ETP-KA/2022-09

Fast Simulation of Electromagnetic Calorimeter Responses using Deep Learning

Zur Erlangung des akademischen Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
(Dr. rer. nat.)
von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Jubna Irakkathil Jabbar

aus Kozhikode, Indien

Tag der mündlichen Prüfung: 29. July 2022
Referent: Prof. Dr. Günter Quast
Korreferent: Prof. Dr. Florian Bernlochner

Contents

TERMS AND DEFINITIONS

- Fast Simulation - Simulation of Particle showers using deep learning models.
- GAN - Generative Adversarial Network
- WGAN - Wasserstein Generative Adversarial Network
- VAE - Variational Autoencoders
- GEANT4 - A software package used for simulation of particle showers.
- Tensorflow - Python-based software package to build deep learning models.
- Pytorch - Python-based software package to construct deep learning models
- Generative models - The deep learning models which learn the probability distribution of a dataset through training and try to generate new images which look like datasets.
- Epoch - Epoch is a step where entire datasets have been passed to the model once for training.
- E_1/E_9 - The ratio of the amount of energy deposited in the innermost crystal to that of the 9 inner crystals.
- E_9/E_{21} - The ratio of energy deposited in the inner 9 crystals to that of all the crystals except the crystals in the corners.
- E_9/E_{25} - The ratio of energy deposited in the inner 9 crystals to that of all 25 crystals in 5 × 5 crystals.
- E_{16}/E_{100} - The ratio of energy deposited in the inner 16 crystals to that of all crystals in 10 × 10 crystals.
- E_{196}/E_{900} - The ratio of energy deposited in the inner 196 crystals to that of all crystals in 30 × 30 crystals.
- N_{Hits} - Distribution of the number of active crystals in the crystal array.

- ECL - Belle II electromagnetic calorimeter.
- basf2 - The software framework developed for the Belle II analysis.
- $\mathrm{P_x}, \mathrm{P_y}$ - The impact position labels.

1 INTRODUCTION

High energy physics is one of the branches of physics that have made remarkable discoveries over the years. Particle physics explores different elementary particles, their interaction with the fundamental forces, and the other parameters involved in their measurements. It tries to unravel various mysteries in the universe like the big bang and dark matter. The Standard Model is an important concept of particle physics. However, it lacks completeness since it fails to explain various phenomenons like the abundance of matter compared to antimatter in the universe. To dive deep into these topics, collider experiments like ATLAS, CMS or Belle II were built to record collisions of elementary particles at high energies. To conduct these experiments and draw conclusions about the Standard Model a vast amount of computing resources are required.

In high energy particle collision experiments, calorimeters play a crucial role in detecting neutral particles and measure their energy. To understand and calibrate the calorimeter detector response, sophisticated simulations are required. Such simulations can be carried out for instance with simulation toolkits like the Geant4. But the high computational complexity and time consumption of these simulations pose a challenge. Run time and resource consumption of the simulation process increase as the energy and the number of particles involved in the shower process increase. As an example for this, Figure 1.1 shows the CPU resource requirement for the various sub-detectors of the Belle II experiment. The typical simulation time of an entire event is approximative 0.8 seconds.

The last decade saw the rapid progress of the area of machine learning and neural networks and their application to various fields. In particle physics, machine learning has an established use for classification and regression tasks [2] [3]. More recently, neural networks have been used for fast simulations and reconstruction tasks [4]. For fast simulations, generative neural networks are often employed to generate detector responses based on

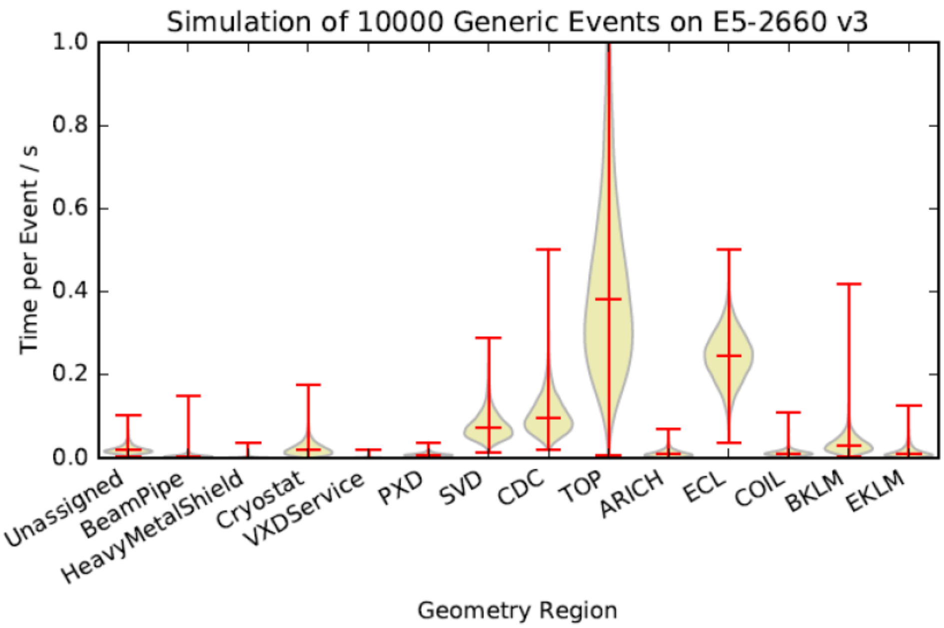


Figure 1.1.: Estimated CPU resources required for the different parts of the Belle II detector [1].

specific training data sets, with the goal to replace the resoruce intensive nominal MC simulation. Two succesful implementations of this can be found in Refs. [5, 6]: The first article on "Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network" investigates the fast simulation of electromagnetic showers of electrons in a configuration of high granularity calorimeter (HGCAL) prototype of the CMS experiment. The second article, "Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed" similarly studies the fast simulation on a prototype calorimeter for the proposed International Large Detector (ILD) of a future e^+e^- Higgs factory.

This thesis aims to develop a fast simulation of the MC particle shower responses of the Belle II electromagnetic calorimeter (ECL) using deep learning models. The simulation of the ECL detector response contributes to a significant fraction of the overall simulation time (see Figure 1.1): on average about 0.2 seconds are needed to fully simulate a particle shower. The Belle II experiment aims to record 50 times more collision data than its predecessor experiment Belle [7]. To analyse the anticipated $50\,\mathrm{ab}^{-1}$ of collisions, large MC samples of at least a similar size or more will be needed. This represents a challenge for the Belle II collaboration in terms of future computing needs. Fast simulations could play an important role in reducing the simulation time. The studies carried out in the context of this thesis make use of simulations based on the Geant4 simulation toolkit. To implement fast simulations, generative networks such as the Wasserstein Generative Adverserial Network (WGAN), Variational Autoencoders (VAE) and combinations of both are investigated. The network structures are implemented and trained using the Tensorflow and Pytorch software frameworks. The rest of this thesis is structured as follows:

Chapter 2 gives a brief description of calorimetry and also provides details of a standalone Geant4 calorimeter crystal setup with a pixel size of 5 × 5. This setup is used to produce particle shower datasets to train the various models.

Chapter 3 introduces the concept of Generative Adversarial Networks and the architecture of Wasserstein Generative Adversarial Networks. The results of the application of a WGAN architecture to the task of simulating fast electron shower responses and their limitations are discussed. By comparing to the fully simulated electron shower response from Geant4 an optimal WGAN architecture is selected, which is further studied using the full Belle II electromagnetic calorimeter. The chapter concludes with a discussion of the results of applying the fast simulation to hadronic showers and their challenges.

Chapter 4 focuses on Variational Autoencoders. The limitation of GAN is the lack of an encoder network to provide inference on a training data set [8]. This shortcoming can be improved upon by combining VAE and GAN architectures. These model variants have the ability to learn specific features of a training data set and thus have the potential to improve the quality of fast simulations. Several training approaches are investigated and two promising ones are presented.

Chapter 5 analyses the application of one of the VAE-WGAN models in the context of a highly granular electromagnetic calorimeter. Two configurations are studied with a 100 and 900 pixels, respectively. The shower response for electrons ranging from 1 to 5 GeV are studied. For the 900 pixel calorimeter also higher energies are considered. Chapter 6 studies the application of the Information-Distillation Generative Adversarial Network (ID-GAN) on the high-energy 900-pixel shower images [9].

1.1. Software Packages and Versions

This section briefly specifies the software packages and versions used for the studies in this thesis.

- GEANT4 simulation toolkit, Version 10.04.p03
- Tensorflow, version 1.13.0
- Pytorch, version 1.9.0+cu102, version 1.9.0+cu111
- Tensorboard, Versions 1.15.0, 2.4

BRIEF INTRODUCTION INTO CALORIMETERY

2

2.1. Calorimeter Resolutions

Calorimeters can be used to measure the energy of particles and play an important role in identifying particles in high-energy physics. The deposition of energy in a calorimeter can be a destructive process, where the primary particle generates numerous secondary particles through a chain of inelastic reactions from interacting with the detector material. The signals produced during these interactions, such as charge flow or scintillation light, can be used to estimate the energy of the primary particle [10]. Electromagnetic calorimeters are used to measure the energy of electrons, positrons, and photons, and hadronic calorimeters for energy measurements of hadrons. The energy of a charged particle can also be estimated by measuring its momentum in a magnetic field. The equations explained in this chapter and the main information about calorimetery are taken from [10] and [11]. The energy resolution σ_E of electromagnetic and hadronic calorimeters is approximately given as [11]:

$$\frac{\sigma_{\mathrm{E}}}{\mathrm{E}} \approx \begin{cases} \frac{2\text{–}15\%}{\sqrt{\mathrm{E/GeV}}} & \text{for} \quad \text{electromagnetic} \\ \frac{35\text{–}120\%}{\sqrt{\mathrm{E/GeV}}} & \text{for} \quad \text{hadronic} \end{cases} \tag{2.1}$$

with E denoting the energy of the primary particle.

The rest of this section briefly explains the working concept of electromagnetic calorimeters introducing homogenous calorimeters, electromagnetic and hadronic shower development, energy resolution, and the signal detection processes.

2.1.1. Electromagnetic Calorimeters

Electromagnetic calorimeters measure the energy of particles through electromagnetic showers generated during their interaction with matter. These showers have short shower depths that confines them within the calorimeter. The radiation length X_0 determines the depth of the electromagnetic shower and in effect the depth of electromagnetic calorimeter. The materials of the calorimeter are chosen so as to minimize the X_0 to contain the entire shower and energy inside. The energy deposited is proportional to the energy of the primary particle and is estimated through various signals produced in the detector material.

2.1.1.1. Homogenous Calorimeters

Homogeneous calorimeters comprise a single medium that serves the purpose of both detecting signals (active material) and absorbing energy (absorber material). One of the main advantages of these calorimeters is that they could detect the shower energy from everywhere inside the medium, and also provides the same responses from all parts of the detector, which results in a good energy resolution. The different homogeneous calorimeter classifications are semiconductor calorimeters, Cherenkov calorimeters, scintillator calorimeters, and Noble liquid calorimeters.

2.1.1.2. Electromagnetic Shower Development

Figure 2.1.: Diagrams of bremsstrahlung (left) and pair production (right) [11].

Quantum Electrodynamics explains the process of interaction of electrons and photons with matter. Figure 2.1 illustrates the diagram of bremsstrahlung (left) and pair production processes. The presence of the nucleus is depicted as Ze in the diagram. The bremsstrahlung process is the primary source of energy loss for electrons greater than around 10 MeV and photons with energy around 10 MeV generate electron-positron pairs [10]. For low energy, electrons lose energy primarily on collision with atoms and molecules, leading to ionization and thermal excitation, whereas Compton scattering and the photoelectric effect are the main source of energy loss for photons in the low energy range [10]. Figure 2.2 shows the energy loss for electrons and positrons in lead [12]. The radiation length $\mathrm{X_0}$ is one of the main parameters of electromagnetic showers and is defined as the average distance an electron has to propagate in material to possess 1/e times its initial energy [10]. $\mathrm{X_0}$ is expressed as [10]

$$\mathrm{X_0\,(g/cm^2)} \simeq \frac{716\,\mathrm{g\,cm^{-2}A}}{\mathrm{Z(Z+1)\ln(287/\sqrt{Z})}}, \tag{2.2}$$

where Z is the atomic number, and A is the atomic mass of the interacting material. Approximately $dE/dx_{(brems)} \simeq E/X_0$ [10]. The rate of energy loss for electrons via the bremsstrahlung process is almost proportional to the energy, whereas via ionization, it logarithmically varies with the energy of electron. The critical energy E_c can be defined as the energy at which the energy loss rate of electron via bremsstrahlung process and ionization process becomes equal [12] [13].

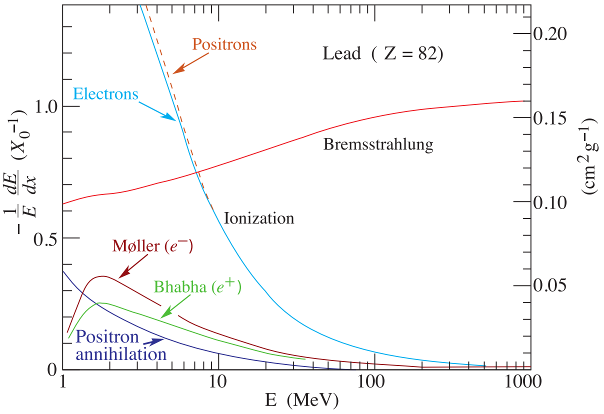


Figure 2.2.: Fractional energy loss of electrons and positrons per radiation length in lead, as a function of energy [12].

The Heitler model explains the evolution of electromagnetic showers in a simple way. According to the model, until the critical energy E_c, electrons and photons interact with matter only via bremsstrahlung and pair production, respectively [11]. After exceeding E_c, the electrons lose their energy through the ionization process. The model defines that the total energy deposited in the matter E_0, is proportional to the total number of electrons and positrons generated during the interaction, which is given as [11]

$$N_{total} \approx \frac{E_0}{E_c}, \tag{2.3}$$

and the total path length of the shower is represented as [11]

$$s_{total} \approx \frac{E_0}{E_c} X_0. \tag{2.4}$$

Figure 2.3 shows the shower development of incident particles after each radiation length X_0 for the model. The energy of the outgoing particle gets halved, and the number of outgoing particles gets doubled, at each step of X_0. For bremsstrahlung or pair production happening at each step of X_0, the energy of the particles for these interactions is given by [11]

$$\text{Bremsstrahlung} : E_e(n\,X_0) = E_\gamma(n\,X_0) = \frac{1}{2}E_e[(n-1)\,X_0], \tag{2.5}$$

$$\text{Pair Production} : E_{e^+}(n\,X_0) = E_{e^-}(n\,X_0) = \frac{1}{2}E_\gamma[(n-1)\,X_0]. \tag{2.6}$$

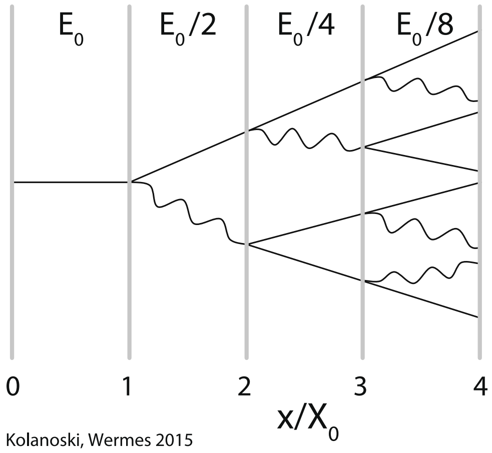


Figure 2.3.: Heitler model of shower development. After a traveled distance of X_0, the particle interacts and two new particle with half the energy of the original particle are produced [11].

Once the energy of the shower reaches the critical energy $\mathrm{E_c}$, the energy remaining is deposited without any more radiation processes. The maximal number of particles in the shower $\mathrm{N_{max}}$, and the maximal length of the shower $\mathrm{t_{max}}$, are given as [11]

$$\mathrm{N_{max}} = \frac{\mathrm{E_0}}{\mathrm{E_c}}, \tag{2.7}$$

$$\mathrm{t_{max}} = \frac{\ln \mathrm{E_0}/\mathrm{E_c}}{\ln 2}. \tag{2.8}$$

Equation (2.7) shows that $\mathrm{N_{max}}$ is linearly proportional to the $\mathrm{E_0}$, and thus can be interpreted as a measure of the energy deposited. Equation (2.8) shows that $\mathrm{t_{max}}$ logarithmically evolves with the energy, which makes it useful while dealing with high energies in the calorimeter [11].

2.1.1.3. Hadronic Showers

When hadrons traverse through matter, their energy degradation happens through a series of strong interactions with the material, resulting in hadronic showers and energy depositions. The simulation of hadronic showers is more complex than electromagnetic showers due to the diverse interactions hadrons exhibit. The shape of hadronic showers varies for different primary particle types and their travel distance is typically comparatively larger than electromagnetic showers, which is why in addition to electromagnetic calorimeters, hadronic calorimeters are used in many hermetic detectors. Hadronic calorimeters are designed such that charged hadrons deposit the majority of their energy inside them. They are often placed outside of tracking detectors and after the electromagnetic calorimeter. Due to fluctuations in the hadron shower shape and their low response, the energy resolution typically is worse for hadrons measured in a hadronic calorimeter than for leptons and photons measured in a electromagnetic calorimeter at comparable energies. In high energy cascades, the hadron interacts with the nucleus of the medium through an inelastic process and a significant amount of energy is transferred to secondary particles, which may produce additional particles through further inelastic interactions. The hadronic shower also occurs via intra-nuclear cascades and evaporation processes. During intra-nuclear cascades, the interaction between highly excited nuclear elements inside the nucleus produce particles until its energy drops below the inelastic threshold region [11]. During evaporation processes, the highly excited nucleus transmits energy through the evaporation of nucleons and nuclear fragments by processes like fission. Pions and nucleons are the predominant secondary particles produced in hadronic showers. An additional electromagnetic component is also often present due to the production of π^0 particles.

2.1.1.4. Energy Resolution

The energy resolution of an electromagnetic calorimeter can be parameterized as [11]:

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \bigoplus \frac{b}{E} \bigoplus c, \tag{2.9}$$

- Symbol $\bigoplus$ refers to the sum in quadrature
- a refers to the stochastic term, which is estimating the size of statistical fluctuations of the shower development
- b refers to the electronic noise term
- c refers to other error sources in the calorimeter, for instance mechanical errors, electronic errors or calibration errors

For homogeneous calorimeters, where the active material and passive absorber are the same, the energy deposited by a monochromatic beam of particles does not fluctuate for different events and the intrinsic energy resolution can be better in most cases than the statistical expectation [10]. The electronic noise term emerges from the signal read-out devices of the calorimeter. The noise term is less for the light signal-based calorimeters if the first step devices of the electronic chains are photosensitive devices that give a high-gain multiplication of signals. The contribution of noise is larger when the signal are charges as the first readout device would be a preamplifier. The contribution of the electronic noise term to the energy resolution increases as the energy of the primary particle decreases. The constant term is independent of the energy of the incident particle. This term emerges from the imperfections related to the geometry of the detector and instrumental readout system. The energy response from the different parts of a large detector can vary due to irregular geometry, imperfections in the detector structure and readout systems, radiation damage, etc., that leads to the constant term in energy resolution. The constant term contributions are managed at a level of 1% or less in electromagnetic calorimeters [10].

2.1.1.5. Signal Detection

Energy deposited in an electromagnetic calorimeter material by a primary particle can be estimated for example through current measurements, scintillation light measurements or

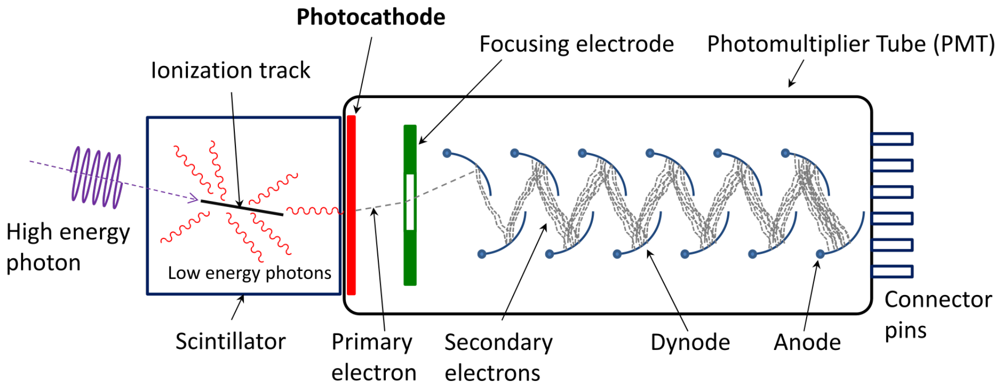


Figure 2.4.: Diagram of a photomultiplier tube. The scintillator part of a PMT produces the light which traverses through a photocathode. The resulting photoelectrons are multiplied and transmitted as current. The image is obtained from [14].

Cherenkov radiation. The intensity of the scintillation light produced in the calorimeter crystals directly provides a measure of the energy deposited by the primary particle. With for example photomultiplier tubes (PMT) or positive intrinsic negative diodes (PIN) the light yield of scintillation light can be measured.

Figure 2.4 shows the schematic diagram of a PMT: once the light leaves the calorimeter crystal, it first interacts with a photocathode. The electrons in the photocathode absorb enough energy from the light to overcome the threshold energy of the material and flow from the photocathode to the anode. Quantum efficiency, which is the measure of the effectiveness of a photocathode to produce these photoelectrons, varies depending on the material and the wavelength of the light. These electrons then enter a vacuum medium maintained at high voltage, creating a high electric field. This space comprises several dynodes and, the accelerating electrons in the vacuum medium collide with these dynodes and multiplies the photoelectrons that leave the photocathode. The accumulated charge from these collisions is transmitted as current and is proportional to the energy deposited in the crystals [15].

2.2. Belle II Electromagnetic Calorimeter

The Belle II Electromagnetic Calorimeter (ECL) is designed for the detection of photons, the determination of energy of photons and charged particles like electrons and the identification of electrons. It is also used for the detection of K_L particles along with KLM detector, event triggering and the online and offline measurement of luminosity. The ECL consists of 8736 thallium dopped caesium iodide crystals (CsI(Tl)) of 98 different shapes. Each crystal has a dimension of approximately $6 \times 6 \times 30$ cm^3. The radiation length of the crystals is around 16.1 X_0. The detector is divided into three regions: a barrel region that is 3 m long with an inner radius of 1.25 m; and two endcap regions at z = 1.96 m (forward) and z = -1.02 m (backward). The barrel part of the ECL has 6624 CsI(Tl) crystals of 29 different types (based on their dimension and shape), and the endcap regions

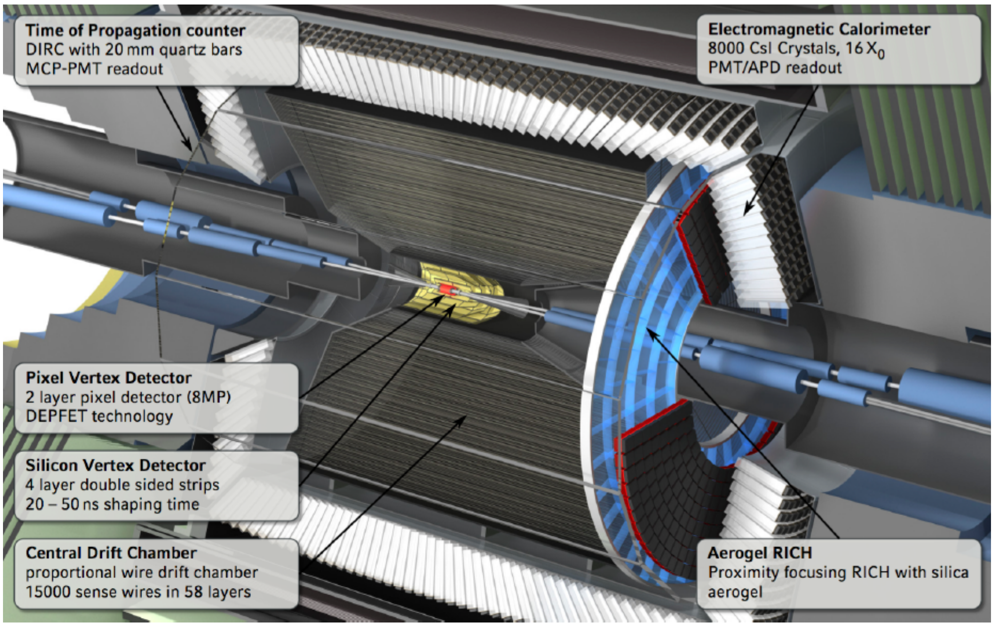


Figure 2.5.: Schematic view of the Belle II detector [16].

in total have 2112 CsI(Tl) crystals of 69 different types. A schematic view of the Belle II detector is shown in Fig. 2.5, where the location of the electromagnetic calorimeter inside the detector is indicated. The polar angle of the ECL ranges from 12.4° to 155.1°. Crystals of the ECL are wrapped with Gore-Tex porous Teflon of thickness 200 μm and covered in aluminized polyethylene of thickness 50 μm. Photodiodes are glued to the crystals to detect the scintillation light signal produced by the electromagnetic showers of particle interactions. Since Belle II is a high luminosity collider experiment, radiation damage can affect the calorimeter crystals. To handle this problem, waveform sampling and pipelined readout are utilized in the experimental setup [17]. The energy of photons ranges between a few MeV to a few GeV [17]. For photons, the energy resolution of the calorimeter changes from 2.5% at 100 MeV to 1.7% at 5 GeV [18]. The Belle II detector does not contain a hadronic calorimeter. Hence, the ECL is used in combination with other detectors to identify charged hadrons based on their shower signature and energy depositions.

2.2.1. CsI(Tl) crystals

When particles pass through a calorimeter, they interact with the material via various processes and deposit energy. Scintillators are materials that emit light after absorbing energy. The intensity of the light emitted is proportional to the energy deposited in the scintillator and therefore allows an energy measurement. Scintillator materials are characterized by their features like light yield and scintillation decay time. The light yield is defined as the number of photons produced per unit of energy deposited, and the decay time is the time taken by the scintillator material to emit $1 - e^{-1}$ of the light converted from the energy deposited [15]. Fast scintillation decay time helps in determining the time of the particle when it passes through the scintillator material. Experiments like BaBar, Belle II, CLEO, and KTeV use CsI crystals that have a short radiation length [10] and also provide a good pulse shape discrimination for different particle energies. Doping of pure CsI crystals with a small amount of Tl (dopant), gives a significantly higher light yield. The Belle II and the BaBar experiments focus on the highly efficient reconstruction of low energy photons and π^0 particles originating from the decay of B-mesons. To achieve a good signal-to-background ratio for these decays, the detector needs to have good energy resolution. The CsI(Tl) crystals with their good resolution serve this purpose.

2.3. Geant4 Simulation

The Geant4 [19] [20] [21] simulation framework is used to model the development of particle showers and their propagation when they interact with detector materials. The framework provides various packages and classes to simulate the particle interaction process. Accuracy of the simulation depends on building an exact geometry of the required detector, correctly defining the required materials and physics processes and setting the right initial conditions like particle type, momentum, and energy.

The detector geometry is constructed using the Geometry package [22], which contains three main volumes — solid, logical, and physical. Each volume is properly placed to avoid overlapping errors. This package also offers various materials to be chosen from. The Detector package handles information related to the sensitive part of the detector like energy deposited in the material, time and position of the particle interaction. A hit is an object that holds the sensitive detector information in Geant4. The Primary Particle package handles the properties of primary particles like type, energy, position, through classes like G4ParticleGun and GeneralParticleSource [22]. Tracking particle interactions is an essential process in Geant4. The separation between two consecutive interaction points of a particle is called a step. A sequence of such steps forms the *track* of a particle. Each step stores information like kinetic energies and physics processes.

Tracking every secondary particle of an entire decay cascade caused by a primary particle through a calorimeter element, can be highly CPU intensive as interactions like Bremsstrahlung do generate a large amount of secondary particles.

To overcome this problem, Geant4 stops tracking particles beyond a user-defined range called *range cut*. Thus an optimized range cut value will stop the production of certain secondary particles leading to a trade-off between the accuracy of the simulation and its simulation time [22].

2.4. Geant4 Standalone Setup

At Belle II, electromagnetic showers of electrons and photons are mostly contained in 5×5 ECL crystals. In this thesis, we thus focus on the fast simulation of such a crystal array. To obtain a training sample for the various fast simulation methods, a standalone simulation of 5×5 array is implemented using Geant4. The development of the Geant4 standalone setup is inspired from the examples of [23] [24]. Each crystal has a fixed shape and is of dimension 6 cm × 6 cm × 30 cm resulting in a crystal array with a dimension of 30 cm × 30 cm × 30 cm. The crystals are simulated as CsI(Tl) material. Unlike the

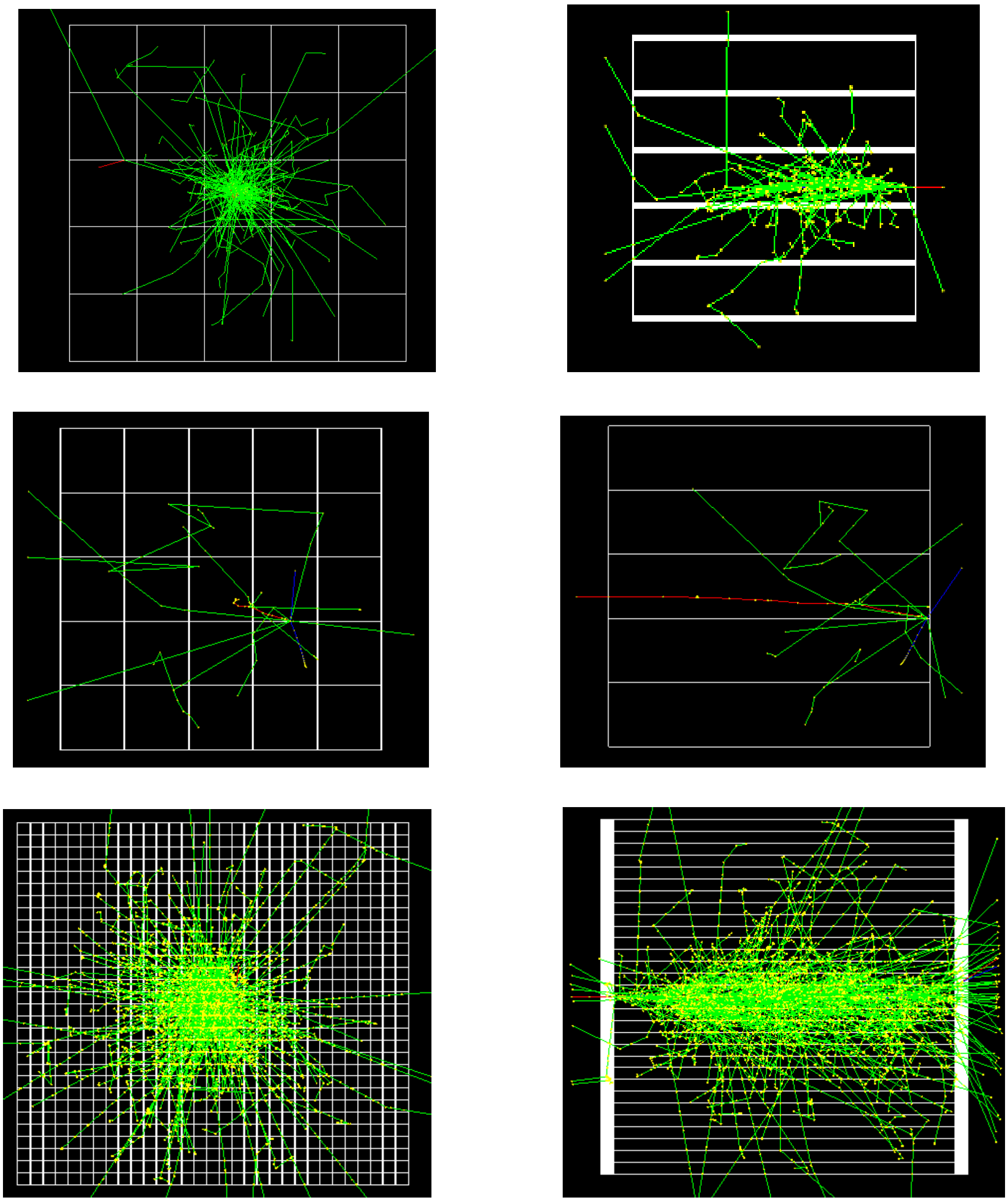


Figure 2.6.: Front view and side view of the simulated shower for 0.5 GeV primary electron in 5 × 5 crystal array (top), 0.5 GeV primary pion in 5 × 5 crystal array (middle), and 5 GeV primary electron in 30 × 30 crystal array (bottom).

actual Belle II ECL, where the size of the crystal front varies between 44.5 mm to 70.8 mm and differs from the size of the crystal rear [25], the standalone setup has a fixed crystal size. The ECL is immersed in the magnetic field of the Belle II solenoid, what impacts the energy deposition of electron showers. This is absent in the Geant4 standalone simulation. The Belle II ECL is also affected by beam and luminosity background and electronic noise. Such effects are also not incorporated in the standalone Geant4 simulation.

For the data set used to train the various fast simulation methods, the energy deposited in

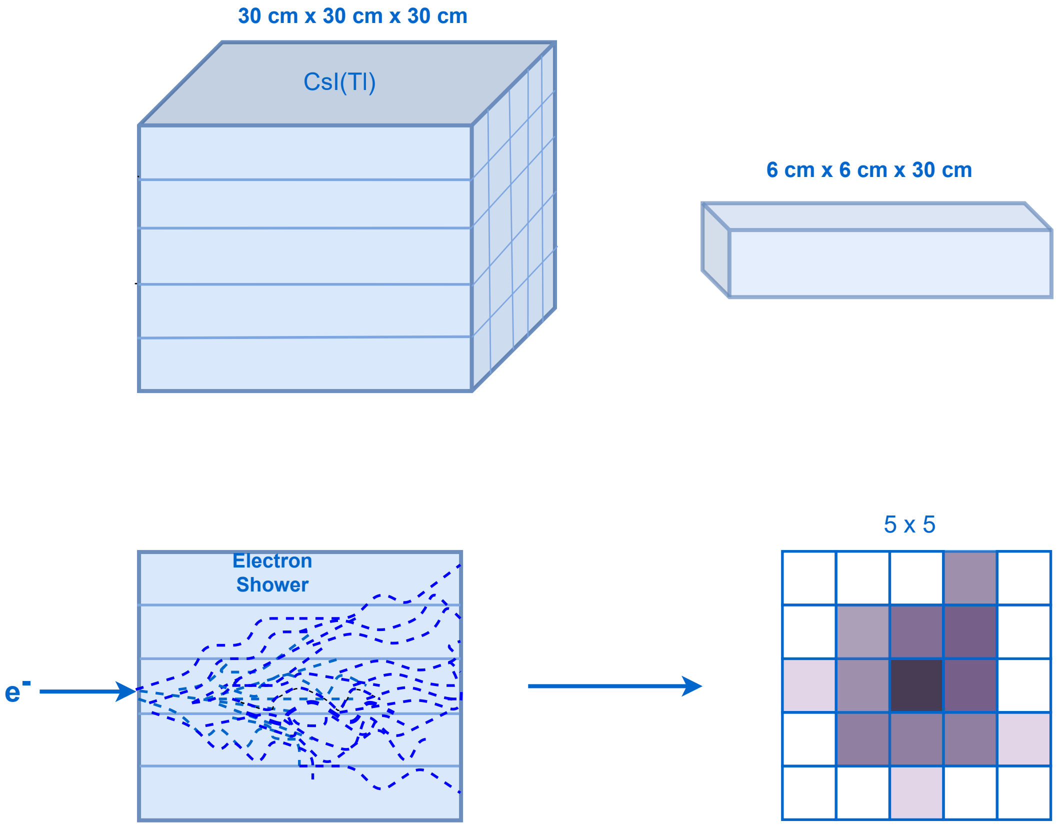


Figure 2.7.: Pictorial representation of an electron showers in a 5 × 5 crystal array and conversion of the information of the shower into training data.

each of the crystals is recorded. Other aspects, such as the propagation of light produced in the crystals and the detection via PIN photodiodes are not simulated.

Primary electrons of energies varying from 0.5 GeV to 2.5 GeV in discrete steps of 0.5 GeV are simulated and are used as the training data set for the algorithms described in Chapter 3 and 4. Due to the absence of a magnetic field in the simulation, electrons deposit the majority of their energy in the crystal array. An additional dataset with charged pions with energies ranging from 0.5 GeV to 1.5 GeV is simulated to study hadronic showers in the crystall array.

For the studies presented in Chapter 5 a more granular crystal array is implemented of the same dimensions: a 30 x 30 pixel array is used to emulate a high granularity calorimeter. The energy range is extended to 5 GeV to increase the deposited energy of individual crystals. Figure 2.6 shows the simulated shower for a 0.5 GeV primary electron in the 5 × 5 crystal array (top), a 0.5 GeV primary pion in the 5 × 5 crystal array (middle) and a 5 GeV primary electron in the 30 × 30 crystal array (bottom).

The process behind generating training datasets for the fast simulation algorithms is

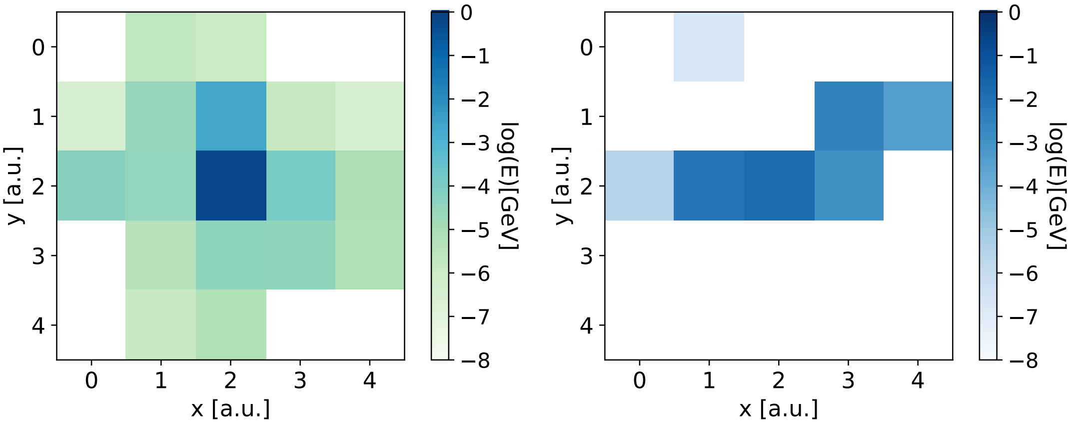


Figure 2.8.: Image of an electron of energy 1 GeV (left) and a pion of energy 0.5 GeV (right).

illustrated in Figure 2.7 for primary electrons: the primary particles are simulated using the *G4ParticleGun* class of Geant4. The impact position is smeared around the center of the crystal array using a Gaussian distribution with standard deviations of 3 cm in the x and y directions, respectively. A typical primary electron in the simulated range deposits more than 90% of its energy into the crystal array. Geant4 provides detailed information about the simulated process, however, only the deposited energy of each crystal is used to build a training data image. Examples for a primary electron and a primary muon training event are shown in Figure 2.8. The figure displays a 3 dimensional histogram, and stores for each crystal in the array the deposisted energy. As the simulated histograms are similar to images, both terms will be used to describe them. Similarly, pixels and crystals will be used interchangably.

WASSERSTEIN GENERATIVE ADVERSARIAL NETWORK

3

3.1. Deep learning Generative Models

Generative models are feed-forward neural networks that generate new samples from noise vectors like Gaussian or uniform distributions by updating model parameters via training. This has become a robust method in machine learning. The model is trained to learn the probability distribution of the training samples through a set of loss functions which help the distribution of training samples and the generated samples to come closer. As a result, a random noise vector is transformed into a meaningful representation and new samples that match the training samples are generated.

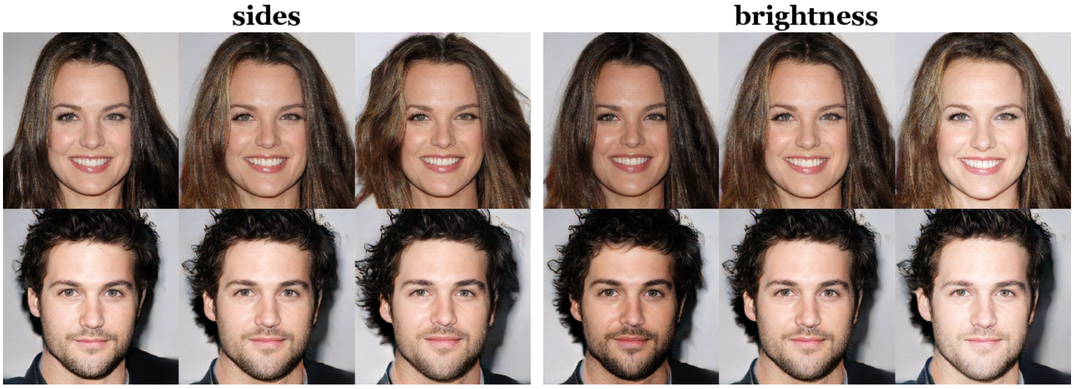


Figure 3.1.: The images of human faces generated by ID-GAN trained on the CelebA-HQ data set [9]. Features of the human faces like hair and brightness are controlled in the images.

Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) are the two most powerful generative models commonly used today. The following sections briefly describe about WGAN and VAE models that are used in a wide range of applications in various domains. Generation of the pictures of non-existing people [26], cartoon characters, animated movies, novel music audios, are some of the most common applications of GAN and VAE. They are likewise employed in science, for the synthesis of chemical compounds and molecules [27]. Detecting a medical tumor in health care [28], which can lead to a significant cost reduction for such procedures, is one of the few advanced applications of the generative models in daily life. The recent advancements in machine learning have empowered the model to produce images and samples with very high resolution and accuracy [9]. Information Distillation Generative Adversrial Network (ID-GAN) is an improved model of GAN. Figure 3.1 shows the high-quality images generated by ID-GAN trained on the CelebA-HQ dataset [9]. Thus generative models are now experimented with and explored in many fields to generate new samples of interest reducing manual work, cost, and time required for the processes.

3.2. Generative Adversarial Network (GAN)

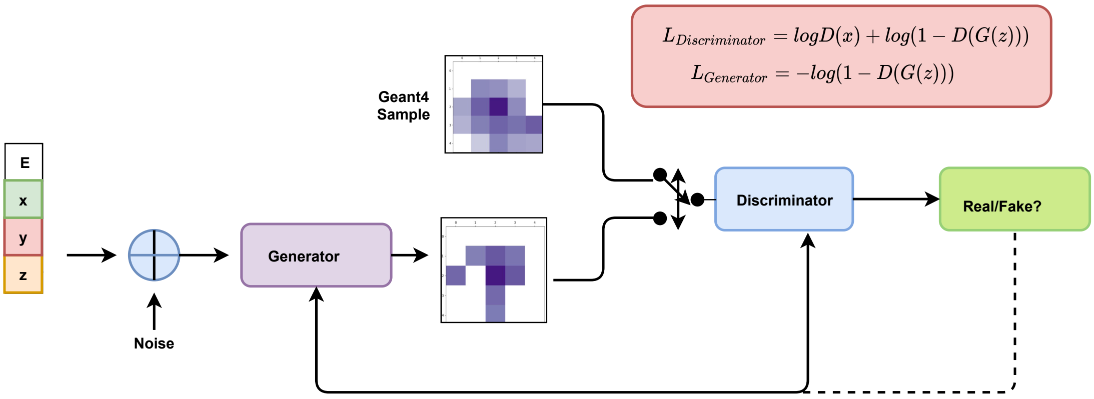


Figure 3.2.: Diagram of GAN architecture. It includes a Generator network and a Discriminator network. E, x, y, and z represent the conditioning labels. Both the networks are trained on the loss functions depicted in the diagram.

A generative adversarial network is a model that combines a generator and a discriminator network, trained separately and competing against each other. Figure 3.2 illustrates the different components of the GAN network and the loss functions they are trained with. The generator is assigned to generate fake images from random Gaussian vectors and the conditioning labels representing the true values of the images. The discriminator is trained to differentiate real and fake images. Since the generator requires the discriminator to classify the generated images as 'real,' it minimizes a cost function. Thus the objective of the generator is given as [29]

$$\mathrm{Loss_G} = -\,\mathbb{E}_{\mathrm{z}\sim \mathrm{p_z(z)}} \log(1 - \mathrm{D(G(z))}). \tag{3.1}$$

Here G refers to the generator, D the discriminator, and z the random noise vector.

- The term G(z) refers to the output of the generator with the noise vector z as the input
- D(G(z)) refers to the output of the discriminator with the generated images as the input.
- $\mathbb{E}_z$ is the expected value over all the random noise vectors given to the generator.
- $p_z(z)$ is the defined prior for input noise vector.

The discriminator maximizes the sum of log-likelihood functions of the discriminator's probability outcomes so as not to be fooled by the generator. The corresponding loss function of the discriminator is given by [29]:

$$\mathrm{Loss_D} = \mathbb{E}_{\mathrm{x}\sim \mathrm{p_{data}(x)}} \log \mathrm{D(x)} + \mathbb{E}_{\mathrm{z}\sim \mathrm{p_z(z)}} \log(1 - \mathrm{D(G(z))}) \tag{3.2}$$

where

- x is the real image.
- D(x) is the output of the discriminator with real images as the input.
- D(G(z)) is the output of the discriminator with the generated images as the input.
- $\mathbb{E}_x$ is the expected value over all the data samples used for training.

If the true and generated distributions are far apart, the discriminator could effortlessly differentiate them. But during their training, the generator gets better, and tries to deceive the discriminator by synthesizing images that closely resemble the real datasets. In cases when the generator produces perfect images, the probability of real and fake outputs are equal. So eventually, during each step of the training, the generator model weights are updated to minimize the loss function (Eqn.(3.1)), and the discriminator network weights are updated to maximize the loss function (Eqn.(3.2)). This min-max game between the generator and the discriminator defines the term 'Adversarial Networks.' Numerous issues encountered while training the GAN network are discussed below.

- **Mode Collapse:** The generator is expected to generate diverse images from random noises. But when the discriminator gets stuck in a local minimum, it is easy for the generator to find a credible fake image that would fool the discriminator. The generator could also easily reproduce the same image repeatedly, and the discriminator may never learn to get out of this local minimum. This issue of GAN is identified as mode collapse, where only similar-looking images are produced out of the generator instead of generating diverse images.
- **Vanishing Gradient:** When the discriminator is extremely good, it never allows the generator to learn and improve, due to diminished gradients and thus leading to unlearned generators.
- **Convergence Problem:** GANS usually suffers from the non-convergence of the model. The model parameters oscillate and never converge.
- **Balance between Models:** Balance between the generator and discriminator is challenging to achieve and the models are susceptible to hyperparameters.

3.3. Wasserstein Generative Adversarial Network

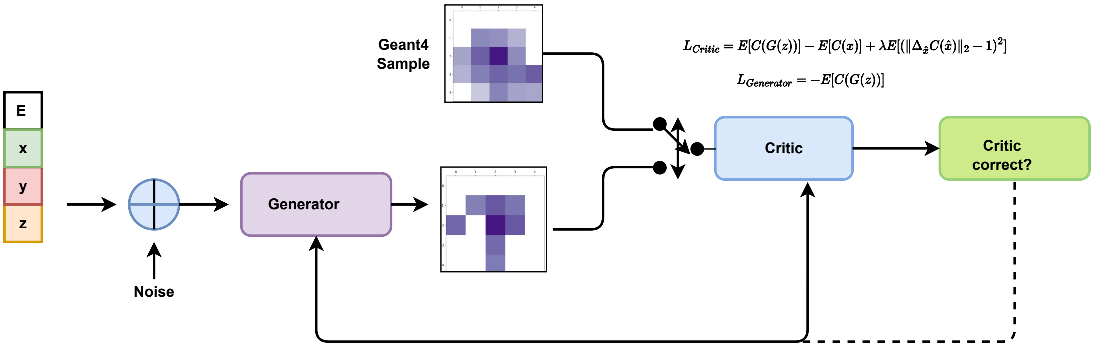


Figure 3.3.: Diagram of WGAN architecture. It contains a Generator network and a Critic network. The model is trained on the loss function calculated based on the Wasserstein distance and an additional gradient penalty term.

The Wasserstein Generative Adversarial Network (WGAN) was developed as an improvement to the GAN. This model is trained with a different loss function that applies the Wasserstein distance. In Ref. [30] the following definition was given: "It is also called Earth Mover's distance, short for EM distance, because informally it can be interpreted as the minimum energy cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution". Given that p_r is the distribution of data over real sample x and p_g is the distribution of generator over data x, the Wasserstein distance between p_r and p_g is expressed as [30]:

$$\mathrm{W(p_r, p_g)} = \inf_{\gamma \sim \Pi(\mathrm{p_r}, \mathrm{p_g})} \mathbb{E}_{(\mathrm{x,y})\sim\gamma}[\| \mathrm{x} - \mathrm{y} \|] \tag{3.3}$$

where the set of possible joint probability distributions between p_r and p_g is given by $\Pi(p_r, p_g)$. The *inf* (infimum) shows that the smallest among the cost are chosen. $\| x - y \|$ refers to the distance between the starting point x and the destination point y [30]. The WGAN network uses a critic network instead of the discriminator network in the GAN. The critic performs the same way as the discriminator in the GAN, except the output of the critic is no longer the probability of being real or fake images; instead, it provides a score for real and fake images. The loss functions of the WGAN are given by [31]:

$$\mathrm{Loss_C} = \mathbb{E}[\mathrm{C(G(z))}] - \mathbb{E}[\mathrm{C(x)}] + \lambda\mathbb{E}[(\| \nabla_{\hat{\mathrm{x}}}\mathrm{C}(\hat{\mathrm{x}}) \|_2 - 1)^2] \tag{3.4}$$

and the objective of the generator is:

$$\mathrm{Loss_G} = -\mathbb{E}[\mathrm{C(G(z))}] \tag{3.5}$$

where

- C(x) refers to the output of the critic from real images.
- C(G(z)) refers to the output of the critic from generated images.

- The term $\hat{x}$ refers to a mixture of training images and generated images.
- $\lambda\mathbb{E}[(\| \nabla_{\hat{x}}C(\hat{x}) \|_2 - 1)^2]$ refers to the gradient penalty term of the loss function.

The critic loss function increases the difference between real and generated image scores, and these scores are calculated over a batch of images in an epoch. Figure 3.3 shows the different components of the WGAN network and the loss function with which they are trained. Consequently, the new loss function computes the Wasserstein distance between the distribution of the training dataset and the generated dataset. Equation (3.5) includes the gradient penalty term, which ensures the 1-Lipshitz continuity enforcement in the critic during training.

In conclusion, the WGAN model can improve the stability of the training process compared to the GAN. The smoother gradient of the loss function prevents vanishing gradients and mode collapse[32].

3.4. Training Geant4 Standalone Setup Electron Showers

The following work on the fast simulation of electron showers in Geant4 standalone crystal setup using the WGAN model is inspired from [5] [33], which explores the fast simulation of electron shower responses in a configuration of High Granularity Calorimeter (HGCAL) prototype of the CMS experiment [34]. The HGCAL prototype has seven layers resulting in 2.8 – 16.2 radiation lengths (X_0). The images of the training datasets are constructed in 12 × 15 × 7 pixels. The total number of pixels in the HGCAL image datasets is 1260, whereas, for the case of standalone Geant4 setup, a two-dimensional array of 5 × 5 crystals is studied. In the case of HGCAL, the Geant4 simulation toolkit is used to produce electron shower simulations, consisting of electron showers of energy 20 GeV, 32 GeV, 50 GeV, 80 GeV, and 90 GeV discrete energies. For the evaluation process, 70 GeV electron showers is also simulated. But for the case of standalone setup, the discrete energies of electrons 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV and 2.5 GeV is used, which is a comparatively low energy range.

This section probes the fast simulation of electron showers simulated from the Geant4 standalone crystal setup using the WGAN model. The shower deposited in the 5 × 5 pixel is studied for both WGAN and Geant4 simulation. The section also looks briefly into the performance of the energy regressor network and the position regressor network used to constrain the labels of the generated images. To evaluate the performance of the model, various physics observables for the shower deposited by both approaches are compared. The performance of the model to interpolate a dataset (generate a shower response for untrained energy within the training energy range) is inspected and its agreement is also analyzed.

Thus main objective of this section is to:

- Build a WGAN model for the images of electron showers deposited in the Geant4 5x5 crystal block;
- Train the electron shower datasets of energies 0.5 GeV, 1 GeV, 1.5 GeV and 2.5 GeV using this WGAN model.
- Interpolate particle showers of 2 GeV electrons using the trained model.

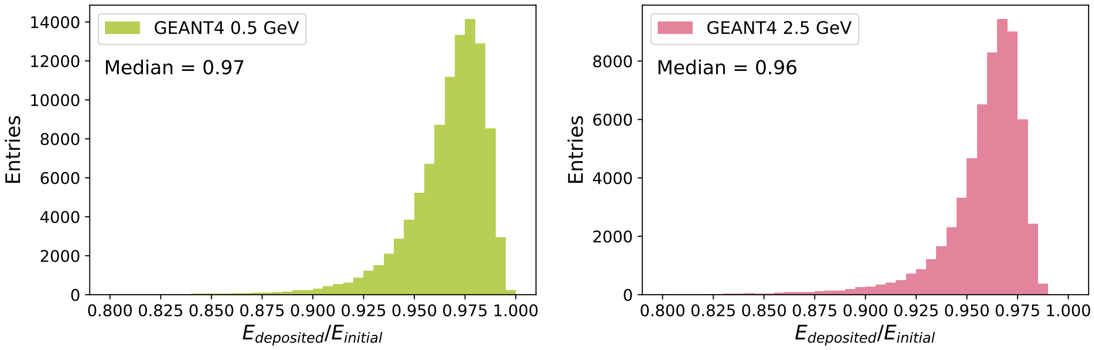


Figure 3.4.: The ratio of electron energy deposited in the crystal block with respect to the incident electron energy for 0.5 GeV (left) and 1 GeV (right).

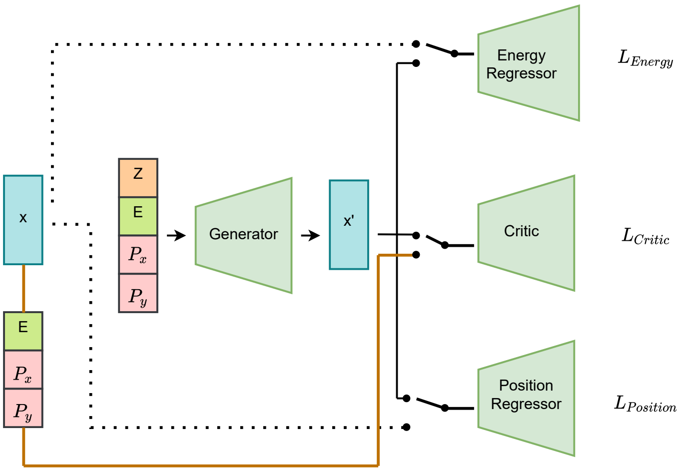


Figure 3.5.: Diagram of a WGAN model combined with an energy regressor network and a position regressor network. The labels representing the true values of the primary particle energy E and position components $\mathrm{P_x}$ and $\mathrm{P_y}$ are given to the generator and critic in addition to the images.

- Discuss the correlation between Geant4 and WGAN-generated images.

The model is trained on 100000 datasets of each energy. A batch size of 250 images is chosen for 300 epochs. Evaluation of the model is performed using 25000 data samples, 5000 from each discrete energy. The pixels with energy less than 1×10^{-3} GeV are substituted as zero energy pixels. The random noise vectors are sampled from a uniform distribution U (-1, 1). The details of the architectures of the optimized networks used to train the model for 5 $\times$ 5 pixel images of electrons are given in Appendix A Tab.A.1, Tab.A.2, and Tab.A.3. Figure 3.4 shows the ratio of energy deposited in the crystal block to the energy of incident electrons, for 0.5 GeV and 2.5 GeV. The energy less than 1×10^{-3} GeV deposited in the crystal block are completely ignored here. It is observed that most of the electron energy is deposited in the crystal itself. The median of the left plot is around 97% and the right plot is around 96%. Figure 3.5 illustrates the configuration of the model, which is trained on the electron data set to reproduce the images of the Geant4 electromagnetic shower simulation. The WGAN model is merged with a position regressor network and an energy regressor model to constrain the energy and position label information in the generated images. The critic network and the regressor networks are trained a few times before training the generator each time. More details of the regressor networks are described in Section 3.4.1 and Section 3.4.2. The development of the model architecture and the training dynamics is inspired from [33]. The model is implemented and trained using Tensorflow, version 1.13.0.

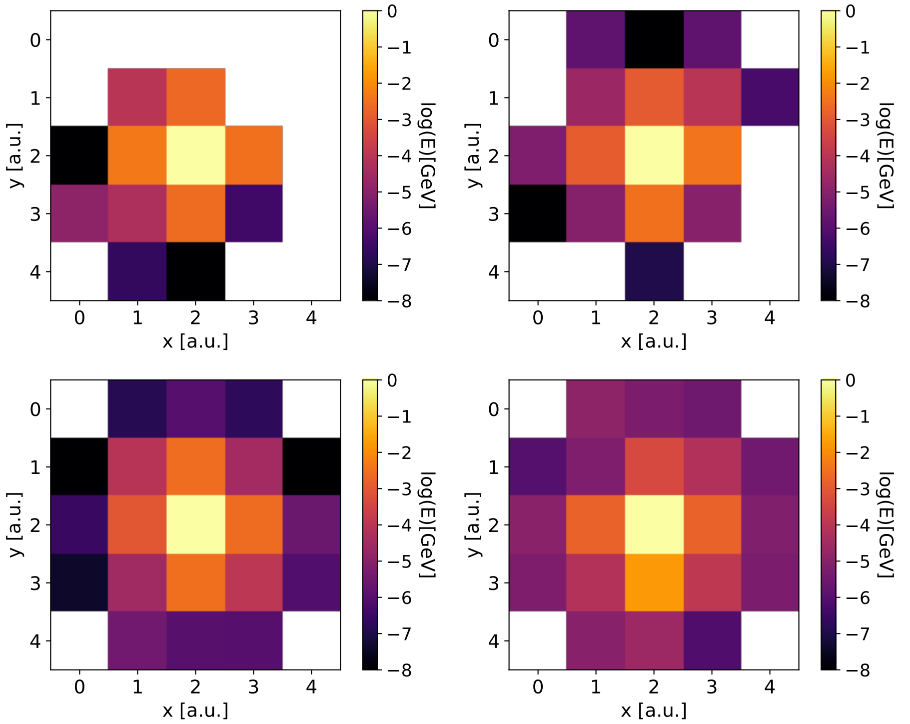


Figure 3.6.: Evolution of shower occupancy patterns in the crystals at different epochs. The plot represents the average of 50 samples of WGAN generated images in epochs 3, 50, 100, 150 (clockwise). As the epoch increase (clockwise), the pixel occupancy gets more prominent. The energy is expressed in a logarithmic scale.

Figure 3.6 shows the average over the 50 samples of WGAN generated images of electrons for different epochs. The energy is represented in the logarithmic scale. Epoch three has fewer occupied pixels for energy reconstruction, and as the epoch increases (clockwise), the pixel occupancy increases. The color of the pixels represents the logarithmic value of deposited energy. Figure 3.7 compares example WGAN generated images (right column) after the final epoch with Geant4 example images (left column) for primary electrons for different energies. The comparison indicates a difference in cell occupancy between the fast and full simulation. As substantiated later, the pixel count of WGAN images is always less than their Geant4 counterpart. This is a limitation of the WGAN simulation.

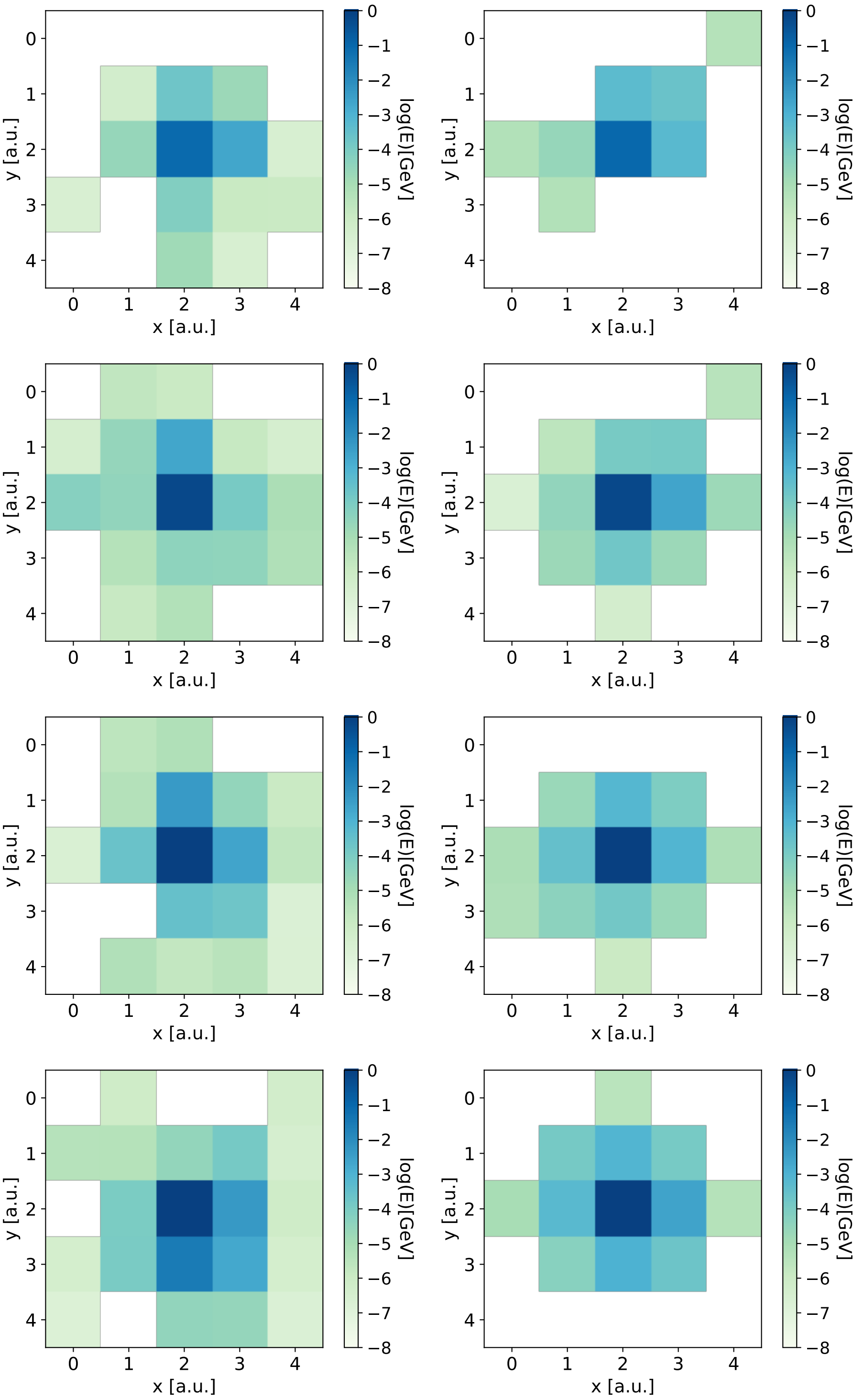


Figure 3.7.: Energy depositions in 5 ×5 crystals Geant4 (left column) and WGAN generated (right column) images of electrons for energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV (top to bottom).

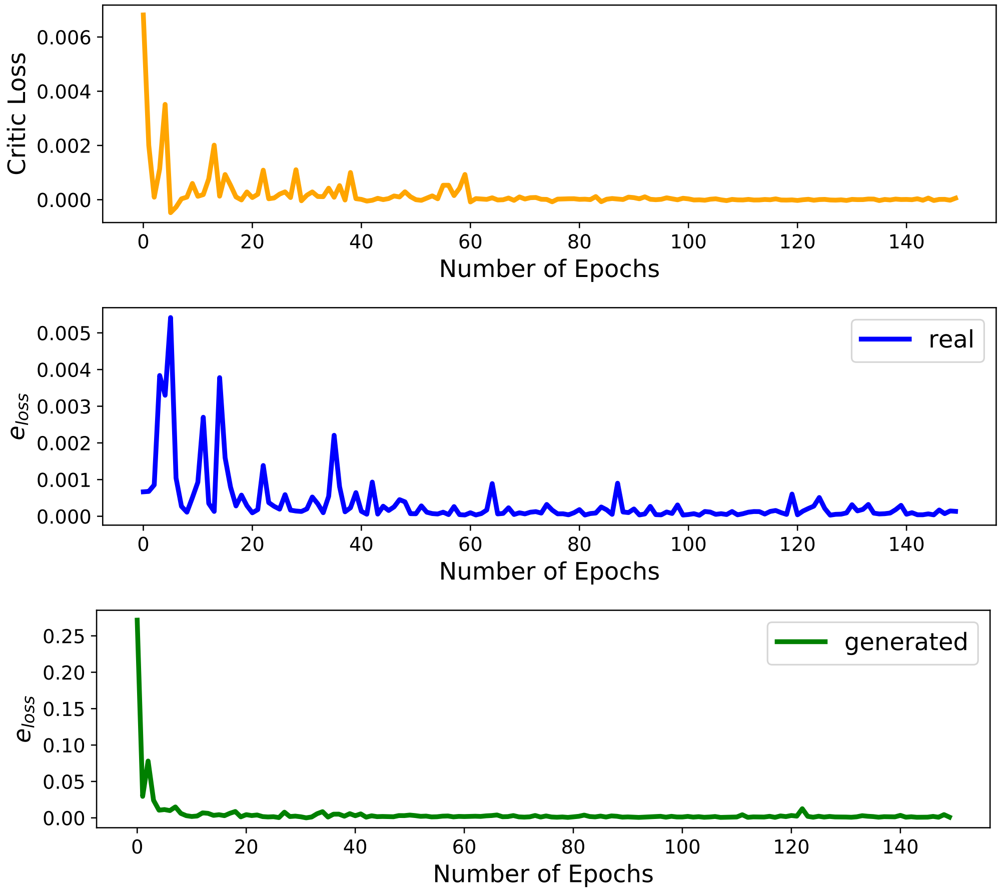


Figure 3.8.: Loss curves for critic network (top) calculated based on Eqn.(3.5), energy regressor network for real/Geant4 images (middle), and energy regressor network for the generated images (bottom).

Another observation is that in the fast simulation the inner pixels of the 5 × 5 array are better reproduced than the outer pixels. This can be attributed to the fact that the inner pixels contain higher energies and the outer ones have lower energies deposited in them. The WGAN seems to have difficulties to model the lower energies.

Figure 3.8 depicts the loss curves of different networks during training. The critic loss (top) converges as the epoch increases. The learning rate for critic for the first 60 epochs is 5×10^{-4} and, then it is reduced to 2×10^{-4} after 60, 1×10^{-4} after 80, and 5×10^{-5} after 100 epochs. The weight of gradient penalty term λ is set as 5. The learning rate for the generator for the first 70 epochs is 1×10^{-3} and then it is reduced to 5×10^{-4}, 2×10^{-4} and 1×10^{-4}, after 70, 90 and 100 epochs respectively [5]. An energy regressor network is added to the WGAN network during training which constrain the energy label information (true values of energy of the primary particle) in the generated images. The plots in the center and the bottom show the loss function graph of the energy regressor network for real/Geant4 simulated images (center) and WGAN generated images (bottom). They converge as the epoch increases, but also an oscillatory behaviour is observed for the loss function of the energy regressor for the Geant4 images. The optimizer used for the training is RMSPropOptimizer. The model is trained around 250 epochs and the results from the best epoch is chosen.

3.4.1. Energy Regressor Network

The true energies of primary particles during the Geant4 simulation are used as labels to condition the generator and the critic [35]. Additionally, an energy regressor model [5] is trained to predict these energy labels of Geant4 simulated and WGAN generated shower images. The energy regressor back propagates the loss function from the Geant4 image for training the energy regressor network. The loss function for the $\mathrm{i^{th}}$ real (Geant4) image based on mean squared error is given as [5]:

$$\mathrm{L_{real,i}} = [\mathrm{ER}(\mathrm{x_i}) - \mathrm{E_i}]^2 \tag{3.6}$$

where $\mathrm{ER}(\mathrm{x_i})$ refers to the output label of the energy regressor when real image $\mathrm{x_i}$ is given

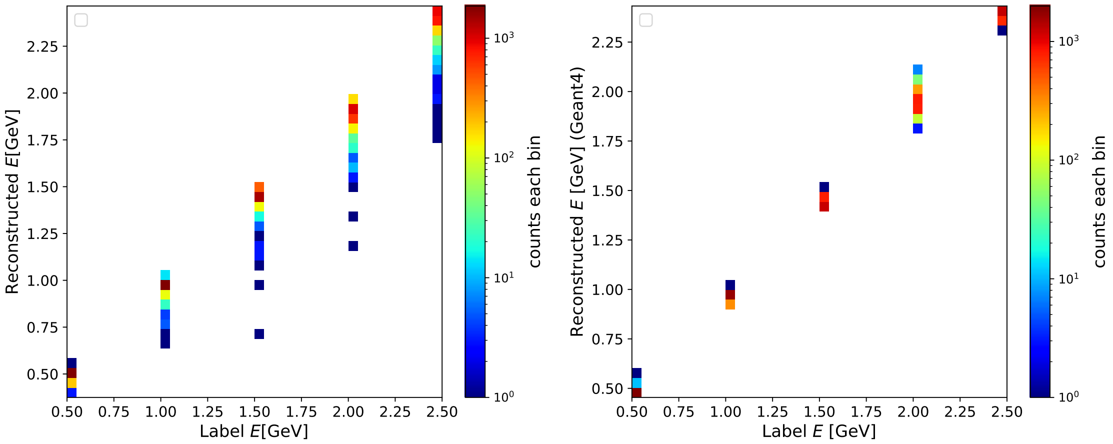


Figure 3.9.: The Energy labels are reconstructed by the energy regressor for the simulated (left) and generated (right) images.

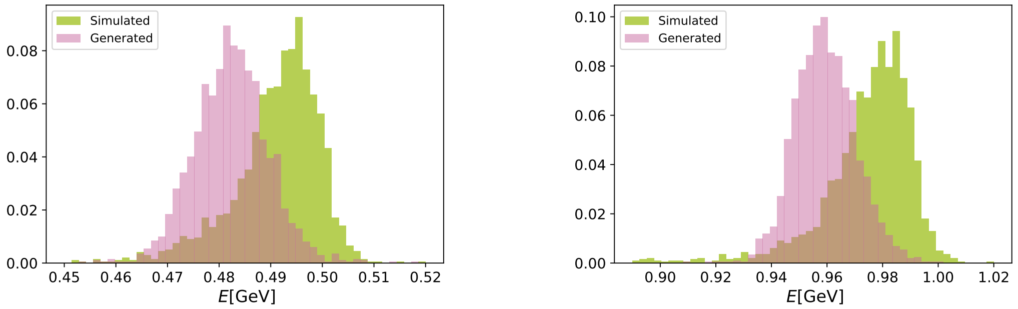
and $\mathrm{E_i}$ refers to the true energy label of the image $\mathrm{x_i}$. The loss function for the fake image based on mean squared error is given as:

$$\mathrm{L_{fake,i}} = [\mathrm{ER}(\mathrm{G}(\mathrm{z_i}, \mathrm{E_i}, \mathrm{P_i})) - \mathrm{E_i}]^2 \tag{3.7}$$

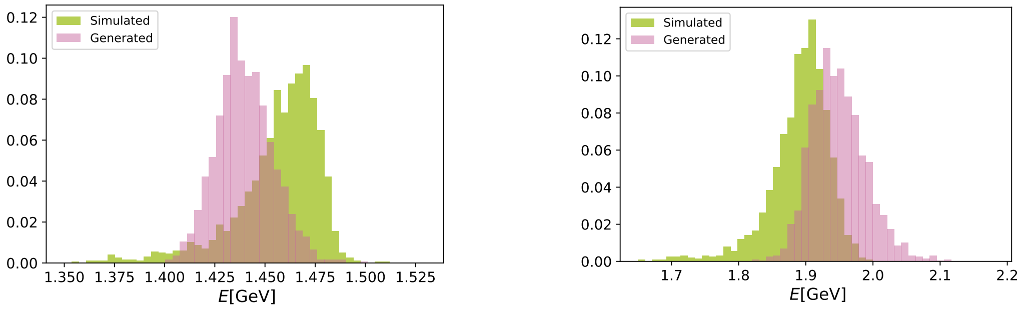
where $\mathrm{ER}(\mathrm{G}(\mathrm{z_i}, \mathrm{E_i}, \mathrm{P_i}))$ refers to the output of the energy regressor when the fake image is given. The letters z, E, P refers to the noise vector, energy label and position label respectively. To constrain the energies of the images generated, the loss function of the generator network is extended by [5].

$$\mathrm{L_{aux}} = \sum_{\mathrm{i}}^{\mathrm{n}} |\mathrm{L_{real,i}} - \mathrm{L_{fake,i}}|.$$

Thus the generator of the model is trained on the Wasserstein distance loss and additional regressor network loss. The loss part to the generator from the regressor network is scaled



(a) 0.5 GeV (b) 1 GeV



(c) 1.5 GeV (d) 2 GeV

Figure 3.10.: The distribution of the energy labels reconstructed by the energy regressor for simulated and generated images for energies 0.5 GeV, 1 GeV, 1.5 GeV and 2 GeV.

in this case with a hyperparameter $\kappa_E = 0.0425$ and a learning rate of 5×10^{-5} is used for training the regressor network. The energy regressor network is only trained for the first 50 epochs. Figure 3.9 shows the reconstruction of energy labels by the energy regressor for Geant4 (left) and WGAN generated (right) images. The x-axis represents the true energy labels of the images, and the y-axis represents the labels reconstructed by the energy regressor. The z-axis represents the number of counts in each bin. It is observed from the plots that the energy regressor performs better for generated images than Geant4 images. Figure 3.10 shows the distribution of energy labels predicted for the Geant4 and WGAN images by the energy regressor for images of 0.5 GeV, 1 GeV, 1.5 GeV, and 2 GeV electron showers. The figures show that the distributions of reconstructed energy labels of Geant4 images and the WGAN-generated images do not overlap entirely.

3.4.2. Position Regressor Network

During the Geant4 simulation, the position of the particle gun is changed randomly between -30 mm and $+30$ mm from the center along the x and y- axes, to obtain a random pattern of energy deposition in the 5×5 crystals (Section 2.4). A major portion of the energy gets deposited into the center crystal. A position regressor network is added to the WGAN model to reconstruct this variation in position. The loss function of the position regressor network is similar to the energy regressor network discussed above (Section 3.4.1). The

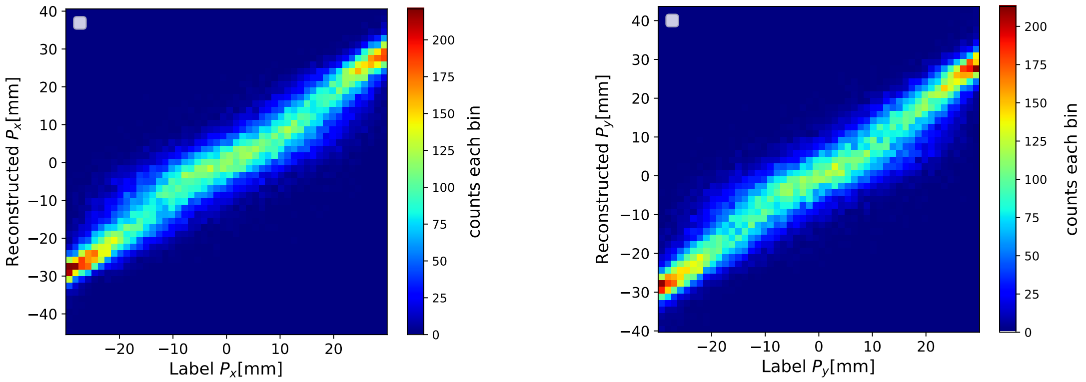


Figure 3.11.: The x (left) and y (right) positions reconstructed by the position regressor for the Geant4 images.

regressor is trained with a loss function calculated from the output of the network when real (Geant4) images are given. The loss function calculated from the position regressor output of fake images is used to train the generator network which helps in the generation of position constrained images. Since the crystal size of the Geant4 setup is large, i.e., 6 cm x 6 cm, it becomes challenging for the position regressor to reconstruct the position of the particle gun.

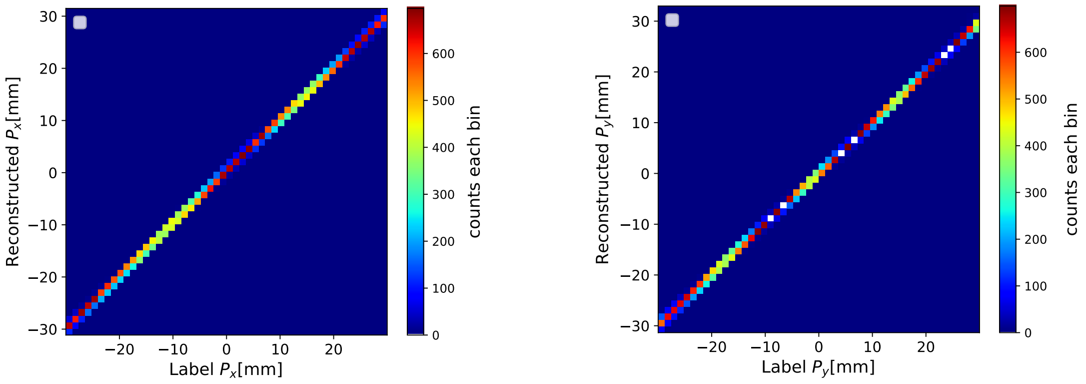


Figure 3.12.: The x (left) and y (right) positions reconstructed by the position regressor for the generated images.

Figure 3.11 shows the position reconstruction of Geant4 images by the position regressor. The x-axis represents the true position labels of the particle gun (P_x and P_y), and the y-axis represents the position labels reconstructed by the position regressor. The z-axis represents the number of counts in each bin. The output of the position regressor from real images shows a histogram with an 's' shape in the middle, and the resolution is worse in the middle caused by the large crystal size [36]. The resolution is much better towards the crystal edges, where the crystals share the energies. Figure 3.12 shows the position (P_x and P_y) reconstruction of WGAN generated images by the position regressor. It is observed that the position of fake images is reconstructed with much better resolution than the actual images, which is not desirable. This happens because the generator repeatedly produces a pattern of images to obtain a minimum penalty from the position regressor. So it can be concluded that the position regressor does not reconstruct the true position labels for 5 × 5 crystals with large crystal size. Due to these reasons, the position regressor is removed in the final training, and the WGAN model is trained with just an additional energy regressor network to constrain the energy labels.

3.4.3. Geant4 and WGAN Shower Comparison

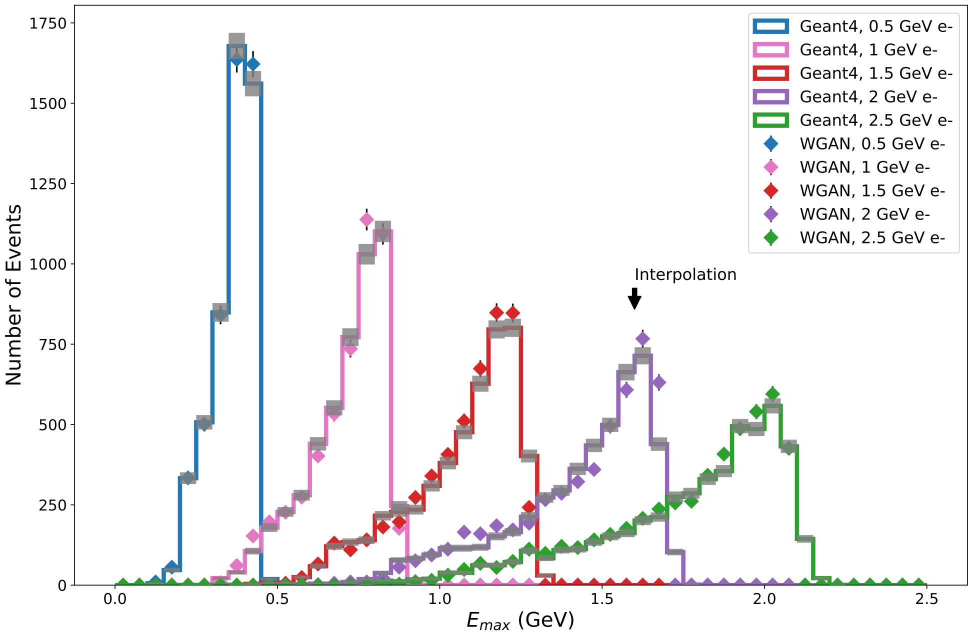


Figure 3.13.: The distribution of E_{max}, the maximum pixel energy for 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV. The images of 2 GeV electrons are interpolated by the model. For the interpolated case, the model produces the expected distribution, which shows that it can generalize this aspect of the electron shower.

Table 3.1.: The p-value calculated using the K-S test corresponding to different energies for different observables.

Observable	0.5 GeV	1 GeV	1.5 GeV	2 GeV	2.5 GeV
E_{max}	0.067	3×10^{-5}	4×10^{-4}	0.078	1×10^{-4}
E_1/E_9	2×10^{-4}	0.02	0.19	3×10^{-61}	3×10^{-5}
Energy active pixels	0.99	0.98	0.10	2×10^{-13}	0.16

Various physics observables are compared for 25000 Geant4 and WGAN electron showers

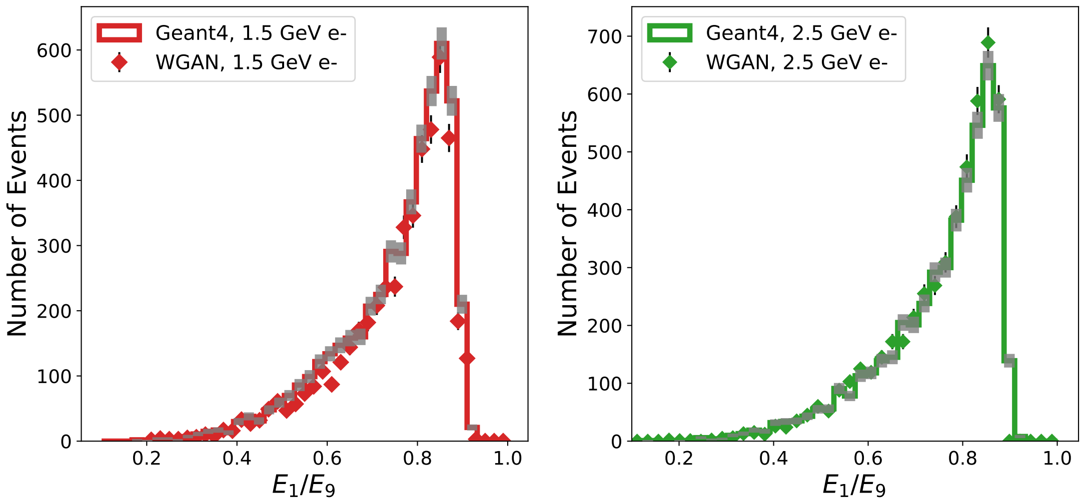


Figure 3.14.: Distribution of E_1/E_9 for 1.5 GeV and 2.5 GeV energies which shows the ratio of energy deposited in the innermost cell to the inner nine cells in the 5 × 5 crystal array.

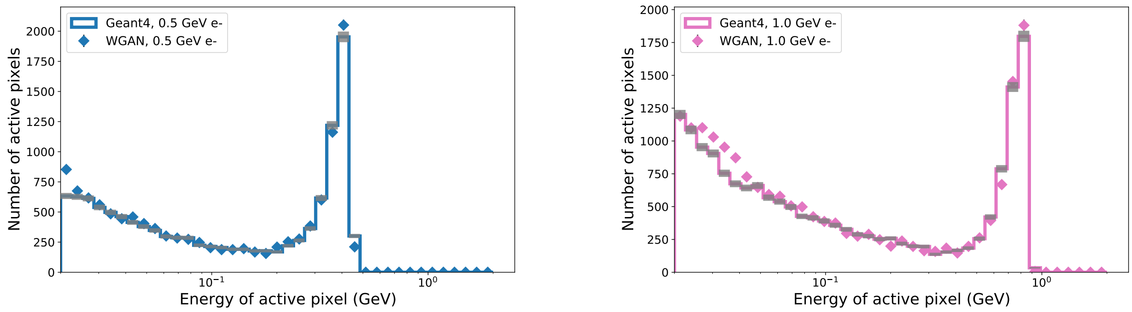


Figure 3.15.: The distribution of energies of active pixels in the 5 × 5 array for 0.5 GeV and 1 GeV electrons.

to examine the accuracy of the showers produced by the WGAN and are discussed below. Only the pixels above the threshold energy of 1 MeV are considered for both WGAN and Geant4 images. The agreement between the distributions of Geant4 and WGAN-generated showers is studied by calculating the p-value using the Kolmogorov–Smirnov test for various observables. The p-value of the distributions corresponding to different physics observables is shown in Tab.3.1.

- Figure 3.13 shows the distribution of the maximum pixel energy deposited in the 5 ×5 crystals for energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV. For the case of an energy of 2 GeV, which is not included in the data set used for the training of the WGAN, is interpolated by the model. The Geant4 and WGAN distribution show a good agreement between them by visual inspection. The p-value for different energies are given in Tab.3.1 for E_{max}. The agreement of interpolated energy 2 GeV seems to be better than other energies. Here the level of significance is set as 0.05. If the p-value is greater than 0.05, then the data samples come from the same distribution.

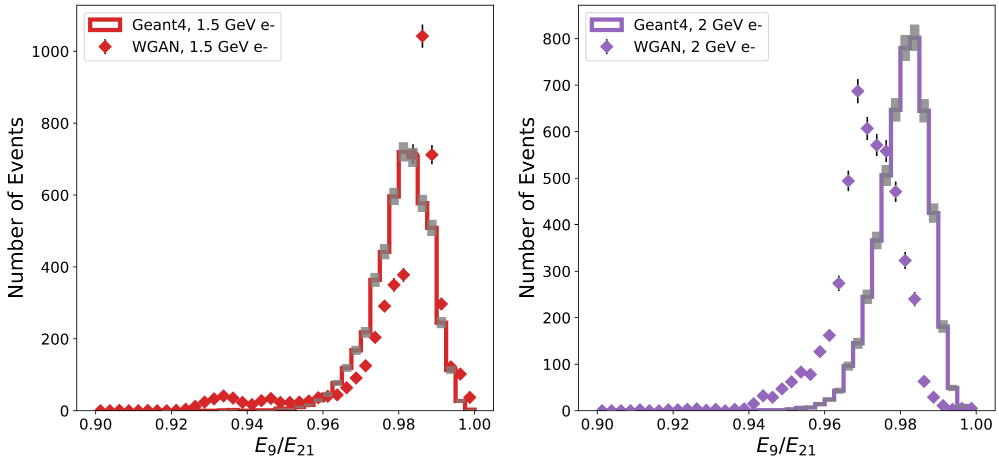


Figure 3.16.: The distribution for E_9/E_{21} shows the ratio of energy deposited in the inner 9 cells to the 21 cells in the 5 × 5 array, excluding the corner cells for 1.5 GeV (left) and 2 GeV (right) energies.

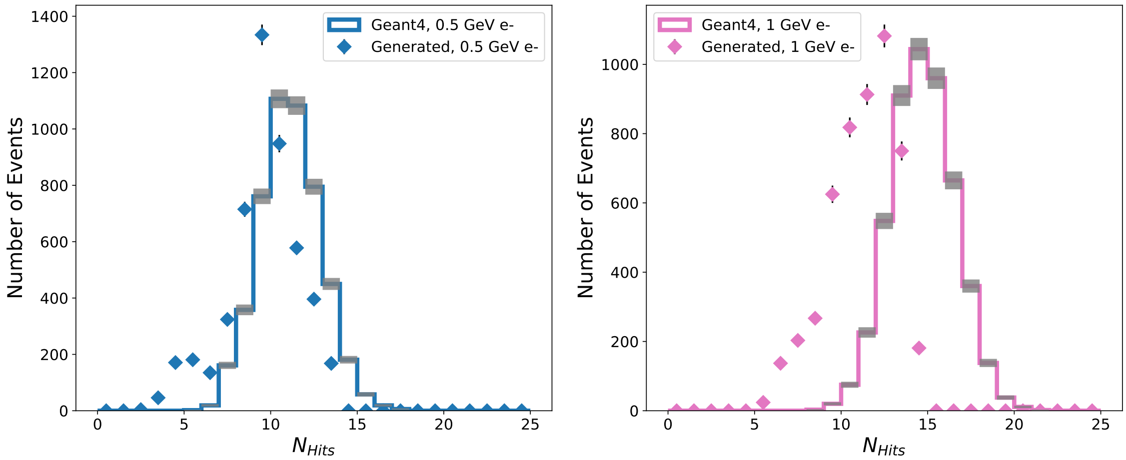


Figure 3.17.: The N_{Hits} distribution, which is the the number of pixels with energies greater than or equal to 1 MeV deposited in the 5 × 5 crystals, for 0.5 GeV and 1 GeV energies.

If the p-value is less than 0.05, they do not come from the same distribution.

- Figure 3.14 illustrates the distribution of the physics variable E_1/E_9 which is an important shower shape variable in Belle II physics analysis. This variable is defined as the ratio of energy deposited in the innermost cell to the inner nine cells of the 5 × 5 crystals. The distribution plotted for 1.5 GeV and 2.5 GeV energies show good agreement between WGAN generated and Geant4 simulated distributions during visual assessment whereas, the p-value calculated through the K-S test is less than 0.05 for 0.5 GeV, 1 GeV, 2 GeV, and 2.5 GeV. Similar plots for other energies are given in Appendix B, section B.2.
- Figure 3.15 illustrates the distribution of all the active pixel energies for 0.5 GeV

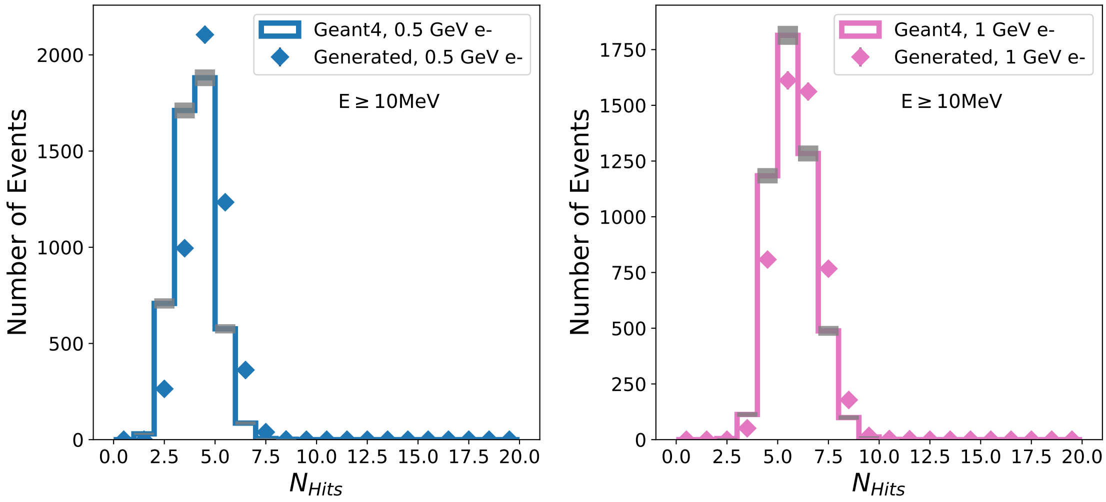


Figure 3.18.: The N_{Hits} distribution with the number of pixels with energies greater than or equal to 10 MeV deposited for 0.5 GeV and 1 GeV images.

and 1 GeV images. Distribution is shown for energies greater than 20 MeV. The distribution shows good agreement between Geant4 and WGAN-generated images for high energy regions, whereas it shows a disagreement towards low energy range as shown in the plot. The p-values for all the energies are greater than 0.05 except for the interpolated 2 GeV energy. The p-values for this variable are calculated using an energy threshold of 50 MeV.

- Figure 3.16 shows the distributions the shower shape variable E_9/E_{21} for 1.5 GeV (left) and 2 GeV (right), which indicates the ratio of energy deposited in the inner 9 cells to the 21 cells in the 5 × 5 crystals, excluding the corner cells. The disagreement observed in these two observables are expected due to the outermost pixels of the 5 × 5 crystals being poorly reconstructed by the WGAN due to negligible energy being deposited. Chapter 4 will discuss this observable.
- Figure 3.17 shows the distributions of the number of hits for 0.5 GeV (left) and 1 GeV (right) which represents the number of pixels with energy greater than or equal to 1 MeV deposited. The disagreement is observed for this observable as the low energy pixels are poorly reproduced by the WGAN. Figure 3.18 shows the plot for the number of hits for 0.5 GeV (left) and 1 GeV (right) for energy greater than or equal to 10 MeV deposited. Here the threshold of the energy is increased from 1 MeV to 10 MeV. A better agreement is observed between the distributions compared to Fig. 3.17.

3.4.4. Computational Performance

The computational performance of Geant4 and WGAN-based simulation is presented in Table. 3.2 for different energies of primary electrons. Figure 3.19 shows the simulation time required per event for primary electrons of energies 0.5 GeV, 1 GeV, 1.5 GeV, 2GeV, and 2.5 GeV via Geant4 and WGAN simulation. The time is expressed on a logarithmic scale.

Table 3.2.: Comparison of the computational performance of Geant4 vs WGAN for the simulation of primary electrons.

	Geant4 (CPU)	WGAN (NVIDIA© GM200)	Speed-up
0.5 GeV	$\mathcal{O}(8\,\text{ms})$	0.3 ms	×27
1 GeV	$\mathcal{O}(15\,\text{ms})$	0.28 ms	×56
1.5 GeV	$\mathcal{O}(24\,\text{ms})$	0.27 ms	×89
2 GeV	$\mathcal{O}(33\,\text{ms})$	0.3 ms	×110
2.5 GeV	$\mathcal{O}(40\,\text{ms})$	0.3 ms	×135

It is observed that for Geant4, the generation time increases as the energy of the particle increases. Whereas for WGAN, the generation time is almost the same for all the energies. There is a computational speed-up for WGAN-generated particles for all the energies.

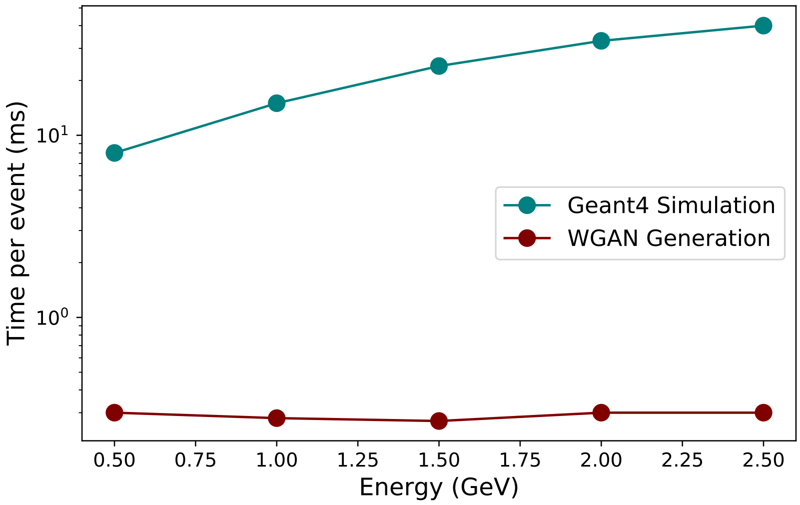


Figure 3.19.: The computational time required for the simulation of primary electrons of energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV via the Geant4 and WGAN approaches.

3.4.5. Summary

In the Geant4 standalone setup, the electron interacts with the crystals and deposits most of its energy inside them. The inner 9 crystals have more energy deposits than the outer crystals. The WGAN model learns and reconstructs the inner pixels easily, whereas it finds difficult to learn the outer pixels due to low energy values. Thus the structure of the image is not entirely reconstructed by the model. To examine the precision of the generated showers, the distributions of E_{max}, E_1/E_9, energy of all the active pixels, E_9/E_{21} and N_{Hits} are compared for Geant4 and WGAN showers. The distributions agree by visual inspection for the variables associated with high energy pixels whereas, for (E_9/E_{21}

and N_{Hits}) involving the outer pixels which contain low energy, the distributions show disagreement. The model was also able to interpolate 2 GeV electron shower responses in crystals which were not part of the training dataset, for E_{max}, E_1/E_9 and enrgy of all the active pixels.

3.5. Training Belle II Electron Showers

The electron shower responses in a standalone crystal setup is an ideal simulation compared to the Belle II electromagnetic calorimeter MC responses, where the presence of beam and luminosity backgrounds and electronic noises cannot be excluded. To explore a more realistic scenario, the WGAN fast simulation is trained on the full detector response from the Belle II ECL simulation.

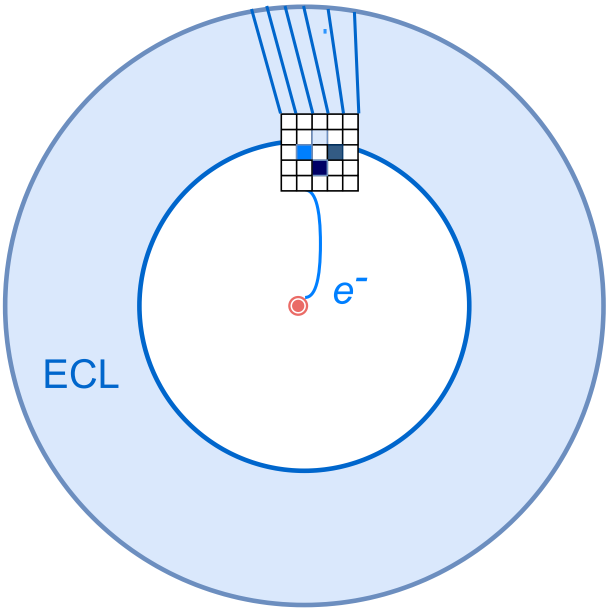


Figure 3.20.: The Belle II Electromagnetic Calorimeter diagram. The electron is shot from the center of the Belle II detector using a particle gun. The electron's trajectory is curved in the Belle II detector due to the presence of magnetic field which is located outside the ECL. The energy of electron is deposited in the crystals of ECL. The deposited energy in the 5 × 5 crystal array is collected and converted to an image.

The basf2 is the software framework developed for the Belle II experiment. This framework provides the capability to load and manage various libraries and modules used for Belle II data processing [37]. For producing the training samples, the detector simulation module, based on Geant4 is used [37]. The particle gun conditions for the simulation are set as given in Tab.B.1 in Appendix B. To simulate realistic data taking condition during the generation of MC samples, different beam background levels can be introduced. Samples are produced with a standard estimate of beam background which is approximately equal to the one associated with the current luminosity. The polar angle coverage $\theta_{\mathrm{generated}}$ is given between 11.4° – 156.1° and the azimuth angle $\phi_{\text{generated}}$ is set as 0° – 360°. The energy of electron showers used for training are 0.5 GeV, 1 GeV, 1.5 GeV and 2.5 GeV. The 2 GeV electron shower, which is not included in the training set, is used for the evaluation of the simulation performance. These showers are used to inspect if the model can interpolate the energies.

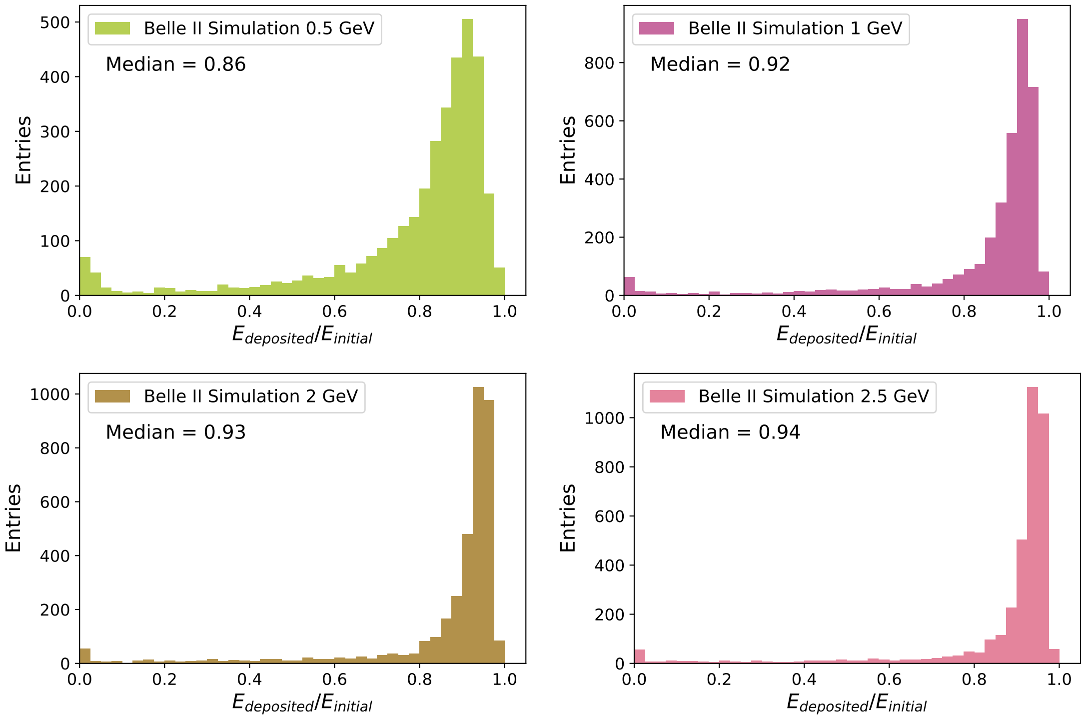


Figure 3.21.: The ratio of electron energy deposited in the 5 × 5 crystal of the Belle II ECL to the incident electron energy for 0.5 GeV, 1 GeV, 2 GeV, and 2.5 GeV.

Figure 3.20 shows a pictorial diagram of the Belle II ECL (blue region) and the electron responses in a 5 × 5 array of crystals. The particle gun is placed at the nominal Belle II beam interaction point and electrons are simulated according to the conditions described in Tab. B.1 Appendix B. The effects of the following subdetectors, pixelated silicon sensors (PXD), silicon strip sensors (SVD), central drift chamber (CDC), and Time-Of-Propagation (TOP) counters are included during the simulation of primary electrons in the ECL. This results in energy loss of the primary electron before it reaches the ECL. The electron hits the ECL crystals. The function eclcaldigitExtEnergy defined in the basf2 is used to read out the energy deposited in the 5 × 5 crystals. Figure 3.21 shows the ratio of energy deposited in the ECL cluster to the energy of primary electrons. The median of the distribution shifts from 0.86 to 0.94 as the energy of the primary particle increases from 0.5 GeV to 2.5 GeV. For the low-energy electrons like 0.5 GeV, the path length is longer due to the effect of the magnetic field resulting in an increased energy loss. The shower images which contain very low energy deposits are not included in the training datasets.

Once the training samples are collected, the previously developed WGAN architecture is used to train these images. The labels such as energy of the primary particle, $\theta_{\text{generated}}$,

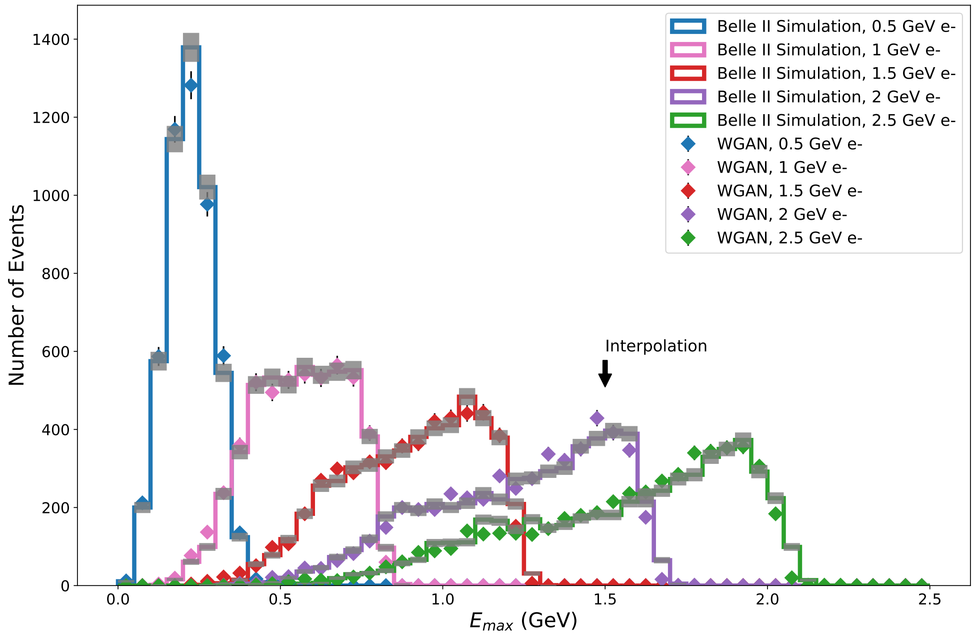


Figure 3.22.: The distribution of the maximum pixel energy in the Belle II ECL 5 × 5 crystals for 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV energies. The images of 2 GeV electrons are not included in the training dataset.

$\phi_{\mathrm{generated}}$, x and y vertex are given to condition the network. As discussed in Section 3.4, an energy regressor is added to the network to constrain the energy of the showers. Here unlike in Section 3.4, the loss function of the energy regressor for the $\mathrm{i^{th}}$ real (Geant4) image based on mean squared error is given as:

$$\mathrm{L_{real,i}} = [\mathrm{ER(x_i) - ET_i}]^2, \tag{3.9}$$

where $\mathrm{ER(x_i)}$ represents the output of the energy regressor when real image $\mathrm{x_i}$ is given and $\mathrm{ET_i}$ refers to the value of total energy deposited in the the image $\mathrm{x_i}$. The loss function of the fake image based on mean squared error is given as:

$$\mathrm{L_{fake,i}} = [\mathrm{ER(G(z_i, E_i, P_i, A_i)) - ET_i}]^2 \tag{3.10}$$

where $\mathrm{ER(G(z_i, E_i, P_i, A_i))}$ refers to the output value of the energy regressor when the fake image is given. The letters z, E, P and A refers to the noise vector, energy label, vertex label and angle label respectively. To evaluate the WGAN generated images, physics observables such as $\mathrm{E_{max}}$, $\mathrm{E_1/E_9}$, the energy of all the active pixels, $\mathrm{E_9/E_{21}}$ and $\mathrm{N_{Hits}}$ are compared for Belle II simulation and WGAN showers. The results and observations are discussed below. The agreement between the distributions is confirmed by calculating the p-value through the Kolmogorov–Smirnov test. The different physics observables and their p-values are given in Tab. 3.3.

- Only the pixels equal to or above the threshold energy of 1 MeV are considered for the shower responses from both Belle II and WGAN images.

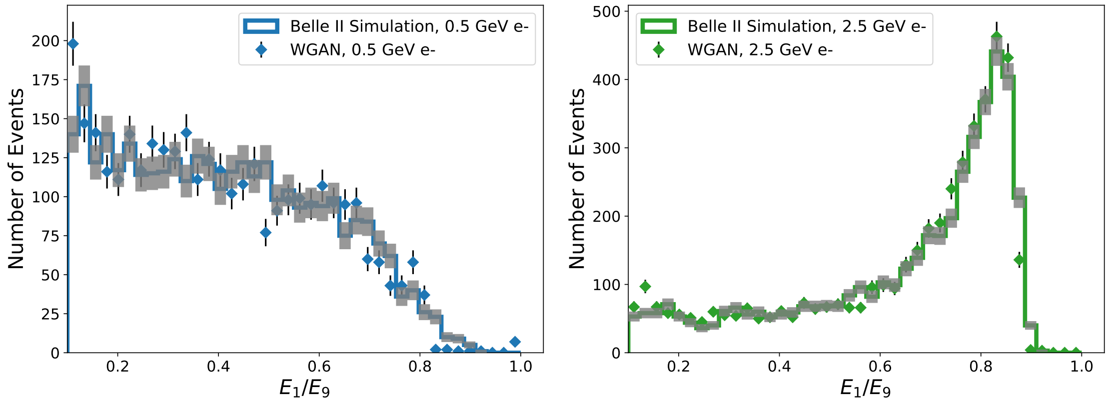


Figure 3.23.: Distribution of E_1/E_9 for 0.5 GeV and 2.5 GeV energies which shows the ratio of energy deposited in the innermost cell to the inner nine cells in the 5 × 5 Belle II ECL crystals.

Table 3.3.: The p-values for different energies for various observables calculated using the K-S test.

Observable	0.5 GeV	1 GeV	1.5 GeV	2 GeV	2.5 GeV
E_{max}	0.11	0.34	0.06	8×10^{-3}	0.011
E_1/E_9	9×10^{-12}	3×10^{-6}	5×10^{-6}	3×10^{-5}	1.5×10^{-5}
Energy active pixels	0.74	0.98	0.69	4×10^{-4}	5×10^{-6}

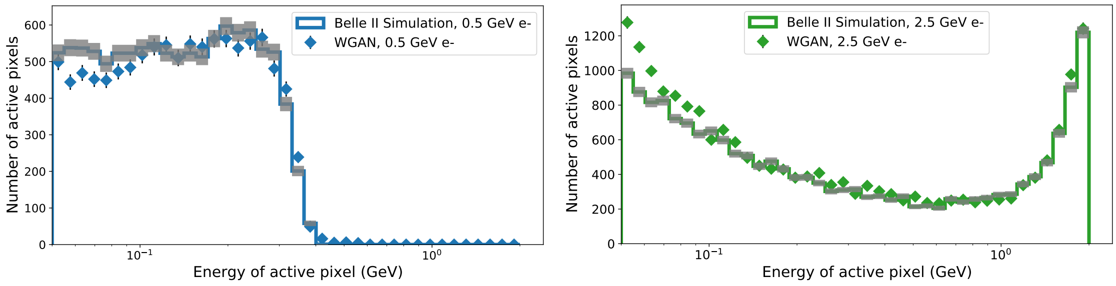


Figure 3.24.: The distribution of energies of every active pixel in the 5 × 5 Belle II ECL crystals for 0.5 GeV and 2.5 GeV electrons.

- Figure 3.22 shows the maximum pixel energy distribution of electrons of energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV in the 5 × 5 ECL crystals. The 2 GeV electrons are interpolated by the model and are not a part of the training datasets. Statistical uncertainties are added for the Geant4 distribution (grey bars) and the WGAN distribution (black bars). By visual inspection, both the distribution agrees quite well. The p-value given in Tab. 3.3 for E_{max} is greater than 0.05, for all the energies except 2 GeV and hence shows a good agreement between the distributions. Here the level of significance is set as 0.05.

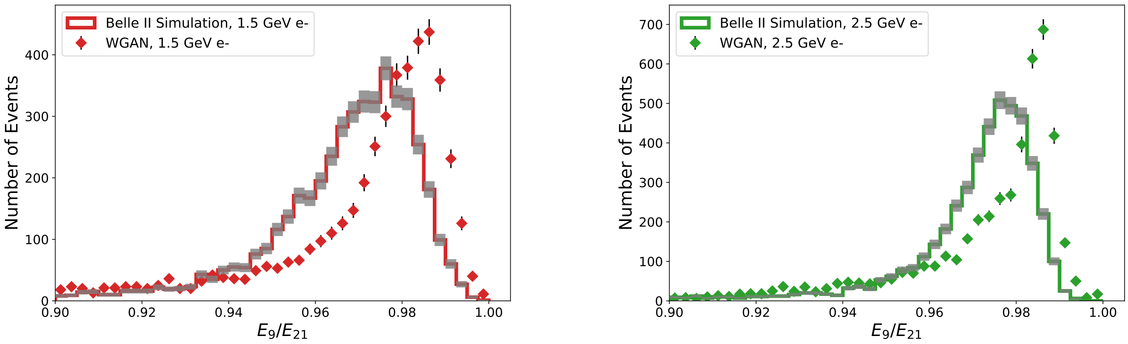


Figure 3.25.: The distribution of E_9/E_{21} in 5 × 5 Belle II ECL crystals for 1.5 GeV (left) and 2.5 GeV (right) electrons.

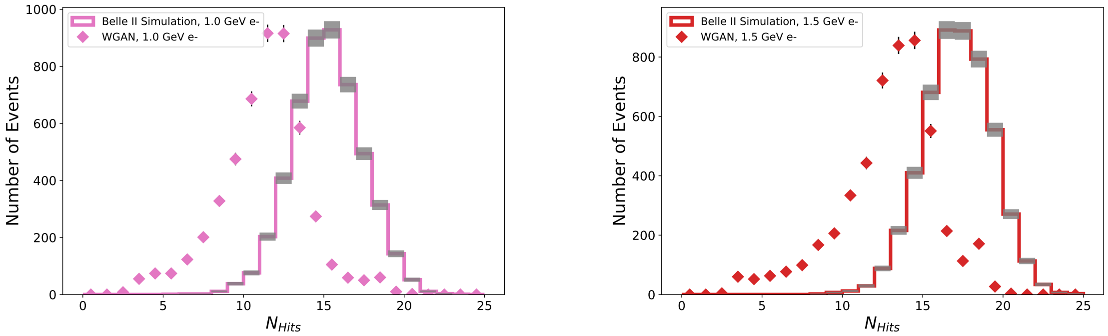


Figure 3.26.: The distribution of N_{Hits} in 5 × 5 Belle II ECL crystals for 1 GeV (left) and 1.5 GeV (right) with energies greater than or equal to 1 MeV deposited in the crystals.

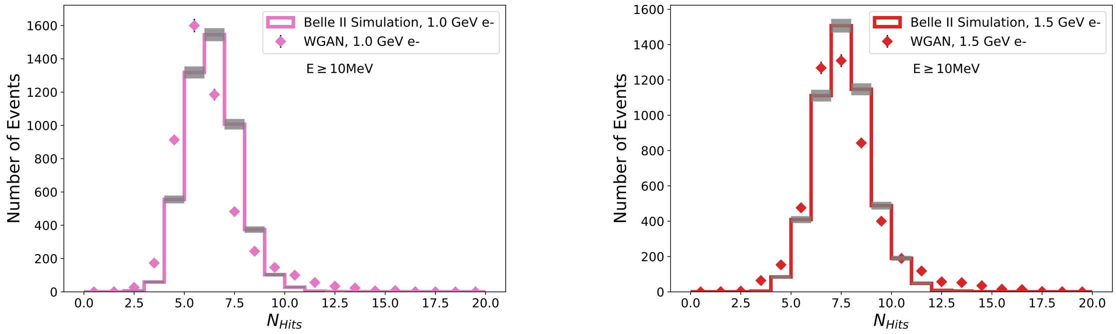


Figure 3.27.: The distribution of N_{Hits} in 5 × 5 Belle II ECL crystals for 1 GeV (left) and 1.5 GeV (right) with energies greater than or equal to 10 MeV deposited in the crystals.

- Figure 3.23 shows the distribution of E_1/E_9 for 0.5 GeV and 2.5 GeV Geant4 and generated images. By visual estimation, both the distribution agrees. The p-value given for E_1/E_9 for different energies are less than 0.05, and hence shows the data points are not from the same distribution. Plots for other energies are given in the

Appendix B, section B.3.

- Figure 3.24 shows the distribution of energies of every active pixel for 0.5 GeV and 2.5 GeV energies. Distribution is shown for energies of active pixels greater than 50 MeV. Distributions disagree towards the lower energy range as shown in the plots. Except for the 2 GeV and 2.5 GeV energies, the p-values are greater than 0.05, which shows that the data points of simulated and WGAN generated are from the same distributions. The p-values for this variable are calculated with an energy threshold of 50 MeV.
- Figure 3.25 shows the distributions of the variable E_9/E_{21} for 1.5 GeV (left) and 2.5 GeV (right). The distributions show a disagreement between Geant4 and generated showers. This is due to the failure of WGAN in the reconstruction of low-energy pixels.
- Figure 3.26 shows the distributions of the number of hits for 1 GeV (left) and 1.5 GeV (right), which denotes the number of active pixels 5 × 5 pixels with an energy greater than or equal to 1 MeV deposited. Figure 3.27 shows the plot for the number of hits for 1 GeV (left) and 1.5 GeV (right) for energies greater than or equal to 10 MeV deposited in the crystals. When the threshold of the energy is increased from 1 MeV to 10 MeV, a better agreement is observed between the distributions.

3.5.1. Computational Performance

Table 3.4.: Comparison of the computational performance of BelleII ECL simulation vs WGAN for the generation of 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV and 2.5 GeV electrons.

	Belle II Simulation (CPU)	WGAN (NVIDIA© GM200)	Speed-up
0.5 GeV	$\mathcal{O}(45\,\mathrm{ms})$	0.3 ms	×150
1 GeV	$\mathcal{O}(85\,\mathrm{ms})$	0.26 ms	×327
1.5 GeV	$\mathcal{O}(126\,\mathrm{ms})$	0.24 ms	×525
2 GeV	$\mathcal{O}(178\,\mathrm{ms})$	0.27 ms	×659
2.5 GeV	$\mathcal{O}(208\,\mathrm{ms})$	0.3 ms	×693

Table 3.4 shows the computational time required for generating electrons using the Belle II ECL simulation framework, and the WGAN method for different energies of primary electrons. There is a difference in the time required for simulation of different energies of electrons in the case Belle II ECL simulation, compared to the Geant4 standalone setup. Figure 3.28 shows the simulation time required per event for different energies of primary electrons through Belle II and WGAN simulation. The time is expressed on a logarithmic scale. The simulation time required per event increases as the energy increases, for the case of the nominal Belle II simulation approach. For the case of WGAN, the time required is almost the same for the different energies. The WGAN also offers a speed-up of about 700 times compared to the Belle II simulation.

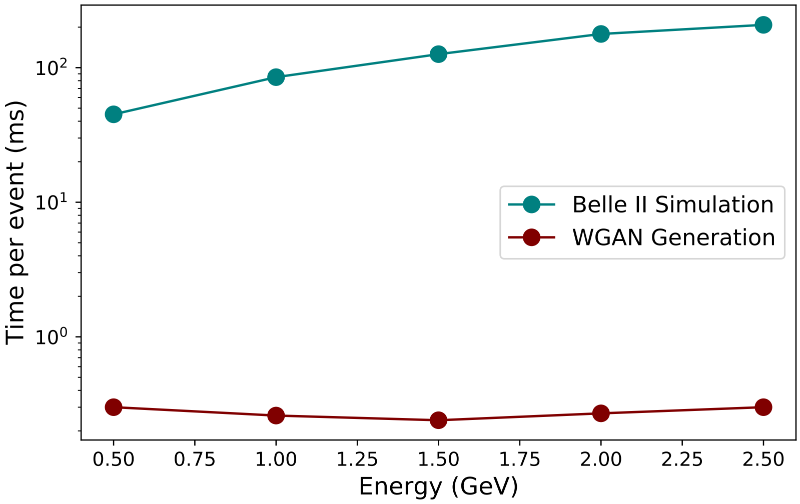


Figure 3.28.: The computational time required per event for the simulation of primary electrons of different energies via the Belle II simulation and WGAN approaches.

3.6. Training Hadronic Shower Simulation of Geant4 Standalone Setup

To study the simulated hadronic shower responses and its fast simulation, negatively charged pion showers (π^-) are simulated in Geant4 standalone setup. The fast simulation of pion showers in electromagnetic calorimeters involves the following challenges.

- It is expected that around 37% of incident hadrons will not interact since the crystals are about one interaction length deep [38]. Thus a large number of pions traverse the detector length without interacting. This is illustrated in the top image of Fig. 3.29.
- As shown in the center image of Fig. 3.29, some of the remaining incident pions deposit a small portion of their energy inside the crystals and the rest outside.
- The rest of the pions deposit their entire energy or majority of their energy inside the crystals as shown in the third image of Fig. 3.29.
- Training datasets are made with pions that had their hadronic interaction inside the crystals.
- Since pions are minimum ionizing particles, the energy deposited in the crystals peak around 200 MeV for every discrete energies, which will make it challenging for the WGAN model to distinguish them.
- The energy range of pions in the Belle II experiment are a few GeV, and the energy deposited by them in the crystals is very less which makes it difficult for the WGAN model to reproduce them. The Belle II detector does not have a separate hadronic calorimeter, so the ECL and other subdetectors is used to identify pions for the experiment. In the case of fast simulation of electrons, producing low energy pixels is found to be challenging for the WGAN model. This is expected to be the case with pions as well where the innermost cell contains most of the energy compared to the outer cells.

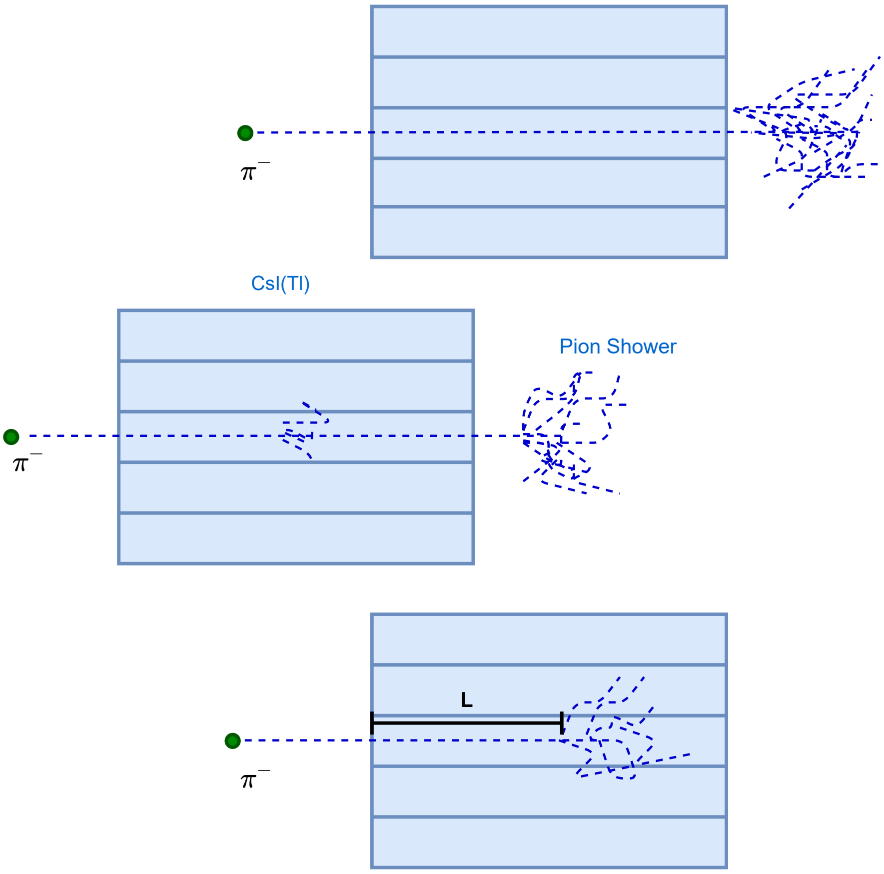


Figure 3.29.: Pion showers in the calorimeter crystals. The topmost image shows the case where the pion does not interact inside the crystals. The center image depicts the case of hadronic interaction happening inside the crystals where some of the energy is deposited in the crystals. The bottom image depicts the case of hadronic interaction happening inside the crystals where the pion deposits the entirety or majority of its energy inside the crystals. The term L denotes the first hadronic interaction length.

3.6.1. Pion Energy Responses in the Geant4 Standalone Setup

For the study of the fast simulation of hadronic showers, pions of energies 0.5 GeV, 0.75 GeV, 1 GeV, and 1.5 GeV are selected to train the model. This choice of pion energies ensures that most of the particles interact with the crystals. The simulation of pion showers for these energies in the Geant4 5 × 5 crystal setup is performed as discussed in Section 2.4. Figure 3.30 shows the average of around 120000 Geant4 images for 0.5 GeV and 1.5 GeV pion showers. Energy is depicted on a logarithmic scale. The figure shows that the cells, with the exception of the centermost pixel, contain a very small amount of deposited energy. The first hadronic interaction depth (L) variable inspired from [39] is used in the Geant4 program to define the point of first hadronic interaction inside the crystals. It is the distance between the entry point of the primary particle in the crystal to the first hadronic interaction point. Once the training samples are made, they are trained on the

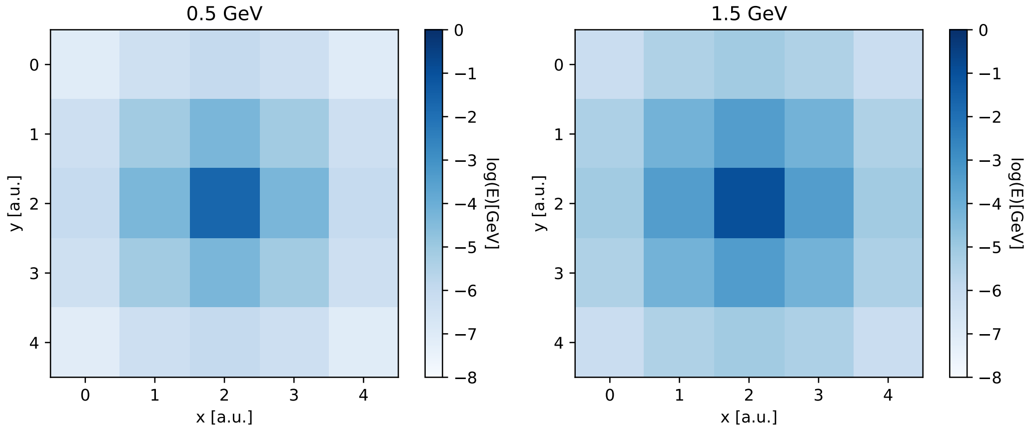


Figure 3.30.: The Figure depicts the average of Geant4 simulated pion showers in the 5 × 5 standalone crystal setup. Negatively charged pions of energy 0.5 GeV (left) and 1.5 GeV (right) are shown in the diagram.

WGAN model with an additional energy regressor. The details of the architecture used to train pion images are given in Appendix A, Tab.A.4, A.2, and A.5. Here the energy of the primary particle and the variable, first hadronic interaction depth, are given as the labels to the generator and the critic. Providing L to the generator and critic as a label to condition the model while training significantly improves the performance. For the pion dataset, the particle gun position is fixed at the center to keep the training simple. Energy less than 1 MeV deposited in the pixels are not considered for both WGAN and Geant4 images.

3.6.2. Results

- Figure 3.31 shows the distribution of maximum pixel energy in the 5 × 5 crystals for energies of 0.5 GeV, 0.75 GeV, 1 GeV, and 1.5 GeV. It is observed that the peak of the distribution is around 200 MeV for all the energies.
- Figure 3.32 shows the distribution of active pixels in the crystal array. The agreement between the WGAN and Geant4 distributions declines towards low energy regions.
- Figure 3.33 shows the distribution of E_1/E_9 for 0.5 GeV and 1.5 GeV with large disagreement. This is because most of the pixels with low energy deposits are not simulated adequately in the generated images.
- Figure 3.34 shows the energy deposition in 5 × 5 crystals for corresponding Geant4 and WGAN generated pions of different energies. As expected, WGAN fails to simulate adequately low-energy pixels.

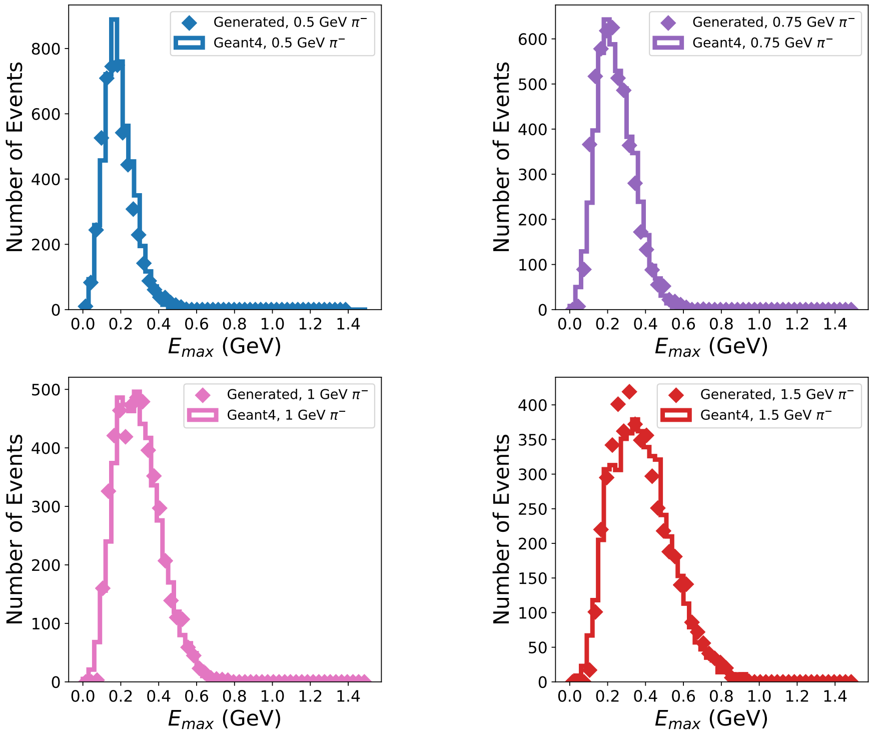


Figure 3.31.: The distribution for the maximum pixel energy of pions for energies 0.5 GeV, 0.75 GeV, 1 GeV and 1.5 GeV.

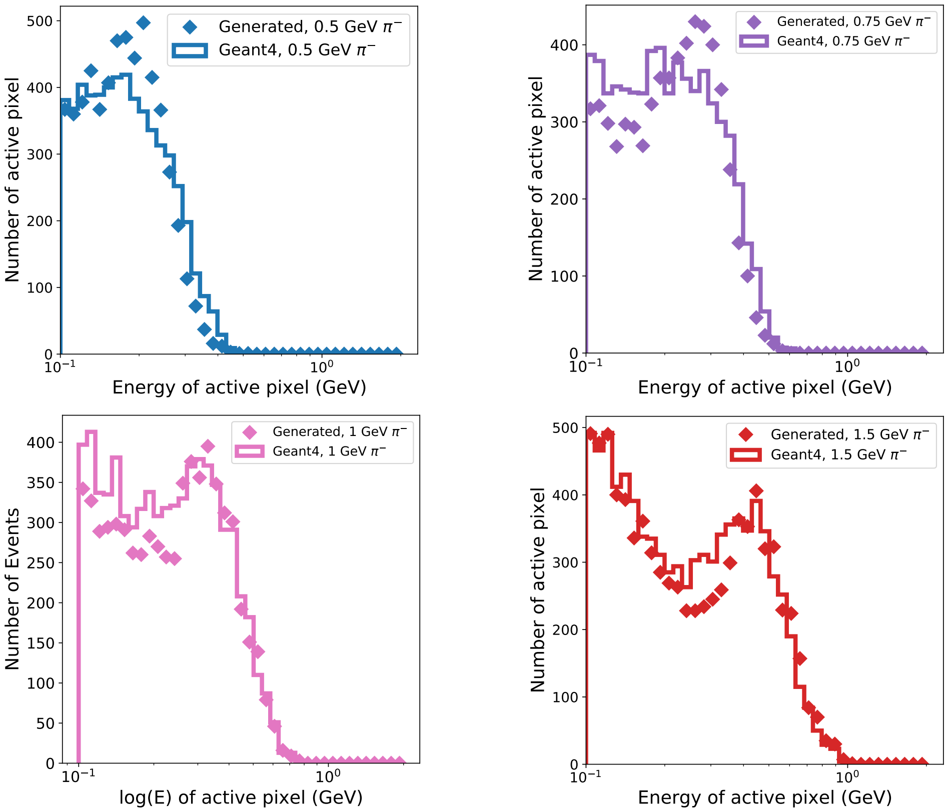


Figure 3.32.: The distribution of energies of active pixels in the crystal array. The agreement between Geant4 and WGAN distribution worsens towards the low-energy regions of the pion shower.

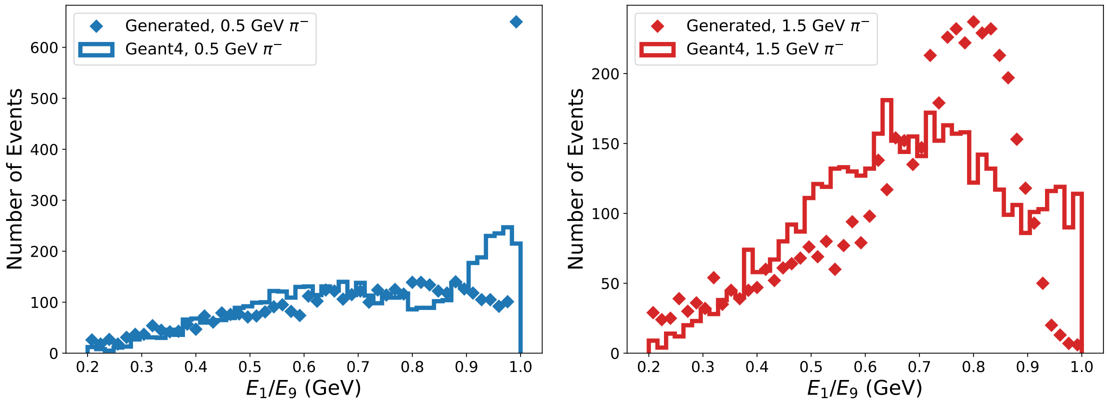


Figure 3.33.: Distribution of E_1/E_9 for 0.5 GeV and 1.5 GeV pions.

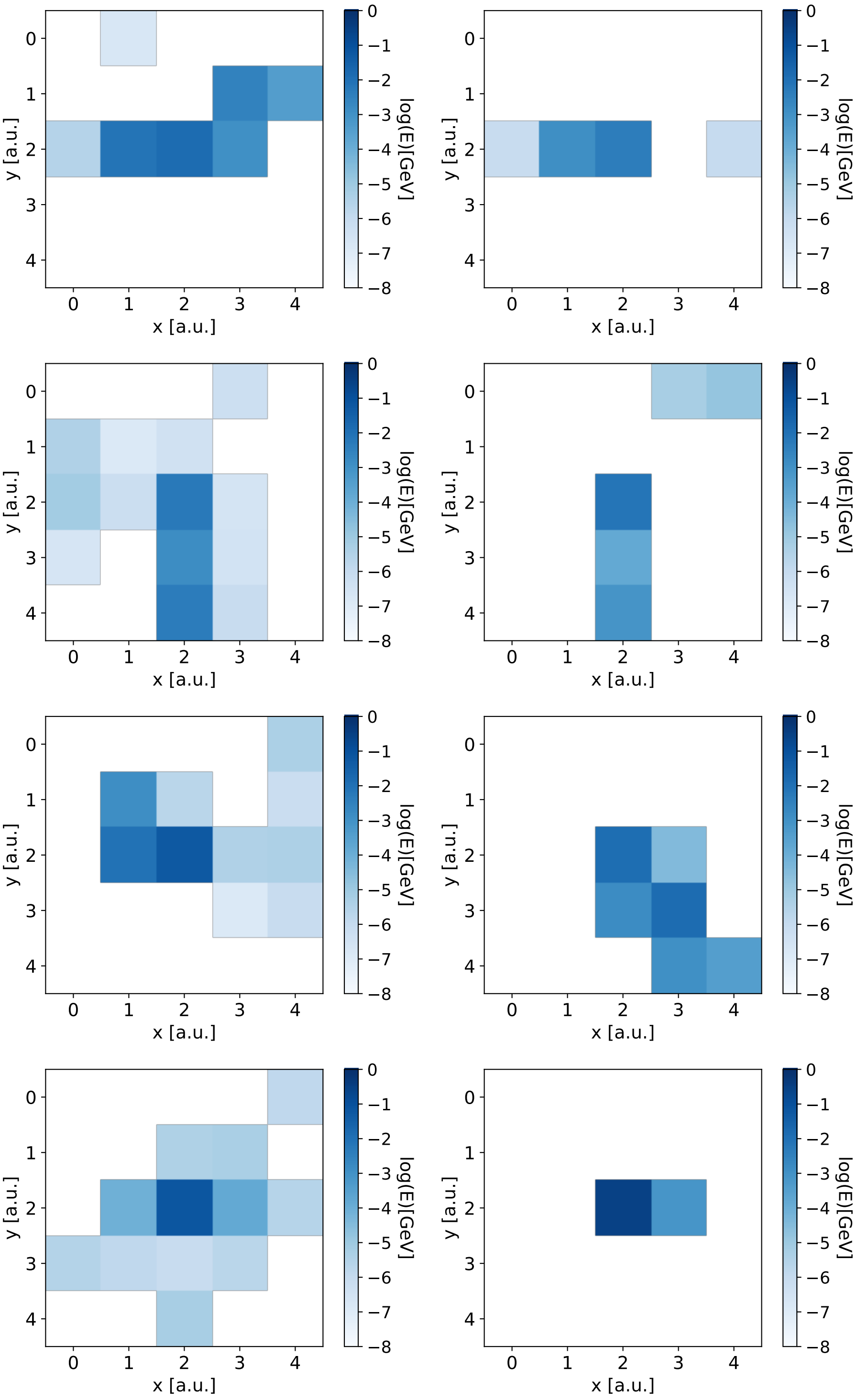


Figure 3.34.: The images show the energy depositions in 5 × 5 crystals for corresponding geant4 (left column) and WGAN (right column) generated images of pion showers for energies 0.5 GeV (1st row), 0.75 GeV (2nd row), 1 GeV (3rd row), and 1.5 GeV (4th row).

3.6.3. Computational Performance

The computational time required for the simulation of negatively charged pions of energies 0.5 GeV, 0.75 GeV, 1 GeV, and 1.5 GeV via the WGAN and Geant4 method is given in Tab.3.5. Since the energies of primary particles are less, the simulation time via Geant4 is comparatively less. As the energy increases from 0.5 to 1.5 GeV, the computational time increases for the Geant4 approach. But for the WGAN approach, the time remains almost the same for all the energies which are clearly shown in Fig 3.35.

Table 3.5.: Computational performance of Geant4 vs WGAN for the simulation of negatively charged pions.

	Geant4 (CPU)	WGAN (NVIDIA© GM200)	Speed-up
0.5 GeV	$\mathcal{O}(16\,\mathrm{ms})$	0.33 ms	×48
0.75 GeV	$\mathcal{O}(24\,\mathrm{ms})$	0.34 ms	×71
1 GeV	$\mathcal{O}(28\,\mathrm{ms})$	0.34 ms	×82
1.5 GeV	$\mathcal{O}(36\,\mathrm{ms})$	0.33 ms	×109

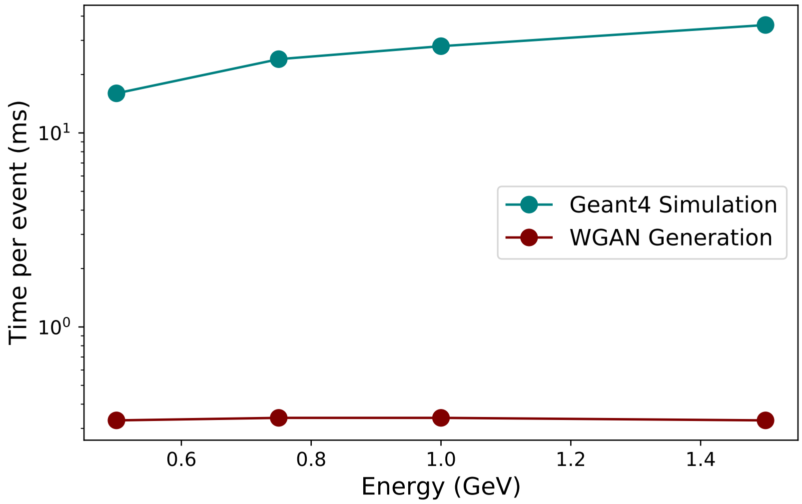


Figure 3.35.: The computational time for the simulation of primary pions of different energies through the Geant4 and WGAN approaches.

3.6.4. Summary

The simulation of pion showers in the Geant4 standalone setup is challenging due to the low range of energies deposited inside the crystals. The conditioning of the networks with the true value of first hadronic interaction depth of the primary particle has helped the model to an extend to achieve better performance. The plots of various physics observables like E_{max} and cell energy distribution show a considerable agreement between Geant4 and WGAN generated showers. The shower shape variable E_1/E_9 is poorly reconstructed, which is expected due to the low range of energies deposited. The image produced by WGAN has significantly fewer pixels. Thus the fast simulation of pion showers in the Geant4 standalone is not executed successfully.

4

VAE-WGAN

4.1. Variational Autoencoder (VAE)

Variational Autoencoder (VAE) [40] [41] is a generative network that consists of a probabilistic encoder, a probabilistic decoder, and a reparameterized learned latent space that capture the essential features of the dataset. Fig 4.1 illustrates the different components of the VAE architecture. The training dataset x is passed to the encoder, which encodes them as a distribution. The latent space z sampled from this distribution is forwarded to the decoder network to reconstruct the real dataset. The loss function of the VAE is given as [42]

$$\mathrm{L}_{\mathrm{VAE}} = \mathbb{E}_{\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})}[\log \mathrm{p}_\theta(\mathrm{x}|\mathrm{z})] - \mathrm{D}_{\mathrm{KL}}[\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z})] \tag{4.1}$$

where $\mathrm{q}(\mathrm{z}|\mathrm{x})$ represents the encoder network that gives an approximate posterior distribution. The ϕ encapsulates the parameters of the encoder network. The term $\mathrm{p}(\mathrm{x}|\mathrm{z})$ represents the decoder network that gives a likelihood distribution. The θ encapsulates the parameters of the decoder network. The term $\mathrm{p}(\mathrm{z})$ represents the prior distribution. The term $\mathbb{E}_{\mathrm{z}\sim\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})} \log \mathrm{p}_\theta(\mathrm{x}|\mathrm{z})$ defines the reconstruction part of the loss function. The term $\mathrm{D}_{\mathrm{KL}}[\mathrm{q}(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z})]$, Kulback-Leibler divergence (KLD) loss also called the latent loss, shows the KL divergence between an encoded gaussian distribution and the standard normal distribution (prior distribution). This loss term regularises the latent space by bringing the distributions $\mathrm{q}(\mathrm{z}|\mathrm{x})$ and $\mathrm{p}(\mathrm{z})$ closer. The RHS of (4.1) is called the evidence lower bound (ELBO). The objective of the variational inference is to maximize the ELBO.

Usually, the prior distributions $\mathrm{p}(\mathrm{z})$ are chosen to be a standard normal distribution. The encoder encodes input x to mean μ and standard deviation σ resulting in a normal

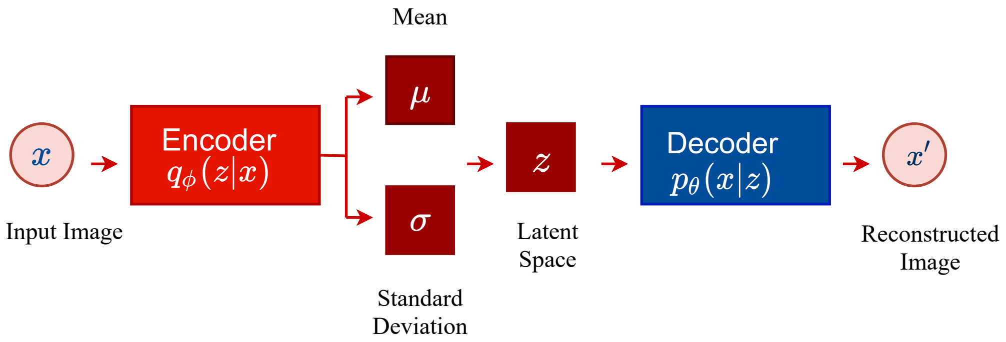


Figure 4.1.: The pictorial representation of the Variational Autoencoder. It contains an encoder that encodes essential features of the dataset to μ and σ from which the latent space variable z is sampled. The vector z is then passed to the decoder to decode the dataset x from z.

distribution from which the latent space z is sampled. If the size of z vector is n, there will be n number of corresponding μ and σ. The KLD loss tries to bring the encoded distribution closer to the prior distribution. The completeness and the continuity of the latent space are ensured through regularization, which helps to attain a meaningful latent space while generating and interpolating new samples.

- **Completeness:** Every point in the latent space should give a meaningful result on decoding. If the latent space is incomplete, the interpolation of some points in the latent space can lead to a completely meaningless result. The image on the left of Fig. 4.2 depicts an incomplete latent space. Even though each class of the datasets is encoded,they exist far apart from each other in the latent space, creating a vacuum space between them. When a point in this vacuum space is interpolated, it provides a meaningless result that does not belong to any dataset classes. Regularization helps to encode data classes closer to each other in the latent space as shown in the image on the right of Fig. 4.2.
- **Continuity:** Two points close to each other in a latent space should generate comparable outputs when decoded. As shown in the image on the right of Fig. 4.2, the points that belong to a particular class boundary can decode the data distribution of that class. However if a point in the latent space belongs to a region between two classes, it should decode to a data distribution similar to the distribution of these two classes.

Beta VAE model is an extension of VAE [44] for which the loss function is modified to

$$\mathrm{L_{VAE}} = \mathbb{E}_{\mathrm{z}\sim \mathrm{q}_\phi(\mathrm{z}|\mathrm{x})}[\log \mathrm{p}_\theta(\mathrm{x}|\mathrm{z})] - \beta \mathrm{D_{KL}}[\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z})]. \tag{4.2}$$

Here β refers to the weight of the latent loss term. If β is 0, the loss function reduces to that of a simple autoencoder with only the reconstruction term. On the other hand, if β

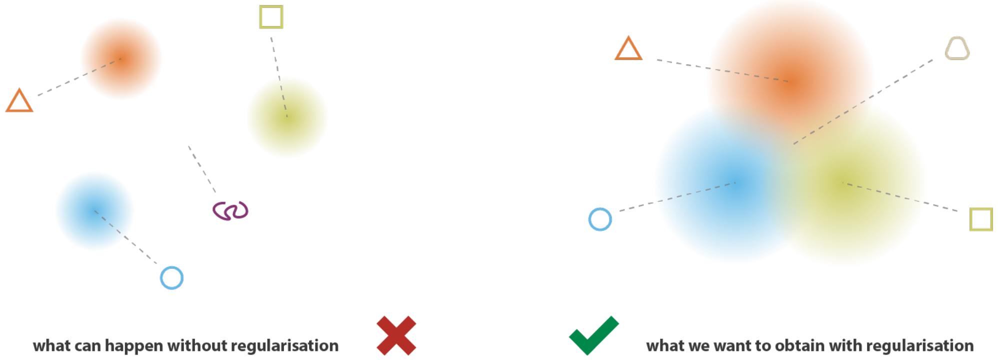


Figure 4.2.: The figure depicts the continuity and completeness of latent space. The image on the left shows the distributions of different classes that are encoded far apart in the latent space, giving meaningless output during interpolation. This image lacks continuity and completeness. The image on the right shows an encoded latent space of different classes with overlapped distributions which is complete and continuous [43].

is 1, the loss function of a VAE is obtained. The value of β is a hyperparameter that is adjusted to achieve a balance between the reconstruction loss and the KL loss. During training, if the model falls into the local optimum of the loss function, it suffers from posterior collapse. Consequently the variational posterior $\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})$ coincides with the prior $\mathrm{p}_\theta(\mathrm{z})$ and the KL loss collapses to 0. During this time decoder disregards the latent space z while reconstructing the dataset [42].

Mutual information between the latent space z and the reconstructed data is represented as I_q. The value of I_q greater than 1 means the decoder collects information from latent space z while generating the output. If $0 < I_q < 1$, then the decoder does not take any information from z while reconstructing the output. The details to calculate I_q between z and x are described in [45]. The method of calculation of mutual information used in this thesis is from [46] [47]. When posterior collapse happens, similar to the latent space, the I_q also collapses near to 0. Thus balancing the KL loss and the reconstruction loss is required to avoid posterior collapse. Mutual information-based loss function is also discussed in [48].

4.2. Training of VAE

The VAE model is trained using electron shower images of 5×5 pixels. The reconstruction loss in the VAE loss function is obtained here by calculating the mean square error of each pixel in the dataset image and the decoded image. This minimizes the difference between real and reconstructed datasets. The training suffers from posterior collapse when the value of β for the loss function given in Eqn.(4.2) is 1. Balancing the KL loss term and the reconstruction loss term is necessary to avoid this problem. Trainings are executed for different values of β. For the β value of 3×10^{-5}, the KL loss collapse issue is mitigated and as a result, the mutual information between the latent space variable z and the x are

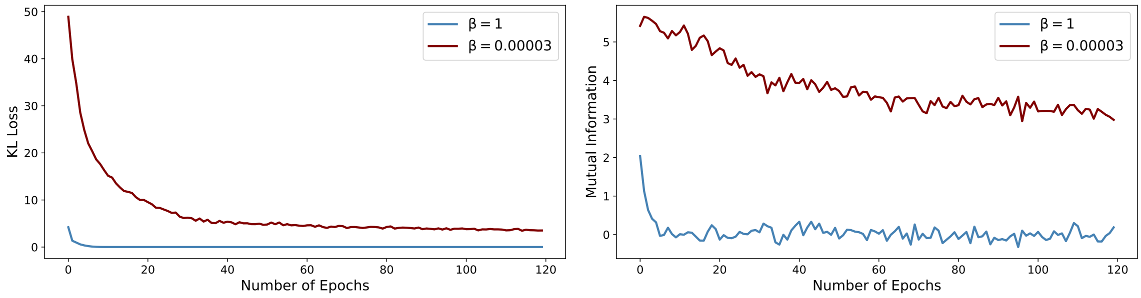


Figure 4.3.: The image on the left shows the KL loss and the image on the right shows the Mutual Information (I_q) for different values of β. When $\beta = 1$, the KL loss and I_q collapses to 0 after a few batches of training, whereas when $\beta = 3 \times 10^{-5}$, the variables maintain a positive value.

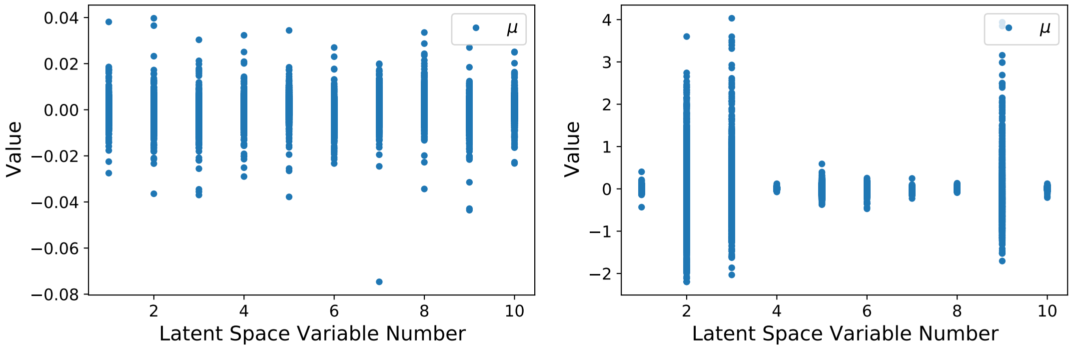


Figure 4.4.: The image on the left shows the encoded μ for $\beta = 1$ where no information content is stored in any of the ten latent space variables. The image on the right shows the encoded μ for $\beta = 3 \times 10^{-5}$, where three out of ten latent space variables have information contained in them.

improved. Fig. 4.3 shows the I_q between latent space z and data x for the β values 1 and 3×10^{-5}. When $\beta = 1$, I_q collapses to around 0, and when $\beta = 3 \times 10^{-5}$, I_q is a positive value which indicates that the decoder has collected information from latent variable z while reconstructing the datasets.

The encoded mean for the two different cases is shown in Fig. 4.4. The image on the left shows the encoded μ when $\beta = 1$. Obviously, there is no information in any of the ten latent space variables. The image on the right illustrates that three variables out of 10 contain most of the information when $\beta = 3 \times 10^{-5}$.

The t-SNE plot of latent spaces for the model with two different β values are shown in Fig. 4.5. The image on the left has β =1, which is contains no gap but lacks information content in the encoded variables as shown in Fig. 4.4 (left). The image on the right has $\beta = 3 \times 10^{-5}$, the latent space is not regularized as it is not complete and has gaps between the distribution, however, the encoded variables contain information as shown in Fig. 4.4(right). The figure depicts the arrangement of the latent space for discrete energies.

The comparison between the distribution of a latent z variable and the Standard Normal

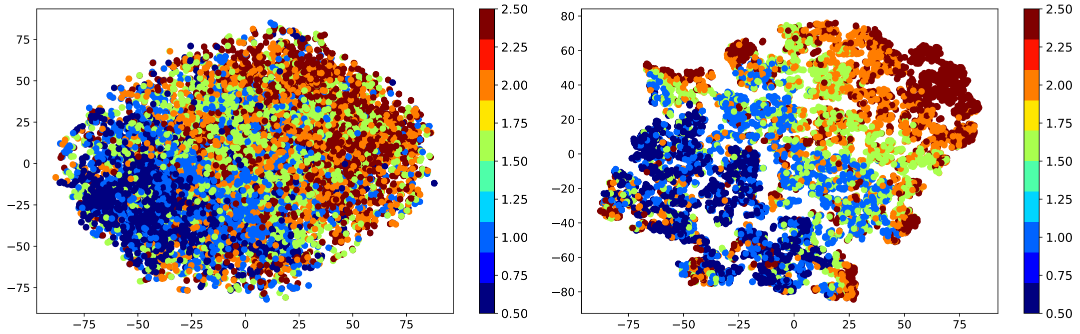


Figure 4.5.: Image on the left shows the t-SNE plot for the latent space for values of $\beta = 1$. The image on the right shows the t-SNE plot of the latent space for $\beta = 3 \times 10^{-5}$ which is not well regularized. The different colors represent different energies.

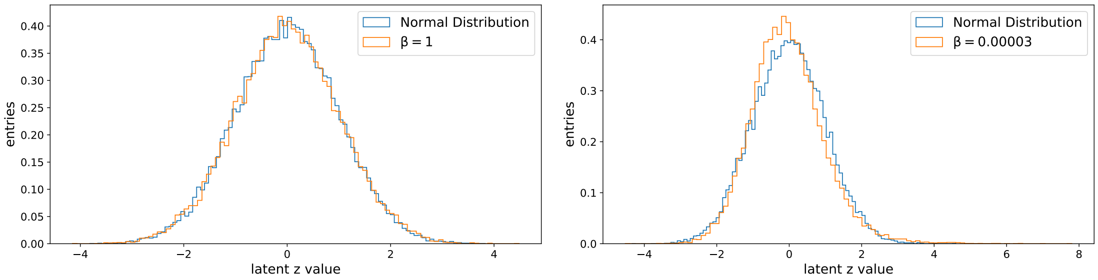


Figure 4.6.: The plots show the deviation between the high information containing z variable distribution and the Normal distribution N(0,1). When $\beta = 1$ (left), both distributions coincide with each other. When $\beta = 3 \times 10^{-5}$ (right), the distribution of the z variable deviates from the Standard Normal distribution N(0,1).

distribution N(0,1) is shown Fig. 4.6. The distributions come closer for $\beta = 1$ and moves apart when $\beta = 3 \times 10^{-5}$.

4.3. VAE–WGAN

The absence of an encoder for the inference of data samples is one of the drawbacks of a GAN [8]. One way of addressing this problem is to combine an encoder network with a WGAN. For a better understanding of latent space and also to solve the above-mentioned physics observables issue in Section 3.4.3 with a WGAN, a model combining a WGAN and a VAE is studied. The VAE part in the model ensures a meaningful latent space, and the GAN part generates high fidelity samples. The training dataset consists of electron shower images of 5 × 5 pixels along with energy and position labels simulated from the Geant4 standalone crystal setup. The discrete energies from 0.5 GeV-2.5 GeV energies are contained in the dataset. The 2 GeV electron showers are used to interpolated the model. The model is implemented using Tensorflow, version 1.13.0. The training of the model is explored in different training dynamics, and the two of them which provide an improvement in the result are explained in this section.

4.3.1. Training Dynamics

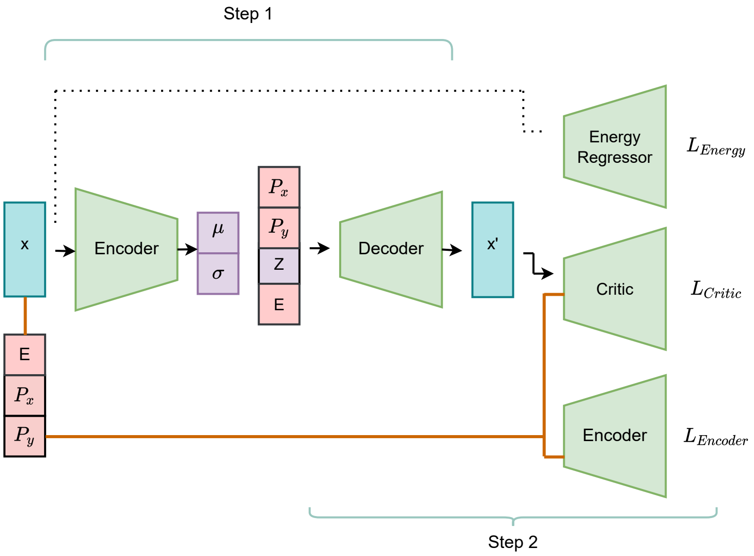


Figure 4.7.: The VAE-WGAN model. The training dynamics of the model is divided into two steps. In the first step, the VAE part of the model is trained. The second step involves the training of the decoder and the critic as a WGAN model along with the energy regressor and the encoder. Energy and position labels are given to the network during training.

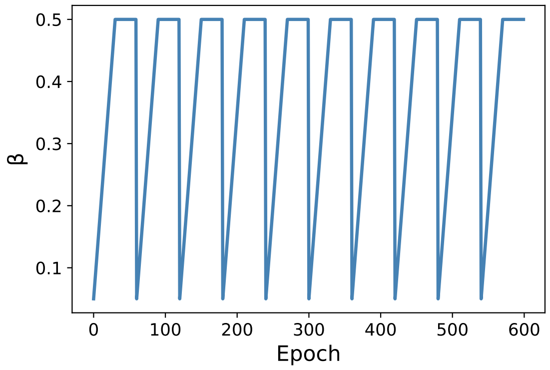


Figure 4.8.: The plot shows the value of β defined in Eqn.(4.2) which varies periodically through cyclic annealing for different epochs.

Figure. 4.7 depicts the diagram of a VAE- WGAN model. It is composed of an encoder, a decoder, a critic, and an energy regressor network. The training is performed in two steps. In the first step, the VAE part of the model is trained (i.e., the encoder and decoder) on the following loss function.

$$\mathrm{L_{VAE}} = \mathrm{L_{Reconstruction}} - \beta \mathrm{D_{KL}}[\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})||\mathrm{p}_\theta(\mathrm{z})] - \gamma \mathrm{I_q}. \tag{4.3}$$

The value of β in the equation is varied through cyclic annealing (Fig. 4.8) for different epochs instead of providing as a constant value which has shown to alleviate the KL- vanishing problem [49]. Along with this periodic β annealing, the loss function is regularized using the mutual information ($\mathrm{I_q}$) term, which together alleviates the KL vanishing problem of VAE model used here. The term $\mathrm{I_q}$ is scaled with a hyperparameter γ. The training images along with energy and position labels are given to the encoder. The decoder is given the reparametrized z variable along with energy and position labels. The primary focus of this step is on the encoder part of the model as the main objective is to develop a latent space that learns well the features of the dataset.

After training VAE for 50 epochs, the second part of the training begins where the WGAN part (i.e., the decoder and the critic) is trained along with the encoder and the energy regressor as shown in Figure. 4.7. In this part, the decoder acts as the generator network for the WGAN. The latent space vector z is provided as the random noise to the generator. The loss function for the training is defined as

$$\mathrm{Loss_G} = -\alpha \mathbb{E}[\mathrm{C}(\mathrm{G}(\mathrm{z}))] + \kappa_\mathrm{E} \mathrm{Loss_{Energy}} + \beta \mathrm{Loss_{KLD}} (4.4)$$

where $\mathrm{Loss_G}$ refers to the loss function of the generator which consists of the feedback from the critic, encoder, and energy regressor. The term C(G(z)) refers to the output of the critic from generated images. The loss functions for the critic and the energy regressor are given in Eqn.(3.5) and Eqn.(3.6) respectively. The α, κ_E and β are the scaling hyperparameters for the generator loss, energy regressor loss and the KLD loss respectively. The architecture of the optimized model are discussed in Appendix A Tab.A.1, Tab.A.3, Tab.A.6, and Tab.A.7 and the details of some hyperparameters used are given in Tab.A.19. The learning rate and the optimizers used for the network are discussed in Section 5.3. The energy regressor is trained for 250 epochs. Once the training is completed, the decoder part is used to generate the new samples.

The model is trained on the electron shower datasets of the Geant4 standalone crystal setup. The training can conducted in two different ways. In the first method, the value of β is cyclically annealed till a certain epoch around 250 and then set to value 1. When β reaches 1, the KL-loss collapses to 0 and the K-L loss is no longer contributed to the generator. In the second or alternate method, the value of β is cyclically varied throughout the training and the K-L loss is contributed to the generator loss throughout the training. Figure. 4.9 compares the Geant4 images (left column) and VAE-WGAN generated images (right column) for 0.5 GeV, 1 GeV, 1.5 GeV and 2.5 GeV energy electrons. The VAE-WGAN image shows a lesser occupancy compared to the Geant4 images. Multiple trainings are performed for both methods and the results of the best of six trainings are explained below, for the first training method where the value of β is cyclically annealed till epoch 250. The model has been trained between 500 and 1500 epochs. However, it is observed that training for a higher number of epochs doesn't bring any significant difference to the results. Different physics observables are compared for Geant4 and VAE-WGAN generated images from the best epoch.

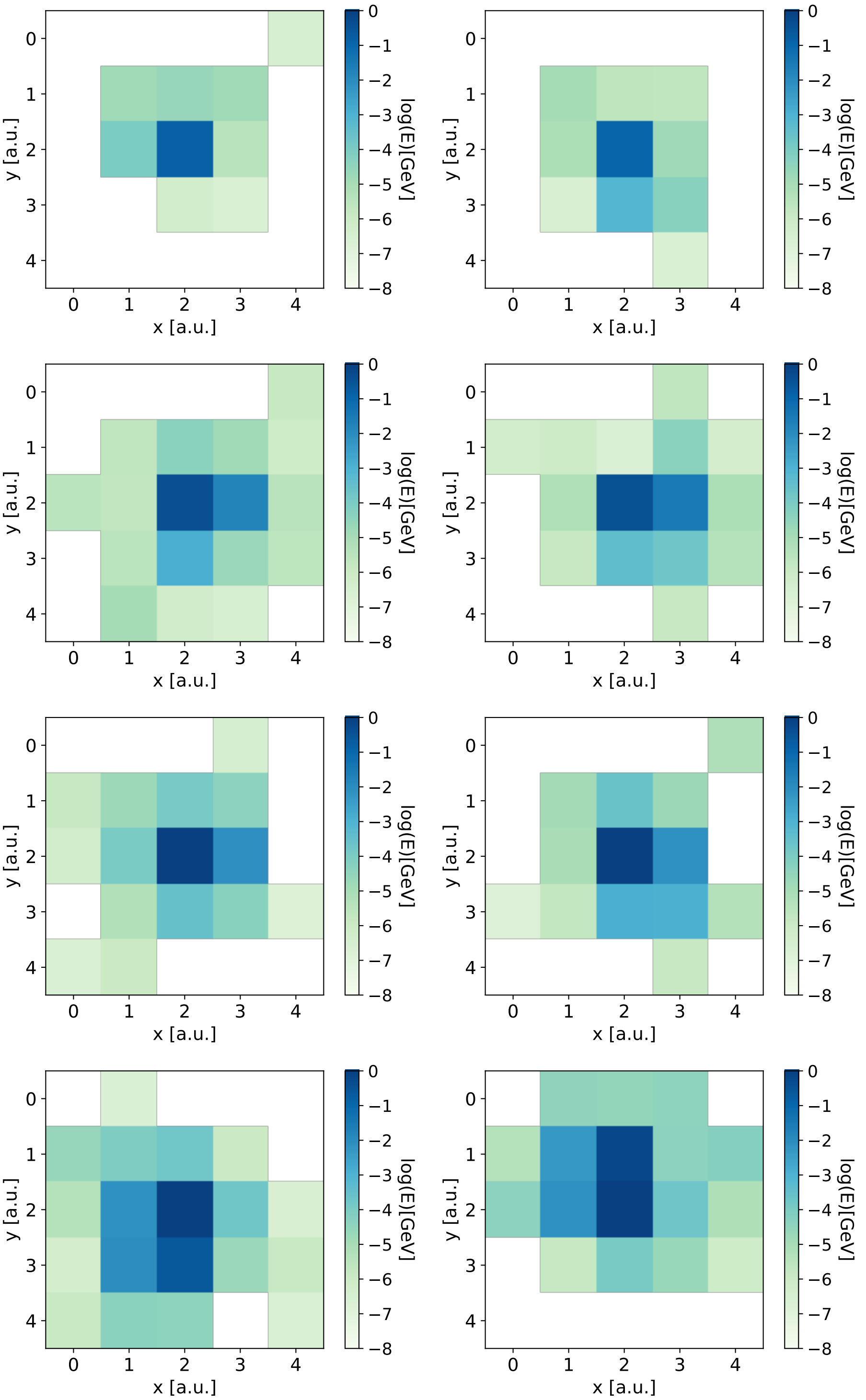


Figure 4.9.: Energy depositions in 5 × 5 crystals Geant4 (left column) and VAE-WGAN generated (right column) images of electrons for energies 0.5 GeV, 1 GeV, 1.5 GeV, 2.5 GeV (top to bottom).

4.3.2. Results

Table 4.1.: The p-value corresponding to each energies for different observables calculated using Kolmogorow-Smirnow-Test.

Observable	0.5 GeV	1 GeV	1.5 GeV	2 GeV	2.5 GeV
E_{max}	0.06	0.7	0.1	6×10^{-5}	0.01
E_1/E_9	0.004	0.003	0.38	4.22×10^{-6}	0.006
E_9/E_{21}	1.7×10^{-8}	2×10^{-30}	1×10^{-69}	3×10^{-27}	2×10^{-25}
Energy active pixels	0.99	0.8	0.99	0.36	0.99

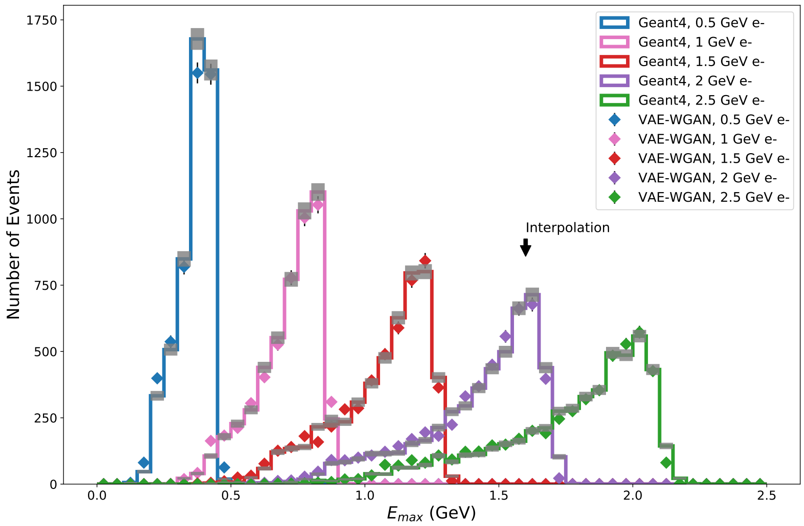


Figure 4.10.: The distribution of maximum value of energy in the 5 × 5 pixels for different energies. The distribution of 2 GeV electron is interpolated by the model.

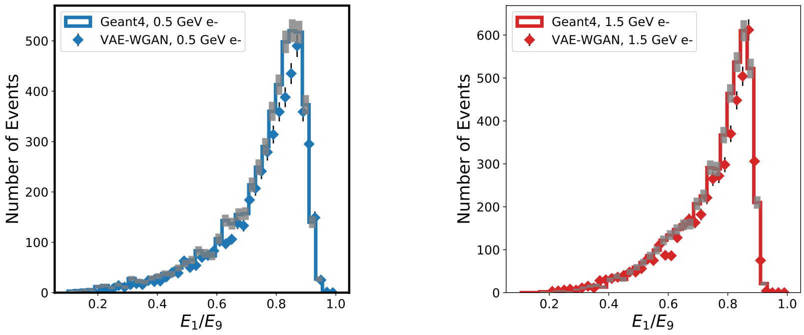


Figure 4.11.: The distribution of E_1/E_9 for 0.5 GeV and 1.5 GeV energy.

- Figure. 4.10 shows the distribution of the maximum value of energy deposited in the crystals for different energies. The Geant4 and generated distributions show

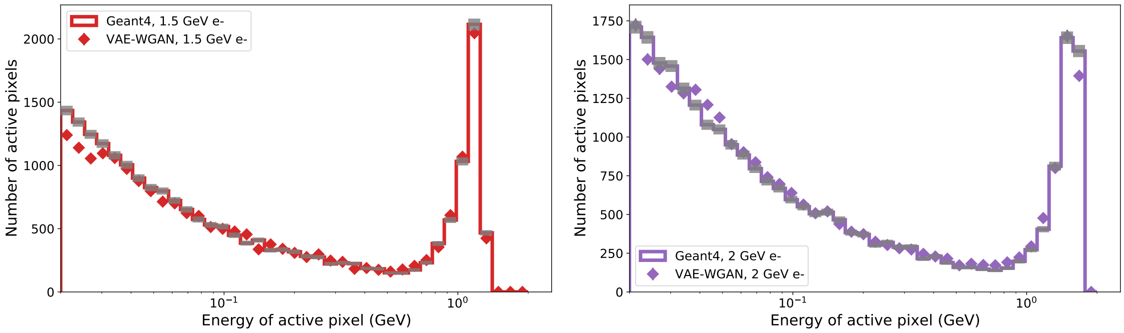


Figure 4.12.: The distribution of energies of active pixels in the 5 × 5 images for 1.5 GeV and 2 GeV electrons. The 2 GeV images are interpolated by the model.

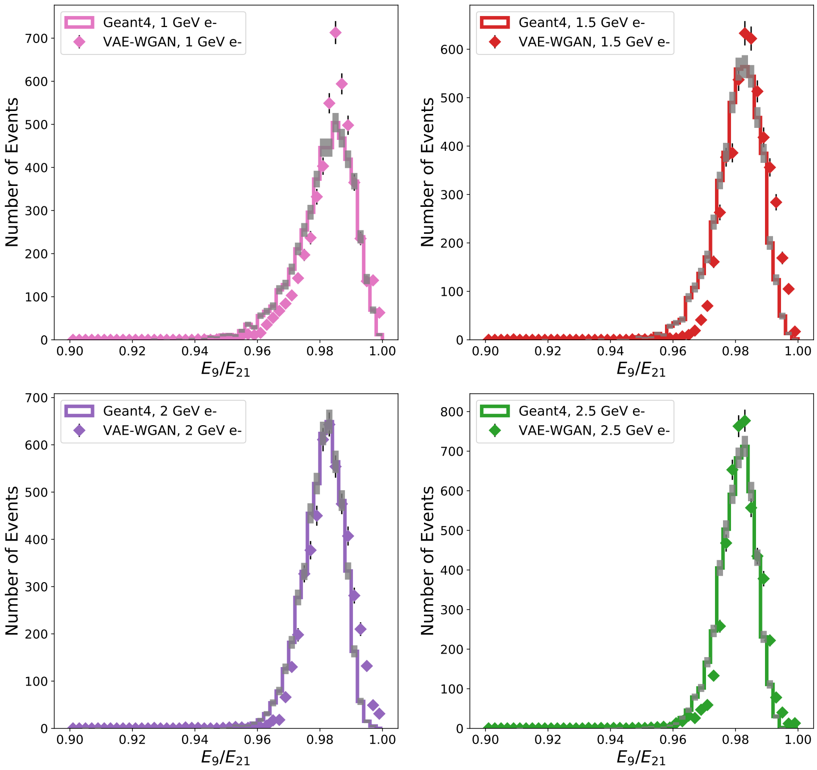


Figure 4.13.: The plots shows the distribution of E_9/E_{21} for energies 1 GeV, 1.5 GeV, 2 GeV and 2.5 GeV. The distribution shows improvement compared to the WGAN distributions.

good agreement for this variable. Table. 4.1 shows the p-value of distributions corresponding to each energy for different observables, calculated using the K-S test. The p-value of 0.5 GeV, 1 GeV, and 1.5 GeV is greater than 0.05 which indicates that the data points are from the same distribution.

- Figure. 4.11 shows the distribution of E_1/E_9 variable for 0.5 GeV and 1.5 GeV electrons. By visual assessment, this variable shows a good agreement for both the

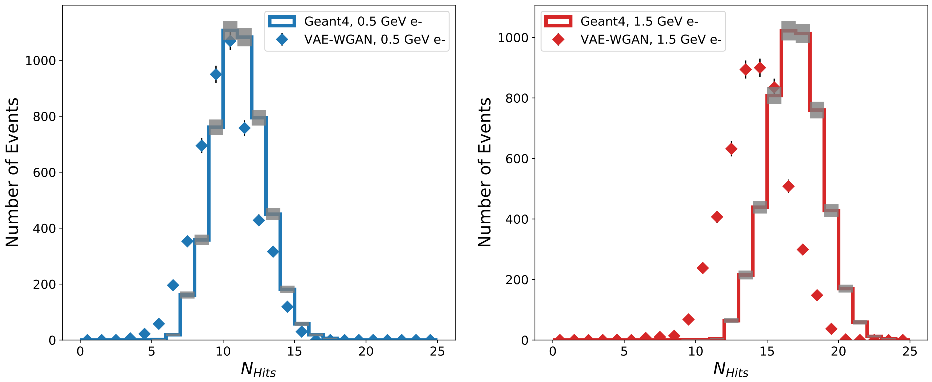


Figure 4.14.: The plots show the distribution of number of pixels with energies greater than or equal to 1 MeV deposited (N_{Hits}). The results show minor improvement compared to WGAN.

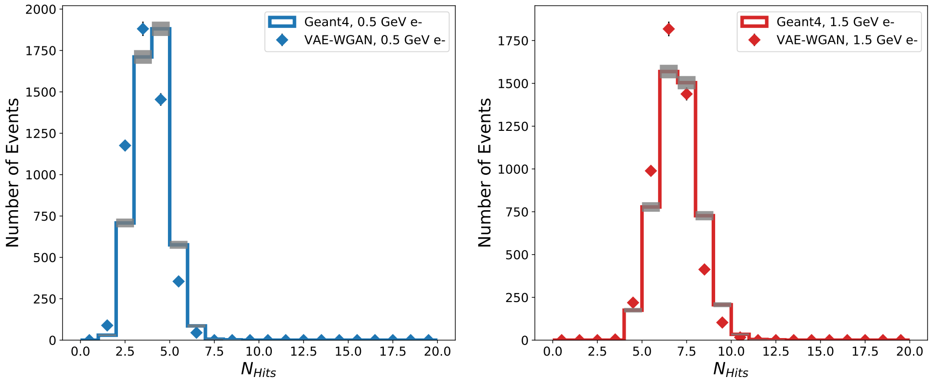


Figure 4.15.: The plots show the distribution of number of pixels with energies greater than or equal to 10 MeV deposited (N_{Hits}).

Geant4 and VAE-WGAN distributions, but the p-values are less than 0.05 for all the energies except 1.5 GeV.

- Figure. 4.12 illustrates the distribution of all the active pixel energies for 1.5 GeV and 2 GeV. Distribution is shown for active pixels with energies greater than 20 MeV. This observable shows agreement between the Geant4 and VAE-WGAN distributions. The p-values are calculated for data points with an energy threshold of 40 MeV. The p-values for all the energies are greater than 0.05 which indicates that the data points are from the same distributions.
- Figure. 4.13 shows the distribution of the physics variable E_9/E_{21} for energies 1 GeV, 1.5 GeV, 2 GeV and 2.5 GeV. The distributions of this observable are observed to be very sensitive and generates some varying distributions during the different epochs of training since it involves low-energy pixels. In Figure. 4.13, this variable shows an improvement compared to the results of WGAN-generated images. But the p-values of the data points are less than 0.05 that shows they are not from the same distribution.

- Figure. 4.14 shows the distribution of the number of pixels with energies greater than or equal to 1 MeV deposited (N_{Hits}) for 0.5 GeV (left) and 1.5 GeV electrons. This variable also shows an improvement compared to the WGAN generated images. Figure 4.15 shows the distribution for the number of hits for 0.5 GeV (left) and 1.5 GeV (right) with energy greater than or equal to 10 MeV deposited. When the threshold is increased from 1 MeV to 10 MeV, a better agreement is observed between the distributions compared to Fig. 4.14
- Even though the variable E_9/E_{21} shows an improvement, the N_{Hits} does not show a significant agreement. In some cases, it is observed that instead of generating more low-energy outer pixels in the generated images, the model tricks by generating a few outer pixels with comparably high energy in it (see Fig. 4.9). The other plots are given in Appendix B, section B.4.

4.3.3. Kernel Density Estimator (KDE) and Sampling

Decreasing the value of β for improving the information content in encoded variables causes the distribution of latent space variable z to deviate from the Standard Normal distribution as shown in the right image of Fig. 4.6. In this case, if the random noise distribution is sampled from the standard normal distribution during the generation process, the correlation between the latent space and the physics observables of the particle shower will be impaired. This is prevented by sampling the random noise vector from the encoded latent space through the Kernal Density Estimation (KDE) technique [50]. Here a large number of training datasets are encoded and the latent space z vector is collected. This latent space can be concatenated with their corresponding energy and position labels in the training dataset. A new set of latent z vectors are generated through the KDE method. When the energies in the datasets are discrete, the latent spaces are selected for these energies using a suitable window. These new sets of latent spaces are then used for generating new showers using the trained model. This retains the correlation between the latent space and the physics observables. Thus this is an alternate way to sample the random noise for the decoder during generation.

4.3.4. Computational Performance

Table 4.2.: Comparison of the computational performance of Geant4 vs VAE-WGAN for the simulation of primary electrons.

	Geant4 (CPU)	VAE-WGAN (NVIDIA© GM200)	Speed-up
0.5 GeV	$\mathcal{O}(8\,\mathrm{ms})$	0.33 ms	×24
1 GeV	$\mathcal{O}(15\,\mathrm{ms})$	0.32 ms	×47
1.5 GeV	$\mathcal{O}(24\,\mathrm{ms})$	0.31 ms	×77.4
2 GeV	$\mathcal{O}(33\,\mathrm{ms})$	0.34 ms	×97
2.5 GeV	$\mathcal{O}(40\,\mathrm{ms})$	0.32 ms	×125

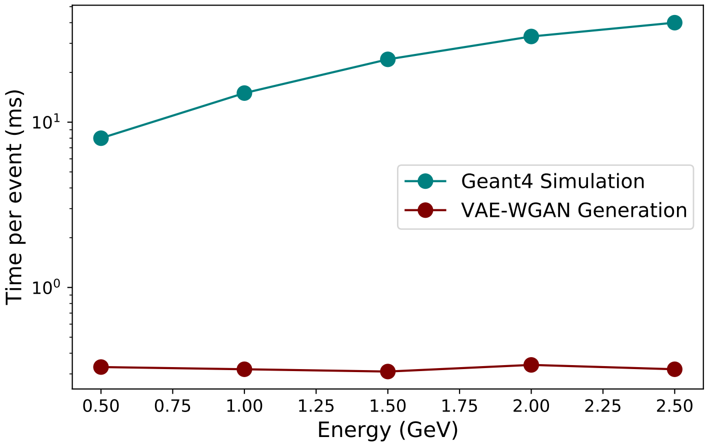


Figure 4.16.: The time required for the simulation of primary electrons of energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV via the Geant4 and VAE-WGAN approaches.

The computational time required for Geant4 and VAE-WGAN is given in the Table 4.2 for different energies of primary electrons. Figure 4.16 shows the simulation time required per event for electrons of different energies via Geant4 and VAE-WGAN simulation. The time is expressed on a logarithmic scale. It is seen that for Geant4, the generation time increases as the particle energy increases. Whereas for VAE-WGAN, the time for the generation of an event is almost the same for all the energies. The computational speed-up for each energy while using VAE-WGAN simulation is given in Table 4.2.

4.3.5. Summary

The VAE-WGAN model is used to train the electron showers of the standalone Geant4 crystal setup. The model is trained in two different methods as explained in the previous sections. The agreement between the Geant4 distribution and the VAE-WGAN generated distribution for the variable E_9/E_{21} shows an improvement compared to WGAN, but the variable N_{Hits} does not exhibits a significant improvement. Thus the precision of the showers need further improvement. The variable E_9/E_{21} is observed to be very sensitive during training. The results are chosen from the best of 6 multiple trainings.

HIGH GRANULARITY CALORIMETER RESPONSES

5

High granularity calorimeters are a promising technology foreseen to be used in many future HEP experiments [51]. Its benefits are: excellent spatial resolution, better cluster disentanglement, and better identification of electromagnetic interactions from hadronic interactions. The High granularity calorimeter upgrade for the endcaps of the CMS experiment (HGCAL) with about 6 million channels is one such implementation [34]. The VAE-WGAN model is studied with high granular crystal responses for different energies of electrons and different levels of granularity. This study is categorized into three parts, as discussed below.

- **Part 1** : Low energy electron (1-5 GeV) responses in high granular calorimeter crystal setup (30 × 30 pixels).
- **Part 2** : Low energy electron (1-5 GeV) responses in medium granular calorimeter crystal setup (10 × 10 pixels).
- **Part 3** : High energy electron (10-25 GeV) responses in high granular calorimeter crystal setup (30 × 30 pixels).

Figure 5.1 shows the diagram of the Geant4 crystal setup and its segmentation for medium and high granularity crystal blocks. The image on the left shows a block of 30 cm × 30 cm × 30 cm divided into 100 crystals (100 pixels) each of size 3 cm × 3 cm × 30 cm. This is designated as the medium granularity crystal block. The image on the right shows the same 30 cm × 30 cm × 30 cm crystal block divided into 900 crystals (900 pixels) each of size 1 cm × 1 cm × 30 cm and is designated as the high granularity crystal block. The training datasets for each part mentioned above are simulated using the Geant4 toolkit, as explained in Section 2.4.

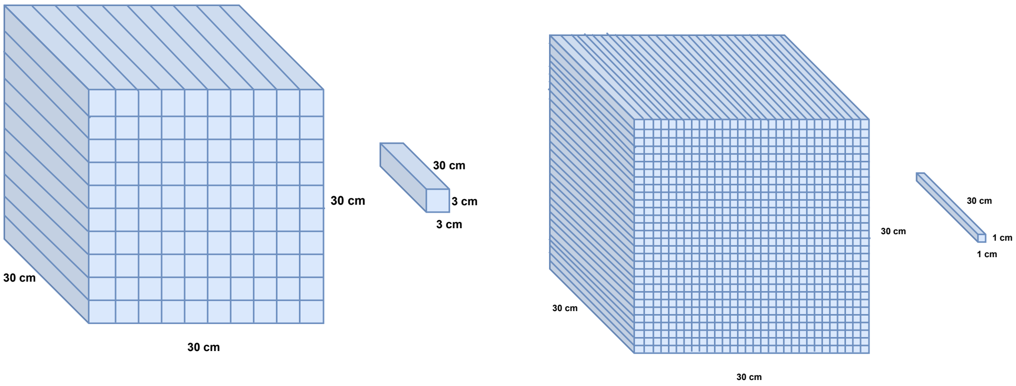


Figure 5.1.: A medium granularity crystal block with 100 pixels (left) and a high granularity crystal block with 900 pixels (right).

5.1. Low energy and high granularity

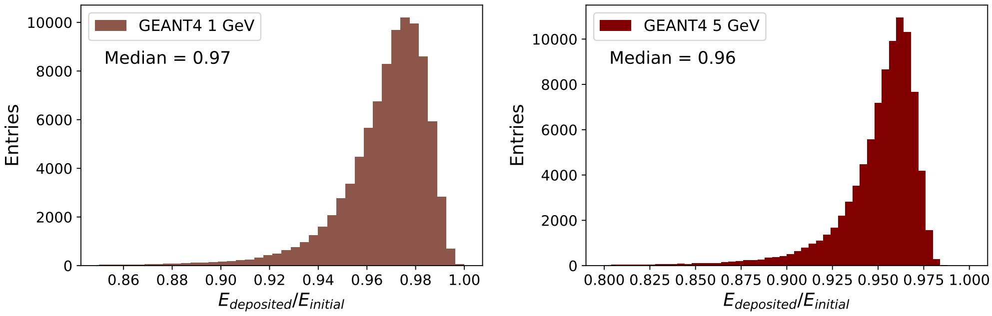


Figure 5.2.: The ratio of energy deposited in the 30 × 30 crystals to that of the energy of the primary electron for 1 GeV (left) and 5 GeV (right). The distribution shifts slightly to the left for 5 GeV electron.

Figure 5.2 shows the ratio of energy deposited in the 30 × 30 crystals to that of the energy of the primary particle (1 GeV and 5 GeV electrons). As the energy increases, the peak of the distribution shift towards the left, indicating that the percentage of energy deposited inside the crystals decreases. Figure 5.3 compares Geant4 images and VAE-WGAN generated images for 1 GeV and 5 GeV electrons in 30 × 30 pixels after the final epoch. For the Geant4 images, most of the other cells contain small energy deposits except for the few innermost cells. This results in a poor reconstruction of the energy deposit by the model.

Electrons of energy 1 GeV, 2 GeV, 3 GeV, and 5 GeV are chosen for training, which is the Belle II energy range. The granularity of the crystal setup is 30 × 30 pixels with a total of 900 channels, where each crystal has a dimension of 1 cm × 1 cm × 30 cm. The segmentation is done along the x- and y-axes without any longitudinal segmentation. The details of the architecture of the model are discussed in Appendix A Tab.A.14, Tab.A.15, Tab.A.16,

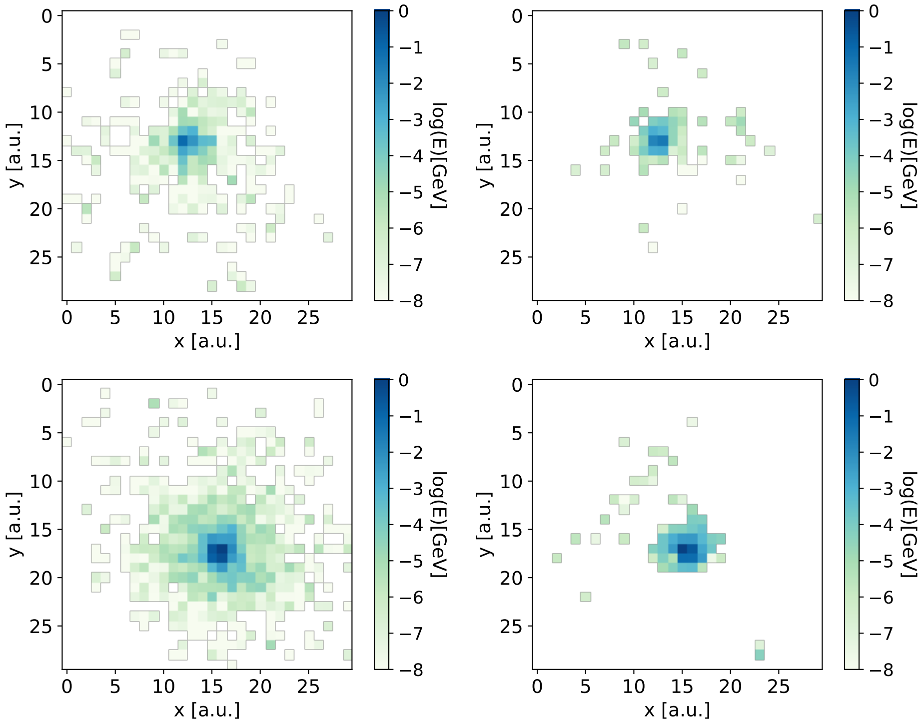


Figure 5.3.: Comparison of energy deposited for Geant4 (left column) and model generated (right column) images of electrons for energies 1 GeV and 5 GeV electrons (top to bottom) in 30 × 30 pixels.

Tab.A.17, and Tab.A.18. Since the electrons are of low energies and the number of pixels is large, the energy deposited in each pixel is minimal. This can make the reconstruction of energies by the deep learning model considerably difficult. Since the crystal size is very small, the reconstruction of the position of the primary particle can also be studied. The position of the particle gun is changed randomly between +40 mm and -40 mm from the center during the Geant4 simulation. These position values of the electron shower are given as the true position labels for the model during training.

The physics observables for the Geant4 and VAE-WGAN images are compared below.

- Figure 5.4 shows the distribution of the maximum value of energy deposited by the electrons of energies 1 GeV, 2 GeV, 3 GeV, 4 GeV, and 5 GeV in the 30 × 30 crystals. The 4 GeV electron distributions are used to interpolate the model. The Geant4 and the VAE-WGAN generated distributions show an enormous disagreement due to the low energies being deposited in each pixel.
- Figure 5.5 shows the distribution of active pixel energies for 1 GeV and 4 GeV showers. This observable also shows a disagreement between the Geant4 and VAE-WGAN showers.

In conclusion, fast simulation of electrons with energies ranging from 1 GeV - to 5 GeV (Belle II energy range) is difficult using a high granularity calorimeter, as energy deposited in each pixel would be very small.

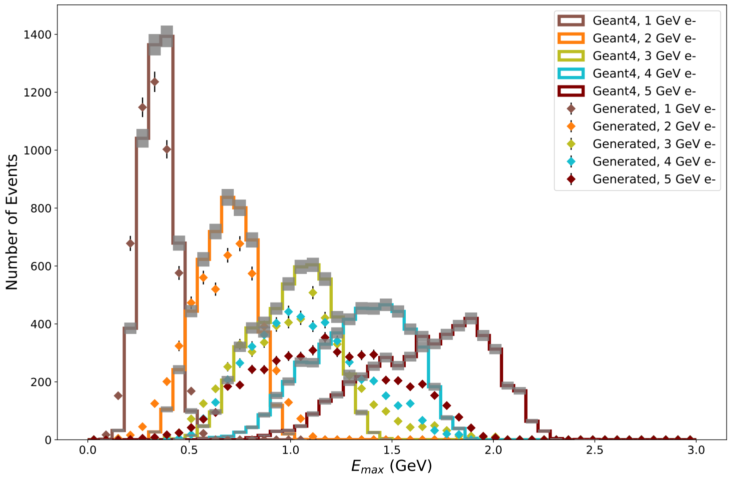


Figure 5.4.: The distribution for the maximum value of energy, E_{max} deposited in the high granularity 30 × 30 pixels. The distributions of Geant4 and VAE-WGAN have a significant disagreement. The 4 GeV showers are used for interpolating the model.

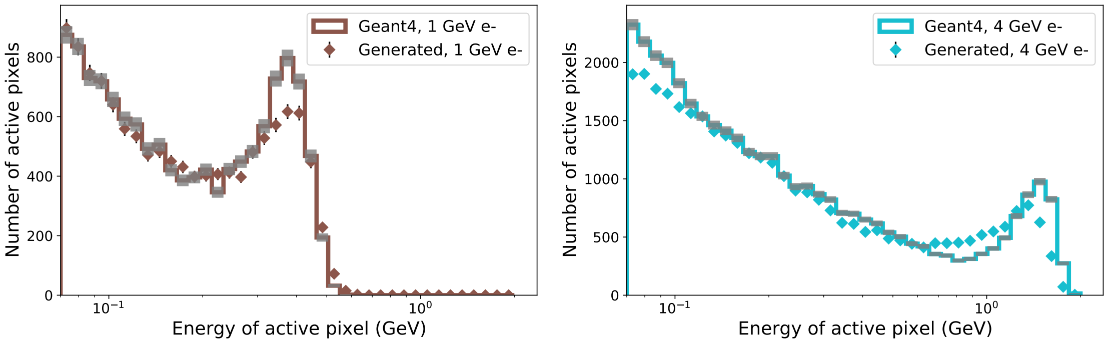


Figure 5.5.: Distributions of energies in all the active pixels of the high granularity 30 × 30 array for 1 GeV and 4 GeV electrons. The distributions show disagreement.

5.2. Low energy and medium granularity

To overcome the difficulty of reproducing low energy electron responses in high granularity pixels, as discussed in Section 5.1, the granularity of the crystal setup is reduced to 10 × 10 pixels, as shown in Fig 5.1 (left), where each crystal has a dimension of 3 cm × 3 cm × 30 cm. The segmentation is provided along the x- and y-axes without any longitudinal segmentation. So in total, there are 100 pixels. The position and energy information of the primary particle is collected as labels along with the energy deposited in the crystals. These Geant4 images are trained using the VAE-WGAN model, whose architecture is described in Appendix A Tab.A.8, Tab.A.9, Tab.A.10, Tab.A.11, Tab.A.12, and Tab.A.13. The training dynamics, learning rate, training epochs, and optimizers used for the VAE-WGAN model for medium granularity crystals are discussed in Section 5.3. These are the same for the models of medium granularity and high granularity crystals. The hyperparameters used for training the model are also discussed in Appendix A Tab.A.20. Since the number of crystals is more prominent and the size of each crystal is smaller compared to the 5 × 5 crystal images (Section 3.4.2), the position reconstruction of the primary particle becomes relevant. Hence the position regressor is also added to the network. The training of the position regressor network is discussed in Section 3.4.2. The loss from the position regressor network is added to the generator loss (see Eqn.(4.4)) with a scaling hyperparameter κ_{P}. Random vectors sampled from the standard normal distribution are used for generating new showers. The results of the medium granularity calorimeter on low-energy electrons are discussed below.

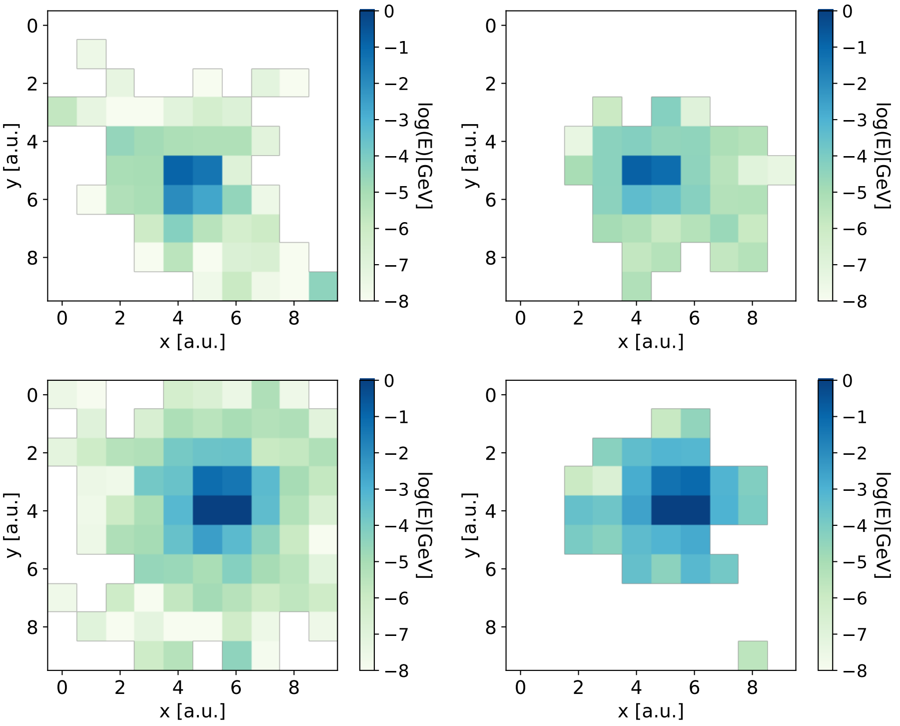


Figure 5.6.: Comparison of energy deposited for corresponding Geant4 (left column) and model generated (right column) images of electrons for energies 1 GeV and 5 GeV electrons (top to bottom) in 10 × 10 pixels.

Figure 5.6 shows the comparison Geant4 and VAE-WGAN generated images for 1 GeV and 5 GeV electron showers in 10 × 10 crystals for the final epochs. The images generated by the VAE -WGAN model have fewer active pixels than the Geant4 images. However, some of the reconstructed pixels in the outer layers contain more energy deposited in them compared to the Geant4 pixels. Thus it can affect the agreement of some physics observables. The pattern of the inner pixels is seen to be reconstructed better than the outer pixels as the energy deposited in these pixels is larger compared to the outer pixels.

The results of the position regressor are shown in Fig. 5.7. The histograms in the first row are the distribution of the reconstructed position (x and y) vs. the true position labels for the Geant4 images. The distribution shows a straight line as required, but the resolution of the pixels becomes distorted at some regions along the distributions. The histograms in the bottom row show the distribution of the reconstructed position (x and y) vs. the

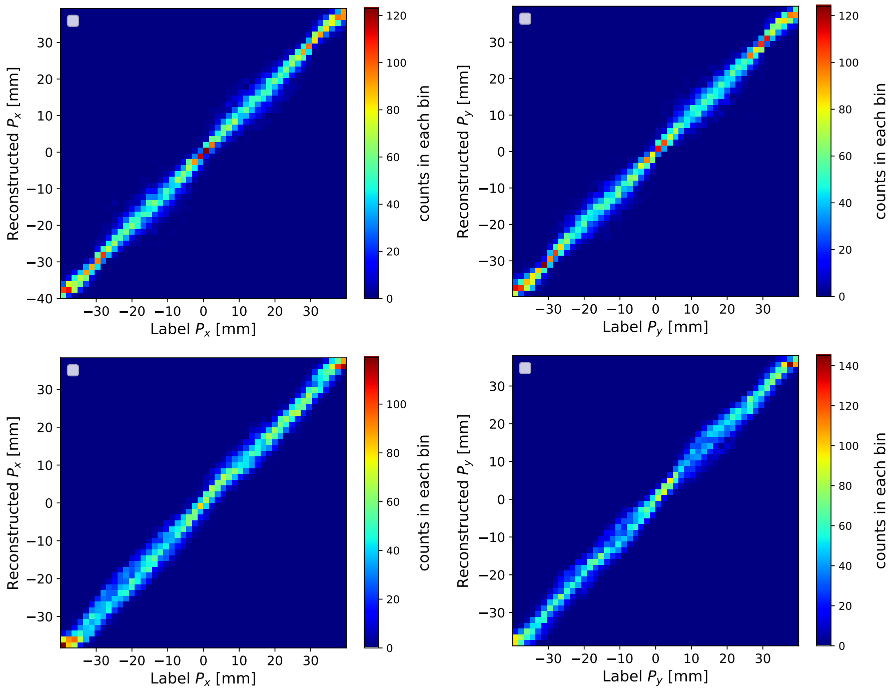


Figure 5.7.: The x (left column) and y positions (right column) reconstructed by the position regressor vs. the true labels for Geant4 images (top row) and the generated images (bottom row) for 10 × 10 pixels. The plot shows a slightly fuzzy straight line, indicating that the reconstructed position varies slightly from the true labels.

true position labels for the generated images. The resolution of the distribution gets much worse at some points compared to the Geant4 distribution. This indicates that the position regressor has reconstructed the position labels of Geant4 images better than the generated images. However, it is relevant to note that the resolution gets better as the crystal size decreases. The ‘s’ shape observed due to the large crystal size for 5 × 5 pixels in Fig. 3.11 is not visible here.

The correlation between the Geant4 and VAE-WGAN images for various physics observables is discussed below for the medium granularity crystal setup. Multiple trainings are performed and the results are taken from the best of six trainings. The model is trained for 500 epochs and the results are taken from the best epoch. The energy depositions less than 1 MeV is not considered here for the Geant4 and VAE-WGAN images.

- Figure 5.8 shows the distribution of the maximum value of energy deposited by the electrons of energies 1 GeV, 2 GeV, 3 GeV, 4 GeV, and 5 GeV in the 10 × 10 crystals. The 4 GeV electron distributions are used to interpolate the model. Since the number of pixels is lesser compared to the high granularity crystal setup (900 pixels), each pixel has a relatively higher value of energy deposited for the same energies. Hence the E_{max} distribution shows agreement between Geant4 and VAE -WGAN generated images.
- Figure 5.9 shows the distribution of all the active pixel energies for 1 GeV, 2 GeV, 3

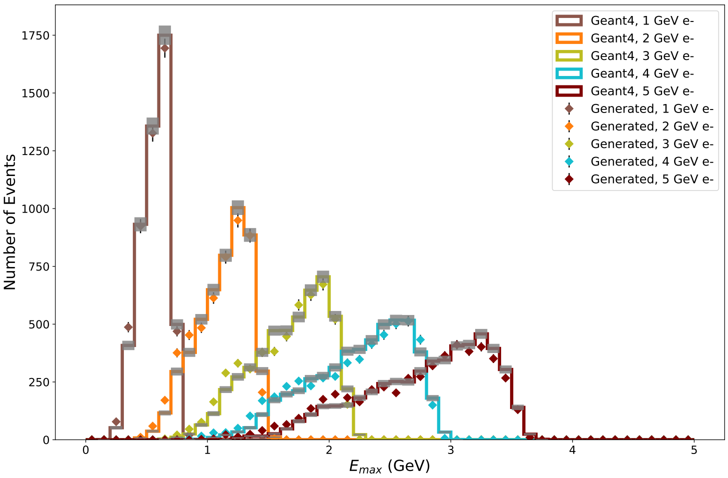


Figure 5.8.: The distribution for the maximum value of energy, E_{max} deposited in the 10×10 pixels. The distributions of Geant4 and VAE-WGAN exhibit agreement. The 4 GeV showers are used for interpolating the model.

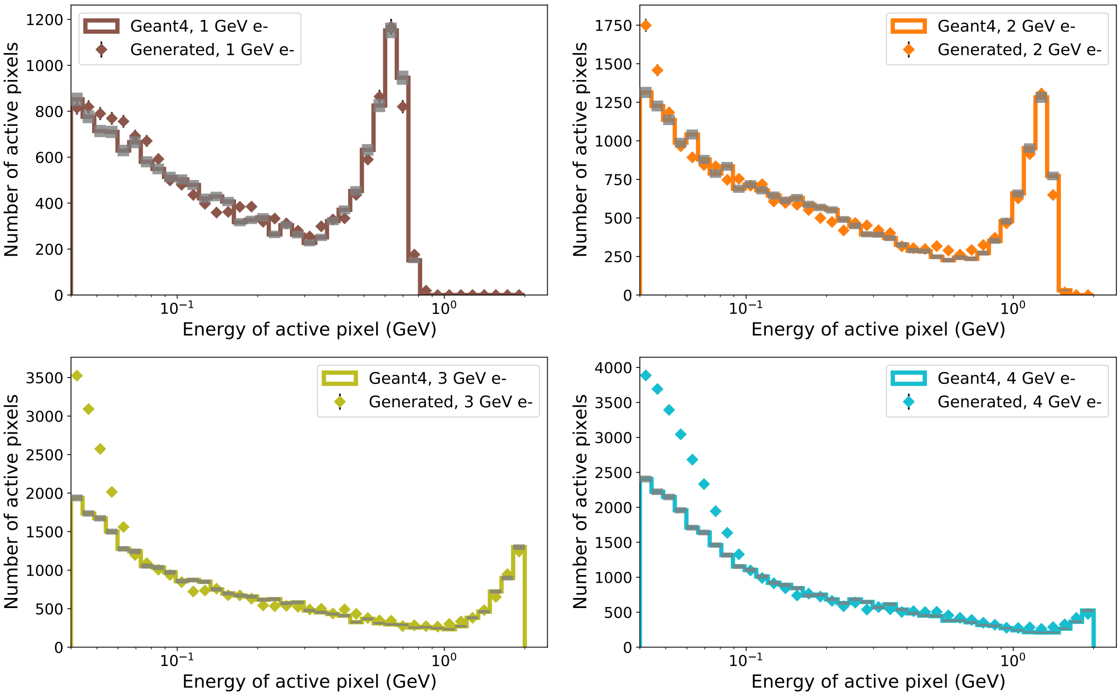


Figure 5.9.: Distributions of energies in all the active pixels of the 10 × 10 array for 1 GeV, 2 GeV, 3 GeV, and 4 GeV electrons. The distributions show disagreement towards the low energy range.

GeV and 4 GeV showers. This observable shows a disagreement between the Geant4 and VAE -WGAN showers towards the end of the distribution where the low range of energies is plotted. Energy range from 40 MeV are shown in the distribution.

- The spread of energy in the crystals is an important feature to be considered. This is evaluated using a variable E_{16}/E_{100}, which measures the energy deposited in the inner 16 pixels to that of the total 100 pixels. This is similar to the observable E_9/E_{21} discussed in the previous chapters. Both these observable measure how well the outer pixels with low energy are reconstructed in the generated images. Figure 5.10 shows the distributions of E_{16}/E_{100} observable for 1 GeV, 2 GeV, 3 GeV and 4 GeV energies. The model interpolates the distribution of 4 GeV energies. This variable is observed to be sensitive during different epochs of training and shows a variation during different epochs. The outer layers with low energy deposits are not reconstructed precisely for the generated images. Instead of generating more low-energy outer pixel, the model generates a few pixels with more energy content in it in the outer layers (see Figure 5.6).
- Figure 5.11 shows the distributions of N_{Hits} for 1 GeV and 3 GeV energies. The distribution shows poor agreement between Geant4 and model-generated images. The E_{16}/E_{100} shows a comparitively better agreement than the N_{Hits}, which indicates that even though less number of outer active pixels are reconstructed in the generated images, they comprise a higher value of energy deposit compared to the Geant4 images as shown in Fig. 5.6.

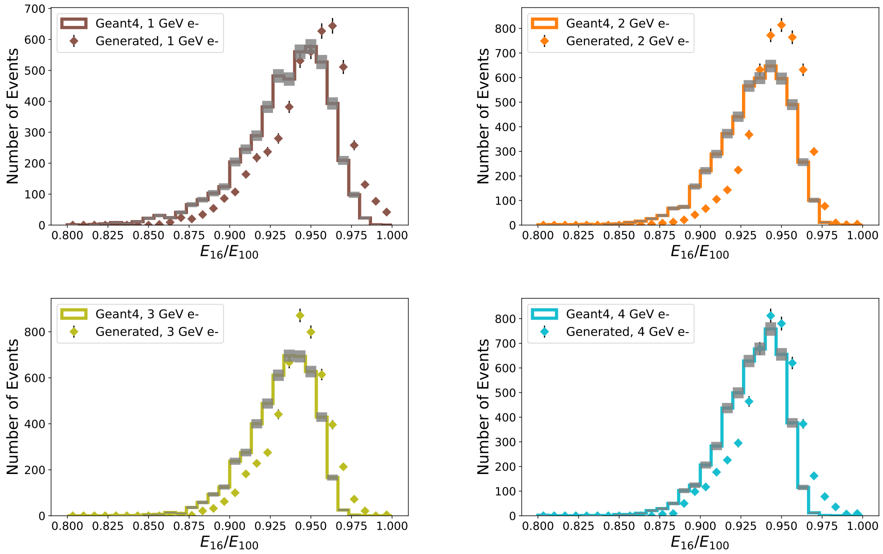


Figure 5.10.: Distributions for E_{16}/E_{100} in the 10×10 pixels for 1 GeV, 2 GeV, 3 GeV and 4 GeV electrons. The distributions does not shows a perfect agreement.

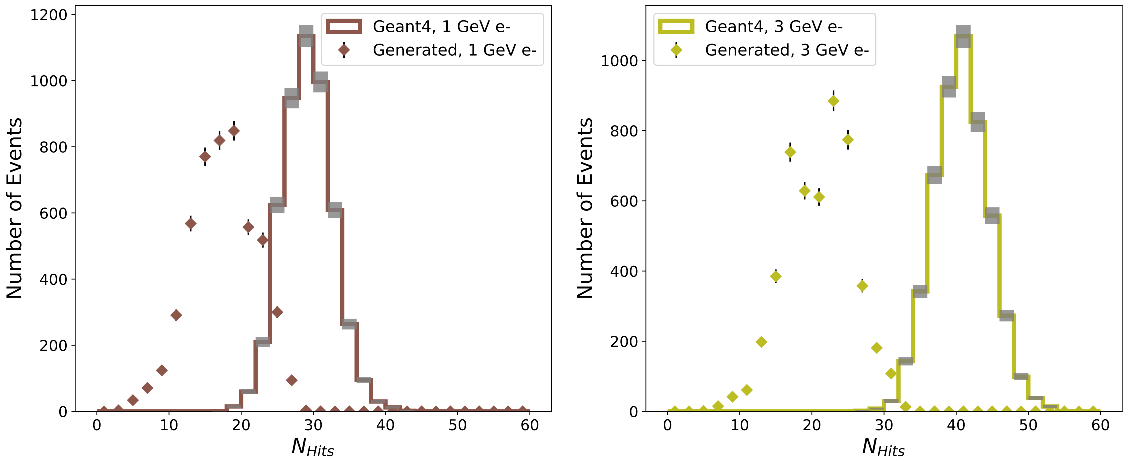


Figure 5.11.: The figure shows the N_{Hits} distribution for 1 GeV electron and 3 GeV electron showers. The plot illustrates the number of active pixels with energies equal to or greater than 1 MeV deposited.

In conclusion, for the low-energy electrons in medium granularity crystals, the model still finds it challenging to generate the outer pixels with low energy deposition with good precision. It is also observed that the reconstruction of the position information by the position regressor improves as the granularity increases.

5.3. High energy and High Granularity

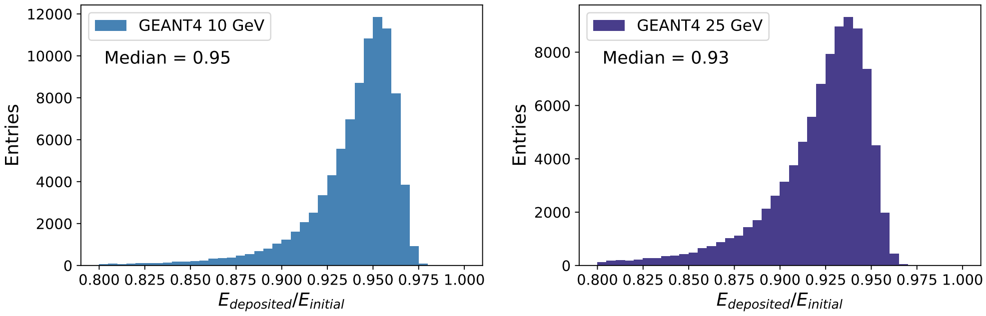


Figure 5.12.: The figure shows the ratio of energy deposited in the 30 × 30 crystals to that of the energy of the primary electron for 10 GeV (left) and 25 GeV (right). The distribution shifts slightly to the left for 25 GeV electron.

This part of the chapter explores the responses and the fast simulation of high-energy electrons 10 GeV, 15 GeV, 20 GeV, and 25 GeV in a 30 × 30 crystal (high granularity) calorimeter setup. Figure 5.12 shows the ratio of energy deposited in the 30 × 30 crystals to that of the energy of the primary particle for 10 GeV and 25 GeV electrons. For the 25 GeV electron shower, the peak of the distribution shift towards the left compared to the 10 GeV electron shower, implying that the amount of energy deposited inside the crystals decreases. Since the energy of the primary electron is high, each pixel has a considerable amount of energy deposited in it. Since the size of the crystals is smaller, adding the position regressor network to the model becomes more relevant, which can contribute the generation of fake images. The architecture used to train the model is given in Appendix A Tab.A.14, Tab.A.15, Tab.A.16, Tab.A.17, and Tab.A.18. During the training, the VAE part training is performed till the first 50 epochs, and then the WGAN, energy regressor, position regressor, and encoder network are trained to 500 epochs. The learning rate of the VAE, the position regressor, and the energy regressor are given as 1×10^{-5}, 5×10^{-5}, and 5×10^{-5} respectively. The regressor networks are trained only till the 250th epoch. The learning rate of the generator is 1×10^{-3} till the 70th epoch and is reduced to 5×10^{-4}, 2×10^{-4} and, 1×10^{-4} after 70, 90, and 100 epochs respectively. The learning rate of the critic is 5×10^{-4} till the 60th epoch and, is changed to 2×10^{-4}, 1×10^{-4} and 5×10^{-5} after 60, 80, and 100 epochs respectively. The value of β is obtained through cyclic annealing. The optimizer used for the training of VAE and the position regressor is Adam optimizer, and the training of energy regressor and WGAN loss uses RMSPropOptimizer. The hyperparameters used for the training are also given in Appendix A Tab.A.21. In this case, for generating the new showers, noise vectors are sampled from the Kernal Density Estimate of encoded latent space vector, as explained in Section 4.3.3 [50]. In Figure 5.13 the left plot shows the deviation between the standard normal distribution and one of the latent space variables that contains the high information, for 24000 samples. Thus if the standard normal distribution is provided as the random noise vector during the generation process, the correlation between the latent space and physics variables would be affected. So to preserve this correlation, new latent space variables are generated via the KDE method and sampled. The right plot in Figure 5.13 shows the agreement of one of the high information containing encoded latent z variable and the latent space variable generated via the KDE method. Energy less than 1×10^{-2} GeV is removed from both Geant4 image pixels and the generated image pixels during evaluation. Thus the threshold of energy here is 10 MeV. The model is trained for 500 epochs.

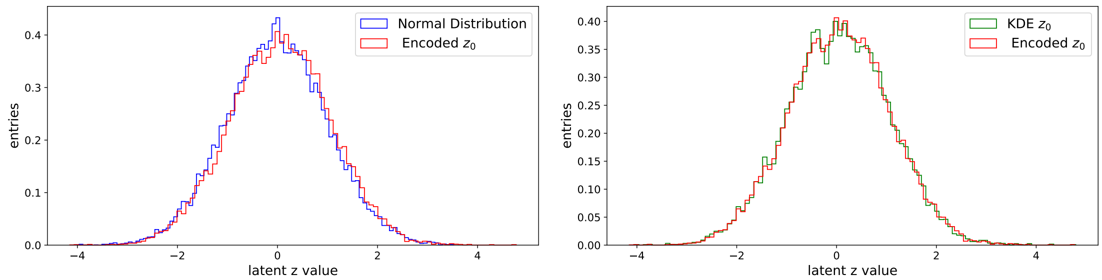


Figure 5.13.: The left plot shows a standard normal disribution and one of the high KLD latent variable. The right plot shows the distribution of one of the high KLD latent variable and the latent variable generated through Kernal Density Estimation.

Figure 5.14 shows the corresponding Geant4 and VAE-WGAN generated images for 10 GeV and 25 GeV electron showers in 30 × 30 crystals. The images generated by the VAE-WGAN model have fewer active pixels than the Geant4 images by visual inspection.

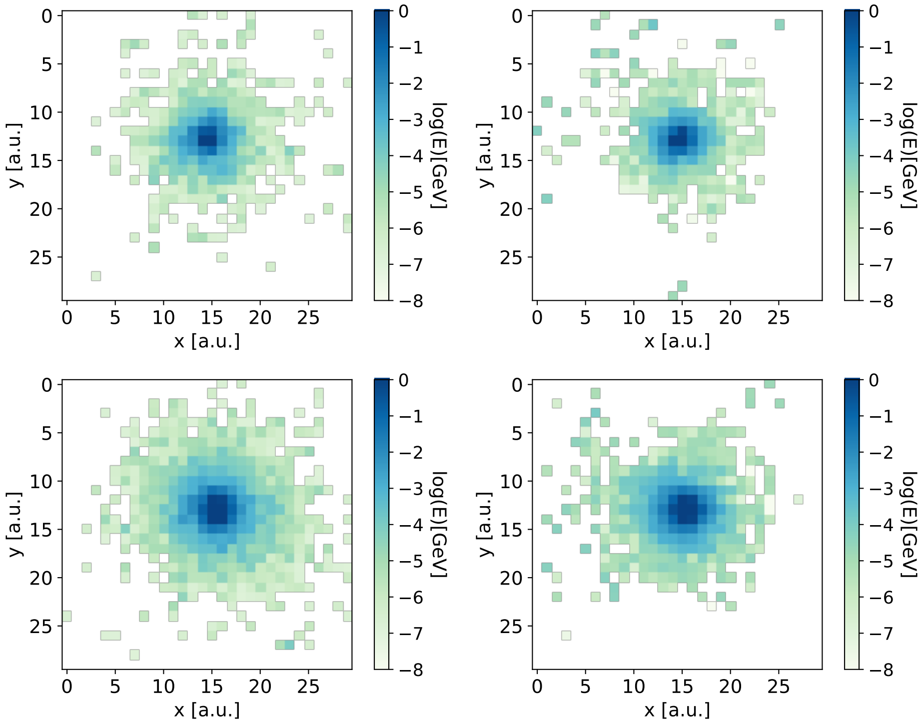


Figure 5.14.: Energy deposited for corresponding Geant4 (left column) and model generated (right column) images of electrons for energies 10 GeV and 25 GeV electrons (top to bottom) in 30 × 30 pixels.

Table 5.1.: The p-value from K-S test corresponding to energies for different observables.

Observable	10 GeV	13 GeV	15 GeV	20 GeV	25 GeV
E_{max}	0.46	0.13	0.56	0.46	0.003
Energy active pixels	0.02	3×10^{-14}	4×10^{-15}	9×10^{-21}	9×10^{-41}

Thus it can affect the N_{Hits} observable. However, the pattern of the inner pixels is seen to be reconstructed well as the energy deposited in these pixels is larger than the outer pixels.

The correlation between the Geant4 and VAE-WGAN images for various physics observables is discussed below for the high energy-high granularity crystal setup. The p-value of the distributions for various physics observables is calculated using the K-S test and is given in Tab.5.1. Multiple trainings are performed and the results from best out of 6 trainings are shown below.

- Figure 5.15 illustrates the plot for true positions vs. the positions estimated (P_x, P_y) by the position regressor network for simulated and generated images. The straight-line plot indicates that the position regressor reconstructed the x and y positions of the primary particle. This is different from the position reconstruction of 5 × 5 pixels with an ‘s’ shape in the middle. The reason being the crystal size small, which facilitates the network to reconstruct the positions of the primary particle.

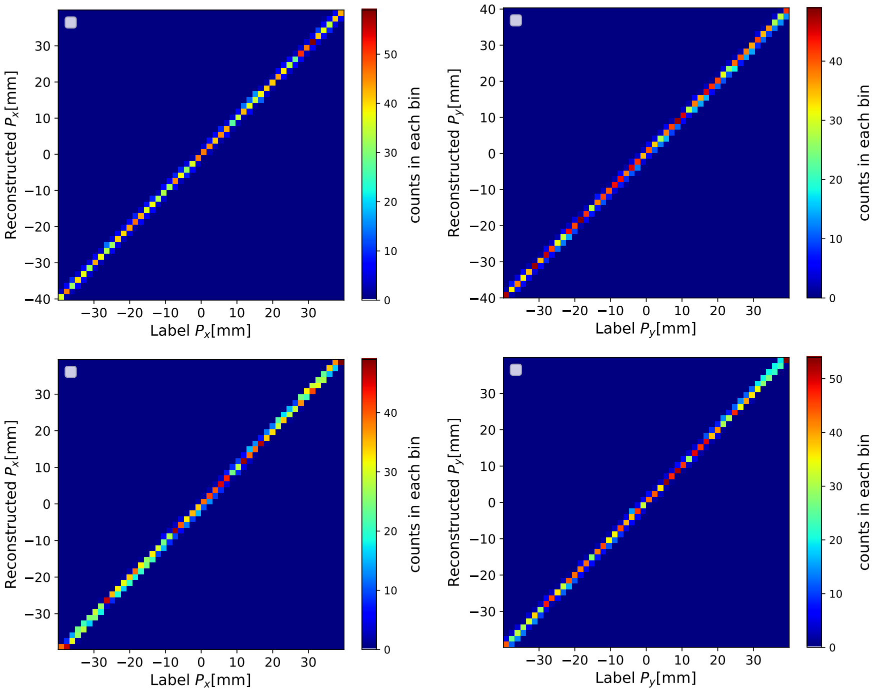


Figure 5.15.: The x (left column) and y positions (right column) reconstructed by the position regressor for Geant4 images (top row) and the generated images (bottom row). The plot shows a straight line which indicates that the position regressor reconstructs the position labels of the images.

- Figure 5.16 shows the distribution of the maximum value of energy deposited by the electrons of energies 10 GeV, 13, 15 GeV, 20 GeV, and 25 GeV in the 30 × 30 crystals. The distribution is plotted on 25000 evaluation samples. The 13 GeV electron distributions are used to interpolate the model. Here, most of the pixel has a relatively good value of energy deposited in it. The E_{max} distribution shows agreement between Geant4 and VAE -WGAN generated images. The Tab.5.1 shows the p-value for E_{max} for all the energies. Except for 25 GeV, all other showers have a value greater than the level of significance of 0.05. Hence it indicates that the data samples of these energies are from the same distribution.
- Figure 5.17 shows the distribution of all the active pixel energies for 10 GeV, 13 GeV, 15 GeV, and 25 GeV showers. The model interpolates the 13 GeV showers. This observable shows an agreement between Geant4 and VAE-WGAN generated images by visual inspection, whereas the calculated p-values are less than 0.05 which implies that the data points are not from the same distribution. The pixels with energy equal to or greater than 1×10^{-2} GeV are considered in the figure.
- The spread of energy in the crystals is evaluated here using a variable E_{196}/E_{900}, which measures the energy deposited in the inner 196 pixels to that of the total 900 pixels. This observable can convey how well the outer pixels with low energy are reconstructed in the generated images. Figure 5.18 shows the distributions of E_{196}/E_{900} observable for 10 GeV, 13 GeV, 15 GeV and 25 GeV energies. The model

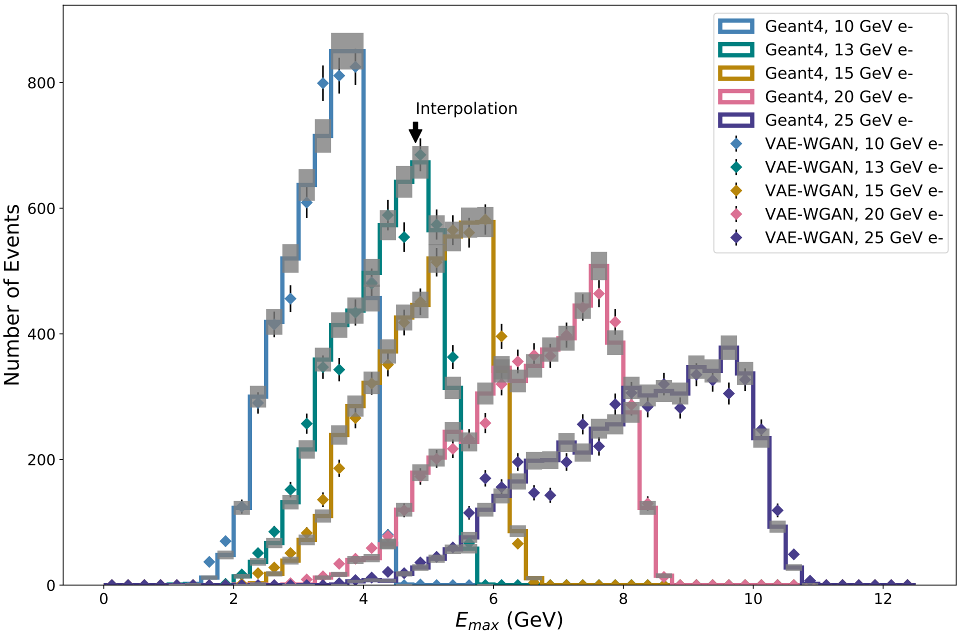


Figure 5.16.: The distribution for the maximum value of energy, E_{max} deposited in the 30×30 pixels. The distributions of Geant4 and VAE-WGAN shows agreement. The 13 GeV showers are used for interpolating the model.

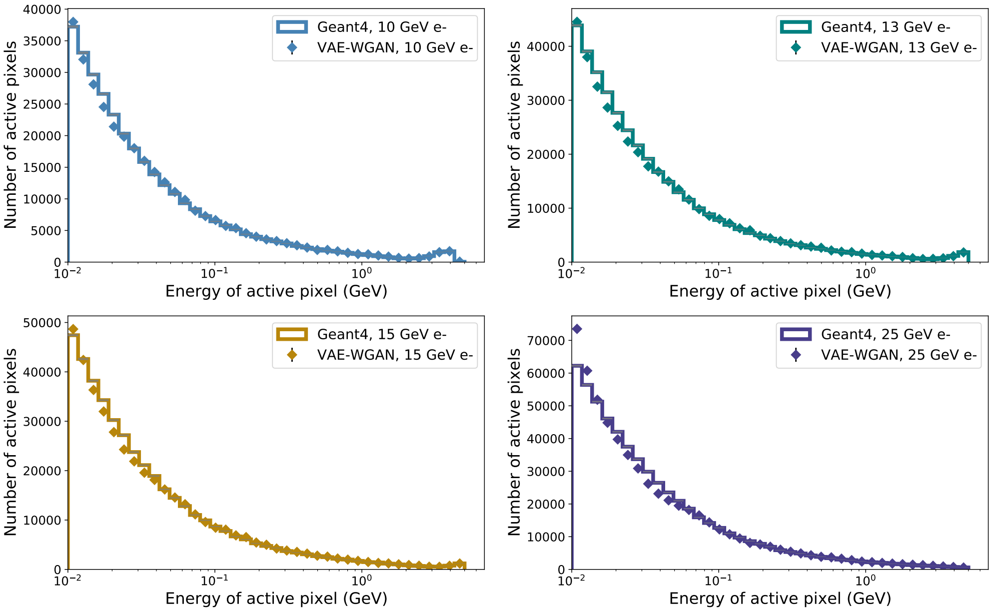


Figure 5.17.: Distribution of energies of all the active pixels of the 30×30 array for 10 GeV, 13 GeV, 15 GeV and 25 GeV electrons. The distributions show agreement between Geant4 and VAE-WGAN generated images.

interpolates the distribution of 13 GeV energy. The distributions does not show a precise agreement. The distribution of this variable is observed to be sensitive and shows some variation during different epochs.

- Figure 5.19 shows the distributions of N_{Hits} for 10 GeV, 13 GeV, 20 GeV and 25 GeV energies. The distribution shows the lack of precision for the model-generated images.

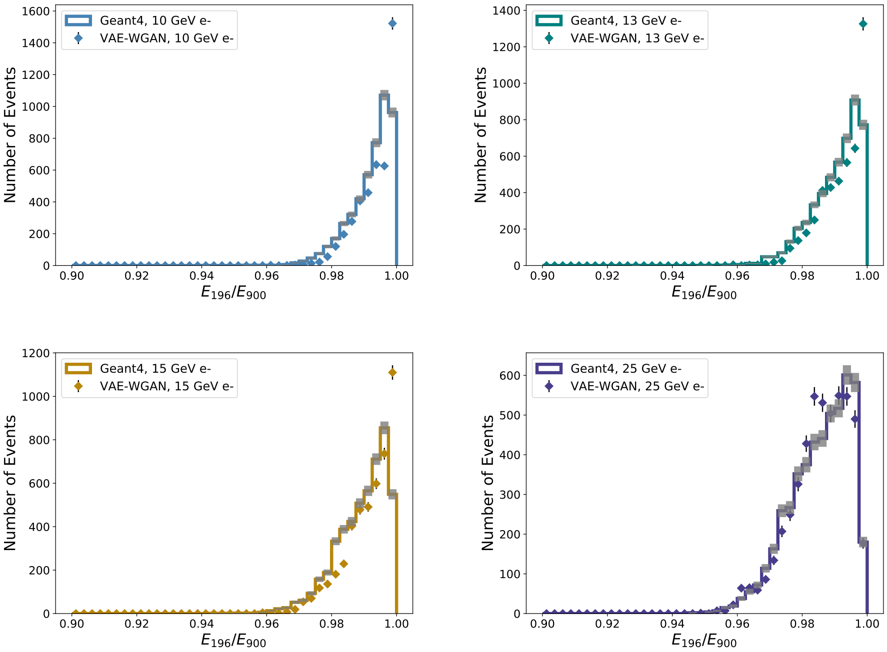


Figure 5.18.: Distributions for E_{196}/E_{900} in the 30 × 30 pixels for 10 GeV, 13 GeV, 15 GeV and 25 GeV electrons. The distributions does not shows a perfect agreement.

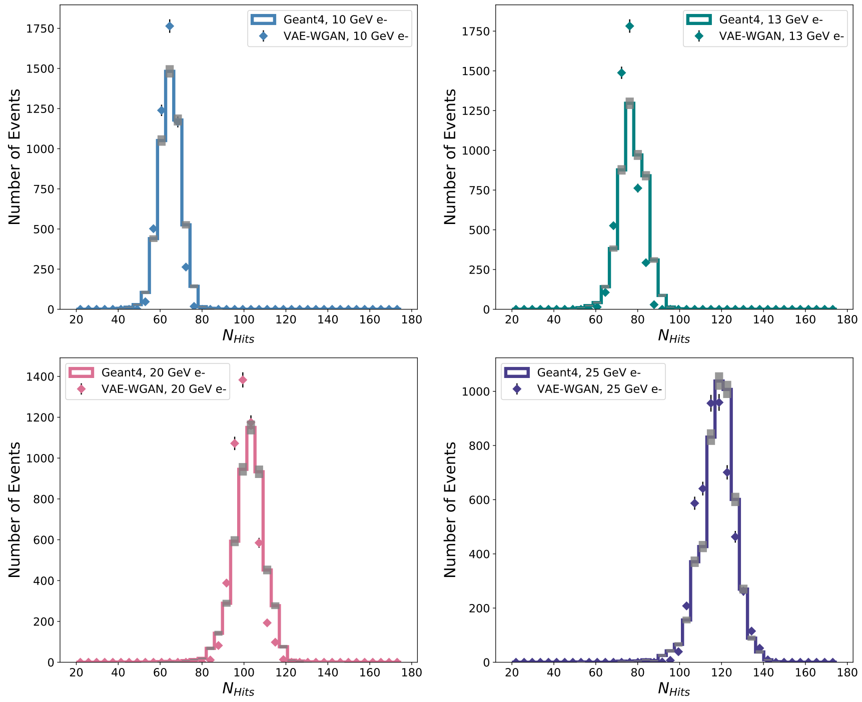


Figure 5.19.: The figure shows the $\mathrm{N_{Hits}}$ distribution for 10 GeV, 13 GeV, 20 GeV and 25 GeV electron showers. The plot illustrates the number of active pixels with energy equal to or greater than 10 MeV deposited.

In conclusion, for the high-energy electrons in 30 × 30 crystals, high-energy pixels are reconstructed by the VAE-WGAN model. However, the reconstruction of low-energy pixels still needs improvement.

5.3.1. Computational Performance

Table 5.2.: Comparison of the computational performance of Geant4 vs VAE-WGAN for the simulation of primary electrons.

	Geant4 (CPU)	WGAN (NVIDIA© GM200)	Speed-up
10 GeV	$\mathcal{O}(245\,\mathrm{ms})$	1.6 ms	×153
13 GeV	$\mathcal{O}(353\,\mathrm{ms})$	1.43 ms	×247
15 GeV	$\mathcal{O}(370\,\mathrm{ms})$	1.5 ms	×247
20 GeV	$\mathcal{O}(523\,\mathrm{ms})$	1.5 ms	×349
25 GeV	$\mathcal{O}(2640\,\mathrm{ms})$	1.47 ms	×1796

The computational time needed for the simulation of high energy electrons in high granularity crystal block via Geant4 and VAE-WGAN-based approaches is given in Tab.5.2.

Figure 5.20 compares the time per event for primary electrons of energies 10 GeV, 13 GeV, 15 GeV, 20GeV, and 25 GeV for Geant4 and WGAN simulation. The time is expressed on a logarithmic scale. For the Geant4 approach, the time increases as the energy of the particle increases. For the WGAN approach, for all the energies, the time is almost the same. The computational speed-up acquired using VAE-WGAN is also given in Tab.5.2.

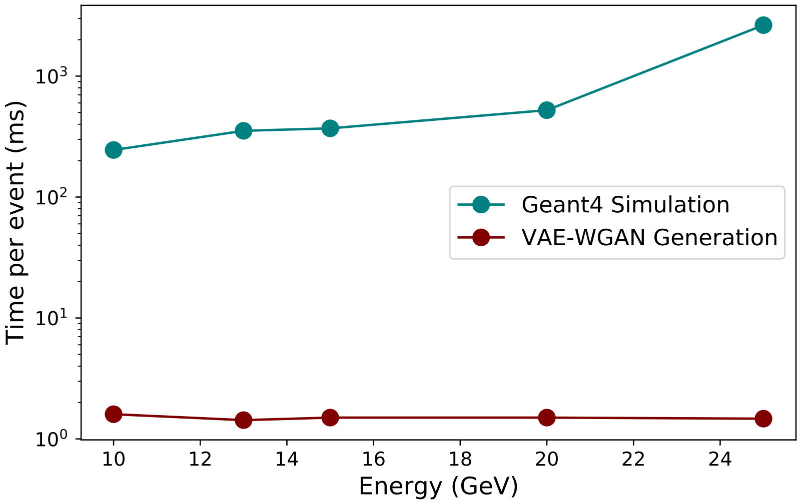


Figure 5.20.: The computational time required for the simulation of primary electrons of energies 0.5 GeV, 1 GeV, 1.5 GeV, 2 GeV, and 2.5 GeV via the Geant4 and WGAN approaches.

5.4. Summary

The VAE-WGAN model is investigated for high-granular shower responses. The fast simulation of electrons for energy range 1 GeV- 5 GeV in 30 × 30 crystals are studied in the initial stage. It is observed that the energy of the pixels is small compared to the granularity of the image. In conclusion, the fast simulation of low-energy shower responses in high granularity calorimeters is challenging for the VAE-WGAN model. In the second stage, the fast simulation of electron energy responses of 1 GeV-5 GeV in a medium granularity, i.e. 10 × 10 crystals are studied. The results are improved compared to the low-energy, high granular case, but the precision of the generated showers still needs further improvement. The performance of the position regressor network used also shows an improvement compared to the case of 5 × 5 pixels, since the crystal size is smaller, however the resolution of the position regressor output has to be improved. Finally, the fast simulation of high-energy electron responses in 30 × 30 pixels is studied. The position of the primary particle is successfully reproduced by the position regressor of the model. Hence it can be concluded that, the position reconstruction improves as the crystal size decreases. However, the precision of the generation of low-energy pixels with the model needs to be improved.

6

INFORMATION DISTILLATION GAN

Variational Autoencoders are powerful generative models for learning features of the input dataset whereas Generative Adversarial Networks are known for the synthesis of good fidelity images. Developing different architectural designs by combining these two models and experimenting with different training dynamics is highly popular nowadays in the deep learning field. This chapter explores the Information Distillation Generative Adversarial Network (ID-GAN) based on the paper [9] [52]. The model focuses on improving the disentangled representation learning and increasing the fidelity of the generated images. A diagram of the model is illustrated in Fig. 6.1. The model has a VAE part which comprises an encoder q_ϕ and a decoder p_θ and also a GAN part with a generator G_ω and a discriminator C_ψ. The training dynamics of the model are classed into two steps. The first step involves the training of the VAE alone. Since ID-GAN has a separate decoder and generator network, each can focus on its own tasks. This takes the burden off of the decoder to generate images of more complex patterns, which in turn helps the encoder to learn the disentanglement learning better. In other words, the encoder and decoder can focus entirely on disentangled representation studies.

The second step of training involves freezing the weights of the parameters of the encoder and training the generator and the discriminator networks. Unlike previous models, here the noise variable given to the generator is composed of two distinct variables, i.e., $z = (s, c)$. The variable s is the random noise vector ($s \sim p(s)$) to capture the correlated variation factors. The variable c is the disentangled latent space variable ($c \sim q_\phi(c|x)$) which is the output of the encoder network. This variable captures the independent factors of the variation. The generated images and real images are passed to the discriminator to differentiate between real and fake images. The generated images are also passed to the previously trained encoder to produce disentangled information representations. The

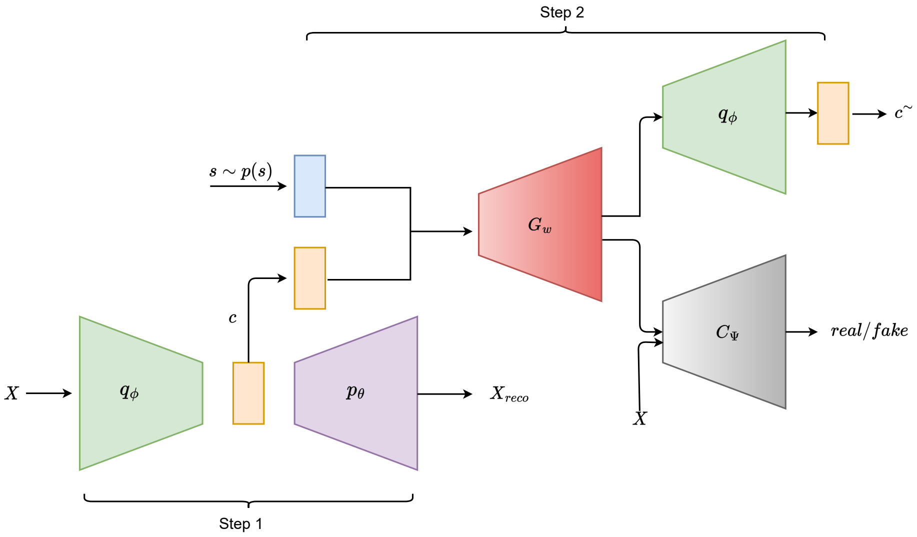


Figure 6.1.: It consists an encoder (q_ϕ), a decoder (p_θ), a generator (G_ω) and a discriminator C_ψ. The training of the model is divided into the training of the VAE part and training of the GAN part. The noise vector consists of two different variables s and c.

feedback given by the discriminator and the information loss calculated from the encoded information is used to improve the generator. For electron shower simulation, an additional energy regressor network and a position regressor network are added to the model to constrain the energy and position of the primary particle. In this chapter, the ID-GAN model from the paper [9] which uses the GAN and VAE is trained and its results are presented. In the ID-GAN paper [9], the model is trained using 64 × 64 images and then extended to 256 × 256 and 1024 × 1024 images. For the study of calorimeter responses, the model is trained using 30 × 30 images obtained from the Geant4 simulated shower responses of 15 GeV, 20 GeV, 25 GeV, and 30 GeV electrons. The model is then interpolated using 23 GeV electron showers.

6.1. IDGAN

The ID-GAN network explored here comprises VAE and the GAN network as published in [9]. The model architecture and training dynamics are inspired from [52]. The VAE part of the model is trained on the loss function given as (see Eqn.(4.2)):

$$\mathrm{L_{VAE}} = \mathrm{L_{reconstruction}} - \beta \mathrm{D_{KL}}[\mathrm{q}_\phi(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z})], \tag{6.1}$$

where each term of the equation is discussed in Section 4.1. The reconstruction loss is obtained here by calculating the mean square error of each pixel in the real image and the reconstructed image. The value of β in Eqn.(6.1) is set as 0.01. The learning rate of the encoder network and the decoder network is set to 1×10^{-4}. The optimizer used for training is Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The VAE part is trained for around

280 epochs. Once the training of the VAE is completed, the training of the GAN part is performed for the high fidelity synthesis of images based on the following objective [9]

$$\min_{\mathrm{G}} \max_{\mathrm{D}} \mathrm{L}_{\mathrm{GAN}}(\mathrm{D}, \mathrm{G}) - \lambda \mathrm{R}_{\mathrm{ID}}(\mathrm{G}), \tag{6.2}$$

where $\mathrm{L}_{\mathrm{GAN}}$ refers to the GAN loss function, defined as [9]

$$\mathrm{L}_{\mathrm{GAN}}(\mathrm{D}, \mathrm{G}) = \mathbb{E}_{\mathrm{x} \sim \mathrm{p}(\mathrm{x})}[\log \mathrm{D}(\mathrm{x})] + \mathbb{E}_{\mathrm{s} \sim \mathrm{p}(\mathrm{s}), \mathrm{c} \sim \mathrm{q}_{\phi}(\mathrm{c})}[\log(1 - \mathrm{D}(\mathrm{G}(\mathrm{s}, \mathrm{c}))], \tag{6.3}$$

and $\mathrm{R}_{\mathrm{ID}}(\mathrm{G})$ refers to the variational lower bound of mutual information between the latent variable c and reconstructed image. It is given as [9]

$$\mathrm{R}_{\mathrm{ID}}(\mathrm{G}) = \mathbb{E}_{\mathrm{c} \sim \mathrm{q}_{\phi}(\mathrm{c}), \mathrm{x} \sim \mathrm{G}(\mathrm{s}, \mathrm{c})}[\log \mathrm{q}_{\phi}(\mathrm{c}|\mathrm{x})] + \mathrm{H}_{\mathrm{q}_{\phi}}(\mathrm{c}). \tag{6.4}$$

Since the term q_{ϕ} is fixed in Eqn.(6.4), the term $H_{q_{\phi}}(c)$ is considered as a constant. The weights of the encoder network are frozen during the second step of training. For the case of the electron shower dataset, to train the generator network, additional feedback from energy regressor and position regressor networks are considered in the Eqn.(6.2), as explained in Section 3.4.1 and Section 3.4.2. The learning rate used for the generator, the discriminator, the energy regressor, and the position regressor are 1×10^{-3}, 5×10^{-4}, 5×10^{-5}, and 5×10^{-5} respectively. The optimizer used for all the networks trained for the GAN part is Adam with $\beta_1 = 0$ and $\beta_2 = 0.999$. The weight κ_{E} for the feedback from the energy regressor network added to the generator loss is set as 0.01. The weight κ_{P} for the feedback from the position regressor network added to the generator loss is set as 0.01. The weight λ for the encoder information is set as 1×10^{-5}. The details of the architectures of the optimized networks are given in Appendix A Tab.A.22, Tab.A.23, Tab.A.24, Tab.A.25, and Tab.A.26. The GAN part is trained for about 350 epochs. The threshold energy of a pixel is 10 MeV. Energy less than 10 MeV are not considered. The model is implemented and trained using Pytorch, version 1.9.0. The results of the ID-GAN are given below.

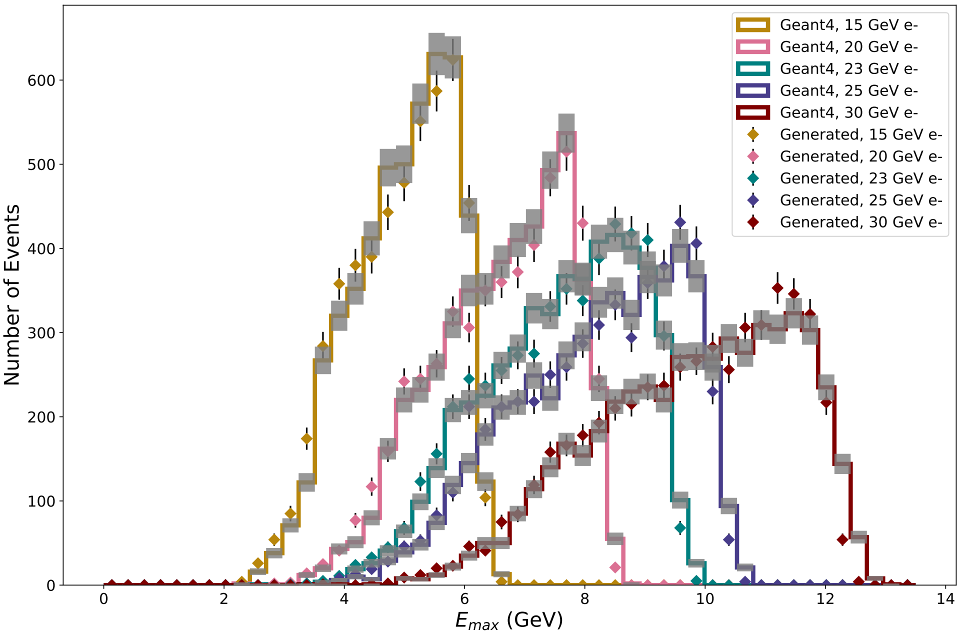


Figure 6.2.: The distribution for the maximum value of energy, E_{max} deposited in the 30×30 pixels. The distributions of Geant4 and ID-GAN images show agreement. The 23 GeV generated showers are interpolated by the model.

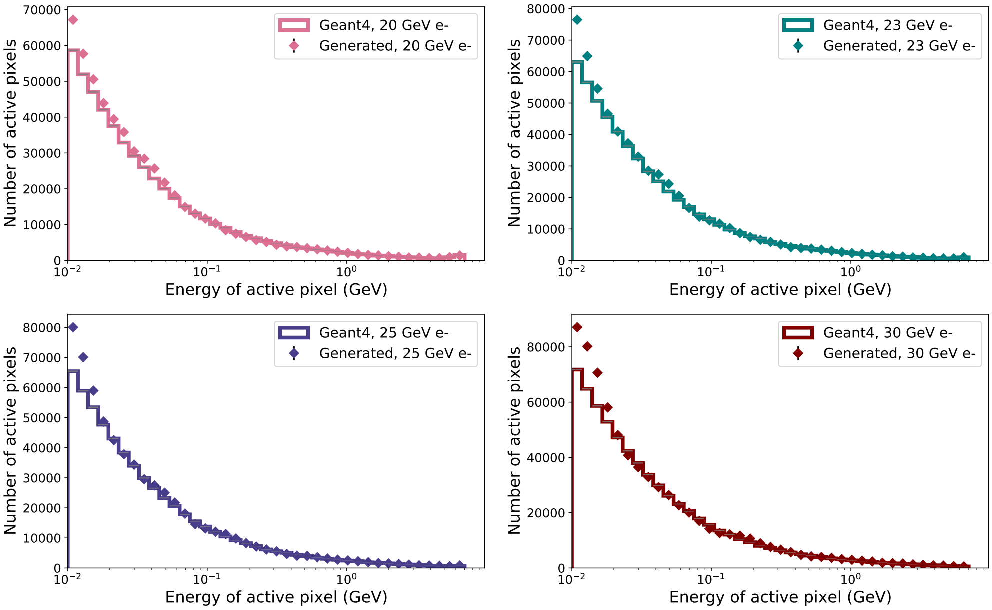


Figure 6.3.: Distribution of all the active pixel energy in 30×30 array for 20 GeV, 23 GeV, 25 GeV, and 30 GeV electrons. The distributions show disagreement towards low energy region.

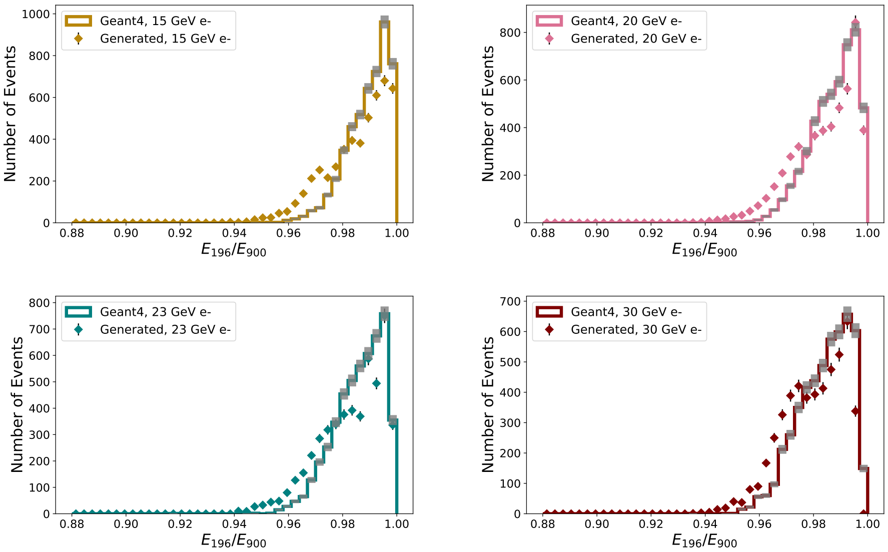


Figure 6.4.: The plots for the observable E_{196}/E_{900} for 15 GeV, 20 GeV, 23 GeV, and 30 GeV electrons. The distributions show disagreement between the Geant4 and ID-GAN generated images.

- Figure 6.2 shows the maximum value of energy deposited by the electrons of energies 15 GeV, 20 GeV, 23 GeV, 25 GeV, and 30 GeV in the 30 ×30 crystals. The distributions are plotted using 25000 evaluation samples. The 23 GeV electron is used to study the interpolation of the model. The E_{max} distributions show agreement between Geant4 and ID-GAN results.
- Figure 6.3 shows the distribution of all the active pixel energies for 20 GeV, 23 GeV, 25 GeV, and 30 GeV electron showers. Here the observable shows an agreement between Geant4 and ID-GAN generated images for higher energy ranges and starts to diverge towards the low energy range.
- The spread of energy in the crystals is evaluated using the observable E_{196}/E_{900}. Figure 6.4 shows the distributions of E_{196}/E_{900} observable for 15 GeV, 20 GeV, 23 GeV, and 30 GeV energies. This observable shows a disagreement between the Geant4 and the ID-GAN generated images.
- Figure 6.5 shows the distributions of N_{Hits} for 15 GeV, 20 GeV, 23 GeV, and 30 GeV energies. The agreement between Geant4 and IDGAN-generated images for the distribution is yet to be improved. The pixels with energy less than 10 MeV are not considered during the evaluation.

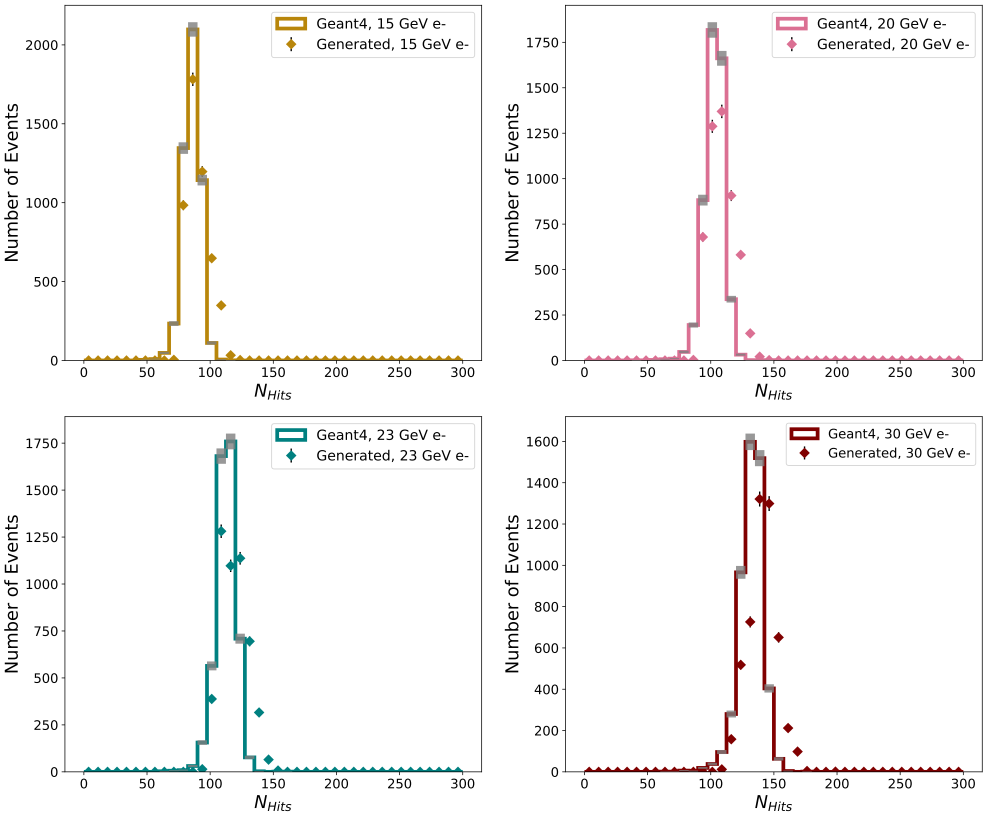


Figure 6.5.: The figure depicts the distribution of number of active pixels in 30 × 30 images. The distribution shows a disagreement between the Geant4 and the ID-GAN generated showers.

6.2. Analysis of the Datasets

For the studies of fast simulation of calorimeter responses, datasets with different pixel sizes and energy ranges are used. The different datasets are as follows:

- Images of 5 × 5 pixels with energy deposits ranging from 0.5 GeV to 2.5 GeV.
- Images of 10 × 10 pixels with energy deposits ranging from 1 GeV to 5 GeV.
- Images of 30 × 30 pixels with energy deposits ranging from 1 GeV to 5 GeV.
- Images of 30 × 30 pixels with energy deposits ranging from 10 GeV to 25 GeV.

For the β VAE model, the value of β_{KLD} plays a vital role in regularizing the encoded latent space. This coefficient balances the KL term and the reconstruction term in the ELBO loss. The KL loss term (KLD) indicates the information contained in the latent space. The value of KLD increases when the β_{KLD} value decreases resulting in the latent space distribution deviating from the Standard Normal distribution. As the β_{KLD} value increases, the value of KLD decreases, and the latent space converges towards the Normal distribution. The information contained in each of these different datasets along with evaluating the β_{KLD} required for balancing the ELBO loss are studied in this section. This is performed by passing the images to the VAE network of the ID-GAN model used for the studies and obtaining the encoded latent space variables. The KLD is calculated via

$$\mathrm{D_{KL}}(\mathrm{Z_i}||\mathrm{N}(0,1)) = -\frac{1}{2}(1+\log(\sigma_\mathrm{i}^2)-\mu_\mathrm{i}^2-\sigma_\mathrm{i}^2) \tag{6.5}$$

[50], where μ_i and σ_i refers to the encoded variables and $\mathrm{Z_i}$ is the distribution of corresponding latent space variable and $\mathrm{z_i}$ is the sampled value. The KLD thus calculated are plotted for each latent variable (inspired from [50]). The architectures of encoder and decoder networks used for training different datasets are shown in Appendix A Tab.A.22, Tab.A.23, Tab.A.27, Tab.A.28, Tab.A.29, and Tab.A.30. The optimizer used for training is Adam ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) and the learning rate of the networks is set to 1×10^{-4}.

6.2.1. Low energy 5 × 5- pixel images

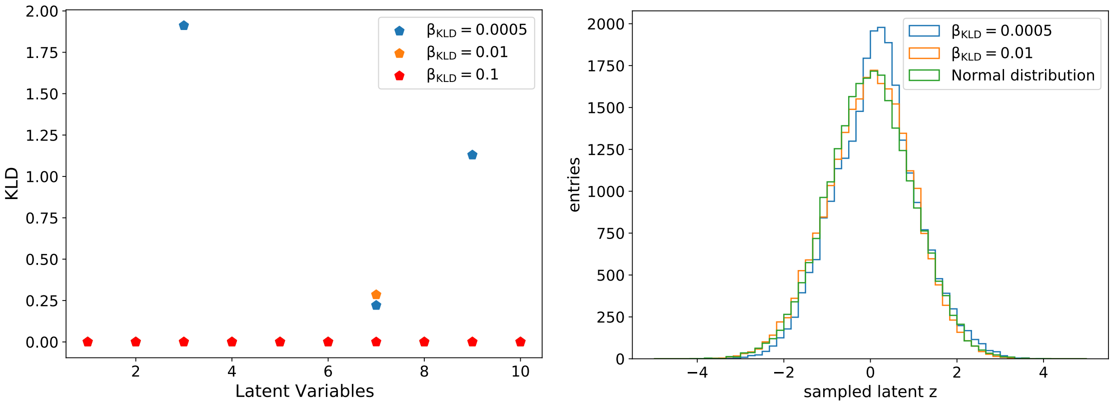


Figure 6.6.: The figure on the left shows the plot for KLD vs. latent variable for β_KLD = 0.0005, 0.01 and 0.1. The figure on the right depicts the distribution of the sampled latent z of the highest information encoded latent variable, and the standard normal distribution.

The images have energies of 0.5 GeV, 1 GeV, 1.5 GeV 2 GeV or 2.5 GeV deposited in 25 pixels. The training of the VAE is performed with $\beta_\mathrm{KLD} = 0.0005$, 0.01 and 0.1. The left plot in Fig. 6.6 depicts the KLD vs. the latent variables for $\beta_\mathrm{KLD} = 0.0005$, 0.01 and 0.1. Out of ten variables, latent variables three and nine contain most of the information at $\beta_\mathrm{KLD} = 0.0005$. For $\beta_\mathrm{KLD} = 0.01$, variable seven contains the highest information among ten variables however, the value of KLD is low. For $\beta_\mathrm{KLD} = 0.1$, the value of KLD in the plot illustrates that none of the variables contain useful information. The right plot in Fig. 6.6 compares the sampled latent z distribution of the highest information containing latent variable for 25000 showers with the standard normal distribution. For $\beta_\mathrm{KLD} = 0.0005$, these distributions deviate from one another, whereas for $\beta_\mathrm{KLD} = 0.01$, the distribution is comparatively closer to the standard normal distribution.

6.2.2. Low energy 10 × 10- pixel images

The images consist of 100 pixels and handle deposits of low energies such as 1 GeV, 2 GeV, 3 GeV, 4 GeV or 5 GeV. In other words, low energy values are distributed in medium

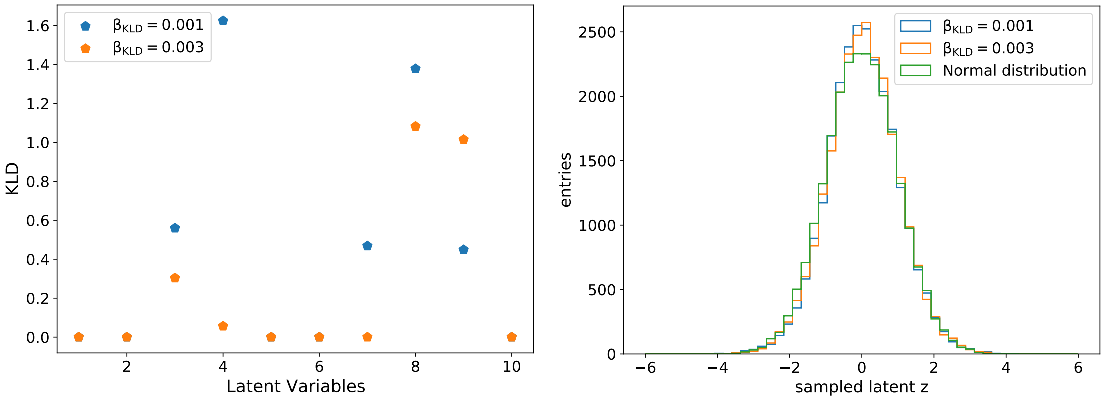


Figure 6.7.: The left plot shows KLD vs. latent variables for β_{KLD} = 0.001 and β_{KLD} = 0.003. The right plot shows the distribution of the highest information containing sampled latent z variable for different β_{KLD} values and the standard normal distribution.

granularity images. The training is executed with $\beta_{\text{KLD}} = 0.001$ and 0.003. The information contained in each latent variable is illustrated in Fig. 6.7. The plot on the left shows KLD vs. latent variables for $\beta_{\text{KLD}} = 0.001$ and 0.003. For $\beta_{\text{KLD}} = 0.001$, variables 4 and 8 encode most of the information. For $\beta_{\text{KLD}} = 0.003$, variable 8 encodes the highest information. The plot on the right compares the distribution of the sampled latent z of the highest information encoded variable for $\beta_{\text{KLD}} = 0.001$, 0.003, with the Standard Normal distribution for 25000 events. The distribution of the sampled latent z deviates from the standard normal distribution for both $\beta_{\text{KLD}} = 0.001$ and $\beta_{\text{KLD}} = 0.003$.

6.2.3. Low energy 30 × 30- pixel images

This case handles energies of 1 GeV, 2 GeV, 3 GeV, 4 GeV or 5 GeV deposited in images with 900 pixels. The energy here is very small compared to the number of pixels of the images and consequently, each pixel gets a very small amount of energy in it. The training is conducted for $\beta_{\text{KLD}} = 0.005$ and 0.01. The plot on the left of Fig. 6.8 shows KLD vs. the latent variables for $\beta_{\text{KLD}} = 0.005$ and 0.01. For $\beta_{\text{KLD}} = 0.005$, variables 7 and 8 encode most of the information. For $\beta_{\text{KLD}} = 0.01$, variable seven has the highest KLD among ten variables, but the value of KLD is low. The plot on the right compares the distribution of the highest information-containing sampled latent z variable for 25000 events for $\beta_{\text{KLD}} =$ 0.01, 0.005, with the Standard Normal distribution. The distribution of the sampled latent z variable at $\beta_{\text{KLD}} = 0.005$ deviates more from the standard normal distribution than the distribution of the sampled latent z at $\beta_{\text{KLD}} = 0.01$.

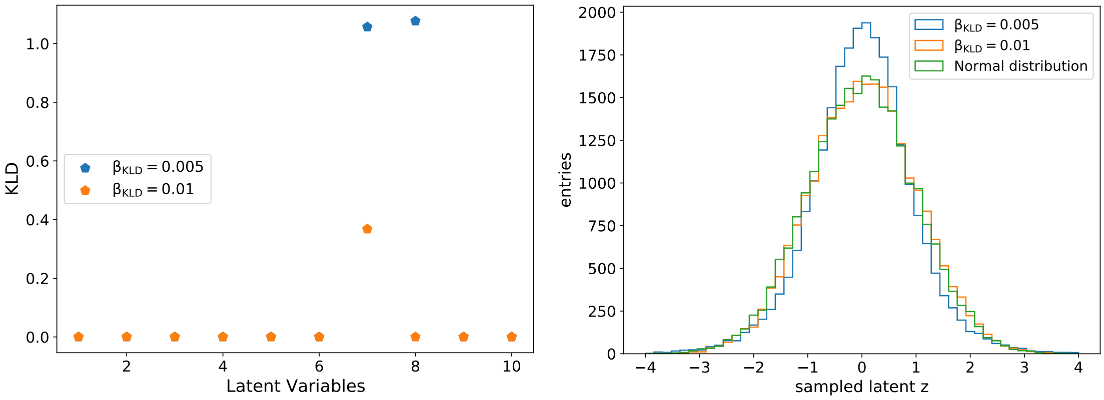


Figure 6.8.: The left plot shows KLD vs. latent variable. The right plot shows the distribution of the highest information containing sampled latent z variable and the Standard normal distribution.

6.2.4. High energy 30 × 30- pixel images

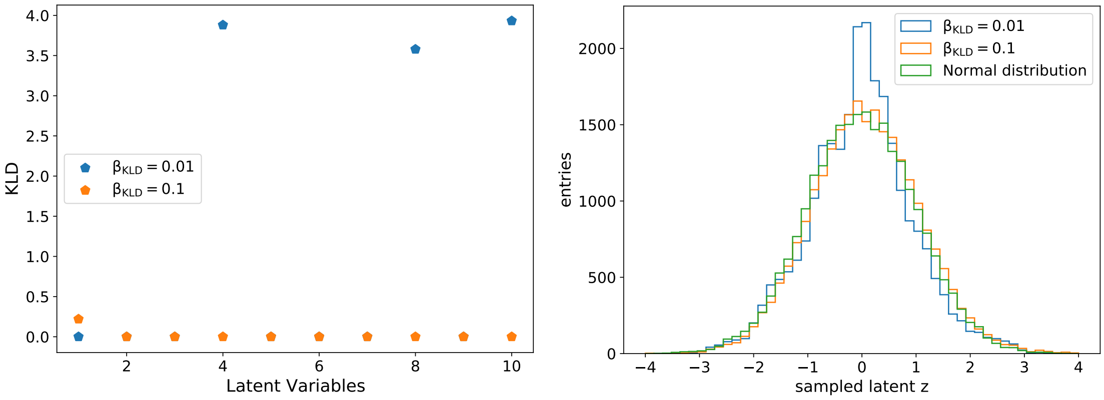


Figure 6.9.: The left plot shows KLD vs. latent variable. Three latent variable contains most of the information for the case of $\beta_{\text{KLD}} = 0.01$. The right plot shows the distribution of the highest information encoded sampled latent z variable for different β_{KLD} values and the standard normal distribution.

This case tests higher energies of 10 GeV, 13 GeV, 15 GeV, 20 GeV, or 25 GeV being deposited in 900-pixel images. Unlike the case of low energy, here energy is comparable to the pixel number, thus depositing a sufficient amount of energy in each pixel. The training of the VAE is performed for $\beta_{\text{KLD}} = 0.01$ and 0.1. The left plot in Fig. 6.9 shows KLD vs. the latent variables for $\beta_{\text{KLD}} = 0.1$ and 0.01. For $\beta_{\text{KLD}} = 0.01$, variable 4, 8 and 10 includes most information content. For $\beta_{\text{KLD}} = 0.1$, the latent variable one has a KLD value greater than 0, however, is very low. The plot on the right compares the distribution of the highest information containing sampled latent z variable for $\beta_{\text{KLD}} = 0.01$, 0.1, with the standard normal distribution. For the case of $\beta_{\text{KLD}} = 0.1$, the distribution of the sampled latent z variable deviates lesser compared to the deviation of the distribution of the sampled latent z variable at $\beta_{\mathrm{KLD}} = 0.01$, from the standard normal distribution. It is observed that the high energy 30×30- pixel image datasets have a higher KLD for a comparatively high value of β_{KLD} among the different sets of datasets.

6.3. Summary

This chapter explores the fast simulation of high-granular high-energy electron shower dataset using ID-GAN. The model is able to reproduce high energy pixels, but fails in reproducing the low energy pixels.The chapter also explores the KLD values and β_{KLD} for different datasets used for the thesis.

7

CONCLUSION

The monte carlo simulation of particle shower responses in an electromagnetic calorimeter incurs a significantly high amount of computational resource and time. Developing a fast simulation approach using deep learning methods to tackle this problem is the primary objective of this thesis. The studies were focused on the Belle II electromagnetic calorimeter responses and their particle energy range. To explore the fast simulation of electromagnetic showers, electron responses in a standalone crystal setup are simulated using Geant4, to be used as the datasets for training a WGAN model. These datasets were compared to the showers generated by the model against various physics observables to evaluate the performance of the model. It is observed that the model performs well in generating high-energy crystal responses, with the corresponding observables showing a good agreement between Geant4 and the model-generated showers. The generation of low energy-containing pixels appears challenging for the model, and the observables related to these pixels exhibit a disagreement.

To study the fast simulation of electromagnetic showers in a more realistic scenario, the WGAN model developed is trained on the electron responses from the Belle II ECL. The MC simulated events are from an early Belle II run. The energy deposited in the 5×5 crystals of the ECL are recorded and transformed into the training images. The correlation between Geant4 and WGAN-generated images also shows a good agreement for the observables related to high-energy pixels, whereas it disagrees with observables pertaining to low energy pixels. For the studies of the fast simulation of hadronic showers in the electromagnetic calorimeter crystals, negatively charged pion shower responses from the Geant4 standalone setup are simulated and used as the datasets to train the WGAN model. The main challenge here is the low energy deposited by the pions. To improve the performance, the model is conditioned by providing the hadronic interaction length in the crystals as a label. Since WGAN has difficulty reproducing the low energy pixels and most of the pixels of pion showers contain low energy deposition, the physics observables, except for a few, show a disagreement during the evaluation.

To improve the feature learning of the training dataset and increase the fidelity of the generated images, a VAE-WGAN model that combines the VAE and the WGAN is developed. Different training dynamics to optimize the model were tested, and the two promising ones are explained in the thesis. The generated showers show an improvement for some physics observables compared to the WGAN. This model and its training dynamics are used further for the studies of fast simulation of high granularity calorimeter responses in three different configurations: First, the model is investigated for low-energy electrons in high granularity crystals (30 × 30 pixels). Proper responses are not generated as there is insufficient energy deposition in the majority of the the pixels. Next, low-energy electrons in medium granularity crystals (10 ×10 pixels) are tested, and it is still found to be challenging to generate the outer pixels with low energy deposition and the precision is to be further improved. Lastly, the model is tested for high-energy electrons in high-granularity crystals (30 × 30 pixels), and the results are discussed.

The thesis also explores the ID-GAN model and it is trained on the high-energy high-granularity dataset. The thesis involves different types of training datasets based on energy responses and granularity and explores their latent space properties.

In conclusion, regarding the computational performance, the models exhibits a speed-up (about a range of 100-700 times for Belle II ECL), whereas the precision of the generated showers through the fast simulation method requires further improvement, especially for low-energy pixels. For experiments at the LHC and Pierre Auger, with high backgrounds which require an enormous amount of time for MC simulation, the compromise of precision against speed-up may be acceptable. However, for the Belle II experiment, where the required simulation time is much smaller in comparison, it is not worth compromising the precision of the generated showers at this time.

APPENDIX A

A.1. Architecture of Networks

This section provides the information about the architecture of the networks used for training datasets of 5 x 5 pixels, 10 x 10 pixels and 30 x 30 pixels.

Table A.1.: Critic network for 5 × 5 electron images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256 maps	1 × 1	SAME	×	leaky ReLU
Convolution	3 × 3	128 maps	1 × 1	SAME	×	leaky ReLU
Convolution	3 × 3	64 maps	1 × 1	VALID	×	leaky ReLU
Convolution	2 × 2	32 maps	1 × 1	VALID	layer norm	leaky ReLU
Reshape						
Concatenate Labels						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.2.: Generator network for 5 × 5 electron and pion images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Concatenate input + Labels						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	144 nodes	N/A	N/A	×	leaky ReLU
Reshape 3 × 3 × 16						
Transposed Convolution	5 × 5	64 maps	2 × 2	VALID	batch norm	leaky ReLU
Convolution	3 × 3	128 maps	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	1 maps	1 × 1	VALID	batch norm	ReLU

Table A.3.: Energy Regressor network for 5 × 5 electron images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256 maps	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	128 maps	1 × 1	VALID	batch norm	leaky ReLU
Reshape 512						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.4.: Critic network for 5 × 5 pion images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256 maps	1 × 1	VALID	×	leaky ReLU
Convolution	2 × 2	128 maps	1 × 1	VALID	layer norm	leaky ReLU
Reshape 512						
Concatenate Labels						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.5.: Energy Regressor network for 5 × 5 pion images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256 maps	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	128 maps	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	64 maps	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	32 maps	1 × 1	VALID	batch norm	leaky ReLU
Reshape 128						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.6.: Encoder network for 5 × 5 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	2 × 2	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	64	1 × 1	VALID	batch norm	leaky ReLU
Reshape 576						
Concatenate E, P_x, P_y						
Dense	N/A	20 nodes	N/A	N/A	×	×

Table A.7.: Decoder network for 5 × 5 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	144	N/A	N/A	×	leaky ReLU
Reshape 3 × 3 × 16						
Transposed Convolution	5 × 5	64	2 × 2	VALID	batch norm	leaky ReLU
Convolution	3 × 3	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	1	1 × 1	VALID	batch norm	ReLU

Table A.8.: Critic network for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	5 × 5	256	1 × 1	SAME	×	leaky ReLU
Convolution	3 × 3	128	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	32	1 × 1	SAME	layer norm	leaky ReLU
Reshape 3200						
Concatenate E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	×
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.9.: Encoder network VAE-WGAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	32	1 × 1	VALID	×	leaky ReLU
Reshape 512						
Concatenate E, P_x, P_y						
Dense	N/A	20 nodes	N/A	N/A	×	×

Table A.10.: Encoder network VAE-WGAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	128	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	SAME	batch norm	leaky ReLU
Convolution	3 × 3	32	1 × 1	VALID	×	leaky ReLU
Reshape 2048						
Concatenate E, P_x, P_y						
Dense	N/A	20 nodes	N/A	N/A	×	×

Table A.11.: Decoder network VAE-WGAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	144	N/A	N/A	×	leaky ReLU
Reshape 3 × 3 × 16						
Transposed Convolution	3 × 3	16	2 × 2	SAME	batch norm	leaky ReLU
Transposed Convolution	3 × 3	32	2 × 2	SAME	batch norm	leaky ReLU
Transposed Convolution	3 × 3	64	2 × 2	SAME	batch norm	leaky ReLU
Convolution	5 × 5	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	16	1 × 1	VALID	batch norm	leaky ReLU
Convolution	3 × 3	1	1 × 1	VALID	×	ReLU

Table A.12.: Energy Regressor VAE-WGAN network for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	32	1 × 1	VALID	×	leaky ReLU
Reshape 800						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.13.: Position Regressor network VAE-WGAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	2 × 2	32	1 × 1	VALID	×	leaky ReLU
Reshape 800						
Linear	N/A	2 nodes	N/A	N/A	×	×

Table A.14.: Critic network for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	5 × 5	256	1 × 1	SAME	×	leaky ReLU
Convolution	3 × 3	128	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	32	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	16	1 × 1	SAME	layer norm	leaky ReLU
Reshape 14400						
Concatenate E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.15.: Encoder network for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	10 × 10	256	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	layer norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	×	leaky ReLU
Reshape 800						
Concatenate E, P$_x$, P_y						
Dense	N/A	20 nodes	N/A	N/A	×	×

Table A.16.: Decoder network for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P$_x$, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	144	N/A	N/A	×	leaky ReLU
Reshape 3 × 3 × 16						
Transposed Convolution	3 × 3	16	2 × 2	SAME	batch norm	leaky ReLU
Transposed Convolution	3 × 3	32	2 × 2	SAME	batch norm	leaky ReLU
Transposed Convolution	3 × 3	64	2 × 2	SAME	batch norm	leaky ReLU
Transposed Convolution	3 × 3	64	2 × 2	SAME	batch norm	leaky ReLU
Convolution	5 × 5	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	batch norm	leaky ReLU
Convolution	7 × 7	16	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	1	1 × 1	VALID	×	ReLU

Table A.17.: Energy Regressor network for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	10 × 10	256	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	×	leaky ReLU
Reshape 800						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.18.: Position Regressor network for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	10 × 10	256	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	128	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	64	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	batch norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	×	leaky ReLU
Reshape 800						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.19.: The hyper parameters for the VAE-WGAN model for 5 × 5 images.

Hyperparameter	Value
γ	0.6
α	7.5
κ_{E}	0.0425
Batch Size	250

Table A.20.: The hyper parameters for the VAE-WGAN model for 10 × 10 images.

Hyperparameter	Value
γ	0.6
α	7.5
κ_{E}	0.0425
κ_{P}	0.0425
λ	5
Batch Size	250

Table A.21.: The hyper parameters for the VAE-WGAN model for 30 × 30 images.

Hyperparameter	Value
γ	0.6
α	7.5
κ_{E}	0.01
κ_{P}	0.01
λ	5
Batch Size	250

Table A.22.: Encoder network for 30 × 30 images ID-GAN.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	(2,2)	32	(2, 2)	(1, 1)	×	ReLU
Convolution	(4, 4)	64	(2, 2)	(1, 1)	×	ReLU
Convolution	(4, 4)	64	(2, 2)	(1, 1)	×	ReLU
Convolution	(4, 4)	256	(1, 1)	Default	×	ReLU
View() 256						
Concatenate E, P$_x$, P_y						
Linear	N/A	20 nodes	N/A	N/A	×	×

Table A.23.: Decoder network for 30 × 30 images ID-GAN.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
c + E, P$_x$, P_y						
Linear	N/A	256 nodes	N/A	N/A	×	ReLU
View (256, 1, 1)						
Transposed Convolution	(4, 4)	64	(1, 1)	Default	×	ReLU
Transposed Convolution	(4, 4)	64	(2, 2)	(1, 1)	×	ReLU
Transposed Convolution	(4, 4)	32	(2, 2)	(1, 1)	×	ReLU
Transposed Convolution	(2, 2)	1	(2, 2)	(1, 1)	×	ReLU

Table A.24.: Discriminator network for ID-GAN 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	(10, 10)	256	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(3, 3)	128	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(3, 3)	64	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(2, 2)	32	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(3, 3)	16	(1, 1)	SAME	layer norm	leaky ReLU
Reshape 14400						
Concatenate E, P$_x$, P_y						
Linear	N/A	100 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.25.: Energy Regressor and Position regressor network for ID-GAN 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	(5, 5)	256	(1, 1)	VALID	batch norm	leaky ReLU
Convolution	(5, 5)	128	(1, 1)	VALID	batch norm	leaky ReLU
Convolution	(5, 5)	64	(1, 1)	VALID	batch norm	leaky ReLU
Convolution	(3, 3)	32	(1, 1)	VALID	batch norm	leaky ReLU
Convolution	(3, 3)	32	(1, 1)	SAME	batch norm	leaky ReLU
Convolution	(5, 5)	16	(1, 1)	SAME	batch norm	leaky ReLU
Reshape 4096						
Linear	N/A	1 nodes	N/A	N/A	×	×

Table A.26.: Generator network ID-GAN for 30 × 30 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	256	N/A	N/A	×	leaky ReLU
Reshape 16 × 4 × 4						
Transposed Convolution	(3, 3)	16	(2, 2)	Default	batch norm	leaky ReLU
Transposed Convolution	(3, 3)	32	(2, 2)	Default	batch norm	leaky ReLU
Transposed Convolution	(3, 3)	64	(2, 2)	Default	batch norm	leaky ReLU
Convolution	(5, 5)	32	(1, 1)	VALID	batch norm	leaky ReLU
Convolution	(5, 5)	16	(1, 1)	VALID	×	leaky ReLU
Convolution	(2, 2)	1	(1, 1)	VALID	×	ReLU

Table A.27.: Decoder network ID-GAN for 5 × 5 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P_x, P_y						
Linear	N/A	144 nodes	N/A	N/A	×	leaky ReLU
Reshape 16 × 3 × 3						
Transposed Convolution	(5, 5)	64	(2, 2)	Default	layer norm	leaky ReLU
Convolution	(3, 3)	128	(1, 1)	VALID	layer norm	leaky ReLU
Convolution	(3, 3)	1	(1, 1)	VALID	×	ReLU

Table A.28.: Encoder network ID-GAN for 5 × 5 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	(3, 3)	256	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(3, 3)	128	(1, 1)	SAME	layer norm	leaky ReLU
Convolution	(3, 3)	64	(1, 1)	VALID	layer norm	leaky ReLU
Convolution	(2, 2)	32	(1, 1)	VALID	layer norm	leaky ReLU
Reshape 128, Concatenate labels						
Linear	N/A	20 nodes	N/A	N/A	×	×

Table A.29.: Encoder network for ID-GAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
Convolution	3 × 3	256	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	128	1 × 1	SAME	layer norm	leaky ReLU
Convolution	3 × 3	64	1 × 1	SAME	layer norm	leaky ReLU
Convolution	2 × 2	32	1 × 1	SAME	layer norm	leaky ReLU
Reshape 3200						
Concatenate E, P_x, P_y						
Linear	N/A	500 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	20 nodes	N/A	N/A	×	×

Table A.30.: Decoder network for ID-GAN for 10 × 10 images.

Operation	Kernal	Features	Stride	Padding	Normalization	Activation
z + E, P_x, P_y						
Linear	N/A	10 nodes	N/A	N/A	×	leaky ReLU
Linear	N/A	256	N/A	N/A	×	leaky ReLU
Reshape 16 × 4 × 4						
Transposed Convolution	3 × 3	16	2 × 2	Default	layer norm	leaky ReLU
Transposed Convolution	3 × 3	32	2 × 2	Default	layer norm	leaky ReLU
Convolution	5 × 5	64	1 × 1	VALID	layer norm	leaky ReLU
Convolution	5 × 5	32	1 × 1	VALID	×	leaky ReLU
Convolution	2 × 2	1	1 × 1	VALID	×	ReLU

APPENDIX B

B.1. Particle Gun Condition

Table B.1.: The particle gun conditions for simulating electron showers in Belle II ECL.

Particle Gun Condition	
basf2	`release-04-01-04`
Beam background level	**BG×1**
Experiment Number	1003
e^- samples	1 particle per event
$\theta_{\text{generated}}$	$11.4^\circ < \theta_{\text{generated}} < 156.1^\circ$
$\phi_{\text{generated}}$	$0^\circ < \phi_{\text{generated}} < 360^\circ$
x Vertex	$\mathcal{N}(0,\ 5\ \text{cm})$
y Vertex	$\mathcal{N}(0,\ 5\ \text{cm})$
z Vertex	0

B.2. WGAN vs. Geant4 standalone setup

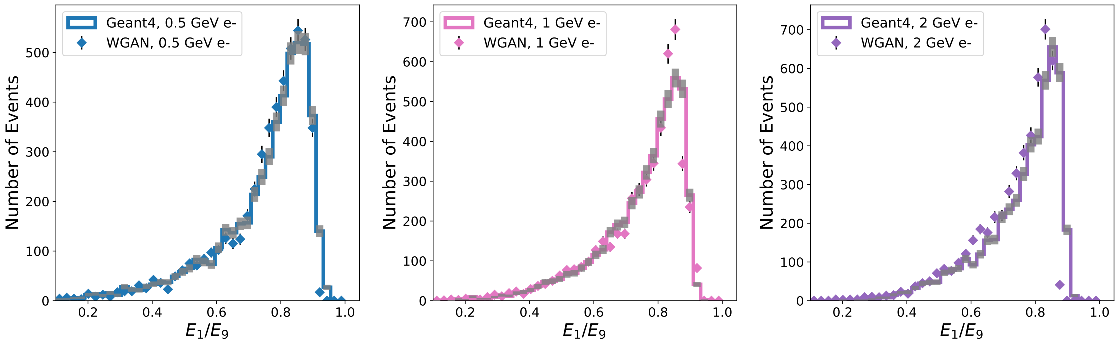


Figure B.1.: Distribution of E_1/E_9 for 0.5 GeV, 1 GeV and 2 GeV energies which shows the ratio of energy deposited in the innermost cell to the inner nine cells in the 5 × 5 crystal array.

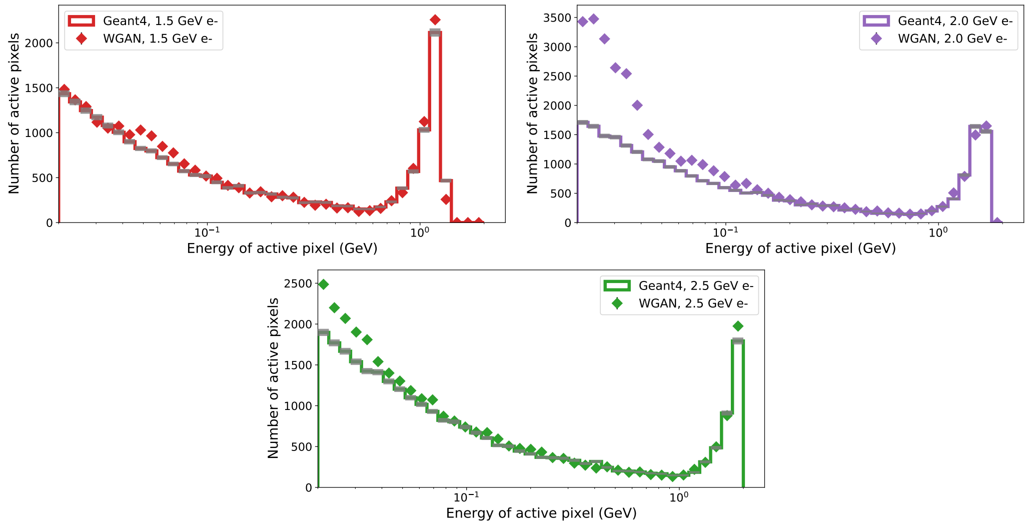


Figure B.2.: The distribution of energies of all the active pixels in the 5 × 5 crystals for 1.5 GeV, 2 GeV and 2.5 GeV electrons. Distribution is shown for energies greater than 20 MeV.

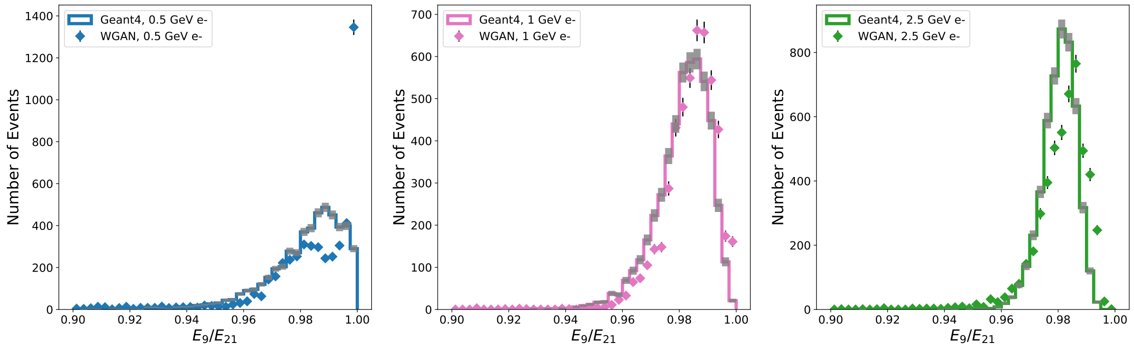


Figure B.3.: The distribution of E_9/E_{21} for 0.5 GeV, 1GeV and 2.5 GeV energies.

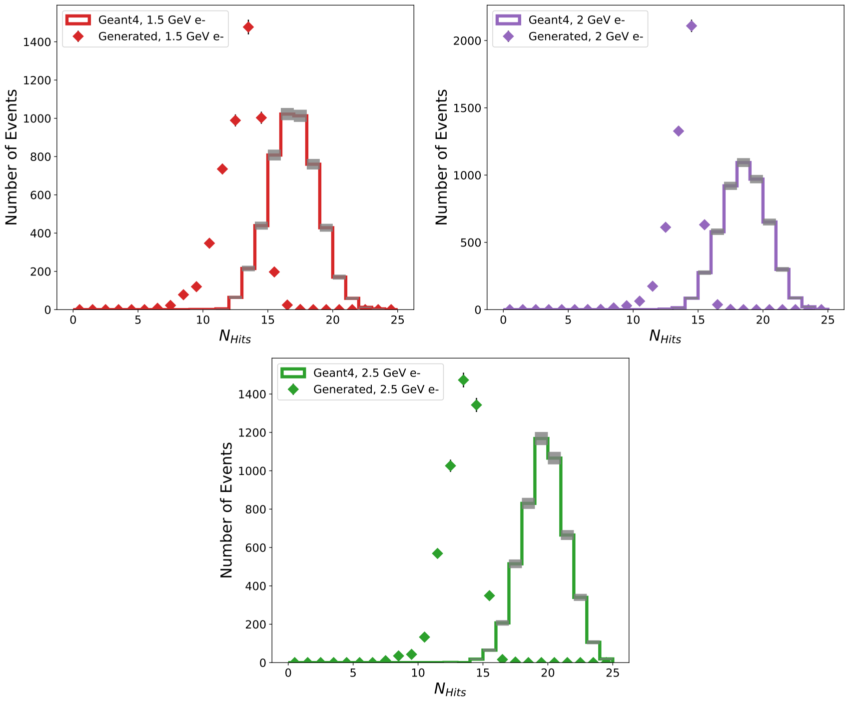


Figure B.4.: The distribution of $\mathrm{N_{Hits}}$ (for energies greater than or equal to 1 MeV deposited) for 1.5 GeV, 2 GeV and 2.5 GeV electrons.

B.3. WGAN vs. Belle II Simulation

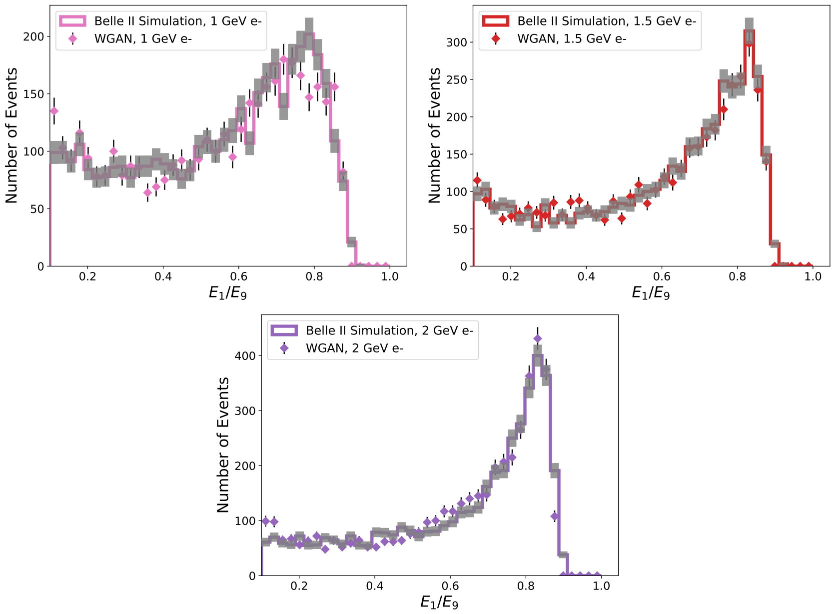


Figure B.5.: Distribution of E_1/E_9 for 1 GeV, 1.5 GeV and 2 GeV energies.

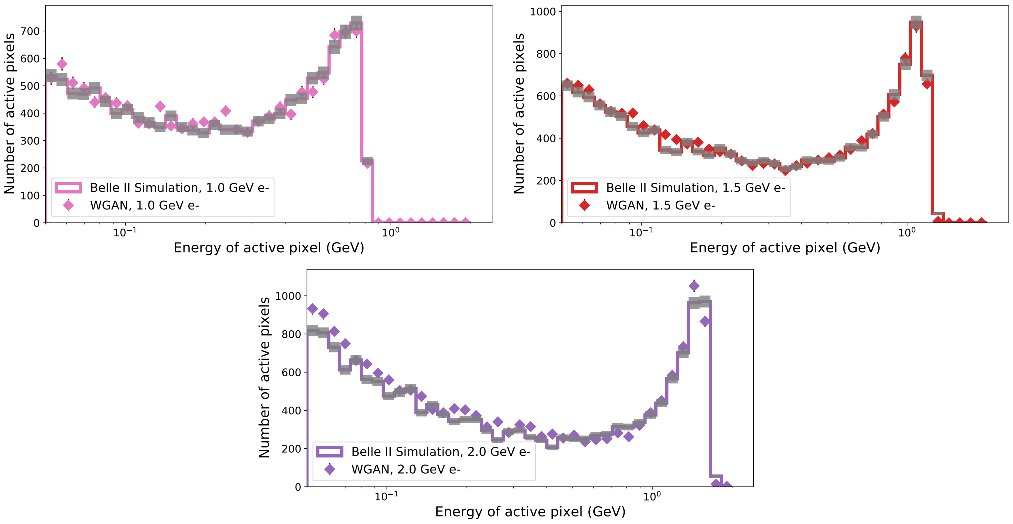


Figure B.6.: The distribution of energies of every active pixel in the 5 × 5 Belle II ECL crystals for 1 GeV, 1.5 GeV and 2 GeV electrons. The distribution is shown for energies greater than 50 MeV.

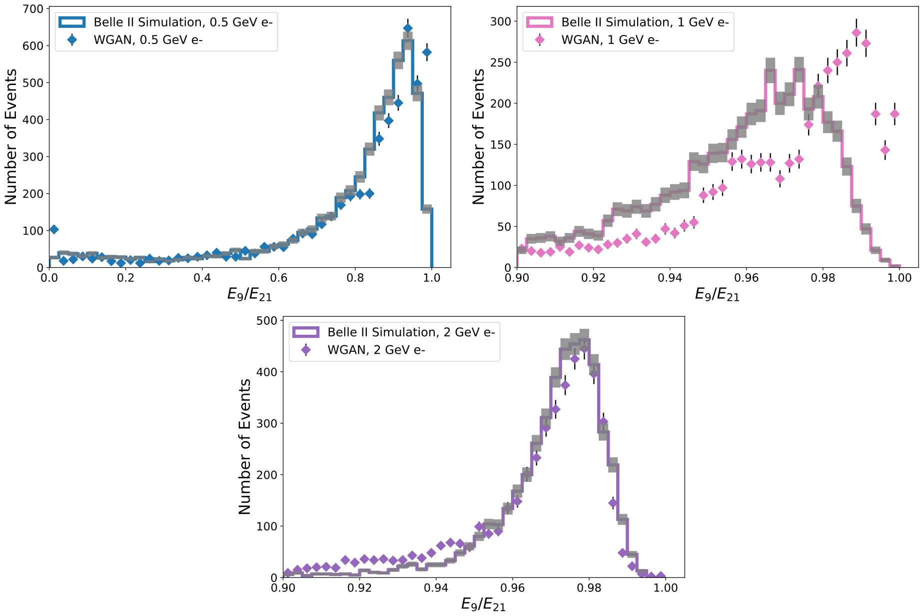


Figure B.7.: The distribution of E_9/E_{21} in 5 × 5 Belle II ECL crystals for 0.5 GeV, 1 GeV and 2 GeV electrons.

B.4. VAE-WGAN vs. Geant4 Simulation

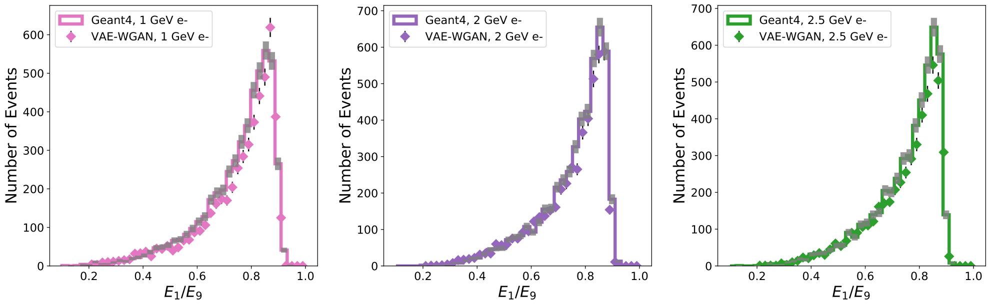


Figure B.8.: Distribution of E_1/E_9 for 1 GeV, 2 GeV and 2.5 GeV energies.

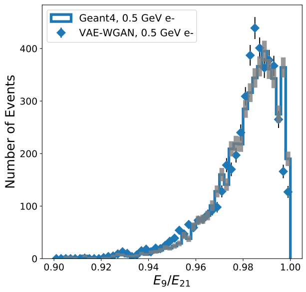


Figure B.9.: The distribution of E_9/E_{21} in 5 × 5 crystals for 0.5 GeV electrons.

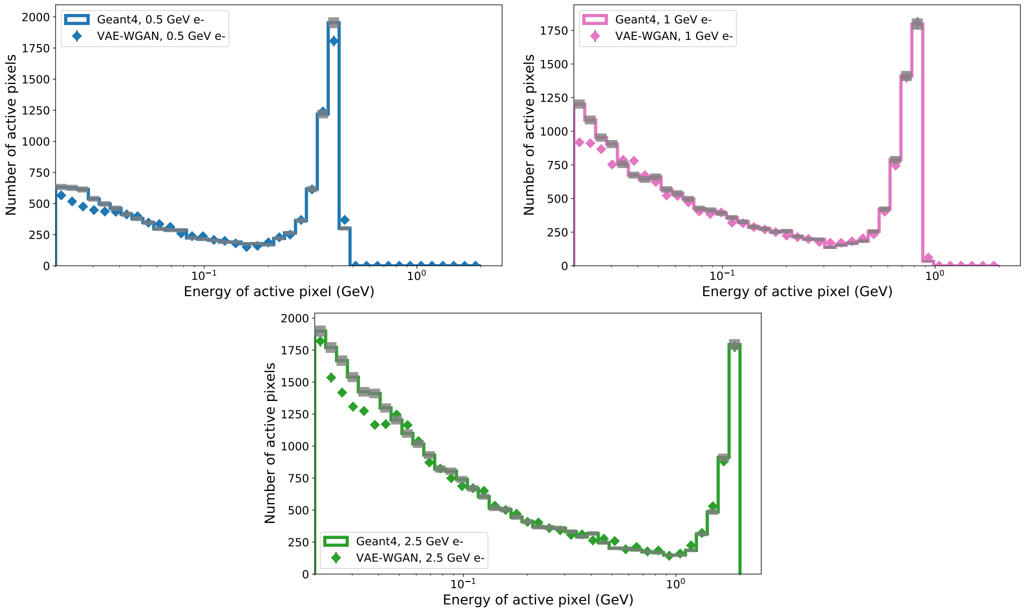


Figure B.10.: The distribution of energies of every active pixel in the 5 × 5 crystals for 0.5 GeV, 1 GeV and 2.5 GeV electrons. The distribution is shown for energies greater than 20 MeV.

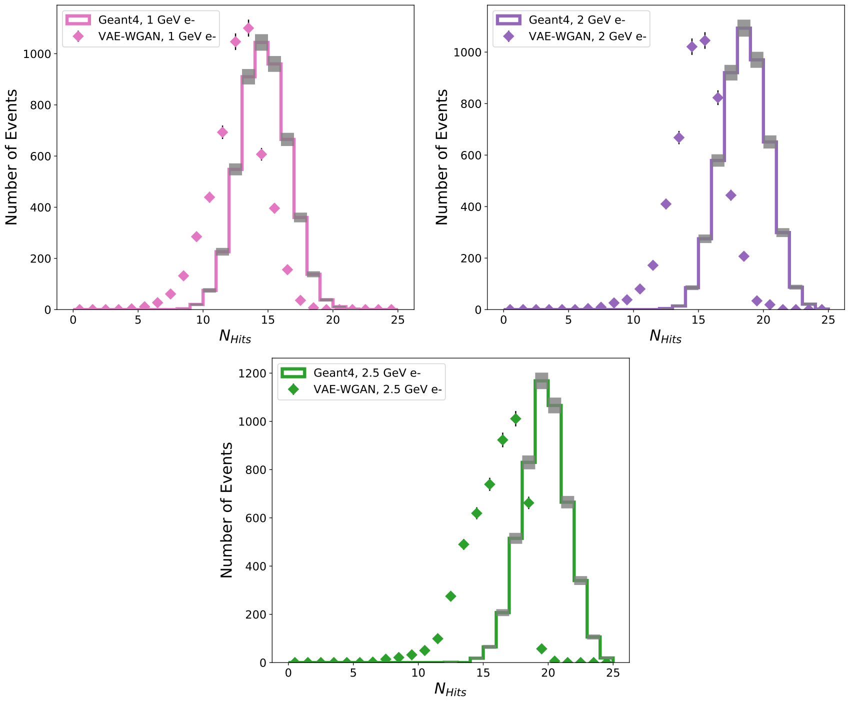


Figure B.11.: The distribution of N_{Hits} (for energies greater than or equal to 1 MeV deposited) for 1 GeV, 2 GeV and 2.5 GeV electrons.

C

BIBLIOGRAPHY

[1] Private communication Thomas Kuhr, image from Martin Ritter.

[2] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos, “Exploring End-to-end Deep Learning Applications for Event Classification at CMS,” *EPJ Web of Conferences* **214** (01, 2019) 06031, 01, 2019.

[3] D. Tilotta, “Regression Deep Neural Networks for top-quark-pair resonance finding in the dilepton channel: feasibility study and foreseen improvements over traditional analysis techniques with the ATLAS experiment at the LHC,” 2020. `https://cds.cern.ch/record/2763283`. Presented 14 Dec 2020.

[4] **Pierre Auger**, J. Glombitza, “Air-Shower Reconstruction at the Pierre Auger Observatory based on Deep Learning,” *PoS* **ICRC2019** (2020) 270, 2020.

[5] M. Erdmann, J. Glombitza, and T. Quast, “Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network,” *Comput. Softw. Big Sci.* **3** no. 1, (2019) 4, `arXiv:1807.01954 [physics.ins-det]`, 2019.

[6] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, *et al.*, “Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed,” *Comput. Softw. Big Sci.* **5** no. 1, (2021) 13, `arXiv:2005.05334 [physics.ins-det]`, 2021.

[7] “Super KEKB and Belle II.”. `https://www.belle2.org/project/super_kekb_and_belle_ii`. Accessed on 2022-05-19.

[8] J. Zhu, D. Zhao, B. Zhou, and B. Zhang, “{LIA}: Latently Invertible Autoencoder with Adversarial Learning,” 2020. `https://openreview.net/forum?id=ryefE1SYDr`.

[9] W. Lee, D. Kim, S. Hong, and H. Lee, "High-Fidelity Synthesis with Disentangled Representation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds., pp. 157–174. Springer International Publishing, Cham, 2020.

[10] C. W. Fabjan and F. Gianotti, "Calorimetry for particle physics," *Rev. Mod. Phys.* **75** (2003) 1243–1286, 2003.

[11] H. Kolanoski and N. Wermes, *Particle Detectors*. Oxford University Press, 6, 2020.

[12] P. D. Group, P. A. Zyla, R. M. Barnett, J. Beringer, *et al.*, "Review of Particle Physics," *Progress of Theoretical and Experimental Physics* **2020** no. 8, (08, 2020) , `https://academic.oup.com/ptep/article-pdf/2020/8/083C01/34673722/ptaa104.pdf`, 08, 2020. `https://doi.org/10.1093/ptep/ptaa104`. 083C01.

[13] M. J. Berger and S. M. Seltzer, "Tables of energy losses and ranges of electrons and positrons.". `https://ntrs.nasa.gov/citations/19650002905`, Accessed on 2022-05-19.

[14] "Photomultiplier tube.". `https://en.wikipedia.org/wiki/Photomultiplier_tube`. Accessed on 2022-05-19.

[15] S. Longo and J. M. Roney, *Measurements of the Radiation Hardness of CsI(Tl) Scintillation Crystals and Comparison Studies with Pure CsI for the Belle II Electromagnetic Calorimeter*. PhD thesis, Victoria, University of Victoria, Victoria, 2015. Presented on 10 09 2015.

[16] Private communication Pablo Goldenzweig.

[17] "Electromagnetic calorimeter.". `https://belle2.jp/electromagnetic-calorimeter`. Accessed on 2022-05-19.

[18] **BELLE II calorimeter group**, B. Shwarz, "Electromagnetic Calorimeter of the Belle II detector," *PoS* **PhotoDet2015** (2016) 051, 2016.

[19] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, *et al.*, "Geant4—a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506** no. 3, (2003) 250–303, 2003. `https://www.sciencedirect.com/science/article/pii/S0168900203013688`.

[20] J. Allison, K. Amako, J. Apostolakis, H. Araujo, *et al.*, "Geant4 developments and applications," *IEEE Transactions on Nuclear Science* **53** no. 1, (2006) 270–278, 2006.

[21] J. Allison, K. Amako, J. Apostolakis, P. Arce, *et al.*, "Recent developments in Geant4," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **835** (2016) 186–225, 2016. `https://www.sciencedirect.com/science/article/pii/S0168900216306957`.

[22] S. Guatelli, D. Cutajar, B. Oborn, and A. Rosenfeld, "Introduction to the Geant4 Simulation toolkit," *AIP Conference Proceedings* **1345** (05, 2011) 303–322, 05, 2011.

[23] Geant4, "Geant4: A Simulation Toolkit.". `https://github.com/Geant4/geant4/tree/master/examples/basic/B4/B4c`. Accessed: 2022-05-19.

[24] J. Allison, K. Amako, J. Apostolakis, P. Arce, *et al.*, "Recent developments in Geant4," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators,*

Spectrometers, Detectors and Associated Equipment **835** (2016) 186–225, 2016. `https://www.sciencedirect.com/science/article/pii/S0168900216306957`.

[25] J. Gemmler, *Development and Deployment of a Deep Neural Network based Flavor Tagger for Belle II*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2020.

[26] “This Person Does Not Exist.”. `https://thispersondoesnotexist.com`. Accessed on 2022-05-19.

[27] D. H. Nguyen and K. Tsuda, “Generating reaction trees with cascaded variational autoencoders,” *The Journal of Chemical Physics* **156** (01, 2022) , 01, 2022.

[28] A.-I. Albu, A. Enescu, and L. Malagò, “Tumor detection in brain mris by computing dissimilarities in the latent space of a variational autoencoder,” 2020.

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, *et al.*, “Generative Adversarial Networks,” 2014. `https://arxiv.org/abs/1406.2661`.

[30] L. Weng, “From GAN to WGAN,” *CoRR* **abs/1904.08994** (2019) , 1904.08994, 2019. `http://arxiv.org/abs/1904.08994`.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” 2017. `https://arxiv.org/abs/1704.00028`.

[32] J. Hui, “GAN — Wasserstein GAN & WGAN-GP.”. `https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490`, Accessed on 2022-05-19.

[33] T. Quast, “Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network.”. `https://github.com/ThorbenQuast/HGCAL_TB2017_WGAN`. Accessed on 2022-05-19.

[34] **CMS**, “The Phase-2 Upgrade of the CMS Endcap Calorimeter,” tech. rep., CERN, Geneva, 2017. `https://cds.cern.ch/record/2293646`.

[35] A. Odena, C. Olah, and J. Shlens, “Conditional Image Synthesis With Auxiliary Classifier GANs,” 2016. `https://arxiv.org/abs/1610.09585`.

[36] T. Awes, F. Obenshain, F. Plasil, S. Saini, *et al.*, “A simple method of shower localization and identification in laterally segmented calorimeters,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **311** no. 1, (1992) 130–138, 1992. `https://www.sciencedirect.com/science/article/pii/0168900292908582`.

[37] “basf2 software documentation- simulation.”. `https://software.belle2.org/sphinx/recommended-training/simulation/doc/index.html`. Accessed on 2022-05-19.

[38] R. W. Novotny, “Scintillation Detectors in Nuclear and High-Energy Physics.”. `https://indico.desy.de/event/9679/contributions/90963/attachments/61201/74226/04_Novotny_Scintillation_Detectors.pdf`. Accessed: 2022-05-19.

[39] T. Quast, “Geant4 Based Event Display & Standalone Simulation.”. `https://github.com/ThorbenQuast/HGCal_TB_Geant4`. Accessed on 2022-05-19.

[40] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” 2013. `https://arxiv.org/abs/1312.6114`.

[41] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," 2014. `https://arxiv.org/abs/1401.4082`.

[42] C. Yan, S. Wang, J. Yang, T. Xu, and J. Huang, "Re-balancing variational autoencoder loss for molecule sequence generation," 2019. `https://arxiv.org/abs/1910.00698`.

[43] J. Rocca and B. Rocca, "Understanding Variational Autoencoders (VAEs).". `https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73`. Accessed on 2022-05-19.

[44] I. Higgins, L. Matthey, A. Pal, C. Burgess, *et al.*, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*. 2017. `https://openreview.net/forum?id=Sy2fzU9gl`.

[45] M. D. Hoffman and M. J. Johnson, "Elbo surgery: yet another way to carve up the variational evidence lower bound.". `http://approximateinference.org/accepted/HoffmanJohnson2016.pdf`. Accessed: 2022-05-19.

[46] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," 2019. `https://arxiv.org/abs/1901.05534`.

[47] J. He, "Aggressive training of inference network.". `https://github.com/jxhe/vae-lagging-encoder`. Accessed: 2022-05-19.

[48] X. Chen, Y. Duan, R. Houthooft, J. Schulman, *et al.*, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016. `https://arxiv.org/abs/1606.03657`.

[49] H. Fu, C. Li, X. Liu, J. Gao, *et al.*, "Cyclical annealing schedule: A simple approach to mitigating kl vanishing," 2019. `https://arxiv.org/abs/1903.10145`.

[50] E. Buhmann, S. Diefenbacher, E. Eren, F. Gaede, *et al.*, "Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network," *EPJ Web Conf.* **251** (2021) 03003, `arXiv:2102.12491 [physics.ins-det]`, 2021.

[51] **CALICE Collaboration**, N. Tsuji, L. Liu, T. Torimaru, T. Mori, and W. Ootani, "Study on Granularity Optimization for ILD Hadron Calorimeter," *JPS Conf. Proc.* **27** (2019) 012015. 4 p, 2019. `https://cds.cern.ch/record/2727526`.

[52] W. Lee, "ID-GAN.". `https://github.com/1Konny/idgan`. Accessed: 2022-05-19.

ACKNOWLEDGEMENT

Firstly, I want to thank supervisors Prof. Dr. Günter Quast and Prof. Dr. Florian Bernlochner for all the guidance and support for my work. I would also like to express my gratitude to my Mentor, Prof. Dr. Matthias Steinhauser for his support and advice during this period. I want to convey my most sincere gratitude to Dr. Pablo Goldenzweig for his advice, valueable comments, patience, help, motivation, and personal support during my research period. Without his constant motivation, suggestions, and support during hard times, this work would not have been possible. I also want to sincerely thank Prof. Dr. Torben Ferber for all his valuable suggestions, corrections, and ideas on my work. This has helped me a lot.

I would like to sincerely acknowledge: Dr. Jochen Gemmler, Dr. William Sutcliffe, Dr. Thorben Quast, Erik Buhmann, Marcel Köpke, and Dr. Matthias Schnepf for their valuable guidance and support. I learned a lot from Thorben Quast, and it was such a fabulous experience to work with him. Thank you for introducing me to the field of deep learning. Working with Marcel during the initial period of my Ph.D. has helped me develop a good knowledge of machine learning.

Also, my genuine thanks to Dr. Felix Metzner, Dr. Raynette van Tonder, Moritz Bauer, Patrick Ecker, and Abtin Narimani for helping proofread my thesis. I also thank all my colleagues and Frau Bärbel Bräunling from the bottom of my heart for all their help and support for making my experience at KIT a memorable one.

Last but not least, I would like to thank my family, my best friend Lisha Fathima, my husband and all my friends, for their love and emotional support during this entire journey. Sincere love and gratitude to my father who stood by my side when the times got hard during the Covid.

This thesis was written with the help of LaTeX using an adapted version of a class originally by Roland Bless, Timo Rohrberg and Thorsten Haberecht. The color code follows the corporate design of the KIT (`https://www.sek.kit.edu/ci_cd.php`).