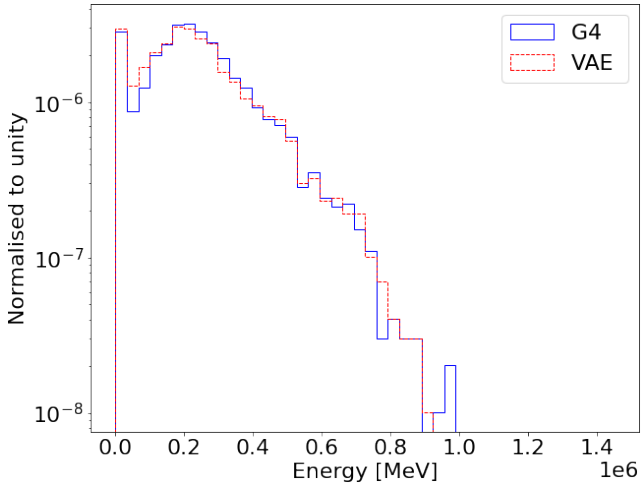
(a) 1 GeV (b) 2 GeV (c) 4 GeV



(d) 8 GeV (e) 16 GeV (f) 32 GeV

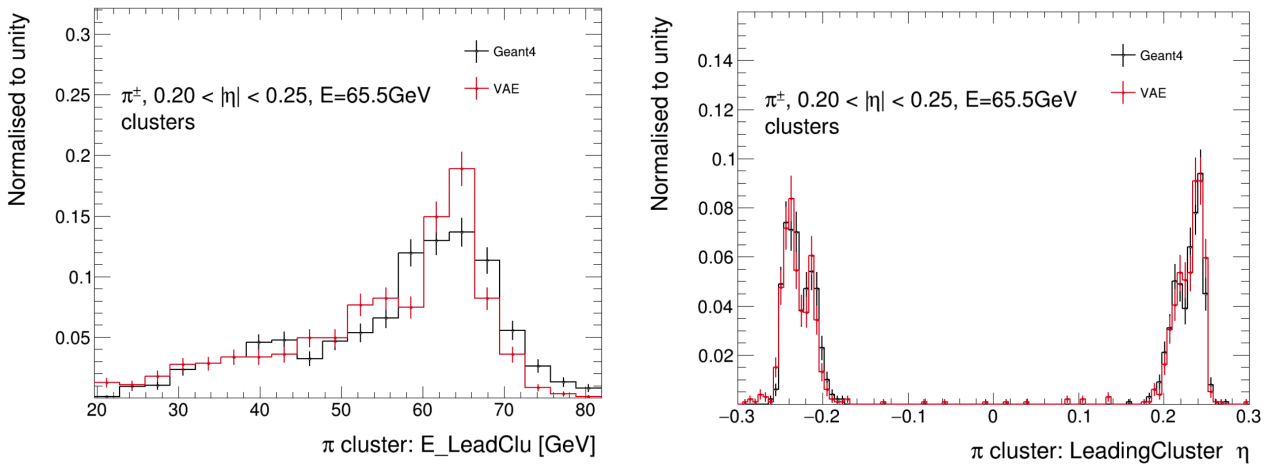


(g) 65 GeV (h) 132 GeV (i) 500 GeV



(j) 1000 GeV

Figure 102: TileBar0 distributions for pions with all the energy range from 1 GeV to 1 TeV in the $0.2 < |\eta| < 0.25$ region. The full detector simulation (solid blue line) is compared to VAE (dashed red line).



(a) (b)

Figure 103: Reconstructed energy of the leading cluster (a) and the learding cluster η for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

Configuration	η coverage	Relevant layers
1	0< $\|\eta\|$ <1.35	0,1,2,3
2	1.35< $\|\eta\|$ <1.5	0,1,2,3,4,5,6,17,18
3	1.5< $\|\eta\|$ <3.5	4,5,6,7,8,17
4	3.05< $\|\eta\|$ <3.25	6,7,8,21
5	3.25< $\|\eta\|$ <4.8	21,22
6	4.8< $\|\eta\|$ <5	21,22,23

Table 9: Proposed configurations with their η regions for the different relevant layers.

10 Centroid-level FastCaloVSim

Extending the model towards learning to generate showers in the whole calorimeter can not be easily done based on cells or (r, α) voxels. Centroid-level FastCaloVSim is a full parametrization of the ATLAS calorimeter, which provides a geometric structure for the Geant4 hits for training and generating showers. This parametrization is based on the ML clustering algorithm K-means. Figure 104 shows a schematic representation where the idea consists of applying the K-means on the Geant4 hits in order to build a set of K cluster centers or K centroids, where for each centroid there is a volume space that groups all the close by Geant4 hits.

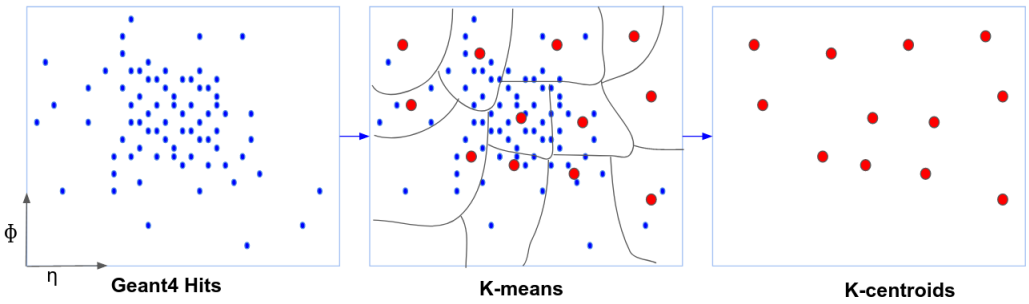


Figure 104: Schematic representation of deriving K-centroids from Geant4 hits using the K-means algorithm. The red points represent the K-centroids.

10.1 ML based Voxelization Procedure

The ML voxelization allows us to define a structured input per calorimeter layer to cover all the angular η regions. This voxelization is designed to be independent of the truth energy and the calorimeter geometry.

10.1.1 Cluster Analysis

Cluster analysis or clustering is an unsupervised ML approach to partition a large dataset of observations (data points) into subsets or clusters. One cluster represents an area of the dataset feature space where a set of observations share similar characteristics. This can be based on similarity or distance metrics between observations.

A comparison study between the most popular clustering algorithms is summarized in Table 10 and Figure 105. It allows us to select the most suitable algorithm based on the output cluster shape and the convergence time

As described in Section 6.6, the K-means algorithm groups the data points into K disjoint clusters, where each point is assigned to the cluster with the nearest mean. The center of the cluster is referred to as the centroid. The mean shift algorithm [197] groups the data points into clusters by shifting points towards the highest density of data points (mode) in the feature space. The spectral algorithm [198] formulates the clustering approach by constructing a similarity graph using a distance metric, where each vertex in this graph represents a data point. The hierarchical Ward, builds a hierarchy set of nested clusters using a successive merging or splitting. The agglomerative clustering [199] on the other hand, builds a hierarchy of clusters as a tree in a bottom up approach: one data point is one cluster, and the individual clusters are grouped successively until achieving the desired number of clusters. The Gaussian mixture technique is based on the expectation-maximization algorithm to learn a Gaussian mixture model from the input data points. Every data point is then assigned to the most probable Gaussian.

Table 10 summarizes the differences between the six algorithms cited above. The geometry is the mathematical definition where a Riemannian manifold is considered flat if its curvature is everywhere zero [200]. In two-dimensional space, for example, flats are points and lines. According to this definition of data geometry, the Geant4 hits in a two-dimensional space form a two-dimensional flat manifold in R^2. As a result, algorithms such as K-means or Gaussian mixture are more suited for deriving the clusters of these hits. Considering all the events, truth energies and η, the dataset becomes very large (order of millions) which requires the use of an adapted clustering algorithm offering the necessary scalability. The time for the algorithm to converge can also be a bottleneck for a clustering application.

Figure 105 shows the resulting clusters of the six algorithms using 600,000 hits from EMB2. The time, in seconds, is reported for each of the algorithms. The mean shift algorithm is not adapted for the definition of the ML voxels, since it creates clusters of dense areas of hits in two-dimensional feature space. The agglomerative clustering used is based on the average linkage and the L1 norm (sum of the absolute values) affinity as

Algorithm	Parameters	Scalability	Geometry	Clustering metric
K-means	Number of clusters (K)	Large dataset	Flat	Within-cluster sum-of-squares
Mean shift	Kernel bandwidth	Not scalable	Non-flat	Distances between points
Spectral clustering	Number of clusters	Not scalable	Non-flat	Graph distance such as the nearest neighbor
Hierarchical clustering (Ward)	Number of clusters or distance threshold	Large dataset	Flat and Non-flat	Distances between points
Agglomerative clustering	Number of clusters or distance threshold, linkage, type, distance	Large dataset when a connectivity matrix is jointly used	Flat and Non-flat	Any pairwise distance
Gaussian mixture	Many	Not scalable	Flat	Mahalanobis distance

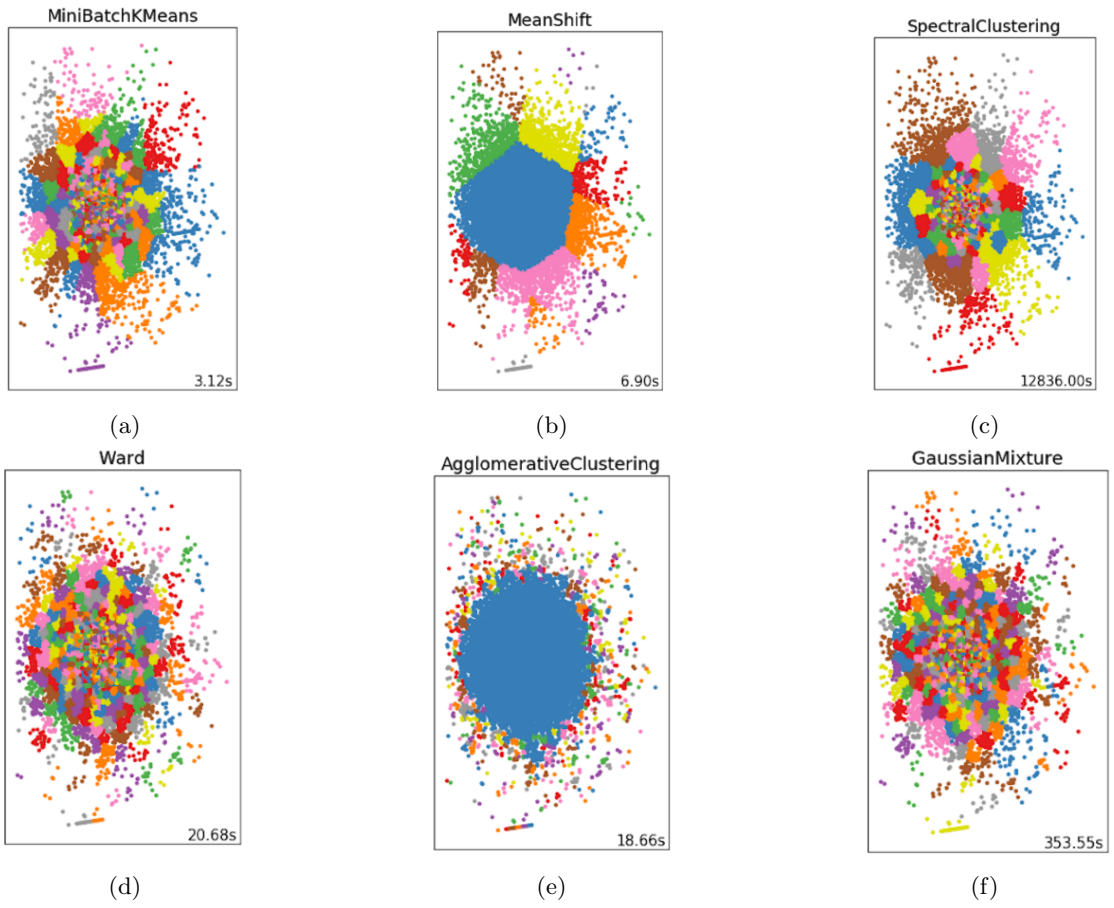
Table 10: Overview of the most well known clustering algorithms.

parameters. The linkage defines the distance computed between sets of data points. The algorithm uses this linkage to merge the sets, minimizing the distance. The average linkage refers to using the average of the distances of each data point of the these sets. The affinity, on the other hand, represents the metric used to compute the distance. This distance is computed for every pair of hits in the dataset, resulting in a quadratic complexity.

Similarly, spectral clustering is computationally very expensive, with a complexity of $O(N^3)$. Given the execution time of the various clustering algorithms on the considered sample, the optimal choice is clearly MiniBatchKmeans, a batch version of the K-means algorithm. However, CPU time is not the only selection criteria in this case. The algorithmic approach of the K-means procedure iterates on the dataset until finding optimal centroids. These centroids are particularly interesting since they can be used as clusters representatives in the ML voxelisation process.

10.1.2 Pipeline from Geant4 Events to VAE Simulated Events

Figure 106 summarizes the development pipeline for shower generation with VAE trained on preprocessed Geant4 showers using the K-means algorithm. Each of the rectangles represents an implemented function. **Geant4 reader** is a C++ function to read the Geant4 ntuples and saves only the hit's information in the relevant layers as flat ROOT trees. These saved files contain as well the calorimeter cell identifiers. **HDF5 builder** function saves 2D hits information of $d\eta \times d\phi$ per layer for all events per energy and η. The saved hits fall in the range [0,dr], where dr is the value in which 99.99% of the hit's energy is contained. **K-means applier** performs the K-means algorithm using the HDF5 files. It is applied to each of the relevant layers. In fact, the number of hits per event is of the order of millions, and combining all events from all energies and η slices results in billions of hits per layer. In order to reduce the number of statistics, only a subset of events is selected randomly per energy and η. The output of the K-means is the list of the centroids and for each of the hits in the input its corresponding centroid. Since we reduced the statistics to make the K-means run faster, the **Hits to centroids assigner** function allows assigning, for all the hits, their closest centroid from the output list of the K centroids. This derived ML shower representation is validated using the **validator** function. It is a set of two validation steps: a standalone and an Athena-integrated based. In the former, the Geant4 hits are assigned to the calorimeter cells using the cell IDs as a key to retrieve the relative cell positions. This assignment is compared to the Geant4 hits to centroids to cells assignment. The comparison at the cell level allows us to assess the performance of reproducing the detector response. The Athena-integrated step with a dedicated FastCaloVSim service allows us to validate the approach in a complex and complete environment. It is based on comparison of high level observables only available after reconstruction within the Athena environment. Since the model is trained on the full detector, the training is performed on a massive dataset and therefore in order to allow an efficient memory loading, the training generator input to the **VAE trainer** function is used. It consists of using TFRecord file format, a binary storage format of Tensorflow. Binary files occupy less space on disk and as a consequence take less time for I/O operations. The training generator, in this case, loads only a batch of events at the same time into memory.



(a) (b) (c)

(d) (e) (f)

Figure 105: Comparison of clustering algorithms. The same number of Geant4 hits 600.000 in EMB2 is used to apply each of the clustering algorithm. The convergence time for each algorithm is reported in the bottom right.

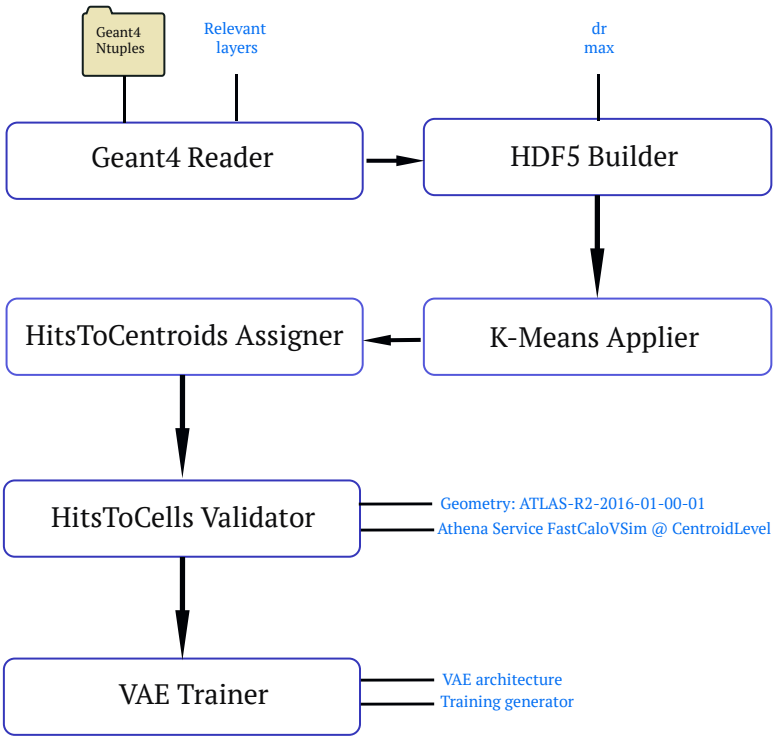


Figure 106: Development pipeline of the FastCaloVSim approach at the centroid level: from Geant4 showers to VAE simulated showers.

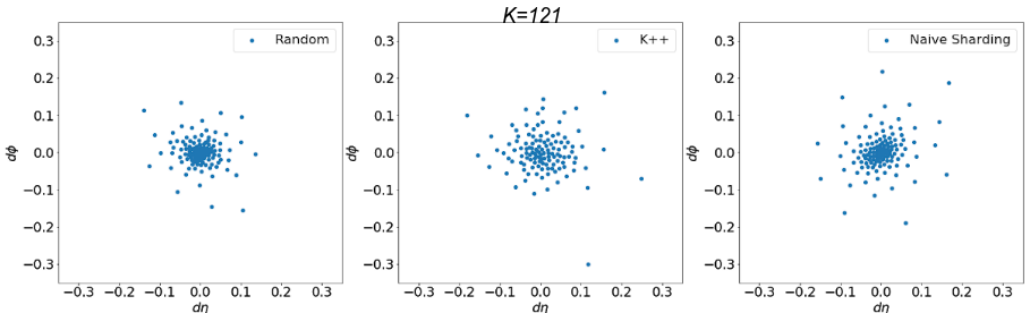
10.1.3 K-means Application, Challenges and How to Overcome Them

Let G be the geometry of the ATLAS calorimeter (see Figure 44): G is segmented into angular regions. η_i is the i^{th} angular region which contains a succession of l layers with i$\in$[1,100] and l not a fixed number, it depends on the η_i region. Counting all the layers in all the η_i regions gives a total of 24 layers. The idea behind using the K-means consists of deriving a 2D image per layer, where the image would be the same, independently of the η_i region. In order to satisfy this condition, the K-means is applied to cluster the Geant4 hits, where each hit is defined in a 2D space of $d\eta$ and $d\phi$. These values represent the relative positions with respect to the extrapolated position of the truth particle. They are clustered separately for each relevant layer of the calorimeter per particle type. Table 11 and 13 summarize the considered relevant layers for photons and pions respectively. Each centroid is in 2D, and it can be seen as a pixel in an image. For the energy values, afterwards, for each of the centroids, this is the sum of the hit energies belonging to the same cluster.

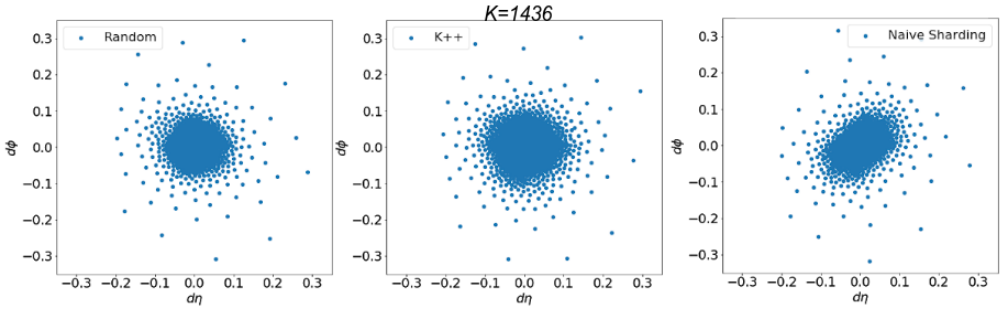
Technically, the K-means algorithm is slow. An alternative approach to reduce the computation time is the mini-batch K-means (shown in Figure 105). It is a K-means algorithm variant which uses mini-batches or subsets of the input data. These subsets are randomly sampled in each iteration of the algorithm. In all the following sections, the K-means refers to the mini-batch K-means.

In the literature, the performance of the K-means algorithm is shown to be dependent on two main factors: the initial cluster center's position and the instance order. In fact, the initial positioning of the centroids controls the number of clustering iterations until convergence, the better the initialization is, the faster and more efficient the algorithm is. There are different methods to set the initial centroids, such as a random initialization, which consists of selecting random K points from the input dataset. The K-means++ initializer [202] on the other hand, starts with a first centroid randomly selected, and the subsequent centroids are chosen from the input dataset using a proportional probability to the squared distance away from a given point's nearest existing centroid. Another initialization method is naive sharding which is based on computing a summed value representing all the attribute value of a point. This value allows sorting all of the points of the input dataset. This data is accordingly divided in a horizontal way into K parts or shards. The initial attributes of each shard are summed independently and their respective mean value is calculated. The mean values represent the centroids.

The three initializers are tested on Geant4 hits in EMB2. Figure 107 shows the three types for low and high

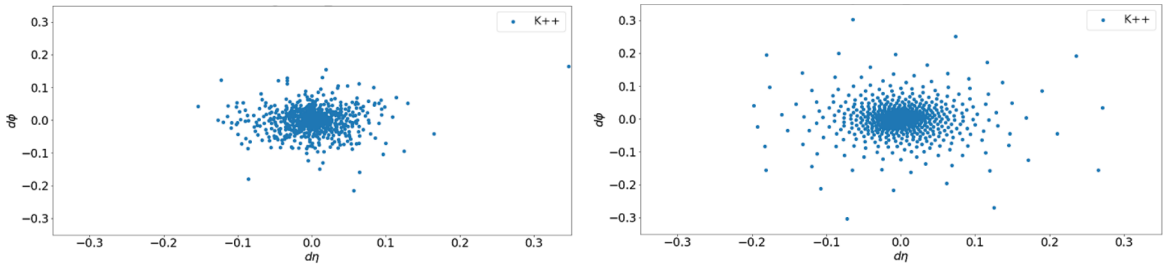


(a) K=121



(b) K=1436

Figure 107: K-means outputs using three different initializers with K=121 (a) and K=1436 (b).



(a) (b)

Figure 108: Output of the K-means clustering with a batch size of (a) 100 and (b) 1000.

values of K. The intuition behind finding the best approach is to spread out the initial centroids to cover as well the hits in the outer regions. Using K=121, both random and Naive sharding techniques have a high density of centroids in the core region and less in the outer region. K++ on the other hand, spreads better the centroids. For higher K, the same pattern is present. In fact, the K++ method, tries to separate the centroids from each others.

One of the important features when applying the K-means approach is the size of the subset of the data points at each iteration, known as the batch size. Figure 108 shows the impact on the final clustering output. It all relates to how much variety of data points the algorithm can use to derive the center of the clusters. The variety includes hits in the outer region in order to better cluster the hits space and get a correct mapping to cells afterwards. This shows that having a large size would lead to wider spread of the centroids because the batch set contains more diverse hits than the small batch. The idea consists of carefully choosing the batch size, not too small to avoid ignoring relevant hits forming the tails and not too big to avoid the time-consuming convergence.

10.1.4 From K Clusters to K Voronoi Polygons

The centroids can be visualized using the Voronoi diagram. It is an encoding structure of proximity information. It allows us to visualize the centroids and their respective areas, where each area forms a polygon. Its construction algorithm is based on half plane intersection repeated for every centroid in the input. Figure 109 shows the construction of a Voronoi polygon: $C1, .., 5$ represent five centroids, the blue lines are the half plane intersections between centroids and the red triangle is called a Voronoi cell or polygon. It defines a set of points that are close to each centroid. Using this diagram, finding the closest centroid to a point is easy by

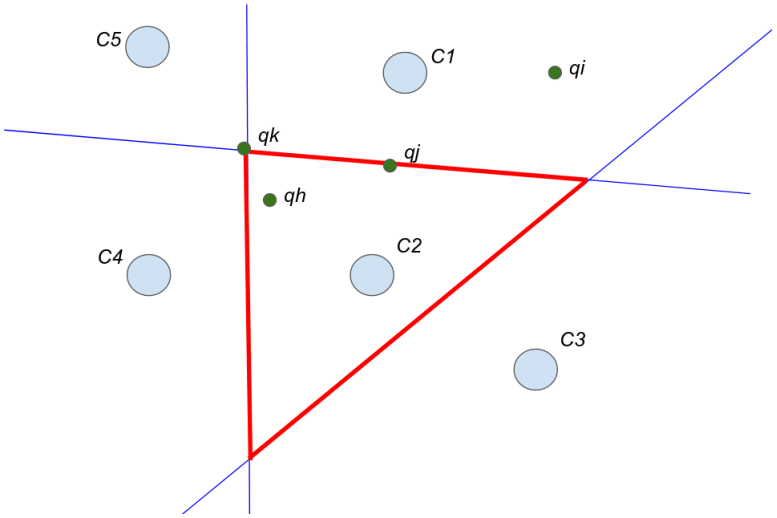
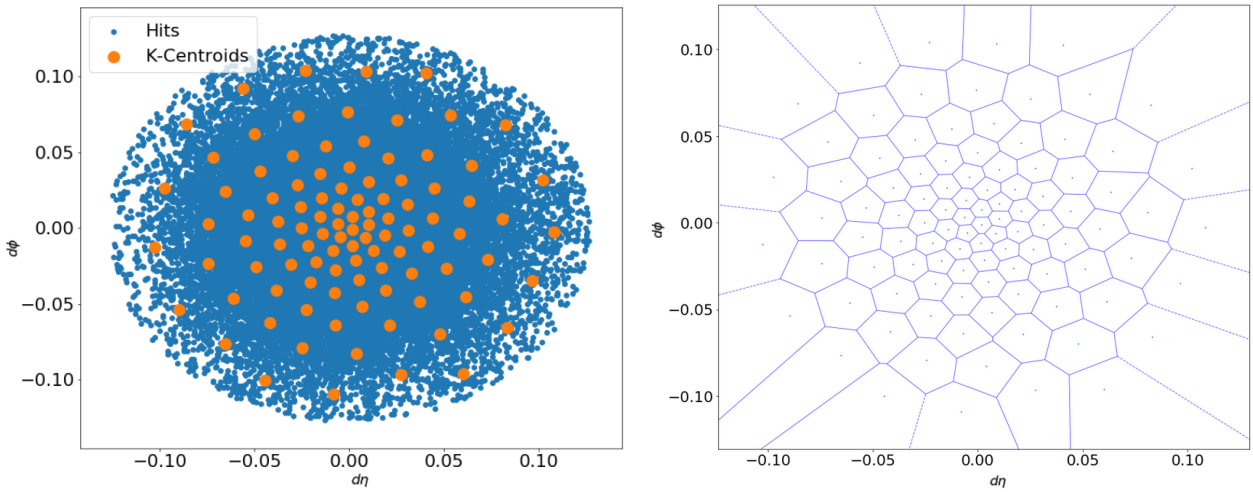


Figure 109: Schematic representation of a Voronoi diagram. C_i represent the clusters centroids and q_i the data points. The blue lines denote the half plane intersections, while the red triangle defines the Voronoi polygon.



(a) (b)

Figure 110: (a) K-means (K=100) applied on Geant4 hits of 100 photon showers in EMB2 (b) Voronoi polygons constructed using the centroids from the K-means application.

construction: the closest centroid to qi is $C1$, and for qh is $C2$. Points on the edge have the same distance to both centroids, as is the case for qj. This case can be extended, if a point is located at the intersection of many Voronoi polygons, such as the qk point.

Figure 110(a) shows 100 centroids derived from the K-means and Figure 110(b) represents the Voronoi polygons for each of the K centroids. The area of the polygons gets larger when moving out from the center (0,0). This is explained by the fact that in the core the number of centroids is higher and therefore building half plane intersection between close by centroids forms a smaller region of space compared to distant centroids.

10.2 Centroid-level FastCaloVSim for Photons

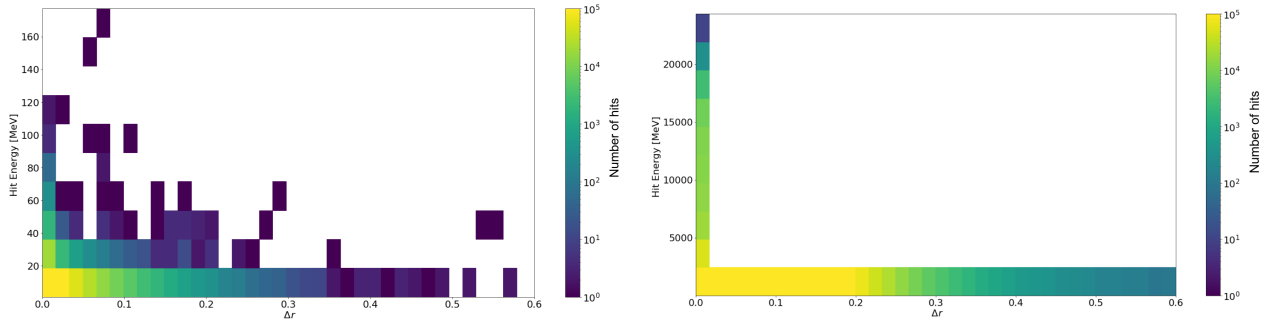
The training process of the VAE to simulate showers in the full detector starts with photon particles because they are characterized by a simpler shower development compared to pions. The number of centroids is defined with an optimization search technique based on a two-step validation as explained in Section 10.1.2.

10.2.1 Data Preprocessing

Table 11 details the relevant layers for photons and for each layer the corresponding η slice where the layer is considered as relevant. Therefore, for photons, the K-means is applied for each of these layers.

Relevant layer	η coverage
0	$0< \|\eta\| <1.55$
1	$0< \|\eta\| <1.55$
2	$0< \|\eta\| <1.55$
3	$0< \|\eta\| <1.4$
4	$1.35< \|\eta\| <1.85$
5	$1.35< \|\eta\| <2.7$
6	$1.35< \|\eta\| <3.5$
7	$1.50< \|\eta\| <3.5$
8	$1.55< \|\eta\| <3.45$
9	$3.2< \|\eta\| <3.4$
12	$0< \|\eta\| <1.05$
17	$1.25< \|\eta\| <1.75$
18	$1.3< \|\eta\| <1.55$
21	$3< \|\eta\| <5$
22	$3.25< \|\eta\| <5$
23	$4.85< \|\eta\| <5$

Table 11: List of relevant layers for photon particles with their respective η coverage.



(a) (b)

Figure 111: Hit energy as function of dr. The colors represent the density of hits per rings of dr for photons with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$ with an energy of (a) 1 GeV and (b) 1 TeV.

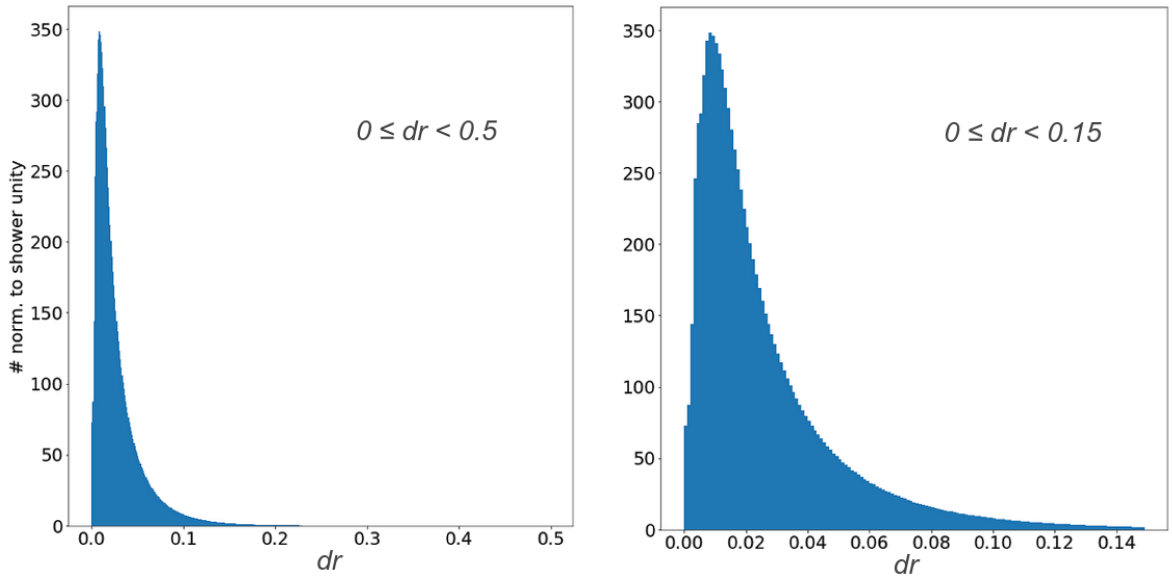
Let (0,0) in $d\eta \times d\phi$ be the center of the shower per layer (Figure 110 illustrates centered showers in EMB2). The energy of the shower develops from center to edge. This can be interpreted as a development in a circle, forming rings of dr. The innermost dr contains most of the energy deposition, with most of the hits entries. Figure 111 shows hits energies in MeV units as function of dr with a color coding for the number of hits in each dr for photons of 1 GeV and 1 TeV in $0.2 < |\eta| < 0.25$ in EMB2. Since the outermost dr rings contain very little energy deposition and in order to reduce the number of hits inputs to the K-means, the Geant4 hits are chosen in a dr which contains more than 99% of the energy per particle energy and η. Figure 112 shows the dr distribution normalized to the number of showers, with two cuts on 0.5 and 0.15. The dr cut is based on quantile definition from the distributions of dr weighted by the energy of the hits. This selection considers all the energies and all η. Table 12 shows an example of an η slice, the dr values of four quantiles with a low, medium, and high energies of 1 GeV, 65 GeV and 1 TeV respectively. In order to globally contain all the hits, a dr value of 0.5 is used.

Quantile	1 GeV	65 GeV	1 TeV
0.995	0.135	0.102	0.096
0.998	0.171	0.129	0.123
0.999	0.204	0.156	0.144
0.9999	0.455	0.244	0.227

Table 12: dr with different quantile values for 1 GeV, 65 GeV and 1 TeV for photon particles with $0 < |\eta| < 0.05$.

10.2.2 Validation Approach

In the K-means algorithm, finding the optimal number of centroids K per layer is an optimization problem. The sum of the K values for all the layers would define the structure of the input and output layers of the VAE



(a) (b)

Figure 112: dr distribution normalized to shower unity with (a) $0 \leq dr < 0.5$ and (b) $0 \leq dr < 0.15$.

model and the subsequent optimized number of nodes per hidden layer. Minimizing K allows as well to minimize the number of trainable parameters. The validation approach consists of testing how well the ML voxelization can accurately represent Geant4 samples. It is based on a series of validation plots from an ML to a physics perspective and from a standalone to Athena-based implementation, including cell-to-cell, event-to-event and distribution-to-distribution comparisons.

From an ML perspective, evaluating the cluster quality allows us to get an idea of how meaningful the clusters are. A good clustering is the one where data points of a cluster are close to each other and far from the other clusters. The inertia is one of the most well known metrics for quality evaluation. To determine an optimal number of centroids, the inertia metric defines the optimal value based on the elbow, the point after which the score starts to decrease in a linear way. In other words, it represents the inflection point on the curve at which increasing the K value leads to a very small decrease in the inertia score. This point indicates that the model performs its best fit. Figure 113 shows the inertia values as a function of the number of clusters. Fourteen K values and their corresponding inertia values are shown in this figure. The elbow value corresponds to the value 100.

The ML validation is not enough alone. The right hit energy has to be mapped to the right cell. Therefore, the validation is also based on a mapping comparison. A mismapping can happen when a Voronoi polygon is larger than a calorimeter cell. This means that the centroid of this Voronoi polygon is closer to the center of a wrong calorimeter cell than the correct one. This mismapping can also happen when the centroid falls at the edge of a cell. Figure 114 shows an example where (a) represents an annotation of calorimeter cells with an energy deposition in EMB3 for one event of 65 GeV in $0.2 < |\eta| < 0.25$. The three blue points in (b) are Geant4 hits which are mapped to the cell number 15. Applying K-means on the Geant4 hits in EMB3 with K=100 results in having large Voronoi polygons than the calorimeter cells in the outer region. This leads to a wrong hit assignment where the third hit is not mapped to cell 15 but instead to cell 16. This wrong assignment would cause an unbalance in the energy per cell and therefore the shower shape variables.

Physics wise, defining the resolution to simulate showers and validating the K-means approach does not rely on a single metric. It instead consists of finding the optimal K with a minimal distortion in the shower shape distributions when comparing the mapping from hits to cells against hits to centroids to cells. The optimization is also based on comparing the area of cells versus the area of the Voronoi polygons to fulfill the condition of deriving higher granularity Geant4 shower representations than the cells. This last comparison can be visualized in Figure 115, where the Voronoi diagram is shown for two different K values along with the calorimeter cells (red points represent the center of the cells) in EMB2 from a single event of photons with an energy of 132 GeV in $0.8 < |\eta| < 0.85$. With a low K value, this representation is much closer to the cells where a polygon has almost a cell area. On the other side, with K=1000, we can clearly see that a number of polygons can be grouped together to be mapped into a single cell.

One of the first quantities to look at is the energy ratio deposition in the core 7×7 cells in EMB2. Figure 116 shows the ratio values using two different values of K, 100 and 300 for photons with 65 GeV energy in

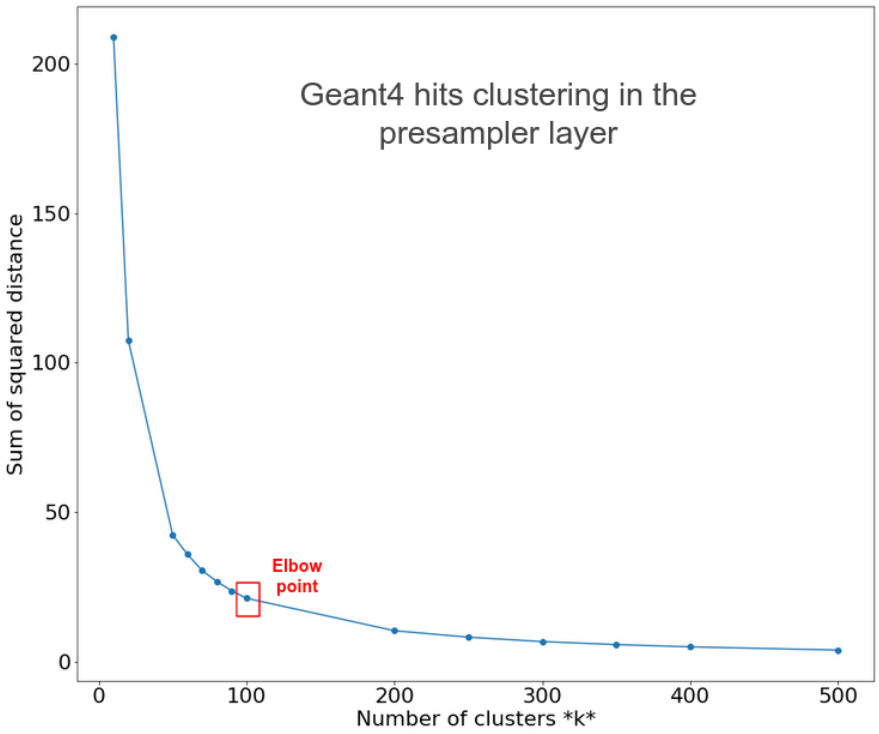
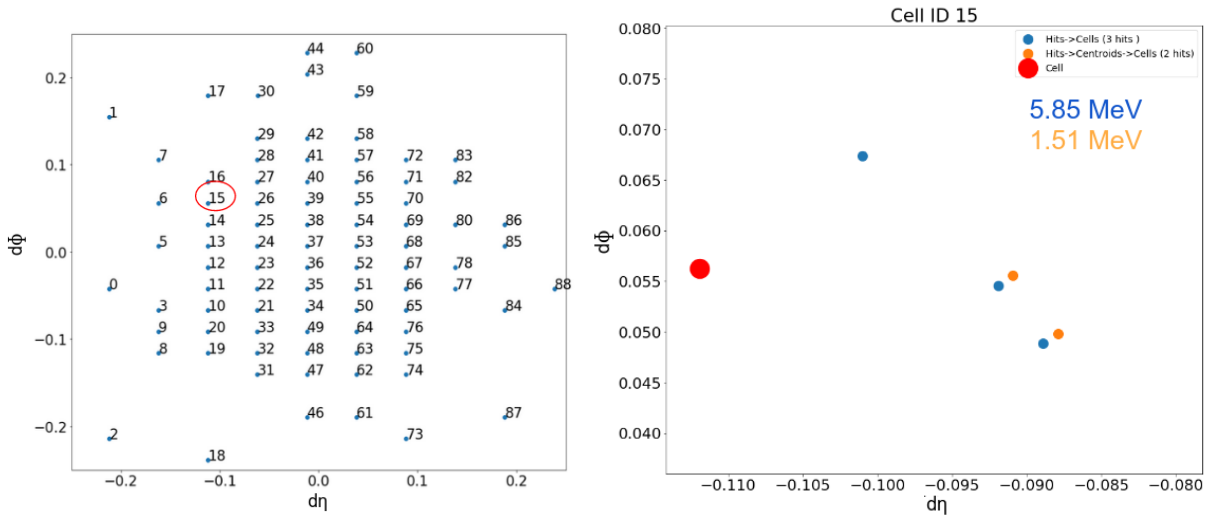
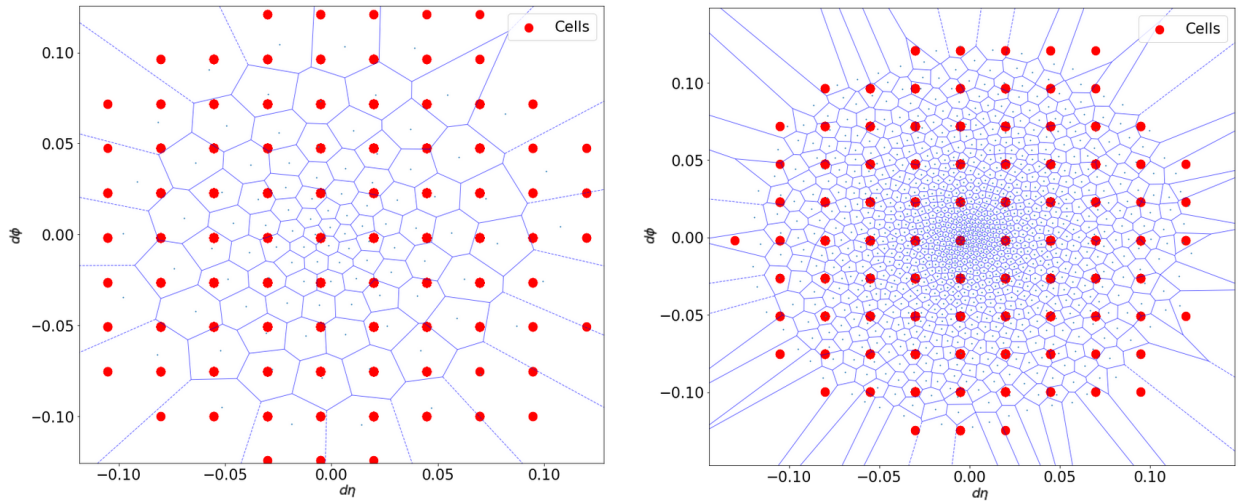


Figure 113: Inertia values as function of the number of clusters for K-means clustering algorithm applied on Geant4 hits in the presampler layer .



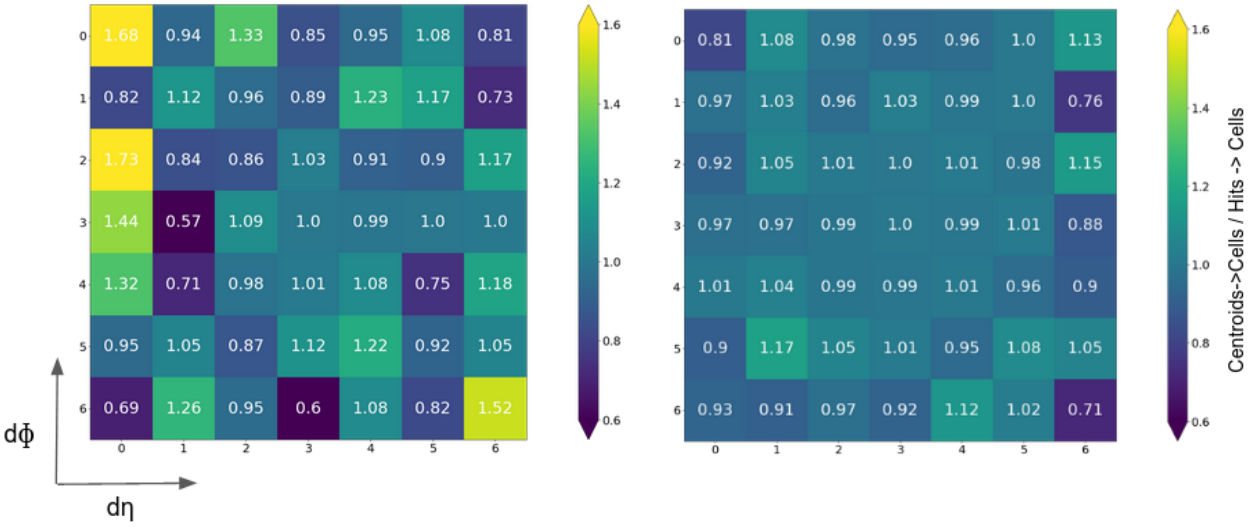
(a) (b)

Figure 114: (a) Cell annotation for a single event in EMB3 (b) Hits in cell 15. The values reported in (b) represent the sum of hit energies. The red point represents the center of the cell 15. The shift between the blue and orange points is added manually for visibility.



(a) (b)

Figure 115: Voronoi polygons with (a) K=100 and (b) K=1000. The red points are the center of cells of a single photon event in EMB2.



(a) (b)

Figure 116: Ratio of energy deposition of (centroids to cells) to (hits to cells) in 7×7 cells in EMB2 with (a) K=100 and (b) K=300

$0.2 < |\eta| < 0.25$. Having more granular definition allows us to accurately assign the energy in the correct cell, as opposed to K=100 where the ratio values are not close to 1. This is due to the large size of the Voronoi polygons compared to cells specifically in the outer region from the core as shows the Figure 115.

The standalone validation relies also on distributions of $d\eta$ and $d\phi$. Figures 117 and 118 represent the distribution of $d\eta$ and $d\phi$ respectively weighted by the energy of the cell. The plots show the weighted distributions for three different K values. K=100 is clearly to be discarded since it can only reproduce the core, not the tails for both $d\eta$ and $d\phi$ distributions. This is also reflected on the $\chi 2$ statistic in the legend, where it is higher for K=100. The $\chi 2$ statistic used in the $\chi 2$ test is computed as $\chi 2 = \sum_i \frac{(O_i - E_i^2)}{E_i}$, where O_i are the observed values and E_i are the expected values.

Figure 119 shows the $\chi 2$ of event to event comparison as a function of the truth energies from 1 GeV to 1 TeV for two different K values. This figure shows that having a larger K leads to a better agreement, and the decrease in the $\chi 2$ range values is visible.

To further assess the quality of the centroids definition and for optimization purposes, the FastCaloVSim service in Athena is implemented to take the centroids as hits and assign them to calorimeter cells. The optimization is performed per relevant layer.

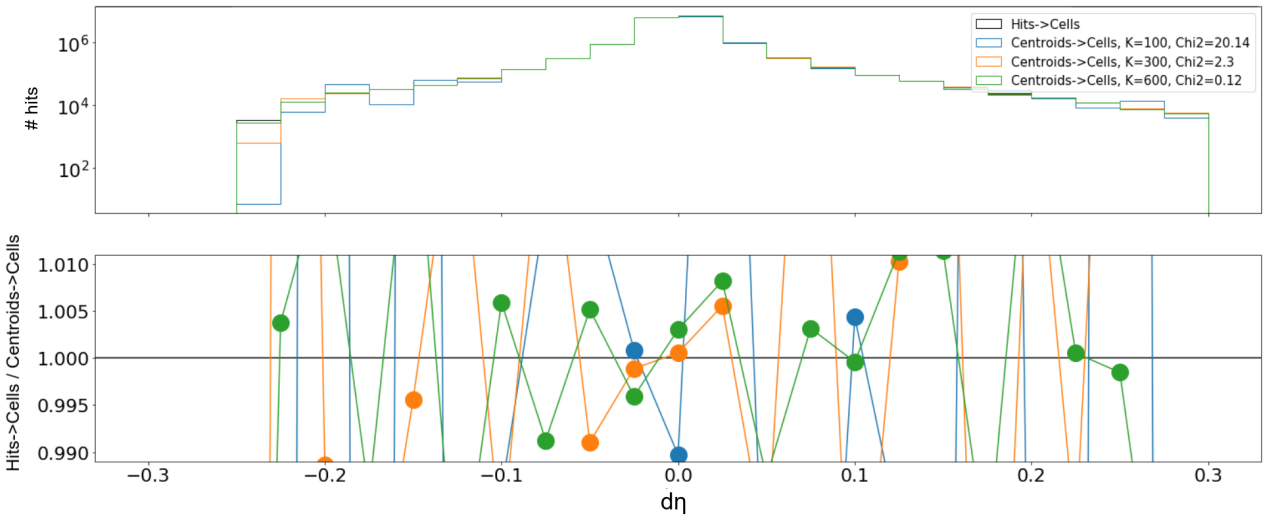


Figure 117: $d\eta$ distribution weighted by hit energy. The full detector simulation of hits mapped to cells (black line) is compared to centroids to cells mapping with K =100 (blue line), K=300 (orange line) and K=600 (green line). The ratio plot is zoomed within 1% from the reference ratio 1.

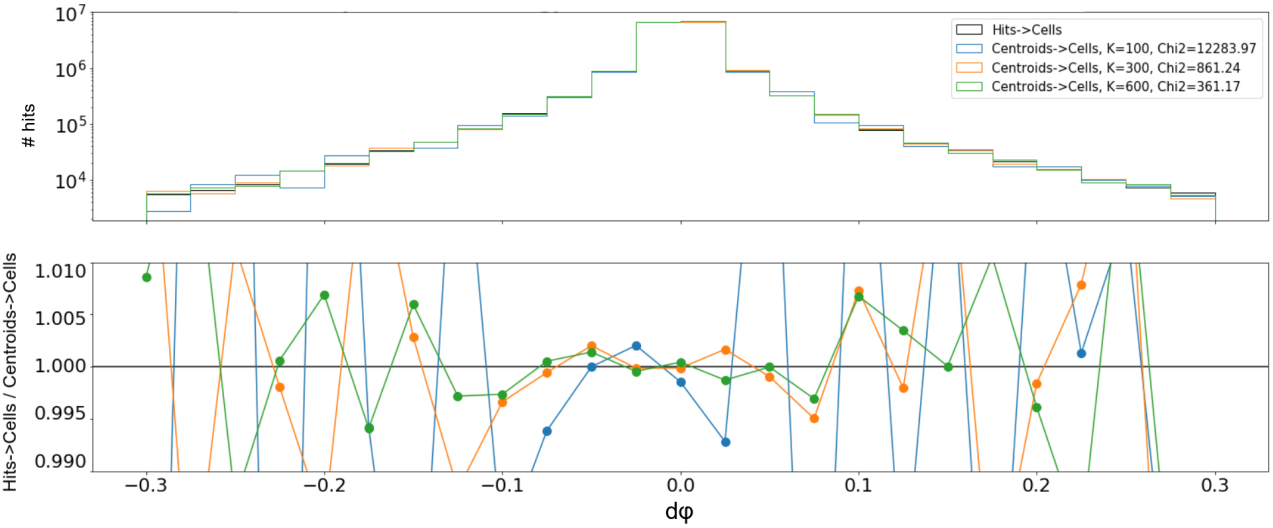
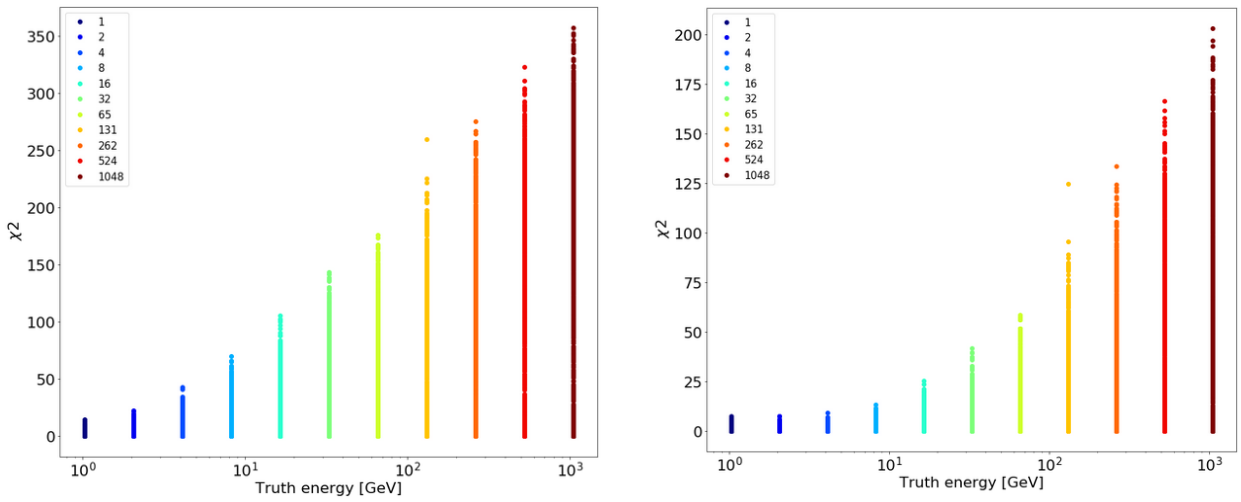
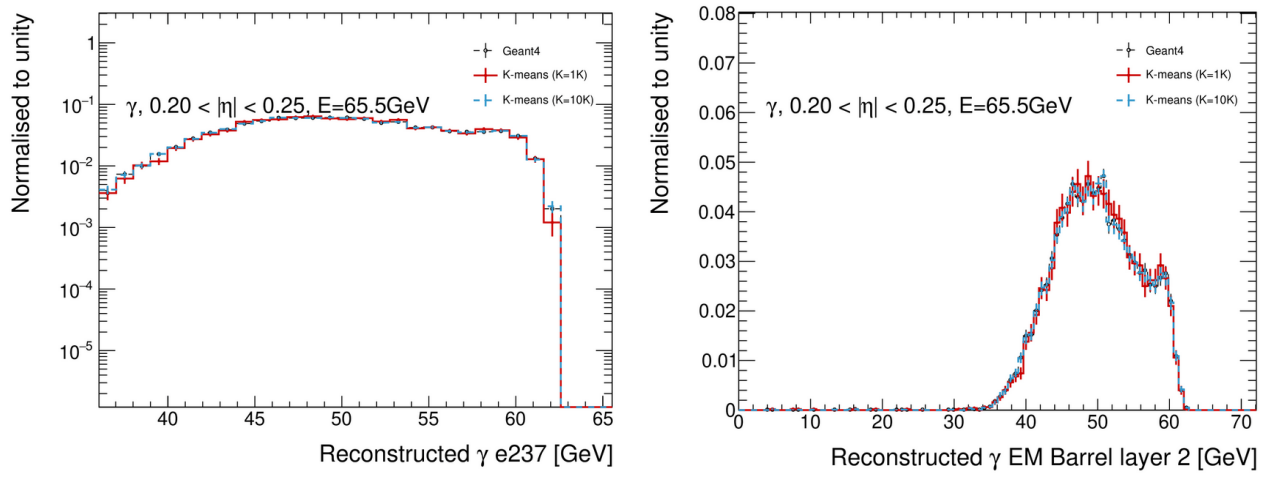


Figure 118: $d\phi$ distribution weighted by hit energy. The full detector simulation of hits mapped to cells (black line) is compared to centroids to cells mapping with K =100 (blue line), K=300 (orange line) and K=600 (green line). The ratio plot is zoomed within 1% from the reference ratio 1.



(a) (b)

Figure 119: χ^2 of event-to-event comparison of hits mapped to cells compared to centroids to cells as funtion of the truth energy (a) K=100 and (b) K=300. The legend shows the color coding for the truth energy values going from 1 GeV to 1048 GeV.



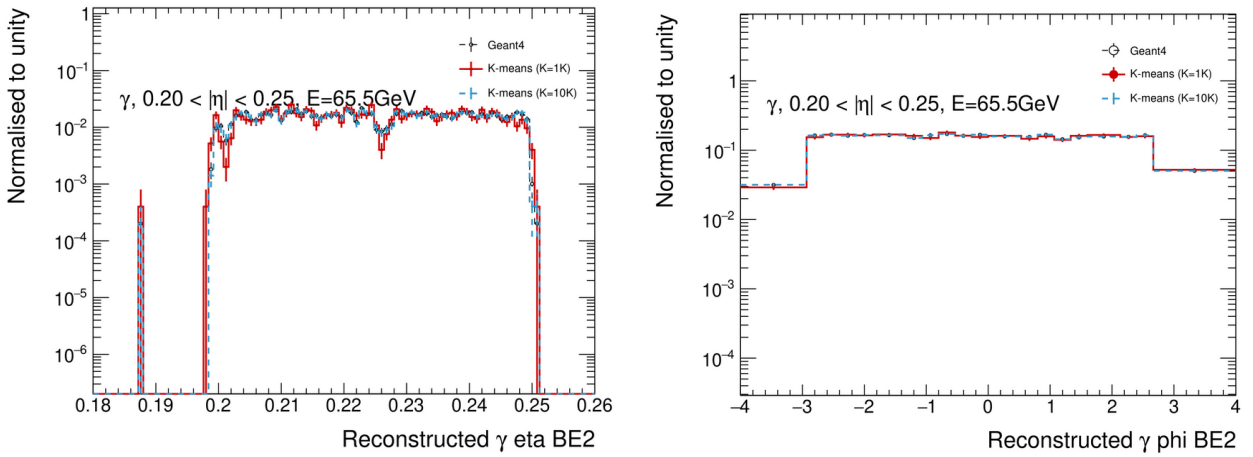
(a) (b)

Figure 120: Reconstructed (a) energy in 3×7 cells in EMB2 (e237) and (b) energy in EMB2 for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a K-means with K=1k (solid red line) and a K-means with K=10k (solid blue line).

A more efficient approach adapted for optimizing K per layer is reversed: instead of starting with a low K value and increasing it to find the optimal value, the reverse optimization approach consists of setting K to a very high value such as 10000, very granular to perfectly reproduce the Geant4 hits and then decreasing K (with a step of 100) until reaching an optimal value with only a small deviation from Geant4 distributions using metrics such the χ2 statistic. Generally, distributions representing only energy quantities are well reproduced with lower K values. The validation is then more based on the shower shape observables. Figures 120, 121 and 122 represent distributions for the Athena-integrated validation for K=10000 where the Geant4 agreement is perfect. The K value for EMB2 is chosen to be 1000. It represents a good trade-off between the number of centroids and the accuracy of Geant4 agreement. The same process is applied for all the relevant layers defined in Table 11.

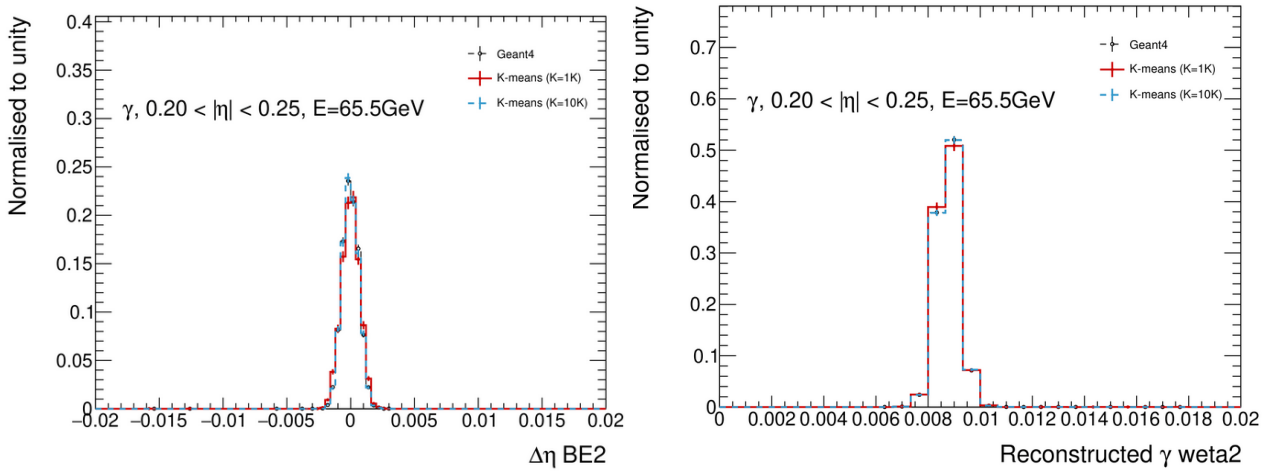
10.2.3 Model Design and Training Procedure

FastCaloVSim at centroid level is trained using all the relevant layers defined in Table 11. The K values based on the above optimization approach are 200, 1000, 1000, 200, 500, 500, 500, 500, 500, 200, 500, 500, 500, 500, 500, 500, 500 for layers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 17, 18, 21, 22 and 23 respectively. This results in having



(a) (b)

Figure 121: Reconstructed (a) η and (b) ϕ for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a K-means with K=1k (solid red line) and a K-means with K=10k (solid blue line).



(a) (b)

Figure 122: Reconstructed (a) $\Delta\eta$ and (b) weta2 for photons with an energy of approximately 65 GeV in the range $0.20 < |\eta| < 0.25$. The energy depositions from a full detector simulation (black markers) are shown as reference and compared to the ones of a K-means with K=1k (solid red line) and a K-means with K=10k (solid blue line).

a total of 8600 centroids per photon shower. Following the same VAE design at cell or voxel level, the VAE learns to reconstruct the shower vector containing all the 8600 centroids energy ratios with one entry for the total energy fraction and 16 entries representing the energy per layer fractions. Therefore, the VAE model for photons at centroid level is trained to reconstruct 8617 nodes of energy ratios. In a particular η region, for all showers where the layer is not considered as relevant, the ratio values of the centroids of that layer are set to be zero. By setting this value, all showers independently of the truth energy and the η region have the same structure with 8617 values. The design of the model's architecture is similar to the (r, α) voxel model in Figure 94 in terms of number of layers, layer structure and activation functions. The model is also conditioned on both energy and η with $0< |\eta| <5$. The number of nodes per layer, derived from a hyperparameter tuning, is set to be 1000, 500, 200, 100 for the encoder's hidden layers and the reversed order for the decoder. The encoder learns to represent a shower at the centroid level in 100D. To generate new showers, the decoder (as a generator) would be fed with 100D vector of uncorrelated Gaussians along with the conditions of energy and η.

10.2.4 Generation Performance

To assess the generation performance of the VAE, both steps of standalone and Athena based validations are presented in this section.

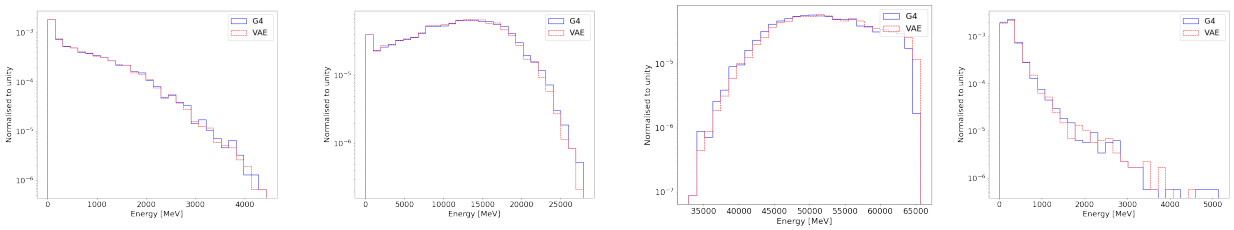
Figures 123, 124 and 125 represent the performance of the VAE compared to Geant4 for particles with 65 GeV in $0.6< \eta <0.65$, $2.95< \eta <3$ and $4.65< \eta <4.7$ respectively. All the relevant layers (16 for photons) and the total energy, defined as the sum of the energy in all the centroids, distributions are reported. The VAE is able to learn and reproduce the energy distributions across layers and η slices with a mismodeling for the FCAL of the total energy caused by deviations in the energy of layer 22 which is an impact of a low statistics regime. One of the key important checks is the distribution of energies per layer where the layer is not relevant for the given η region. In the training inputs, the energy values of these layers are set to zero. Figure 123 for example, shows the distributions for EME1 and FCAL0 layers which are both non-relevant layers in the slice $0.6< |\eta| <0.65$.

Other total energy distributions for particles with 65 GeV and 524 GeV energies for six different η slices are shown in Figure 126 and 127 respectively. In general, the total energy distribution generated by the VAE is observed to agree with Geant4. Moreover, the overall performance of the VAE for photons with 65 GeV and 1024 GeV energies is summarized in Figure 128. The ratios of the means and RMS of the total energies of VAE to Geant4 are shown as function of η. Overall, the VAE can reproduce all the total energies across the η regions with a similar level of agreement.

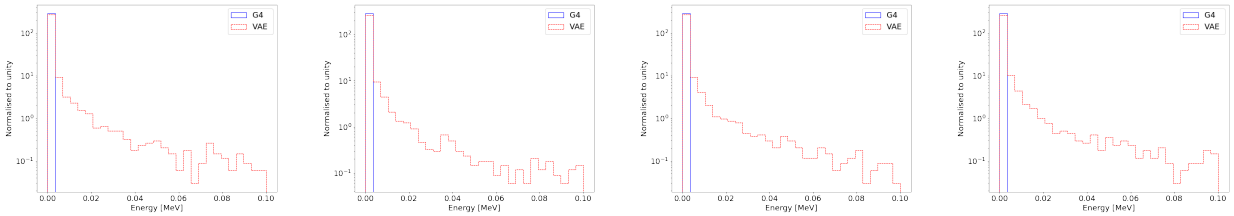
For the performance of the 1 TeV particles, the main η regions where the VAE reproduces larger RMS than those in the full simulation, are the ones close to the transition regions such as the case for η close to 1.7. Figure 129 (b) and (e) shows the total energy distributions for these regions. The Geant4 distributions of total energy in these regions are non-Gaussian and therefore show a different shape characterized by a double peak compared to neighboring distributions to $\eta \approx 1.7$ and $\eta \approx 2.5$ in Figure 129. The transition regions represent the central end-cap calorimeters and the end-cap forward calorimeters, which both are characterized by a reduced calorimeter coverage. In fact, there are many complex factors that can influence the performance in these regions such as the change of geometry that can produce a sample that is not completely uniform

An Athena FastCaloVSim service is implemented to simulate energy depositions of the centroids. Similarly to the cell or voxel level services, the generator simulates an energy ratio value for each of the centroids for all the relevant layers considered for photons. By rescaling to the truth energy and then the total energy per layer, this latter value is used to rescale the centroid's value. Each of the centroids is considered as a hit, and it is assigned to an ATLAS calorimeter cell.

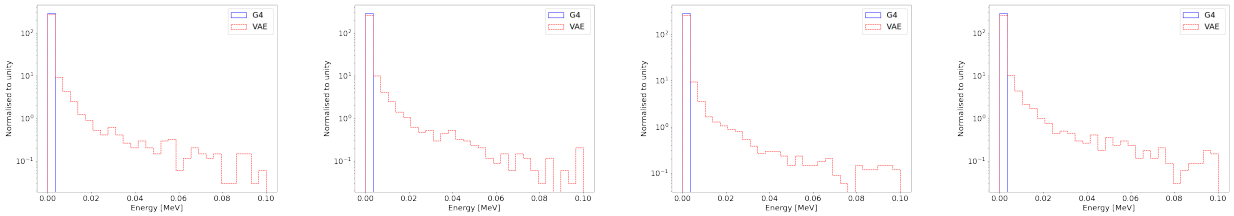
The next plots show the performance of the VAE on single photons with an energy of 65 GeV in $0.2< |\eta| <0.25$. Figure 130 shows the reconstructed sum cell energies in a window of 3×2 cells for EMB1 (a) and 7×7 for EMB2 (b). These plots show the good performance of the VAE in reproducing the energy per cell in the core region of two electromagnetic layers of the ATLAS calorimeter. The VAE trained on the centroids reproduces as well the layer energy. Figure 131 shows the reconstructed energies for EMB0 (a), EMB1 (b), EMB2 (c) and EMB3 (d) as they contain most of the energy deposition for a typical photon shower in $0.2< |\eta| <0.25$. In the VAE, learning to encode an input shower is a key feature during which the correlations are modeled. The correct modeling of these correlations allows to better reproduce some of the key shower observables such as the total energy. Figure 132 is a visualization of correlations between the ECAL layers for both Geant4 and VAE. To further assess the performance of VAE in reproducing the correlations between the layers, Figure 133 represents η distributions for EMB1 and EMB2 and the difference in η distribution between these two layers. DeltaE and Eratio distributions, shown in Figure 134(d) and (r) respectively, are also well reproduced by the VAE. These two quantities are computed using e2tsts1, emins1 and emaxs1 with their respective distributions shown in Figure 134 (a), (b) and (c). All these quantities are defined in Table 5.



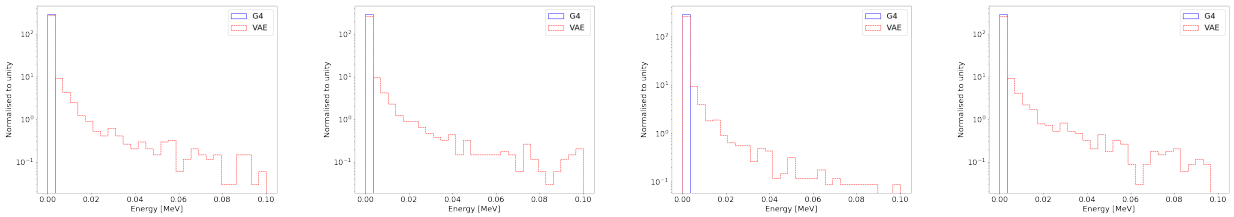
(a) Layer 0 (b) Layer 1 (c) Layer 2 (d) Layer 3



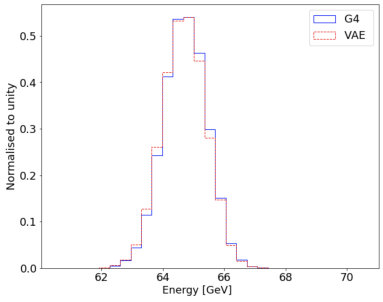
(e) Layer 4 (f) Layer 5 (g) Layer 6 (h) Layer 7



(i) Layer 8 (j) Layer 9 (k) Layer 12 (l) Layer 17

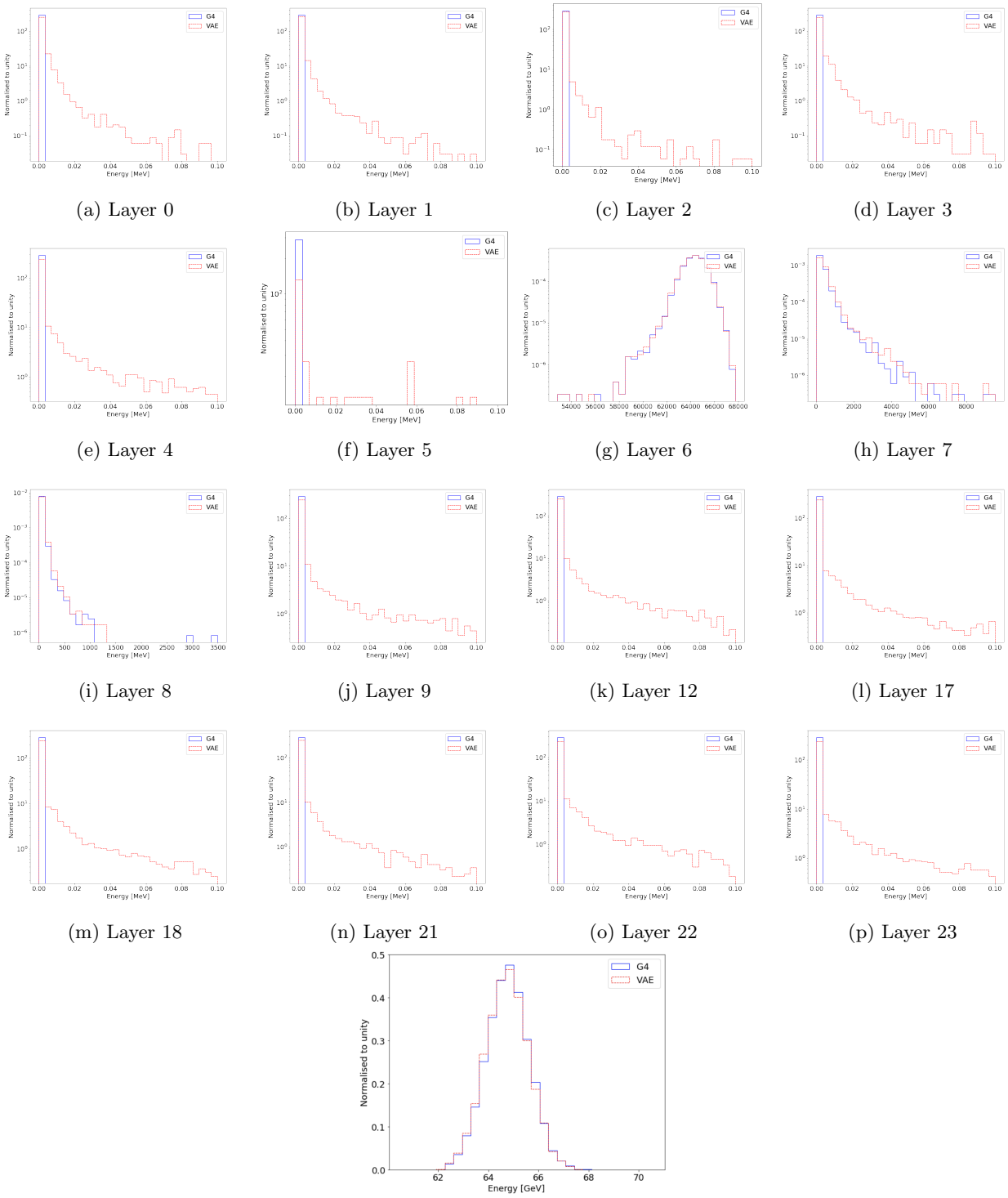


(m) Layer 18 (n) Layer 21 (o) Layer 22 (p) Layer 23



(q) Total energy

Figure 123: Energy per layer and the total energy for photons with an energy of 65 GeV in the $0.6 < \eta < 0.65$ range. The full detector simulation (blue line) is compared to the VAE (red line). Here are shown all the 16 relevant layers for photons reported in Table 11. For $0.6 < \eta < 0.65$ the most of the energy is deposited in layer 0, 1, 2 and 3. The other layers do not have an energy deposition, and this is shown as a single energy bin (blue line).



(a) Layer 0 (b) Layer 1 (c) Layer 2 (d) Layer 3

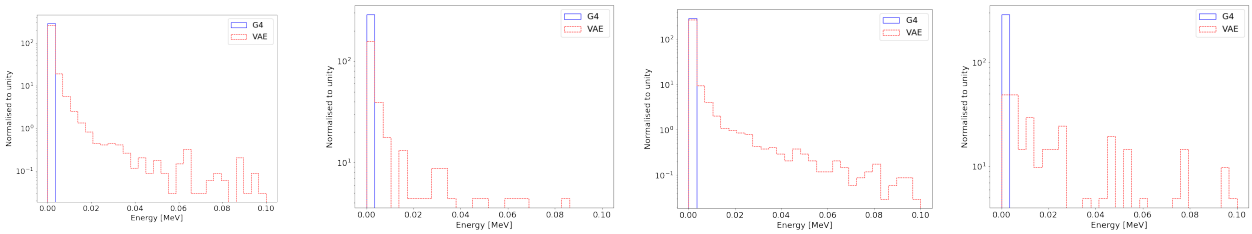
(e) Layer 4 (f) Layer 5 (g) Layer 6 (h) Layer 7

(i) Layer 8 (j) Layer 9 (k) Layer 12 (l) Layer 17

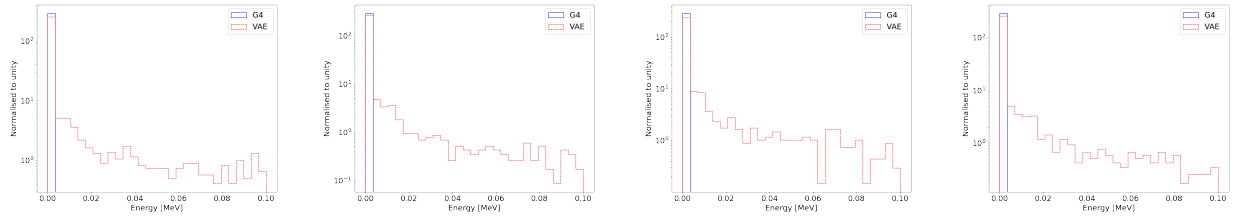
(m) Layer 18 (n) Layer 21 (o) Layer 22 (p) Layer 23

(q) Total energy

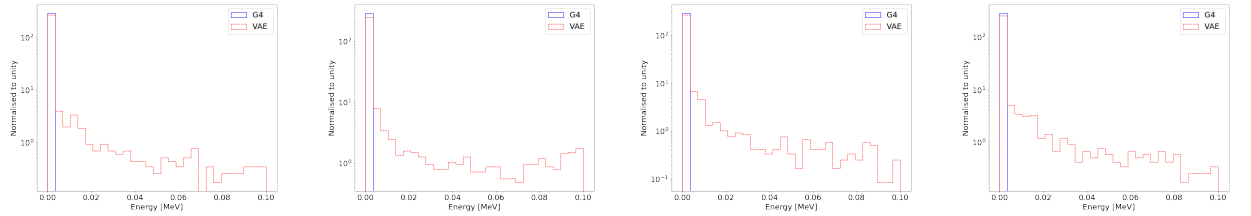
Figure 124: Energy per layer and the total energy for photons with an energy of 65 GeV in the $2.95 < \eta < 3$ range. The full detector simulation (blue line) is compared to the VAE (red line). Here are shown all the 16 relevant layers for photons reported in Table 11. For $2.95 < \eta < 3$ the most of the energy is deposited in layer 6, 7 and 8. The other layers do not have an energy deposition, and this is shown as a single energy bin (blue line).



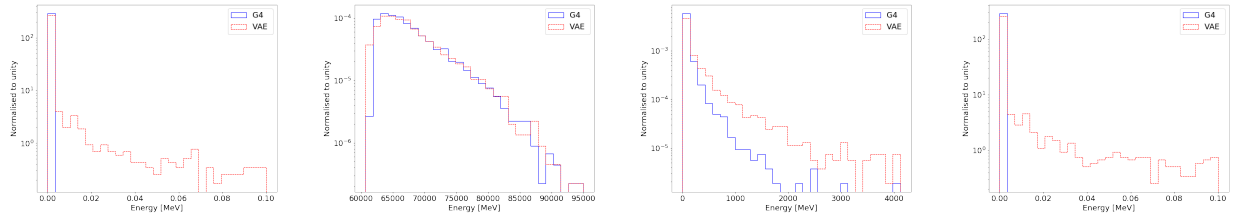
(a) Layer 0 (b) Layer 1 (c) Layer 2 (d) Layer 3



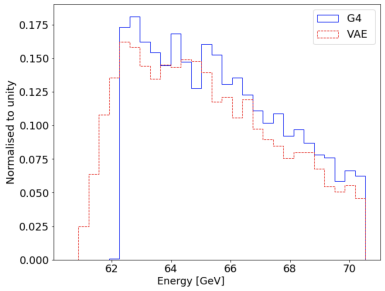
(e) Layer 4 (f) Layer 5 (g) Layer 6 (h) Layer 7



(i) Layer 8 (j) Layer 9 (k) Layer 12 (l) Layer 17

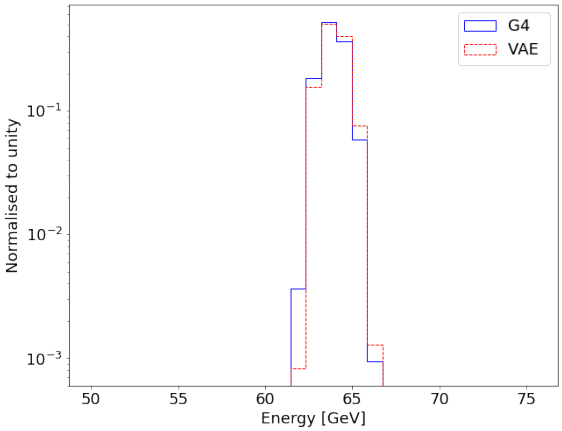


(m) Layer 18 (n) Layer 21 (o) Layer 22 (p) Layer 23

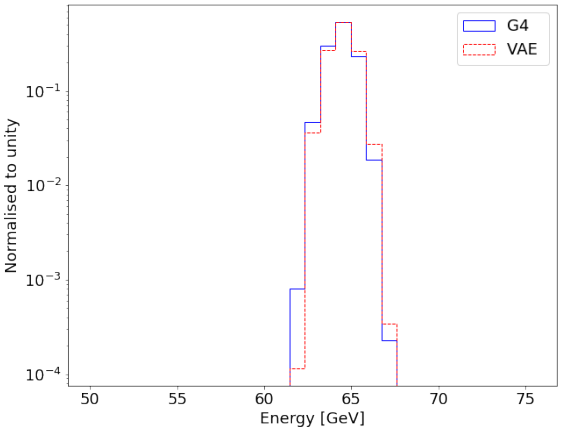


(q)

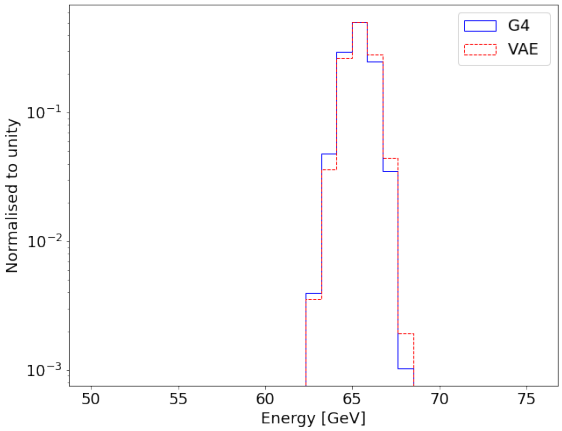
Figure 125: Energy per layer and the total energy for photons with an energy of 65 GeV in the $4.65 < \eta < 4.7$ range. The full detector simulation (blue line) is compared to the VAE (red line). Here are shown all the 16 relevant layers for photons reported in Table 11. For $4.65 < \eta < 4.7$ the most of the energy is deposited in layer 21 and 22. The other layers do not have an energy deposition, and this is shown as a single energy bin (blue line).



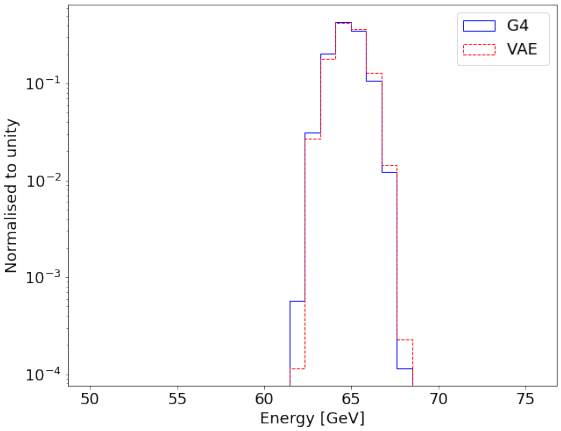
(a) 0.1< $|\eta|$ <0.15



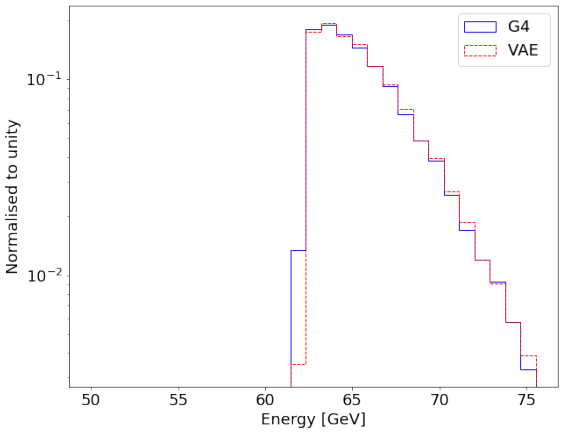
(b) 0.95< $|\eta|$ <1



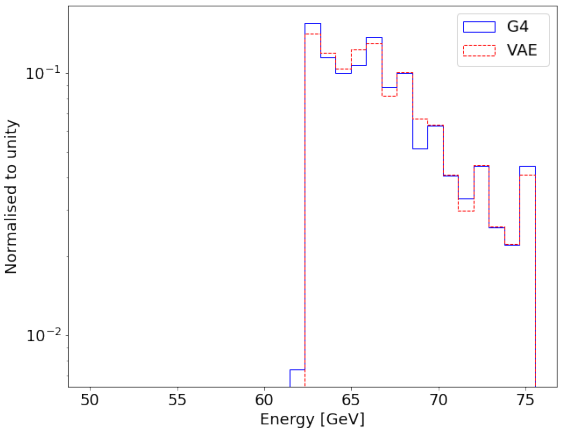
(c) 2< $|\eta|$ <2.05



(d) 2.7< $|\eta|$ <2.75

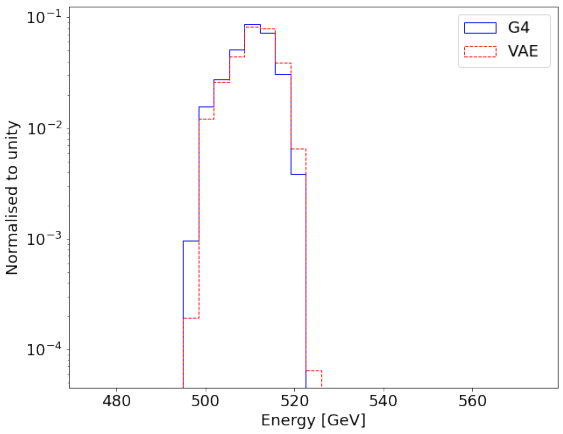


(e)

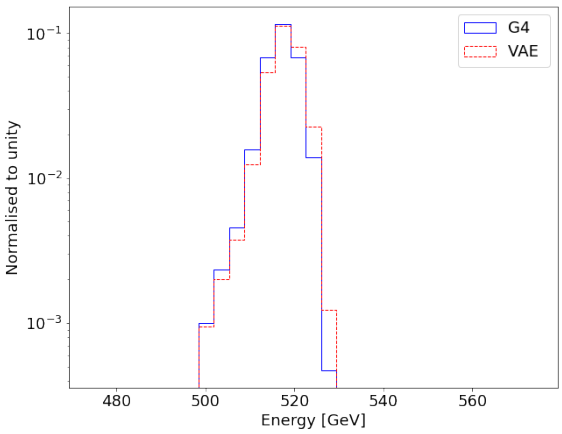


(f) 4.85< $|\eta|$ <4.9

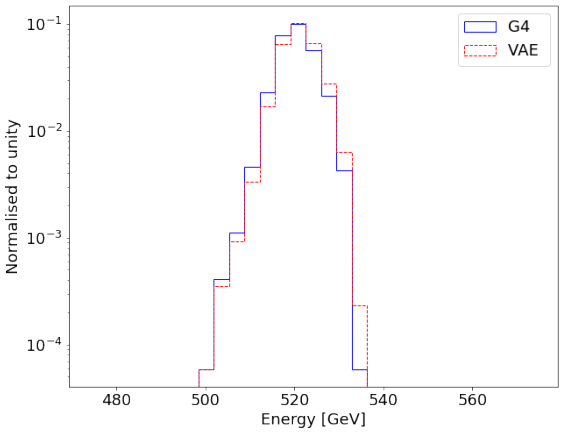
Figure 126: Total energy deposited in the calorimeter (a) for photons with an energy of 65 GeV. The energy depositions from a full detector simulation (solid blue line) are shown as reference and compared to the ones of a VAE (dashed red line)



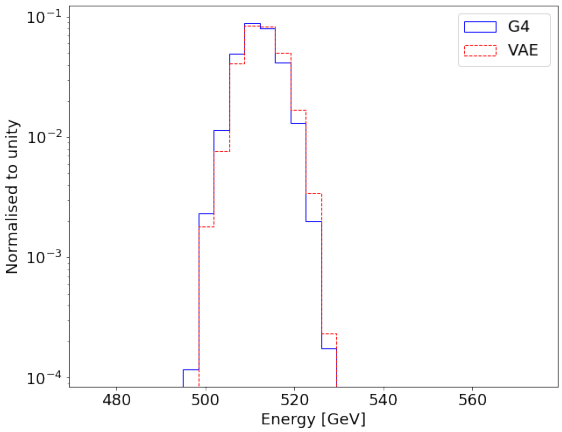
(a) 0.2< $|\eta|$ <0.25



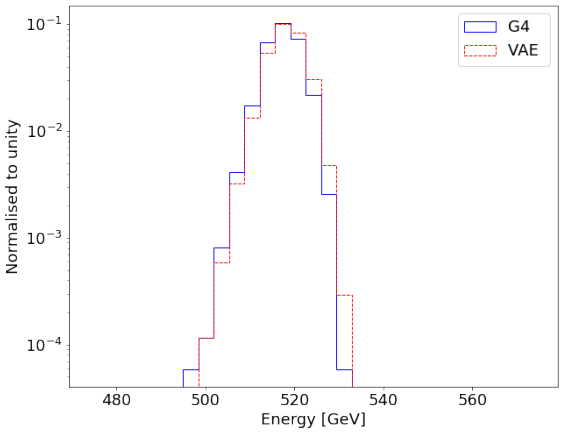
(b) 0.65< $|\eta|$ <0.7



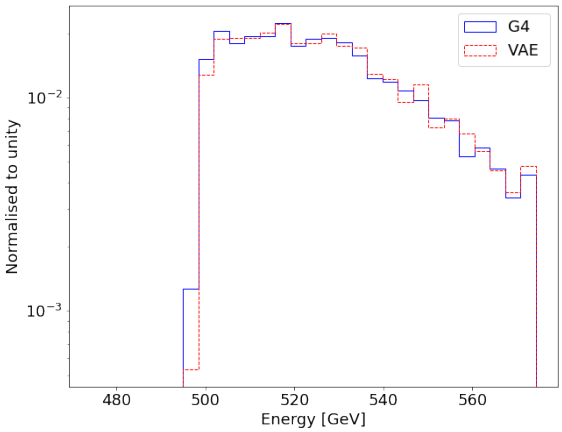
(c) 2.05< $|\eta|$ <2.1



(d) 2.65< $|\eta|$ <2.7

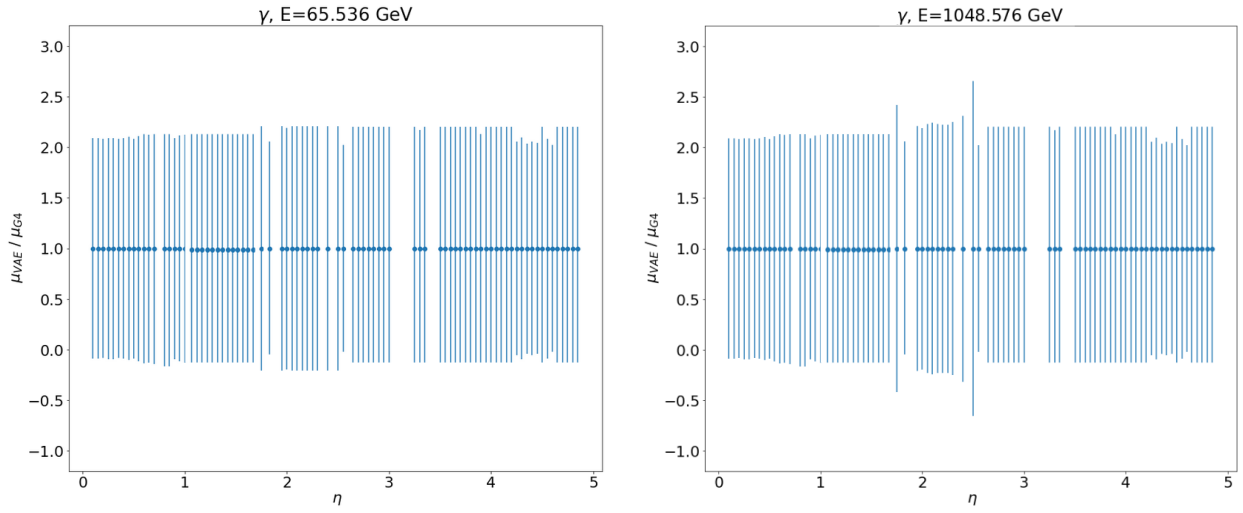


(e) 3< $|\eta|$ <3.05



(f) 4.45< $|\eta|$ <4.5

Figure 127: Total energy deposited in the calorimeter (a) for photons with an energy of 524 GeV. The energy depositions from a full detector simulation (solid blue line) are shown as reference and compared to the ones of a VAE (dashed red line)



(a) (b)

Figure 128: Ratio of the mean of the VAE total energy response to the full detector simulation for photons with an energy of approximately 65 GeV (a) and approximately 1048 GeV (b) as function of η. The error bar shows the ratio of the RMS of the total energy response.

Figure 135 summarizes the performance of the VAE compared to Geant4 in terms of total energy and transverse momentum (pt) distributions. It shows the correct modeling of (a) the sum of the reconstructed energy in the core cells, where only cells in 3×3, 15×2, 5×5 and 3×5 in EMB0, EMB1, EMB2 and EMB3 are respectively considered. More than 90% of the shower photon energy is contained in these core cells.

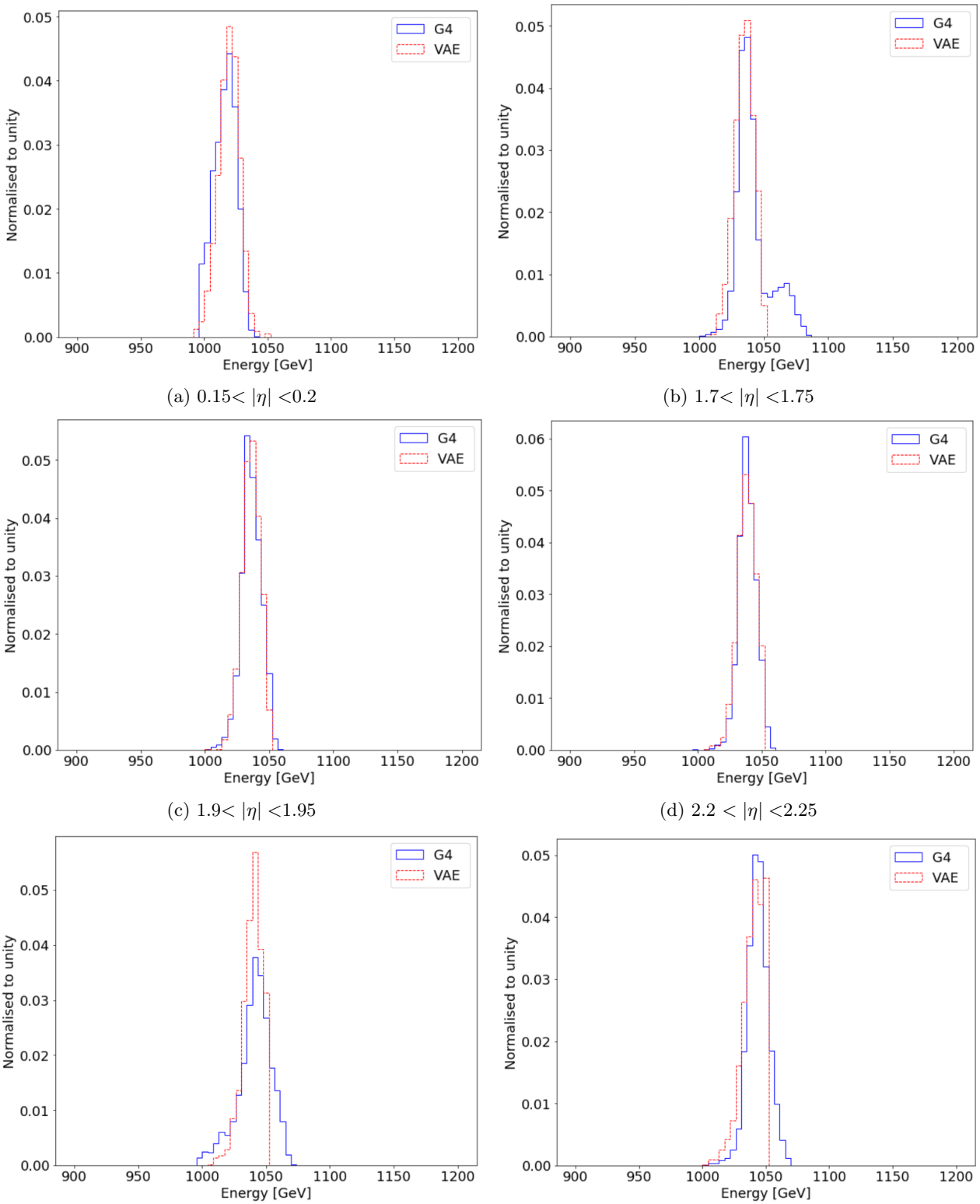
Figure 136 represent the reconstructed variables η, ϕ and pos. A good shower simulator is required to well reproduce all the shower shape variables. This can be further probed with $EGamma$ variables such as $Reta$, $Rphi$ and $Rhad$. In Figure 137 $Reta$ and $Rphi$ evaluate the energy ratio between the core cells with respect to a wider window selection per layer. $Rhad$ represents the fraction of the E_T leakage into hadronic calorimeter with exclusion of energy in TileGap3 with respect to the E_T. Fractional shower shape variables such as $f1$, $f3$ and $fracs$ are shown in Figure 138. The VAE reproduces both $f1$ and $f3$ defined as the fraction of reconstructed energies in EMB1 and EMB3 respectively. Moreover, the shower shape in the shower core shown as $fracs$ agrees with the Geant4 distribution. The width of the shower can be visualized in Figure 139 through the shower width in EMB1 $weta$ and the lateral width in EMB2 $weta2$. It can also be visualized as a shower width in EMB1 without the corrections on particle impact point inside the cell $widths1$ and a shower width in a window of $d\eta \times d\phi$ $wtots1$. These results show that the Athena based observables are accurately modeled by the VAE.

10.3 Centroid-level FastCaloVSim for Pions

All the previous steps of data preprocessing, K-means application, validation and optimization are also applied for pions. Table 13 details the relevant layers for pions and for each layer the corresponding η slice. Unlike photons, all the calorimeter layers are relevant for pions. The number of centroids per layer is set to be 200, 1000, 1000, 200, 200, 200, 200, 200, 200, 200, 200,200, 500, 500, 500, 200, 200, 200, 200, 200, 200, 200, 200, 200 for the 24 layers in order. The number of centroids in the TileBar layers 0,1, and 2 corresponding to layers 12, 13 and 14, is more than two times higher compared with photons to better capture the hadronic shower development in these layers.

As a first qualitative validation of single pion simulation, the standalone validation is based on evaluating the performance of reproducing the average energy ratios as function of dr. Figure 140 shows the energy ratio of centroids to the energies per layer for EMB0 (a), EMB1 (b), EMB2 (c) and TileBar0 (d). Moreover, the energy per layer for EME2 and TileBar0 can be seen in Figures 141 and 142 across different η slices.

On the Athena side, the validation of single pions is based on reconstructing their showers as jets. This reconstruction is derived from the ATLAS topological cell clustering [203] based on the anti-k_t algorithm [206] which groups close energies with a high probability of belonging to the same shower. These grouped clusters are an efficient visualization to describe the topology of the energy depositions. The radius R considered for this algorithm and which ATLAS uses is 0.4 (and 1). In a hadronic shower, hadrons generated in inelastic interactions can even travel significant distances and generate sub-showers outside the direct neighborhood of the calorimeter cell containing the initial hadronic interaction. Therefore, the topo-clusters can contain only



(a) 0.15< $|\eta|$ <0.2

(b) 1.7< $|\eta|$ <1.75

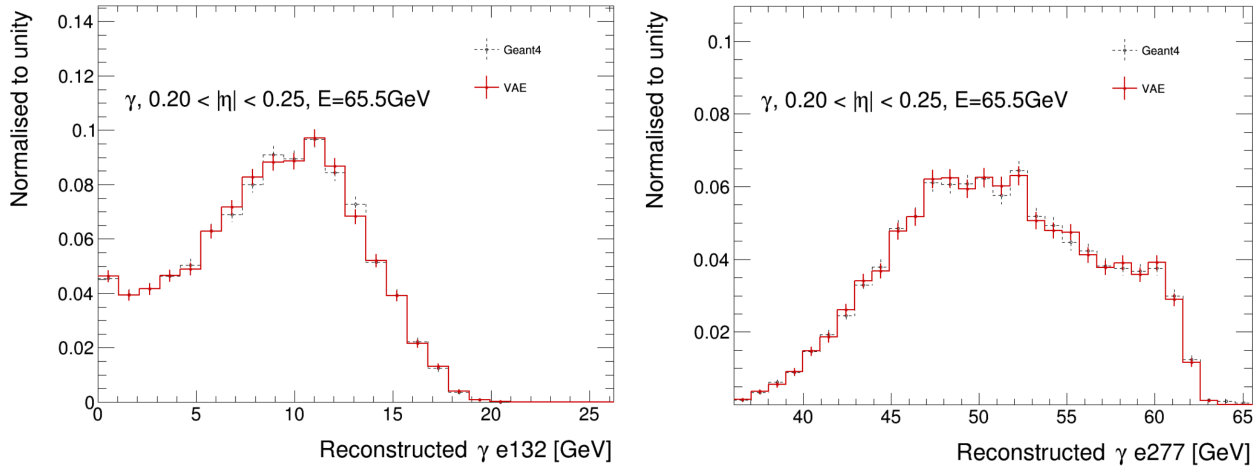
(c) 1.9< $|\eta|$ <1.95

(d) 2.2 < $|\eta|$ <2.25

(e) 2.5< $|\eta|$ <2.55

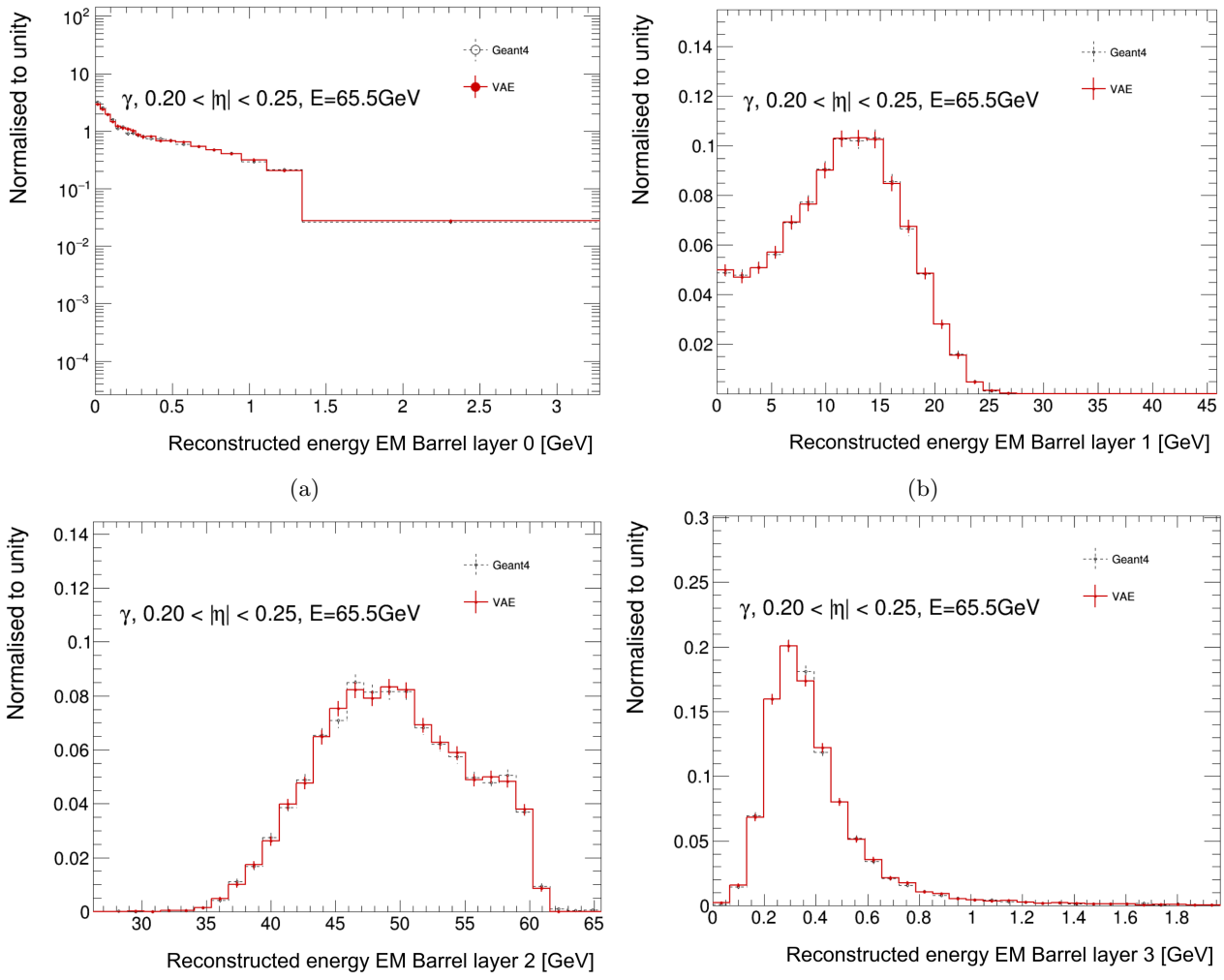
(f) 2.8< $|\eta|$ <2.85

Figure 129: Total energy response of the calorimeter to photons with an energy of 1048 GeV. The calorimeter response for the Full detector Simulation (solid blue line) compared to FastCaloVSim (dashed red line).



(a) (b)

Figure 130: Reconstructed energy in a selected window of 3×2 in EMB1 (a) and 7×7 (b) in EMB2 for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).



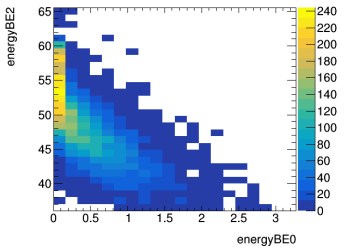
(a) (b)

(c) (d)

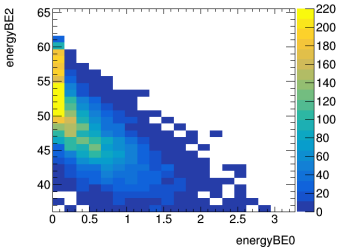
Figure 131: Reconstructed photon energy in (a) EMB0, (b) EMB1, (c) EMB2 and (d) EMB3 layers energy for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line)

(a)

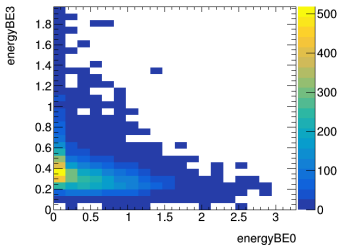
(b)



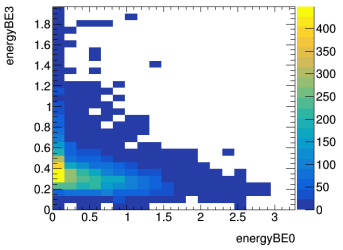
(c)



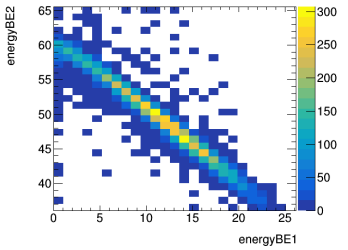
(d)



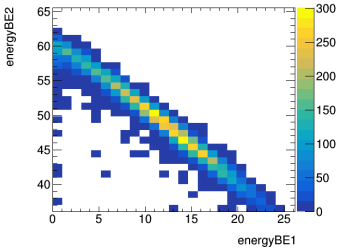
(e)



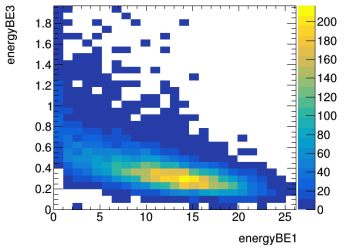
(f)



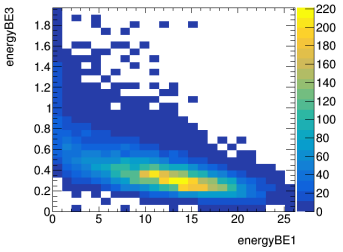
(g)



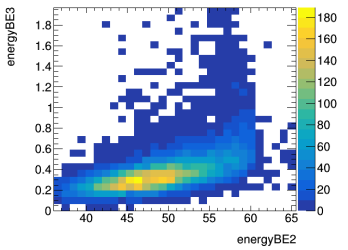
(h)



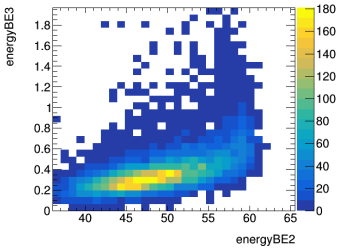
(i)



(j)

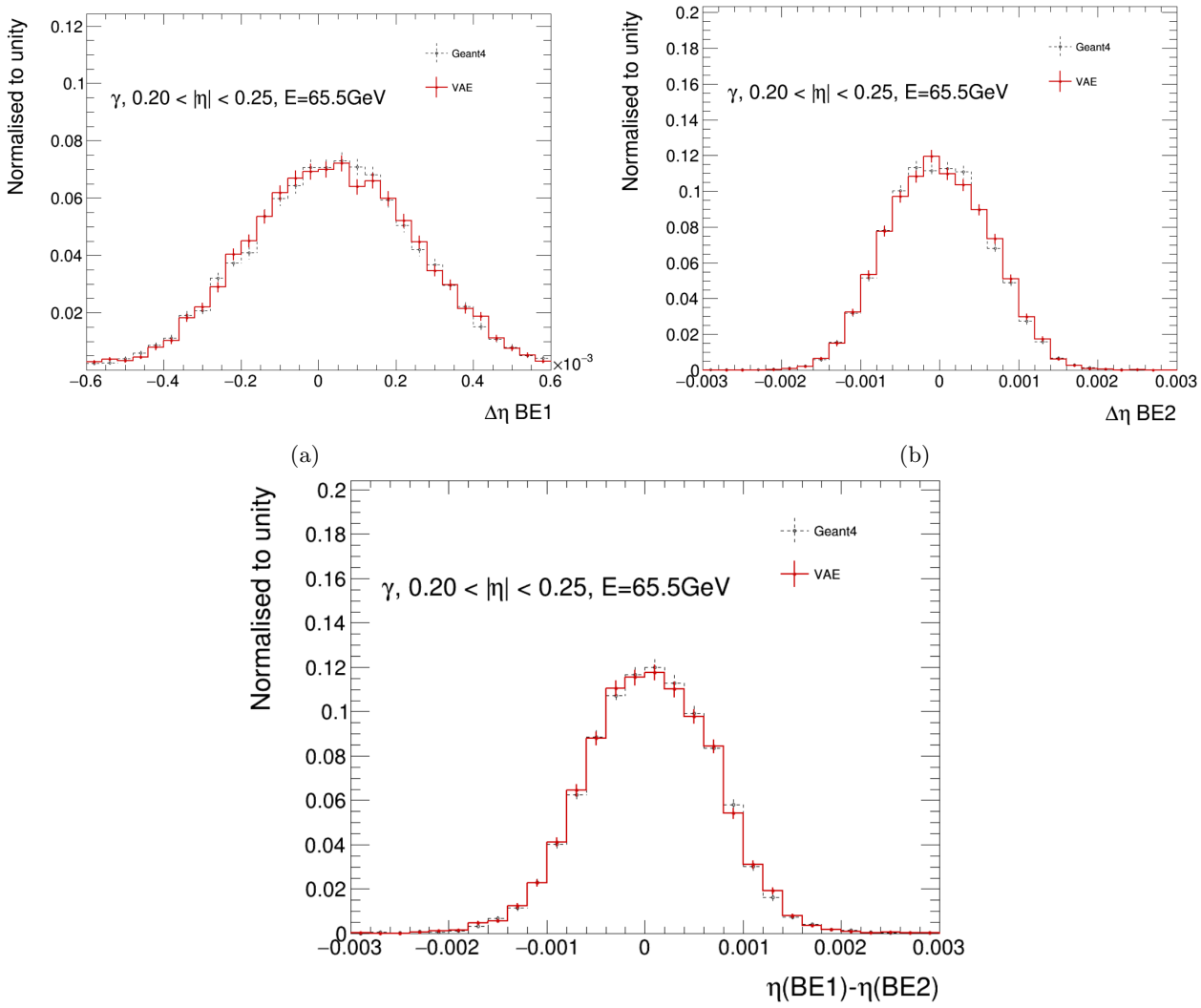


(k)



(l)

Figure 132: Correlations between deposited energy in the four ATLAS electromagnetic layers for photons with an energy of approximately 65 GeV in the $0.2 < |\eta| < 0.25$ range. The full detector simulation (left column) is compared to VAE (right column): (a, b) EMB1 vs EMB0, (c, d) EMB2 vs EMB0, (e, f) EMB3 vs EMB0, (g, h) EMB2 vs EMB1, (i, j) EMB3 vs EMB1, (k, l) EMB3 vs EMB2. The numbers on the colorbar represent the number of events.



(a) (b)

(c)

Figure 133: Position resolution along η for EMB1 (a) and EMB2 (b) layers and η difference between EMB1 and EMB2 for photons with an energy of approximately 65 GeV in the $0.2< |\eta| <0.25$ range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

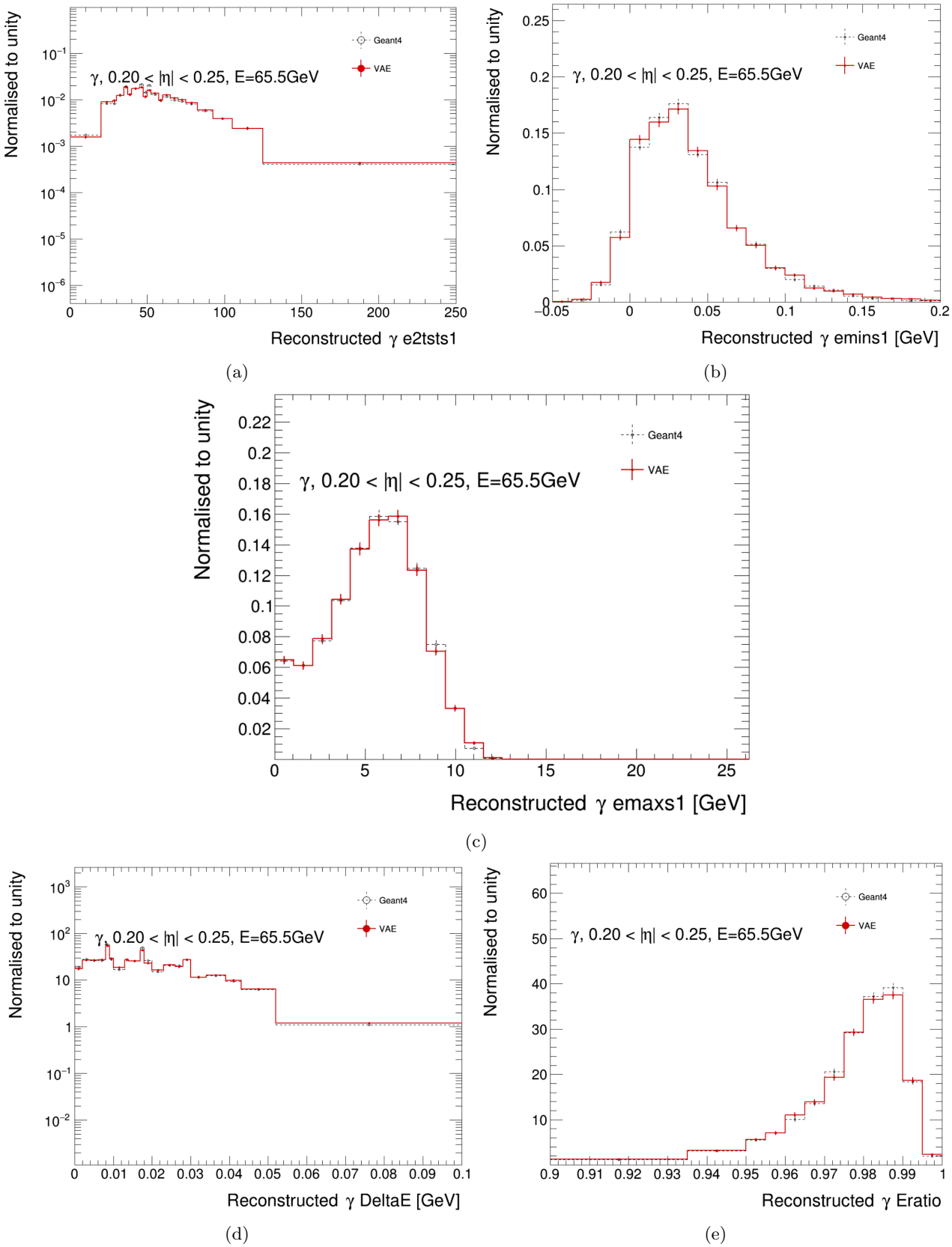


Figure 134: Shower shapes (d) DeltaE and (e) ERatio distributions along with the corresponding quantity distributions (a) $e2tsts1$, (b) $emins1$ and (c) $emaxs1$ used to derive them for photons with an energy of approximately 65 GeV in the $0.2< |\eta| <0.25$ range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

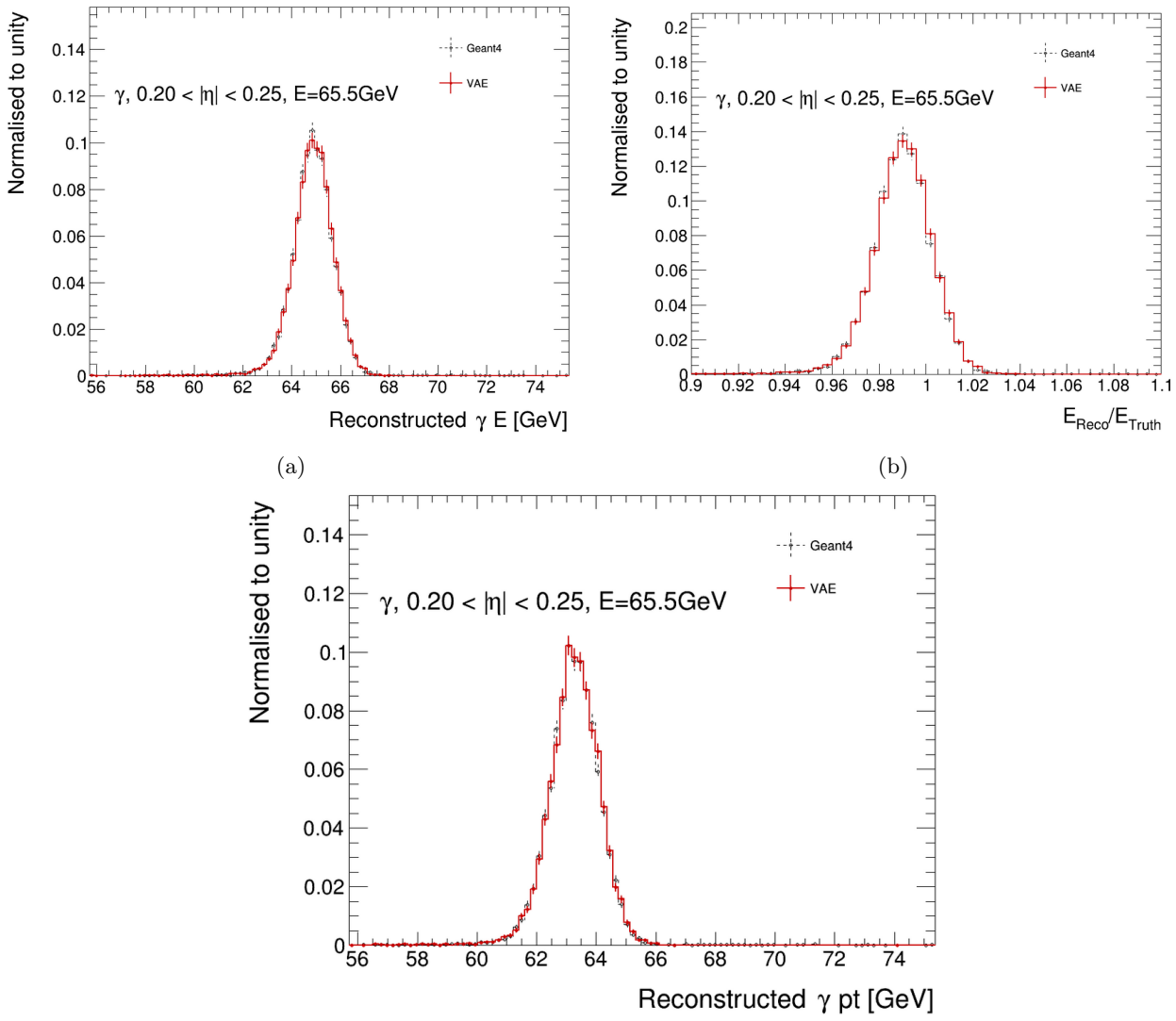


Figure 135: Reconstructed energy in the core (a), ratio of reconstructed energy to the truth energy (b) and *pt* (c) for photons generated with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

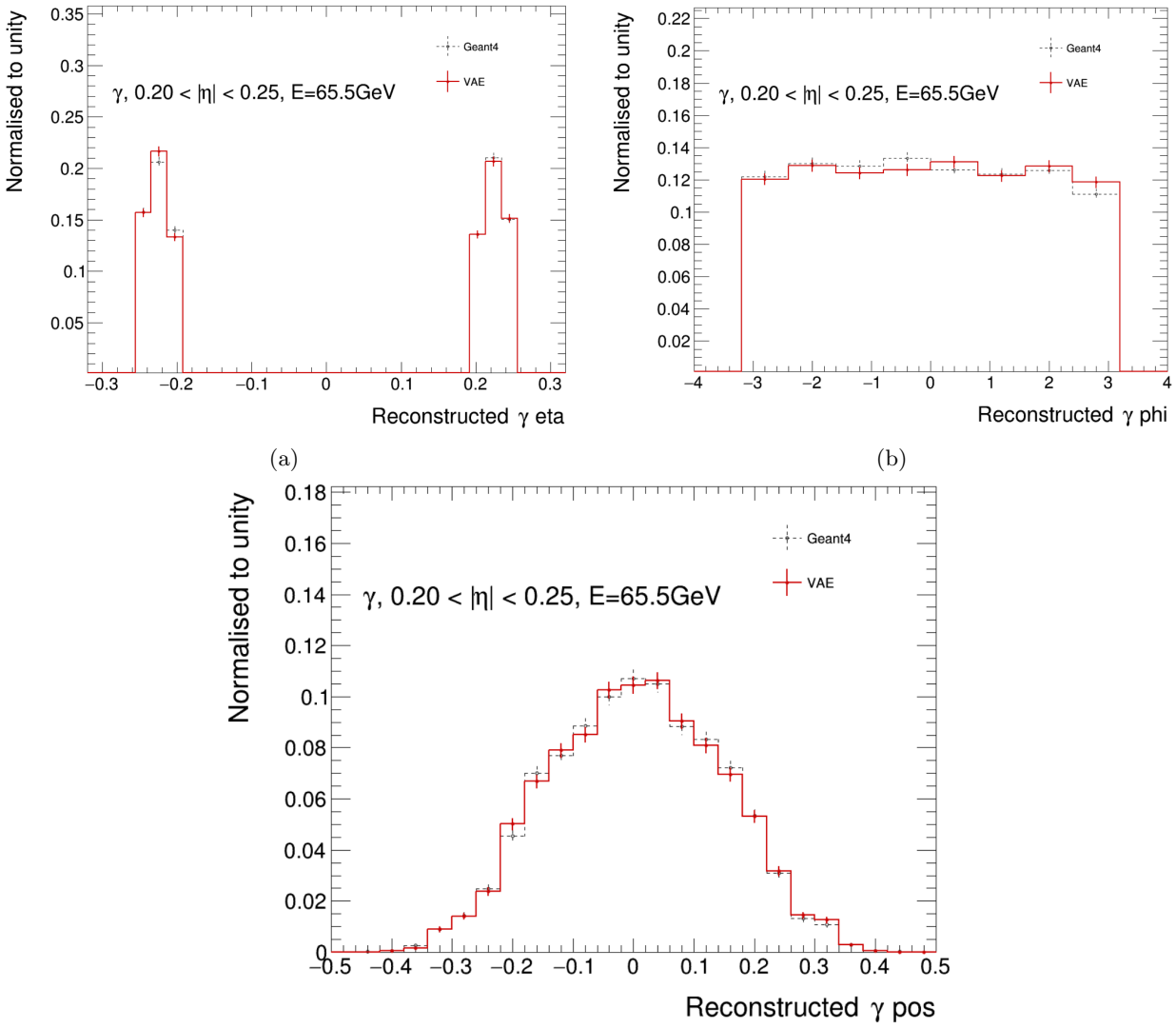
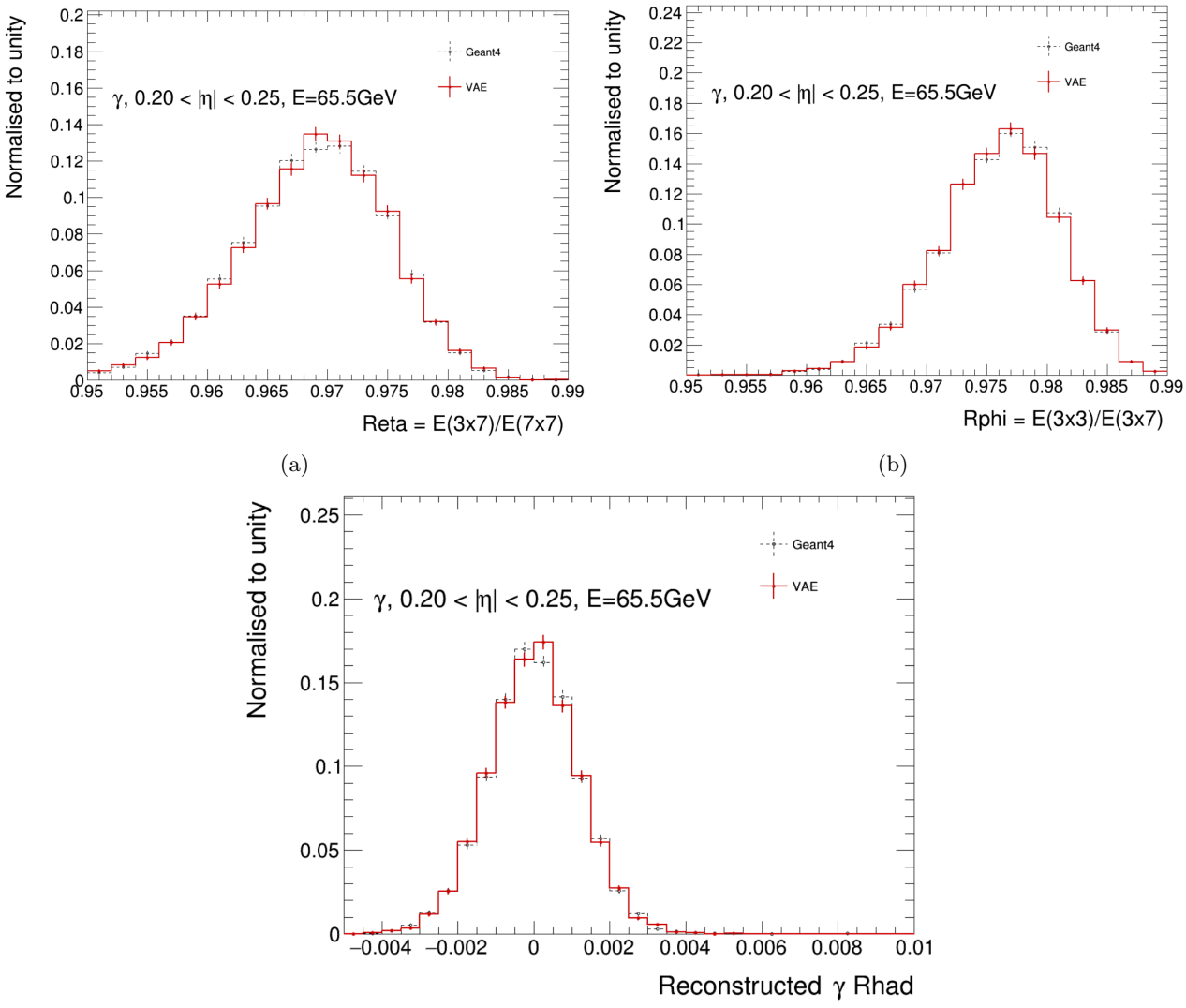


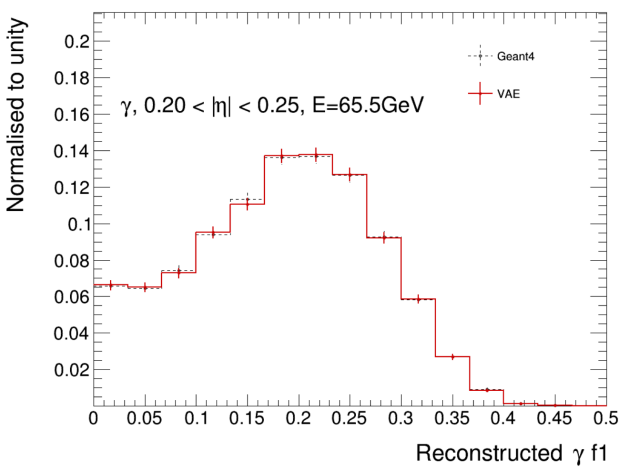
Figure 136: Reconstructed (a) η, (b) ϕ and (c) *pos* distributions for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).



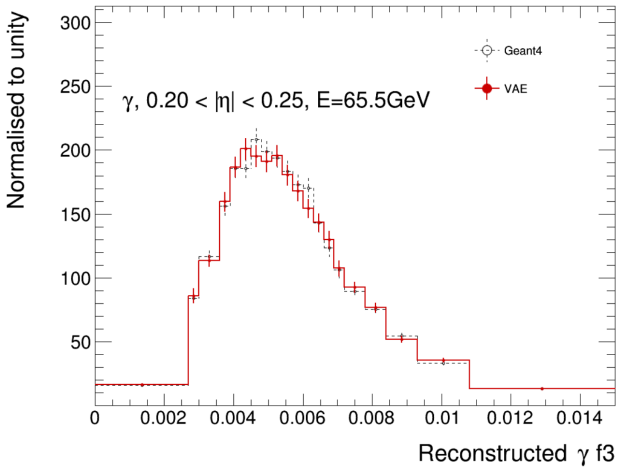
(a) (b)

(c)

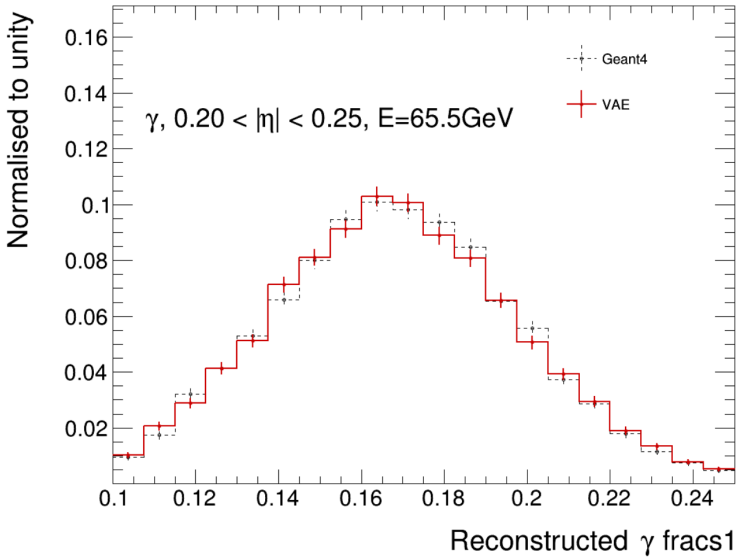
Figure 137: Reconstructed (a) *Reta*, (b) *Rphi* and (c) *Rhad* distributions for photons with an energy of approximately 65 GeV in the $0.2 < |\eta| < 0.25$ range. The full detector simulation (dashed black line) is compared to VAE (solid red line).



(a) f1



(b) f3



(c) fracs

Figure 138: Reconstructed (a) $f1$, (b) $f3$ and (c) $fracs$ distributions for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

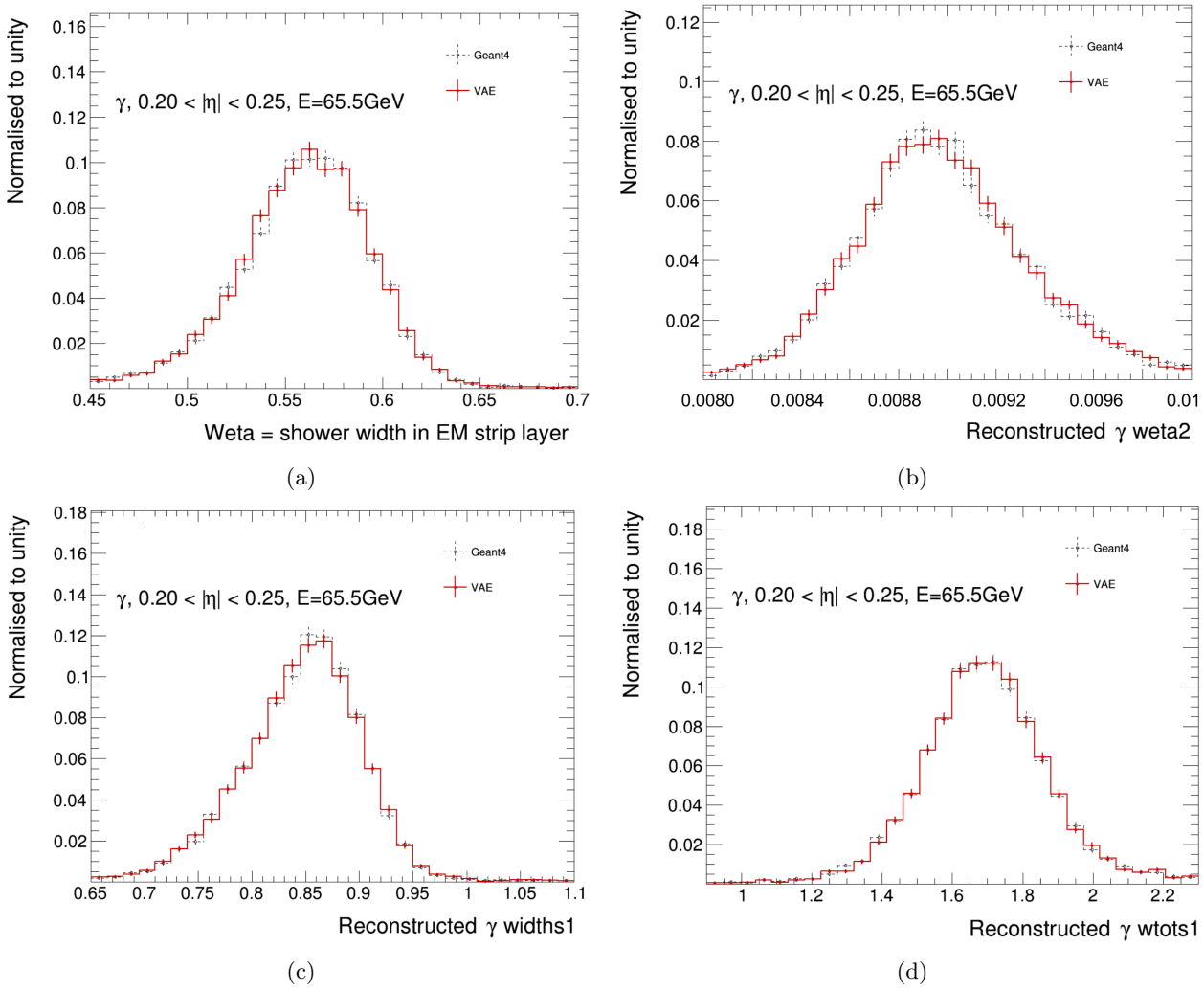


Figure 139: Reconstructed (a) *weta*, (b) *weta*2, (c) *widths*1 and (d) *wtots*1 distributions for photons with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to VAE (solid red line).

a fraction of the hadronic shower [203]. Reproducing the cluster variables is a very challenging task which requires an accurate modeling of the right number of clusters in the right part of the detector with all the complex correlations. The next set of plots show the FastCaloVSim performance for pions of 65 GeV in the $0.2< |\eta| <0.25$ range.

One of the key distributions for evaluating the performance of the VAE is reported in Figure 143 as the leading cluster energy. Overall, the VAE can reproduce a very close shape to Geant4 over the whole energy range. This can also be seen in terms of ratios of the leading cluster energy to the truth energy. In the same figure, the shape of leading cluster p_T distribution is also reasonably reproduced. A good agreement is seen in distributions of η and ϕ shown in Figure 144.

Figure 145 shows the center λ, where up to 200 cm it represents clusters that have centroids in EMB2, beyond 1 m most of the cluster energy is in the tile layers. The extra peak around 1.5 m represent the clusters starting in TileBar1 (layer 13) and the tail is the energy sharing with TileBar2. The figure shows as well the ΔR measuring how far the pions are to the center. Overall, the distribution is well reproduced.

Secondary cluster moments such as $< \lambda^2 >$ and $< r^2 >$ are shown in Figure 146. The correct modeling of $< \lambda^2 >$ and $< r^2 >$ allows good performance in reproducing both longitudinal and lateral quantities for all the clusters. Some discrepancies are visible in both plots. Low values of $< \lambda^2 >$ indicate the presence of narrow clusters and the peak at zero indicates Minimum Ionizing Particles (MIPs), which have a rate of mean energy loss close to the minimum, like clusters. For $< r^2 >$, the agreement to Geant4 is better across all the $< r^2 >$ range compared to all the clusters. The VAE can also reproduce with some discrepancies the reconstructed energy of topo-clusters as shown in Figure 147 and the clusters E_T shown in Figure 148. The modeling of E_T can also be seen as a function of η shown in Figure 149, where the correlation structure is reproduced.

To evaluate the performance of jet reconstruction, Figure 150 shows the distribution of the jet P_t. FastCaloVSim can reproduce the average and the RMS of this distribution with some disagreement compared to Geant4. Figure 151 illustrates the jet mass and the reconstructed jet tau21. Although this is a case of single particles, where these two quantities are small, it is interesting to see that they can be reproduced. Figure 152 represents modeling of the number of clusters for the reconstructed jet. This value is well modeled, especially at high multiplicity.

Quality variables such as the isolation and $< Q >$ in the liquid argon are reported in Figure 153 which are reproduced by the VAE. In the topo-clustering algorithm [203], the noise is implicitly removed. This leads to a signal loss and therefore alters the calorimeter response. This loss appears at the topo-cluster boundary. The correction of this effect is sensitive if this loss can be included in a neighboring cluster or if they can be ignored. A topo-cluster with an isolation coefficient of zero means that the cluster is surrounded by other clusters. On the other hand, a coefficient of one means that the cluster is completely isolated. $< Q >$ refers to the total charge produced by an energy deposition of particles.

FastCaloVSim at the centroid level is trained on a range of energy from 1 GeV to 1 TeV. An extrapolation to 2 TeV particles demonstrates the learning capacity of the model. Figure 154 shows the secondary cluster moments $< \lambda^2 >$ and $< r^2 >$ and the ratio of the jet leading cluster energy to jet p_T. In general, the plots show a reasonable agreement to the full simulation.

10.4 Centroid-level FastCaloVSim Performance on Di-jets

After the performance evaluation on single particles, this section describes the evaluation on jets. Even if the performance on single particles is not at the percent level, an energy response simulator can still be able to reproduce reconstructed objects. This evaluation is based on a di-jet sample where a filtering is applied to only contain leading jets with a p_T of about 2.5 TeV. These samples contain also additional jets to cover the full energy range. To run FastCaloVSim on these samples, a full detector parametrization is created with basic analysis and definition of cuts and defining FastCaloVSim as the simulator. This global evaluation of jets is also a global evaluation of FastCaloVSim to interpolate and extrapolate in the full energy range. The observables for the evaluation process are similar to the pion ones. Figures 155, 156, 157 158 show the performance of FastCaloVSim on these physics samples compared to a reference ATLAS fast simulation AFII. Many improvements are demonstrated through these plots compared to AFII that can be explained by a better learning of the correlations with an ML model. The number of cluster in the subleading jet has significantly improved.

10.5 Summary and Discussion

This chapter described the VAE model trained on centroids derived using the K-means clustering algorithm. This model extends the cell and voxel models in terms of layers and η regions, where all layers and angular

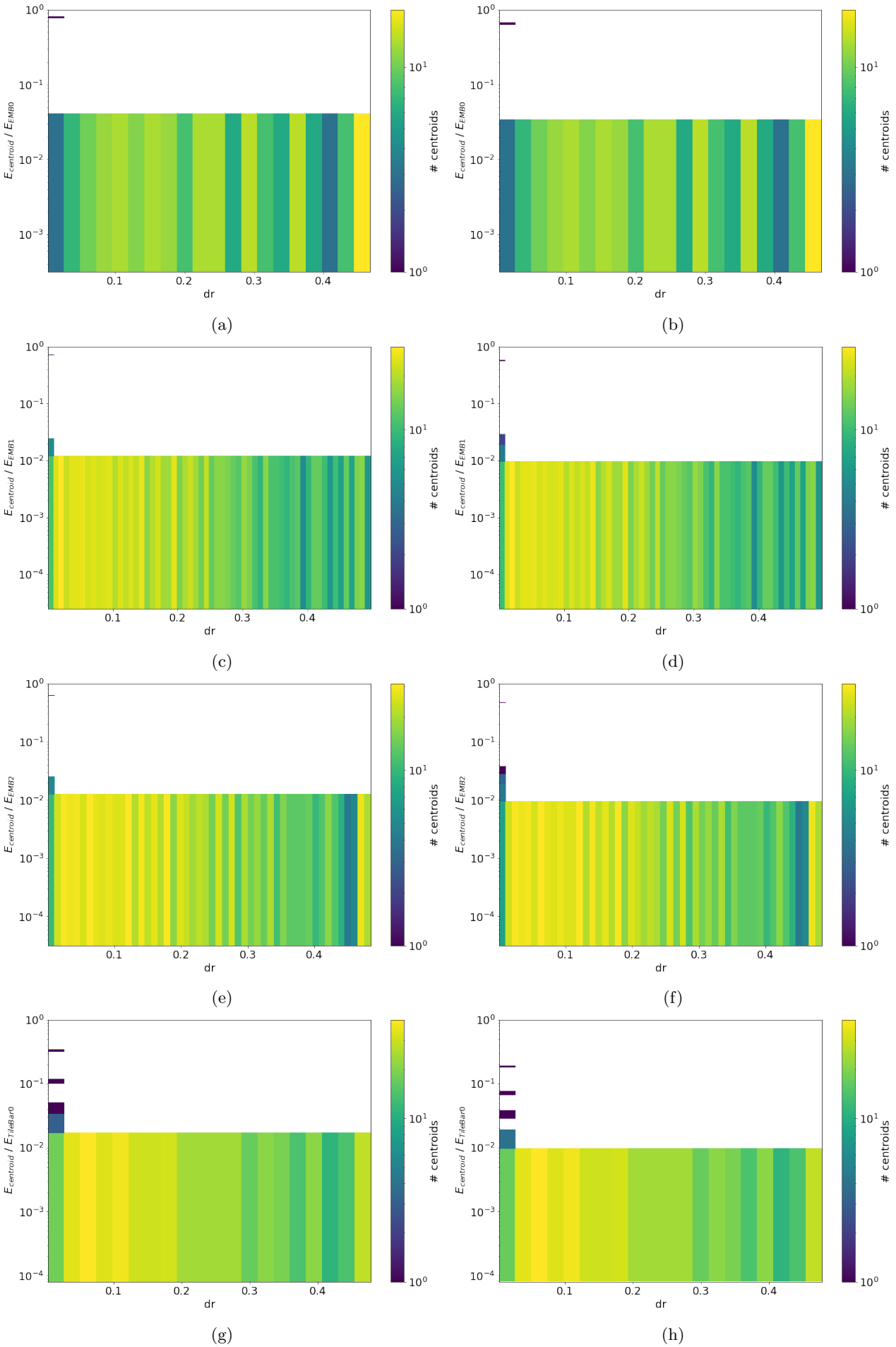
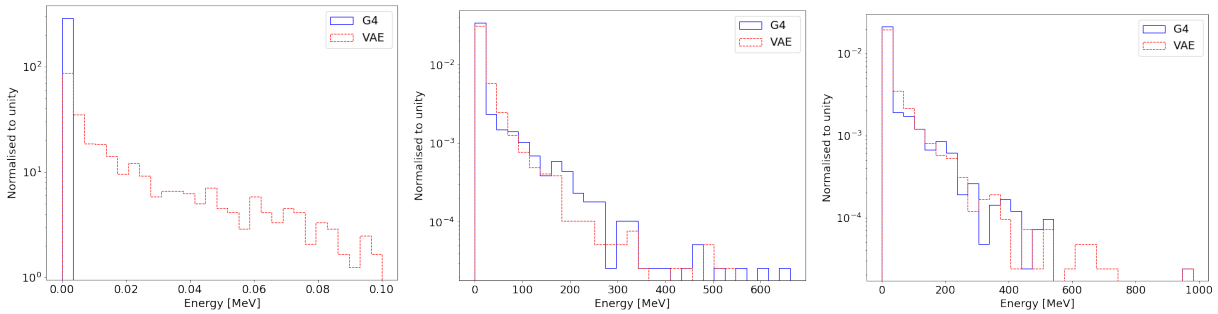
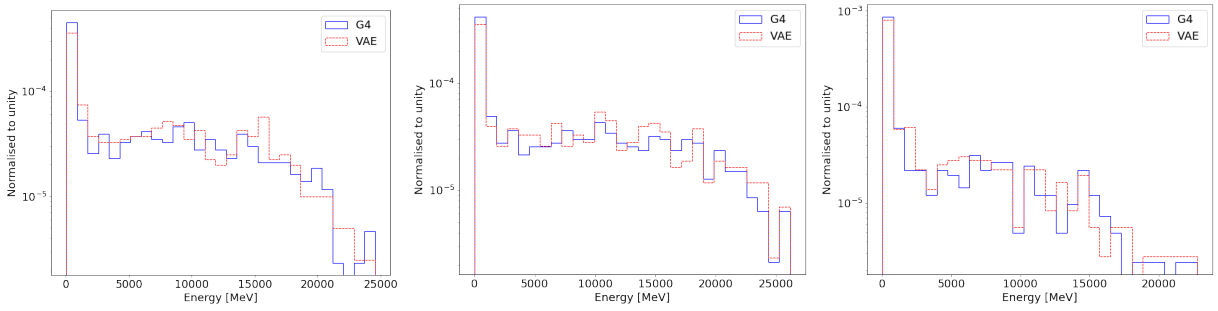


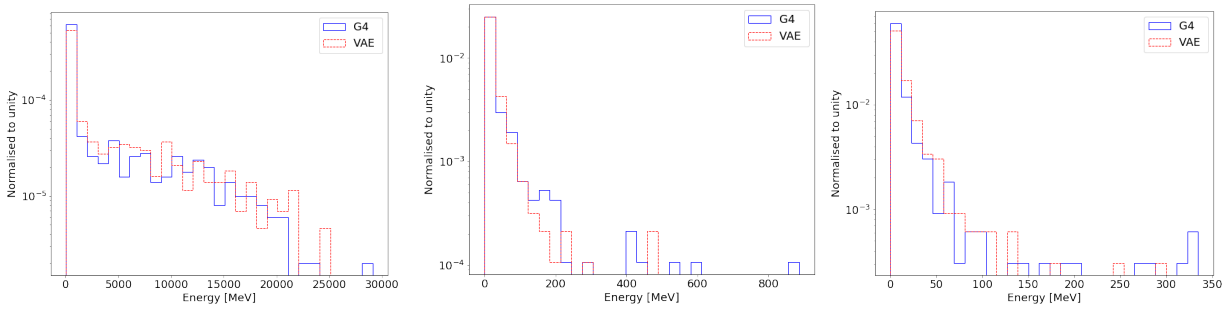
Figure 140: Average centroid energy as of *dr* function for (a, b) EMB0, (c, d) EMB1, (e, f) EMB2 and (g, h) TileBar0, for pions with an energy of approximately 65 GeV in the $0.2< |\eta| <0.25$ range. Left column represents Geant4 and right column VAE.



(a) 0.25< $|\eta|$ <0.3 (b) 1.2< $|\eta|$ <1.25 (c) 1.25< $|\eta|$ <1.3

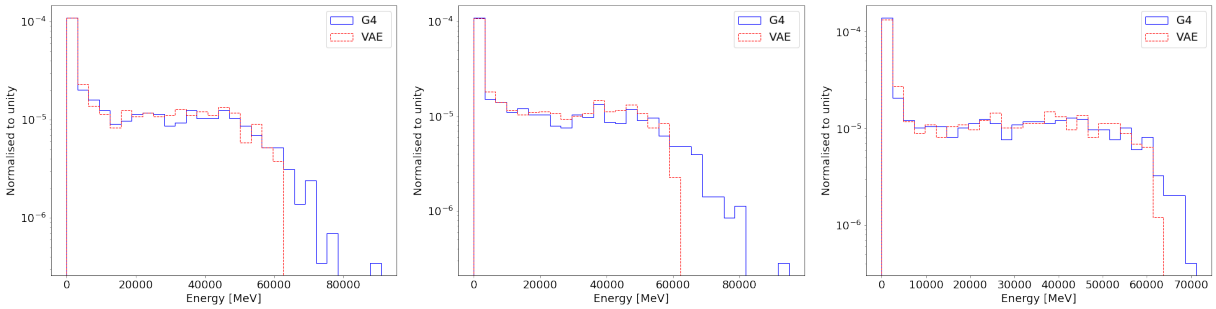


(d) 1.65< $|\eta|$ <1.7 (e) 1.7< $|\eta|$ <1.75 (f) 1.9< $|\eta|$ <1.95

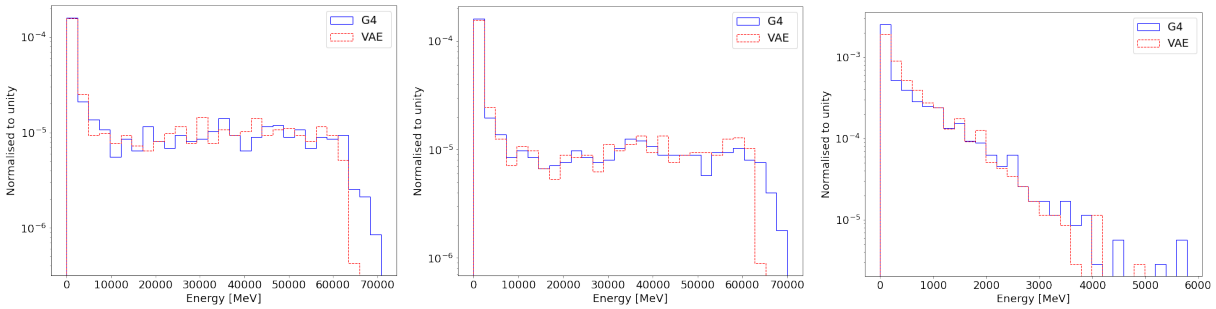


(g) 2.05< $|\eta|$ <2.1 (h) 2.65< $|\eta|$ <2.7 (i) 2.75< $|\eta|$ <2.8

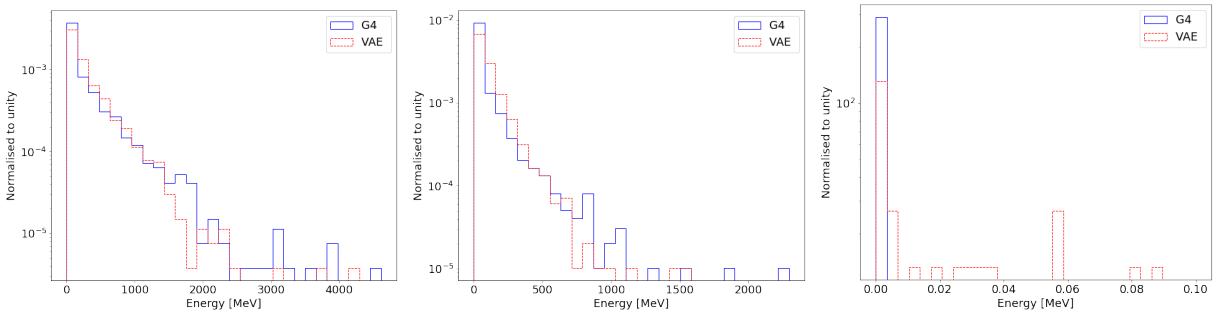
Figure 141: Energy deposited in layer EME1 for pions with an energy of approximately 65 GeV. The full detector simulation (solid blue line) compared to the VAE (solid red line). For (a) EME1 is not a relevant layer in the 0.25< $|\eta|$ <0.3 region.



(a) 0.05< $|\eta|$ <0.1 (b) 0.1< $|\eta|$ <0.15 (c) 0.25< $|\eta|$ <0.3

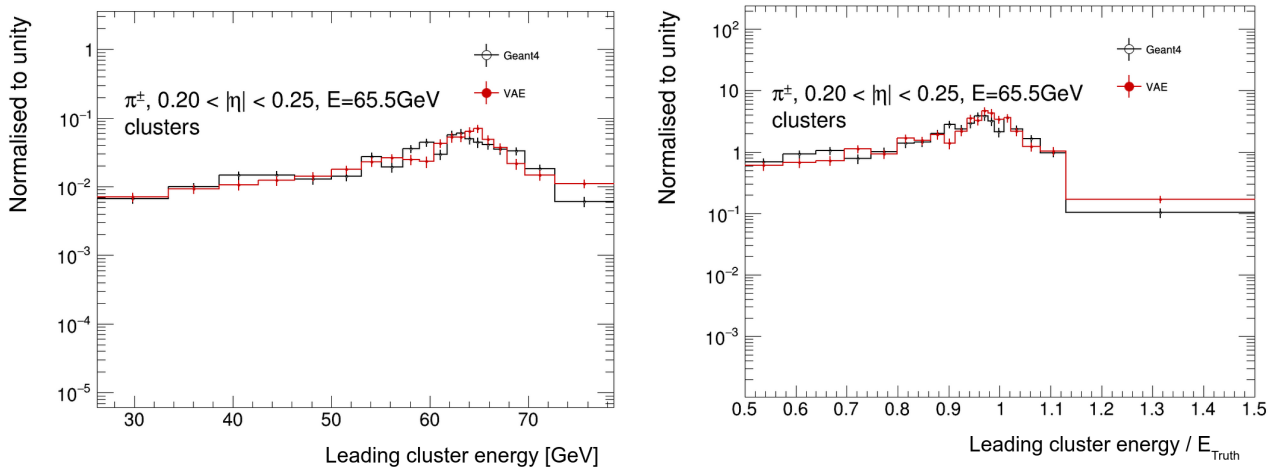


(d) 0.45< $|\eta|$ <0.5 (e) 0.55< $|\eta|$ <0.6 (f) 1< $|\eta|$ <0.15

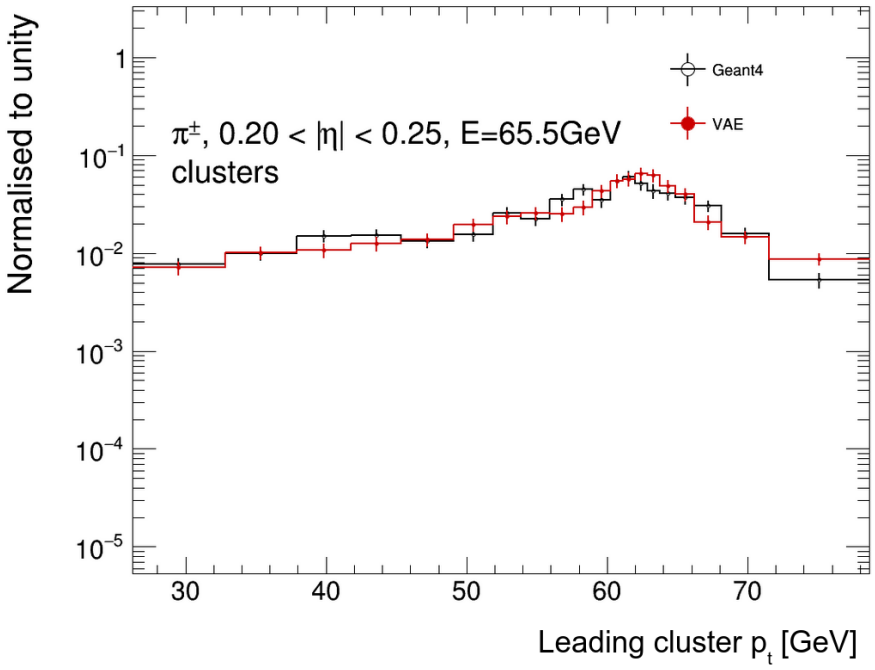


(g) 1.15< $|\eta|$ <1.2 (h) 1.25< $|\eta|$ <1.3 (i) 3.15< $|\eta|$ <3.2

Figure 142: Energy deposited in layer TileBar0 for pions with an energy of approximately 65 GeV. The full detector simulation (solid blue line) compared to the VAE (solid red line). For (i) TileBar0 is not a relevant layer in the 3.15< $|\eta|$ <3.2 region.



(a) (b)

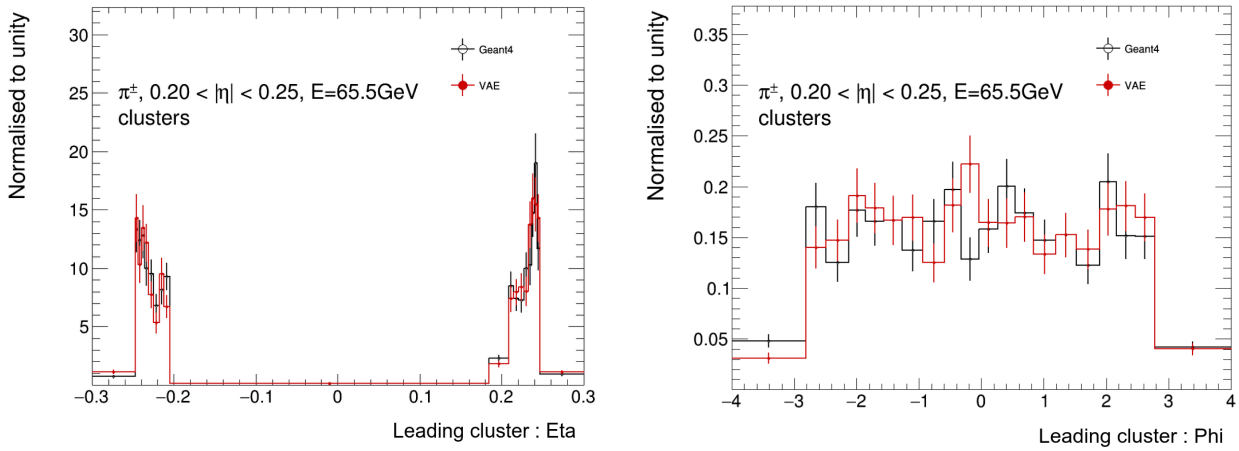


(c)

Figure 143: (a) Leading cluster energy, (b) leading cluster energy over truth energy and (c) cluster p_T for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

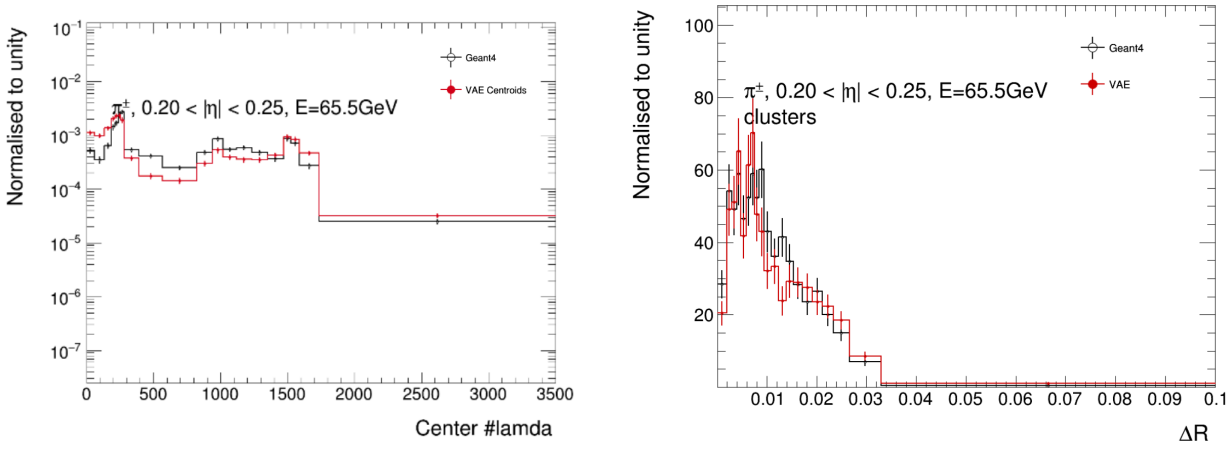
Relevant layer	η coverage
0	$0< \|\eta\| <3.5$
1	$0< \|\eta\| <3.5$
2	$0< \|\eta\| <3.5$
3	$0< \|\eta\| <3.5$
4	$1.55< \|\eta\| <2.1$
5	$0.85< \|\eta\| <4$
6	$0.8< \|\eta\| <4.9$
7	$1.25< \|\eta\| <4.9$
8	$1.2< \|\eta\| <3.9$
9	$1.25< \|\eta\| <5$
10	$1.45< \|\eta\| <5$
11	$1.55< \|\eta\| <5$
12	$0< \|\eta\| <1.55$
13	$0< \|\eta\| <1.6$
14	$0< \|\eta\| <0.85$
15	$0.8< \|\eta\| <1.4$
16	$0.7< \|\eta\| <1.4$
17	$0.65< \|\eta\| <3.1$
18	$0.75< \|\eta\| <2.3$
19	$0.75< \|\eta\| <2$
20	$0.75< \|\eta\| <1.6$
21	$2.4< \|\eta\| <5$
22	$2.6< \|\eta\| <5$
23	$2.85< \|\eta\| <5$

Table 13: List of relevant layers for pions particles with respective η slide definition



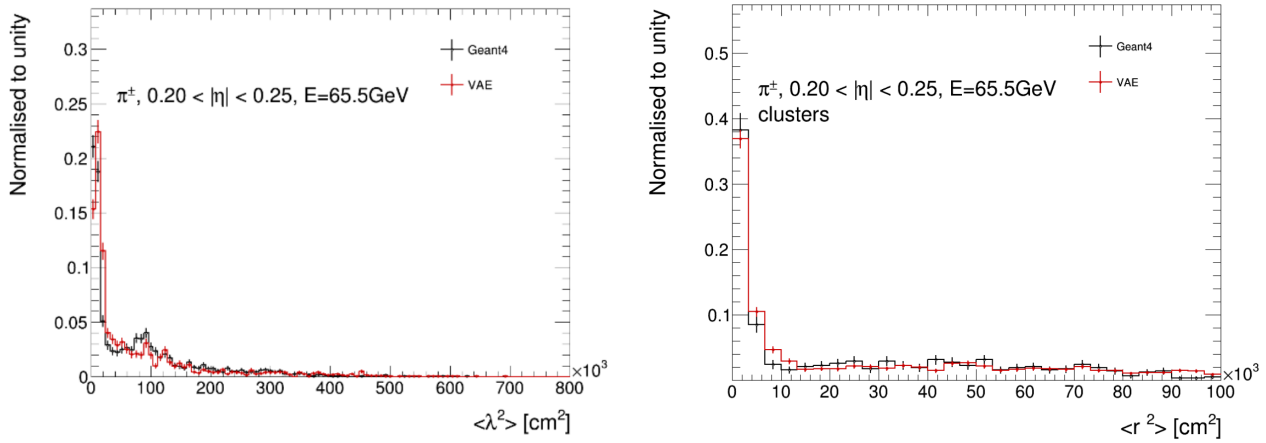
(a) (b)

Figure 144: Leading cluster (a) η and (b) ϕ distributions for pions with an energy of approximately 65 GeV in the $0.2< |\eta| <0.25$ range. The full detector simulation (black line) is compared to VAE (red line).



(a) (b)

Figure 145: (a) Center λ and δR distributions for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).



(a) (b)

Figure 146: Cluster moments (a) $< \lambda^2 >$ and (b) $< r^2 >$ for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

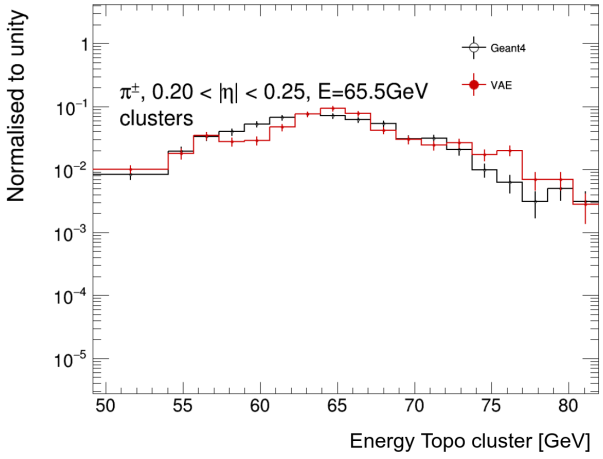


Figure 147: Topo-cluster reconstructed energy and energy RMS distributions for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

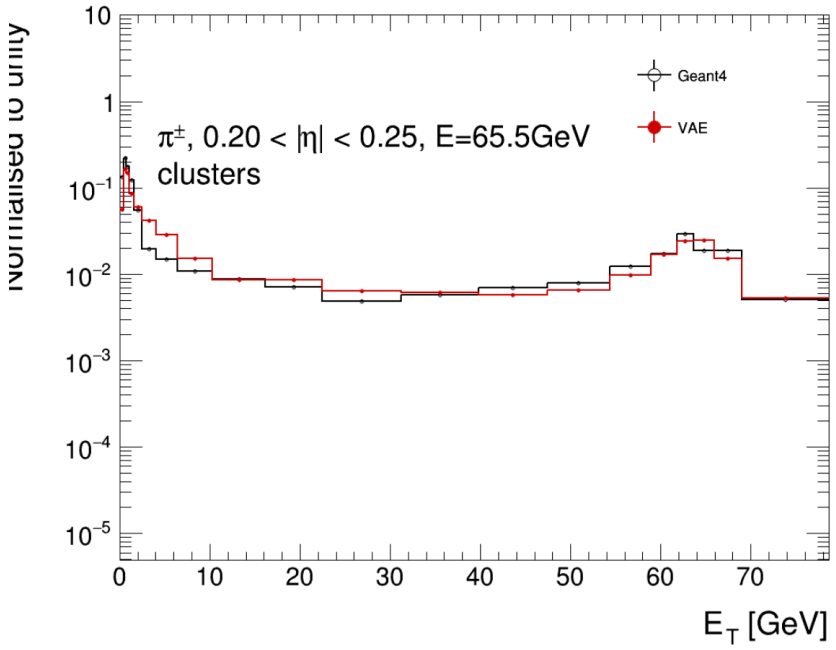
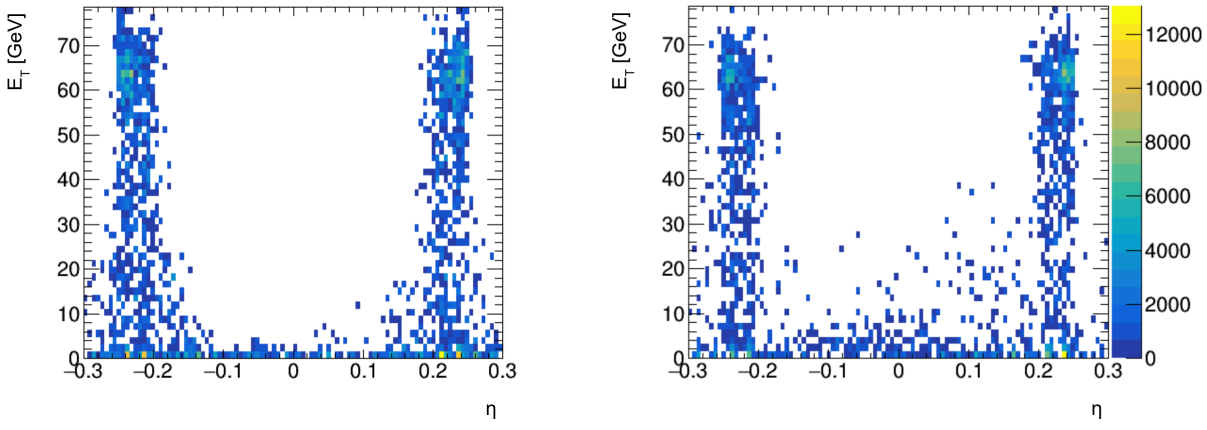


Figure 148: E_T distribution for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).



(a) (b)

Figure 149: E_T as function of *eta* for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (a) is compared to VAE (b). The number in the colorbar represent the number of entries in the 2D histogram.

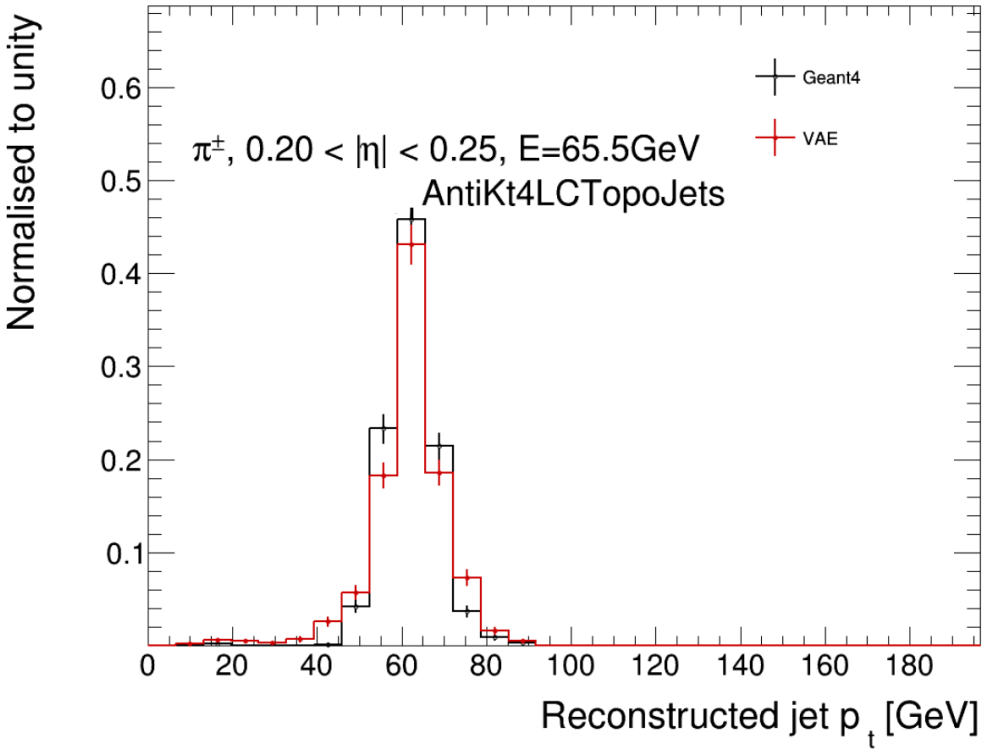
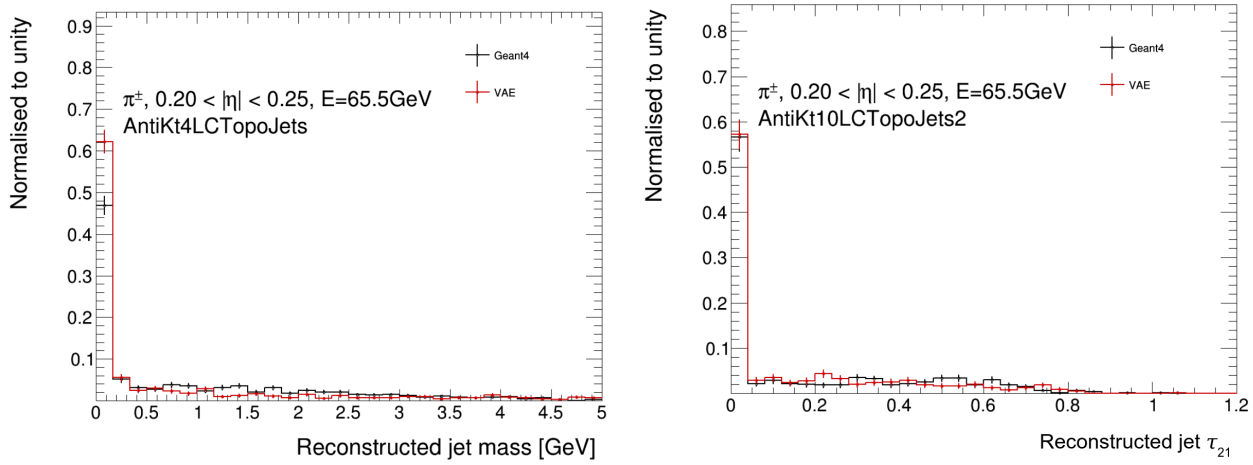


Figure 150: Reconstructed jet p_t for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).



(a) (b)

Figure 151: Reconstructed (a) jet mass (b) jet tau21 distributions for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

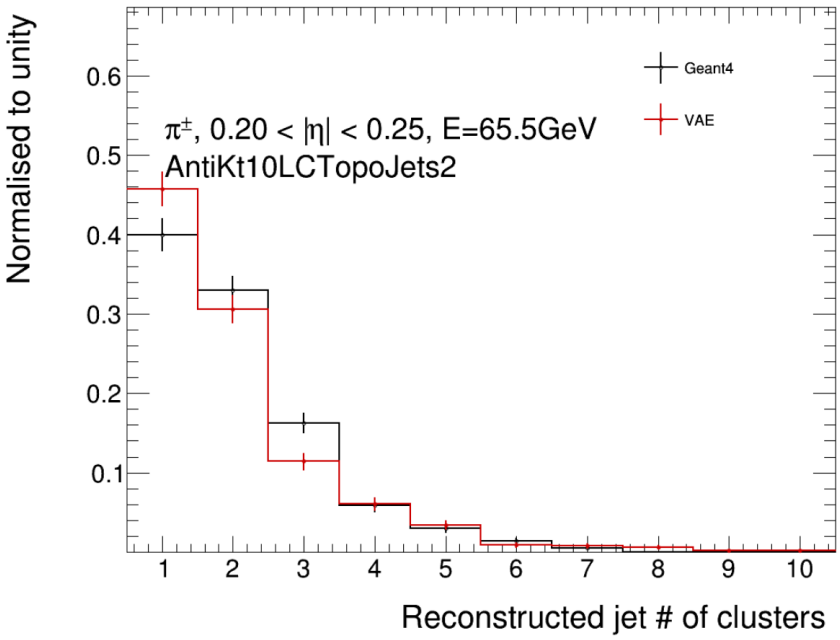
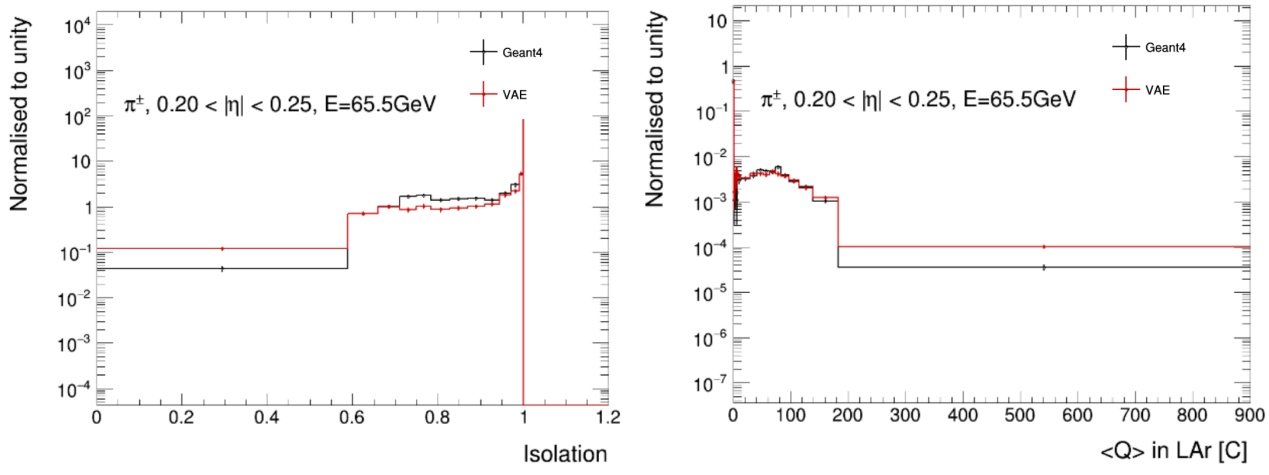


Figure 152: Reconstructed jet number of clusters for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).



(a) (b)

Figure 153: (a) isolation and (b) AVG Lar Q distributions for pions with an energy of 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (black line) is compared to VAE (red line).

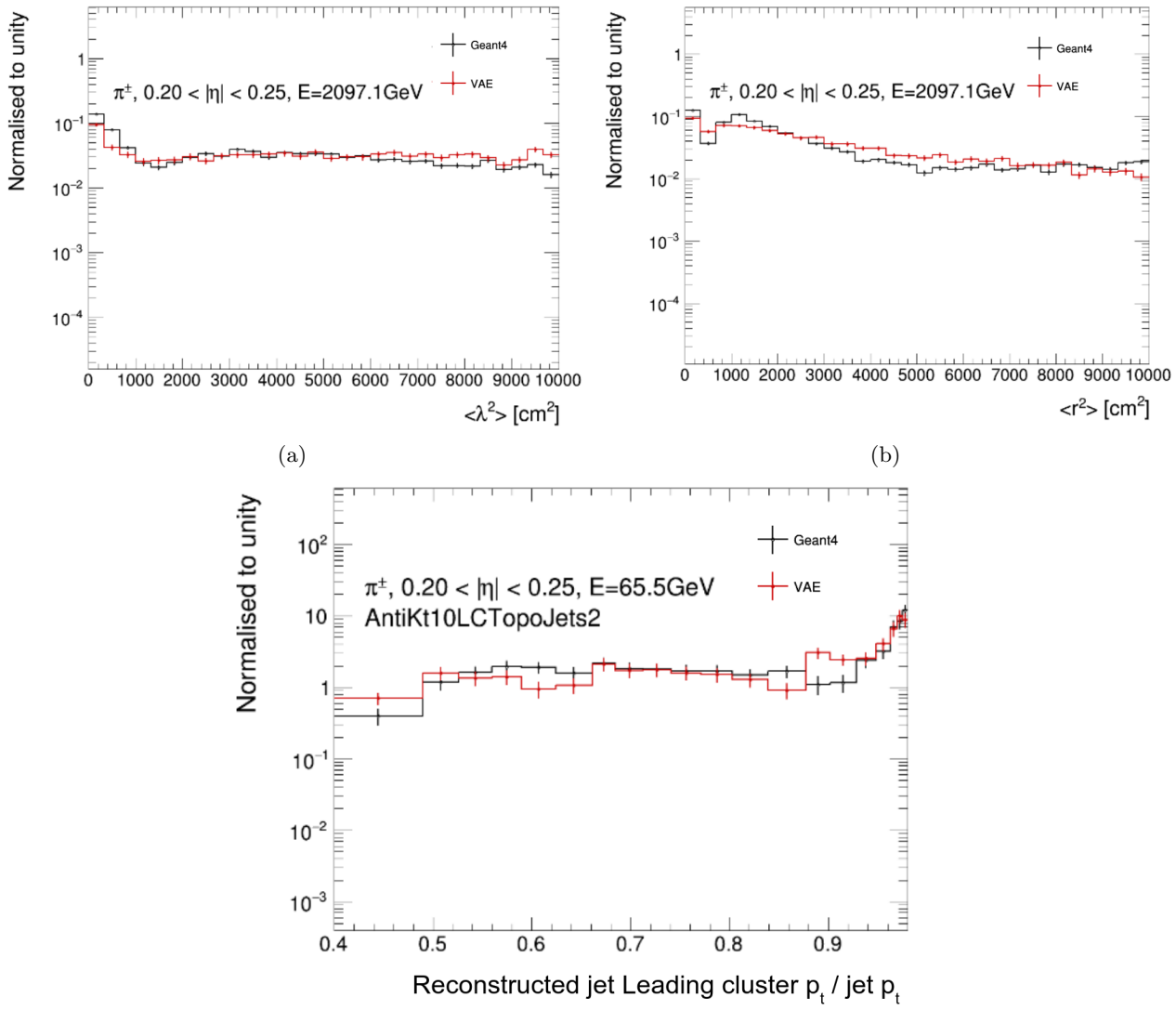


Figure 154: Secondary cluster moments (a) $< \lambda^2 >$ and (b) $< r^2 >$ and (c) the ratio of the jet leading cluster p_t over the jet p_t for pions with an energy of 2097.1 GeV in the $0.2< |\eta| <0.25$ range. The full detector simulation (black line) is compared to VAE (red line).

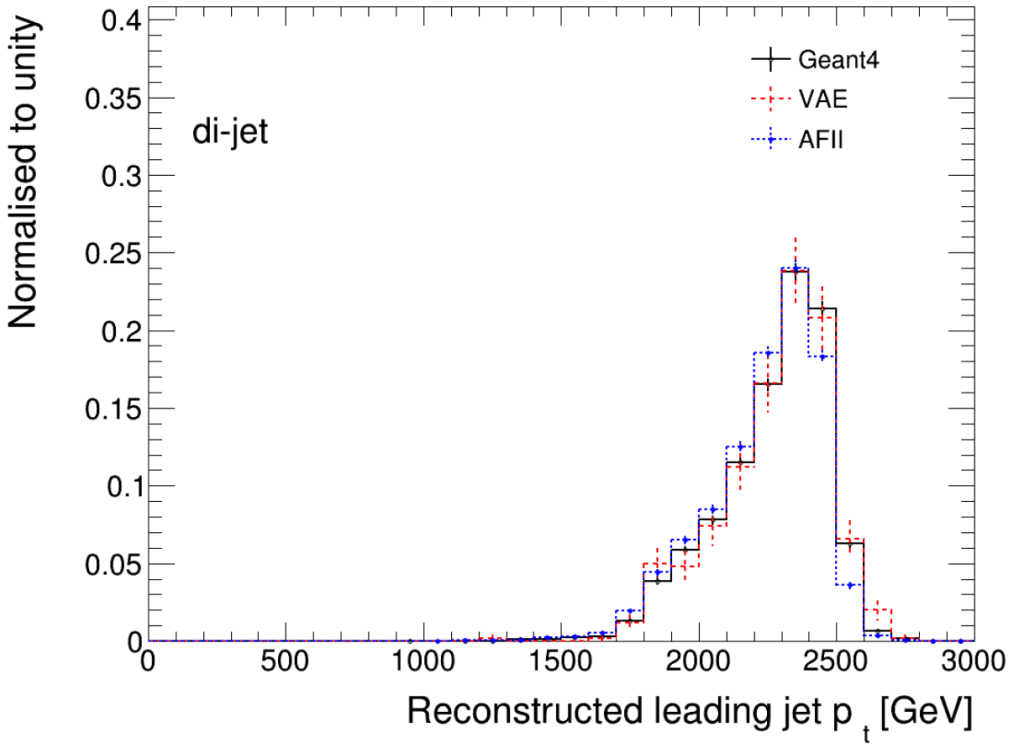
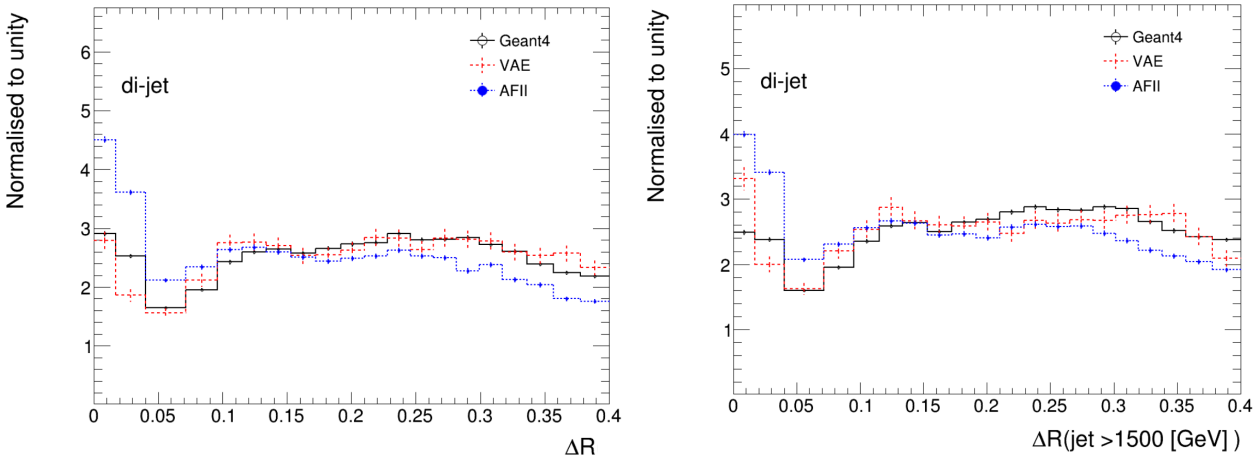


Figure 155: Jet p_T of the reconstructed leading jet in the di-jet sample. The full detector simulation (solid black line) is compared to VAE (dashed red line) and AFII (dashed blue line).



(a) (b)

Figure 156: (a) ΔR of the leading jet, (b) ΔR of jet>1500 GeV in the di-jet sample. The full detector simulation (solid black line) is compared to VAE (dashed red line) and AFII (dashed blue line).

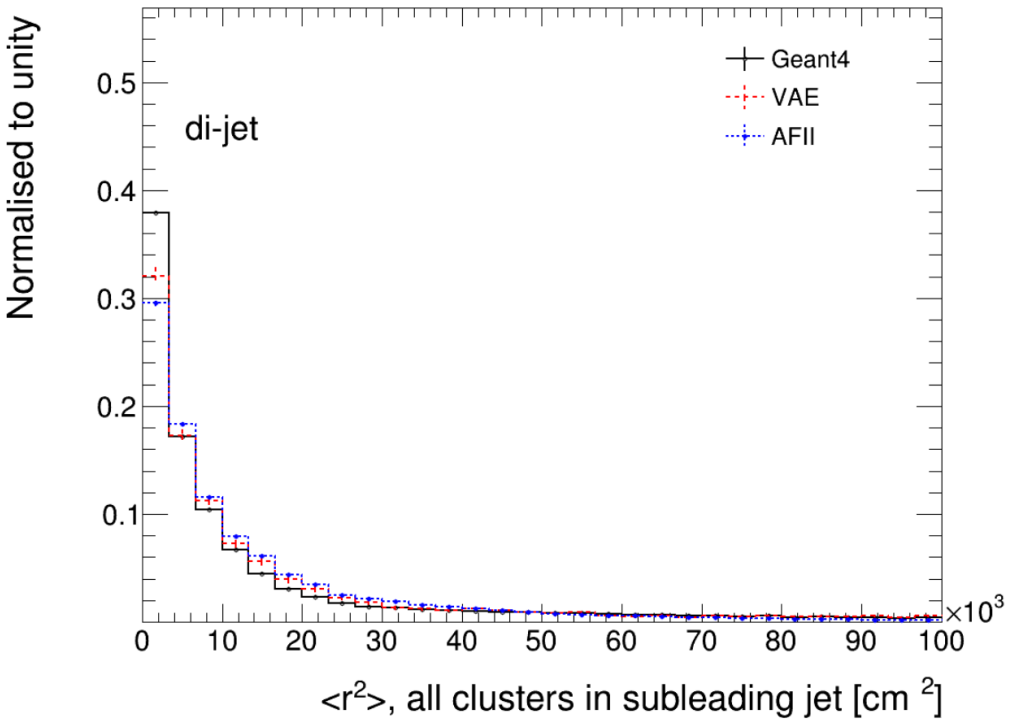


Figure 157: $< r^2 >$ in the subleading jet. The full detector simulation (solid black line) is compared to VAE (dashed red line) and AFII (dashed blue line).

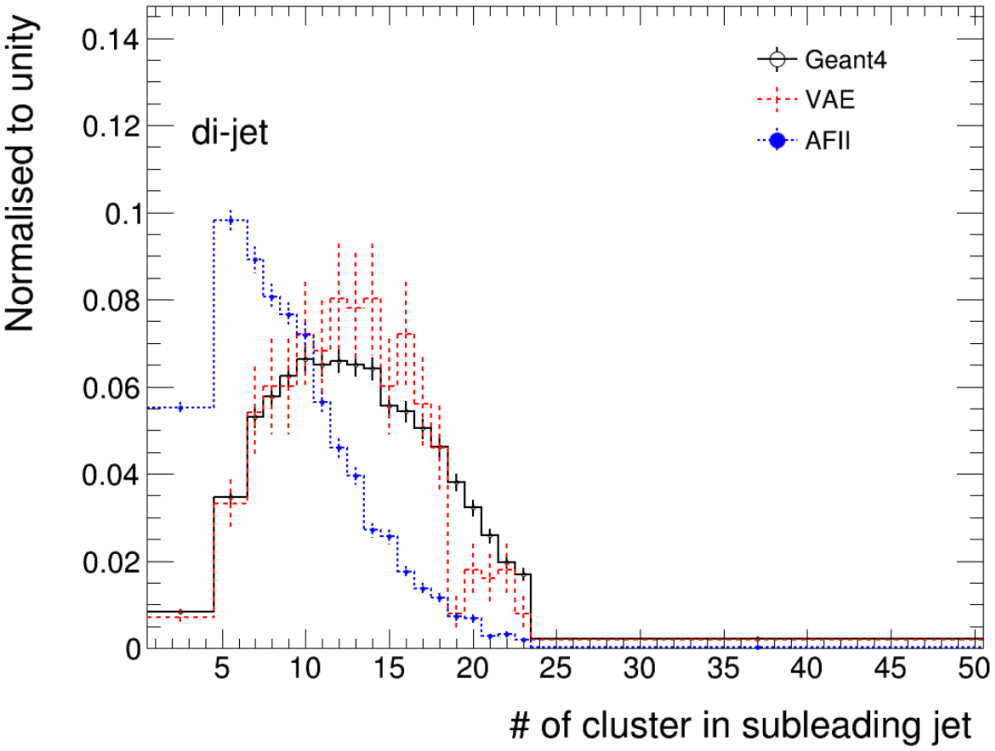


Figure 158: Number of clusters in the reconstructed subleading jet in the di-jet sample. The full detector simulation (solid black line) is compared to VAE (dashed red line) and AFII (dashed blue line).

regions of the ATLAS calorimeter are considered.

The first section of the cluster analysis study summarized a comparison between six of the most known clustering algorithms. It showed that K-means (MiniBatch variant) is the most suitable algorithm to use due to its scalability, adaptability and fast convergence time. This section described as well the development pipeline starting from Geant4 events to VAE simulated events. The pipeline translates all the implemented functions of preprocessing the Geant4 events using the K-means algorithm to validation of this clustering to the final VAE training. The same section presented how the clustering algorithm is used to derive the shower representations per calorimeter layer. Moreover, the K-means challenges study shows the impact of two main parameters of the algorithm: the initial positioning of the centroids and the batch size. The K++ initializer showed a better representation of the centroids. The section also presented the Voronoi polygons used to visualize these centroids and their clusters.

The second section summarized all the steps to train the VAE model on photons. The validation approach of the ML voxelization is detailed. The ML validation consisted of first evaluating the cluster quality using the inertia score. However, this score can not be, alone, a representative metric of a shower representation. To this end, a physics validation is used to ensure that the mapping from hit energies to cell energies is correct and that the shower shape is preserved. Thus, this validation is based on a set of shower observables compared between the Geant4 hits to cells mapping and the Geant4 hits to centroids to cells mapping. The physics validation itself is first presented in a standalone format where only the mappings are compared without running the full simulation chain. To further assess the quality of this ML voxelization, the validation based on running the full simulation and reconstruction chain, allows us not only to quantify the performance of the output K-means but also to optimize the number of centroids K.

The third section of the chapter, is dedicated to the VAE model trained on pions. All the validations and optimizations detailed for photons are applied for pions. The unique particle properties that pion showers exhibit, such as the complex correlations make it more difficult to optimize. In addition to that, there are further distributions for which it is hard to analyze the impact of the model parameters on them, such as the secondary cluster moments.

Overall, the VAE demonstrated a good performance over a wide range of shower observables. Some mismodelings are seen for photons in the total energies, mostly in the transition regions of the calorimeter for very energetic particles (1 TeV) and some η slices in the FCAL region. For pions the VAE shows overall shows a good performance of reproducing the cluster level, jet variables and di-jets with more discrepancies compared to the photon model.

11 FastCaloVSim : Summary, Comparison and Possible Improvements

From cells to voxels to centroids

Chapters 8,9 and 10 described the FastCaloVSim approach to simulate the energy response of the ATLAS calorimeter. The VAE model in each chapter optimized and extended the previous one in terms of input data structure, particle type, energy, and η slice.

FastCaloVSim at the cell level simulates only photon particles of logarithmically spaced energies from 1 GeV to 262 GeV at a central η slice. In its first version, the model training is performed on cell energies and then is performed on cell energy ratios. This reparametrization of the inputs allows the VAE to better model the total energy as well as the energy per layer. In parallel, training on ratios is augmented with physics knowledge by using a weighting term for each reconstructed feature encoded as the inverse of the standard deviation of the input distribution. These reparametrizations, physics knowledge incorporation and model optimizations led to a good performance of the VAE in reproducing the Geant4 shower distributions.

FastCaloVSim at the cell level offers many orders of magnitude in computational speedups compared to Geant4. Figure 159 shows a comparison of the simulation time of a single photon event of FastCaloVSim versus Geant4. Both simulations are performed on the same CPU hardware. In the full simulation, the time scales linearly with the particle energy, and it is significant for high energies. For FastCaloVSim there is a flat dependence as the showers are generated in the same manner for all energy points.

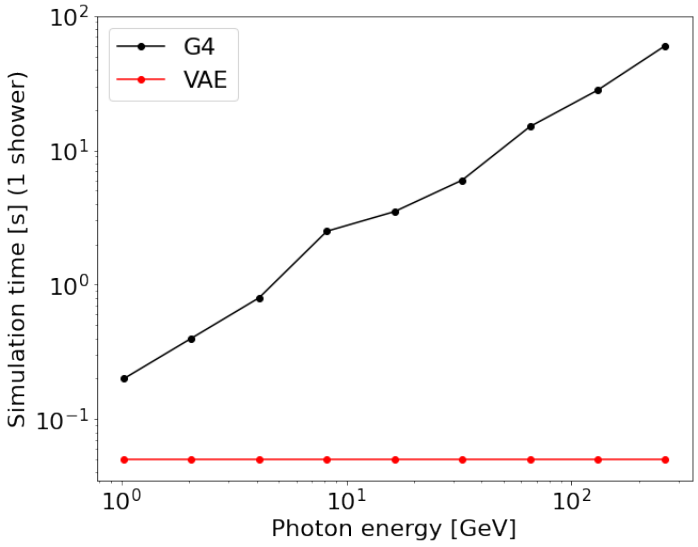


Figure 159: Simulation time for the VAE model (at cell level). The full detector simulation (black line) is compared to VAE (red line).

Moving from cells to (r,α) voxels is a motivated choice to be less dependent on the detector geometry. Therefore, extending the η range of the Geant4 training events is feasible with higher granularity voxels than the cells. Training on the voxels showed a good modeling of Geant4 observables for both photons and pions across energy and η.

The centroid level model shed light on how to design an ML simulator trained on Geant4 showers represented in volume spaces derived using an ML algorithm (K-means). This ML representation in fact is a motivated choice to build a flexible structure which can include showers from all η regions of the ATLAS calorimeter independently of the geometry. The results have shown good agreement to Geant4 on single particles and on dijets.

One of the key features that helped validate and optimize the clustering output of the K-means is the Athena based validation to compare the distributions of the Geant4 hits mapped to calorimeter cells and Geant4 hits grouped into K centroids and then mapped to cells. This validation is also used to compare between the (r,α) voxels and the centroids. Figure 161 shows two different quantities of shower observables in EMB2 with a K of 500. For the energy per layer, the difference is not noticeable. On the hand, in the lateral width $weta2$, the shift of the voxelization is significant. This variable is calculated with a window of 3×5 cells in EMB2 which implies that the granularity of the center region is sufficient to reproduce the width structure. The impact is not only related to the number of volume spaces in a layer, but also the size of these volumes and their distribution

	Cells	Voxels	Centroids
Number of calorimeter layers	4	4	16
Number of truth energies	9	11	11
η regions	1	16	100
Size of FastCaloVSim input/output	276 + 5	2424 + 5	8600 + 17
Size of the latent space	5	10	100
Simulation time for one event (s)	0.05	0.06	0.2
Resident memory in Athena (Gb)	0.15	0.7	1.7
Virtual memory in Athna (Gb)	2.1	4.04	6.59

Table 14: Comparison between VAE models using cells, voxels and centroids. The memory footprint shown here includes the full memory from an Athena job with Athena uses 2 GB for all the initialization and setup.

around the center of the shower. In addition, Figure 162 shows the E_{ratio} variable quantifying the evenly shared energy between the maxima of energy deposition in EMB1 comparing Geant4 to the output of the VAE. Even though having in total 1600 voxels versus 1000 centroids in EMB1, this does not lead to a better performance of this variable. Figure 47 shows a representation of the volume spaces of the cell, voxel, and centroid approaches where clearly a majority of the volumes for the centroid definition are in the core of the shower as opposed to the voxels where the same number of voxels is considered per r ring. Although both centroid and voxel definitions provide a variable size of the volumes around the center of the shower, the centroid sizes are not the same in the two dimensions, as shown in Figure 162(e). This means that in the case where the assignment of an energy from a single centroid to a cell is performed wrongly, it might not affect the other assignments too much because of this variable size. On the other hand, for voxels the size is the same within a ring, which can lead to multiple wrong assignments of multiple energies.

The number of volume spaces increases when moving from cells to voxels to centroids, and the VAE model evolves with each of the approaches. This includes the width of the input, output and hidden layers of the model. As a consequence, significant differences are seen between these models. To give an idea on this matter, Table 14 summarizes the major difference between the three models for photon particles.

The size of the input and output layers for the centroid based model is 8600 in addition to 17 fractions to represent the total energy and the energies per layer. This size requires a wider network compared to the voxels or the cells. A reduced input size is generally recommended in ML applications to ensure a better performance and avoid the model from overfitting. Nevertheless, in our case high granularity inputs allows not only to define a global model in terms of η, but also to better capture the structure of the shower in both core and tail regions. In order to avoid the model from overfitting, the regularization technique of batch normalization is used.

A key important quantity reported in Table 14 is the size of the latent space, which also evolves with the size of the input/output layers. The last two comparisons are related to the simulation time and the memory footprint. The time to simulate one shower increases from an approach to another. For the centroid model all showers from all energies are simulated in 0.2 s which is faster than Geant4.

The memory footprint, on the other hand, is a direct effect of the way the decoder is stored and loaded. The weights and architecture of the decoder are stored as a single JSON file. This representation of data is used in the C++ Athena framework for inference. The performance of the underlying serialization (and de-serialization) process is highly dependent on the type of the file. The current implementation of the LWTNN library supports only JSON files. JSON (JavaScript Object Notation) is a text format which is easily readable, self-contained and where everything is embedded with the object. These metadata details add further load on payload. Furthermore, the JSON structure has not a clear definition of types or in other words it is a simply type system (i.e. only numbers). This agnostic feature can have serious consequences on memory performance, especially at high data volumes, where the cost of serializing and de-serializing is expensive. If, for example, the weights of the model are unsigned 32-bit integers, JSON does not provide data type specification. As a result, when parsing the JSON file, a larger amount of memory is allocated for each of the values that is really needed.

In order to better optimize the memory footprint, alternative file storage formats can be considered, such as protocol buffers (protobuf). This file can be used within the ONNX runtime [208] for inference and deployment in Athena. Protobuf are implemented differently than JSON files to make the serialization operation faster. It is a binary file which improves the speed of loading, and thus it takes less space and bandwidth. Studies in [209] and [210] compare different libraries for object serialization in terms of size and execution time. Figure 160 shows these two quantities. The size of the binary serialized file in protobuf is more than two times smaller than the JSON file. The execution time of object serialization on the other hand, is more than five times less for protobuf than for JSON. Moreover, the ONNX runtime implements a set of optimization components such as graph optimizations and quantization which can drastically reduce the complexity of the model and

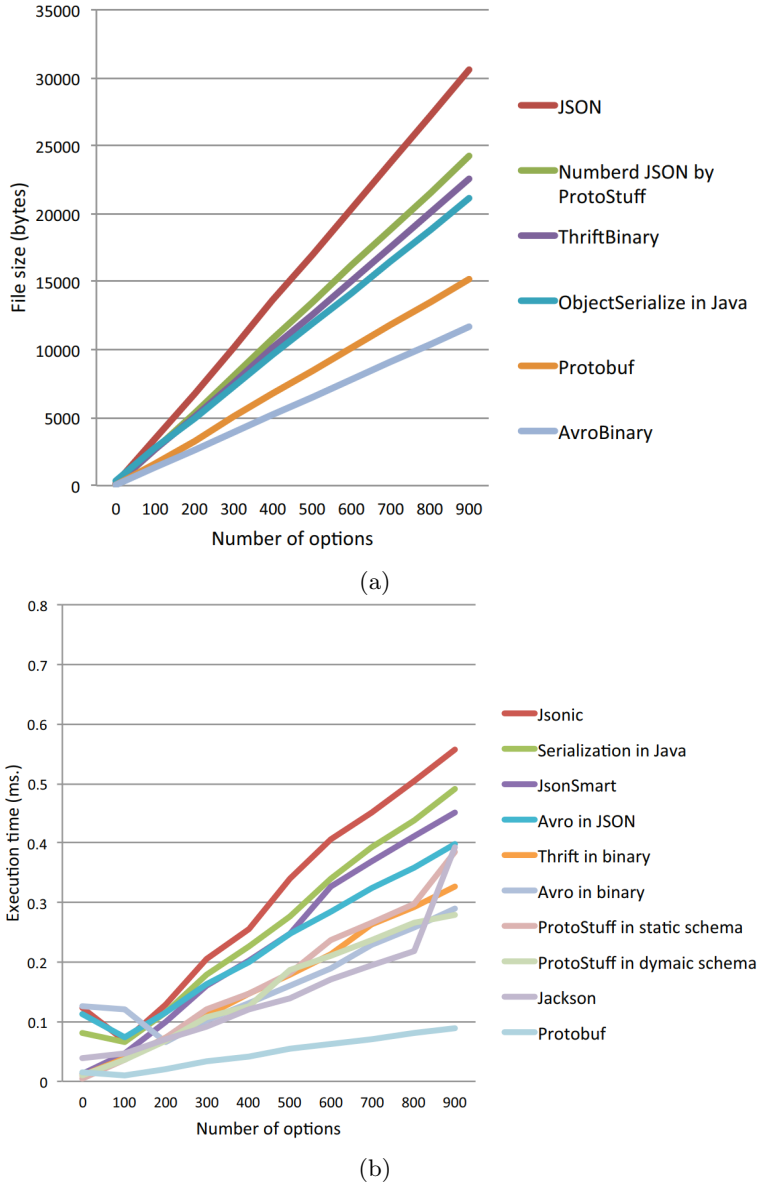


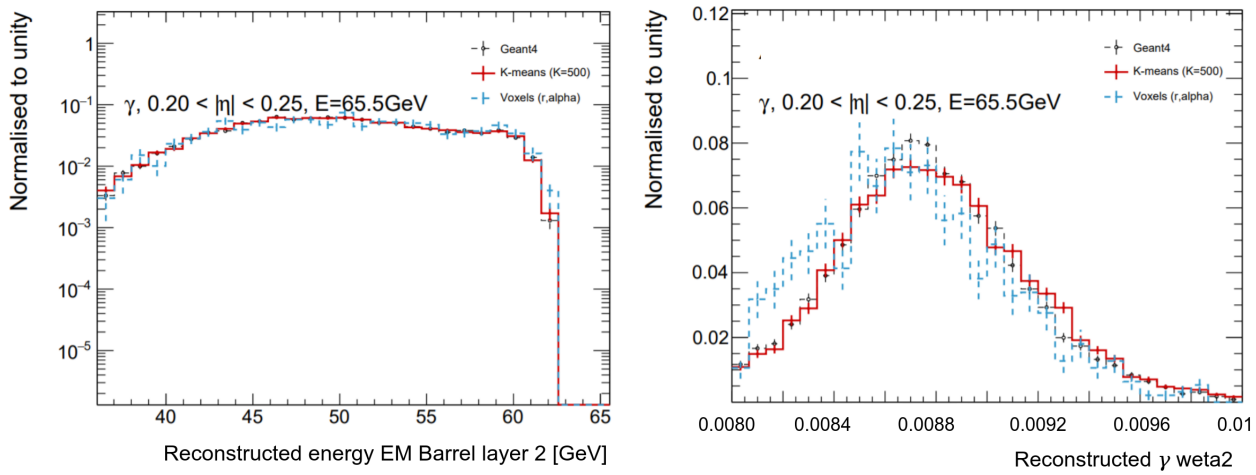
Figure 160: (a) Size of object serialization, (b) Execution time of object serialization [210]

therefore the memory footprint and the simulation time. Another alternative to reduce these two features is to use knowledge distillation [219]. This consists of transferring the higher knowledge capacity from a large to a smaller model.

Novel K-means alternatives

The clustering results have shown that the majority of the clusters are centered in the core region. This is explained by the fact that the algorithm computes the means to all data points and therefore shifts the centroids towards the highest density population. For future developments, a better definition of the shower resolution in the core and the tail region can lead to more accurate simulator.

We implemented and tested new versions of K-means clustering algorithm in order to control the distribution of the number of centroids in the core and in the tail. 2-steps K-means, for example, consists of first defining a $dr_{threshold}$ where most of the energy is deposited (95%). A first K-means step is applied on hits with $dr_{core} \leq dr_{threshold}$ and the second step is applied on hits with $dr_{threshold} \leq dr_{tail}$. Figure 164 illustrates the dr distribution weighted by the energies of hits in EMB2 comparing the hits to cells assignment with the hits to centroids to cells. By applying only the standard K-means with K equals to 100, it is clear that at large dr only few centroids are formed and because of their large size their energies can be assigned to the wrong cells. Using 2-steps K-means allowed to better reproduce the structure of the tail. However, the K value of both core and tail should be carefully chosen to avoid creating, if $K_{tail} >> K_{core}$, a bias of successive high and low densities of centroids as shown in Figure 163 when K_{tail}=500 and K_{core}=100.



(a) (b)

Figure 161: (a) Energy in EMB2 and (b) weta2 distributions for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to K-means (solid red line) and voxels (dashed blue line).

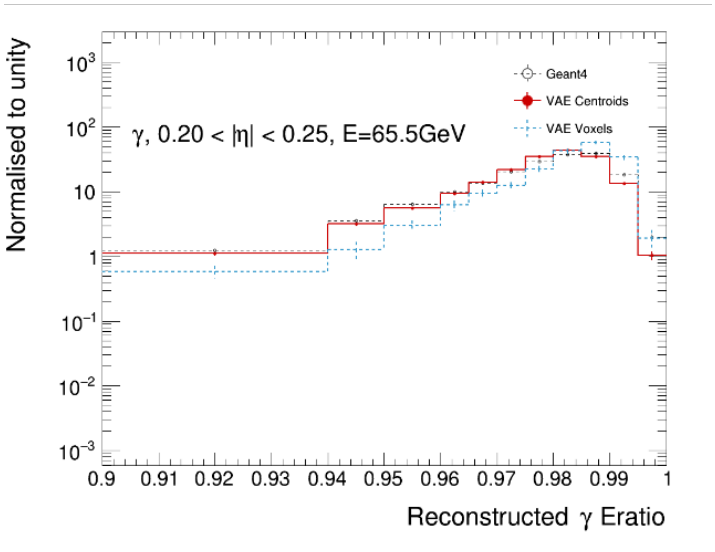


Figure 162: Eratio distributions for (a) VAE centroids and (b) VAE voxels for pions with an energy of approximately 65 GeV in the 0.2< $|\eta|$ <0.25 range. The full detector simulation (dashed black line) is compared to K-means (solid red line) and voxels (dashed blue line).

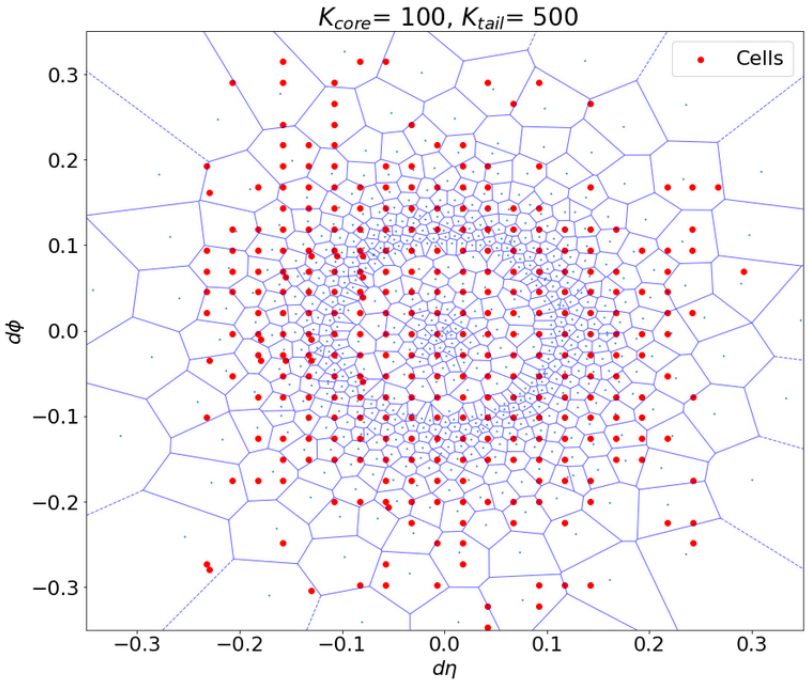


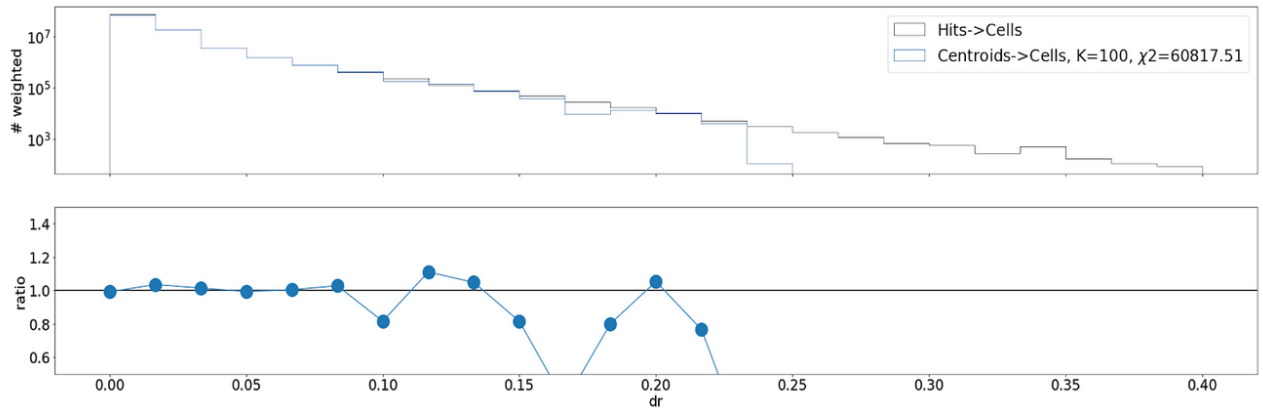
Figure 163: Voronoi diagram derived from 2-steps K-means with K_{core}=100 and K_{tail}=500.

An adaptive K-means is an automatic alternative if the resolution is computed based on the area of the volume spaces of the clusters w.r.t. the area of the cells. The algorithm then computes the area of these volumes using the Voronoi polygons. The adaptation consists of modifying the stopping criteria such that the K-means stops when all $f\times$ polygon areas $\leq$ cell areas, where f is a factor of choice on how granular the definition should be. However, with K being an input parameter to the algorithm, it is not guaranteed to converge and reach this stopping criteria. Given the shortcomings of the previous techniques, we propose a novel technique named Hierarchical K-means. It starts by applying K-means with K=3, computes the area of the polygons and in the next iteration if $f\times$ polygon area $\leq$ cell area then apply K-means with K=3. The stopping criteria is the same as the adaptive K-means definition. Figure 165 shows the algorithm application over the iterations. The red circles represent the areas where the stopping criteria is not reached when computing the area of the polygons after each iteration. Compared to all the other implementation of K-means, the hierarchical version is the fastest. Moreover, its convergence is guaranteed. On the other hand, the number of centroids increases quickly.

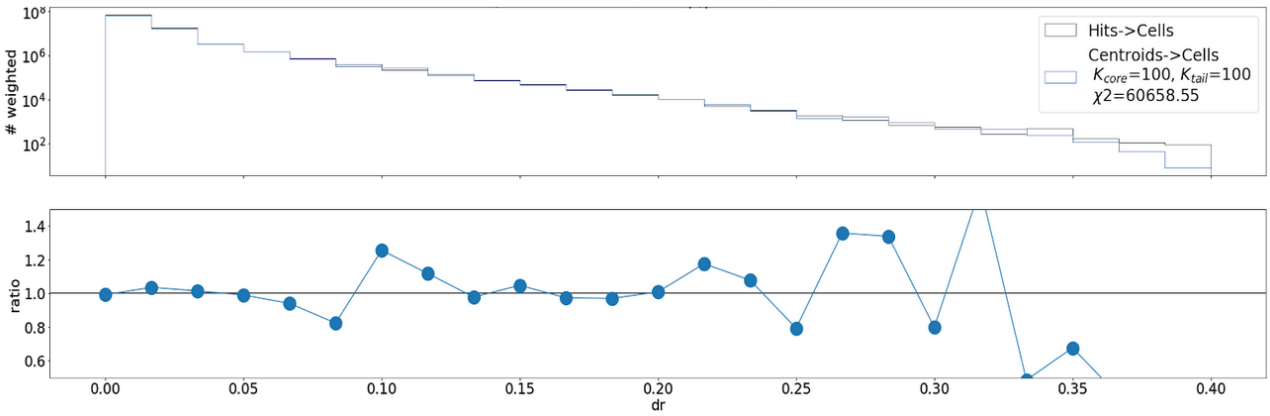
The hierarchical K-means should be the most optimized shower representation, since it uses a clustering algorithm that is independent of the number of centroids and a physics stopping criteria. Compared to the standard K-means, the total number of centroids, from all the layers, can be very large. This increases significantly the number of trainable parameters, impacting the training and inference time as well as the memory. In order to be used in production, this approach would need extensive optimizations such as hardware acceleration and model pruning. After these considerable optimizations, the hierarchical K-means could lead to optimal results both in precision and runtime. However, in this thesis, we propose to investigate the trade-off between precision and runtime, accepting a small loss in precision with the faster alternative of using a standard K-means.

Fast simulation approaches in ATLAS

In this section, we present a comparison of fast simulation approaches in ATLAS FastCaloSim (FCS) and FastCaloGAN. FastCaloGAN [211] is another GAN model developed in ATLAS by another group, and it is a different GAN approach from the one presented in Chapter 8. It uses 300 GANs, one for each particle type and η region. FastCaloGAN learns to simulate calorimeter showers using a voxelization procedure in (r, α) which is different compared to the one presented in Chapter 9. The FastCaloGAN voxelization is derived for each particle type, η slice and relevant layer. The relevance is determined using only the 1 TeV energy points with energy fractions larger than 0.1 %. Moreover, only layers which have a large fraction of the total energy are binned along α. This results, for photons and electrons, in having for example 170 voxels in EMB1 with 17 bins in r and 10 bins in α (for FastCaloVSim, the number of centroids is 1000). FastCaloGAN is based on the Wasserstein GAN variant with a gradient penalty term. A schematic representation of its architecture is presented in Figure 166, based on dense layers and a latent space has a dimension of 50.

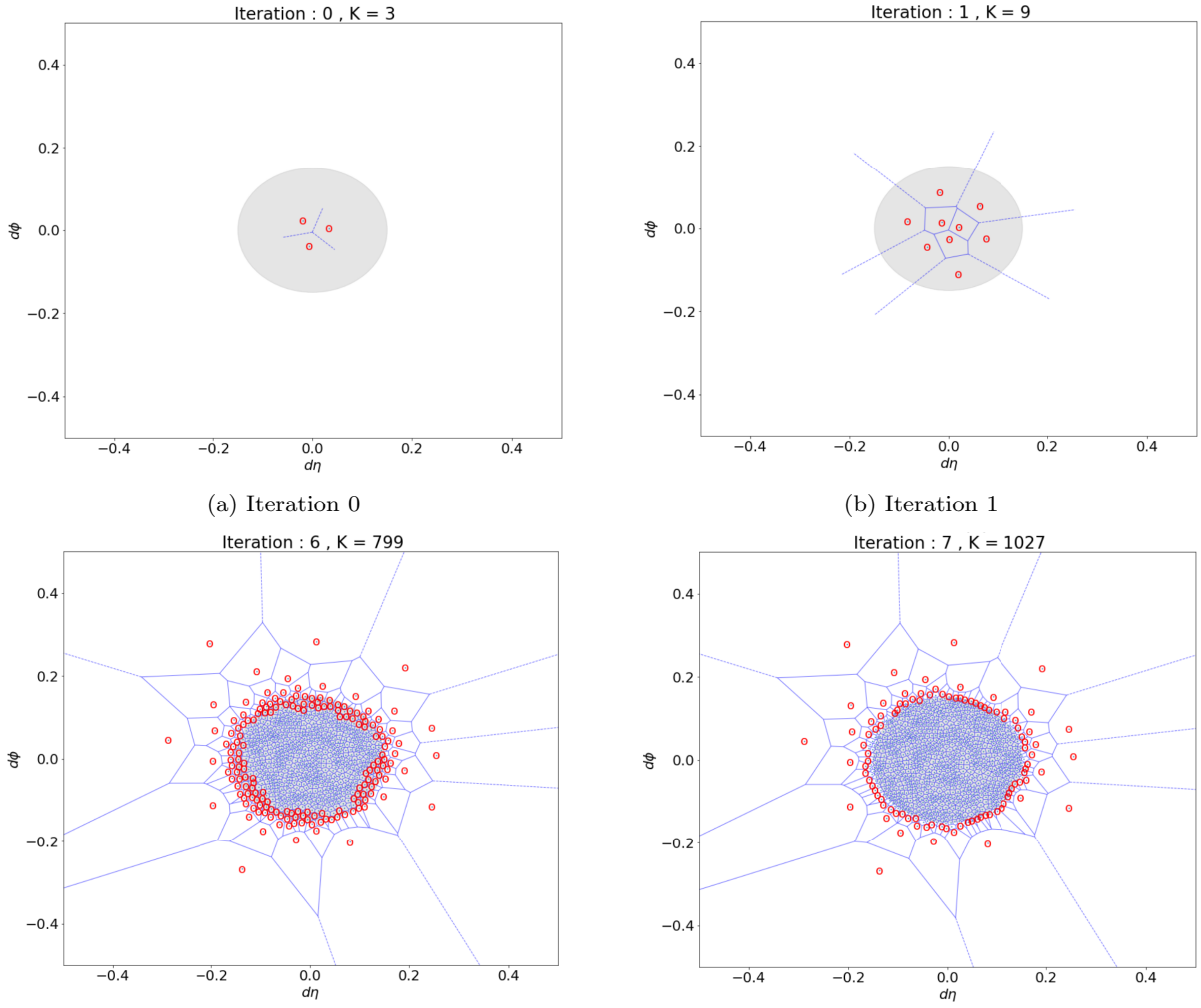


(a)



(b)

Figure 164: *dr* distribution weighted by energy of the hits in EMB2 for photons with an energy of 1048 GeV in the $0.2< |\eta| <0.25$ range, (a) standard K-means and (b) 2-steps K-means.



(a) Iteration 0

(b) Iteration 1

(c) Iteration 6

(d) Iteration 7

Figure 165: Hierarchical K-means over four different iterations. The gray circle represents a dr of 0.2 where it contains 99% of the shower energy deposited in EMB2. The red circles represent polygon areas where the values are greater than the cell area divided by a factor f. In this case, f=4.

Figure 167 shows the performance of FastCaloGAN in the central η region using Athena validation. The energy per layer is better modeled by FastCaloGAN compared to the total energy, where it shows a shift. The same figure contains as well a standalone plot where the means and RMS of each of the total energy distributions across the energy range is reported as a function of the true momentum. This shows a good agreement to Geant4. FastCaloGAN has also shown a good performance on pions. Figure 168 illustrates that the average p_T is better reproduced than the width of the distribution. For the longitudinal and extension $< \lambda^2 >$ of all clusters in the leading jet, at high values $< \lambda^2 >$, can better reproduce the distribution than in the low values regime. For the lateral extension $< r^2 >$, it is equally well modeled in both regimes.

Table 15 summarizes a set of characteristic components of FCS, FastCaloGAN and FastCaloVSim at centroid level. Compared to the FCS approach which factorizes the shower parametrization into longitudinal and lateral energy components for the different energies, an ML approach provides a simpler alternative solution to learn the process of the shower development in the calorimeter as a single step. For the number of particles and energies FCS and FastCaloGAN use a larger range, while for FastCaloVSim only photons and pions were considered with energies ranging from 1 GeV to 1 TeV compared to 15 energy points covering the range from 256 MeV to 4 TeV. Extending these ranges for the FastCaloVSim is quite straightforward.

For the preoprocessing, FCS uses a fixed binning voxelization where a 1 mm bins are used in the r direction for EMB1 and EME1 layers and 5 mm for the other layers. For α, 8 bins are used for all the layers. For FastCaloGAN, having a variable bin width in r can create voxels which are larger than the cells of the calorimeter, and thus can lead to a mis-assignment of energies from voxels to cells. Following the centroid's definition of FastCaloVSim there are about 40 polygons at 2 cell radii in EMB1 which is much more granular than FCS or FastCaloGAN. This high granularity demonstrated a better learning of key shower quantities such as the *Eratio*.

The storage feature in Table 15 refers to what each of the models needs to store in order to generate showers within the Athena framework. For FCS, a parametrization file is saved for each particle type, energy point and η slice, or alternatively they can be saved as a single large file. For the ML models, the generators are saved. FastCaloGAN trains 300 GANs storing all of them independently. For FastCaloVSim, a single model is trained per particle type which means that only this file is saved, which can significantly reduce the storage space. The optimization for FCS or FastCaloGAN needs a dedicated approach for each particle, energy and η slice independently, which requires a lot of manual scanning. On the other hand, for FastCaloVSim any form of optimization is performed on a single model for each particle type.

Since the parametrization or the learning of shower simulation for all approaches is performed using logarithmic spaced discrete energy points, the interpolation between energy points is an important property which allows us to get the full energy spectrum falling between the lowest and highest energy points. The advantage of learning a conditional model on the energy parameter for the ML based approaches, provides a direct solution to this task, where interpolation has shown to be efficient. On the other hand, for FCS, the interpolation is based on a spline functions that fits the total energy response. These functions are also defined for each particle type and each η region.

Different corrections are applied to FCS to accurately describe the simulated events compared to Geant4. The total energy resolution, for example, is corrected by reweighting the FCS distribution to the Geant4 distribution. A second correction consists of applying a calibration when reconstructing photons and electrons. This corrects the total energy modulated in the ϕ direction due to the accordion shape of the EM calorimeter of the ATLAS detector. The third correction, is applied on the total energy of hadrons to differentiate between the response of charged pion and other hadron variants. The residual energy response correction, on the other hand, adjusts the total energy of all particles to match the average energy of Geant4 after all the steps of the ATLAS simulation chain. All these corrections are not included to the ML based solutions. Therefore, all these effects, specifically the ϕ modulation, can be included in future versions of FastCaloVSim to correct for the accordion structure of the calorimeter to have a correct modeling of the Higgs mass.

	FCS	FastCaloGAN	FastCaloVSim
Approach	Factorize shower into components (longitudinal and lateral for the different energy points)	Generative model GAN to learn shower energy deposition	Generative model VAE to learn shower energy deposition
Number of particles	3	3	2
Number of discrete energy points	17	17	11
Preprocessing	Voxels (fixed binning)	Voxels (defined per layer and η)	Centroids (defined per layer)
Storage	Parametrization files per energy point and η slice (or one big parametrization file)	300 GANs	2 VAEs (depends on the number of particles)
Model optimization	Per particle, energy point and η slice	Per particle and η slice	Per particle
Energy interpolation	Fit Spline function	Direct application	Direct application
Number of corrections applied	4	0	0

Table 15: Comparison between fast simulation approaches in ATLAS : FCS, FastCaloGAN and FastCaloVSim.

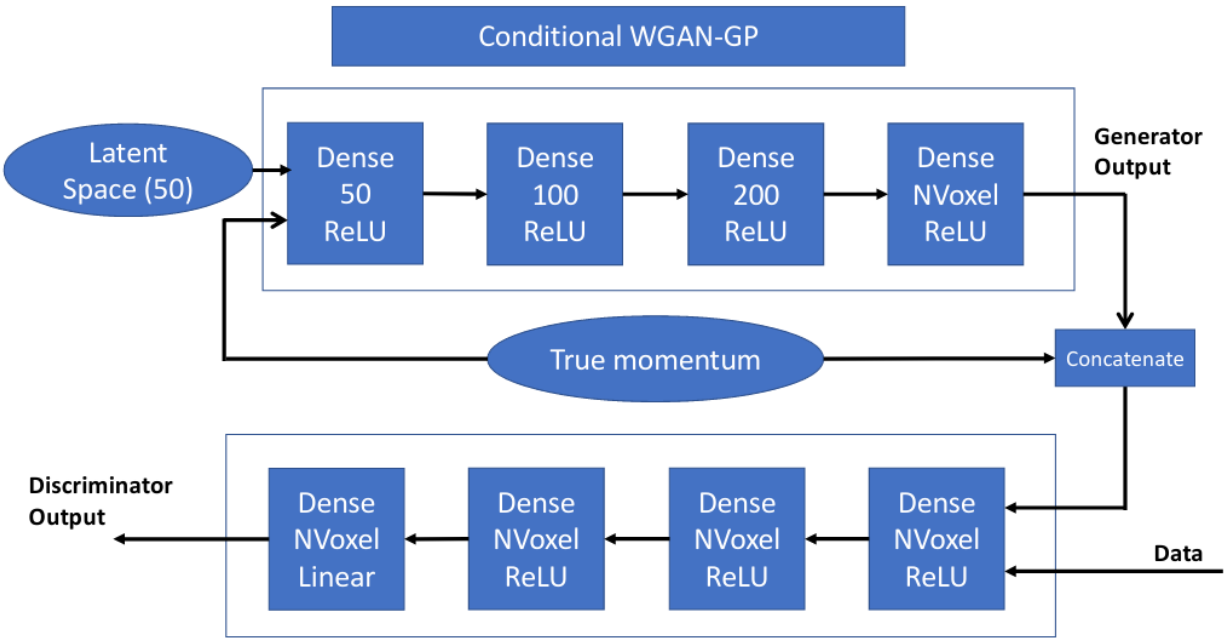
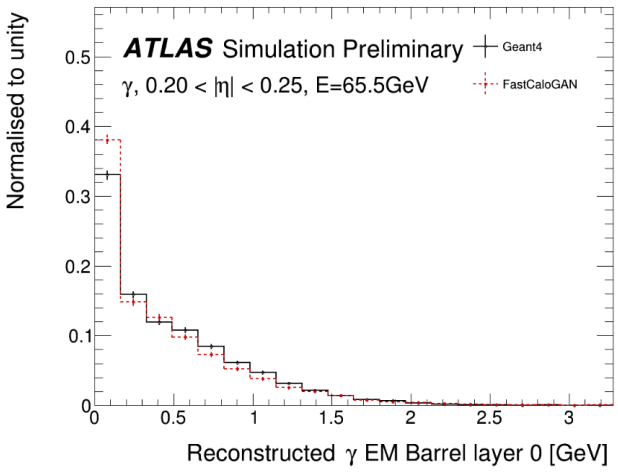
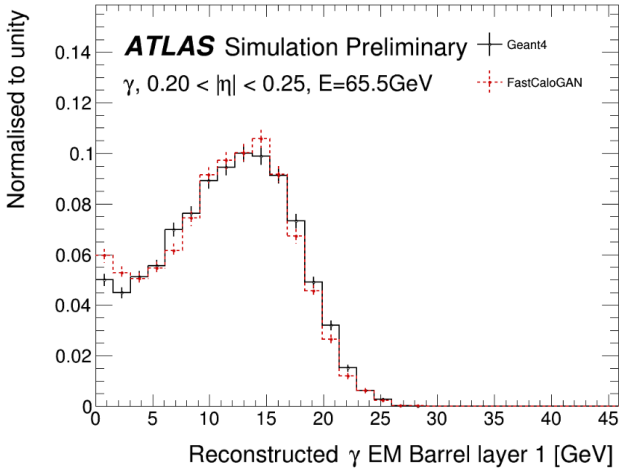


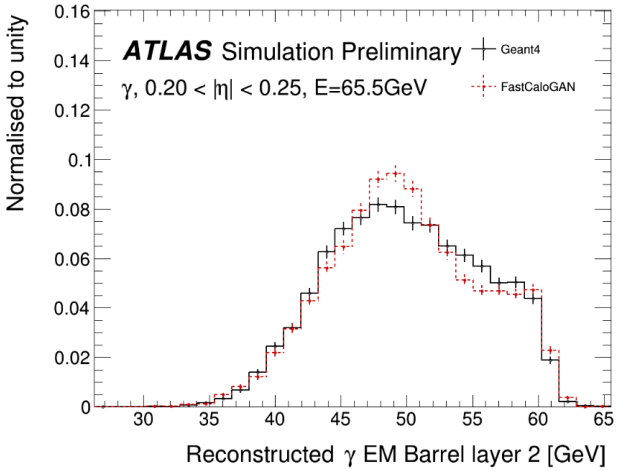
Figure 166: Schematic representation of the architecture of the GANs used by FastCaloGAN [211].



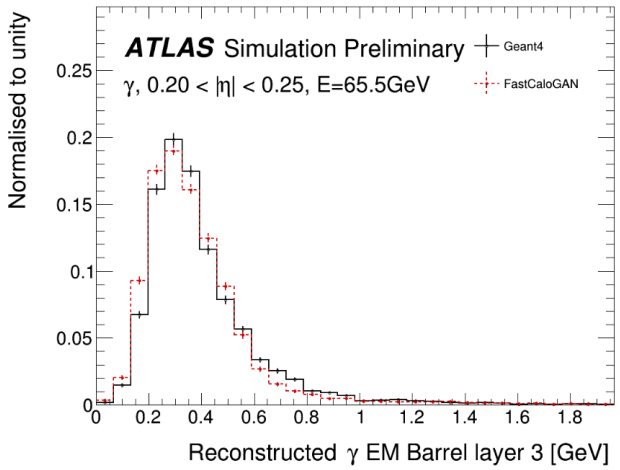
(a) Reconstructed photon energy in EMB0.



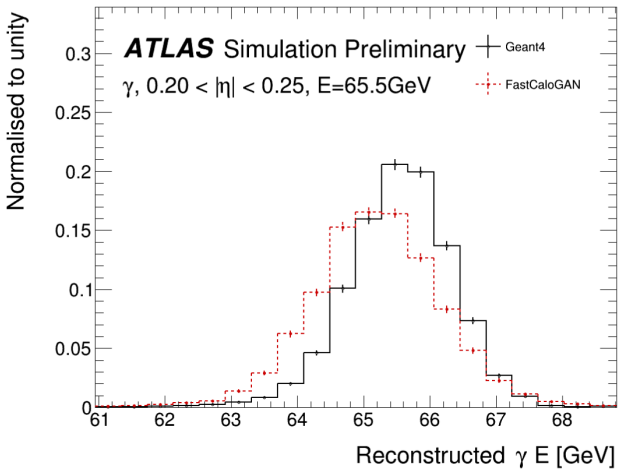
(b) Reconstructed photon energy in EMB1.



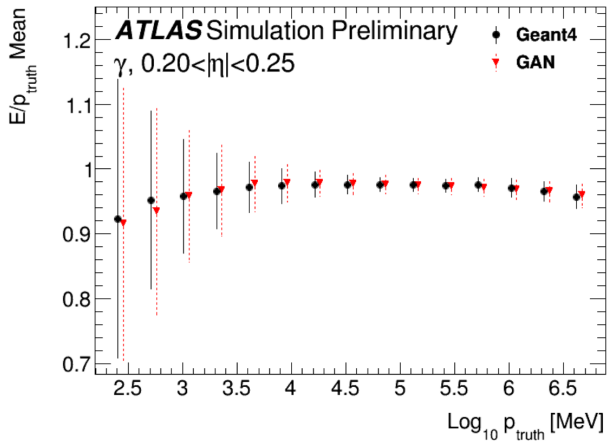
(c) Reconstructed photon energy in EMB2.



(d) Reconstructed photon energy in EMB3.

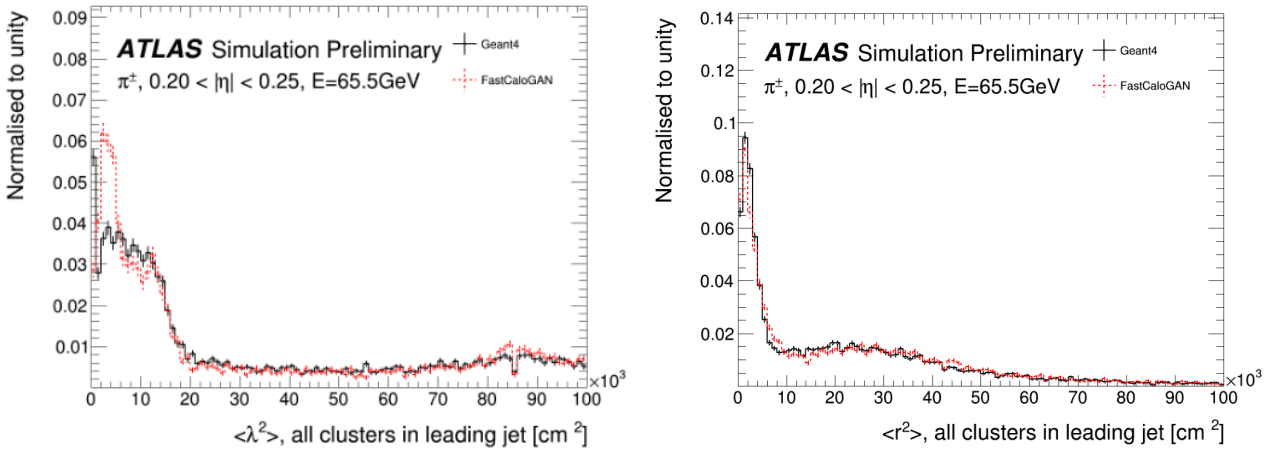


(e) Reconstructed total energy for photons.



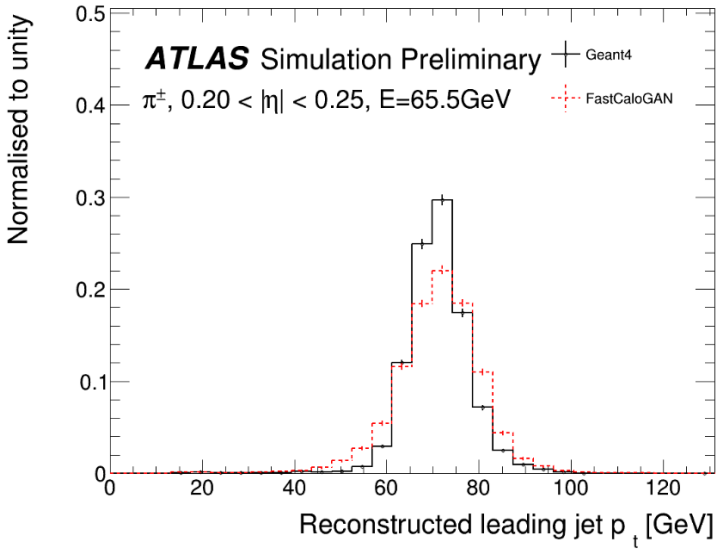
(f) Total energy response normalized to the truth momentum as function of the true momentum. distribution.

Figure 167: Performance of FastCaloGAN for photons in $0.2 < |\eta| < 0.25$ with an energy of approximately 65 GeV. For (f) the error on the data point show the RMS of the total energy [211].



(a)

(b)



(c)

Figure 168: Performance of FastCaloGAN for pions in $0.2 < |\eta| < 0.25$ with an energy of approximately 65 GeV [211].

12 CoVAE: Modeling the Correlated Fluctuations with Variational Autoencoders

FastCaloSim (FCS), described in Chapter 5, is a simulation tool which relies on parametrization of the particle shower development. These parametrizations can be split into longitudinal and lateral. The former describes the outward propagation of energy from the interaction point, and the latter models the shape of each particle shower. As depicted in Figure 33, the lateral parametrization constructs an average shape for each particle type, energy, η and layer using Geant4. This shape is then used during simulation, as a probability distribution from which a finite number of hits are randomly sampled. This number is computed from the energy deposited in each layer and the intrinsic resolution of that layer. Therefore, $N_{hits} \sim Poisson(1/\sigma_E^2)$, where the resolution is defined as $\sigma_E = a/\sqrt{E} + c$, for a given energy E simulated within a calorimeter layer with a (c) being a stochastic (constant) term. For EMB2, for example, the values are 10.1% and 0.2% [54] for a and c respectively. The simulated hits have equal energies with $E_{hit} = E/N_{hits}$. Therefore, the sampling process introduces a statistical fluctuation of $\sigma_S = a/\sqrt{\sum E_{hit}} + c$, away from the average shape. This is also known as the random fluctuation or the uncorrelated noise.

Figure 169 shows, in the left plots, the average shape for a 65 GeV photon and pion in a 5×5 grid of calorimeter cells in EMB2. The right plots show the ratios of single Geant4 events with respect to the left averages. This shows that pions have larger and non-trivial deviations from the average shape. Moreover, pions ratios are characterized by large deviations compared to photon ratios. This illustrates that current shape simulation, with purely random fluctuation, is not sufficient because it neglects the correlated part, where for a given simulated event the energy ratios will look different depending on how the energy is shared between the neighboring cells and layers.

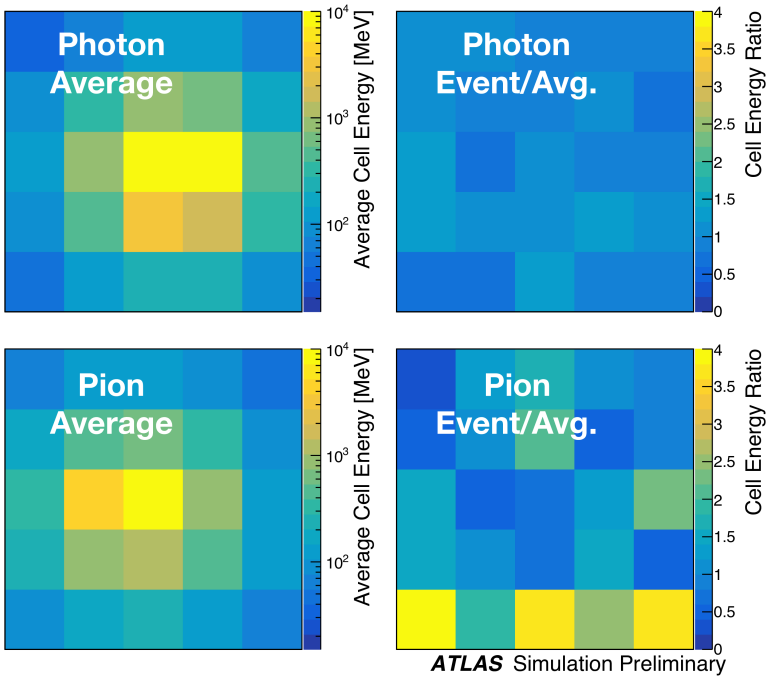


Figure 169: Examples of 65 GeV central photon and pion average shapes (left column) and the ratios of single Geant4 events to the given average (right column) in a select 5x5 grid of calorimeter cells in EMB2. Electromagnetic showers are generally simpler and can be modeled by small deviations from some average shape. Hadronic showers tend to have larger, more non-trivial deviations. Correspondingly, the photon ratio is very close to one in all cells, while the pion ratio has larger deviations [212].

The correct modelling of these fluctuations is a requirement to model substructure and boosted topologies. To address this problem, a CoVAE approach is implemented. It uses a VAE model to learn the differences from the average shower shape. The VAE learns to model event by event fluctuations of shower shape and correlation across cells or voxels. This is achieved by training on energy ratios with respect to the average. These correlated fluctuations are then added on top of FCS as weights.

In this chapter, a VAE model is first designed to learn the correlated fluctuations of photons to describe electromagnetic showers and then pions to describe hadronic showers.

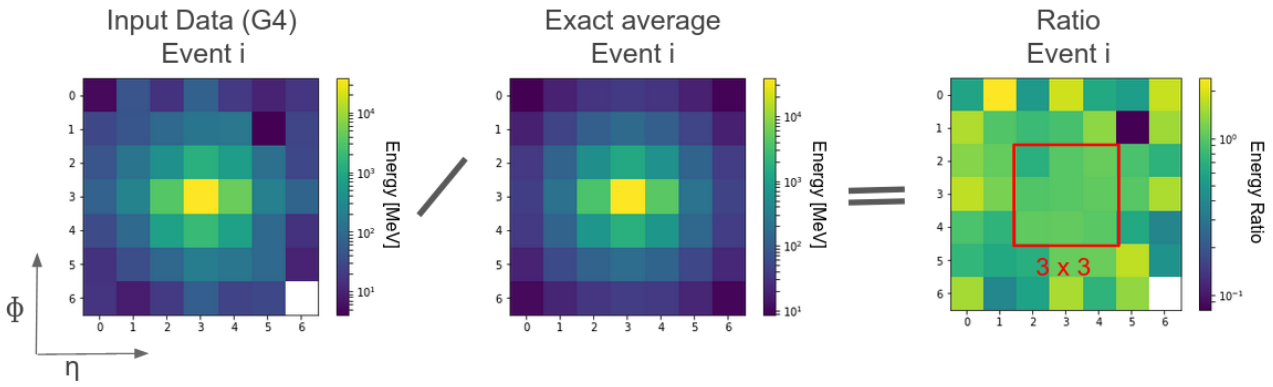


Figure 170: A given Geant4 event of 7×7 cells in EMB2, its exact average and the energy ratios result. The 3×3 cells represent the core cells.

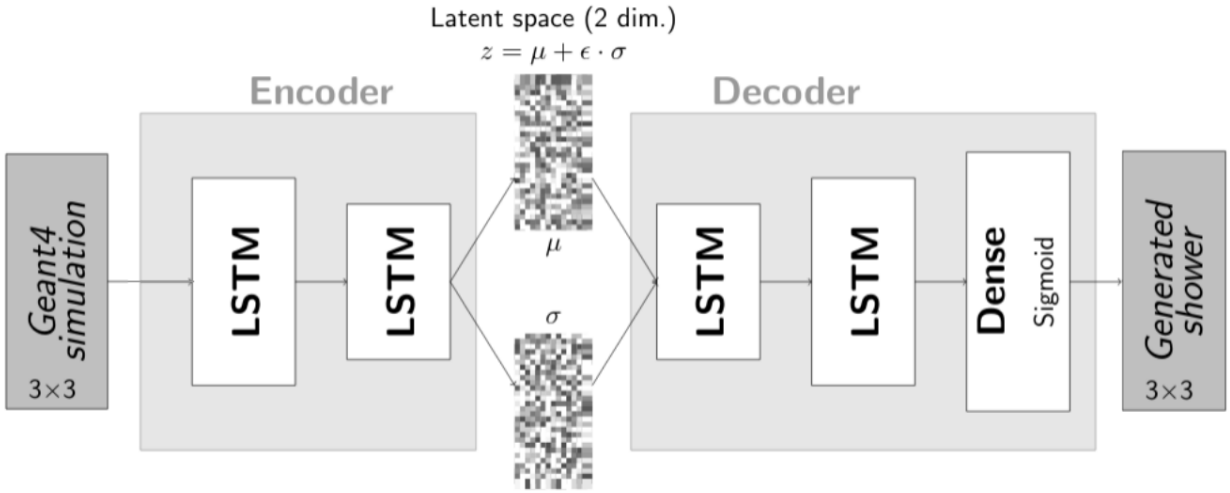


Figure 171: VAE architecture : number of units per layer and number of layers are shown for both the encoder and the decoder.

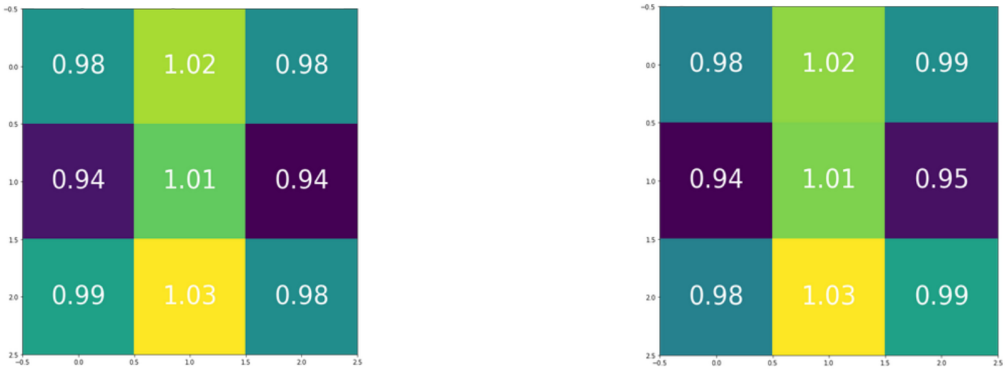
12.1 Modeling the Correlated Fluctuations for Photons

The first prototype to model the correlated fluctuations is based on photons with their shower energy depositions at the cell level. In order to model the deviations of events from the corresponding average, the idea is to train on energy ratios with respect to the average shower shape. It is implemented by running a given event through the simulation chain a large number of times and then build an average shape (or exact average) per calorimeter layer. Figure 170 illustrates a given Geant4 event in a cell energy grid of 7×7 in EMB2 with its exact average and the energy ratios derived from dividing the energies of the event by the energies of the exact average.

The training set is composed of Geant4 showers produced by photons with an energy of 65 GeV. The showers are uniformly distributed in $0.20 < |\eta| < 0.25$. The preprocessing is based on a rectangular selection of 7×7 cells in EMB2, as shown in Figure 170. The figure shows as well the grid of 3×3 cells, which represents the core region that contains most of the shower energy deposition. Therefore, the VAE is only trained on energy ratios of the 3×3 grid.

The architecture of the VAE model is represented in Figure 171. It is composed of LSTM layers for the encoder and the decoder. Using LSTM to learn correlated fluctuations gives the unique advantage of storing the important information for a longer amount of time, i.e., important patterns are persistent.

The choice of the architecture and all the VAE parameters, such as the size of the latent space, is based on an optimization coupled with a validation step. The first validation consists of reproducing the average shower representation across events. In fact, averaging multiple showers allows us to have a more visible topology with specific features such as the smoothness of the energy distribution. The exact average image in Figure 170 shows this pattern. Figure 172 shows the average Geant4 3×3 cell image over all events, compared to the average VAE-generated image. This shows that the VAE reproduces this pattern.



(a) (b)

Figure 172: 3×3 average ratio image from (a) Geant4 and (b) VAE.

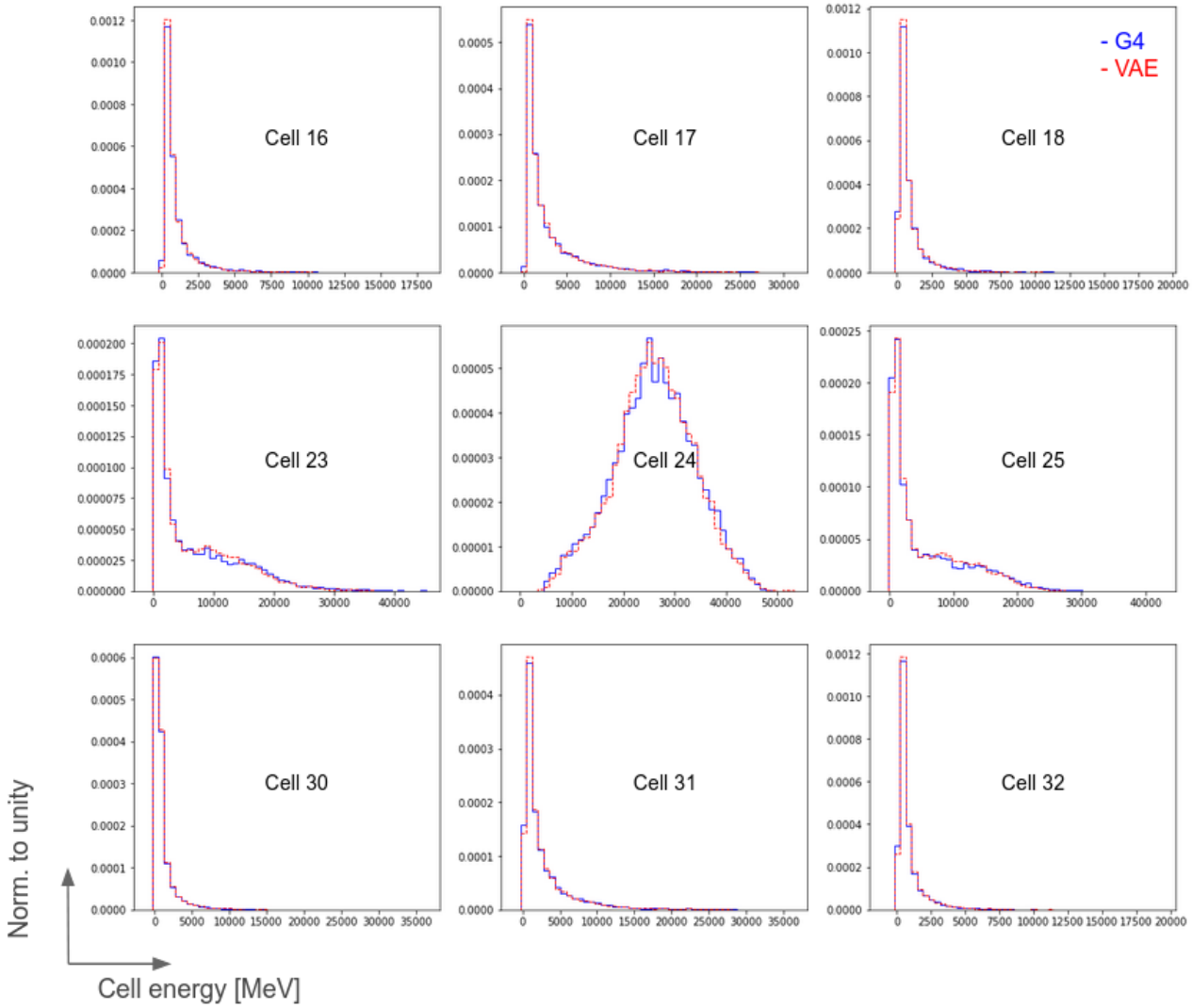


Figure 173: Cell energy distributions for the 3×3 cells grid in EMB2. The energy depositions from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line). The cell numbering here is done with respect to 7×7 grid in EMB2. .

To further assess the performance of the VAE, another validation criteria is to reproduce the shape of the cell energy ratios or alternatively the cell energies of the training inputs. Figure 173 shows the energy distributions for the 3×3 cells in EMB2. These energies are computed by multiplying the ratios by the exact average. All the distributions from the nine cells are well reproduced.

12.2 Modeling the Correlated Fluctuations for Pions

The photon prototype is a proof of concept on using a VAE model to learn the fluctuations. From Figure 169, the unique features that pion showers exhibit results in larger and non-trivial deviations from the average shape. Therefore, developing a dedicated tool to learn the fluctuations is more complex. The next section starts first with cell level information as training input in the process of moving from photons to pions. Then, finer granularity information is used to learn the fluctuations at the voxel level by extending the learning to more calorimeter layers, and extending the energy and η ranges.

12.2.1 CoVAE at Cell level

Pions are characterized by having wider showers compared to photons. Therefore, the selected window of cells in EMB2 should intuitively be larger than 3×3. The core region is set to be 5×5 which contains more than 98% of the total energy deposited in this layer. The VAE training is performed using central pions with a 65 GeV energy.

The uncorrelated-noise is included in the Geant4 events. In FCS, it is applied on top of the reconstructed cell energies to correct the energies. To illustrate the impact of this correction, Figure 174 compares the cell energy ratios from Geant4 to the average energy with added noise divided by the average. This shows that this correction can partially model some cells, especially in the edges, such as cell 0 or cell 24. On the other hand, in the core region, where most of the energy is deposited (cells: 12,11,13,6,7,8,16,18) it can not reproduce the distributions due to the missing correlation information. In fact, the noise component can be used during training and validation. The idea behind using the noise is to simplify the learning process of the network to only focus on modeling the correlations between the cells. Chapter 9 presented the augmented reconstruction loss with the noise.

Another correction applied by FCS after reconstruction consists of renormalizing the energy per event. This normalization allows an exact match of the simulated energy. It consists of adding an energy quantity q to match the Geant4 energy such that $q\sum_{i=1}^{25} E_i(G4_{ea}) = \sum_{i=1}^{25} E_i(G4_{ce})$, where, $E_i(G4_{ea})$ represents the cell energies using the exact average and $E_i(G4_{ea})$ the cell energies from the Geant4 events.

In addition to the distribution of cell energies or ratios, the modeling of the correlated fluctuations can be also validated through correlation coefficients between the core cell (middle) and other cells. Geant4 contains a specific pattern of this statistical measure, constituting a set of negative correlations. An example to illustrate the negative correlation can be seen in Figure 175 which shows the energy values in the 5×5 grid in EMB2 for two pion events with an energy of 65 GeV. A negative correlation in this case comes from how the showers develop in the calorimeter: when most of the energy is contained in the core region the outer regions contain less energy.

Moving from a 3×3 grid for photons to a 5×5 grid for pions as mentioned above, is motivated by the energy deposition. Figure 176 summarizes the correlation coefficients for VAE compared to Geant4 for three different sizes of the grid. In each of the plots, a VAE model is trained to reconstruct the cell ratios with a 2D latent space. Only the input and output layers are different between the four models. These plots show a very promising result towards increasing the size of the grid, where the correlations are well reproduced for the 7×7 grid. However, as shown in Figure 174 cell fluctuations in the outer region can be modeled using the uncorrelated-noise. Therefore, we consider only the 5×5 grid. The same model, as described for the photons in Figure 171, is used to train on the cell energy ratios for pions with an adapted number of nodes in the input and output layers to match the new grid size. A Similar level of agreement is seen for the pion model in learning the distribution of each cell energy ratios. These distributions are plotted in Figure 177.

Following the same process of validation in the previous chapters, after a standalone validation, the shape validation here allows us to access the performance of CoVAE on top of FCS. For this propose, the decoder's architecture and weights are saved and converted into a LWTNN JSON file of size 144.1 KB, which is afterwards used for the shape validation.

Figure 178 shows one of the most important shower observables, the shower width *weta*. Adding the VAE correlated fluctuations on top of FCS improves the agreement with Geant4.

12.2.2 CoVAE at Voxel level

Similar to Chapter 9, the voxel level consists of binning the Geant4 hits using polar coordinates with 8 bins in α, where $\alpha \in [0, 2\pi]$ and 9 bins in r with the first r bin is 5 mm and the others are of order of 20 mm. The energy values of the voxels are saved in 3D histograms (r, α, e), where the third dimension e represents the energy. The exact average in the voxel level case is also stored as a histogram. A first trial consists of using four truth energy values: 8, 16, 32, 65 in $0 < |\eta| < 0.6$. Moreover, for testing the robustness of the model another η

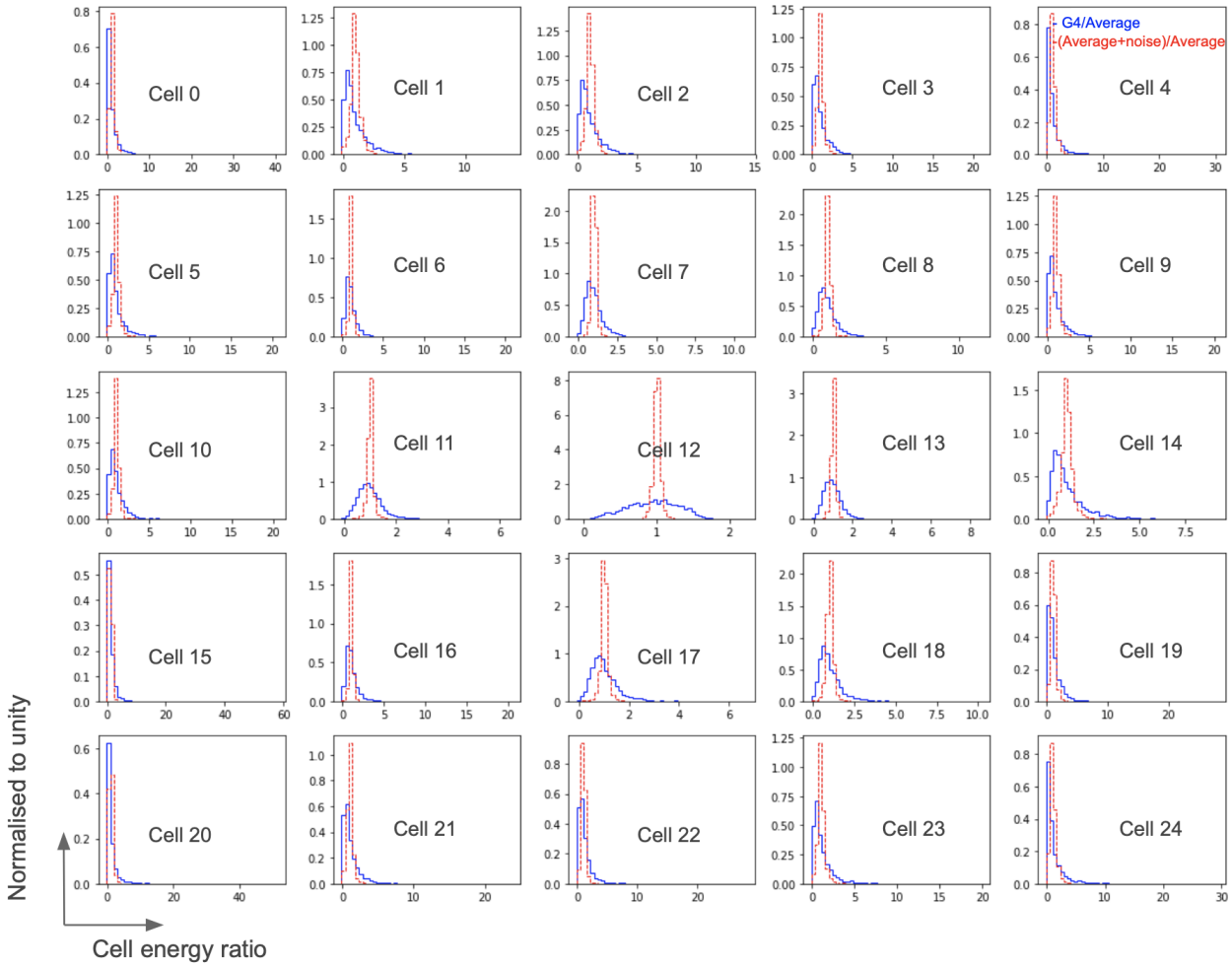


Figure 174: Cell ratio energy distributions for the 5×5 cells grid in EMB2. The ratios from a full detector simulation (blue line) are shown as reference and compared to the ratio of the (average shape + noise) and the average with a normalization factor (red line). The cell numbering here is done with respect to 5×5 grid in EMB2.

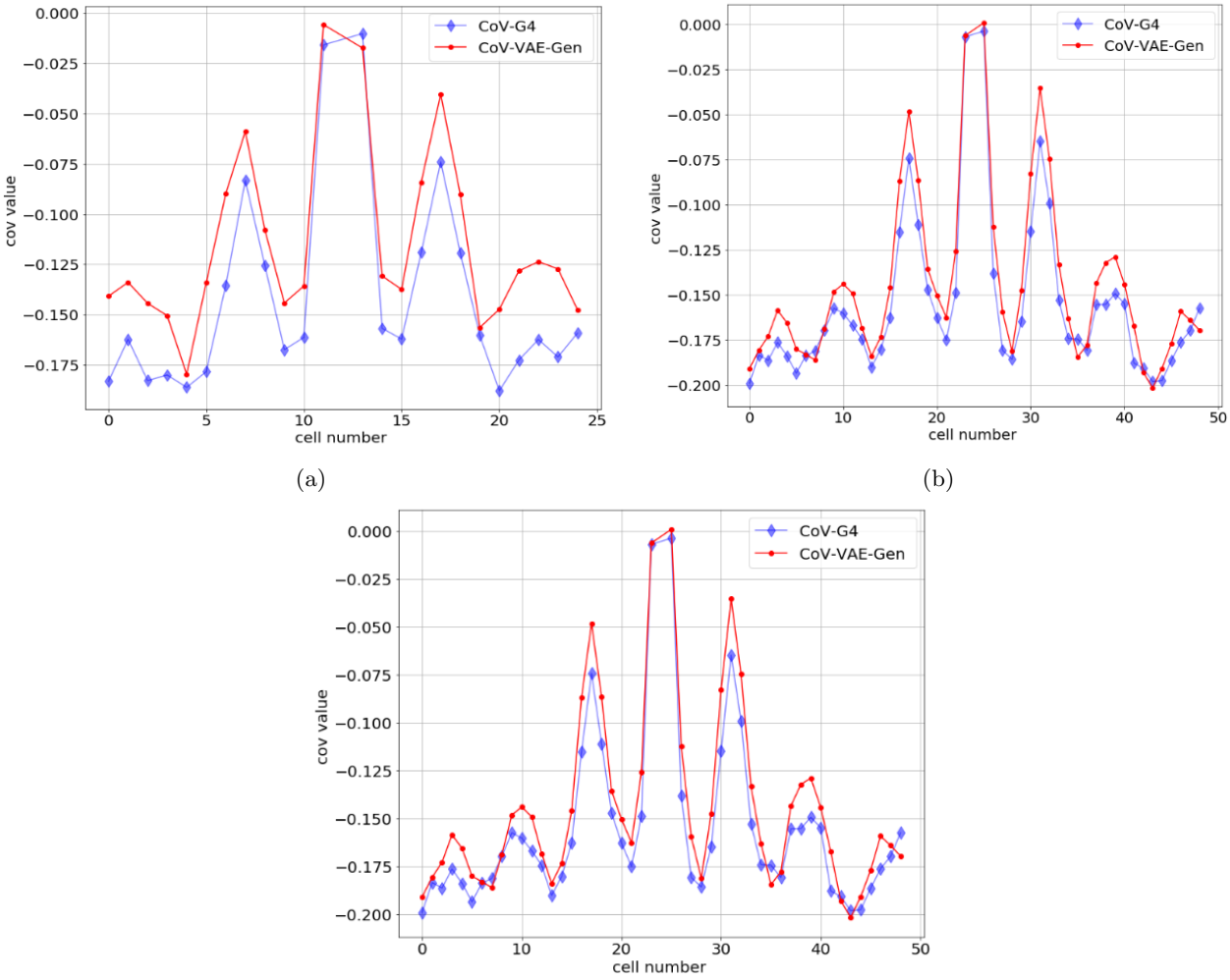
230.297	148.296	132.279	106.19	52.9492
31.8689	106.797	463.599	216.393	28.9776
34.9465	498.376	8150.41	917.835	51.816
109.862	377.44	850.734	383.105	124.722
182.367	70.1307	71.9822	124.049	100.837

(a)

49.9634	100.837	263.175	326.382	478.298
106.19	1215.33	1596.68	855.588	275.553
793.906	2733.14	5516.28	713.5	177.21
355.338	687.957	494.991	133.685	191.182
145.293	317.51	447.624	118.757	18.9655

(b)

Figure 175: Energy values in EMB2 for pions eta 0.2 65 GeV (a) (b) .



(a)

(b)

(c)

Figure 176: Correlation coefficients for Geant4 versus VAE-generated for four different grid sizes in EMB2. (a) 3×3, (b) 5×5, (c) 7×7. The full simulation (blue line) is compared to the VAE (red line). The periodic structure represents the order of cell reading when moving from one row to the next one.

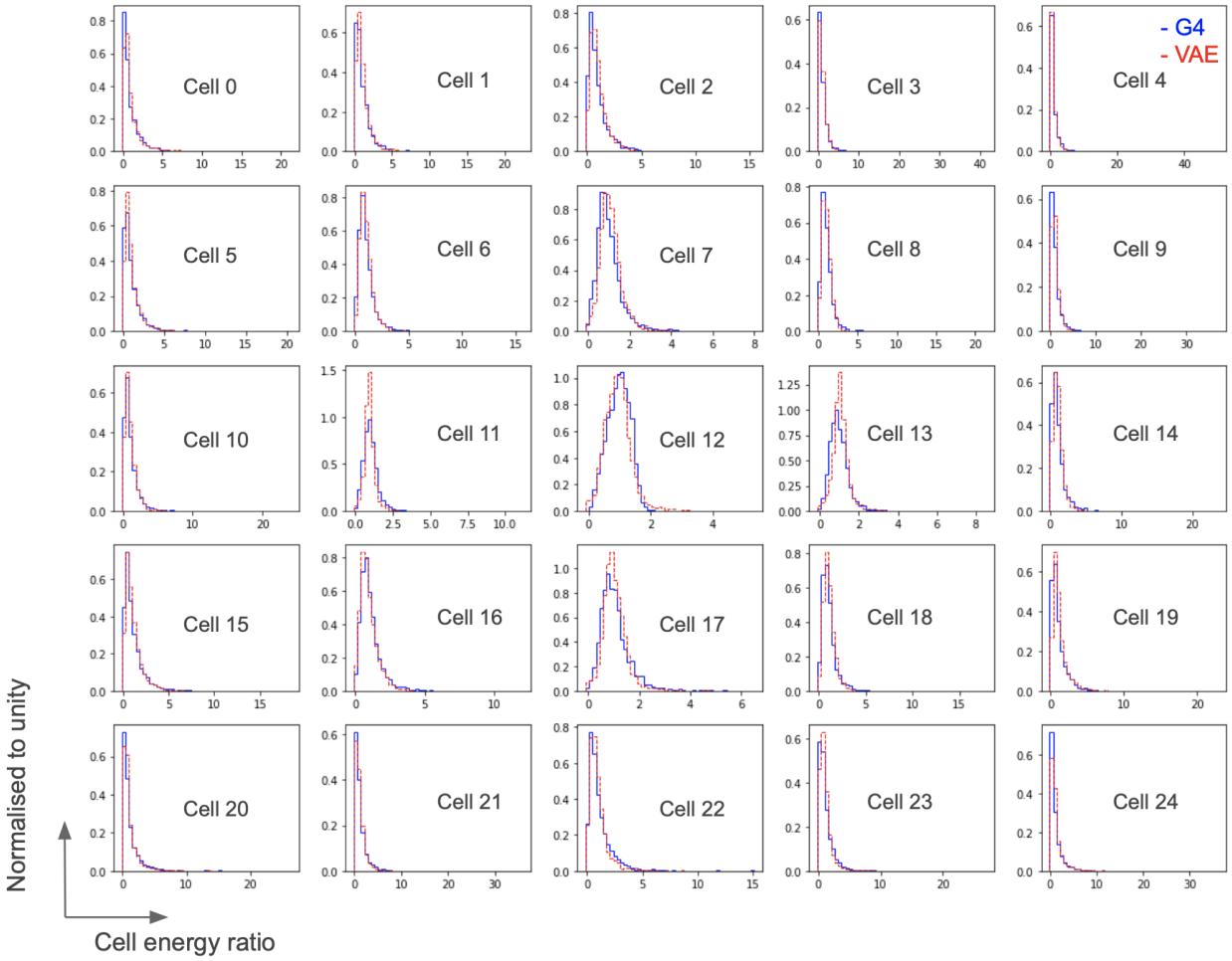
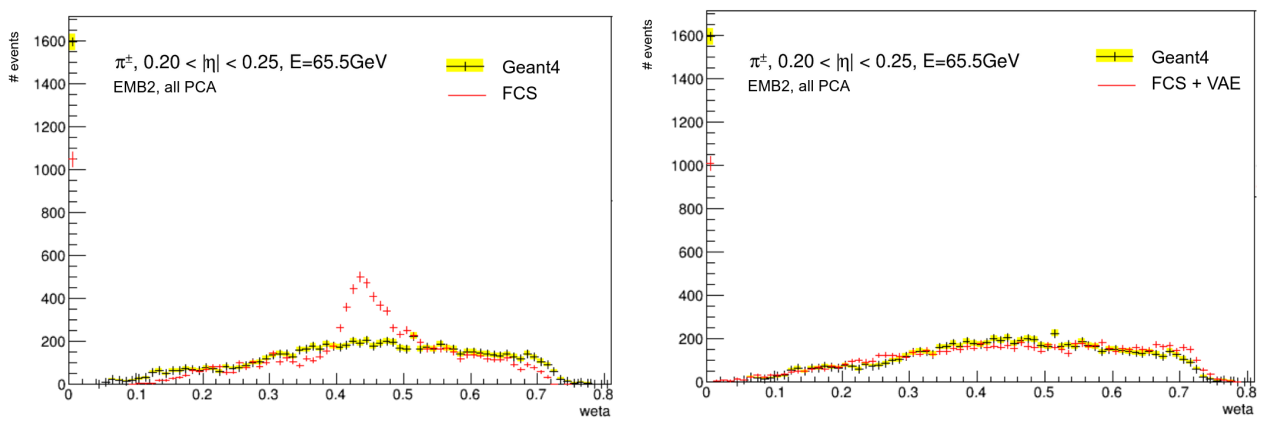
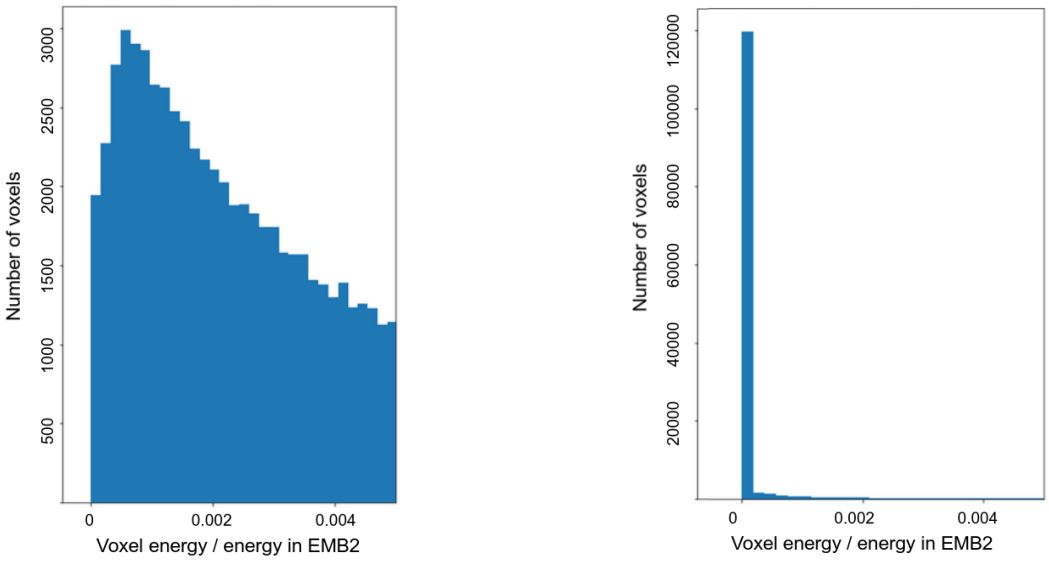


Figure 177: Cell energy ratio distributions for the 5×5 cells grid in EMB2. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line) after adding the noise and the renormalization to the VAE output.



(a) (b)

Figure 178: weta distribution in EMB2 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The full simulation (black line) is compared to FCS (red line in a) and FCS+ VAE-correlated fluctuations (red line in b)



(a) (b)

Figure 179: Voxel energies for the first (a) and the last (b) PCA bins in EMB2.

range is considered where $2.2 < |\eta| < 2.3$ and the energy of the truth particles of 260 GeV. The VAE model is therefore conditioned on both the energy and η.

Following the same logic as the previous section, the uncorrelated-noise (computed with respect to the energy of the voxel in GeV units) is fed to the network when computing the reconstruction loss. The VAE performance is systematically compared to Geant4 and the Gaussian model investigated by another ATLAS group. This comparison is the result of a research collaboration summarized in Reference [212].

The Gaussian model is based on finding a representation of the input data for generating new events. It applies a mapping function f on the inputs to Gaussianize them. This operation results in N dimensional Gaussians with N means and an $N \times N$ covariance matrix. To generate events, a sampling from the cumulative distributions is performed.

Section 5.3.2 described how the longitudinal paramaterization of FCS derives PCA bins of de-correlated energies. The number of PCA bins is five. Each of the bins contains about 20% of the total number of events. Events in the same PCA bin share the same properties of energy deposition. One of these properties can be seen in Figure 179 which compares the voxel energy ratios to the total energy between the first and the last PCA bins. The first bin contains showers which deposited a considerable amount of energy in EMB2 and the last bin contains showers with very low energy deposition in EMB2.

For the training procedure, we use the information of the PCA bins. In order to select the best training approach, we tested conditional and unconditional approaches. The condition here refers to the PCA bin information. The conditioning allows us to have a single model in order to reduce the memory footprint and the optimization process. Since the PCA bins are categorical information, the condition can be defined as a one hot vector where the condition vector of events belonging to the first PCA bin is [1,0,0,0,0] and for the last PCA bin the vector is [0,0,0,0,1]. Figure 180 shows the 8×9 voxel ratios for 65 GeV pions in EMB2 for all PCA bins. The shape of the ratios is not well reproduced. This can be explained by the fact that the different PCA bins represent different features as shown in Figure 179 and different categories of events where PCA1 can be seen as the bin of showers starting their development in EMB2 and PCA5 as the bin of events where their secondaries start after EMB2 (late showers). To further understand the results of Figure 180, the correlation plots in Figure 181 shows that the mismodeling of the energy ratios is caused mainly by a mismodeling of the correlations in the last PCA bins (4 and 5). Unlike at the cell level where the correlations are computed between the core cell and the other cells, there is no clear reference to a center voxel since all the 8 α bins in the first ring represent the core of the shower. Following the same standard in FCS shape, the energy ratios from the first r ring are averaged and set to be the core voxel. The index of each of the other voxels is shown in Figure 182.

On the other hand, when testing training and generation on each PCA bin independently, the performance is enhanced. Figures 183, 184, 185, 186 and 187 show the 8×9 voxel ratios for each of the PCA bins.

The last two PCA bins are dominated by zero peaks, which are a direct consequence of low energy depositions in these bins. This means that the maximum number of bins to consider should be smaller to only contain the

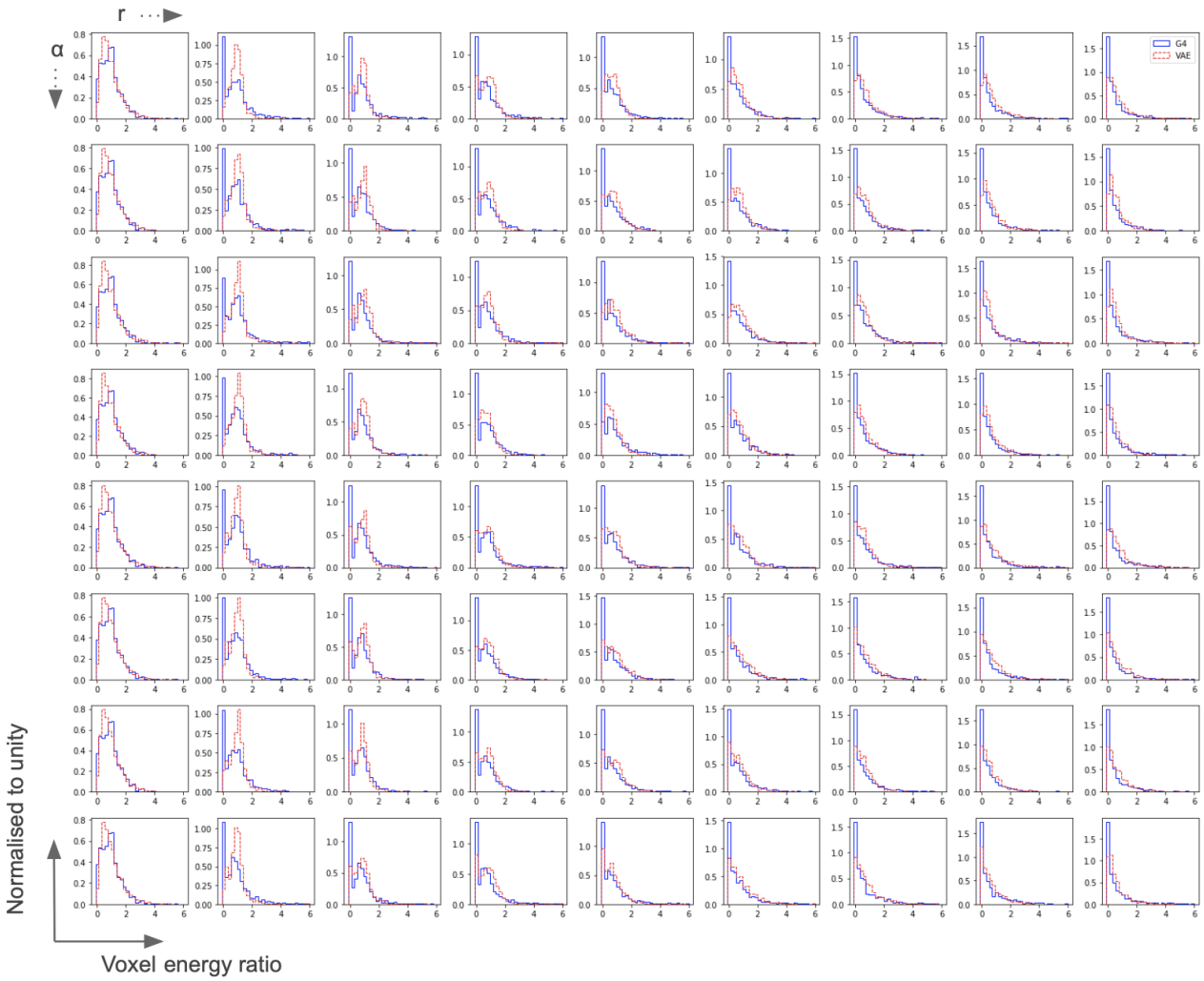
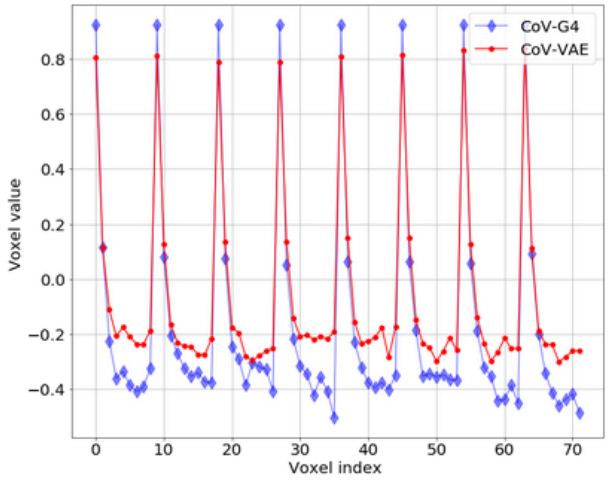
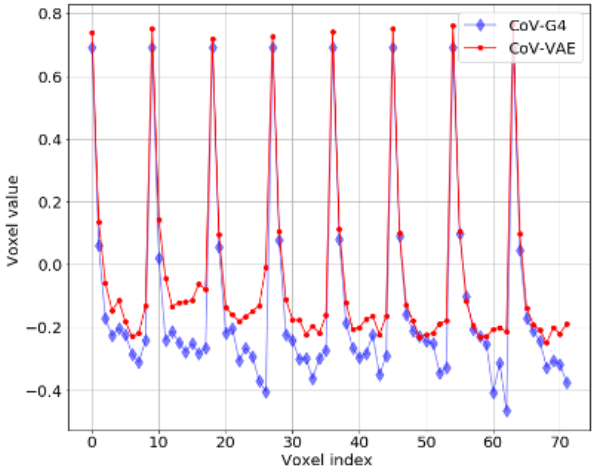


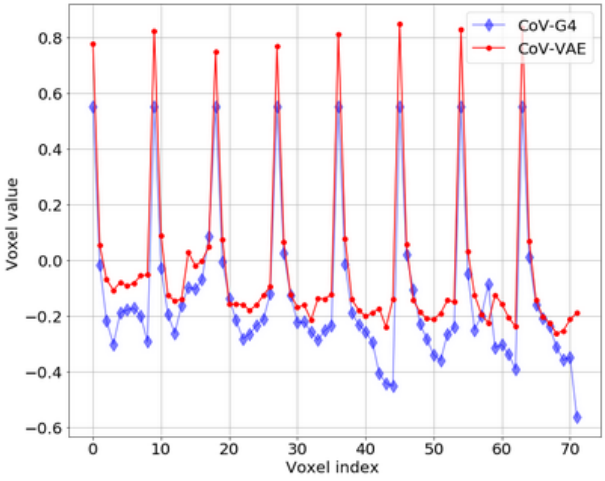
Figure 180: (r, α) voxel energy ratio distributions in EMB2 in PCA1 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).



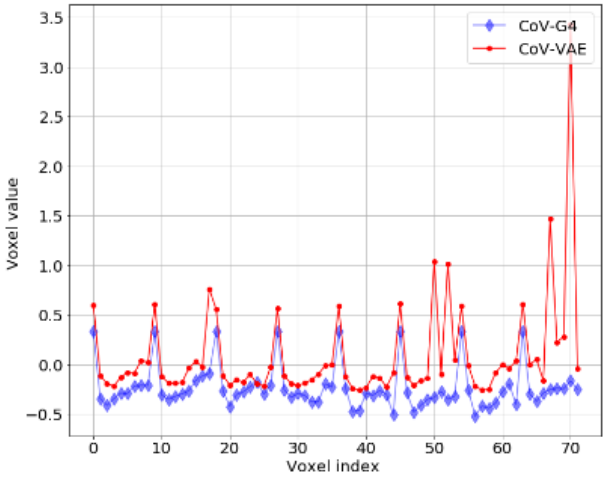
(a) PCA 1



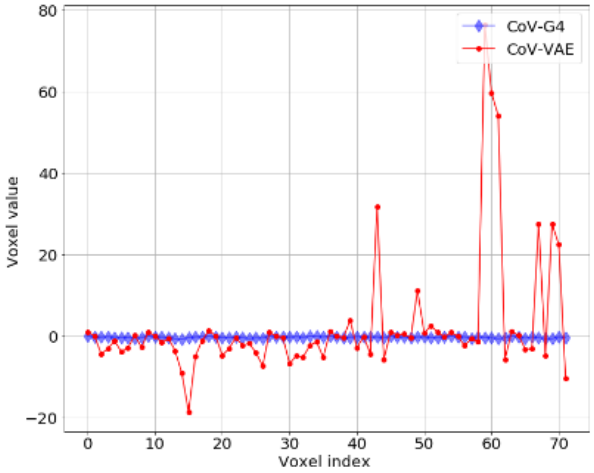
(b) PCA 2



(c) PCA 3



(d) PCA 4



(e) PCA 5

Figure 181: Correlation coefficients for Geant4 versus VAE-generated for (a) PCA 1, (b) PCA 2, (c) PCA 3, (d) PCA4 and (e) PCA 5. The full simulation (blue line) is compared to the VAE (red line). The periodic structure represents the order of voxels when moving from one α bin to the next one. The indices are shown in Figure 182.

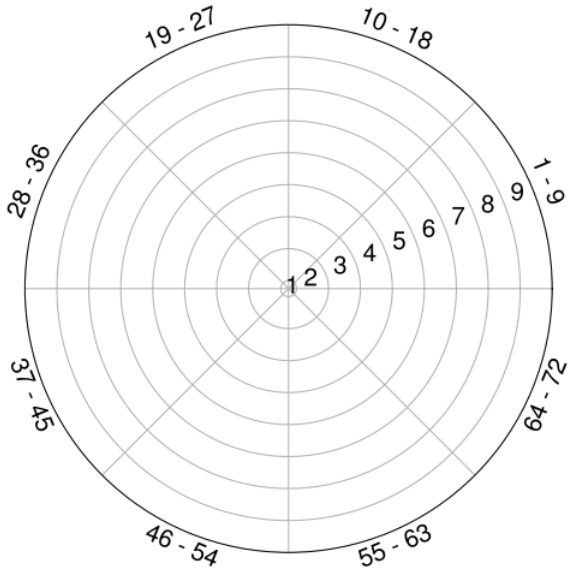


Figure 182: Voxel index.

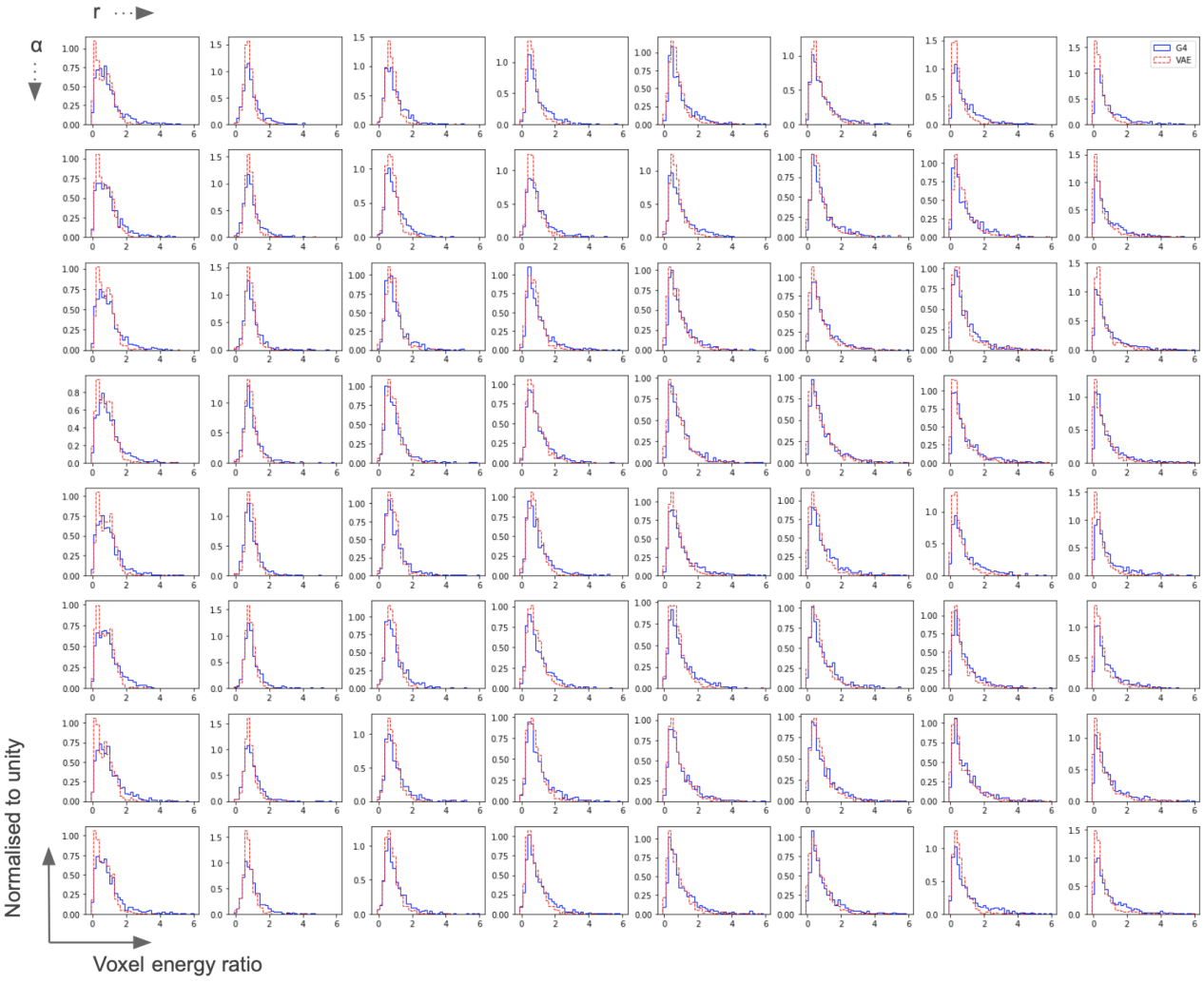


Figure 183: (r, α) voxel energy ratio distributions in EMB2 in PCA 1 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

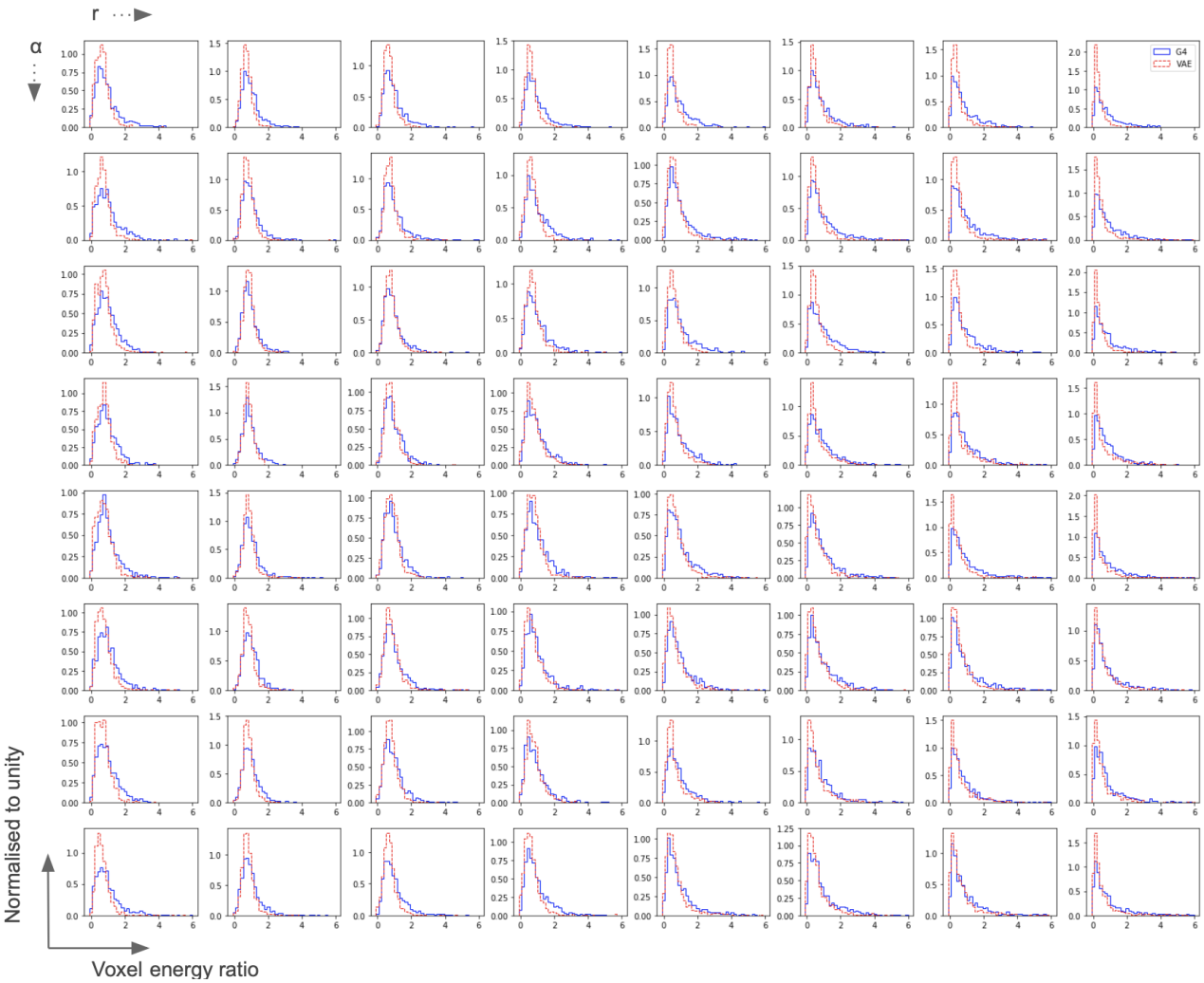


Figure 184: (r, α) voxel energy ratio distributions in EMB2 in PCA 2 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

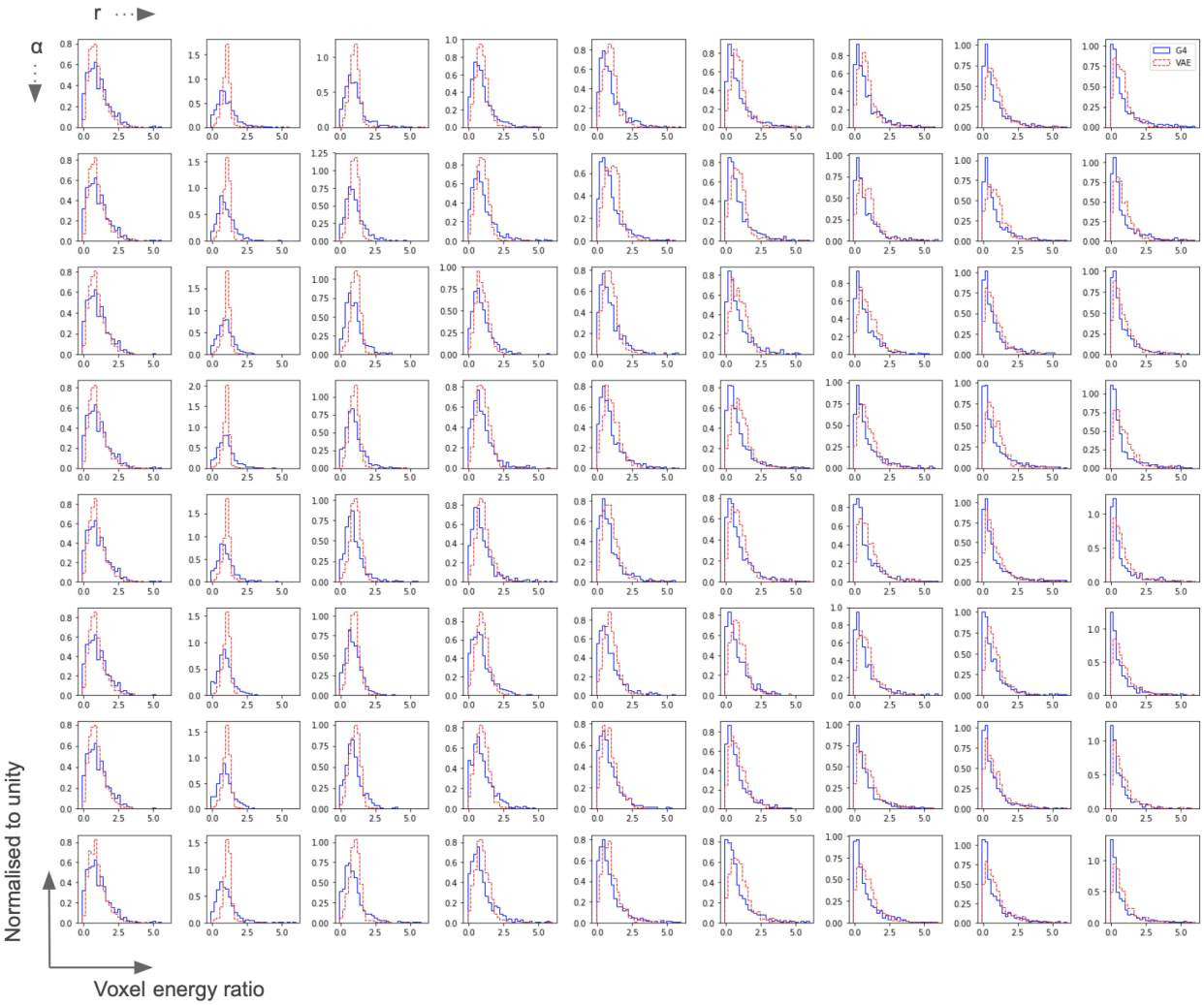


Figure 185: (r, α) voxel energy ratio distributions in EMB2 in PCA 3 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

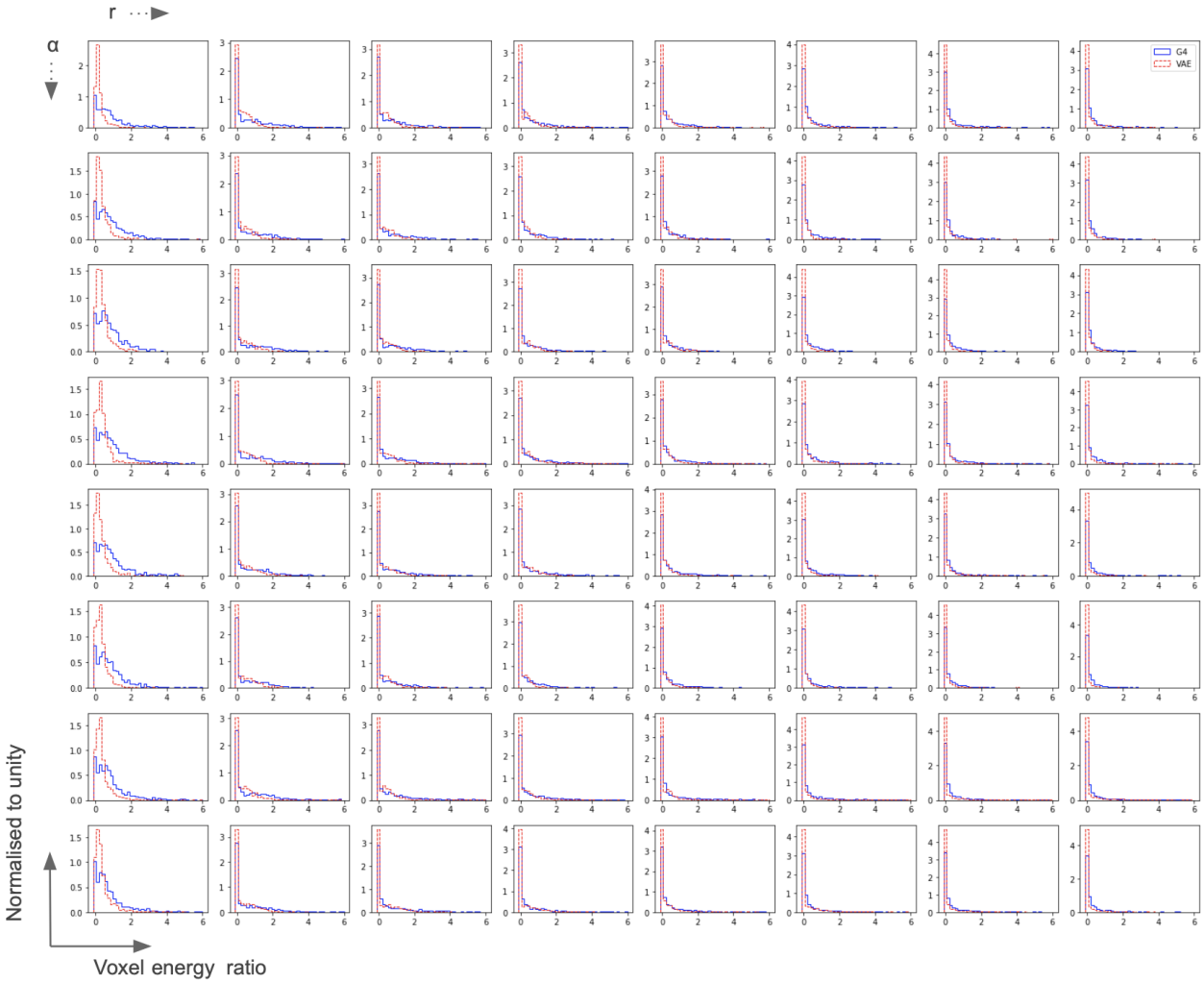


Figure 186: (r, α) voxel energy ratio distributions in EMB2 in PCA 4 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

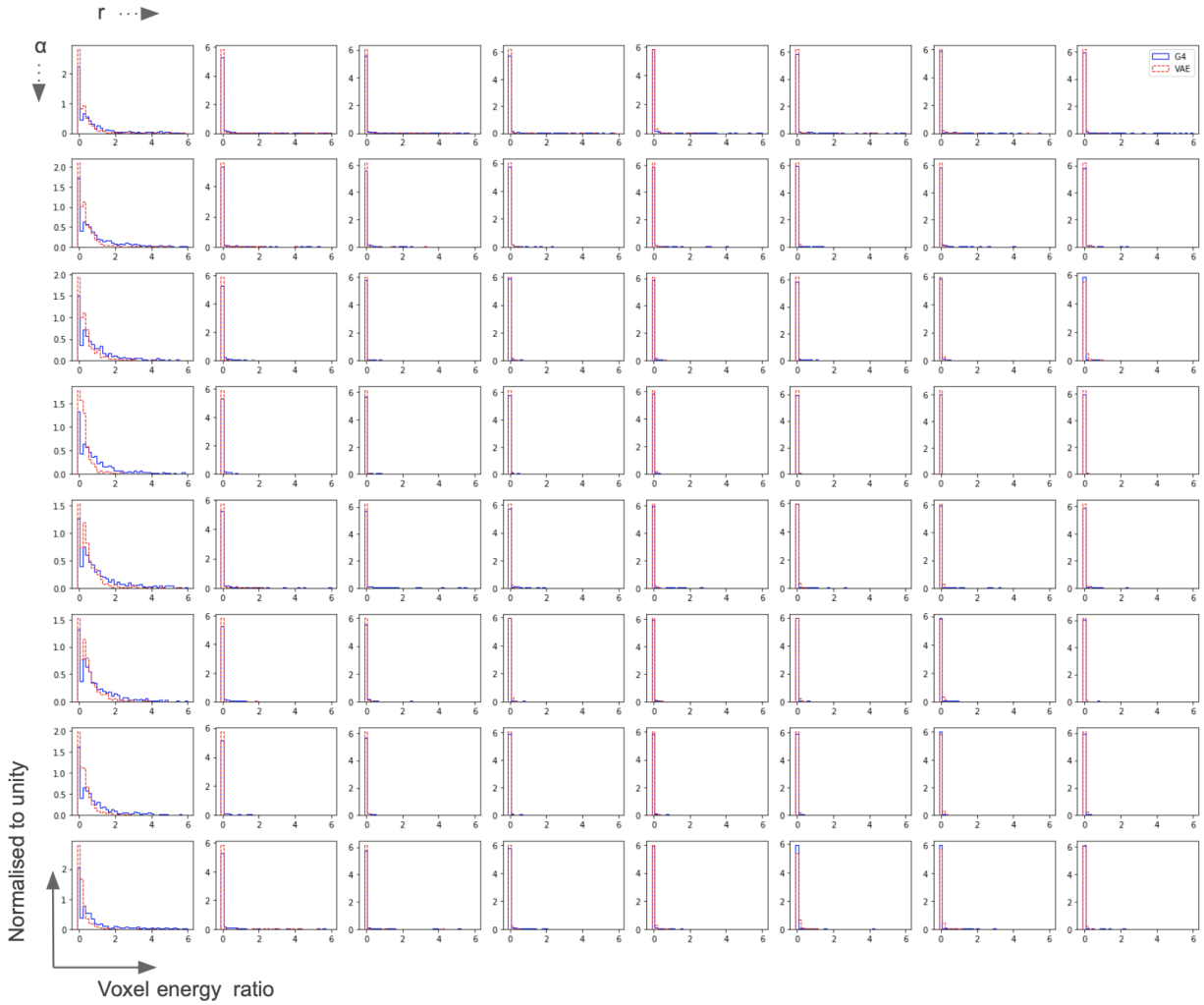


Figure 187: (r, α) voxel energy ratio distributions in EMB2 in PCA 5 for pions with an energy of 65 GeV in the range $0.20 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

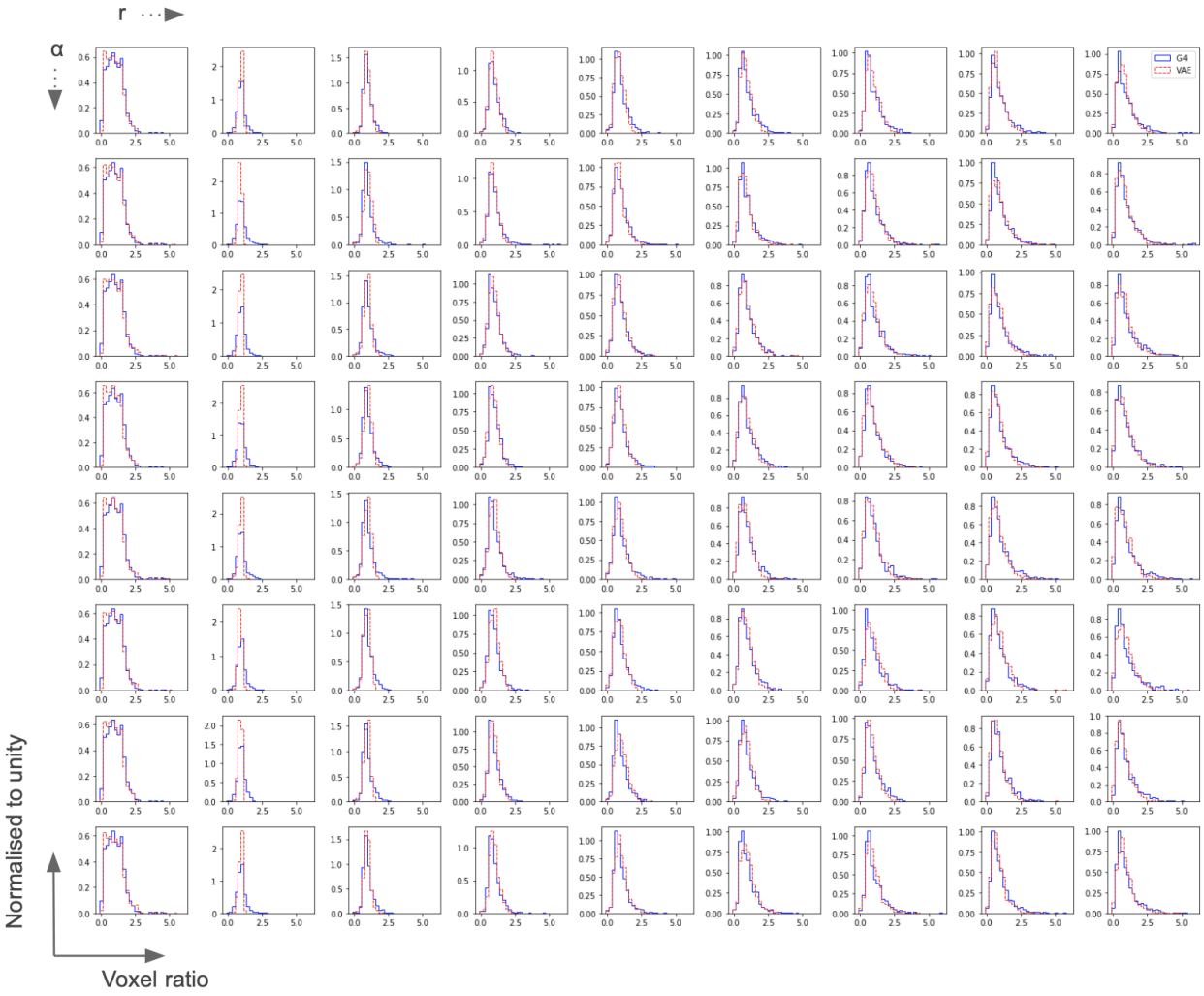


Figure 188: (r, α) voxel energy ratio distributions in EMB2 in PCA 5 for pions with an energy of 262 GeV in the range $2.2 < |\eta| < 2.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

rings with most of the energy deposition.

Since the PCA is performed for each η slice independently, this makes the definition of the PCA bins inconsistent across η. In $2.2 < |\eta| < 2.25$, PCA5 is now the one with most of the energy deposition for EMEC2 layer. Figure 188 shows the 8×9 voxel ratios for PCA5 in EMEC2 layer. The plot shows the model's capacity to learn the fluctuations in different η region with a higher energy.

The plots in Figures 189 to 192 show the results from using the VAE along with the Gaussian method. These represent two techniques to model the fluctuations on top of FCS. The plots compare the RMS fluctuations about the average shape without the correlated fluctuations and with the correlated fluctuations using both techniques. Figure 189 shows the performance for 65 GeV pions in EMB2 in $0.05< |\eta| <0.10$. The agreement of the RMS error bars with Geant4 is well modelled with both techniques. The correlated fluctuations are only modeled up to a fixed distance from the shower center by using an 8×9 bins in (α, r). This is reflected in this plot, where above a distance of 0.1 the agreement does not match. Figures 189 and 190 show the RMS fluctuations about the average shape for two extreme η slices in the considered range of training. Figure 191 shows the performance on another energy range of 16 GeV in $0.20 < |\eta| < 0.25$. Figure 192 shows the same quantity for pions with an energy of 262 GeV in EMEC2 in $2.20 < |\eta| < 2.25$.

The plots described above are the results of a training on voxelized hits in polar (α, r) bins with a 5 mm core in r and 20 mm binning elsewhere. In fact, the 5 mm core is filled with the average value of voxels in this bin across all the 8 α bins when creating the parametrization. On the simulation side, each of the 8 α bins is treated separately. This means that the model is learning redundant information. This leads to a biasing in the ratio to the average shape as shown in Figure 191 where with the VAE correlated fluctuations a slight shift from the ratio value to the average shape (1) is seen. To overcome this, we can instead train on a single voxel

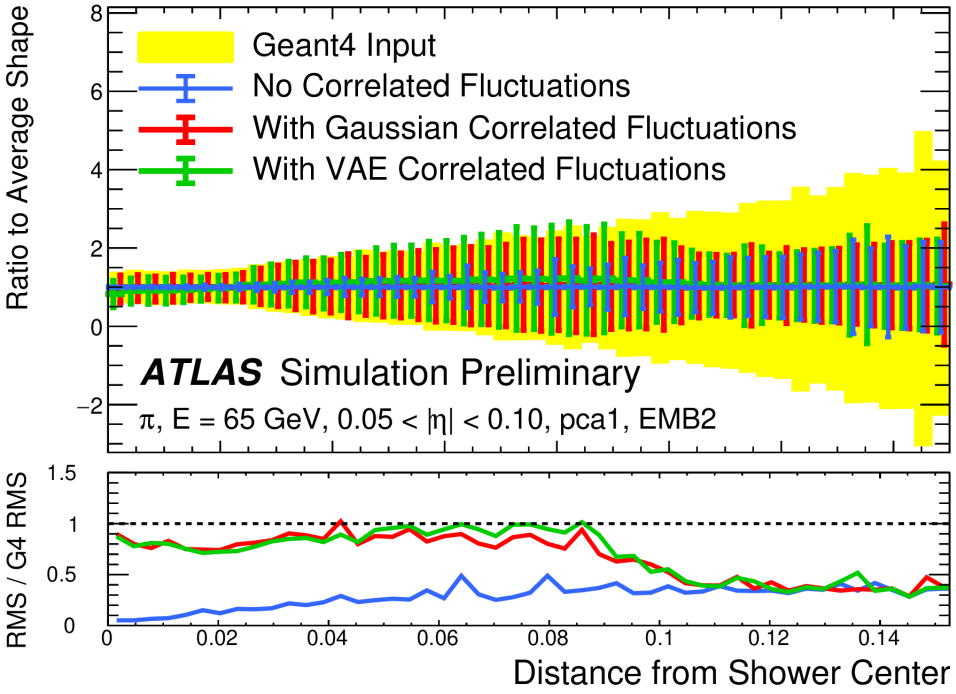


Figure 189: RMS fluctuations about the average shape with the Gaussian fluctuation model (red) the VAE fluctuation model (green) and without correlated fluctuations (blue) for 65 GeV pions in EMB2, $0.05 < |\eta| < 0.10$.

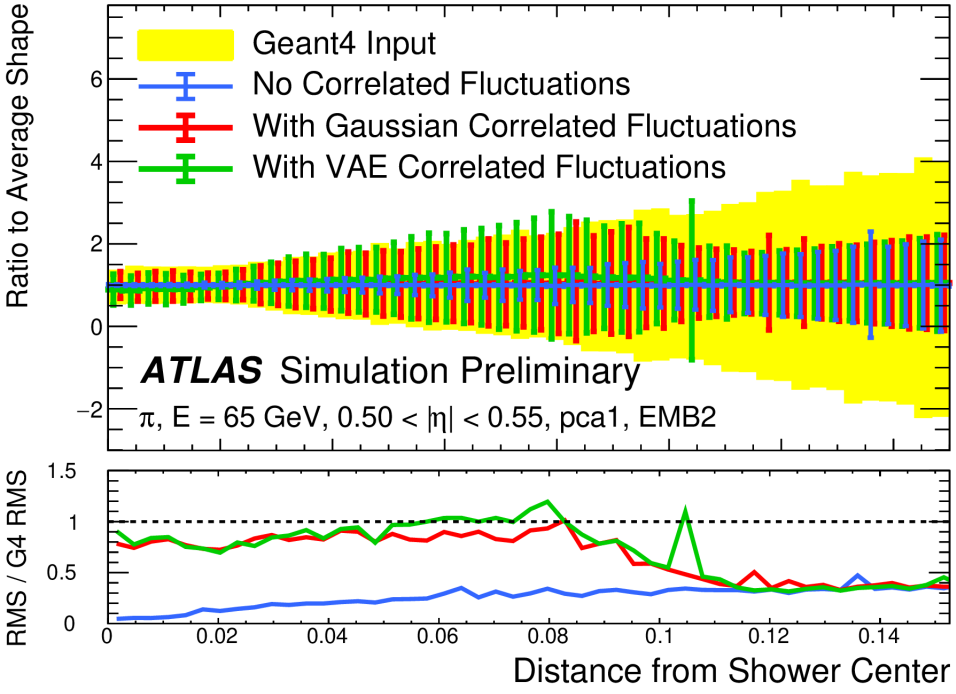


Figure 190: RMS fluctuations about the average shape with the Gaussian fluctuation model (red) the VAE fluctuation model (green) and without correlated fluctuations (blue) for 65 GeV pions in EMB2, $0.5 < |\eta| < 0.55$.

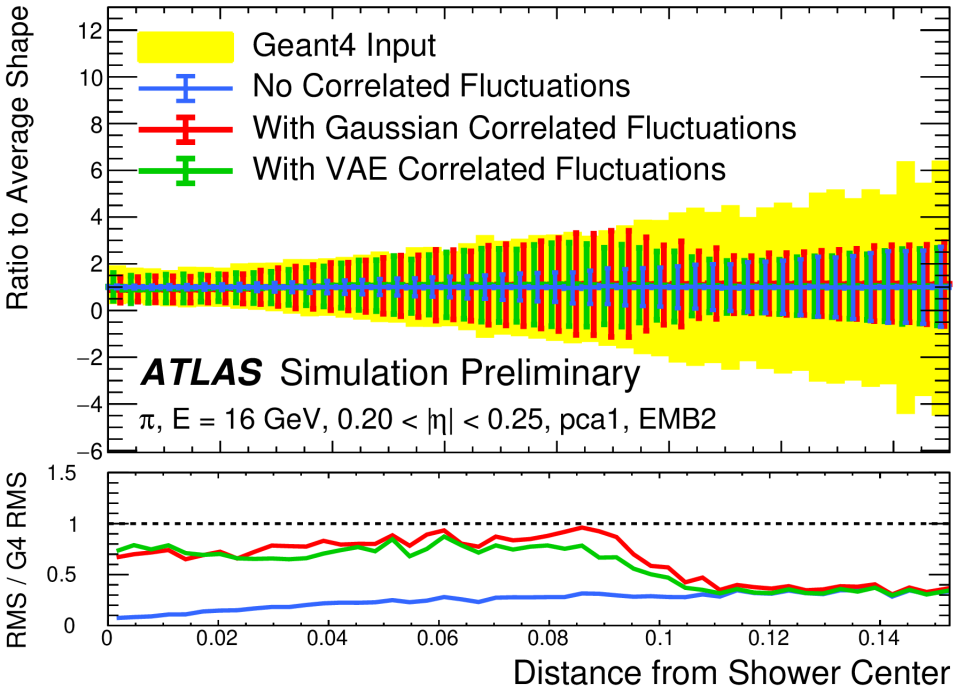


Figure 191: RMS fluctuations about the average shape with the Gaussian fluctuation model (red) the VAE fluctuation model (green) and without correlated fluctuations (blue) for 16 GeV pions in EMB2, $0.2 < |\eta| < 0.25$.

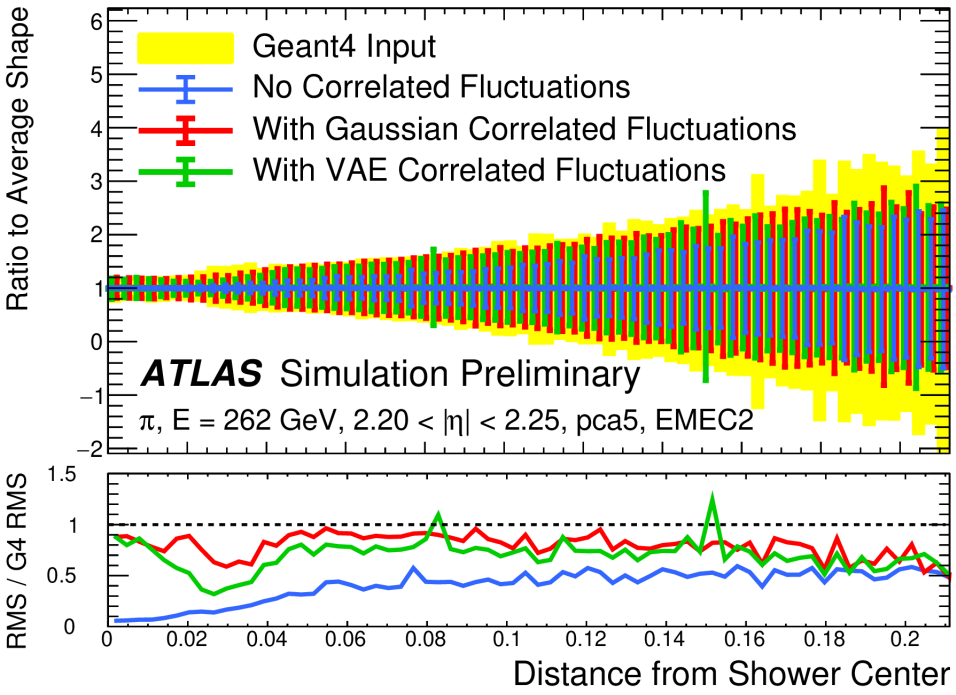


Figure 192: RMS fluctuations about the average shape with the Gaussian fluctuation model (red) the VAE fluctuation model (green) and without correlated fluctuations (blue) for 262 GeV pions in EMB2, $2.2 < |\eta| < 2.25$.

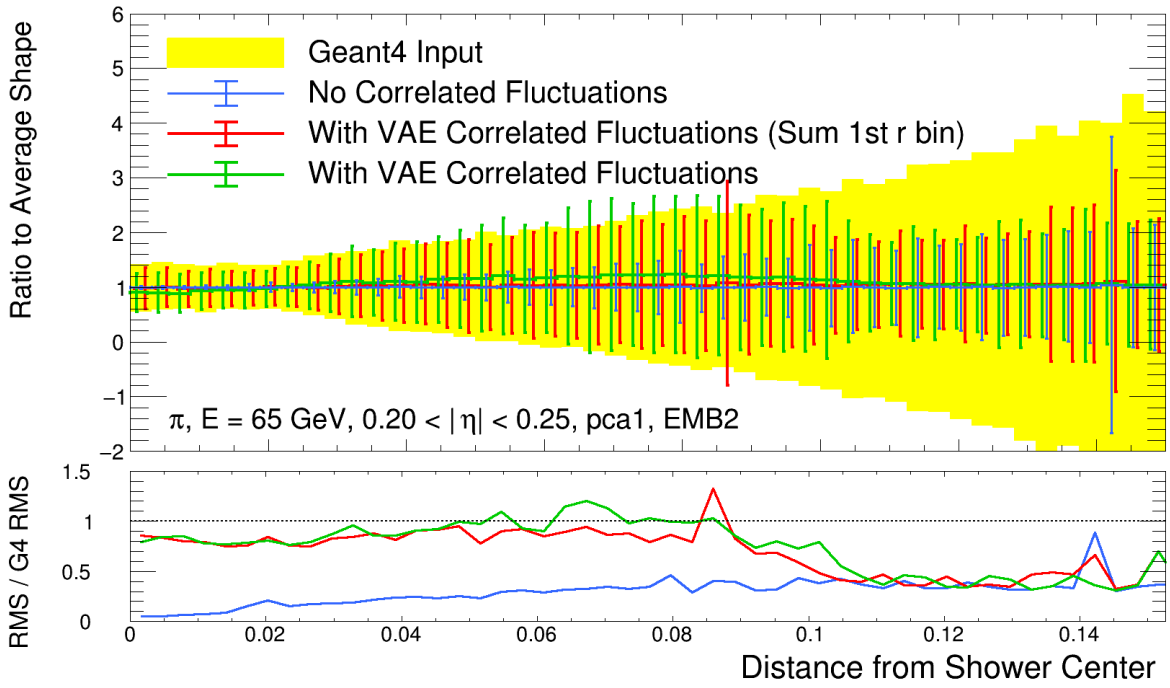


Figure 193: RMS fluctuations about the average shape with the VAE by summing the first r ring (red) of the VAE fluctuation model without summing the first r ring (green) and without correlated fluctuations (blue) for 65 GeV pions in EMB2, $0.2 < |\eta| < 0.25$.

in the core r bin and the remaining 8×8 bins in (α, r). This reproduces the correct ratio value to the average shape as shown in Figure 193 (in red) as compared to the previous training (green).

Towards extending energy and η ranges

The VAE model in the previous section is trained per PCA bin. In order to further extend the energy and η ranges, a PCA consistency study is done to derive regions where the PCA definition is similar. For this purpose, quantities such as the energy per layer can be used. The energy per layer shows the largest variations across the PCA bins compared to the total energy.

Upon defining the regions of η with a consistent PCA definition, a conditional VAE model is trained on voxel energy ratios with energies ranging from 8 GeV to 4 TeV. Energies below 8 GeV are not taken into account because the ratios are dominated by zero values.

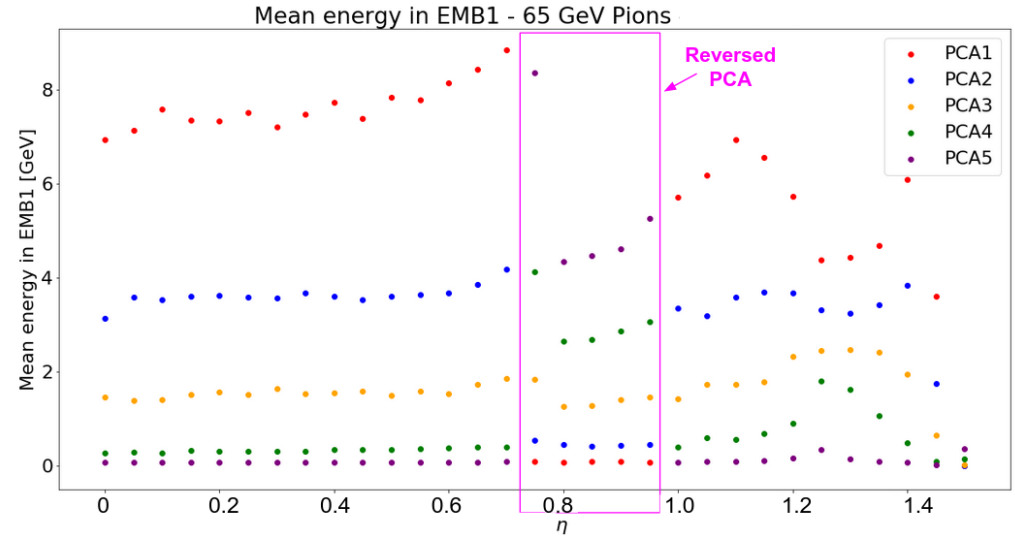
Figure 194 shows the mean energy in EMB1 as function of η for 65 GeV and 2 TeV pions for all the PCA bins. The plot shows that the PCA bin definition using the information of the mean energy in EMB1 is consistent where the PCA ordering is the same. A clear distinction of two PCA_η regions is visible: up to $\eta \leq 0.7$ and starting from $\eta \geq 1$ the PCA bins are consistent, in between the PCA are reversed.

Figures 195, 196 and 197 show the 8×9 voxel ratios distributions for a low, medium and high energy values of 4 GeV, 65 GeV and 4 TeV pions respectively with three different η values. Overall, the model learns the all the distribution shapes.

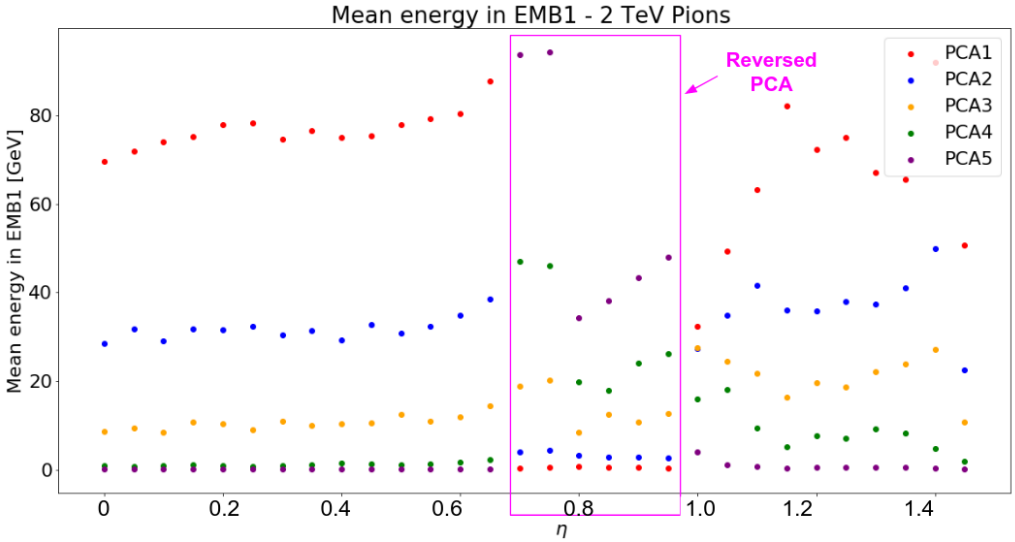
The Athena validation allows us to access the performance of the VAE on top of FCS. With the VAE correlated fluctuations, the energy of clusters in the ATLAS topological cluster reconstruction is better modeled. This can be seen in Figure 198. Figure 199 shows the distance of the cluster to the true pion. The agreement to Geant4 improves when the VAE correlated fluctuations are included. The agreement is also better modeled for the second moment in r and lambda, shown in Figure 200 and 201 respectively.

12.3 Summary and Discussion

FCS relies on longitudinal and lateral parametrization of the particle shower development to realistically model the ATLAS detector response. The longitudinal parametrization describes the energy deposited in each calorimeter layer. Since the energies are highly correlated, FCS uses Principal Component Analysis (PCA) to de-correlate the energies to simplify the parametrization. While FCS models very well the longitudinal quantities of electro-



(a)



(b)

Figure 194: Mean energy in EMB1 as function of the truth η for pions with an energy of (a) 65 GeV (b) 2 TeV.

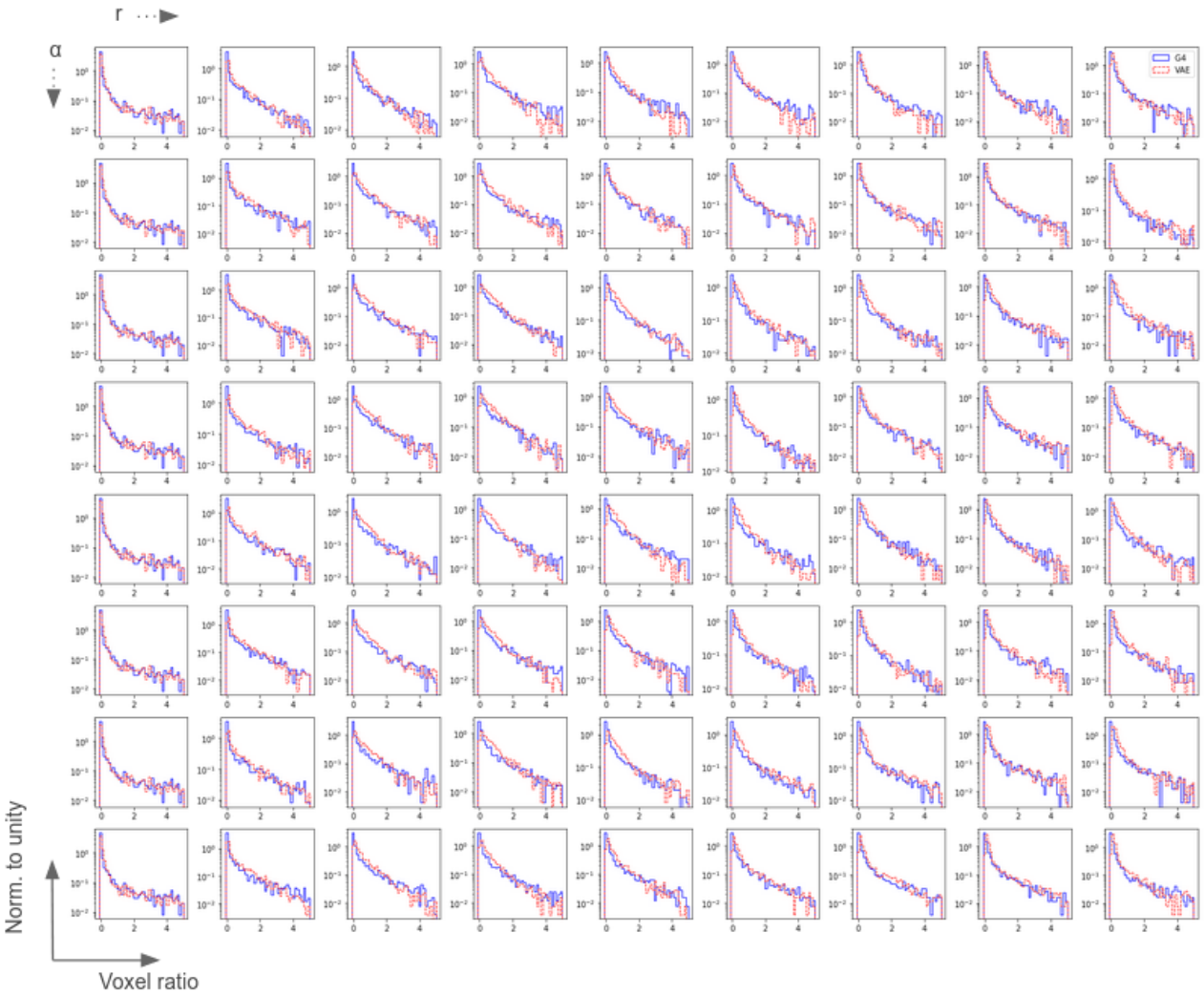


Figure 195: (r, α) voxel energy ratio distributions in EMB2 in PCA 1 for pions with an energy of 8 GeV in the range $0.4 < |\eta| < 0.45$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

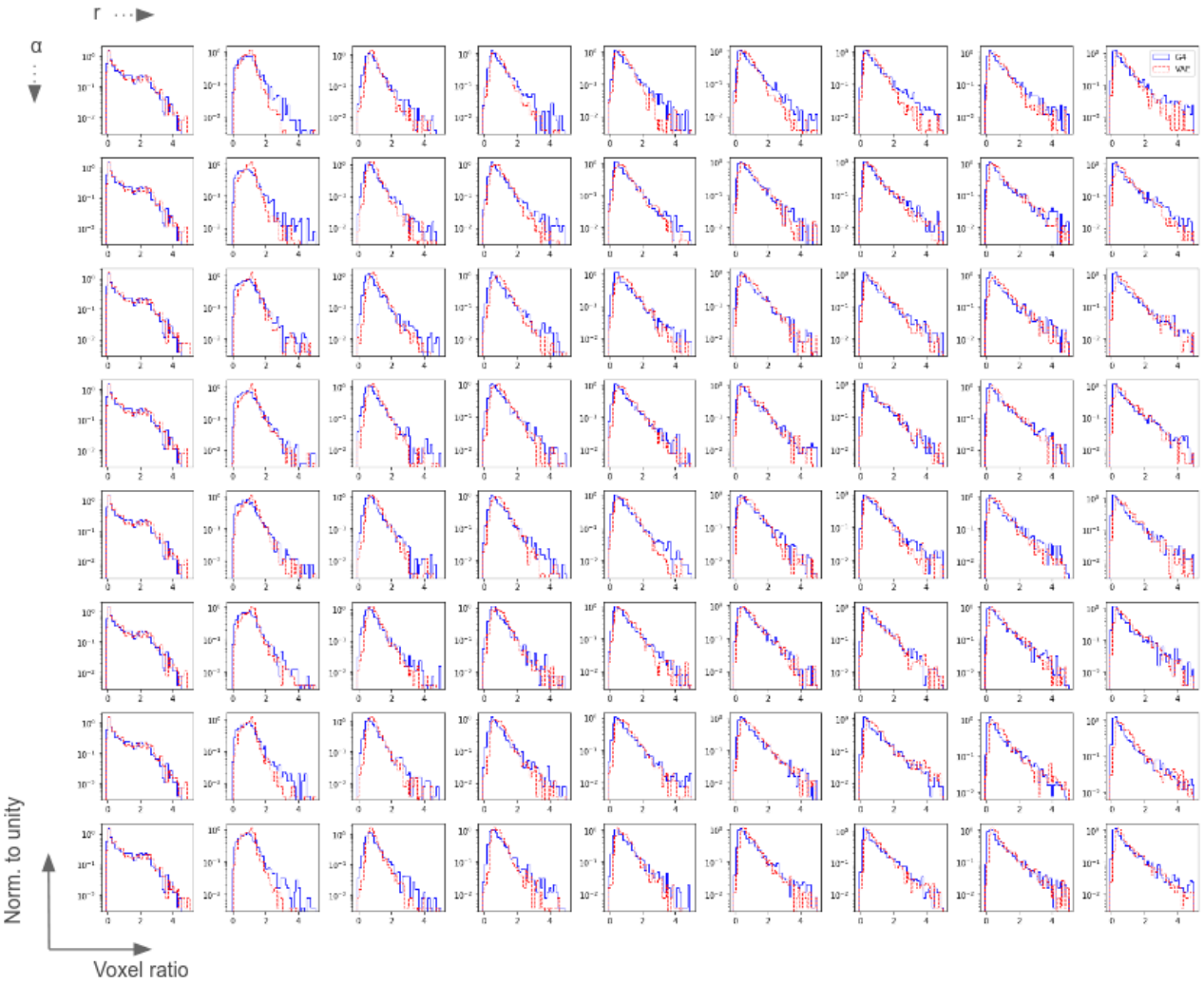


Figure 196: (r, α) voxel energy ratio distributions in EMB2 in PCA 1 for pions with an energy of 65 GeV in the range $0.2 < |\eta| < 0.25$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

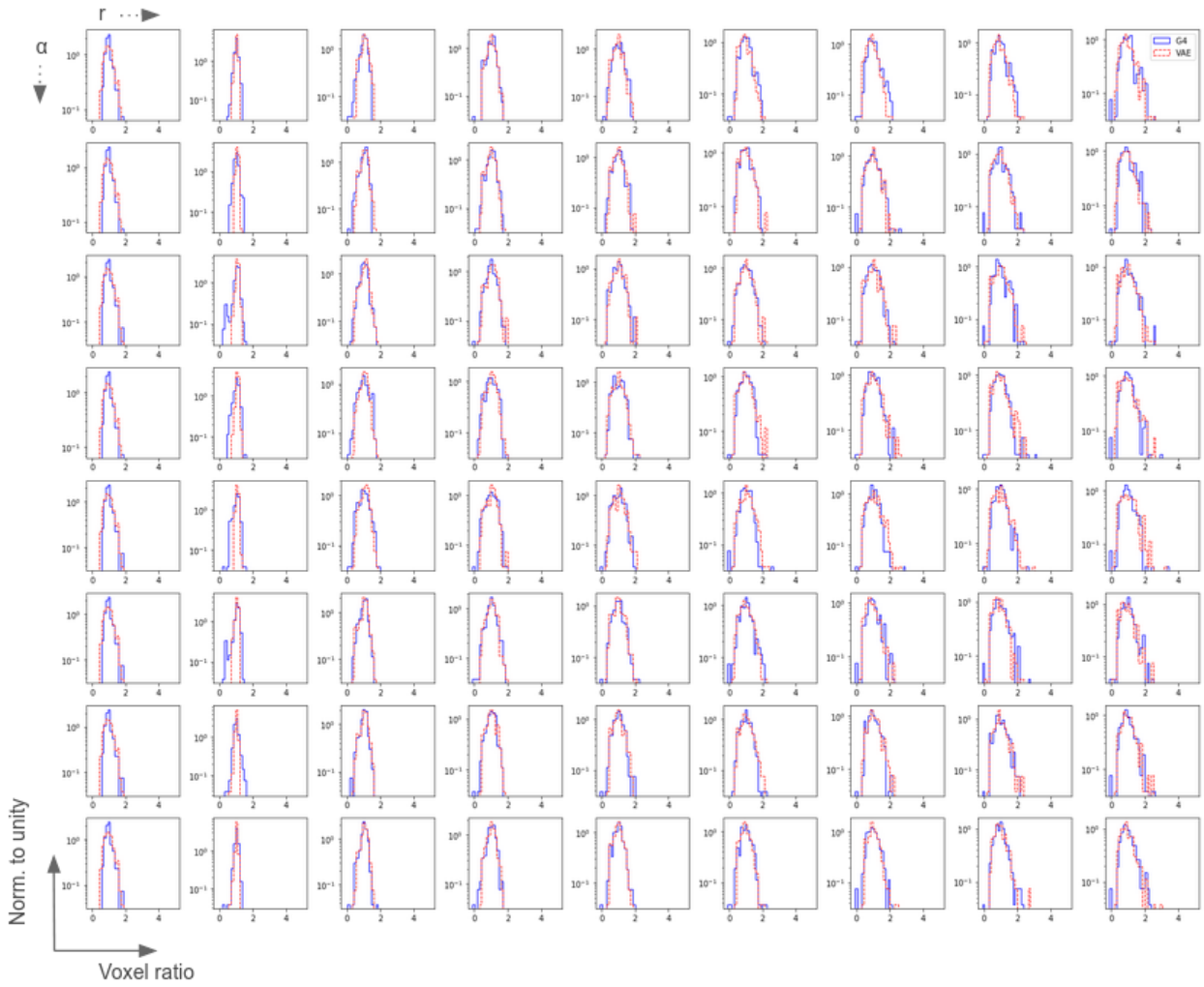


Figure 197: (r, α) voxel energy ratio distributions in EMB2 in PCA 1 for pions with an energy of 4 TeV in the range $0 < |\eta| < 0.05$. The ratios from a full detector simulation (blue line) are shown as reference and compared to the VAE (red line).

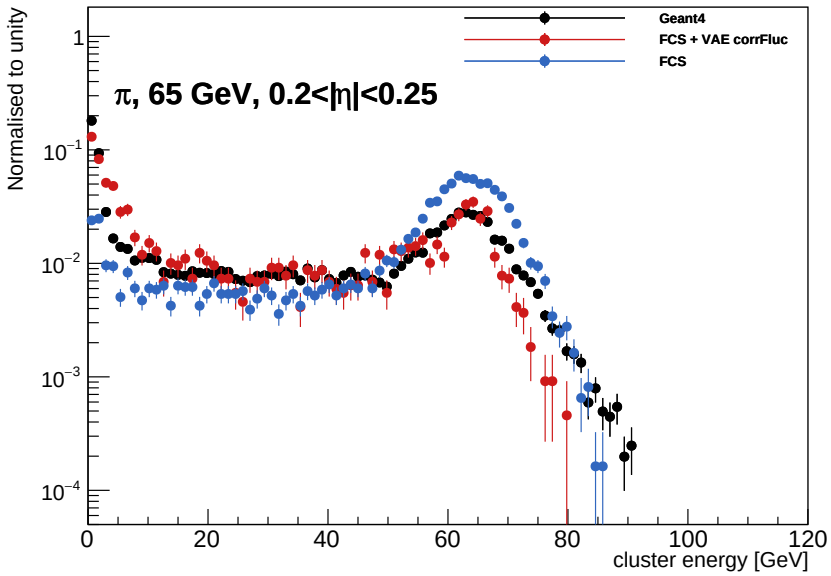


Figure 198: Cluster energy of pions with 65 GeV energy in $0.2 < |\eta| < 0.25$. The full simulation (black markers) is shown as a reference and compared to the ones of FCS (blue markers) and FCS with VAE correlated fluctuations (red markers).

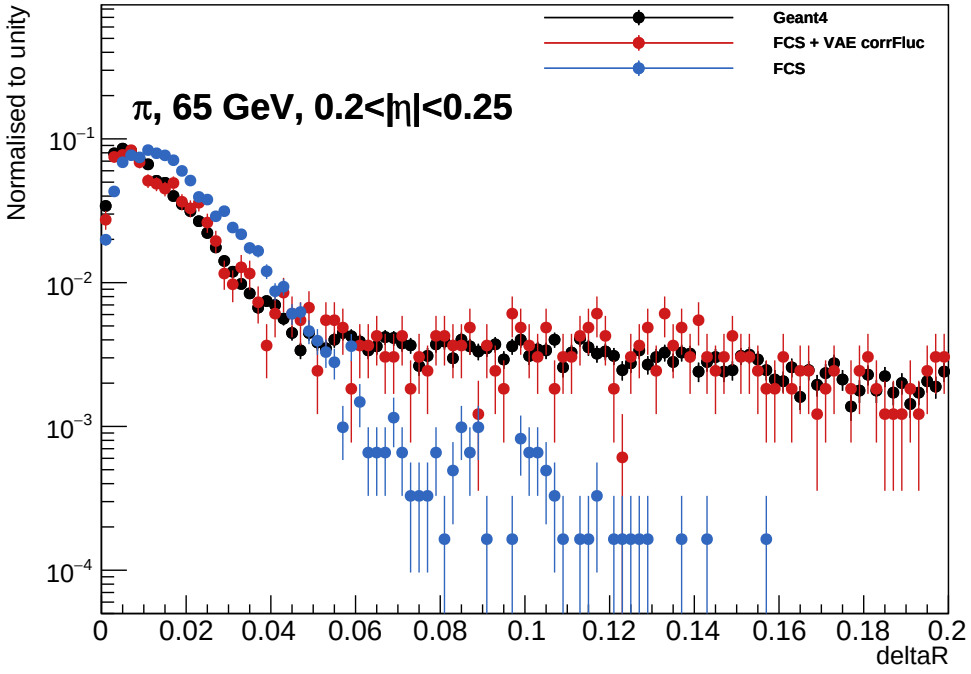


Figure 199: DeltaR of pions with 65 GeV energy in $0.2 < |\eta| < 0.25$. The full simulation (black markers) is shown as a reference and compared to the ones of FCS (blue markers) and FCS with VAE correlated fluctuations (red markers).

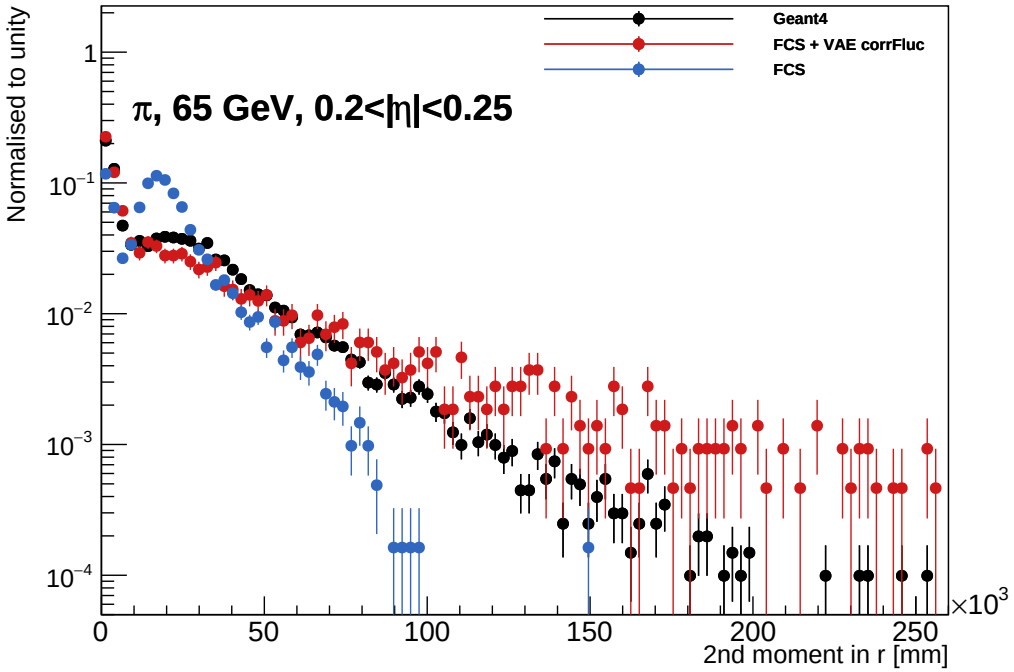


Figure 200: Second moment in r of pions with 65 GeV energy in $0.2 < |\eta| < 0.25$. The full simulation (black markers) is shown as a reference and compared to the ones of FCS (blue markers) and FCS with VAE correlated fluctuations (red markers).

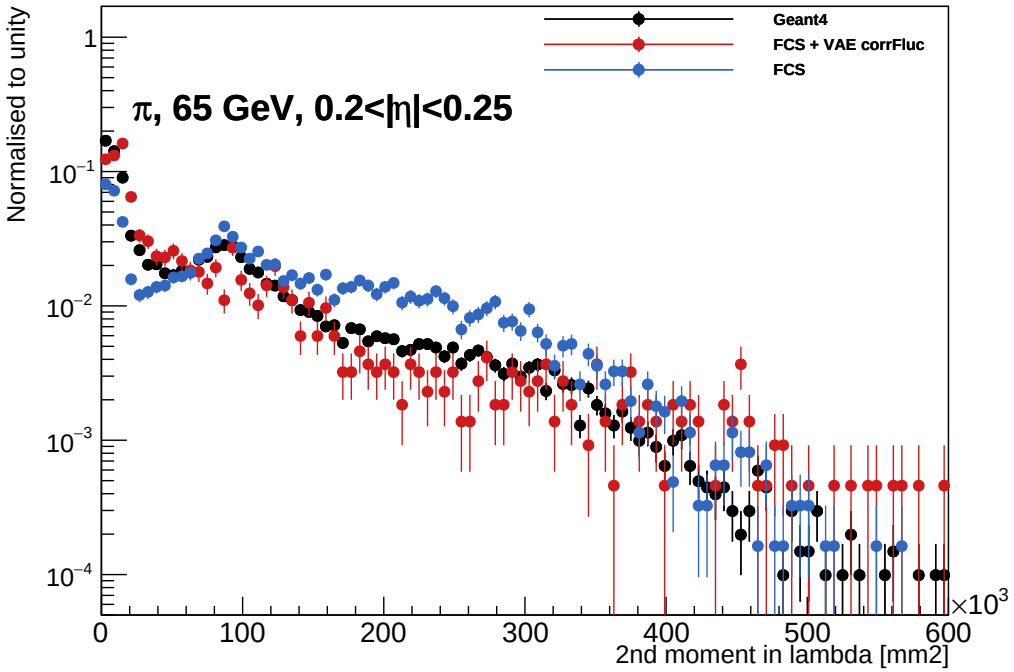


Figure 201: Second moment in lambda of pions with 65 GeV energy in $0.2 < |\eta| < 0.25$. The full simulation (black markers) is shown as a reference and compared to the ones of FCS (blue markers) and FCS with VAE correlated fluctuations (red markers).

magnetic and hadronic showers, for the lateral shape the correlations are not well modeled. These correlations are seen in individual events, which exhibit large fluctuations from the average shower. Modeling correctly these correlated fluctuations is important to model accurately the substructures.

This chapter described the CoVAE approach used to learn the correlated fluctuations using a VAE model. The VAE model was trained on the relative energy of showers to the average shower shape. The output of the VAE was applied on top of FCS as weights in the lateral parametrization.

Throughout the chapter, for the validation of the VAE performance, various quantities were used such as the energy ratio distributions per cell (voxel) and the correlation coefficients between a central cell (voxel) and the neighboring cells (voxels). Moreover, shape validation plots such as energy ratios with the RMS fluctuations about the average shape as function of the distance from the shower center were also used to demonstrate the performance.

The first prototype was based on photon particles at the cell level in a grid of 3×3 for a single calorimeter layer. The model reconstructed very well the input distributions.

Jet and particle reconstruction in the ATLAS calorimeter is based on the structure and connection of the topological energy depositions of pions. Therefore, modeling the correlated fluctuations is important for the correct modeling of the substructure. A first model for pions was designed using the cell level information and then extended to the voxel level with higher granularity information. Similar to the voxelization procedure used to train the FastCaloVSim model on voxels, the Geant4 hits in this chapter, are also binned in polar coordinates (r, α) with 8 and 9 bins in r and α respectively. Moving from cells to voxels, the energy range and η slices are also extended, and therefore the model was conditioned on these two features.

The VAE model at the voxel level was trained on each calorimeter layer and PCA bin separately. In fact, it was found that separating the PCA bins led to better performance than combining them in a single model and augmenting the condition vector with the PCA values.

The total number of voxels to use was also an important part of this study. For example, in EMB2 in the central η region, the last two PCA bins are dominated by zero values because they represent showers with a low energy deposition in that layer. Therefore, the number of voxels can be reduced to only contain the r rings where there is an energy deposition beyond a threshold (1 MeV, for example).

The number of voxels was also reduced differently for all the PCAs. In fact, the voxels of the first r ring were all filled with the same value during the generation of the input files. It resulted in a small shift of the ratio to the average shape. This was solved by considering only a single voxel in the first r ring, and therefore training the model on $1+8\times8$ voxels.

The PCA definition, by design, is not homogeneous over the energies and η regions. This adds a complexity on how to incorporate this information in the training procedure. In order to extend the model capacity to more energy and η ranges, a consistency study on the PCA definition was first conducted. The mean energy per layer was used to derive regions where the PCA definition is consistent. A model per PCA bin and per consistent region was designed, conditioned on the energy and η. The performance of the VAE was shown using standalone validation of voxel energy ratio distributions, where the agreement to Geant4 is well modeled. The Athena based validation allowed us to compare Geant4 to FCS and FCS with the VAE correlated fluctuations. The agreement improved when the correlated fluctuations were included.

The performance of the VAE was also compared to the Gaussian method, which builds a Gaussian representation of the input data. Both methods demonstrate a good performance across multiple regions of the calorimeter. Although using the Gaussian method is simpler to formulate and its implementation relies (mostly) on existing tools from FCS, extending it to include more energies and η is difficult. In fact, the Gaussian method is applied for each sample of energy and η independently, which requires a long processing time. Moreover, it saves for every sample the covariance matrix that is afterwards used to generate the showers. The VAE offers more flexibility by using a conditional learning process and saving only a single model (decoder). Moreover, it is possible to improve the architecture and fine tune the parameters of the model. In fact, a memory optimization study allows the derivation of a light version of the VAE model, which has a 0.5 MB as a memory footprint. The initial model took up to 4.6 MB. The optimization consisted of reducing the total number of nodes of the model while keeping the same performance measured by the RMS fluctuations about the average shape.

Although the consistency study in this chapter showed that the η regions with similar PCA definition can be delimited, this can be biased if the metric used only relies on the mean energy per layer. Furthermore, using a correction of the correlation would require a manual scanning per energy and η to assure the correct correspondence of the fluctuations w.r.t. to the PCA bin. An idea that can lift the burden of a manual scanning is to use an autoencoder to define a new PCA binning where the consistency remains the same across η slices.

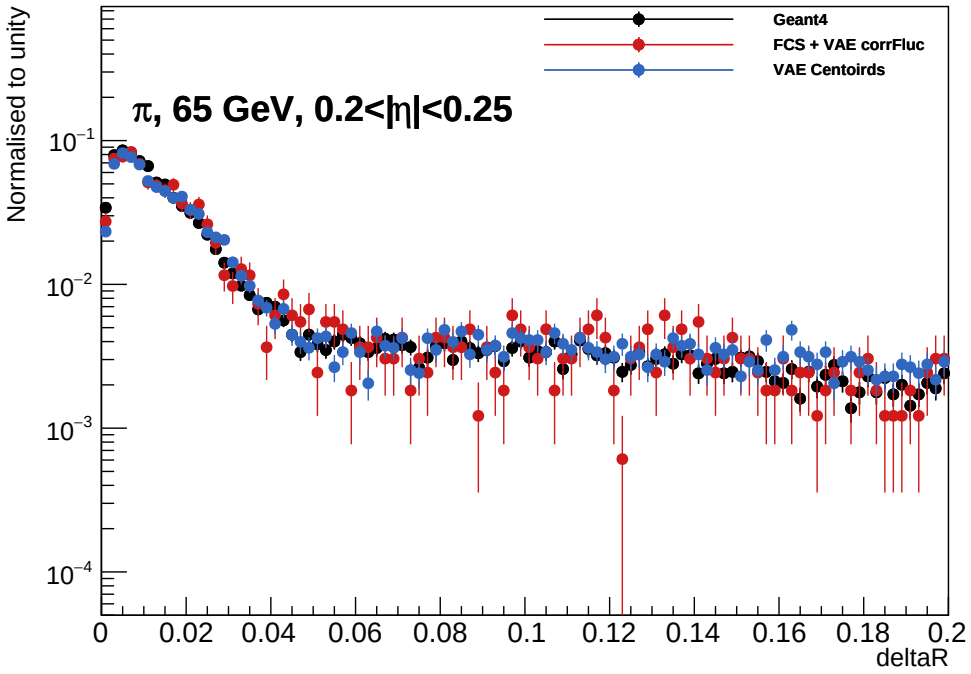


Figure 202: DeltaR of pions with 65 GeV energy in $0.2 < |\eta| < 0.25$. The full simulation (black markers) is shown as a reference and compared to the ones of VAE (FastCaloVSim) trained using centroids (blue markers) and FCS with VAE correlated fluctuations (red markers).

It is also possible to use a single VAE for both the simulation and the fluctuation modeling. Since the start of this work, the VAE has evolved significantly and whereas it was intended to extend the FCS by modeling fluctuations only, we showed in Chapter 10 its potential for an end to end simulation. Compared to the state-of-the-art FCS augmented by the VAE fluctuation modeling, some quantities are even better described, such as the distance of the cluster to the true pion shown in Figure 202.

Conclusions and Future Outlook

Machine learning is now an important research area in high energy physics. It has shown great performance in solving problems such as particle identification and event reconstruction. This thesis followed the design, implementation, and performance evaluation of novel machine learning techniques for calorimeter simulation in the ATLAS experiment.

The following points describe various challenges encountered during this work. They are summarized in such a way to allow future extension of this work and rapid insight into what worked and what did not work, as well as key elements that have a high impact on the development of machine learning models for fast shower simulation.

- Since the dataset used in this research project was initially released as part of an official ATLAS production, the availability of the data was never a concern. In the high energy range however, very few samples were available due to the massive simulation time they require.
- The preprocessing of the data, i.e., defining a shower representation general enough to be compatible with any detector region, was such an important and lengthy component of this work that it is discussed as a section of this chapter.
- The validation of the preprocessing output has to go through the ATLAS simulation chain and not solely rely on a standalone validation. This is due to the number of additional metrics considered in the ATLAS simulation chain such as: shape variable, cluster level variables and reconstructed object validation. These metrics are only available in the ATLAS reconstruction chain, and a standalone validation that would only consider energy accuracy does not fulfill the general ATLAS performance requirements. This process is not necessarily straightforward since the output of a custom preprocessing strategy is not compatible with the ATLAS simulation chain and an additional procedure had to be implemented.
- The encoding of the target variable: the energy of the shower in this context. Different choices are available: training on the raw energies, normalized energies to the calorimeter layer or mapped energies. These differences may seem subtle, but they have great impact on the accuracy of the modeling. A major innovation in this work was the incorporation of normalized energies (per calorimeter layer and per total energy) in the objective function of the model. Without this normalization, the model is unable to correctly reconstruct the total shower energy. Incorporating data knowledge in the model is generally preferred and often superior to the development of a highly complex model. In fact, in the context of fast simulation, the incorporation of domain knowledge is crucial since the alternative, i.e. developing a highly complex network, cannot satisfy the primary constraint of this work: speed.
- The evaluation of the machine learning model. Speed is only the first requirement for the model to be considered. Many more physics performance metrics have to be carefully analyzed for the model to be "accurate". These metrics are referred to as shower observables and are generally evaluated in several distribution plots. This multi-stage evaluation has a significant impact on the optimization of the hyperparameters of the model. An evaluation proxy (Chapter 7) was used to bypass this graphical performance constraint.

The research strategy adopted at the beginning of this thesis defined three main stages: data preprocessing, model design and validation, and finally Athena integration. Figure 203 illustrates the major steps involved at each stage, their connection as well as their respective categories.

Due to the challenges previously described, many models and techniques were proposed and studied in this thesis. The naming and parameters of these models can be confusing to the reader. Nevertheless, these techniques follow an improving trend, i.e., each new technique allows for a higher coverage of the parameter space. For example, FastCaloVSim relying on centroid preprocessing allows for a wider coverage of the pseudorapidity range compared to the cell preprocessing technique. The different techniques (strategies) are compared in Figure 204 as a function of the parameters they act on. The strategies are enumerated according to the chronological appearance in this work. As the strategies evolve, the wider coverage of the four parameters becomes visible.

In the following, the different building blocks of the FastCaloVSim approach are summarized.

Data Preprocessing

In this research project, the processing of the data refers to the definition of a suitable shower representation used as input to the deep learning model. A suitable representation has to be general enough to capture the full range of the detector coverage, but also detailed enough to incorporate different levels of granularity of the detector (resolution). Moreover, this representation has to be designed such that the model is able to learn the desired variables.

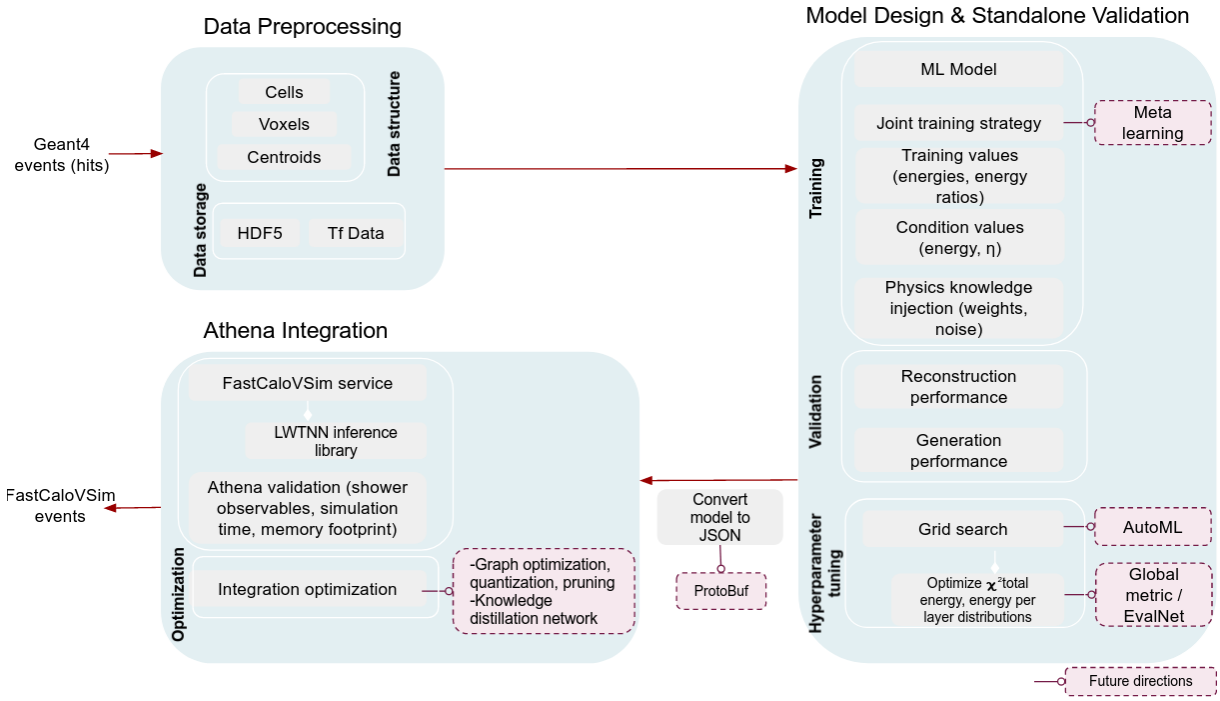


Figure 203: Overview of FastCaloVSim components with implemented models and future research directions.

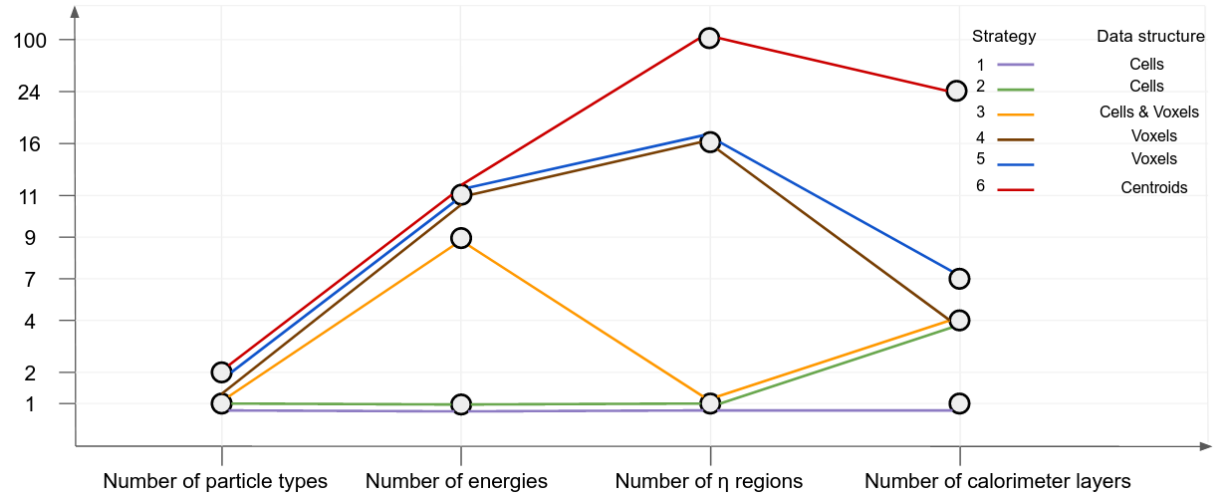


Figure 204: FastCaloVSim training strategies described by the number of particle types, energies, η regions, and calorimeter layers.

The preprocessing is applied on the Geant4 events, where the hits information is provided. This includes the energy, the position, and the direction of the incoming particle as well as its secondaries. The preprocessing uses the hits to derive the shower data structure, which can be one of the following: cells, voxels or centroids. The cell definition was found to be limited because the cells are too coarse to capture the intricate structure of a shower. In addition, they are highly dependent on the geometry of the calorimeter, which implies that the structure definition can only be applied to calorimeter regions where cells have similar shapes. To address this geometry dependence and increase the granularity of the representation, the voxel level was presented as an alternative to cells. The (r,α) voxels have a limited application, which also depends on the η slice.

We then proposed a novel structure definition method based on the K-means clustering algorithm. The definition of the centroids was derived per calorimeter layer. This derivation was independent of the truth energy and η. As a result, it allowed showers with different truth features to have the same structure as the ones the generative model was trained on. Moving from a small η range to the full detector represented a significant transition at many levels. The training strategy was adapted to handle a large number of statistics by only loading in memory events of the current training batch.

Once the data was preprocessed, it was stored in HDF5 files or as TensorFlow datasets. The former was used when the training set fitted in memory (such as cell representation). TensorFlow's datasets were used when the data did not fit into memory and had to be split into batches.

Model Design and Validation

As summarized in Figure 204, the strategy of training the VAE model started with a single particle type (photons), a single energy point, a single η region and a single calorimeter layer, using the cell level definition. Later strategies were built on improving and adapting to new data structures and incoming particles. The last strategy (number 6) extends the previous versions in terms of all features.

In this thesis we demonstrated that improving the performance of the model was not only limited by the architecture or the optimization but also by the structure of the input data: cells to voxels to centroids and using energy ratios instead of energies for the training. Using these ratios improved the performance of the model by capturing key shower quantities, such as the total energy. This reparametrization showed the importance of incorporating existing knowledge into the learning process.

The research direction taken in this thesis was based on building a generalizable model that can generate showers with a specific energy and η direction. This was achieved by incorporating these two features as conditions to the models. Thus, the VAE learned a conditional approximation function of the shower generation process. The conditional learning has also shown its advantages in interpolating or extrapolating to unseen energy points.

We adopted a joint training strategy during which the optimization of the objective function is performed over events across energies and η and where all the available statistics are used. For the model using the centroid information, the training on a GPU, took around seven days. It is possible to drastically reduce the training time by using meta learning [220]. Meta learning, or learning to learn, approaches have shown great performance in recent years. Meta learning improves the learning process using the experience of multiple tasks. The model can then be trained on few examples per energy and η and the task in this case can be to generalize to the η region. Furthermore, to reduce the training time, distributed training with data parallelism can offer orders of magnitude of speed up of the training time.

To analyze the model's performance, we presented two validation steps: a standalone and an Athena based validation. The standalone validation, as Figure 203 shows, is divided into reconstruction and generation. It is used to assess the quality of the decoder's performance in reconstructing a Geant4 shower. This reconstruction was used in the tuning of the model's hyperparameters. The Structure Similarity Index Metric (SSIM) was used to compare the Geant4 shower with the VAE-reconstructed ones. The SSIM was shown to be more efficient in capturing the similarities in the event by event comparison. For the generation, when only the decoder is used, the validation is based on comparing distributions of shower observables such as the energy per layer and the total energy.

For hyperparameter tuning of the model, a grid search was performed in parallel on multiple GPUs. While this approach allowed us to have a good set of parameters for the model, the number of combinations remains limited. For example, the latent space dimensions were varied in the range 1,10,50,100,1000. A very promising future direction is deploying Automatic Machine Learning (AutoML) solution to lift the burden of manual definition of hyperparameters.

In this work, we used a combined χ^2 to compare the distributions of the total energy and the energy per layer as a metric when running the grid search to measure the generation performance of the model. While this metric has been efficient for the photon case, it is interesting to explore a more elaborated metric. The definition of a global metric can include a number of shower observables describing the shape of the shower. This metric can

be used along with the AutoML paradigm. This metric would allow the fine-tuning of the model, especially for the case of pions where complex correlations need to be correctly learned. Another interesting idea would be to use a neural network (EvalNet in Figure 203) as a binary classifier, which, according to the Neyman-Pearson Lemma [221], if $p_{Geant4} = p_{VAE}$ a classifier cannot distinguish data from generated samples.

While the overall performance of FastCaloVSim remains good, the transition regions of the calorimeter are not well described for high energies such as 1 TeV. An interesting idea could be using a weighting technique to add a higher penalty when reconstructing the showers in these regions.

In this thesis, only photon and pion particles were used. A future extension could add electrons. Moreover, the energy range should also be extended to include very low (64 MeV) and very high energy (4 TeV) particles. Deriving the data structure using the K-means clustering algorithm was a novel idea in high energy physics. The newly implemented version "hierarchical K-means" can be used in the future extension of FastCaloVSim to completely alleviate the problem of selecting and validating the number of centroids per calorimeter layer. Moreover, the VAE models presented in this thesis are trained without the correction on energy resolution for the accordion structure of the calorimeter. This effect can be corrected for a future version of FastCaloVSim to be able to assess its performance on $H \to \gamma\gamma$.

For the model design, more complex architectures can be investigated, such as a BIB-AE [222] approach which unifies several generative models. For the validation samples, more tests can be run using, for example, $H \to \gamma\gamma$ to access the performance of a wide range of physics samples in a more dense environment.

Athena Integration

Another important module of the FastCaloVSim was its integration into the ATLAS Athena framework. A new service was implemented to perform the inference using the VAE model.

This model was converted into a JSON file to be used during the inference handled by the LWTNN library. The memory footprint and the simulation time were also quantified as a function of the complexity of the model: the size of the input/output layers, the following depth and width of the hidden layers. Using libraries such as ONNX with a model converted to a binary file (ProtoBuf) was shown to alleviate the memory footprint. Moreover, if the decoder network is too large and complex, optimization techniques such as graph optimizations, quantization and pruning can be used to reduce the complexity. Also, techniques such as "Knowledge Distillation" can be a good alternative, to compress the knowledge contained in a large model into a much smaller one.

With all the optimizations mentioned above, FastCaloVSim would be very fast and light and ready to use in the production framework. A number of these extensions are already being investigated by the author within the EP-SFT group at CERN. The general goal being to develop an agnostic detector simulation model which can be used as a fast simulation plugin in Geant4.

References

[1] S. Weinberg,The Quantum Theory of Fields, vol. 1. Cambridge University Press, 1995.

[2] P.Langacker, The Standard Model and Beyond, https://www-taylorfrancis-com.ezproxy.cern.ch/books/9781315170626.

[3] D.H. Perkins (1982) Introduction to High-Energy Physics. Addison-Wesley: 22. https://archive.org/details/IntroductionToHighEnergyPhysics/page/n31/mode/2up.

[4] Braibant, S., Giacomelli, G., Spurio, M. (2011). Particles and fundamental interactions: an introduction to particle physics. Springer Science Business Media.

[5] The Standard Model infographic developed during CERN webfest 2021. https://cds.cern.ch/record/1473657.

[6] Bialynicki-Birula, I., Bialynicka-Birula, Z. and Ter Haar, D. (1975). Quantum electrodynamics, Pergamon. https://cds.cern.ch/record/106670.

[7] Higgs, P. W. (1964). BROKEN SYMMETRIES, MASSLESS PARTICLES, AND GUAGE FIELDS. Phys. Letters, 12.

[8] Englert, F., Brout, R. (1964). Broken symmetry and the mass of gauge vector mesons. Physical Review Letters, 13(9), 321.

[9] Nason, P. (2018). The top quark mass at the LHC. arXiv preprint arXiv:1801.04826.

[10] Glashow, S. L. (1961). GLASHOW 1961. Nucl. Phys, 22, 579.

[11] Weinberg, S. (1967). WEINBERG 1967. Phys. Rev. Lett, 19, 1264.

[12] Salam, A. (1994). Weak and electromagnetic interactions. In Selected Papers Of Abdus Salam: (With Commentary) (pp. 244-254).

[13] Rajasekaran, G. (2014). Fermi and the theory of weak interactions. Reson 19, 18–44. https://doi.org/10.1007/s12045-014-0005-2.

[14] Atlas Collaboration. (2012). Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. arXiv preprint arXiv:1207.7214.

[15] Chatrchyan, S., and al. (2012). Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. Physics Letters B, 716(1), 30-61.

[16] ATLAS Collaboration. Summary plots from the ATLAS Standard Model physics group. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CombinedSummaryPlots/SM/index.html.

[17] Altarelli, G., Mele, B., Ruiz-Altaba, M. Searching for new heavy vector bosons in pbarp; colliders 1989 Z.

[18] Branco, G. C., Ferreira, P. M., Lavoura, L., Rebelo, M. N., Sher, M., Silva, J. P. (2012). Theory and phenomenology of two-Higgs-doublet models. Physics reports, 516(1-2), 1-102.

[19] D. Barducci, A. Belyaev, S. Moretti, S. De Curtis, and G. M. Pruna,LHC physics of extragauge bosons in the 4D Composite Higgs Model, EPJ Web Conf.60(2013) 20049, arXiv:1307.1782 [hep-ph].

[20] H. Georgi and S. L. Glashow,Unity of All Elementary Particle Forces, Phys. Rev. Lett.32(1974) 438–441.[22]

[21] H. Fritzsch and P. Minkowski,Unified Interactions of Leptons and Hadrons, Annals Phys.93(1975) 193–266.

[22] D. Pappadopulo, A. Thamm, R. Torre, and A. Wulzer,Heavy Vector Triplets: Bridging Theoryand Data, JHEP09(2014) 060, arXiv:1402.4431 [hep-ph].

[23] ATLAS Collaboration. Search for high-mass diboson resonances with boson-tagged jets in proton-proton collisions at $\sqrt{s}$= 8TeV with the ATLAS detector, JHEP12(2015) 055, arXiv:1506.00962 [hep-ex].

[24] CMS Collaboration. Search for massive resonances in dijet systems containing jets tagged as W or Z boson decays in pp collisions at $\sqrt{s}$=8 TeV, JHEP08(2014) 173, arXiv:1405.1994[hep-ex].

[25] ATLAS Collaboration. Combination of searches for WW, WZ, and ZZ resonances in pp collisions at$\sqrt{s}$= 8 TeV with the ATLAS detector, Phys. Lett.B755(2016) 285–305, arXiv:1512.05099 [hep-ex].

[26] Particle Data Group. Dark Matter and Cosmological Parameters, Chin.Phys.C38(2014) 090001.

[27] Super-Kamiokande Collaboration, Y. Fukudaet. al.,Evidence foroscillation of atmospheric neutrinos,Phys. Rev. Lett.81(1998) 1562–1567 [arXiv:hep-ex/9807003].

[28] S. F. King,Neutrino mass models, Rept. Prog. Phys.67(2004) 107–158[arXiv:hep-ph/0310204].

[29] Searching for Electroweak SUSY. https://atlas.cern/updates/physics-briefing/searching-electroweak-susy.

[30] Livingston, M. S.; Blewett, J. (1969). Particle Accelerators. New York: McGraw-Hill. ISBN 978-1-114-44384-6.

[31] Longer term LHC schedule. https://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm.

[32] The CERN accelerator complex. https://cds.cern.ch/record/2197559.

[33] ATLAS Collaboration. The ATLAS experiment at the CERN Large Hadron Collider, JINST3(2008) S08003

[34] CMS Collaboration. The CMS Experiment at the CERN LHC, JINST3(2008) S08004.

[35] LHCb Collaboration. The LHCb Detector at the LHC, JINST3(2008) S08005.

[36] ALICE Collaboration. The ALICE experiment at the CERN LHC, JINST3(2008) S08002.

[37] TOTEM Collaboration, G. Anelli et al., The TOTEM experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08007.

[38] LHCf Collaboration, O. Adriani et al., The LHCf detector at the CERN Large Hadron Collider, JINST 3 (2008) S08006.

[39] MoEDAL Collaboration, J. Pinfold et al., Technical Design Report of the MoEDAL Experiment.

[40] The ATLAS Collaboration. Official website of the ATLAS Experiment. http://www.atlas.ch/.

[41] ATLAS Collaboration. Luminosity Results for Run2 (2015-2017). https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResults.

[42] Kuger, F. Signal Formation Processes in Micromegas Detectors and Quality Control for large size Detector Construction for the ATLAS New Small Wheel. Diss. Wurzburg U., 2017.

[43] ATLAS Collaboration. Luminosity Results for Run2 (2011-2018). https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2.

[44] C.Y. Wong. Introduction to high-energy heavy-ion collisions. World Scientific, 1994.

[45] High Luminosity upgrade for the LHC, https://hilumilhc.web.cern.ch/content/hl-lhc-project.

[46] ATLAS Collaboration. Letter of Intent for the Phase-I Upgrade of the ATLAS Experiment,Tech. Rep. CERN-LHCC-2011-012. LHCC-I-020. https://cds.cern.ch/record/1402470.

[47] ATLAS New Small Wheels. https://cds.cern.ch/record/2669402.

[48] The New Small Wheels set ATLAS on track for high luminosity, urlhttps://home.cern/news/news/experiments/new-small-wheels-set-atlas-track-high-luminosity.

[49] ATLAS Collaboration,Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment,LHCC-I-023, CERN-LHCC-2012-022

[50] M. Gaillard and S. Pandolfi, CERN Data Centre passes the 200-petabyte milestone. http://cds.cern.ch/record/2276551.

[51] WLCG status, urlhttps://indico.cern.ch/event/877841/.

[52] R. Wigmans, Calorimetry: Energy Measurement in Particle Physics. International Series of Monographs on Physics. OUP Oxford, 2017.

[53] ATLAS collaboration. Liquid argon calorimeter technical design report. CERN-LHCC-96-041.

[54] ATLAS calorimeter performance: Technical Design Report, Technical Design Report ATLAS, 988 Geneva: CERN.

[55] M. Krammer, Detectors for Particle Physics: Calorimeters,

[56] G. Grindhammer and S. Peters, The parameterized simulation of electromagnetic showers in homogeneous and sampling calorimeters, https://cds.cern.ch/record/420530/files/0001020.pdf.

[57] C. W. Fabjan and F. Gianotti, Calorimetry for particle physics, Rev. Mod. Phys. 75 (2003) 1243–1286. https://link.aps.org/doi/10.1103/RevModPhys.75.1243.

[58] N. Ilic. Performance of the ATLAS Liquid Argon Calorimeter after three yearsof LHC operation and plans for a future upgrade.Journal of Instrumentation,9(3),2014.

[59] ATLAS Collaboration, Jet reconstruction and performance using particle flow with the ATLAS Detector, The European Physical Journal C volume 77, Article number: 466 (2017), arXiv:1703.10485 [hep-ph]

[60] C. Lippmann. Particle identification.2018, arXiv:1101.3276 [hep-ex].

[61] ATLAS collaboration. Tile calorimeter technical design report, CERN-LHCC-96-042. `http://cdsweb.cern.ch/record/331062`.

[62] ATLAS collaboration. Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter. `https://cds.cern.ch/record/2285583`.

[63] ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 no. 08, (2008) S08003.

[64] B. Aubert et al., lR D proposal: liquid argon calorimetry with LHC-performance specifications, Tech. Rep. CERN-DRDC-90-31. DRDC-P-5, CERN, Geneva, 1990.

[65] ATLAS Collaboration, A. M. Henriques Correia, The ATLAS Tile Calorimeter, Tech. Rep.ATL-TILECAL-PROC-2015-002, CERN, Geneva, Mar, 2015.

[66] P. Puzo, ATLAS calorimetry, NIMA 494 no. 1, (2002) 340 – 345. Proceedings of the 8th International Conference on Instrumentatio n for Colliding Beam Physics.

[67] A. Hrynevich, Performance of the ATLAS Tile Calorimeter, JINST 12 no. 06, (2017) C06021, `http://stacks.iop.org/1748-0221/12/i=06/a=C06021-`

[68] W. Lampl et al., Calorimeter Clustering Algorithms: Description and Performance, Tech. Rep. ATL-LARG-PUB-2008-002. ATL-COM-LARG-2008-003, CERN, Geneva, Apr, 2008.

[69] ATLAS Collaboration, N. Anjos, The ATLAS Jet Trigger for LHC Run 2, Tech. Rep. ATL-DAQ-PROC-2015-034, CERN, Geneva, Oct, 2015.

[70] Chang, H. M., Procura, M., Thaler, J., Waalewijn, W. J, Calculating track-based observables for the LHC. Physical review letters, 111(10), 102002,2013. arXiv:1303.6637 [hep-ph]

[71] ATLAS Collaboration, G. Aad et al., Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1, Eur. Phys. J. C77 (2017) 490, arXiv:1603.02934 [hep-ex] .

[72] The Expected Performance of the ATLAS Inner Detector, Tech. Rep. ATL-PHYS-PUB-2009-002. ATL-COM-PHYS-2008-105, CERN, Geneva, Aug, 2008.

[73] ATLAS Collaboration, Measurement of the photon identification efficiencies with the ATLAS detector using LHC Run 2 data collected in 2015 and 2016, Eur. Phys. J. C **79** (2019) no.3, 205 doi:10.1140/epjc/s10052-019-6650-6 arXiv:1810.05087 [hep-ex]].

[74] ATLAS Collaboration, Electron and photon energy calibration with the ATLAS detector using data collected in 2015 at $\sqrt{s} = 13$ TeV, Tech. Rep. ATL-PHYS-PUB-2016-015, CERN, Geneva, Aug, 2016.

[75] ATLAS Collaboration, S. D. Jones, The ATLAS Electron and Photon Trigger, Tech. Rep. ATL-DAQ-PROC-2018-014, CERN, Geneva, Jul, 2018.

[76] S. Zenz, Understanding Jet Structure and Constituents:Track Jets and Jet Shapes at ATLAS: ISMD2010.

[77] R. Atkin, Review of jet reconstruction algorithms, Journal of Physics: Conference Series 645 no. 1, (2015) 012008.

[78] ATLAS collaboration. Observation of W https://atlas.cern/updates/physics-briefing/observation-w-pair-from-light

[79] P. Calafiura, W. Lavrijsen, C. Leggett, M. Marino, D. Quarrie,The Athena controlframework in production, new developments and lessons learned, inComputing in highenergy physics and nuclear physics. Proceedings, Conference, CHEP'04, Interlaken,Switzerland, September 27-October 1, 2004(2005).

[80] The Athena Framework. `https://atlassoftwaredocs.web.cern.ch/athena/athena-intro/`.

[81] ATLASCollaboration, ATLAS Computing Tech-nical Design Report,ATLAS-TDR-017, CERN-LHCC-2005-022 (2005).

[82] Bandieramonte, M., Bianchi, R. M., Boudreau, J. (2020). FullSimLight: ATLAS standalone Geant4 simulation. In EPJ Web of Conferences (Vol. 245, p. 02029). EDP Sciences.

[83] S. Agostinelli et al., Geant4 - a simulation toolkit, Nucl.Instr. Methods Phys. Res.A 506(2003) 250–303.

[84] J. Allison et al., Geant4 Developments and Applications,IEEE Transactions on Nuclear Science53(2006) 270–278.

[85] I. Bird et al., Update of the Computing Models of the WLCG and the LHC Experiments.

[86] ATLAS Collaboration. The ATLAS Simulation Infrastructure, Eur. Phys. J. C70(2010) 823,arXiv:1005.4568 [physics.ins-det].

[87] J. D. Chapman, Approaches to speed up Geant4 Simulation in ATLAS, Tech. Rep.ATL-COM-SOFT-2018-005, CERN, Geneva, Mar, 2018.

[88] ATLAS Collaboration. Public results. Computing and Software. `https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults`.

[89] ATLAS Collaboration. The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim,ATL-PHYS-PUB-2010-013, 2010.

[90] G. Barrand et al., GAUDI - A software architecture andframework for building LHCb data processing applica-tions, in Computing in High Energy and Nuclear Physics 2000 Conference (CHEP2000), Pavia, Italy, February 7-11, 2000, ed. M. Mazzucato, (Padua: INFN, 2000)

[91] M. Cattaneo et al., Status of the GAUDI event-processingframework, in Computing in High Energy and NuclearPhysics 2001 Conference (CHEP2001), IHEP, Beijing,China, September 3-7, 2001, ed. H.S. Chen, (e-article,2001).

[92] Binet S, Calafiura P, Jha M K, Lavrijsen W, Leggett C, Lesny D, Severini H, Smith D, Snyder S,Tatarkhanov M, Tsulaia V, Van Gemmeren P and Wash brook A 2012 Journal of Physics: Conference Series 36801 2018.

[93] T. Sj ostrand, S. Mrenna and P. Skands, PYTHIA6.4 physics and manual,JHEP05(2006) 026,[hep-ph/0603175]

[94] T. Gleisberg et al., SHERPA 1.alpha, a proof-of-conceptversion, JHEP02(2004) 056, [hep-ph/0311263.

[95] B.P. Kersevan and E. Richter-Was, The Monte Carloevent generator AcerMC version 2.0 with interfaces toPYTHIA 6.2 and HERWIG 6.5, 2004, [hep-ph/0405247].

[96] E. Barberio, B. van Eijk and Z. Was, PHOTOS: A Univer-sal Monte Carlo for QED radiative corrections in decays,Comput. Phys. Commun.66(1991) 115–128.

[97] S. Jadach, J.H. Kuhn and Z. Was, TAUOLA: A Libraryof Monte Carlo programs to simulate decays of polarizedtau leptons, Comput. Phys. Commun.64(1990) 275–299.

[98] M. Dobbs and J.B. Hansen, The HepMC C++ MonteCarlo event record for High Energy Physics, Comput.Phys. Commun.134(2001) 41–46.

[99] M. Smizanska, S.P. Baranov, J. Hrivnac and E. Kner-inger, Overview of Monte Carlo simulations for ATLASB-physics in the period 1996-1999, ATL-PHYS-2000-025(2000).

[100] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, Event generation with SHERPA 1.1, JHEP 2 (2009) 007, arXiv:0811.4622 [hep-ph].

[101] S.DeanandP.Sherwood,Athena-Atlfast (2008). `http://www.hep.ucl.ac.uk/atlas/atlfast/`.

[102] ATLAS Collaboration,The ATLAS experiment at the CERN Large Hadron Collider, JINST3(2008) S08003

[103] Beckingham, M., Duehrssen, M., Schmidt, E., Shapiro, M., Venturi, M., Virzi, J., ... Yamanaka, T. (2010, October). The simulation principle and performance of the ATLAS fast calorimeter simulation FastCaloSim. In International Conference on Computing in High Energy and Nuclear Physics.

[104] Schaarschmidt, J., ATLAS Collaboration. (2017, October). The new ATLAS Fast Calorimeter Simulation. In Journal of Physics: Conference Series (Vol. 898, No. 4, p. 042006). IOP Publishing.

[105] Ritsch, E. The ATLAS Integrated Simulation Framework,(2013). `https://cds.cern.ch/record/1532476`.

[106] PCA principle and the implementation used for FastCaloSimV2 is described at `https://root.cern.ch/doc/master/classTPrincipal.html`.

[107] Cox, R.T. (1946). Probability, Frequency, and Reasonable Expectation. American Journal of Physics. 14 (1): 1–10. doi:10.1119/1.1990764.

[108] The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[109] Dahl, G. E., Ranzato, M., Mohamed, A., and Hinton, G. E. (2010).Phone recognition with the mean-covariance restricted Boltzmannmachine.

[110] Bengio, Y. (2008). Neural net language models.Scholarpedia, 3(1).

[111] Dinh, Laurent, David Krueger, and Yoshua Bengio. "Nice: Non-linear independent components estimation." arXiv preprint arXiv:1410.8516 (2014).

[112] Pearson, K. On lines and planes of closest fit to systems ofpoints in space.Philosophical Magazine, 1901.

[113] Fisher, Ronald A. The use of multiple measurements intaxonomic problems.Annals of eugenics, 7(2):179–188,193.

[114] N. Tishby, F. Pereira, and W. Bialek. Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing, page 368-377. (1999).

[115] Diederik P Kingma, Max Welling, Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML].

[116] Rezende, D. J., Mohamed, S., Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082.

[117] Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications.

[118] Gelfand, A. and Smith, A. (1990). Sampling based approaches to calculating marginal densities. Journal of the American Statistical Association, 85:398–409.

[119] Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. Journal of the American statistical Association 112.518 (2017): 859-877.

[120] Jordan, M. I., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). Introduction to variational methods for graphical models.Machine Learning, 37:183–233.

[121] Mitchell, Tom. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 0-07-042807-7.

[122] ATLAS Collaboration. Measurement of the tau lepton reconstruction and identification performance in the ATLAS experiment using pp collisions at $\sqrt{s}$=13 TeV.

[123] ATLAS Collaboration. Deep generative models for fast shower simulation in ATLAS (2018).

[124] Kuusela, M., Vatanen, T., Malmi, E., Raiko, T., Aaltonen, T., Nagai, Y. (2012). Semi-supervised anomaly detection–towards model-independent searches of new physics. In Journal of Physics: Conference Series (Vol. 368, No. 1, p. 012032). IOP Publishing.

[125] Research Blog: AlphaGo: Mastering the ancient game of Go with Machine Learning". Google Research Blog. 27 January 2016.

[126] Stefano Carrazza, Frédéric A. Dreyer, Jet grooming through reinforcement learning, Phys. Rev. D 100, 014014, 15/07/2019 [arXiv:1903.09644] DOI: 10.1103/PhysRevD.100.014014.

[127] Goodfellow-et-al-2016, Deep Learning Book, http://www.deeplearningbook.org

[128] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, Nature 323 no. 6088, (1986) 533.

[129] N. Srivastava et al., Dropout: A Simple Way to Prevent Neural Networks from Overfitting, J.Mach. Learn. Res. 15 no. 1, (2014) 1929–1958.

[130] S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, pp. , 448–456. JMLR.org, 2015.

[131] R. Caruana, S. Lawrence, and C. L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, In proceedings of Neural Information Processing Systems.

[132] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio,Generative adversarial networks, ArXiv e-prints(2014) [1406.2661].

[133] Lars Mescheder, Sebastian Nowozin, Andreas Geiger. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. arXiv:1701.04722 [cs.LG].

[134] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In ECCV, 2016.

[135] Raymond Yeh, Ziwei Liu, Dan B Goldman, Aseem Agarwala, Semantic Facial Expression Editing using Autoencoded Flow. arXiv:1611.09961 [cs.CV].

[136] Van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse,Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In Advances In Neural Information Processing Systems, pp. 4790–4798, 2016a.

[137] Van den Oord, Aaron van den, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks.arXiv preprint arXiv:1601.06759, 2016b.

[138] Shengjia Zhao, Jiaming Song, Stefano Ermon. Learning Hierarchical Features from Generative Models. arXiv:1702.08396 [cs.LG].

[139] Mehdi Mirza, Simon Osindero, Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs.LG].

[140] Alec Radford, Luke Metz, Soumith Chintala , Unsupervised Representation Learning with Deep Convolutional Generative. arXiv:1511.06434 [cs.LG].

[141] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN. arXiv:1701.07875 [stat.ML].

[142] Emily Denton, Soumith Chintala, Arthur Szlam, Rob Fergus, Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. arXiv:1506.05751 [cs.CV].

[143] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel, InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. arXiv:1606.03657 [cs.LG].

[144] P. Gallinari, Y. LeCun, S. Thiria, and F. Fogelman-Soulie. Memoires associatives distribuees. InProceedings of COGNITIVA 87, Paris, La Villette, 1987.

[145] Hinton, G. E., Zemel, R. S. (1994). Autoencoders, minimum description length and Helmholtz free energy. In Advances in neural information processing systems (pp. 3-10).

[146] Hinton, G. E., Krizhevsky, A., Wang, S. D. (2011, June). Transforming auto-encoders. In International conference on artificial neural networks (pp. 44-51). Springer, Berlin, Heidelberg.

[147] D.J Rezende, S Mohamed, D Wierstra - arXiv preprint arXiv:1401.4082, 2014.

[148] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs.CL].

[149] Laurent Dinh, Jascha Sohl-Dickstein, Samy Bengio .Density estimation using Real NVP. arXiv:1605.08803 [cs.LG].

[150] Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In AISTATS, volume 1, page 2, 2011.

[151] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In ICML, pages 881–889, 2015.

[152] Kingma, Durk P., et al. "Improved variational inference with inverse autoregressive flow." Advances in neural information processing systems. 2016.

[153] Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In NIPS, 2015.

[154] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In NIPS, 2015.

[155] Siavash Arjomand Bigdeli, Matthias Zwicker. Image Restoration using Autoencoding Priors.arXiv:1703.09964 [cs.CV].

[156] VAE-celebA project. url:https://github.com/yzwxx/vae-celebA.

[157] T. M. Cover and J. A. Thomas.Elements of Information Theory. John Wiley,New York, 1991.

[158] R. D. Ball et al. "Parton distributions for the LHC Run II."JHEP04 (2015), p. 040.doi:10.1007/JHEP04(2015)040. arXiv:1410.8849 [hep-ph].

[159] A. De Simone and T. Jacques,Eur. Phys. J. C79, no. 4, 289 (2019)doi:10.1140/epjc/s10052-019-6787-3 [arXiv:1807.06038 [hep-ph]].

[160] Cerri, O., Nguyen, T. Q., Pierini, M., Spiropulu, M., Vlimant, J. R. (2019). Variational autoencoders for new physics mining at the large hadron collider. Journal of High Energy Physics, 2019(5), 36.

[161] Gleyzer, S., Seyfert, P., Schramm, S. (1807). Machine Learning in High Energy Physics Community White Paper (2019).

[162] V. V. Gligorov and M. Williams. "Efficient, reliable and fast high-level triggering using a bonsai boosteddecision tree."JINST8 (2013), P02013.doi:10.1088/1748-0221/8/02/P02013. arXiv:1210.6861[physics.ins-det].

[163] D. Derkach et al. "Machine-Learning-based global particle-identification algorithms at the LHCb experiment."Proceedings, 18th International Workshop on Advanced Computing and Analysis Techniques inPhysics Research (ACAT 2017): Seattle, United States of America, August 21-25, 2017.

[164] D. Derkach, M. Hushchyn and N. Kazeev, EPJ Web Conf. 214, 06011 (2019). DOI: 10.1051/epjconf/201921406011.

[165] ATLAS Collaboration, Published in:Phys.Rev.D 97 (2018) 7, 072016 e-Print:1712.08895 [hep-ex].

[166] Aiello, Sebastiano, et al. "Event reconstruction for KM3NeT/ORCA using convolutional neural networks." arXiv preprint arXiv:2004.08254 (2020).

[167] CMS Collaboration.,CMS PAS HIG-18-030, CERN, June 24, 2019.

[168] T. Vuillaumeet al.,EPJ Web Conf.214, 06020 (2019). DOI: 10.1051/epj-conf/201921406020.

[169] D. Derkach et al. "Machine-Learning-based global particle-identification algorithms at the LHCb experiment."Proceedings, 18th International Workshop on Advanced Computing and Analysis Techniques inPhysics Research (ACAT 2017): Seattle, United States of America, August 21-25, 2017.

[170] Paganini, Michela, Luke de Oliveira, and Benjamin Nachman. "CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks." Physical Review D 97.1 (2018): 014021.

[171] K. Deja, T. Trzcinski and . Graczykowski EPJ Web Conf. 214, 06003 (2019). DOI:10.1051/epjconf/201921406003.

[172] A. Hoecker et al., arXiv:physics/0703039, 2007.

[173] E. Rodrigues et al., arxiv.org/abs/2007.03577, 2007.

[174] R. Brun and F. Rademakers, Proceedings AIHENP'96 Workshop, Lausanne, Sep.1996, Nucl. Inst. and Meth. in Phys. Res. A, 389, 81-86 (1997).

[175] Higgs challenge. url:https://www.kaggle.com/c/higgs-boson.

[176] TrackMLChallenge. url:https://www.kaggle.com/c/trackml-particle-identification.

[177] ACTS. url:http://acts.web.cem.ch/ACTS/latest/doctindex.html.

[178] J.-R. Vlimantet al.,EPJ Web Conf.214, 06025 (2019). DOI: 10.1051/epj-conf/201921406025.

[179] V. Estradeet al.,EPJ Web Conf.214, 06024 (2019). DOI: 10.1051/epj-conf/201921406024.

[180] D. Derkach, M. Hushchyn and N. Kazeev,EPJ Web Conf.214, 06011 (2019). DOI:10.1051/epjconf/201921406011

[181] ATLAS Collaboration, Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton-proton collision data, JINST 14 (2019) P12006, arXiv: 1908.00005 [hep-ex].

[182] https://twiki.cern.ch/twiki/bin/viewauth/AtlasProtected/EGammaD3PDtoxAOD

[183] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In Advances in Neural Information Processing Systems, pp. 2234–2242, 2016.

[184] D. H. Guest et al.,lwtnn/lwtnn: Version 2.9, version v2.9, 2019 (cit. on p. 25).

[185] ATLAS Collaboration, Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton–proton collision data, JINST 14 (2019) P12006, arXiv: 1908. 00005 [hep-ex] (cit. on p. 25).

[186] ATLAS Collaboration,ATLAS HL-LHC Computing Conceptual Design Report, tech. rep.,970CERN, 2020,url:https://cds.cern.ch/record/2729668(cit. on p. 3)

[187] C. Bozzi,LHCb Computing Resource usage in 2014 (II), Tech. Rep. LHCb-PUB-2015-004.CERN-LHCb-PUB-2015-004, CERN, Geneva, Jan, 2015.https://cds.cern.ch/record/1984010.

[188] The HDF Group, Hierarchical Data Format, version 5, 1997-2018.http://www.hdfgroup.org/HDF5/

[189] J. Allison et al.,Recent developments in Geant4, Nucl. Instrum. Meth.A835(2016) 186

[190] H. W. Bertini and M. P. Guthrie,News item results from medium-energy intranuclear-cascade calculation,Nucl. Phys.A169(1971) 670

[191] . Andersson, G. Gustafson and B. Nilsson-Almqvist,A Model for Low p(t) Hadronic Reactions,with Generalizations to Hadron - Nucleus and Nucleus-Nucleus Collisions,Nucl. Phys.B281(1987) 289.

[192] T. Tieleman and G. Hinton,Lecture 6.5 – RMSProp: Divide the gradient by a running average of its recent magnitude,Coursera: Neural networks for machine learning4(2012) 26.

[193] S.Ioffe and C.Szegedy,Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, pp. , 448–456. JMLR.org, 2015.

[194] Creswell, A., Arulkumaran, K. and Bharath, A.A., 2017. On denoising autoencoders trained to minimise binary cross-entropy. arXiv preprint arXiv:1708.08487.

[195] Pattern Recognition and Machine Learning (Information Science and Statistics)

[196] Rousseaux, Colin G., and Shayne C. Gad. "Statistical assessment of toxicologic pathology studies." Haschek and Rousseaux's Handbook of Toxicologic Pathology (2013): 893-988.

[197] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24.5 (2002): 603-619.

[198] Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." Advances in neural information processing systems. 2002.

[199] Maimon, Oded, and Lior Rokach, eds. "Data mining and knowledge discovery handbook." (2005).

[200] Lee, John M. Introduction to Riemannian manifolds. Springer International Publishing, 2018.

[201] Cheng, Yizong (August 1995). "Mean Shift, Mode Seeking, and Clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence. 17 (8): 790–799

[202] url: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

[203] ATLAS Collaboration, Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1, Eur. Phys. J. C 77 (2017) 490, arXiv: 1603.02934 [hep-ex] (cit. on p. 30).

[204] Deeba, F., Dharejo, F. A., Zawish, M., Memon, F. H., Dev, K., Naqvi, R. A., ... Du, Y. A novel image dehazing framework for robust vision-based intelligent systems. International Journal of Intelligent Systems.

[205] Lv, Z., Xu, Q., Schoeffmann, K., Parkinson, S. (2021, July). A Jensen-Shannon Divergence Driven Metric of Visual Scanning Efficiency Indicates Performance of Virtual Driving. In 2021 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.

[206] M. Cacciari, G. P. Salam and G. Soyez, The anti-k_t jet clustering algorithm, JHEP 04 (2008) 063, arXiv: 0802.1189 [hep-ph] (cit. on p. 30).

[207] M. Cacciari, G. P. Salam and G. Soyez, FastJet user manual, Eur. Phys. J. C 72 (2012) 1896, arXiv: 1111.6097 [hep-ph] (cit. on p. 30).

[208] ONNX runtime. `https://www.onnxruntime.ai/`

[209] S. Popić, D. Pezer, B. Mrazovac and N. Teslić, "Performance evaluation of using Protocol Buffers in the Internet of Things communication," 2016 International Conference on Smart Systems and Technologies (SST), Osijek, Croatia, 2016, pp. 261-265, doi: 10.1109/SST.2016.7765670.

[210] K. Maeda, "Performance evaluation of object serialization libraries in XML, JSON and binary formats," 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, Thailand, 2012, pp. 177-182, doi: 10.1109/DICTAP.2012.6215346.

[211] ATLAS Collaboration. Fast simulation of the ATLAS calorimeter system with Generative Adversarial Networks, ATL-SOFT-PUB-2020-006, 2020. `https://cds.cern.ch/record/2746032`.

[212] FastCaloSim Plots for CHEP conference 2019. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2019-008/.

[213] VAE for photon shower simulation in ATLAS 2019. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2019-007/

[214] ATLAS Collaboration. Deep generative models for fast shower simulation in ATLAS (2021).

[215] Lu, H., Du, M., Qian, K., He, X., Wang, K. (2021). GAN-based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots. IEEE Sensors Journal.

[216] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29, 2234-2242.

[217] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[218] Chollet, F. (2016). Building autoencoders in keras. The Keras Blog, 14.

[219] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

[220] S. Thrun and L. Pratt, "Learning To Learn: Introduction AndOverview," inLearning To Learn, 1998.

[221] Neyman, J., Pearson, E. S. (1933). IX. On the problem of the most efficient tests of statistical hypotheses. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 231(694-706), 289-337.

[222] Voloshynovskiy S, Kondah M, Rezaeifar S, Taran O, Holotyak T, Rezende DJ (2019) Information bottleneck through variational glasses. arXiv:1912.00830 [cs.CV]

Acknowledgements

This thesis has been a true adventure which started with a dream when I, as a undergraduate student, visited CERN for the first time in 2013 during the open days. My PhD journey would not have been possible without the support and encouragement of many amazing people who positively impacted my professional and personal growth. I hope I will not leave anyone out.

Profound gratitude goes to my thesis advisor Prof. Tobias Golling. Thank you for giving me the opportunity to join the ATLAS group at Geneva university. Thank you for all your support, your advice, your valuable questions and comments, for all the fruitful discussions and for your availability and flexibility. All your comments helped me become the researcher I am.

I would like to thank my thesis co-supervisor Prof. Svyatoslav Voloshynovskyy for all the fruitful machine learning discussions, which contributed to building and improving FastCaloVSim. Thank you for all your support and your brilliant ideas! It was a pleasure to work and learn from you. I am also delighted that we had a great collaboration and a fusion of physics and machine learning and which opened doors to many more projects.

I am thankful to Dr. Graeme Stewart for all your help, your support and all the discussions we had nearly one year before the official start of my thesis. Thank you for being enthusiastic about machine learning and thank you for all your continuous support throughout all the years of my PhD journey. I can't thank you enough for your encouragement and for helping me become the researcher I am.

A special thanks goes to Mario Lassnig. Thank you for your precious help way before I started my thesis. Thank you for making this journey possible!

Many studies in this work would not have been possible without the talent and expertise of Michael Duehrssen-Debling, who is a brilliant physicist and a truly dedicated mentor. I cannot thank you enough for all the great discussions and all your great ideas and for making me love calorimetry. This work would not have been the same if I sat anywhere else than next to your office!

I am thankful to Stefan Gadatsch and Johnny Raine for their feedback and constructive input all along this work.

I would like to thank all the present and past members of the ATLAS- UniGe group for all the fruitful discussions.

I would like to thank all the ATLAS FastCaloSim group. Thank you all for all the guidance, inputs and help. A special thanks goes to Michele Faucci Giannelli. Thank you Michele for answering all my questions, for always providing support, help and valuable advice. All the Athena integration part would not have been possible without your help!

I would like to thank Anna Zaborowska for all your support and encouragement. Thank you and Michele for the rehearsal and all the valuable feedback which enormously contributed in better preparing my defense.

I am thankful to everyone in the third floor of building 40 at CERN for their help, the fun times and the laughs we had together. A special thanks goes to Noemi Calace. Thank you Noemi for your advice and your help especially all the ROOT tips and tricks. Grazie mille!

A special thanks to Pauline Gagnon whom I first met during the CERN open days in 2013 where I had the chance for the first time to hear about the higgs boson from a CERN physicist. Thank you Pauline for opening the door of this research journey. Thank you for asking researchers and group leaders about opportunities for me to do a thesis at CERN. Thank you for all your support! Because of this opened door, in the next CERN open days 2019 I participated as an ATLAS underground guide to share my passion with visitors from all around the world!

I would like to thank all my friends and especially, my best friend Sabrina Amrouche. I would not have been able to start this thesis without your support. Thank you for being my office mate, for sharing every step of this journey with me. Thank you for all the experiences we had together, for all the fun times and for being always there!

I cannot thank enough my parents for their support and continuous encouragement. Thank you for always being present for me, thank you for always giving me the strength. I cannot, also, thank enough my two sisters and brother for your support and advice.