Multi-Particle Reconstruction with Dynamic Graph Neural Networks

Shah Rukh Qasim

PhD 2023

Multi-Particle Reconstruction with Dynamic Graph Neural Networks

Shah Rukh Qasim

A thesis submitted in partial fulfilment of the requirements of
Manchester Metropolitan University
for the degree of Doctor of Philosophy

Department of OTEHM and
Department of Computing and Mathematics
Manchester Metropolitan University

IN COLLABORATION WITH

Conseil Européen pour la Recherche Nucléaire (CERN)

2023

Abstract

The task of finding the incident particles from the sensor deposits they leave on particle detectors is called event or particle reconstruction. The sensor deposits can be represented generically as a point cloud, with each point corresponding to three spatial dimensions of the sensor location, the energy deposit, and occasionally, also the time of the deposit. As particle detectors become increasingly more complex, ever-more sophisticated methods are needed to perform particle reconstruction. An example is the ongoing High Luminosity (HL) upgrade of the Large Hadron Collider (HL-LHC). The HL-HLC is the most significant milestone in experimental particle physics and aims to deliver an order of magnitude more data rate compared to the current LHC. As part of the upgrade, the endcap calorimeters of the Compact Muon Solenoid (CMS) experiment – one of the two largest and general-purpose detectors at the LHC – will be replaced by the radiation-hard High Granularity Calorimeter (HGCAL).

The HGCAL will contain $\sim$ 6 million sensors to achieve the spatial resolution required for reconstructing individual particles in HL-LHC conditions. It has an irregular geometry due to its hexagonal sensors, with sizes varying across the longitudinal and transverse axes. Further, it generates sparse data as less than 10% of the sensors register positive energy. Reconstruction in this environment, where highly irregular patterns of hits are left by the particles, is an unprecedentedly intractable and compute-intensive pattern recognition problem. This motivates the use of parallelisation-friendly deep learning approaches. More traditional deep learning methods, however, are not feasible for the HGCAL because a regular grid-like structure is assumed in those approaches.

In this thesis, a reconstruction algorithm based on a dynamic graph neural network called GravNet is presented. The network is paired with a segmentation technique, Object Condensation, to first perform point-cloud segmentation on the detector hits. The property-prediction capability of the Object Condensation approach is then used for energy regression of the reconstructed particles. A range of experiments are conducted to show that this method works well in conditions expected in the HGCAL i.e., with 200 simultaneous proton-proton collisions. Parallel algorithms based on Nvidia CUDA are also presented to address the computational challenges of the graph neural network discussed in this thesis. With the optimisations, reconstruction can be performed by this method in approximately 2 seconds which is suitable considering the computational constraints at the LHC.

The presented method is the first-ever example of deep learning based end-to-end calorimetric reconstruction in high occupancy environments. This sets the stage for the next era of particle reconstruction, which is expected to be end-to-end. While this thesis is focused on the HGCAL, the method discussed is general and can be extended not only to other calorimeters but also to other tasks such as track reconstruction.

Acknowledgements

First, I would like to thank Dr Jan Kieseler, my main PhD advisor. He always offered me timely help and guidance and explained everything I was struggling with. I always enjoyed our discussions and felt free to say, ask, and challenge anything. I was immensely impressed by the fact that he went through my thesis on a short notice, while having to work on the weekend, to provide timely feedback. Second, I would like to thank Dr Maurizio Pierini for his support and for giving me a chance to work on this project. The discussions during the student meetings were of immense help. I have and probably will always look up to him. Third, I would like to thank Dr Raheel Nawaz for trusting me and offering me long-term guidance for both during the PhD and the career after. And on the same note, Dr Ernest Edifor, for filling in Dr Raheel's shoes, when he left Manchester Metropolitan University, and helping me navigate through the university's requirements. And finally, I would also like to thank my colleague, Ema Puljak, for providing feedback on this thesis.

I would also like to thank Dr Ian Fisk and the Flatiron Iron Institute of the Simons Foundation for providing me access to their GPU clusters. These GPU clusters were paramount to the success of this thesis.

The research work required for this thesis was carried out under funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 772369).

Contents

Chapter 1

Thesis Introduction

Humans have been pondering about how the universe works and what it is made of for thousands of years. One way to do so is by studying the most fundamental constituents of matter. This endeavour is called particle physics. While philosophers and theists have been proposing a variety of theories for eons, most of the progress was made over the course of the last few centuries. Many theories were developed, disproven, and reiterated, and finally, in the 1970s, scientists developed the Standard Model (SM) that very accurately describes the most fundamental particles the universe is made of and the laws governing their interactions. Even though the SM has demonstrated remarkable accuracy and predictive power, there are still some anomalies left to be explained, e.g. lack of integration of general theory and matter-antimatter asymmetry. Gigantic particle physics experiments have been setup around the world to test the SM rigorously and find further evidence that explains our current theories.

While the initial experiments were conducted completely manually, computing has slowly evolved into a crucial component of particle physics. Currently, automated algorithms are used everywhere, from detector physics and event selection to reconstruction. Even more recently, machine learning-based adaptive algorithms have replaced some of the more traditional methods to a large extent and their usage is only expected to increase in the coming years.

Among many computational challenges in particle physics, one of the hardest is event reconstruction. So far, classical methods have been employed for this task. Although the ever increasing complexity of this calls for adoption of machine learning based approaches. Even in cases where classical algorithms work well, e.g. track reconstruction, machine learning based approaches will profit from the parallelisation offered by modern GPUs. These optimisations are crucial to meet the computational demands. In some cases, especially for calorimetric reconstruction, often the pattern recognition task in itself is very complex and poses challenges to traditional approaches. The most significant example of this is the future HGCAL at the CMS experiment. Reconstruction in the HGCAL will be studied in this thesis. It poses unique computational challenges and is one of the most complex particle reconstruction problems.

1.1 Thesis structure

Figure 1.1 shows the structure of this thesis. It begins with an introduction of required physics concepts in Chapter 2. It gives a brief account of the history of particle physics and the Standard Model. A higher level overview of the LHC and the CMS experiment is also given in this chapter highlighting the significance of the research that will be presented in the later chapters.

As discussed in Chapter 2, there are many sub-detectors at the CMS. Owing to their significance in their thesis, calorimeters and their physics is discussed in detail in Chapter 3. The HGCAL is also presented in this chapter.

Chapter 4 presents a literature review of the classical reconstruction algorithms used at the LHC. It discusses reconstruction in trackers introduced in Chapter 2 and calorimeters introduced in Chapter 3. Classical approach to reconstruction in the HGCAL is also discussed in this chapter.

Chapter 5 presents an introduction to machine learning and reviews various approaches. Their applications in high energy physics are also presented for a comprehensive literature review.

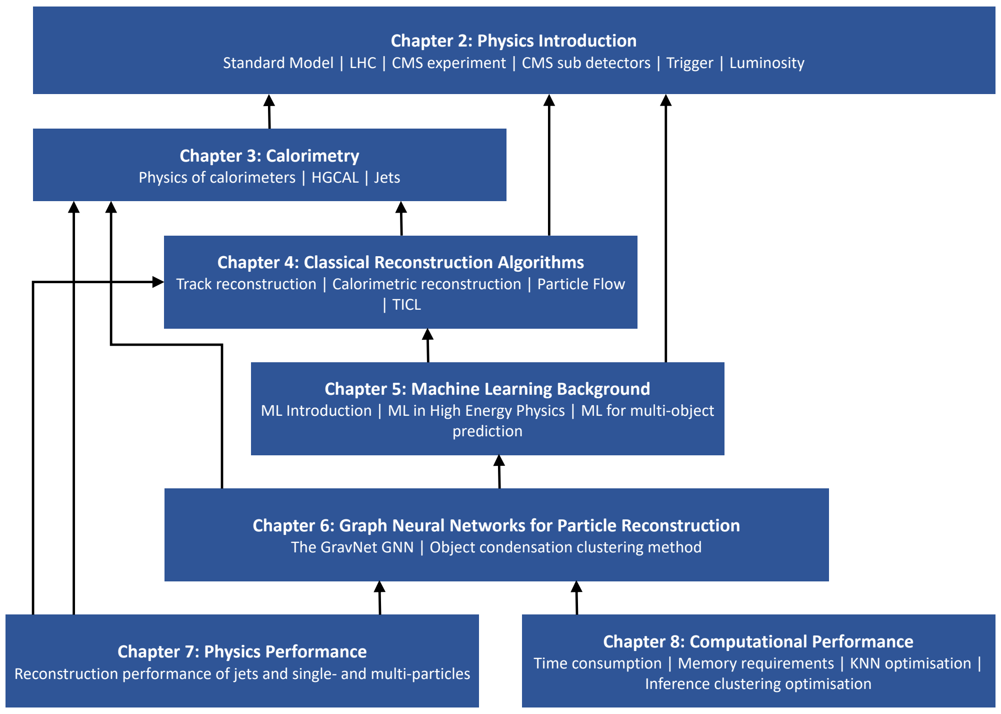

Chapter 2: Physics Introduction
Standard Model | LHC | CMS experiment | CMS sub detectors | Trigger | Luminosity
Chapter 3: Calorimetry
Physics of calorimeters | HGCAL | Jets
Chapter 4: Classical Reconstruction Algorithms
Track reconstruction | Calorimetric reconstruction | Particle Flow | TICL
Chapter 5: Machine Learning Background
ML Introduction | ML in High Energy Physics | ML for multi-object prediction
Chapter 6: Graph Neural Networks for Particle Reconstruction
The GravNet GNN | Object condensation clustering method
Chapter 7: Physics Performance
Reconstruction performance of jets and single- and multi-particles
Chapter 8: Computational Performance
Time consumption | Memory requirements | KNN optimisation | Inference clustering optimisation

Figure 1.1: A diagram showing the flow of all except this and the last chapter.

The graph neural network and the clustering approach presented in this thesis are then discussed in Chapter 6. A detailed physics performance of the presented method is then conducted in Chapter 7 via a range of experiments. A comparison is also performed to the classical approach presented in Chapter 4. Similarly, Chapter 8 discusses the computational performance and requirements of the presented method in different complexities.

Finally, the summary and conclusion of this thesis are presented in Chapter 9.

1.1.1 Note on publication

Some parts of this thesis have been published in a journal article (Qasim, Chernyavskaya, et al., 2022). Most notably, this includes some parts of

Chapter 6 and Section 7.1 in Chapter 7.

1.2 Research problem

There can be as many as 0.2M sensors that register a positive energy and over a thousand particles in each endcap of the HGCAL, making the problem very complex. A sophisticated reconstruction algorithm is needed that can perform particle reconstruction in these challenging conditions. Because of the intractable nature of the HGCAL data, it is also important to use machine learning based approaches. However, the HGCAL has a non-uniform geometry, which makes it challenging to use traditional deep learning approaches like convolutional neural networks. In addition, most of the sensors do not register any signal, which makes the data sparse and adds to the complexity of the problem. It is important to note that reconstruction in the HGCAL signifies one of the two most complex reconstruction tasks in particle physics currently, where the other one is track reconstruction. Prior to the research presented in this thesis, only one reconstruction algorithm for the HGCAL exists. It is a classical approach and is presented in Chapter 4, in Section 4.5.

1.2.1 Research contributions presented in this thesis

With the contributions listed below, the need for a reconstruction algorithm that works in harsh conditions of the HGCAL is addressed in the research presented in this thesis:

1. A novel reconstruction algorithm based on graph neural networks and point cloud segmentation is presented in Chapter 6, the first algorithm to do single-shot calorimetric reconstruction in the high-luminosity conditions.

2. The method is shown to work in 200 pileup[1] on a toy calorimeter

[1]Pileup refers to the number of primary proton-proton collisions. This is further explained in Chapter 2, in Section 2.3.

based on the HGCAL, as well as in a multi-particle environment in the HGCAL directly in Chapter 7.

3. A framework of truth-prediction matching to study the reconstruction performance is presented in Chapter 7, in Section 7.2.2.

4. A detailed reconstruction performance analysis of the new method is conducted in Chapter 7.

5. The reconstruction performance is also compared to the only other reconstruction algorithm that exists for the HGCAL in Chapter 7, in Section 7.2.3.

6. Optimisations are presented that improve the computational performance of the method to make it compatible with the offline computing requirements of the CMS experiment in Chapter 8, in Section 8.2.2 and Section 8.2.3.

7. Computational performance is studied in detail in different pileup conditions in Chapter 8 in Section 8.1.

The following works were published as a direct result of this thesis:

1. Qasim, S. R., Chernyavskaya, N., Kieseler, J., Long, K., Viazlo, O., Pierini, M., and Nawaz, R. End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks. European Physical Journal C 82, 753 (2022).[2]

2. Bhattacharya S, et al. GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter. 20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (2021).[3]

[2]Qasim, Chernyavskaya, et al. (2022)
[3]Bhattacharya et al. (2022)

3. Qasim, S. R., Long, K., Kieseler, J., Pierini, M., and Nawaz, R. Multi-particle reconstruction in the High Granularity Calorimeter using object condensation and graph neural networks. 25th International Conference on Computing in High Energy and Nuclear Physics (2021).[4]

[4]Qasim, Long, et al. (2021)

Chapter 2

Physics Introduction

This chapter will present an introduction to physics. First, an overview of the recent history of particle physics and the adoption of computing algorithms is given in Section 2.1, followed by a brief description of the Standard Model in Section 2.2. The Large Hadron Collider and the CMS experiments are described in Section 2.3 and Section 2.4, respectively. The High Luminosity upgrade of the LHC is then discussed in Section 2.5. Finally, Section 2.6 summarises the chapter.

2.1 A historical perspective

One of the earliest particle detection apparatuses was developed by J. J. Thomson in the early 20th century. It was based on a cathode ray tube and led to the discovery of the electron (e^-). The gold foil experiment, conducted by Rutherford, is another example of a very early particle detection apparatus. The nucleus was discovered this way, marking one of the first significant milestones in the development of particle physics (Rutherford, 1911).

Around 1920, cloud chambers were first used as particle detectors and they dominated particle physics experiments for the next 30 years. Saturated water vapours are employed in cloud chambers; and whenever a charged particle passes through, condensation occurs, leaving behind a vis-

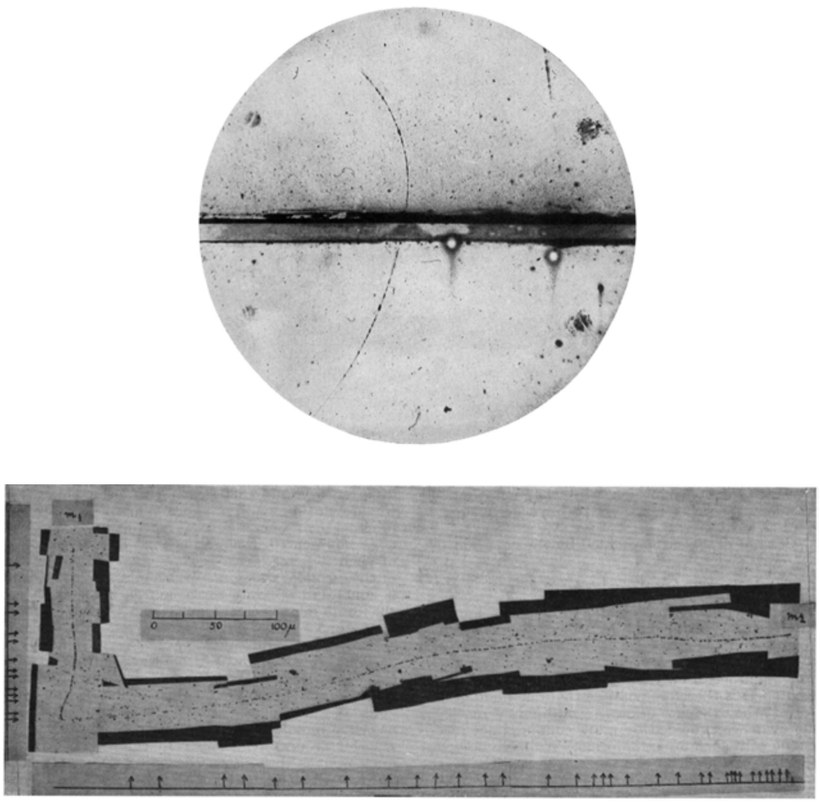


Figure 2.1: Top: Discovery of the positron in 1933 in a cloud chamber (Anderson, 1933). Bottom: Discovery of the pion in 1947 with photographic emulsion (Lattes et al., 1947).

ible ion trail. The discovery of positrons (e^+) was the first major physics discovery that employed cloud chambers, and it was done by Carl Anderson. Cloud chambers later evolved into bubble chambers. They also work on a similar principle, but instead of condensation, bubbles are formed by the charged particles. Photographic emulsions were another type of early detectors that were placed at high altitudes where charged particles make ionisation tracks directly on photographic plates. Many more particles, muons, pions, kaons, etc were also discovered by these photographic techniques. Fig. 2.1 depicts the tracks left by particles in these photographs that led to the discoveries of positrons and pions. In all these detectors, photographs of the ionisation tracks were taken, developed, and the particle tracks were manually identified. After doing so, the momenta of the particles was estimated by studying the curvature of the tracks. Cameras were placed at different angles and the track identification was done only in 2D. Theoretically, two cameras are required to find the tracks and their parameters; however, a third one was usually added for robustness and cross-checking. In the prior days, the process was completely manual, i.e. tracks were spotted by eye on photographic films, and then rulers were used for finding the associated parameters. Later, film measuring machines were invented that allowed an operator to slide a marker along on the xy plane and mark various vertices using electronic controls. Pattern recognition programs were then added to these machines that made them, at least partially, automated.

However, in 1968, Georges Charpak developed the multi-wire proportional chamber (Charpak et al., 1968). It was the first wire chamber consisting of an array of wires. This allows for a very fast electronic readout of particle tracks, increasing the detection frequency by three orders of magnitude. UA1 detector built at CERN used this technology, allowing 3D reconstruction of the events.

At the same time, computers were also scaling up in power, enabling automation of many tasks, including track reconstruction. Algorithms based on Kalman filters (Kalman, 1960) started taking the stage in the 80s.

Apart from detection, the second major part of particle physics experiments is the production of particles, also called the beam. On this front, until 1930s, physicists were using natural phenomena, such as radioactivity or cosmic rays, to study particles, but these methods didn't offer a fine control over the production of the particles. This started to become a limitation in the discovery of new physics. To overcome this, Ernest Lawrence invented the cyclotron. In cyclotrons, charged particles go in a curved trajectory and are accelerated with the help of an electric field, increasing the kinetic energy in each cycle. Pions were first synthetically produced in a cyclotron. An improved version of cyclotron, called the synchrotron, was proposed in 1944 by Vladimir Veksler; and one of the earliest synchrotron based physics experiment was constructed at Lawrence Berkeley National

Laboratory called Bevatron. Bevatron was able to perform collisions at a center-of-mass energy of around 6 GeV. This led to the discovery of antiprotons and antineutrons.

The modern particle accelerators and colliders are descendants of the cyclotron. They are paired with much improved particle detectors, computing devices, and reconstruction algorithms. This enables us to reach much higher energy and frequency. Some of the famous experiments include Tevatron, SLAC, and the Large Hadron Collider.

2.2 The Standard Model

Except for gravity, three fundamental forces of nature (electromagnetic, weak, and strong interactions) are unified under a theory called the Standard Model (Oerter, 2006). The Standard Model comprises quantum theories and special relativity, and is a very accurate description of the nature of the universe and the laws governing them. Under standard model, the universe is comprised of six quarks and six leptons that together make up the fermions. All the known matter in the universe is comprised of the fermions. While the leptons exist independently, the quarks can only be found as part of composite particles. The proton, for example, is a composite particle consisting of two up quarks and one down quark. The interactions between the fermions are governed by five force-carrying particles. Photons (γ) mediate the electromagnetic interactions and gluons (g), the strong interaction. The other three bosons (W, H, Z) govern the weak interactions. Examples of particle interactions include particle decay, annihilation, and pair production.

Except for g and γ, all other particles in the Standard Model have a mass. Another property of a particle is its electric charge. The electrons (e^-) and the positron (e^+) are examples of charged particles, while the neutrino (ν) and γ are examples of neutral particles. The elementary particles of the Standard Model are illustrated in Fig. 2.2, along with their mass, charge, and spin. Spin is another property of a particle that describes an intrinsic

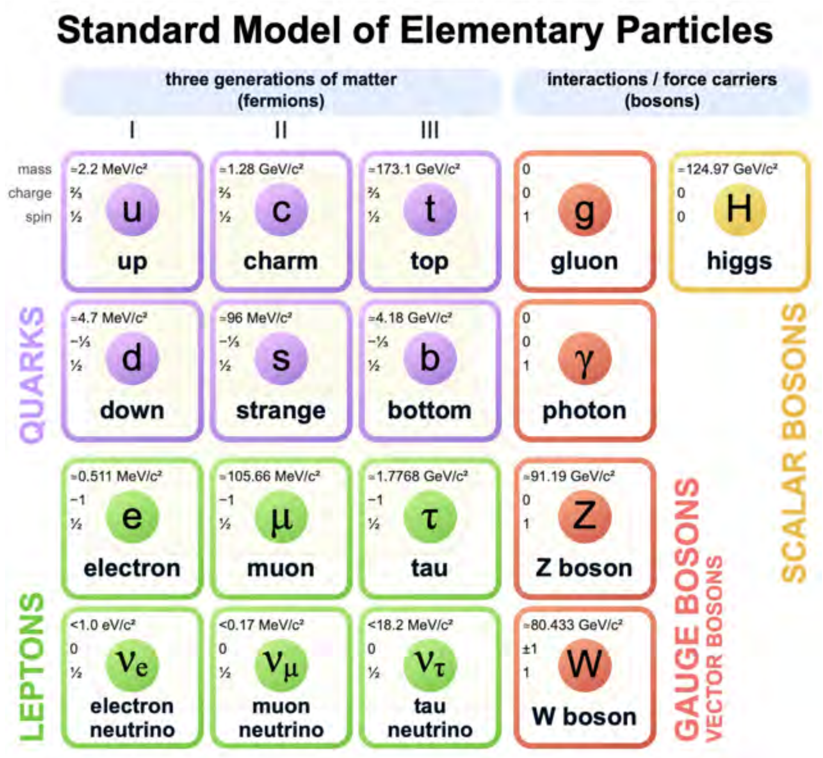


Figure 2.2: The Standard Model particles and their properties (Wikimedia Commons, 2019).

angular momentum.

Hadrons are the particles that are made up of quarks. Two hadrons, the neutron (n) and the proton (p), make up most of the mass of the universe. Other examples of long-lived hadrons are pions (π^+, π^+).

Under special relativity, the speed of light (c) and the laws of physics remain the same anywhere in the universe and at any speed. This leads to the concept of frame of reference, i.e. measurements of quantities, such as speed, energy, and length of an object, depend on the observer. Special relativity also defines mass (m) and energy as equivalent quantities under the following relationship:

$$E = mc^2 = m \; . \tag{2.1}$$

Here, as common in high energy physics, the constant c is set to 1 under "natural units". This relationship holds for a frame of reference under which an object is at rest. Therefore, m and E are called the rest mass and the rest energy, respectively. E and m are generally measured in prefix multipliers of electronvolts. An electronvolt is the kinetic energy gained by an electron as it travels through a potential difference of 1 V and it is used as a unit of energy and mass.

2.3 The Large Hadron Collider (LHC)

Most of the Standard Model particles do not exist independently and need to be created in high-energy collisions for physics studies, as permitted by Equation 2.1. The total energy of a system depends on the frame of reference. For a particle travelling with momentum p relative to a certain frame of reference, the total energy of the system in that frame of reference can be evaluated as:

$$E = \sqrt{(mc^2)^2 + p^2c^2} = \sqrt{m^2 + p^2} \ . \tag{2.2}$$

Here, natural units, i.e. $c = \hbar = 1$, have been used. A quantity called center-of-mass energy or invariant mass ($\sqrt{s}$) can be used to describe the energy of a system that remains the same in all frames of reference. In a two-particle system, $\sqrt{s}$ can be calculated as:

$$\sqrt{s} = \sqrt{m_1^2 + m_2^2 + 2(E_1E_2 - \boldsymbol{p}_1 \cdot \boldsymbol{p}_2)} \ . \tag{2.3}$$

Here, m_1 describes the rest energy of the first particle and m_2, the second particle. In a low-energy interaction, the velocity of the two particles with respect to each other is very low ($p_2 \approx 0$), resulting in $E_2 = m_1$. Equation 2.3 reduces to:

$$\sqrt{s} = \sqrt{m_1^2 + m_2^2 + 2m_1m_2} = m_1 + m_2 \ . \tag{2.4}$$

However, if one of the particles has a high kinetic energy ($E_2 >> m_1 = m_2$), $\sqrt{s}$ takes the following form, describing the total energy of the system in fixed-target particle accelerators:

$$\sqrt{s} \approx \sqrt{2m_1E_2}\ . \tag{2.5}$$

In particle accelerators where the colliding particles have both been accelerated in opposite directions with approximately the same momentum, $\boldsymbol{p_1}.\boldsymbol{p_2} = p_1p_1\text{cos}(180°) \approx -E_1^2$, and therefore $\sqrt{s}$ can be defined as follows:

$$\sqrt{s} \approx 2E_1\ . \tag{2.6}$$

A Standard Model particle called the Higgs boson (H^0) was theorised in 1964 (Englert and Brout, 1964; Guralnik et al., 1964; Higgs, 1964) and by the start of the 21st century, it was the last missing piece of the Standard Model yet to be confirmed. Many experiments, including the Large Electron-Positron (LEP) collider (Djouadi, 1995), sought its discovery; however, they could not reach the required $\sqrt{s}$ to produce the boson. It was the one of the main goals of construction of the LHC that began in 1993 in the same tunnels where the LEP was hosted. The construction took over 15 years to finish and the initial tests were run in 2008 (Myers, 2012).

The LHC is a proton-proton collider, where two beams of protons are accelerated in opposite directions. Fig. 2.3 shows a picture of the LHC and its two beams. The total energy of the collision can hence be defined by Equation .2.6. As of 2022, the highest energy we have reached at the LHC is 6.8 TeV per beam, resulting in total center-of-mass energy of 13.6 TeV and it is expected to further increase to 14 TeV. This energy is used in creating new particles that are observed in the detectors. Unlike an electron-positron annihilation, these high-energy collision events at the LHC are significantly more complex, and create hundreds of particles in every collision.

In 2012, the Higgs boson was discovered at CERN when the LHC reached $\sqrt{s} = 8$ TeV (ATLAS Collaboration, 2012; CMS Collaboration, 2012). With $\sqrt{s} > 13$ TeV, at which the LHC is currently operating, it is the world's

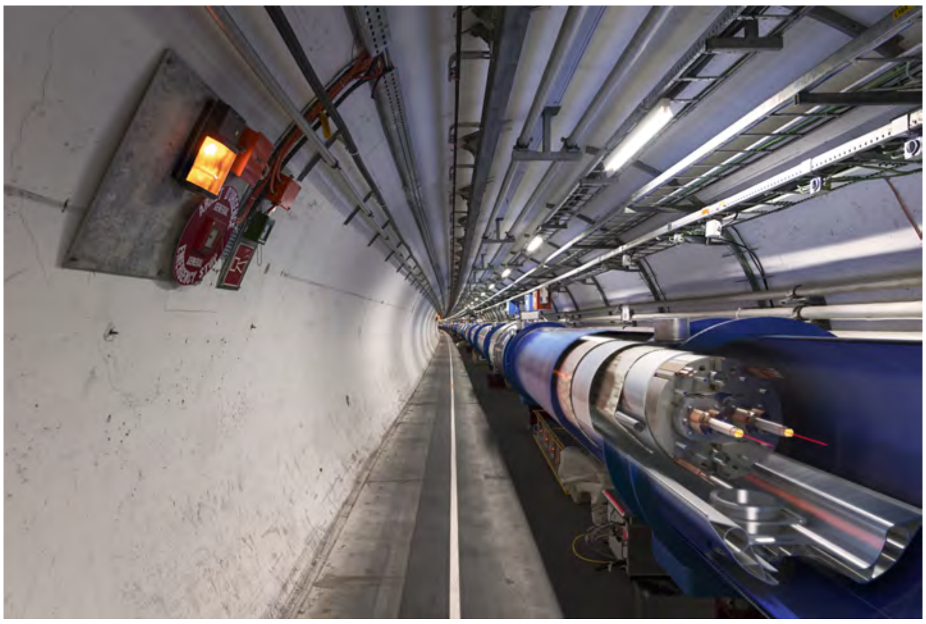


Figure 2.3: A picture of the LHC (Mobs, 2019) showing a cut of the dipole and the two beam pipes in which two sets of protons are accelerated in opposite directions (Dominguez, 2014).

largest particle accelerator, and is widely thought of as the most complex machine ever built. For a small fraction of the time, lead ions (Pb^+) are also accelerated instead of protons to study quark-gluon plasma (Müller et al., 2012).

Figure 2.4 schematically shows the entire journey of the protons as they pass through various accelerators before finally getting injected into the LHC. The acceleration of protons begins with hydrogen gas at an accelerator called LINAC4 (Linear Accelerator 4). When these protons enter the LHC, their energy is 450 GeV where it is further pushed to > 6.5 TeV. The two beams of protons are collided with $\sqrt{s} > 13$ TeV at the centre of different detectors, called experiments. The center of the detector where the collisions occur is called the interaction point. Some experiments at the LHC accelerator complex are CMS (Compact Muon Solenoid)[1], ATLAS (A

[1]CMS Collaboration (2008)

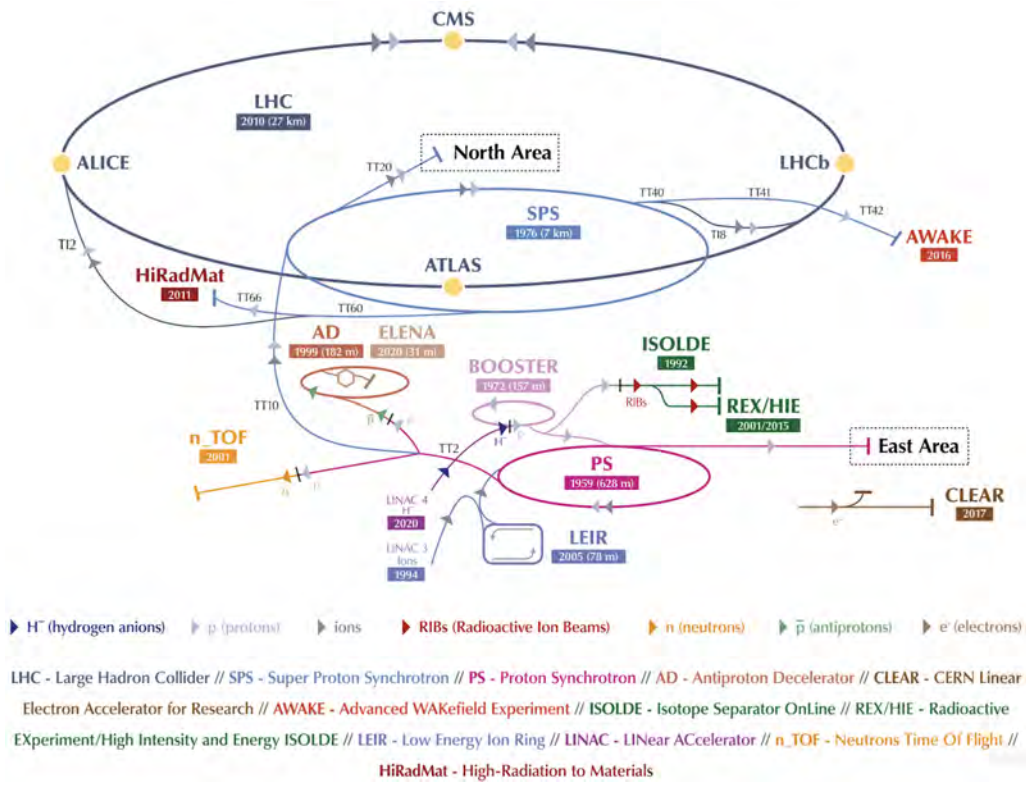

CMS
LHC
2010 (27 km)
ALICE
North Area
LHCb
TT20
TT40
TT41
TT42
SPS
1976 (7 km)
TI8
AWAKE
2016
TI2
ATLAS
HiRadMat
2011
TT66
TT60
AD
1999 (182 m)
ELENA
2020 (31 m)
ISOLDE
1992
BOOSTER
1972 (157 m)
RIBs
REX/HIE
2001/2015
TT10
n_TOF
2001
TT2
East Area
PS
1959 (628 m)
LINAC 4
H⁻
2020
CLEAR
2017
LINAC 3
Ions
1994
LEIR
2005 (78 m)
H⁻ (hydrogen anions)
p (protons)
ions
RIBs (Radioactive Ion Beams)
n (neutrons)
p̄ (antiprotons)
e⁻ (electrons)
LHC - Large Hadron Collider // SPS - Super Proton Synchrotron // PS - Proton Synchrotron // AD - Antiproton Decelerator // CLEAR - CERN Linear Electron Accelerator for Research // AWAKE - Advanced WAKefield Experiment // ISOLDE - Isotope Separator OnLine // REX/HIE - Radioactive EXperiment/High Intensity and Energy ISOLDE // LEIR - Low Energy Ion Ring // LINAC - LINear ACcelerator // n_TOF - Neutrons Time Of Flight // HiRadMat - High-Radiation to Materials

Figure 2.4: The accelerator complex of the LHC. First, hydrogen ions ($\mathrm{H} + e^- \rightarrow \mathrm{H}^-$) are accelerated at the Linear Accelerator 4 (Arnaudon et al., 2006). They are stripped of their two electrons at the booster to form protons ($\mathrm{H}^- \rightarrow \mathrm{H}^+ + 2e^-$). These protons ($\mathrm{H}^-$) then make their way through two accelerators, the Proton Synchrotron and the Super Proton Synchrotron, before entering the LHC.

Toroidal LHC Apparatus) [2], LHCb (LHC beauty) [3], and ALICE (A Large Ion Collider Experiment)[4]. LHCb and ALICE are specialised experiments focusing on the beauty quark and heavy ion collisions, respectively. CMS and ATLAS are general-purpose experiments, and the largest ones in size. The goal of a detector is to perform event reconstruction, i.e. find all the particles that were formed in a collision event. For every particle, the following quantities are reconstructed:

[2]ATLAS Collaboration (2008)
[3]LHCb Collaboration (2008)
[4]Alice Collaboration (2006)

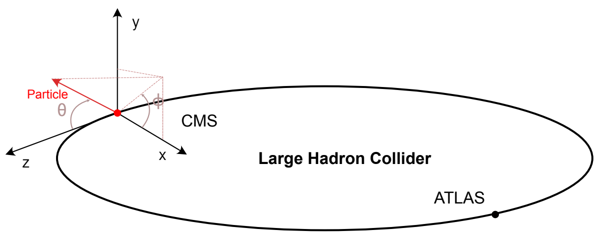


Figure 2.5: Schematic representation of the coordinate system (Qasim, Chernyavskaya, et al., 2022)

1. The type – identification or ID,
2. Direction,
3. Energy,

and if it is possible, also the time when the particle leaves its signature in the detector. This is called 5D reconstruction (3D position, energy, and time).

The protons at the LHC travel in groups, called bunches. Each bunch is $\sim$ 30 cm in length and the gap between two bunches is $\sim$ 7.5 m. In each bunch, there are approximately 10^{11} protons. As these particles travel at approximately the speed of light, we get a collision frequency of 40 MHz. A set of collisions in one bunch crossings can also be referred to as an event. Out of 10^{11} protons pairs, only $\sim$ 40 proton-proton (pp) collisions currently occur in an event. This is often referred to as pileup (N_{PU}) i.e. 40 pileup event means an event with 40 proton-proton interactions. The number of pp collisions per bunch is a random process following a Poisson distribution with $\lambda = N_{\mathrm{PU}}$.

2.3.1 Collider kinematics

A right-handed Cartesian coordinate system is used at the LHC as a basis and it is shown in Fig. 2.5 with the interaction point at the centre $(0, 0, 0)$. z-axis is placed along the beam pipe, while x and y axes are oriented towards the centre of the LHC and upwards, respectively. The Cartesian coordinate system is then extended to include three detector coordinates, θ, ϕ and η. θ is the angle from the beam axis. ϕ is the polar angle on the (x, y) plane. Pseudorapidity (η) (Wong, 1994) is another measure of the angle relative to the beam axis and can be calculated as a function of θ. These quantities can be evaluated as follows:

$$r = \sqrt{x^2 + y^2 + z^2} \ , \tag{2.7}$$

$$\theta = \arccos(z/r) \ , \tag{2.8}$$

$$\eta = \log(\tan\frac{\theta}{2}) \ , \tag{2.9}$$

$$\phi = \text{arctan2}(y, x) \ . \tag{2.10}$$

Fig. 2.6 shows energy scattering and number of particles as a function of θ and η in 14 TeV pp collisions. From the interaction point, $0 < \theta < \pi/2$ and $\eta < 0$ in one direction of the beam pipe and in the other direction, $\pi/2 < \theta < \pi$ and $\eta > 0$. It can be observed that most of the energy is scattered very parallel to the beam pipe, and is hard to notice on the θ spectrum. However, as η scales exponentially with θ close to the beam pipe (Equation 2.9), the energy scattered and the number of particles is within an order of magnitude across the η spectrum. Therefore, η is the preferred way to measure the angle from the beam pipe. Almost all the particles that ought to be observed in the detectors are coming directly from the collision point. Thus, it is generally enough to express a particle's direction in only two degrees of freedom, using η and ϕ. The distance between two particles is generally described as the Euclidean distance in the (η, ϕ) space.

Many of the primary particles are unstable and quickly decay into even more particles, increasing the complexity of the event. A detector is generally designed as symmetric because, in expectation, the same amount of energy is scattered on both sides of the beam pipe. Similarly, the energy scattering and the particle production are uniform across the ϕ spectrum as well.

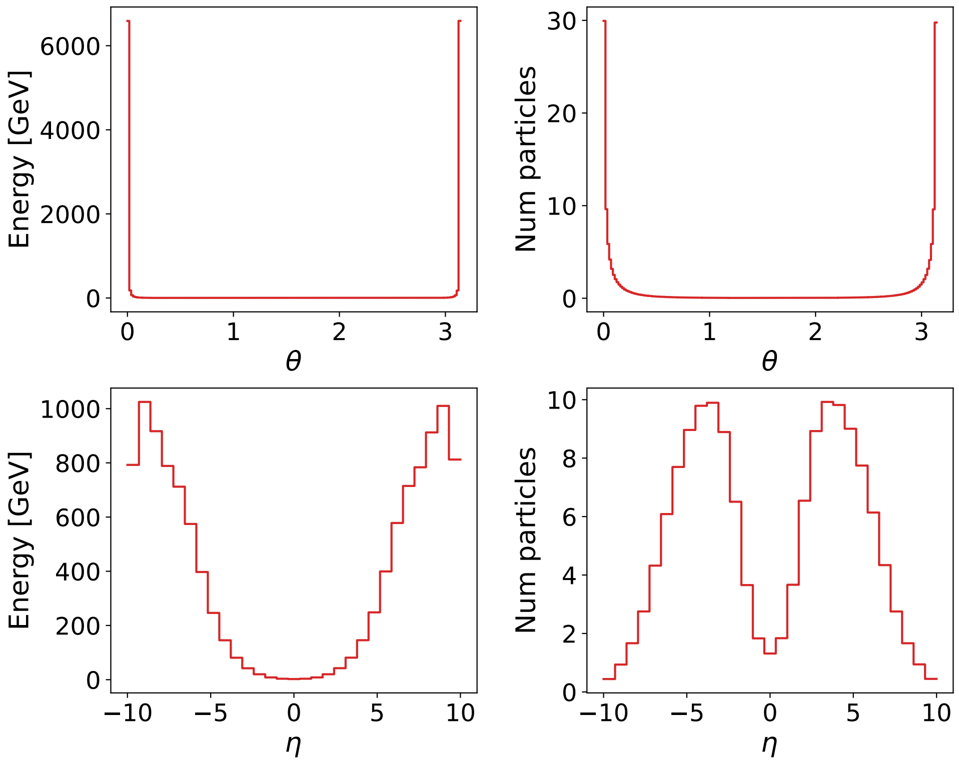


Figure 2.6: Expected number of primary particles and their energy in a 14 TeV pp collision distributed as a function of η and θ.

For the high-energy particles, $E = p$ according to Equation 2.2. However, only the momentum perpendicular to the beam pipe, called the transverse momentum (p_T), can be constrained in the hadronic collisions:

$$p_T = \frac{E}{\cosh(\eta)} \ , \tag{2.11}$$

$$\sum_{x \in P} p_T(x) = 0 \ . \tag{2.12}$$

Here, P represents the set of particles that are produced in an event.

2.4 CMS (Compact Muon Solenoid)

The CMS detector is one of the two general-purpose experiments setup at the LHC. The detector is located 100 m underground in Cessy, France. In 2012, alongside the ATLAS experiment, it confirmed the existence of the Higgs boson. It is a giant detector with a weight of $14 \cdot 10^6$ kg and spans 21 m, 15 m, and 15 m along the length, width, and height, respectively. As shown in Fig. 2.7, the detector is wrapped around the beam pipe where bunch crossings occur exactly in the centre.

Based on η, the detector can be divided into two distinct regions, the barrel and the endcap region. The barrel region is the region of the detector that wraps around the beam pipe with $|\eta| < 1.5$. The particles with momentum direction perpendicular to the beam pipe go into the barrel region. The endcap region captures the particles with direction parallel to the beam pipe with $|\eta| > 1.5$.

In Fig 2.8, the path of different particles is shown as they progress through different parts of the CMS detector. There is a 4 tesla superconducting magnet that comes after the calorimeters and before the muon chambers. The trajectory of the muons is inverted in the muon chambers as the direction of the magnetic field lines is inverted.

2.4.1 Tracker

Closest to the interaction point is the silicon tracker (Hartmann, 2007). It runs 0.28 m in length along the beam pipe. Only the charged particles leave traces on the sensors of the tracker. A single charged particle leaves many hits that are arranged in a curve, called a track. In a high-pileup event, thousands of tracks (representing charged particles) are present. An algorithm based on the Kalman Filter is used to separate out all the tracks from each other. This process is called track reconstruction. The tracker also makes use of the strong magnetic field that is present in the CMS detector. As charged particles interact with magnetic fields, their path is bent into a curve. The lower the momentum a particle has, the more it is

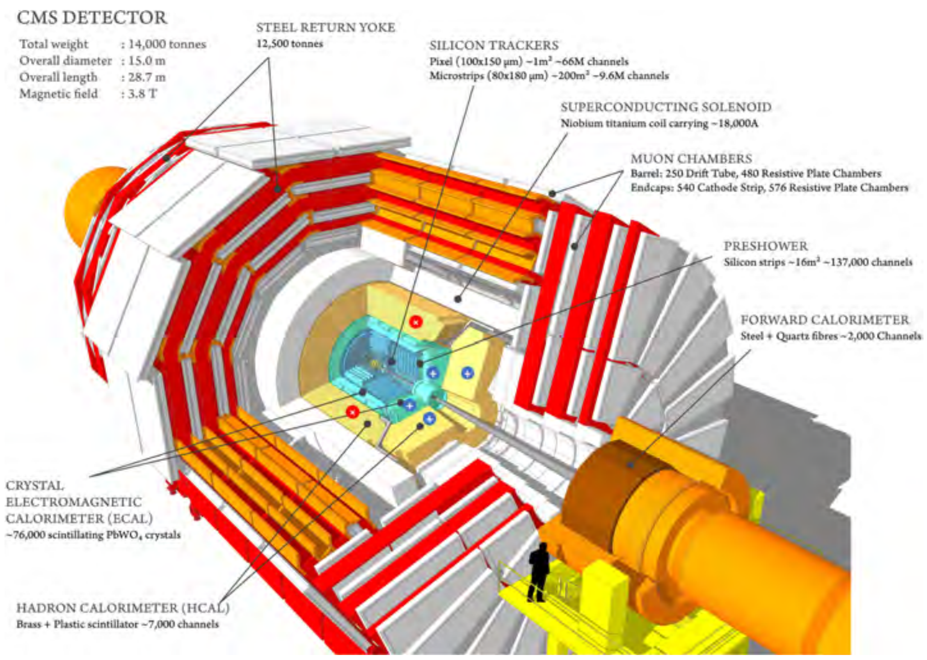

CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm) ~1m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 540 Cathode Strip, 576 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO₄ crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

Figure 2.7: An architecture diagram of the CMS detector, adapted from (Sakuma and McCauley, 2014). The endcap region is marked by + in blue colour while the barrel region is marked by $\times$ in red colour.

curved and vice versa. This property is used to estimate the momentum of the particles. The relative momentum resolution decreases with an increase in momentum almost linearly.

2.4.2 Calorimeters

Most of the particles, with muons as the most notable exception, shower in the calorimeter. A clustering algorithm is used to reconstruct all the showers in an event. The energy of a particle is then estimated based on the shower and the deposited energy on the sensors. Calorimetry is discussed in detail in Chapter 3, owing to its significance to this thesis. A brief account of the calorimeters at the CMS experiment is also presented.

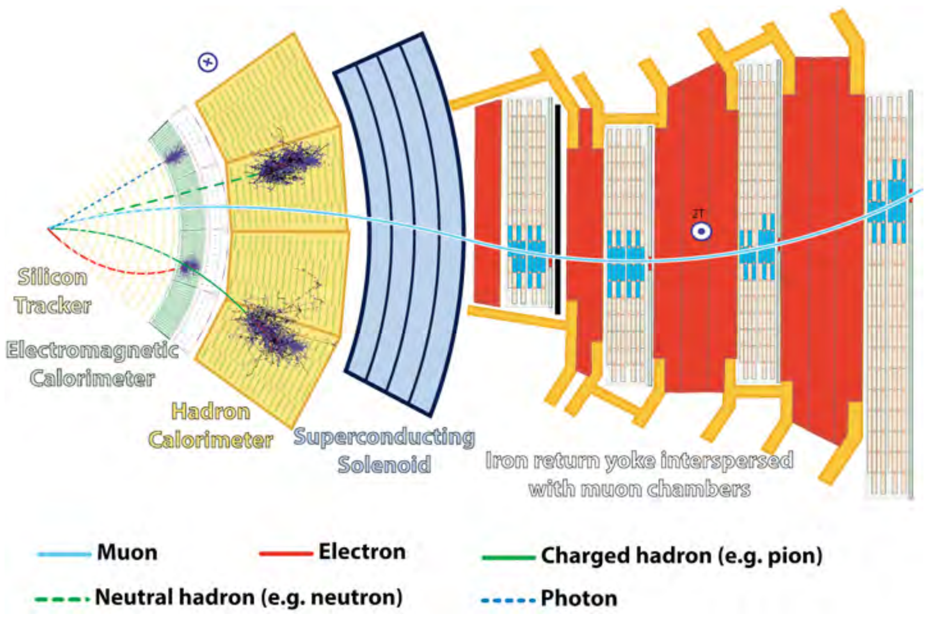

Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with muon chambers
2T
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

Figure 2.8: Longitudinal cross-sectional view of different particles in the CMS detector Barney, 2015. The interaction point is on the left side of the figure where the particles are originating. The trajectory of the charged particles is curved, while the neutral particle (neutral hadron) travels in a straight line. The muon interacts minimally with the matter and has the longest path, as it travels into the muon chambers, the detectors placed the farthest from the interaction point.

2.4.3 Muon chambers

Muons are charged particles and as so, they leave tracks in the tracker, but they interact minimally with the matter. On a typical day, about 10k/m^2, mostly originating from the sun, pass through the Earth's surface, and the majority of them continue straight through to the other side. The CMS detector has muon chambers that are specifically used for the detection of muons. The detector also gets its name, Compact Muon Solenoid, because of these chambers. Muons reconstruction is done in combination with track reconstruction. All the high p_{T} tracks from the tracker are considered as muon candidates that are picked from the muon chambers.

2.4.4 Trigger mechanism

There are 40M collision events occurring every second at the interaction point, and it is not possible to store all of them. A system called trigger is employed, which selects a fraction of these events for storage. The selection is based on maximising the likelihood of finding interesting events, such as those not explained by the current physics models. The CMS detector employs a two-tier trigger system: the first tier is called L1 trigger and the second tier, high-level trigger (HLT).

The L1 trigger brings down the collision rate to 100 kHz, which is further decreased to 400 Hz by the HLT. The L1 trigger is deployed online where the system has to make a decision within a few microseconds. Therefore, an embedded system comprising custom processor boards is used for this purpose. The algorithms employed at the L1 trigger are also very simplified. The HL trigger employs much more sophisticated algorithms that run on a computing farm comprising tens of thousands of standard CPU cores. While this is currently done using classical algorithms, GPUs have recently been added to the computational farm to profit from highly parallel machine learning algorithms for the future. Although a simplified form of reconstruction is already done at the HLT stage, the events that are output of the HLT are stored on disks for long-term usage and full event reconstruction. The magnitude of the stored data for offline computing is on the scale of tens of petabytes per year.

The trigger mechanism introduces a selection bias into the data recorded by the detectors. An event that does not have this bias is called a minimum bias event (Field, 2011).

2.5 The High Luminosity LHC (HL-LHC)

Luminosity in particle colliders refers to collision rate, and is defined as:

$$L = \frac{1}{\sigma}\frac{dN}{dt} \ . \tag{2.13}$$

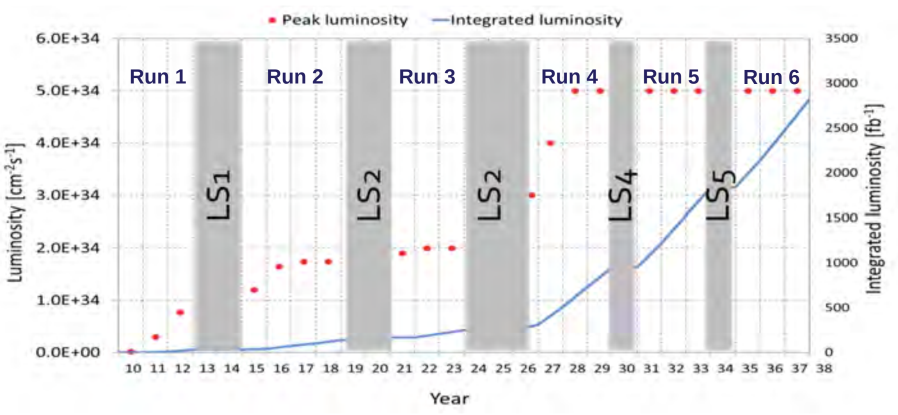


Figure 2.9: LHC projected time table where the long shutdowns (LS) and runs are highlighted. Instantaneous and integrated luminosity is also shown (Schmidt, 2016).

Here, σ is the cross section of the beam pipe at the interaction point, and N is the number of collisions. Luminosity is measured in m^{-2}s^{-1} in the System International (SI) units.

A term called integrated luminosity (L_{int}), measured in fb^{-1} (m^{-2} in SI units), and defined as

$$L_{\text{int}} = \int_t L \cdot dt \ , \tag{2.14}$$

directly measures the amount of data that has been collected over a certain period of time. Fig. 2.9 shows both the instantaneous and integrated luminosity of the LHC. Both the ATLAS and the CMS experiment will together generate around 300 fb^{-1} of data by the end of Run 3. The HL-LHC is the next milestone in particle physics. The aim of the HL-LHC is to accumulate 3000 fb^{-1} of luminosity by the 2030s by increasing the pileup to 200. The detectors at the LHC are receiving major upgrades to compensate for the high radiation and to increase the spatial resolution. At the CMS detector, the barrel calorimeters and the muon systems will receive major electronic upgrades while the tracker will be completely replaced. The end-cap calorimeters are also getting replaced by a new calorimeter called the HGCAL.

2.6 Summary

This chapter provides a high-level overview of the physics and the LHC to highlight the significance of the research that will be presented in later chapters. Owing to a wide body of research conducted over the 20th century, we now have the Standard Model of particle physics which describes the laws of nature very precisely. Particle accelerators such as the LHC probe the universe at the smallest scale to test the Standard Model rigorously and to find explanation for anomalies that are yet to be explained such as the existence of dark matter.

There are many experiments at the LHC and the CMS experiment is one of the two large and general-purpose experiments. It is composed of many sub detectors and readout from all of them is combined for full event description. Most of the particles are radiated into the endcap regions and get absorbed in the endcap calorimeters. This highlights the significance of reconstruction algorithms in the endcap region.

Although the collisions at the LHC occur at a frequency of 40 MHz, it is not possible to store all of them. A trigger mechanism selects the important events and data is stored at a rate of 1 kHz. The stored events are fully reconstructed offline. Physics analysis is performed on the reconstructed events to search for new physics and to find new particles. The research presented in this thesis is focused on offline computing.

The particles that are collided at the LHC are protons. Currently, approximately 40 proton-proton collisions occur simultaneously. This is referred to as 40 pileup. To observe more rare phenomena, under the High-Luminosity LHC (HL-LHC) project, the experiments at the LHC are receiving major upgrades and will increase the pileup to 200. The CMS experiment plans to install a new High-Granularity Calorimeter (HGCAL) at its endcaps to enable reconstruction in a highly dense 200-pileup environment. The HGCAL, and calorimeters in general, will be discussed in detail in Chapter 3.

Chapter 3

Calorimetry

As this thesis is dedicated to calorimetric reconstruction, the physics of calorimeters is discussed in detail in this chapter. Calorimeters can be divided into two types: electromagnetic and hadronic. After Section 3.1 introduces the concepts common to both both types, the physics of electromagnetic and hadronic calorimeters is then discussed separately in Section 3.2 and Section 3.3, respectively. Afterwards, the HGCAL is discussed in Section 3.4 that contains both electromagnetic and the hadronic parts. Section 3.5 is dedicated to the discussion of jets. Finally, Section 3.6 summarises the chapter.

3.1 Introduction

Apart from embedding particles in a magnetic field, like in the tracker, another way to measure their energy is by a detector called the calorimeter. Unlike the tracker where the particles interact minimally, when a particle enters a calorimeter, it is generally fully absorbed in the material and a cascade of secondary particles, called a shower, is produced. A readout of its interaction with the material gives an estimate of the energy of the particle. The science of measuring energy in this way is called calorimetry. Calorimeters offer two advantages:

1. Unlike the tracker where only charged particles leave tracks, calorime-

ters measure the energy of all the particles, including neutral ones.

2. The energy reconstruction performance improves following a $\sqrt{E}$ relationship, where E is the energy of the incident particle. This is in contrast to the tracker, where the energy reconstruction performance degrades linearly with an increase in momentum.

Calorimeters can be divided into two distinct types: electromagnetic and hadronic. Electromagnetic calorimeters measure the energy of photons, electrons and positrons, whereas energy of the hadronic particles is measured by hadronic calorimeters. The hadronic showers are much longer than the electromagnetic showers and the physics of electromagnetic and hadronic showers, and therefore of the respective calorimeters is different. Calorimeters can be classified as either homogeneous calorimeters or sampling calorimeters. In homogeneous calorimeters, all the material is used for the readout and in sampling calorimeters, the material is divided into a series of active and passive layers. The passive layers have no readout capability. The sampling fraction (f_{sample}) of a sampling calorimeter is defined as the fraction of the energy that a particle deposits in the active part of the calorimeter.

Factors, such as the electronic noise and fluctuations in the cascade of particles in a shower and the fraction particles produced in the active vs passive region of the calorimeters, introduce an uncertainty in the reading. The resulting quality of the energy measurements is defined by two quantities, mean response and resolution. A response is defined as $E_{\text{pred}}/E_{\text{true}}$. Here, E_{true} is the energy of the incident particle and E_{pred}, the energy recorded by a calorimeter. The response of a series of events is then averaged to compute the mean response $\mu\left(E_{\text{pred}}/E_{\text{true}}\right)$. The response can be corrected a posteriori by applying an energy-dependent scaling. However, the energy resolution, measured as the standard deviation of the response distribution $\sigma\left(E_{\text{pred}}/E_{\text{true}}\right)$, cannot be improved in such a simple manner and is therefore more important for measuring the final physics performance. The mean response and resolution are shortened to μ and σ, respectively. It is typical to divide resolution by the mean response to get a mean-corrected resolution (σ/μ), which is often expressed as a percentage (%) quantity. A lower value of resolution signifies better performance.

3.2 Electromagnetic calorimeters

When travelling through material, an electron loses energy by two processes: bremsstrahlung and ionisation. Bremsstrahlung, German for braking radiation, is the dominant factor above a certain energy threshold (ϵ), and ionisation dominates below it. The threshold is called critical energy and is $\sim$ 7 MeV for electrons travelling through lead. This is also shown in Fig. 3.1. ϵ for solids can be approximated as follows:

$$\epsilon = \frac{610}{Z + 1.24}\ \text{MeV}\ . \tag{3.1}$$

Here Z is the atomic number of the material.

High-energy electrons produce photons via bremsstrahlung, which potentially decay into e^-e^+ through pair production (Motz et al., 1969). At low energy ranges, ionisation occurs when electrons interact with molecules and atoms in the material. Photons lose their energy mainly through pair production at high energy ranges. At low energy ranges, photoelectric effect and Compton scattering are the causes of the energy loss. While the phenomenology of shower development is very complex, the main features of electromagnetic showers can be described via simple parametric functions that depend on a single parameter called the radiation length (X_0). It can be approximated as follows and is a characteristic of the material, expressed in g.cm^{-2}:

$$X_0 = \frac{716 \cdot A}{Z(Z+1) \cdot \log(287/\sqrt{Z})}\ \text{g.cm}^{-2}\ . \tag{3.2}$$

Here, A refers to the mass number of the nucleus and Z is defined above.

An electron's energy is reduced by $1/e$ after travelling a distance of X_0 through a material and a photon's energy is reduced by the same amount after travelling a distance of $\frac{7}{9}X_0$. Therefore, the depth of an electromagnetic shower, expressed in radiation length, can be approximated as follows:

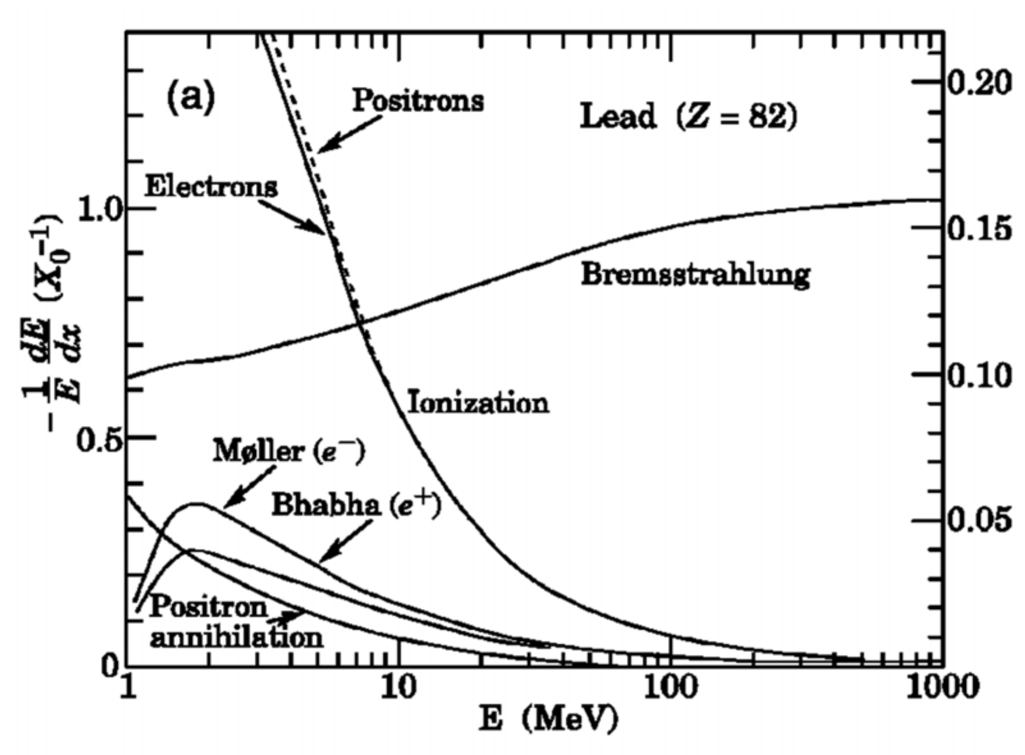


Figure 3.1: Fractional energy lost in lead by electrons and positrons as a function of incident energy (Hagiwara et al., 2002). After $\sim$ 7 MeV, bremsstrahlung is the dominating factor.

$$t_{\max} \approx \ln\frac{E}{\epsilon} + t_0 \ . \tag{3.3}$$

This is called the longitudinal length of the particle shower. It scales logarithmically with the incident energy of the particle (E). This is a useful feature because it restricts the depth of the material required to construct an electromagnetic calorimeter. t_0 takes the value of 0.5 for electrons and positrons, and -0.5 for photons. Electromagnetic calorimeters are typically very compact devices, e.g. the crystal ECAL at the CMS detector is only 23 cm, corresponding to $26 \cdot X_0$.

The spread of a particle shower transverse to its longitudinal axis is measured by the Molière radius (R_M), defined as:

$$R_M \approx \frac{21X_0}{\epsilon}\ \mathrm{g}\cdot\mathrm{cm}^{-2}. \tag{3.4}$$

The radius ϵ is measured in MeV here. R_M represents the average lateral spread of electrons at critical energy after travelling a distance of X_0. About 90% of an electromagnetic shower is contained in a cylinder centered around the longitudinal axis with a radius of R_M, and is only a few centimetres in most calorimeters. It can be observed that the Molière radius is energy independent.

The length of a particle produced in the shower cascade before it interacts is called its track length. The sum of track lengths of all the charged particles, T_0, is proportional to the energy of the incident particle, as given below. T_0 can be measured, for example, by detecting the light in a scintillating material. This principle is used to approximate the incident energy, as follows:

$$T_0 \propto X_0\frac{E}{\epsilon}. \tag{3.5}$$

The resolution of an ideal calorimeter, in which T_0 can be perfectly measured, only depends on the fluctuations in T_0. These fluctuations are a stochastic process and therefore:

$$\frac{\sigma}{\mu} \propto \frac{1}{\sqrt{T_0}} \propto \frac{1}{\sqrt{E}}. \tag{3.6}$$

In a realistic calorimeter, many other factors that degrade the performance are also present. There, the resolution can be summarised as:

$$\frac{\sigma}{\mu} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c. \tag{3.7}$$

The three terms in this equation are the above-defined stochastic term, a noise term and a constant term, respectively. Here, $\oplus$ is the quadratic sum, i.e. $x \oplus y = \sqrt{x^2 + y^2}$.

The stochastic term is very small in homogeneous calorimeters because it is mainly defined by fluctuations in the approximation of T_0. The term is much higher in sampling calorimeters, where the energy that is deposited in the active part of the detector fluctuates on an event-by-event basis. Increasing f_{sample} will decrease the contributions of the stochastic term. The following relationship holds in the sampling calorimeters:

$$\frac{\sigma}{\mu} \propto \frac{1}{\sqrt{N_{\text{active}}}} \propto \sqrt{\frac{t}{E}}. \tag{3.8}$$

Here, t is the thickness of the absorber material, expressed in radiation lengths and therefore, the larger the absorber fraction, the higher is the stochastic term.

The noise term defines electronic fluctuations in the readout equipment. Like the stochastic term, the noise term also has an inverse relationship with the energy of the incident particle and is inversely proportional to the sampling fraction. Calorimeters that use photo detection to collect the readout signal have a lower noise term than the calorimeters, where charge is used for signal generation.

The constant term is the noise contribution that is independent of the energy of the incident particle. It is mainly a function of the construction of the calorimeter. An irregularly shaped detector can have a high constant term, for instance, as the non-uniformity introduces channel-to-channel calibration errors. The constant term can also increase in an environment with a presence of nearby particle. The constant term is the dominating factor at high energies as the other two terms substantially decrease at higher energies because of their inverse relationship with the incident energy.

After combining all the terms, typically, the resolution can be approximated with the following relation:

$$\frac{\sigma}{\mu} = \frac{d}{\sqrt{E/\text{GeV}}}\%. \tag{3.9}$$

The factor d typically varies between 5 and 20, depending on the calorimeter.

3.2.1 Types of electromagnetic calorimeters

One class of electromagnetic calorimeters is semiconductor calorimeters, where detectable electrical signals are generated by electron-hole pairs produced by charged particle. Both silicon and germanium can be used as the semiconductor material. Semiconductor calorimeters are radiation hard and are a good choice for high occupancy environments. However, diode-depletion of thick material to build readout material is both challenging and expensive, and thus, most semiconductor calorimeters are designed as sampling calorimeters with thin active layers.

The second class of electromagnetic calorimeters is called Cherenkov calorimeters. Cherenkov light is produced when an electron or a positron traverses a transparent material at a speed greater than c/n, where c is the speed of light and n is the refractive index of the material. Photo detectors are used for the signal readout. Cherenkov calorimeters are only made as homogeneous calorimeters and have generally worse resolution compared to other homogeneous calorimeters. Common materials for Cherenkov calorimeters are lead glass (PbO) and lead fluoride (PbF_2).

Scintillation calorimeters are another class of calorimeters that also operate by detecting light produced as a result of radiation. Charged particles produce electron-hole pairs in the conduction and valence bands of the material. When the electrons return to the valence bands, detectable photons are produced. The threshold for this to occur is smaller in Scintillation calorimeters compared to Cherenkov calorimeters and thus, a large amount of light is produced. Therefore, Scintillation calorimeters have a better resolution. Scintillation is used in both sampling and homogeneous calorimeters. Examples of materials used in Scintillation calorimeters are $PbWO_4$, CsI and BGO.

Another class of calorimeters is Noble-liquid calorimeters. Here, a noble gas, such as Ar, Kr and Xe is cooled to cryogenic temperatures. Both scintillation and ionisation occur in these calorimeters. Excellent energy resolution can be achieved by only collecting charge in these calorimeters. Ar is most commonly used in sampling calorimeters. It has the advantage

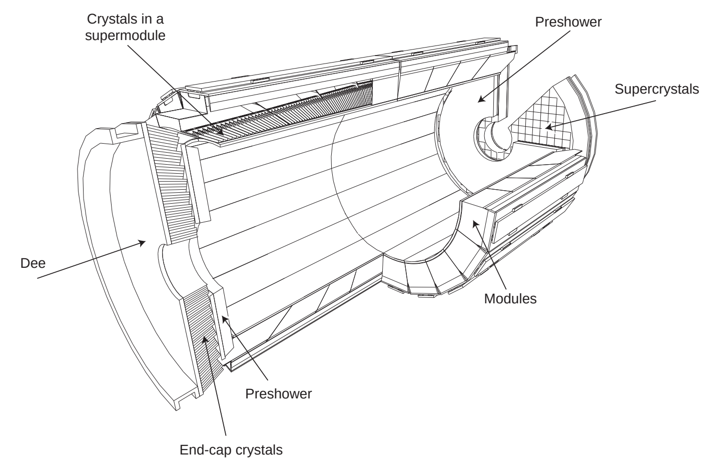


Figure 3.2: A visual layout of the CMS ECAL (CMS collaboration, 2010a).

of being intrinsically radiation hard and allows easier segmentation.

3.2.2 CMS Electromagnetic Calorimeter

The electromagnetic calorimeter of the CMS detector (ECAL) is a homogeneous calorimeter. It is a scintillation calorimeter and uses $PbWO_4$ crystals. A visualisation of the ECAL is shown in Fig. 3.2. The cylindrical part that wraps around the beam pipe ($|\eta| < 1.479$) is the barrel section (EB) whereas the two circular parts at the edges ($1.653 < |\eta| < 3$) form the endcap section (EE). The EE section also has two sampling pre-shower layers in front of it. There, silicon and lead are used as active and passive materials, respectively. The resolution of the ECAL (Equation 3.7) can be defined as follows (CMS collaboration, 2014):

$$\frac{\sigma}{\mu} = \frac{2.8\%}{\sqrt{E}} \oplus \frac{12\%}{E} \oplus 0.3\% \ . \tag{3.10}$$

3.3 Hadronic calorimeters

The physics of the hadronic showers is significantly more complex than electromagnetic showers. Fig 3.3 shows the spectra of particles produced when 100 GeV protons travel through lead. It illustrates the complexity of the physics processes that occur in hadronic showers. While the energetic component is dominated by pions, a large amount of electrons, positrons and photons are also produced. Approximately 1/3rd of the pions will be neutral (π^0). Neutral pions are unstable and decay into two photons ($\pi^0 \rightarrow \gamma\gamma$) before interacting hadronically with the material. Therefore, neutral pions are absent in Fig 3.3, and are shown as electromagnetic particles. The resulting cascade is only an electromagnetic sub-shower; and shows the behaviour described in Section 3.2. As the number of hadronic interactions increases with an increase in the energy of the incident particle, increasingly more interactions will lead to the production of neutral pions (and hence photons). This implies that the fraction of the electromagnetic component (F_{π^0}) of the hadronic shower increases with an increase in the energy of the incident particle. At 100 GeV, F_{π^0} is around 0.5 and increases to 0.7 at 1000 GeV.

In Fig. 3.3, the hadronic particles form the hadronic component of the shower. Some of the positrons and electrons produced here are also considered as part of the hadronic component, as their production occurs with a considerable delay. The hadronic component partially results in what's called invisible energy. The invisible energy can only be detected with a reduced efficiency. If the efficiency of observing an electromagnetic shower is η_e and that of observing a hadronic cascade is η_h, the following fraction of energy induced by a charged pion is visible:

$$E^{\pi}_{\text{vis}} = \eta_e F_{\pi^0} E + \eta_h F_h E \ . \tag{3.11}$$

F_h is the hadronic fraction of the energy i.e. $1 - F_{\pi^0}$. This can be used to evaluate the relative response, $e/\pi = (\frac{E^{\pi}_{\text{vis}}}{E^{e}_{\text{vis}}})^{-1}$ as follows:

$$\frac{E^{\pi}_{\text{vis}}}{E^{e}_{\text{vis}}} = (1 - \frac{\eta_\pi}{\eta_e}) F_h \ . \tag{3.12}$$

The relative response is a characteristic of the material and turns out to be the most important metric for measuring the performance of a hadronic

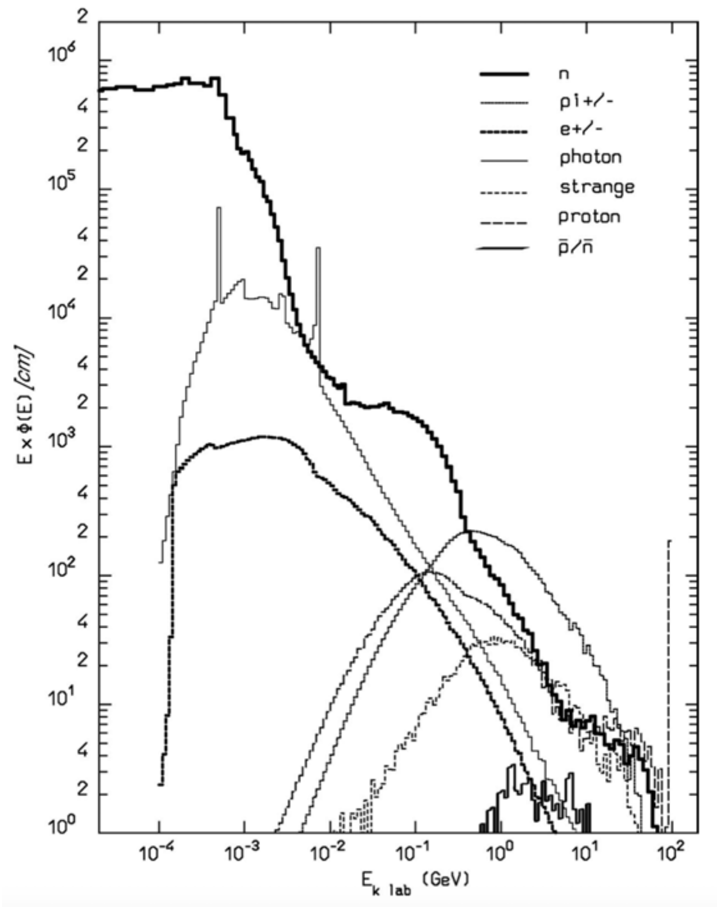


Figure 3.3: The richness of physics produced in the hadronic cascade when 100 GeV protons travel through lead. At low energies, photons, neutrons, electrons and positrons dominate the spectrum. The high energy spectrum is dominated by pions. Figure by (Fabjan and Gianotti, 2003).

calorimeter. The hadron fraction F_h is a random variable with fluctuations occurring on an event-by-event basis. Given the fact that $\eta_\pi \neq \eta_e$, the average response for the hadronic calorimeter cannot be 1 for both electromagnetic and hadronic showers with a linear calibration.

By analogy with electromagnetic calorimeters, where the radiation length (X_0) determines the longitudinal length of the showers, in hadronic calorimeters, nuclear interaction length (λ_I) is the determining factor. It can be approximated as follows:

$$\lambda_I = 35 \cdot A^{1/3} \text{ g} \cdot \text{cm}^{-2} \ . \tag{3.13}$$

The length of hadronic showers, expressed in nuclear interaction lengths, can be approximated as follows:

$$t_{\text{max}} = 0.2 \cdot \log(E/\text{GeV}) + 0.7 \ . \tag{3.14}$$

The longitudinal size of a hadronic calorimeter should be around $10 \cdot \lambda_I$ to ensure adequate containment of the hadronic radiation. This corresponds to a longitudinal length that is over a factor of ten higher than that of the electromagnetic calorimeters. For this reason, the electromagnetic calorimeters are placed in front of the hadronic calorimeters. The lateral profile of the hadronic showers is also larger than the electromagnetic showers. For a 95% containment, a lateral dimension of $1 \cdot \lambda_I$ is required.

The resolution of the hadronic showers is worse than electromagnetic showers and does not scale as $1/\sqrt{E}$. A constant term can be added to approximate the hadronic resolution:

$$\frac{\sigma}{\mu} = \frac{a_1}{\sqrt{E}} \oplus a_2 \ . \tag{3.15}$$

Fluctuations in F_{π^0} on an event-by-event basis are a major part of the hadronic resolution. Furthermore, F_{π^0} is energy dependent, which leads to non-uniform response as the function of incident particle's energy. Additionally, the factors affecting the electromagnetic shower resolution are also present in electromagnetic parts of the hadronic showers.

3.3.1 Types of hadronic calorimeters

Hadronic calorimeters are almost always sampling calorimeters. Their f_{sample} is also very low. Iron (Fe) and lead (Pb) are commonly used passive materials in hadronic calorimeters. In the active part, semiconductors are more commonly used. Liquid-noble gases or scintillators can also be used for the active part.

3.3.2 CMS Hadronic Calorimeter

The hadronic calorimeter at the CMS detector (HCAL) is a sampling calorimeter where brass and plastic scintillators are used as active and passive materials, respectively. Looking from the interaction point, the HCAL is installed behind the ECAL. In Fig. 3.4, its layout is shown.

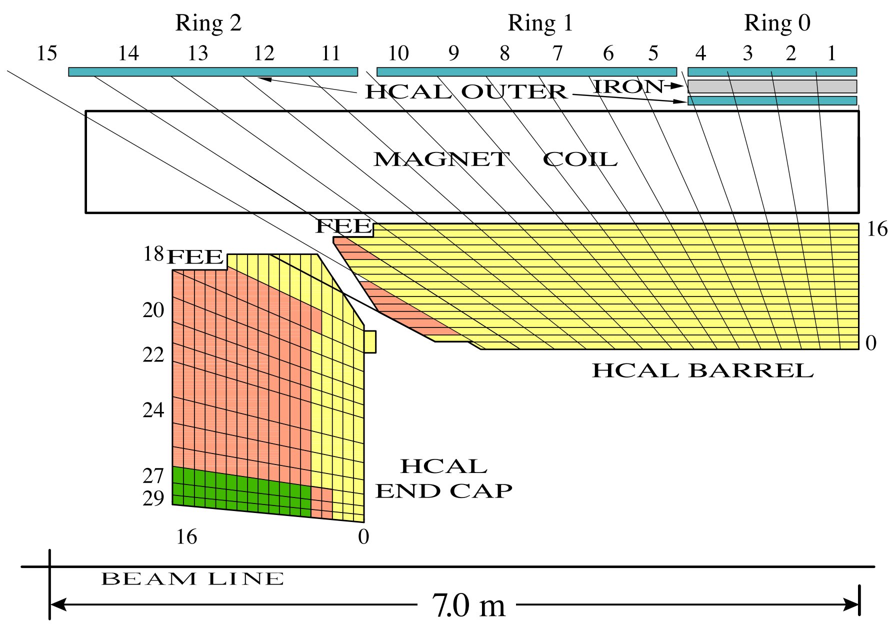


Figure 3.4: A schematic diagram of the CMS HCAL (CMS collaboration, 2010b). The interaction point is at the bottom right side of the figure. Across one η segment, all the cells with the same colour are summed up for the readout.

3.4 HGCAL

The endcap calorimeters of the CMS experiment will be replaced with the HGCAL to sustain the high radiation environment expected at the HL-LHC. Both the HCAL and ECAL will be replaced (CMS collaboration, 2015b). The HGCAL will achieve the following objectives:

- It will be tolerant to increased radiation and suffer minimal degradation in physics performance at 3000 fb^{-1}.
- It will be highly granular, offering fine lateral and longitudinal granularity for adequate separation of showers in 200 pileup environments.
- Add a precise timing signal that was not present in the HCAL and ECAL. This will improve pileup separation performance, and will also aid in the identification of vertex for individual showers.

- Contribute to the L1 trigger.

The HGCAL is the first high granular imaging calorimeter to be ever built. The high granularity will offer the ability to separate showers that are very close to each other. This feature will substantially improve the event reconstruction performance and offer immense physics benefits.

A schematic diagram of the HGCAL is shown in Fig. 3.5. The electromagnetic section of the HGCAL is called CE-E and the hadronic section, CE-H. CE-E corresponds to $26{\cdot}X_0$ and $1.7{\cdot}\lambda_I$. The rest of the HGCAL along the z axis makes up the CE-H which corresponds to $9\lambda_I$. CE-E consists of 28 sampling layers constituting 0.34 m along the z axis. CE-H is divided into two sections. The second section has thicker absorbers compared to the first. In total, there are 24 active layers in the CE-H.

Silicon makes up all the active elements in the CE-E section, while WCu, Cu and Pb act as the absorbers. Stainless steel makes up the majority of the absorber, taking $\sim$ 35 mm between layers in the first stage and $\sim$ 70 mm at the far end of the calorimeter. More detailed description of the layout can be found in CMS Collaboration (2017). The active part of the CE-H consists of silicon as well, except in the low radiation part (low $|\eta|$) where scintillators are used as a cost saving measure.

Laterally, the HGCAL covers the pseudorapidity range $1.5 < |\eta| < 3$. The transverse view of the HGCAL is shown in Fig. 3.6. To maximise the area coverage on circular silicon wafers, the modules are hexagonal in shape. Furthermore, multiple modules are grouped into wedges called cassettes that span either 30° or 60° along the ϕ axis. Thickness along the z axis of the sensors is either 300, 200 or 120 μm. The scintillator section of the CE-H is segmented in the rϕ plane, and is therefore not hexagonal. A module is comprised of either 432 or 192 channels in high and low $|\eta|$ region, respectively. The sizes of the cassettes also shrink with an increase in $|\eta|$, leading to high granularity close to the beam pipe. High spatial resolution is required at high $|\eta|$ due to the presence of a high number of particles. In total, there are 3.9m channels in CE-E, 1.9m and 0.4 in silicon and scintillator parts of the CE-H, respectively. This adds to a total of

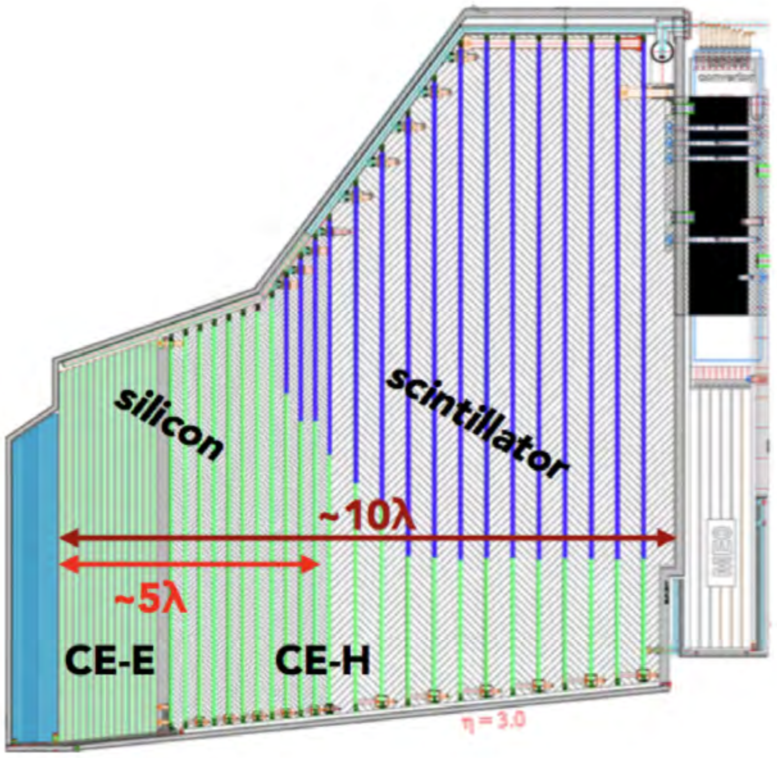


Figure 3.5: A schematic diagram of the HGCAL showing its longitudinal cross section (Martelli, 2017).

about 6 million readout channels (sensors) in both endcaps combined.

In a collisions' event, two values are recorded for each channel: total deposited energy (d_{raw}) and time (t). Deposited energy is the sum of all the charge deposited in one cell by all the particles. If d_{raw} exceeds a certain threshold, the time is recorded with a high resolution for that cell. This implies a certain loss of information in rare cases, i.e., if a higher energy particle hits an already triggered time latch, the recorded time signal will incorrectly correspond to the lower energy particle. The time signal is very useful for disentangling the origin vertices associated with the particles.

The deposited energy is then linearly calibrated using the root mean

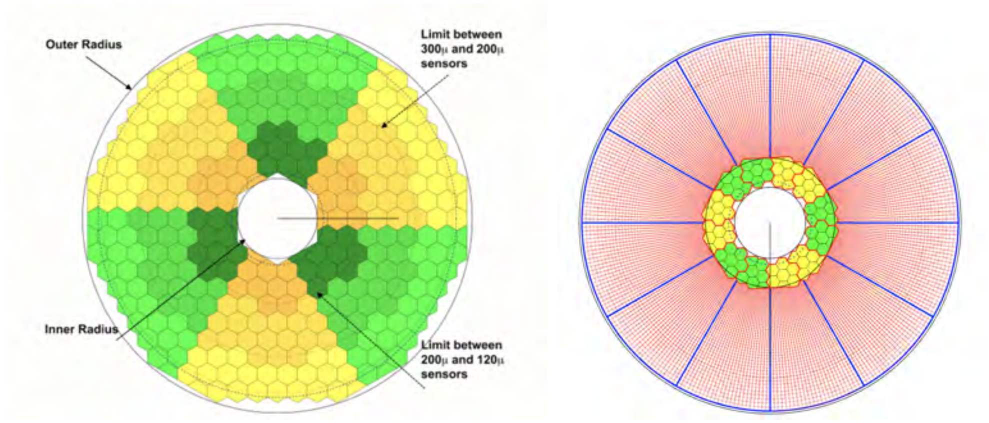

Outer Radius
Limit between 300μ and 200μ sensors
Inner Radius
Limit between 200μ and 120μ sensors

Figure 3.6: Transverse cross section view of the HGCAL (CMS Collaboration, 2017). Left: 9th layer of the CE-E. The sensors are grouped into hexagonal shaped wedges of 60° wedges. Right: 22nd layer of the CE-H. Hexagonal cassettes of silicon are shown in the high $|\eta|$ region whereas uniformly cut wedges across r and ϕ axes are shown in the low $|\eta|$ region.

square (RMS) method as:

$$\hat{W} = \underset{W}{\text{argmin}} \frac{1}{|T|} \sum_{t \in T} \left(\sum_{s \in \mathbb{S}} w_{l(s)} d_{\text{raw}(s,t)} - E(t) \right)^2 . \tag{3.16}$$

T is the ensemble of particles that are used for the calibration. $\mathbb{S}$ labels all the sensors in the calorimeter and $l(s)$, the layer number associated with the sensor s. The calibration vector (W) is an m-dimensional vector where m is the number of layers in the calorimeter. $w_l(s)$ refers to the element in the vector W that is associated to layer corresponding with the sensor s. $E(t)$ refers to the kinetic energy of the particle t. Therefore, $\hat{W}$ is the best estimated calibration vector.

3.5 Jets

In high energy hadronic collisions, such as those at the LHC, a cascade of quarks and gluons, called a parton shower, is radiated from the initial particle. The force between quarks increases with an increase in distance and if a pair of quarks is taken sufficiently far from each other, the potential energy becomes large enough that new particles get created from the vacuum via a process called hadronisation. As the name implies, these particles are generally hadrons and some of them quickly decay into even more particles. Therefore, one quark or one gluon produced in a primary collision creates a spray of high-energy particles that are very close to each other in the $(\eta,\ \phi)$ space. This spray is called a jet. Jets are an integral ingredient for the analysis of particle collisions. Particularly in the forward region, well-resolved individual jet constituents are crucial for a successful pileup removal and for identifying e.g. quark jets over gluon jets in vector-boson-scattering or fusion processes.

Even though jets are created by quarks or gluons, jets are defined only by a clustering algorithm. This implies that the jet clustering algorithm will also label a set of particles as jets that did not originate as a result of hadronisation. The most commonly used jet clustering algorithm is the sequential anti-k_t algorithm (Cacciari et al., 2008).

If there are N particles, a matrix $J \in \mathbb{R}^{N\times N}$ is computed as:

$$J_{ij} = \min({p_T}_i^a, {p_T}_i^a) \times \frac{R_{ij}^2}{R} \ , \tag{3.17}$$

with N as the total number of particles and ij, a pair of particles. The particle pair with the minimum J, $\hat{i}\hat{j}$, is evaluated:

$$\hat{i}\hat{j} = \text{argmin}(J) \ . \tag{3.18}$$

If $J_{\hat{i}\hat{j}}$ is less than ${p_T}_{i'}^a$, with

$$i' = \text{argmin}(\{{p_T}_i^a \forall i \in \{0, 1, 2, ...N\}\}) \ , \tag{3.19}$$

the $\hat{i}th$ and the $\hat{j}$th particles are merged; otherwise, the i'th cluster is considered as a final jet and is subtracted from the list of particles.

R is a hyperparameter, commonly set to a value of 4. R_{ij} is the Euclidean distance in (η, ϕ) space between the ith and the jth particles. a is another hyperparameter and it should be less than 0^1, and is commonly set to -2. It being negative favours high-energy particles. This process continues until either all particles are merged together or until $\min(R_{ij})$ is less than R.

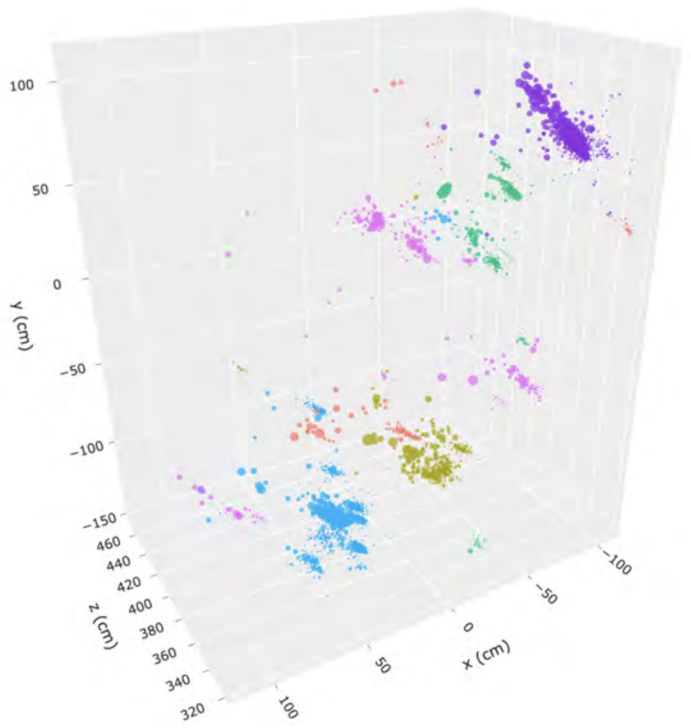
Modern jet clustering algorithms are designed to be infrared and collinear safe. If a jet clustering algorithm is infrared safe, it implies that introduction of soft radiation (i.e. low energy particles) will not impact the set of jets that were produced. Collinear safe jet algorithms are those in which a collinear splitting does not change the jet definition. These properties make jets very useful for studying reconstruction performance of algorithms and compensate for ambiguities in the truth definition.

Hadronisation is a rare process in minimum bias events. To increase the odds of finding high energy hadrons in the forward region, in this thesis, $q\overline{q} \rightarrow t\overline{t}$ are also simulated, where a significant activity in the forward region can be found more commonly. Fig. 3.7 shows two examples of jets clustering algorithm when applied to $q\overline{q} \rightarrow t\overline{t}$ interactions. It can be seen that particles that are close to each other are part of a single jet. This can be compared to Fig. 7.6 in Chapter 7, where an 40 pileup event is shown. A lot more high-energy activity can be observed in only one $q\overline{q} \rightarrow t\overline{t}$ event.

3.6 Summary

There are two ways to detect particles and measure their energy: using trackers and using calorimeters. Minimal material is present in trackers and most particles traverse uninterrupted. Only the charged particles can be observed in trackers. The trajectories of the charged particles get bent due to the presence of a magnetic field and the magnitude of the bending measures their momentum. On the other hand, all particles – whether charged or not – can be observed in calorimeters. The performance of momentum measurement in trackers decreases with an increase in the momentum of the particles whereas, in calorimeters, the energy measurement performance improves with an increase in energy. Therefore, trackers and

[1] < 0 for the anti-k_t algorithm and > 0 for the k_t algorithm



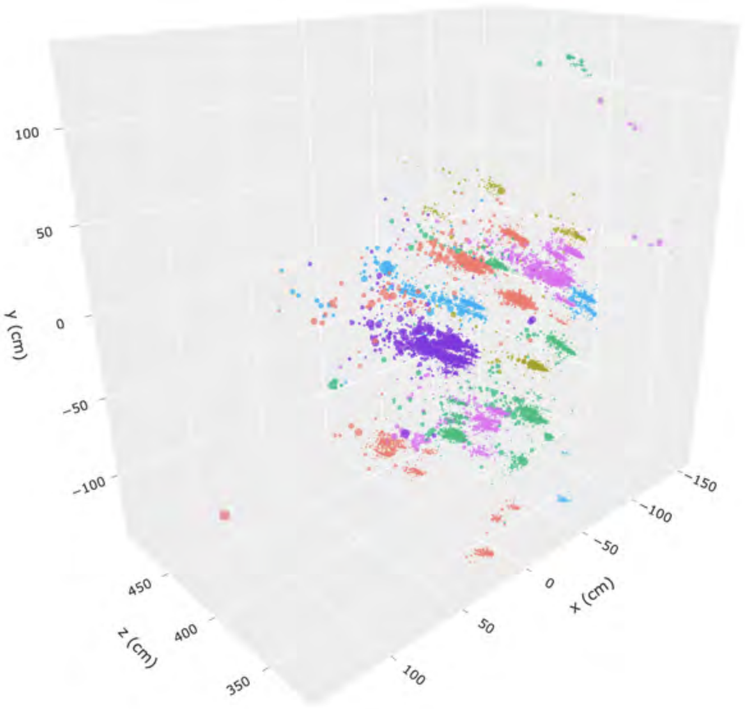


Figure 3.7: Anti-k_t jet clustering algorithm applied to particles produced in two $q\bar{q} \rightarrow t\bar{t}$ interactions. Different colours correspond to different jets.

calorimeters add complementary information and are often used together. In this chapter, the physics of calorimeters is discussed in detail.

Particles are fully absorbed in calorimeters as dense material is present. The signature the particles leave in calorimeters is called particle shower. Calorimeters are divided into many layers and some of them don't have any read-out capability. Such layers are called absorbers. Some layers are made up of active elements, such as silicon, and record the energy of the particles traversing through them. These layers are called active layers.

Photons, electrons and positrons shower very quickly and don't require thick absorbers. These particles are classified as electromagnetic particles and the type of calorimeter used to detect them is called the electromagnetic calorimeter. Other particles, called hadrons, penetrate much more deeply into the material and require thick absorbers. The type of calorimeter used to detect such particles is called the hadronic calorimeter. In large physics experiments, both are used and the electromagnetic calorimeter precedes the hadronic calorimeter.

The HGCAL will consist of both electromagnetic and hadronic sections. It will replace the endcap calorimeters on both sides at the CMS experiment, covering the eta range $1.5 < |\eta| < 3$. It has non-regular geometry as its sensors are of varying sizes and are hexagonal in shape. To enable reconstruction in a highly dense environment with 200 pileup, the HGCAL is highly granular and has 3 million sensors in each endcap.

Jets in particle physics refer to collimated sprays of particles that are produced in high-energy particle collisions via a process called hadronisation. They are also discussed in this chapter as they are detected in calorimeters. Jets are defined by a clustering algorithm and the most commonly used algorithm for clustering is called anti-k_t. It sequentially groups particles together based on their (η, ϕ) distance and favours higher energy particles.

Chapter 4

Classical Reconstruction Algorithms

When electronic readout instruments were introduced in the 70s in particle physics experiments, automated algorithms slowly replaced the human element in tasks such as reconstruction. These algorithms that work without employing modern deep learning techniques are referred to as classical algorithms in this thesis. Even though many tasks are now achieved by modern machine learning based approaches, classical algorithms are still widely used for their simplicity and performance. In this chapter, various classical algorithms used for a number of reconstruction tasks are presented with a two-fold aim: 1) presenting a literature review 2) discussing how full event reconstruction is done at the LHC, setting the stage for the application of the method presented in this thesis. Classical algorithms at both the CMS and the ATLAS experiments are discussed for a comprehensive overview.

Track reconstruction and vertex reconstruction are discussed in Section 4.1 and Section 4.2, respectively. Calorimetric reconstruction is then discussed in Section 4.3. These reconstructed elements from the individual sub-detectors are joined by a linking algorithm that is discussed in Section 4.4. The classical reconstruction algorithm for the HGCAL, TICL, is discussed in Section 4.5. Finally, Section 4.6 summarises the discussions conducted in this chapter.

4.1 Track reconstruction

In 40 pileup, $\mathcal{O}(10^3)$ charged particles travel through the tracker and this will increase to $\mathcal{O}(10^4)$ for the HL-LHC in 200 pileup. Track reconstruction in these high occupancy settings requires highly granular detectors. A tracker has two orders (an order) of magnitude times more readout channels than a typical calorimeter (HGCAL). However, the tracker data is much more sparse owing to the fact that the average number of hits per particle track is only ~ 10. It can be expected that in 200 pileup, $\mathcal{O}(10,000)$ hits will be registered in the tracker. Therefore, the nature of track reconstruction is very different from calorimetric reconstruction.

The CMS tracker consists of two parts: the inner pixel tracker and the outer strip tracker. The inner pixel tracker consists of 66 million pixels over an area of 1 m^2 of silicon, whereas the outstrip detector has 10 million pixels over an area of 200 m^2. The inner pixel tracker at the ATLAS experiment has 80 million pixels over an area of 1.8 m^2 and the strip tracker has 6 million pixels covering an area of 60 m^2. Additionally, the ATLAS experiment has a third layer of the tracker, called the Transition Radiation Tracker (TRT) with 0.3 million readout channels.

At the CMS detector, the track reconstruction begins with local reconstruction, where zero-suppressed signals above certain thresholds are clustered into 3D hits. These hits are then clustered into tracks. An algorithm based on Combinatorial Kalman Filter is used, which is an extension of the Kalman filter (Kalman, 1960). The algorithm is called Combinatorial Track Finder (CTF) (CMS Collaboration, 2014). It is an iterative algorithm where the tracks that are easiest to reconstruct are discovered and masked for the next iterations. There are typically six iterations.

In every iteration, there are multiple steps. The first step is called seeding, and it involves finding a rough estimate of a particle track. A track is defined by five parameters. Three 3D hits are needed for estimation of these parameters. It is also possible to use two hits if a constraint on the origin of the particle is known, e.g. if the particle is originating from the interaction point. With that in mind, seeding is done using highly granular

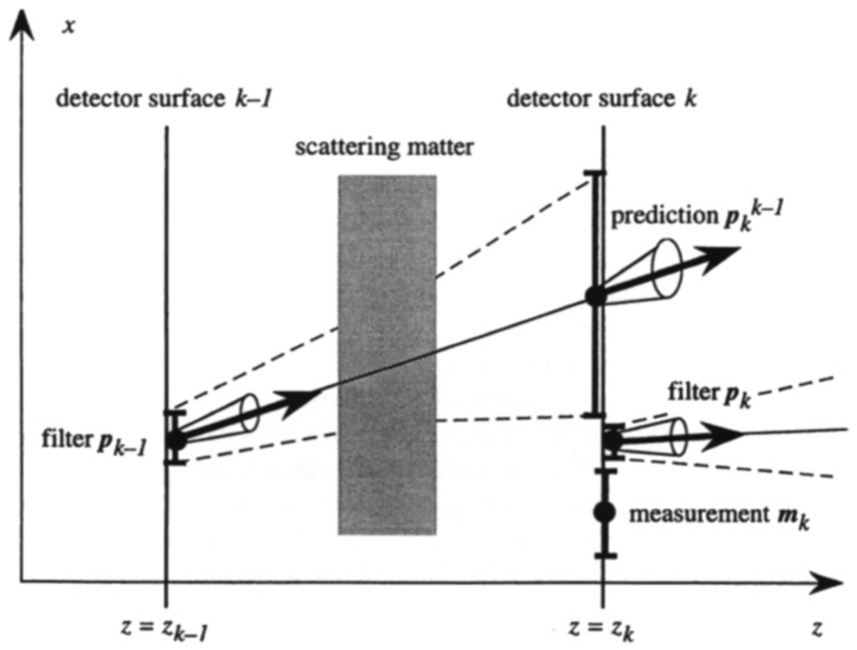


Figure 4.1: Schematic diagram of Kalman filter method (Regler et al., 1996). It is applied at every detector layer iteratively. In this figure, the track has been extrapolated from the $(k-1)$th layer to kth layer with $z = z_k$. The track parameters are then filtered to adapt the prediction p_k^{k-1} according to the measurement m_k. The filtered track parameters produce the state p_k which will be extrapolated at the $(k+1)$th layer in the next iteration.

pixel hits or matched strip hits. A fraction of the strip layers have silicon sensors on the both sides, which are very close to each other. The hits obtained from these layers are called matched strip hits. The second step is called track finding. In the track finding stage, seeds are used to build track candidates. Using the coarse parameters already known by the seeds, the track is propagated in the subsequent layers (generally inside-out), and the new hits are added to each track. The track parameters are updated at each step using a Kalman filter, as visually shown in Fig. 4.1. The third step is called track fitting. At this stage, information from all the hits is combined to get the final estimate of the tracking parameters. A Kalman filter is used again, now in both directions, inside-out and outside-in, and a

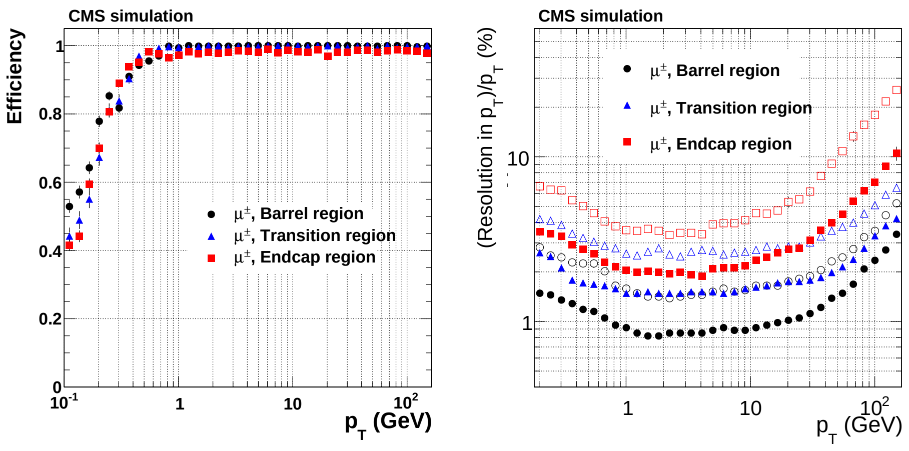


Figure 4.2: Single muon reconstruction performance at the CMS tracker as a function p_T (CMS Collaboration, 2014). Left figure shows efficiency and the right figure shows the p_T resolution. Monte Carlo data with Run 1 settings.

weighted average is used to get the final values of the tracking parameters. The matched strip hits are separated into individual hits to improve the performance at this stage. The last step is called track selection to filter out fake tracks. This is done either by a χ^2 test or by checking compatibility of whether they originated from the interaction point.

The track reconstruction algorithm at the ATLAS experiment is similar to the one used in the CMS experiment and is also based on the combinatorial Kalman filter (Aaboud, Aad, Abbott, Abdallah, Abeloos, S. Abidi, et al., 2017). Instead of one step local reconstruction, local clustering is also part of the iterative process in the ATLAS track reconstruction. A neural network is used to get cluster positions and to identify merged clusters (see Chapter 6).

In Fig. 4.2, single muon track reconstruction performance at the CMS detector is shown. High p_T tracks are easier to reconstruct. The efficiency decreases with an increase in η and is therefore low in the endcap region. The p_T resolution is also shown, which follows the inverse trend and worsens with resolution.

A very similar track finding algorithm, based on Kalman filters, is also used for track reconstruction in the muon chambers. Seeds are built using cathode strip and drift chambers, which are then projected to perform full track reconstruction (CMS collaboration, 2018).

4.2 Vertex reconstruction

The primary vertex refers to the 3D coordinates of the location where a proton-proton collisions occurred. A set of decay products originate from this location as the result of the collision. The direction/position of a particle can be measured with a high precision ($\mathcal{O}(\mu\text{m})$) in the tracker. This fact is important for the reconstruction of the primary vertices. Clustering is performed on the reconstructed tracks to associate them with their primary vertices. The number of primary vertices is a random variable and is not known. At the CMS detector, deterministic annealing (DA) algorithm (Chabanat and Estre, 2005; Rose, 1998) is used for this task. In the DA algorithm, z coordinates of the extrapolated interaction point (also called the point of closest approach) for every track based on the tracking parameters are assigned to a set of assumed vertices. Initially, only one vertex is assumed but is increased iteratively. At the ATLAS experiment, an algorithm called adaptive multi-vertex finder and fitter is paired with deterministic annealing (ATLAS collaboration, 2019; Piacquadio et al., 2008). Here, the tracks not assigned to a vertex are analysed to find the most likely interaction point to build a candidate vertex. Close-by tracks are then assigned to the candidate vertex. It is possible that a track can be assigned to more than one vertex. Once this is done, the tracking parameters of all the tracks assigned to the new candidate vertex are fitted using a Kalman filter and outliers are progressively de-weighted via deterministic annealing.

4.3 Calorimeter clustering

At the CMS detector, a traditional clustering algorithm runs independently on all the calorimeters (i.e. EE, EB, two ECAL preshower layers, HE, HB).

Therefore, for each run, the clusters are only two dimensional. These are much simpler detectors when compared to the HGCAL and hence, use of such algorithms is easier.

As the first step, N seeds are identified as all the cells where 1) the energy deposit is higher than a threshold 2) the energy deposit is higher than on the neighbouring cells. Once the seeds are identified, it is assumed that each of the seeds corresponds to a Gaussian cluster. An iterative expectation-maximisation algorithm based on Gaussian-mixture models re-evaluates the parameters of the associated Gaussian functions. In each iteration, in the first step, the fraction of energy deposit on jth cell by ith cluster is computed as follows (CMS collaboration, 2017):

$$f_{ji} = \frac{A_i \cdot \exp(-(\vec{c_j} - \vec{\mu_i})/(2\sigma^2))}{\sum_i^N A_k \cdot \exp(-(\vec{c_j} - \vec{\mu_k})/(2\sigma^2))} \ . \tag{4.1}$$

Here, c_j and μ_i refer to the position of jth cell and ith seed, respectively, in the (η, ϕ) plane. The initial value of the seed's position (μ_i) and the amplitude (A_i) are simply the position of the cell and the energy deposit on that cell. These are re-evaluated in the second step as follows:

$$A_i = \sum_j^M f_{ji} E_j \ , \tag{4.2}$$

$$\vec{\mu_i} = \sum_j^M f_{ji} E_j \vec{c_j} \ . \tag{4.3}$$

Here, M refers to the number of cells.

Once the algorithm converges, the amplitude and position are taken as the energy and position of the built clusters.

At the ATLAS experiment, a slightly different method is used (Aad et al., 2017). Seeds are identified as follows:

$$|E^{EM}_{\text{cell}}| > S \cdot \sigma^{EM}_{\text{noise,cell}} \ . \tag{4.4}$$

E^{EM}_{cell} and $\sigma^{EM}_{\text{noise,cell}}$ refers to the energy deposited and the average noise on a cell, respectively. Both of these values are taken on the electromagnetic scale. S is a hyperparameter and generally takes the value 4. The seeds are then grown to the neighbouring cells if the following condition is satisfied sequentially in the decreasing order of $|E^{EM}_{\text{cell}}|/\sigma^{EM}_{\text{noise,cell}}$:

$$|E^{EM}_{\text{cell}}| > N \cdot \sigma^{EM}_{\text{noise,cell}} \ . \tag{4.5}$$

N is another hyper parameter that generally takes the value 2. If a neighbouring cell passing the threshold is another seed or belongs to another cluster, the two corresponding clusters are merged. The procedure repeats iteratively until all the neighbouring cells satisfying the condition,

$$|E^{EM}_{\text{cell}}| > P \cdot \sigma^{EM}_{\text{noise,cell}} \tag{4.6}$$

are collected. P is generally set to 0, collecting all the cells with a non-negative energy signal; however, it can be increased to increase the purity of the algorithm. The rule $S > N \geq P$ applies while selecting the hyperparameters.

It is important to note that the absolute value of the energy deposit ($|E^{EM}_{\text{cell}}|$) is taken. This is because negative signals can also be generated in the calorimeters and they can also contribute to the growth and formation of clusters. These negative signals correspond to out-of-time particles and electronic noise and can provide insight into these effects. While reconstructing physics objects, such as jets, only the clusters with positive energy are considered.

4.4 Particle flow algorithm

The tracker and the calorimeters can both measure the energy of particles. At low p_{T}, the tracker provides a better energy resolution than the calorimeters, and vice versa at high p_{T}. Additionally, only the calorimeters can measure the energy of neutral particles. Hence, an improved event description can be obtained after combining information from the tracker and the calorimeters. This holistic approach is called particle flow. The reconstructed muons add further information to improve the global event reconstruction performance.

The particle flow paradigm used in the CMS experiment is detailed by CMS collaboration (2017). The clusters and tracks obtained from various sub-detectors are called particle flow elements and they are joined via a link algorithm. Any pair of particle flow elements can be checked for compatibility. The condition to allow a link depends on the type of sub-detectors. To improve computational time of $\mathcal{O}(N^2)$, only the elements that are close to each other in (η,ϕ) plane are tested, obtained via a k-dimensional tree (Bentley, 1975).

To link a track to a cluster, the last hit of the track is extrapolated to two layers of the pre-shower and to ECAL and HCAL to depths corresponding to the expected longitudinal electron shower profile and one nuclear interaction length, respectively. The track is then linked if the projected position is within an area defined by the union of associated cells of a cluster. The area is enlarged to account for the gaps between the cells and modules of the calorimeters. A link distance is computed in (η, ϕ) plane that defines the quality of the link. If multiple particle flow elements are linked, only the link with the smallest distance is retained. In a similar way, muon tracks from muon chambers are linked to tracks formed in the central tracker. Fig. 4.3 visualises a set of linked particle flow elements.

Dedicated algorithms are also created for specialised processes such as bremsstrahlung photons (CMS collaboration, 2015a), which have a high probability of converting to an e^+e^- pair already in the tracker. If a pair of electron and positron are very close in the η space and a compatibility is found, they are linked. Similarly, a set of tracks can also be linked to each other if they form a secondary vertex comprising at least 3 tracks, one of which is the incoming track.

For electrons and single charged hadrons, the energy measurement from the calorimeter (E) and the momentum measured from the tracker (p) can be combined as follows:

$$E_{\text{corr}} = \frac{\sigma_p^{-1} p + \sigma_E^{-1} E}{\sigma_p^{-1} + \sigma_E^{-1}} \ . \tag{4.7}$$

Here, σ_E and σ_p refer to the expected resolution of the calorimeter and the tracker. This only occurs if both measurements are compatible i.e. if the positions and the momentum/energy do not significantly different.

If the calorimetric energy measurement is significantly higher than the track momentum, a neutral particle is created, which could either be a photon or a neutral hadron, depending on how much energy is in the HCAL compared to the ECAL.

A similar linking algorithm is also employed at the ATLAS experiment (Aaboud, Aad, Abbott, Abdallah, Abeloos, S. H. Abidi, et al., 2017). The metric used for track-cluster linking is defined as follows:

$$\Delta R^{`} = \sqrt{\left(\frac{\Delta\eta}{\sigma_\eta}\right)^2 + \left(\frac{\Delta\phi}{\sigma_\phi}\right)^2} \ . \tag{4.8}$$

σ_η and σ_ϕ are widths of the clusters in η and ϕ spaces, respectively, computed as the standard deviation of the constituent cells of the associated clusters. The cluster that is closest in $\Delta R^{`}$ is linked to a track.

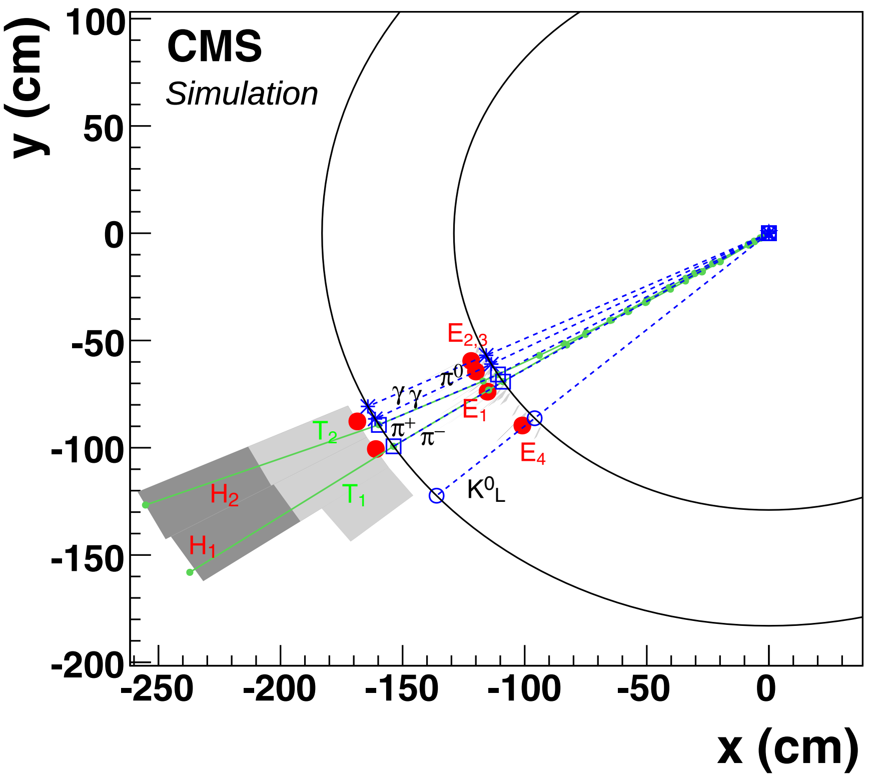


Figure 4.3: A visualisation of particle flow algorithm at the CMS experiment (CMS collaboration, 2017). The two circular surfaces correspond to ECAL and HCAL surfaces. Tracks T_1 and T_2 correspond to two hadronic particles that created showers H_1 and H_2 in the HCAL. In addition, there are also four tracks (shown in blue) that correspond to four electromagnetic showers in the ECAL (E_1,...,E_4).

4.5 The iterative clustering framework (TICL)

In parallel to research conducted in this thesis, a more traditional method, called The Iterative CLustering framework (TICL), uniquely for the HGCAL reconstruction, is also being developed (Cristella, 2021; Pantaleo and Rovere, 2022). It is closely related to the iterative tracking method discussed in Section 4.1. The architecture diagram of the TICL framework is shown in Fig. 4.4.

The first step is clustering, performed independently at different layers

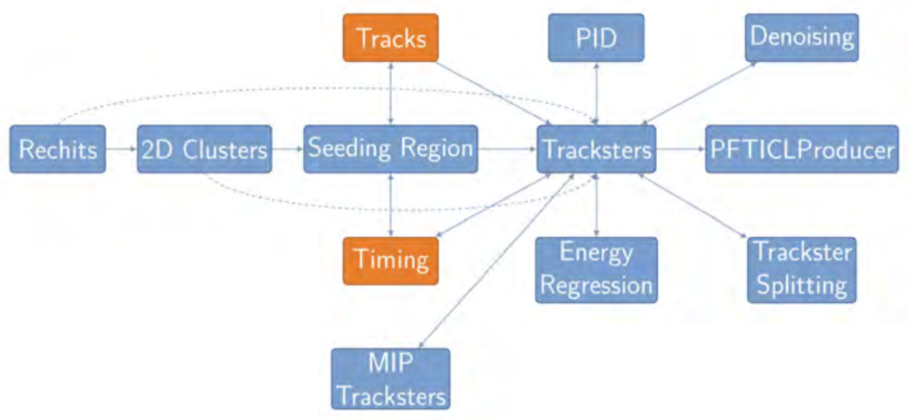


Figure 4.4: An architecture diagram of the TICL framework (Cristella, 2021).

of the HGCAL. This is done via the CLUE algorithm (Rovere et al., 2020). Local energy density (ρ) is computed for each rechit as:

$$\rho_i = \sum_{h \in H} \chi(i,h) E_j \ . \tag{4.9}$$

Here, χ refers to a convolutional kernel which that operates on all the neighbours of a rechit:

$$\chi(i,h) = \begin{cases} 1 & \text{if} \quad \Delta(i,h) = 0 \\ 0.5 & \text{if} \quad \Delta(i,h) \leq d_c \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

Δ is the distance in the (η, ϕ) space.

The CLUE algorithm also computes a quantity (δ) that is defined as distance to the closest neighbour with a higher ρ. Based on these quantities, the algorithm picks the rechits with locally maximum ρ as seeds and labels the ones surrounding as their followers. The rest of the rechits are labelled as noise.

The layer clusters are then linked to form tracksters. This is done by a pattern recognition module that can either be CLUE3D, an extension to the CLUE algorithm, FastJet (Cacciari et al., 2012) or Cellular Automaton. Once the tracksters are established, they are also linked to reconstructed elements from other sub-detectors in a particle-flow like fashion to improve the performance. Finally, the reconstructed clusters are masked and the next iteration is executed.

4.6 Summary

Computation is fundamental to the operation of the LHC and its experiments. In this chapter, some of the classical reconstruction algorithms are discussed.

Charged particles traverse through a tracker in a helical curve. Track reconstruction involves fitting a curve on the sensor hits. It begins with seeds which are collected in the initial highly-dense layers of the tracker. Using Kalman Filters, the track is then progressively extrapolated to the outer layers. Thousands of charged particles traverse through the tracker in modern physics experiments which makes track reconstruction a challenging task. Another pattern recognition task that needs to be performed in the tracker is called vertex finding i.e. estimating the 3D coordinates of the primary proton-proton collisions. Deterministic annealing is generally used for this purpose.

Classical calorimetric reconstruction algorithms generally make use of the grid-like structure of the existing calorimeters and are generally applied independently on different layers. Local maxima of energy deposits on the grid elements are used as seeds which are then grown to find all the clusters. The classical approach to reconstruction in the HGCAL is called TICL. It doesn't rely on a grid-like structure but uses local neighbours to find the seeds independently on different layers which are then linked.

Calorimeters and the tracker add complementary information to perform full event reconstruction. The holistic approach which combines information from different sub-detectors is called particle flow. At the CMS and the ATLAS experiments, track and calorimetric reconstruction are first performed independently. A particle flow algorithm then links the reconstructed tracks to their clusters based on their (η, ϕ) proximity and compatibility in energy.

Owing to the intractability of the data, classical algorithms pose computational challenges. They have to be carefully tuned to account for a wide range of cases that can occur. For example, two calorimetric clusters sometimes have to be linked, and specialisations have to be made for particle decays in the tracker. In some cases, such as for track reconstruction, the classical algorithms work well but are too compute-intensive. The use of machine learning based approaches will alleviate this issue as such approaches can naturally profit from modern parallelisable hardware.

Chapter 5

Machine Learning Background

In this chapter, basic concepts of machine learning are discussed. The discussion is divided into a general overview of artificial neural networks in Section 5.1, followed by convolutional neural networks and graph neural networks in Section 5.2 and Section 5.3, respectively. A literature review of the application of artificial neural networks in high energy physics is also presented in each section. A brief overview of methods used to predict a variable number of objects using neural networks is presented in Section 5.4. Finally, Section 5.5 summarises the chapter and discusses the research gap addressed in this thesis from a broader perspective after concluding the literature review.

5.1 Artificial neural networks

Artificial neural networks (ANNs), or simply neural networks (NNs), are computing systems – loosely inspired by biological neural circuits – that can be trained to learn various properties. While the history of artificial neural network dates back to the 50s (Rosenblatt, 1958), the computers of that time were too slow for wide adoption of the ANNs. With the advancements in computers and the introduction of parallel processors, their usage has skyrocketed in the past two decades. Availability of large-scale datasets has also aided this trend.

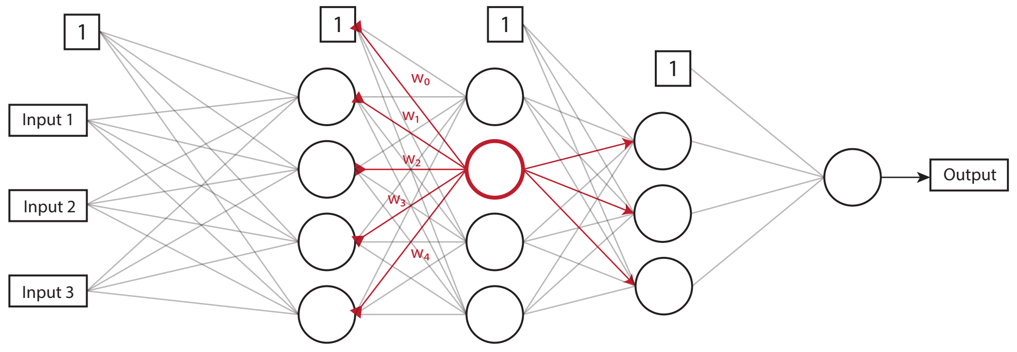


Figure 5.1: An example of three-layer MLP that takes three inputs and produces a single output. One of the neurons is highlighted in red and it is shown how it is taking inputs from four neurons in the preceding layer and multiplying it with a set of learned weights ($\mathbf{w}^\circ$). The output of this neuron is then forwarded to three neurons in the subsequent layer.

ANN is a general term used to describe a wide range of trainable computing systems. The simplest form of ANN is called the Multi-Layer Perceptron (MLP). It comprises a set of neurons that are grouped into different layers. The output of a neuron at a certain layer is defined as follows:

$$l^\circ = \sum_{i=1}^{N} w_i^\circ x_i + w_0 = \sum_{i=0}^{N} w_i^\circ x_i = \mathbf{w}^\circ \cdot \mathbf{x} \ . \tag{5.1}$$

Here, x_0 is always 1 and x_i represents the ith input to the neuron, which is either an input signal or output of another neuron. $\mathbf{w}^\circ$ is the weight vector of the neuron. w_i° corresponds to the contribution of the ith input and w_0 is called the bias term.

At the first layer, every neuron is connected to the input signals, represented as a vector. At the second and the subsequent layers, these neurons are connected to all the neurons in the preceding layer. The layers which are not at the output or the input are often referred to as hidden layers. An example of an MLP is visualised in Fig. 5.1. An MLP is frequently also referred to as a dense neural network, especially if the layers are part of a bigger network.

The weights of all the neurons in a layer can be summarised in a matrix (let's say W) in which the nth row corresponds to the weights of the nth neuron. In a two-layer MLP, the output can be written as $y = W_2(W_1\mathbf{x})$. By the associative property, $W_2(W_1\mathbf{x}) = (W_2W_1)\mathbf{x} = W_3\mathbf{x}$. This defeats the purpose of having multiple layers. A non-linear activation function ϕ is added to prevent this, $y = W_2(\phi(W_1\mathbf{x}))$. Two-layer MLPs with a non-linear activation function are universal function approximators (Hornik et al., 1989).

Some of the widely used activation functions are given below. These activation functions are applied independently to every output of the neural network (l).

1. Rectified linear unit (ReLU)

$$\phi(l) = \max(0, l) \tag{5.2}$$

2. Exponential linear unit (ELU)

$$\phi(l) = \begin{cases} \alpha(\exp(l) - 1) & \text{if} \quad l \leq 0 \\ l & \text{otherwise} \end{cases} \tag{5.3}$$

   Here, α is a hyperparameter.

3. Hyperbolic tangent

$$\phi(l) = \tanh(l) \tag{5.4}$$

4. Sigmoid

$$\phi(l) = \frac{1}{1 + \exp(-l)} \tag{5.5}$$

5.1.1 Training an ANN

The weights of the neural network are initialised with random values. In the modern approaches, it is common to scale the weights by a factor of $1/M$, where M is the number of weights in a layer. This is done via a method called Xavier initialisation (Glorot and Bengio, 2010). A series of examples, called a training dataset, are then processed by a learning algorithm with the goal of finding a set of weights that achieve the best performance on those examples. This process is called training the neural network.

Training is done iteratively. At every iteration, a cost function (also known as loss function – f_{loss}) is evaluated that quantifies the quality of the predictions of the neural network. A lower value indicates better performance. This is generally done by comparing the output of the neural network to a defined truth in the training dataset. The weights of the neural network are then updated to decrease the cost function. If the quantity to be learned is a scalar, the mean squared error (MSE) is a commonly used cost function:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \ . \tag{5.6}$$

y is the output of the neural network and $\hat{y}$ represents the target. The cost is then averaged over all the training examples.

Gradient descent (Cauchy et al., 1847) is used as the standard learning algorithm. Gradient descent updates the weights as follows per iteration:

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \eta\frac{\partial f_{\text{loss}}}{\partial \mathbf{w}} \tag{5.7}$$

$\mathbf{w}$ is the vector representing the weights of all the layers in a neural network. The $-\frac{\partial f_{\text{loss}}}{\partial \mathbf{w}}$ term evaluates the direction in which to update the weights in order to decrease the cost function. The surface representing the values of the cost function on all the possible weight values is called the cost surface. The cost surface is almost never convex and therefore has local minima and inflection points. The gradient of the cost function at a minimum is 0 and is very close to 0 around it. Therefore, it can occur that the gradient descent cannot get out of a minimum, halting the training process. η in Equation 5.7 is called the step length, which defines the scale by which to update the weights. If η is too low, the learning process is slower and the risk of getting into a local minimum is also higher. If η is too high, the learning might not occur at all. Even though η can be set to a fixed value, generally an algorithm called the optimiser is used that deduces the step length in a more sophisticated way to increase the time of convergence. One such algorithm is the momentum optimiser, which takes the following form. m and α are momentum and learning rates, respectively, and are taken as hyper-parameters of an ANN. v_i is velocity, which is evaluated for every parameter at every time step as follows:

$$\mathbf{v}_i = m\mathbf{v}_{i-1} - \alpha * \frac{\partial f_{\text{loss}}}{\partial \mathbf{w}} \ , \tag{5.8}$$

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mathbf{v}_i \ . \tag{5.9}$$

The most widely used optimiser in modern approaches is the Adam optimiser (Kingma and Ba, 2014), and it follows the update process given below:

$$\mathbf{m}_i = \beta_1 * \mathbf{m}_{i-1} + (1-\beta_1)\frac{\partial f_{\text{loss}}}{\partial \mathbf{w}} \ , \tag{5.10}$$

$$\mathbf{v}_i = \beta_2 * \mathbf{v}_{i-1} + (1-\beta_2)\left(\frac{\partial f_{\text{loss}}}{\partial \mathbf{w}}\right)^2 \ , \tag{5.11}$$

$$\hat{\mathbf{m}}_i = \frac{\mathbf{m}_i}{1-{\beta_1}^i} \ , \tag{5.12}$$

$$\hat{\mathbf{v}}_i = \frac{\mathbf{v}_i}{1-{\beta_2}^i} \ , \tag{5.13}$$

$$\hat{\mathbf{w}}_i = \mathbf{w}_{i-1} - \alpha * \left(\frac{\hat{\mathbf{m}}_i}{\sqrt{\hat{\mathbf{v}}_i}} + \epsilon\right) \ . \tag{5.14}$$

Here, β_1, β_2, and ϵ are taken as hyper parameters and are commonly set to 0.9, 0.999 and 10^{-7}, respectively. α is the learning rate, again, commonly set to 10^{-4} or 10^{-3}.

It is not possible to use all the training examples in an iteration if the training dataset is too large. The gradients can be approximated via stochastic gradient descent (SGD) (Bottou and Bousquet, 2007) in that case. In SGD, a randomly sampled subset is sampled from the training dataset without replacement in every iteration. The sampled subset of the training dataset is called the mini-batch. If the training dataset is exhausted during the iterative process, it is referred to as an epoch. The whole dataset is refilled after every epoch. n epochs indicate that a learning algorithm has used every example in the training dataset n times. The SGD approach has an added advantage that it is less likely to getting stuck in a local minimum as fluctuations in the mini-batches yield slightly different gradients.

The process of evaluating the output and the loss of a neural network is called the forward pass. If all the operations performed during the forward pass are differentiable, $\frac{\partial f_{\text{loss}}}{\partial \mathbf{w}}$ can be evaluated through an algorithm called backpropagation (Rumelhart et al., 1986), an application of the gradient chain rule. A matrix multiplication, for instance, is a differentiable operation, and even though the ReLU activation function is not, its gradient can be evaluated through sub-derivatives (Clarke, 1990). The cost function should also be differentiable. In a N-layers neural network, $\frac{\partial c}{\partial \mathbf{w}} = \lfloor \frac{\partial c}{\partial W_1} \rfloor \oplus \lfloor \frac{\partial c}{\partial W_2} \rfloor \oplus \lfloor \frac{\partial c}{\partial W_3} \rfloor \cdots \lfloor \frac{\partial c}{\partial W_N} \rfloor$, with $\lfloor X \rfloor$ representing the matrix flattening operation and $\oplus$, vector concatenation.

$$\begin{aligned}\frac{\partial c}{\partial W_m} &= \frac{\partial \mathbf{o}_m}{\partial W_m}\frac{\partial W_{m+1}}{\partial \mathbf{o}_m}\frac{\partial \mathbf{o}_{m+1}}{\partial W_{m+1}}\cdots\frac{\partial W_{N-1}}{\mathbf{o}_{N-2}}\frac{\partial \mathbf{o}_{N-1}}{\partial W_{N-1}}\frac{\partial \partial W_N}{\partial \mathbf{o}_{N-1}}\frac{\partial \mathbf{o}_N}{\partial W_N}\frac{\partial c}{\partial \mathbf{o}_N} \\ &= \frac{\partial \mathbf{o}_m}{\partial W_m}\frac{\partial W_{m+1}}{\partial \mathbf{o}_m}\frac{\partial c}{\partial W_{m+1}}\end{aligned} \tag{5.15}$$

$\mathbf{o}_n$ is the output of the mth layer of the neural network and W_m, the weights. It can be observed that the gradient calculations at the mth layer can be condensed to use the gradient of the $(m+1)$th layer, by evaluating gradients backwards. This process is called the backward pass. Software libraries such as Tensorflow (Martín Abadi et al., 2015), PyTorch (Paszke et al., 2019) and Jax (Bradbury et al., 2018) can automatically compute the aforementioned gradients and therefore, generally, only the forward pass needs to be programmed.

In summary, the weights of the neural network are updated iteratively via the forward and backward passes until the cost function becomes stable, at which point the neural network is considered as trained. The training dataset is expected to be drawn from a stationary distribution that adequately represents the problem. However, it is possible that the learning process will over analyse the training distribution, and when tested on another dataset sampled from the same distribution, it will not yield equally good performance as on the training dataset. This is called overfitting. On the other hand, if a neural network is unable to sufficiently learn even the sampled training distribution, it is referred to as underfitting. The extent to which a neural network achieves the same performance on various datasets drawn from the problem distribution measures its generalisation capability. A subset of the available data, called the testing dataset, is withheld to measure the generalisation performance of the neural network. The generalisation performance on a testing dataset can only be measured so many times as otherwise, the selection on the basis of testing dataset performance is in itself a training process. This often referred to as data contamination.

5.1.2 Applications in high energy physics (HEP)

The earliest use of ANNs in high energy physics dates back to the 80s. Denby (1988) investigated track and cluster finding with neural networks. The neurons in their neural network were only connected to their neighbouring cells. The tracks were discovered by applying a neural network iteratively until convergence. In every iteration, the nearby track hits were connected by the neural network. This is shown in Fig. 5.2. Application of neural networks for cluster finding and trigger were also studied in the same work. Another work, Peterson (1989), was published at the same time where the authors used a very similar method for the task of track reconstruction. Another early example of the application of neural networks in HEP is by P. Abreu et al. (1992) where the classification of hadronic decays of the Z_0 boson into b and c quark pairs at the DELPHI detector at CERN's LEP (DELPHI collaboration, 1991) using a set of 17 jet features was investigated. The authors used a three-layer MLP with 25 nodes in the hidden layer and employed the MSE loss for the classification task, and the target was either 0 or 1 for the two classes [1]. A more comprehensive review of the usage of ANNs in HEP in the 90s was conducted by Denby (1999).

In a more recent work, the discrimination of jets originating from the

[1]An interesting choice looking from the modern perspective when the cross-entropy loss is the most popular choice for classification tasks

top quark and the other partons was investigated by Almeida et al. (2015) using CMS HCAL data. This was achieved by using a four-layer MLP and the cross-entropy loss. At the ATLAS experiment, MLPs are used for local clustering in the tracker (ATLAS Collaboration, 2014) [2]. The approach is similar to Denby (1988), albeit the neural network being more powerful here. In another interesting application of neural networks, Baldi, Sadowski, et al. (2014) used neural networks to search for exotic particles by training a neural network for signal and background classification. Another recent example of an application of neural networks is for jet substructure classification (Baldi, Bauer, et al., 2016).

It is evident from the above-mentioned studies that dense neural networks have been shown to work for a variety of tasks and are an integral component of high energy physics. Similarly, gradient boosted decision trees have also been shown to yield the state-of-the-art results on many applications. They were first utilised by Yang et al. (2005) for particle identification at the MiniBooNE detector (MiniBooNE collaboration, 2009) and were later crucial to the discovery of the Higgs Boson at both the CMS and the ATLAS experiments (ATLAS Collaboration, 2012; Chen and He, 2015; CMS Collaboration, 2012).

[2]also see Section 4.1

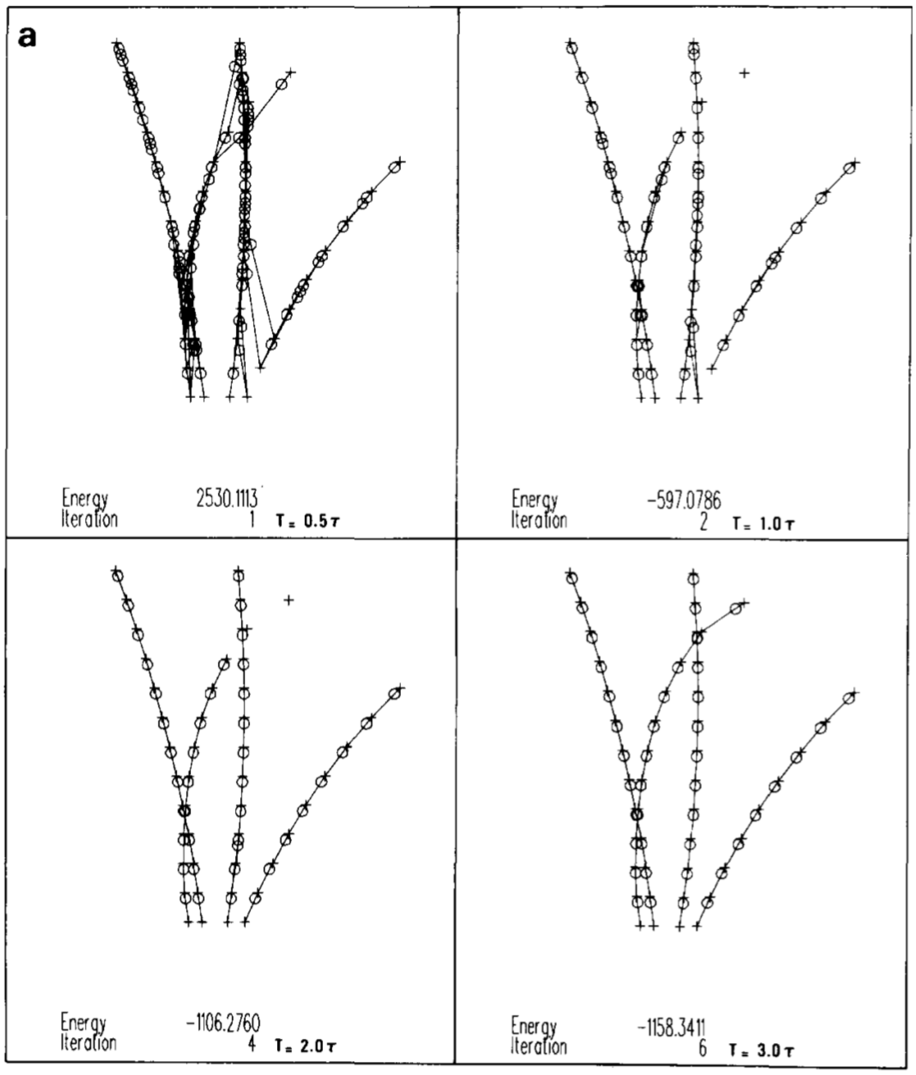


Figure 5.2: Track finding using neural networks in the 80s (Denby, 1988). For a certain track hit, the neural network decides whether to connect to all the other points in its near vicinity defined by a threshold in an iterative fashion. In this example, the neural network has perfectly reconstructed four tracks.

5.2 Convolutional neural networks

The use of MLPs in grid-like data with large dimensions requires an extremely large number of weights. E.g. for a 1000×1000 pixel image, a

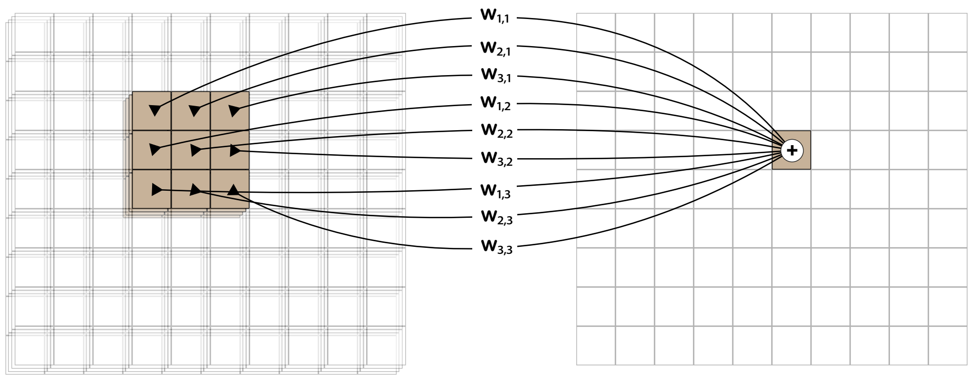


Figure 5.3: A CNN filter: dot product of the feature vector at a certain location is taken with various weights vectors. The resulting scalar numbers are added together and placed in the output grid at the location corresponding to the centre of the filter.

billion parameters are required in just one layer. Convolutional Neural Networks solve this by sharing the weights for neighbouring pixels and exploit the translation equivariance found in most image data. Like an MLP, a CNN is also divided into a set of layers, and within each layer, there are N filters. Each of the filter slides through the grid and produces an output at every grid point, resulting in another grid with N features:

$$l_{x,y} = \phi\left(\sum_{i=0}^{M-1}\sum_{y=0}^{M-1} \mathbf{w}_{i,j} \cdot \mathbf{z}_{x-\lfloor M/2\rfloor,y-\lfloor M/2\rfloor}\right) \, . \tag{5.16}$$

Here, the vector $\mathbf{z}_{k,l}$ refers to input features at location (k,l) of the input grid. The filter weights are also arranged in an MxM grid and the weight vector at location (m,n) is $\mathbf{w}_{m,n}$. M is an odd number. This is also visualised in Fig. 5.3. Zero padding is often applied at the edges to keep the total number of pixels constant at the output of a CNN layer.

In CNN architectures, it is common to reduce the number of pixels with each layer and, at the same time, increase the number of features. This is either done by a pooling operation where every group of neighbouring $c \times c$ pixels are reduced to one pixel by either taking the maximum value of all

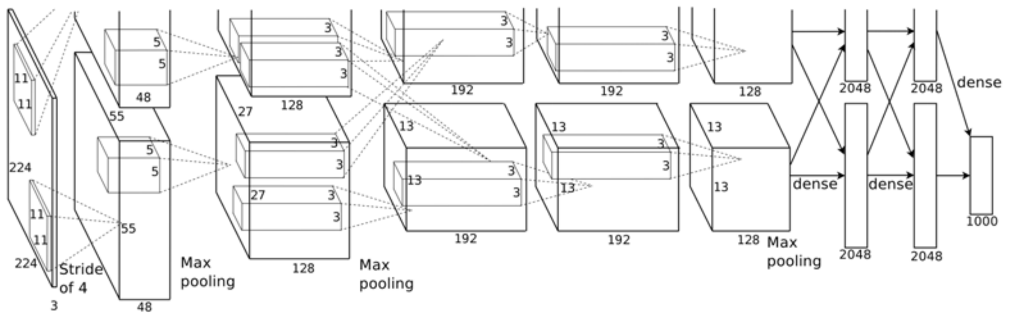


Figure 5.4: AlexNet: The first CNN that achieved ground-breaking results on the ImageNet challenge (Krizhevsky et al., 2012). The number of pixels is decreased with progressive application of CNN filters. At the end, the features are concatenated and passed through a set of dense layers. A 1000-dimensional output signifies 1000 output classes in the ImageNet dataset.

the filters or the average. It is also possible to reduce the dimensionality by applying a strided convolutional filter. In strided convolution, the filters are applied only every nth row and nth column. In a typical classification task, at the end of the last convolutional layer, all the features are flattened into a single vector. An MLP is then applied to this feature vector.

While the CNNs date back to the 50s (Hubel and Wiesel, 1959), their use was mainstreamed in 2012 when AlexNet (Krizhevsky et al., 2012) achieved state-of-the results on the ImageNet challenge (Deng et al., 2009), henceforth beginning the deep learning revolution. The CNN architecture used by Krizhevsky et al. (2012) is shown in Fig. 5.4. In recent years, CNNs have since been applied to medical imaging (Suzuki, 2017), document processing (Gilani et al., 2017), agriculture (Kamilaris and Prenafeta-Boldú, 2018), and many other fields.

5.2.1 Applications in high energy physics

One of the earliest study on application of CNNs in high energy physics was done by (Aurisano et al., 2016) at the NOvA detector at Fermilab (Ayres et al., 2007). A deep CNN was used to classify neutrino events without full reconstruction and improved the efficiency ν_e classification by 40%.

While the neutrino detector produces 3D data, in the study, only X and Y views were taken as input to the CNN. Another earliest study was done by Oliveira, Kagan, et al. (2016) where jet images from calorimeters were taken as input to a CNN to classify QCD jets. The authors also studied correlations of per-pixel input to the output of the network to investigate how the neural network learns the jet substructure.

Komiske, Metodiev, and Schwartz (2017) also used calorimeter images for classification of quark and gluon jets. The input jet image was constructed by summing up calorimeter deposits in the nearby (η, ϕ) region. The architecture is visualised in Fig. 5.5. Many other studies have since been done using CNNs in high energy physics for various classification tasks (Acciarri et al., 2017; Ai et al., 2018; ATLAS Collaboration, 2017; Choi et al., 2019; De Oliveira et al., 2020; Fraser and Schwartz, 2018; Guo et al., 2018; Kasieczka et al., 2017; Komiske, Metodiev, Nachman, et al., 2018; Macaluso and Shih, 2018; Racah et al., 2016; Renner et al., 2017).

Full simulation of physics data using, for instance, GEANT4 (Agostinelli et al., 2003), is a time and computationally intense process. In a very interesting and novel application, Oliveira, Paganini, et al. (2017) showed that generative adversarial networks (GANs) (Goodfellow et al., 2014) can be paired with CNNs to generate jet images much faster than running full GEANT4 simulations. Similarly, it has also been shown that calorimetric data can be generated using GANs (Paganini et al., 2018). Variational autoencoders (Kingma and Welling, 2013) can also be paired with CNNs for the task of fast simulation (Touranakou et al., 2022). Numerous studies have since been done on pairing CNNs with generator networks for fast simulation (Belayneh et al., 2020; Buhmann et al., 2021; Di Sipio et al., 2019; Erdmann et al., 2019; Musella and Pandolfi, 2018; Salamani et al., 2018).

In applications more related to this thesis, CMS Collaboration (2017) used CNNs to improve energy reconstruction performance and for identification of single-particles in the HGCAL. A coarse regular grid was superimposed on the non-uniform data of the HGCAL to get an an input compatible with CNNs. Similar work was also done for the Future Circular

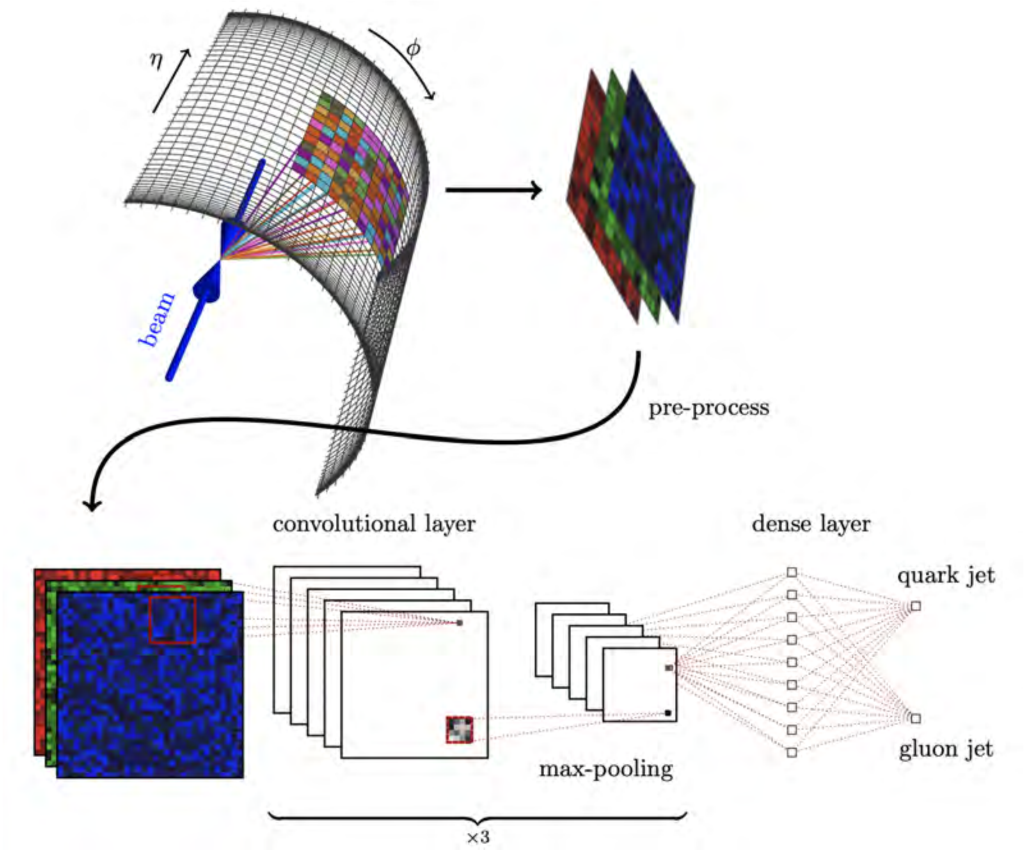


Figure 5.5: The CNN used by Komiske, Metodiev, and Schwartz (2017) to classify jet images into quarks and gluons.

Collider[3] (FCC collaboration, 2019a). In a more recent work, Akchurin et al. (2021) also used CNNs to improve the energy reconstruction performance in standalone calorimeters for single-particles. This is shown in Fig. 5.6. These works relate to the discussion in Section 3.3 and show that software compensation using neural networks can significantly improve reconstruction performance.

In an interesting application, Neubüser et al. (2022) employed CNNs to improve the longitudinal design of calorimeters while taking the software compensation into account.

[3]FCC collaboration (2019b)

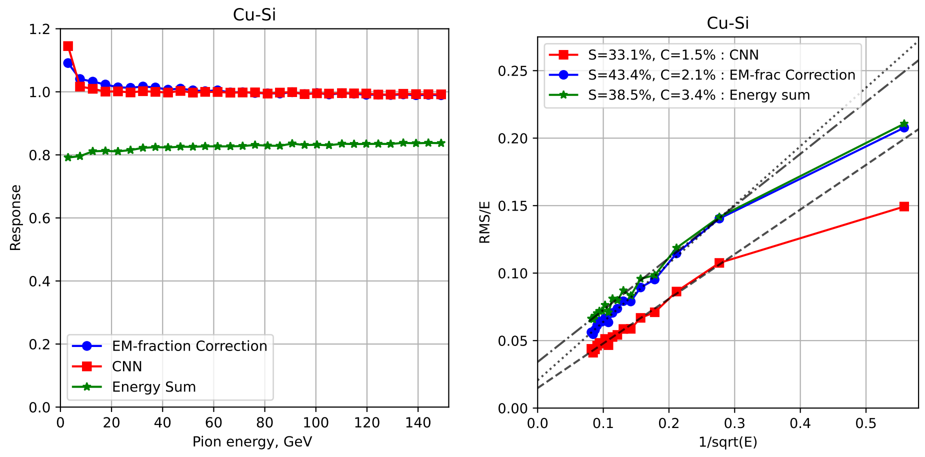


Figure 5.6: A CNN is used to improve the response (left) and resolution (right) of hadron energy reconstruction (Akchurin et al., 2021).

5.3 Graph neural networks

While CNNs are powerful neural networks, not all the data can be represented in a grid-like structure. Examples of such data include molecular data, social networks, or documents. This data can, however, be represented as a graph. A graph is a very generic datastructure, i.e. images, audio clips, sets, can all be represented as graphs. A graph is composed of a set of objects called vertices or nodes. Any pair of nodes can be linked via an edge to signify, for instance, a real-world connection between the connected nodes. The link can either be directional or non-directional. All the nodes that a certain node has an edge with are called neighbours of that node. Mathematically, a graph G is a tuple, comprising a set of vertices V and a set of directed edges E and undirected edges U:

$$G = (V, D, U) \ , \tag{5.17}$$

$$D = \{(x, y) \ni x \in V, y \in V\} \ , \tag{5.18}$$

$$U = \{\{x, y\} \ni x \in V, y \in V\} \ . \tag{5.19}$$

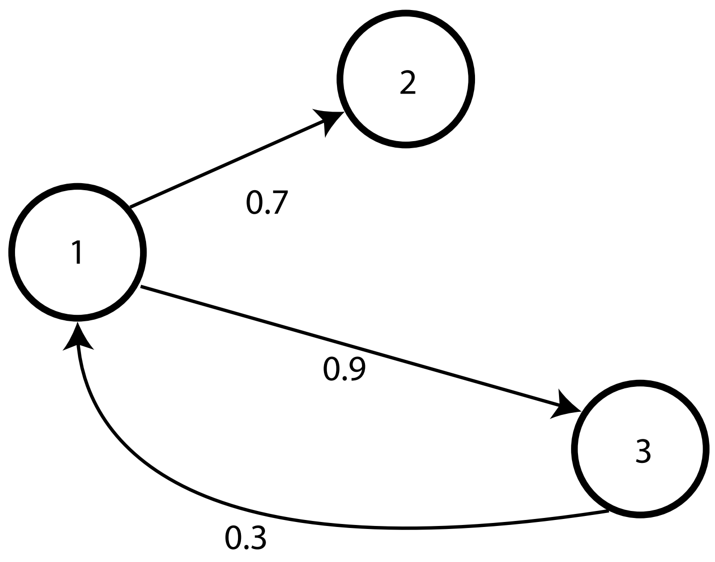


Figure 5.7: A very simple directed graph. Nodes are shown as circles and edges as arrows, representing the direction of the edge. The edges are also weighted.

The graph is called a directed graph if $|U| = 0$ and similarly, an undirected graph if $|D| = 0$.

One or more labelling functions can also be defined on either the edges or nodes to label various properties. It is common to assign a number to all the edges via a labelling function, $w(v) \in \mathbb{R}$, $\forall\ v \in V$, making the graph a weighted graph. An example of a weighted graph is shown in Fig. 5.7. The graph connectivity can also be expressed in the form of an $N \times N$ adjacency matrix A. The diagonal is always 1 and A_{ij} element of the matrix is 1 if ith node is connected to jth node. It can also be any other number in the case of a weighted graph. For undirected graphs, $A^T = A$.

A Graph Neural Network (GNN) is an artificial neural network that operates on graph data. For example, a GNN can take as input a set of molecules in a drug, represented as a graph, and produce the probability that the drug is safe for human consumption at the output. The graph neural network model was introduced in (Scarselli et al., 2008) as an extension of recursive neural networks. The model includes an iterative diffusion algorithm. A processing neural unit is associated with each node of the graph, and information via this unit is propagated according to the graph connectivity. This continues iteratively until the graph reaches the state of equilibrium, at which point, the output of the graph is taken as the output of the neural network. This work was extended by Li et al. (2015) using gated recurrent units (GRUs) (Cho et al., 2014) for propagation of information between the nodes. GRUs are a neural technique commonly used in recurrent neural networks for learning sequences. One class of GNNs is called the spectral graph neural networks, where the spectrum of the adjacency matrix is analysed. In graph convolutional neural networks (Bruna et al., 2013), the convolutional operation is done on the Fourier domain by Eigen decomposition of the Laplacian of the graph. A lot of research has since been done on spectral GNNs (Defferrard et al., 2016; Henaff et al., 2015; Kipf and Welling, 2016).

In non-spectral approaches, operations such as convolution operations are defined directly on the graph (Duvenaud et al., 2015), i.e. on the neighbourhood of nodes. Atwood and Towsley (2016) defined a diffusion operation, i.e. by taking powers of the transition matrix. A more recent GNN architecture is called GraphSAGE (Hamilton et al., 2017). For forward propagation, GraphSAGE samples a fixed number of neighbours for each of the nodes and information is aggregated over these sampled neighbours. The aggregator functions are order invariant. In the study, `mean` and `max` were used for aggregation. Other examples of non-spectral approaches are graph attention networks (Veličković et al., 2017) and message passing neural networks (Gilmer et al., 2017).

The data produced by the HGCAL is only composed of a set of points, i.e. all the hits with energy above a threshold. In this data like this, the graph structure is defined a-priori and has to be dynamically learned. One of the earlier work extending GNNs to such point clouds is the Dynamic Graph Convolutional Neural Network (DGCNN) (Y. Wang et al., 2019). In the DGCNN, the points are transformed to a learned space and neighbours are defined as the K nearest neighbours for each of the node in this learned space. The information is then aggregated on the neighbourhood of graph nodes using the edge convolution operation. In the edge convolution, the convolution operation is performed on edges defined by difference of features of the associated nodes.

Another recent work on points cloud was done by Qasim, Kieseler, et al. (2019). Two GNNs, GravNet and GarNet, were introduced. In GarNet, N virtual nodes (called aggregators) are added to the graph and are connected to all the non-virtual nodes. The information is then aggregated over these nodes by permutation-invariant aggregation functions. The GravNet is similar to the DGCNN. It is used in this thesis and is therefore described in Chapter 7 in more detail.

5.3.1 Applications in high energy physics

Henrion et al. (2017) used Message Passing Neural Networks (MPNNs) (Gilmer et al., 2017) to classify whether the origin of a jet was a W boson or QCD. It was shown that this method outperforms the previously proposed recursive neural networks for the application (Louppe et al., 2019). Abdughani, J. Ren, et al. (2019) searched for stop pair production (i.e. classifying signal from background) using the fully connected MPNN approach. They show that this method outperforms MLPs. The MPNN is also used for event classification by the same authors to study Higgs coupling (Abdughani, D. Wang, et al., 2021; J. Ren et al., 2020).

A GNN architecture called ParticleNet was proposed by Qu and Gouskos (2020). It is based on DGCNN (Y. Wang et al., 2019) hence it operates on points cloud. The authors used it for quark/gluon classification and top tagging. Bernreuther et al. (2021) used the same architecture, ParticleNet, for semi-visible and top boosted top jet classification.

Moreno, Cerri, et al. (2020) and Moreno, Nguyen, et al. (2020) used interaction networks (Battaglia et al., 2016) for multi-class classification of jet, gluons and hadronic decays. Mikuni and Canelli (2020) used Graph Atten-

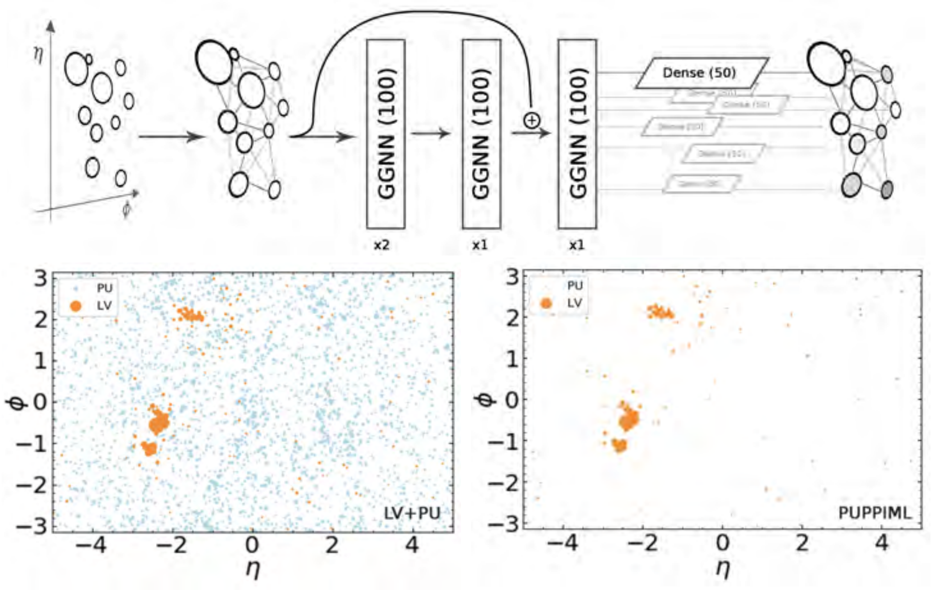


Figure 5.8: Pileup subtraction in 20 PU event an architecture based on Gated Graph Neural Networks (top). The ground truth is shown on the left and the prediction of the GNN is shown on the right (Martínez et al., 2019).

tion Networks (Selvan and Arutchelvan, 2021) for quark-gluon classification and pileup mitigation. Pileup mitigation has also been studied using Gated Graph Neural Networks (Martínez et al., 2019), achieving state-of-the-art performance (Fig. 5.8).

5.4 Neural inference of variable number of entities

For tasks such as classification or regression, a neural network can be designed in a straightforward fashion. For classification, an N-dimensional vector is the output of the neural network, where N is the number of classes. However, many problems require an inference of the variable number of entities. An example of such a task is optical character recognition (OCR) where the number of characters it not known in an image. Specialised techniques need to be developed in order to achieve this task. For OCR, recurrent neural networks are used to generate a sequence (Breuel et al., 2013) until a specific token is generated (Graves et al., 2006).

Another example is object detection or instance segmentation in computer vision. Object detection refers to the prediction of bounding boxes of various objects of interest (e.g. cars and people) in an image. Research on object detection has been going on for decades, starting from feature-extraction methods (Dalal and Triggs, 2005; Lowe, 1999; Zou et al., 2019). Faster R-CNN (S. Ren et al., 2015), proposed in 2014, was the first time an end-to-end deep learning based algorithm was used for this task. A region proposal network generates a set of bounding boxes proposals. Non-maximum suppression (Neubeck and Van Gool, 2006) is used to resolve duplicates by filtering out the redundant bounding boxes. Another example of an object detection network is Single Shot Detection (SSD) (W. Liu et al., 2016), where the need for a proposal network is eliminated.

Instance segmentation is similar to object detection, but instead of predicting bounding boxes, individual pixels are assigned to a set of predicted objects. A recent work on instance segmentation uses SSD to first build bounding boxes; and then assign every pixel to the object associated with each box (Uhrig et al., 2018). Kendall et al. (2018) performed instance segmentation by pointing to the object centre at each pixel; and the object centres are later clustered. In a more recent work (Neven et al., 2019), intersection-over-union is directly optimised. Another recent work on instance segmentation was performed by Zhang and Wonka (2021). Another segmentation technique called Object Condensation was proposed by Kieseler (2020). Object Condensation is more general is also suited for graph data that is not grid-like, and is therefore, used in this thesis. As it is the basis of the GNN method presented in this thesis, it is described in more detail in Chapter 6.

5.5 Summary

In this chapter, introduction to machine learning is presented as well as its various applications in high energy physics. For reconstruction, it was evidenced in Chapter 4 that classical methods are very complex in nature. They have to be carefully tuned to account for a wide range of cases that can occur. For example, two calorimetric clusters sometimes have to be linked, and specialisations have to be made for particle decays in the tracker. The LHC data is also becoming increasingly more intractable in nature, which makes it challenging to use classical algorithms, and motivates the use of more data-driven approaches. The applicability of various data-driven machine learning approaches for a variety of tasks in particle physics was established in this chapter. Further, sub-optimal choices for parameters over multiple steps of classical reconstruction algorithms can have significant impacts on the overall performance, hinting at need for end-to-end neural network based reconstruction. The discussion in Section 5.4 showed that substantial research has been done on similar tasks, such as object detection, in the field of computer vision.

In high energy physics, however, while there is some research on using CNNs for lightweight reconstruction (Pol et al., 2022), no work has been done on neural network based end-to-end reconstruction in complex modern detectors such as the HGCAL prior to the research presented in this thesis. In tasks such as object detection, however, a grid-like structure is assumed. Graphs are a generic data structure and are are well suited for complex data that, for example, cannot be represented in a grid-like format. Therefore, the research presented in this thesis is a novel application of neural inference of a variable number of entities on graph-like data.

Chapter 6

Graph Neural Networks for Particle Reconstruction

The HGCAL data cannot be easily represented in a grid-like format due to non uniformity and sparsity. This motivates the use of graphs. However, the HGCAL data is only a point cloud corresponding to the sensors activated by the particles. The data format is further discussed in Chapter 7, in Section 7.1.2 and Section 7.2. Therefore, the graph structure – through which the information is propagated – need to be learned via a dynamic approach. The state-of-the-art neural network applicable to this type of problem is called the Dynamic Graph Convolutional Neural Network (DGCNN) (Y. Wang et al., 2019). However, given the resource constraints even for offline reconstruction, and the large number of input input that are expected in 200 pileup events, the DGCNN approach is not feasible for the task of HGCAL reconstruction. Another dynamic GNN, GravNet, is faster and has been shown to work very well for calorimetric problems outperforming the DGCNN (Qasim, Kieseler, et al., 2019). Therefore, in this thesis, GravNet is investigated for the reconstruction problem, as described in Section 6.1. Every hit is taken as a graph node as an input in GravNet. These nodes remain the same throughout the execution of the neural network while the graph structure is re-defined using KNN in a learned latent space in every application of the GravNet layer. This graph structure is

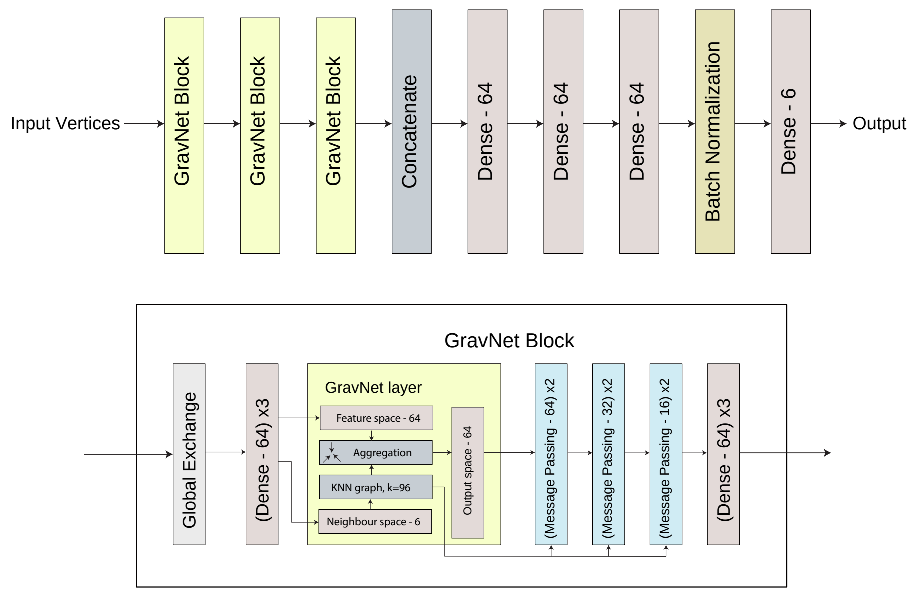


Figure 6.1: Architecture diagram of the graph neural network used for reconstruction

used for the propagation of information.

At the output of the neural network, the objective is to generate the particles which produced the hits that were given as input. In the truth definition, every hit is associated to either a true particle or noise.[1] Therefore, one way to define this problem is as a hit clustering problem. Object Condensation (Kieseler, 2020) is used as the loss function to achieve supervised clustering. This is discussed in Section 6.2.

The training process is presented in Section 6.3. The clusters predicted by the neural network don't need to be evaluated during the training phase. This is required only at the inference time and the method to do so in Section 6.4.

Finally, the summary of the chapter is presented in Section 6.5.

6.1 Neural network

The architecture diagram of the neural network is visualised in Fig. 6.1. It consists of three GravNet blocks, each of which has multiple message passing and dense layers. The output of all the GravNet blocks are concatenated together, and it is followed by more dense layers. To describe the neural computation graph precisely, the input and output features at a layer n are represented as X_n and X_{n+1}, respectively. The vertices are always represented as a set H, and hence X_{n_h} indexes the input feature vector corresponding to node h at the nth layer.

The input to the neural network is a set of detector hits X_0, and each hit is comprised of a column vector. r, η, ϕ (x, y, z) are the boost-invariant cylindrical coordinates (Cartesian coordinates) of the sensor, A is its area, and e is the deposited energy. The input features are not all independent, i.e. x and y, for instance, can be inferred from η, ϕ and z; however, adding them to the input has near-zero impact on the computational cost and can help train the neural network faster. The variable t representing the time when the sensor registered the energy is only present in the HGCAL data (it is set to 0 in the toy calorimeter[2]):

$$X_0 = \{[r, \eta, \phi, x, y, z, \mathrm{A}, e, t]_h \forall h \in H\} \ . \tag{6.1}$$

At the output of the neural network, six values are produced for each hit:

$$X_N = \{[x, \beta', \psi, \phi]_h \forall h \in H\} \ . \tag{6.2}$$

Here, x corresponds to three-dimensional clustering space coordinates, β' is used to evaluate the confidence score, ψ is energy correction factor and ϕ is the local distance scaling factor.

Global exchange is a normalisation layer and does not have any trainable parameters. Mean, maximum, and minimum values are computed independently for all the features across the input set, and concatenated ($\oplus$) with

[1]Procedure 2 discusses this in more detail in Chapter 7.
[2]The toy calorimeter is described later in Section 7.1.1

features of each node, as a form of message passing:

$$X_{n+1} = \big\{ \big(X_{n_h} \oplus \texttt{mean}(X_n) \oplus \texttt{min}(X_n) \oplus \texttt{max}(X_n) \big) \, \forall h \in H \big\} \ . \tag{6.3}$$

The number of features increases by $4\times$ for each node with every application of the global exchange layer.

The dense layer ($\mathcal{D}_*$) is independently applied to all the vertices multiple times in the neural network (as shown in Fig. 6.1). $\mathcal{D}_*$ has not been indexed and refers to a unique dense layer every time it is used. ReLU is used as the activation layer (ϕ) in all the dense layer, except the last layer where there is no activation. Dense layer is also used as a part of the GravNet layer. W_n and b_n are the trainable weights of the dense layer. The number of columns of W_n is equal to the dimensions of the input vector (x) and the number of rows corresponds to the number of output features. b_n is the bias term with the same cardinality as the output features.

$$\begin{aligned} X_{n+1} &= \mathcal{D}_*(X_n) \\ &= \big\{ \big(\phi(W_n X_{n_h} + b_n) \big) \, \forall h \in H \big\} \end{aligned} \tag{6.4}$$

The distribution of the features at a certain neural network layer changes as the neural network learns and updates its weights. This phenomenon is called the internal covariance shift. It slows down the learning process because intermediate neural network layers have to learn to accord with the shifted distribution. A batch normalisation layer (Ioffe and Szegedy, 2015) reduces the internal covariance shift by normalising using mini-batch statistics. $\texttt{E}(X_n)$ and $\texttt{var}(X_n)$ represent the expected value and variance of the features. During the training phase, these are computed over the mini-batch. In the neural network presented in this thesis, the vertices are flattened across multiple inputs and are used independently for the computation of the mini-batch. Moving statistics are used for the computation of

$\mathtt{E}(X_{n_h})$ and $\mathtt{var}(X_{n_h})$ during inference:

$$X_{n+1} = \left\{ \left(\frac{X_{n_h} - \mathtt{E}(X_n)}{\sqrt{\mathtt{var}(X_n)}} \right) \forall h \in H \right\} \ . \tag{6.5}$$

6.1.1 GravNet layer

To dynamically learn the graph structure, in GravNet, the feature space of the input vertices is transformed into two distinct spaces: coordinate space (S) and feature space (F), via dense layers ($\mathcal{D}_*$):

$$S = \mathcal{D}_*(X_n) \ , \tag{6.6}$$

$$F = \mathcal{D}_*(X_n) \ . \tag{6.7}$$

The transformation into the coordinate space (S) is linear, i.e. the dense layer does not have an activation function, and the dense layer used in the feature space transformation uses the ReLU activation. The dimensionality of S is chosen to be lower compared the dimensionality of F. This improves the computational performance by defining neighbours in low-dimensional coordinate space; while the feature information is carried in a high-dimensional space. In the neural network presented in this thesis, the feature space has 64 dimensions and the coordinate space, 6.

A graph is dynamically defined using KNNs, i.e. for a node h, its neighbours are the K nearest node in S.

$$D = \{\cup\{(x, h, d(x, y)) \, \forall x \in \mathrm{KNN}(h)\}\} \ . \tag{6.8}$$

The resulting graph is directed, and the neighbours of a node are pointed towards the said node. The edge weight, defined by labelling function d, is the exponentially decayed Euclidean distance from the node h:

$$d(x, h) = \exp\left(-\|S_x - S_h\|^2\right) \ . \tag{6.9}$$

For every node h, the feature space information is then aggregated by taking mean and max over all its neighbours ($\mathcal{N}(h)$). These functions are called aggregation functions; and any perturbation-invariant function can be used. The aggregated information is weighed by the edge weight as a form of attention mechanism. It is followed by a subtraction from the features of the node h:

$$G_h = F_h - \texttt{mean}(\{d(x,h)F_x \ \forall x \in \mathcal{N}(h)\}) \ , \tag{6.10}$$

$$I_h = F_h - \texttt{max}(\{d(x,h)F_x \ \forall x \in \mathcal{N}(h)\}) \ . \tag{6.11}$$

The weighing operation is necessary for the learning algorithm, as otherwise, the gradients for the computation of S would not exist. Therefore, the gradient descent learning algorithm changes S to reduce the distance between two nodes that require strong information exchange in every iteration. This is called graph topology learning.

Features in G and I are concatenated with the input features to get another feature set O; and finally, O is transformed to the output space X_{n+1} via another dense layer. The activation function for the dense layer at the output is also ReLU.

$$O_h = F_h \oplus G_h \oplus H_h \tag{6.12}$$

$$X_{n+1} = \mathcal{D}_*(O) \tag{6.13}$$

6.1.2 Distance-weighted message passing

The information aggregation process in a GravNet layer can also be separated into a series of message passing layers, which can be stacked into L layers. This increases the depth of the neural network and, therefore, increases its expressive power. In these message passing layers, the same graph, as defined in the GravNet layer, is used.

Beginning with:

$$O_0 = \mathcal{D}_*(X_n) \ , \tag{6.14}$$

and for $i \neq 0$:

$$G_{i_h} = O_{(i-1)_h} - \texttt{mean}(\{d(x,h)O_{(i-1)_x} \; \forall x \in \mathcal{N}(h)\}) \ , \tag{6.15}$$

$$I_{i_h} = O_{(i-1)_h} - \texttt{max}(\{d(x,h)O_{(i-1)_x} \forall x \in \mathcal{N}(h)\}) \ , \tag{6.16}$$

$$O_{i_h} = G_{i_h} \oplus I_{i_h} \oplus O_{(i-1)_h} \ . \tag{6.17}$$

At the last iteration, the O becomes the output of the distance-weighted message layers set:

$$X_{n+1} = I_L \ . \tag{6.18}$$

6.2 Loss function

The object condensation method is extended in this thesis to account for the complexity of the calorimetric data by softening various loss terms. These modifications make the method more robust to stochasticity in the truth definition and when its not clear whether two showers are the same or not. Such problems occur especially more in the presence of pileup where a high number of very low energy particles are presence. The object condensation paradigm embeds all the associated hits into a clustering space (x_h). It must have at least two dimensions to guarantee a monotonous path in the clustering space towards a minimum. Over three dimensions are complicated to visualise. In this thesis, a three dimensional clustering space is used.

For each particle shower, a representative hit, referred to as the condensation point, is learned. The condensation point is determined by the condensation confidence score that is produced for each of the hit (β_h). The condensation confidence score is sigmoid activated and is hence restricted between 0 and 1:

$$\beta_h = \frac{1}{1 + \exp(-\beta_h^{'})} \ . \tag{6.19}$$

Here, $\beta_h^{'}$ is the raw value produced by the neural network for every hit.

All the properties of the shower are aggregated into this condensation point through a payload loss. In the neural network presented in this thesis, the payload loss only consists of an energy correction factor $\psi_h \approx 1$, which is further described below. Finally, a dynamic distance measure φ_h is also introduced, that scales with the distance from other showers and noise hits in the clustering space. This addition allows to keep the cluster coordinate space low dimensional, and therefore interpretable, while introducing another degree of freedom to adopt distances to locally dense environments. The distance is also sigmoid activated and is restricted between 0 and 1. Therefore, in total, the neural network has a 6-dimensional output per hit.

The loss consists of three terms: the potential loss L_V is responsible for creating the clustering space and embedding hits into it, the condensation score loss L_β trains the network to identify the condensation points, and the payload loss L_P creates gradients for the other object properties, in our case the energy correction factor. The relative contribution of the two loss terms is set by a factor s_C, which is taken as $s_C = 1$ in this thesis.

$$L = L_V + s_C(L_\beta + L_P) \tag{6.20}$$

Especially for the hadronic showers, hits that are significantly displaced in position from the shower core are challenging to assign to their incident particle. The mis-assignment of these hits to showers that are closer in space, but initiated by another incident particle, is relatively common and difficult to avoid. Typically, such hits are low energy, so their correct shower association is less critical to estimating the total shower properties. With this in mind, the original object condensation method is adapted to reduce the impact of the mis-association of this class of hits in the network. Mathematically, this also serves to reduce the maximum fluctuations in the gradients. In comparison to Kieseler (2020), the calculation of the clustering charge q_h has also been adapted, and the potential terms are smoothened. The clustering charge is calculated based on β_h with $0 \leq \beta_h \leq 1$. The calculation of q_h is slightly rescaled to avoid strong gradients for $\beta_h \rightarrow 1$ by

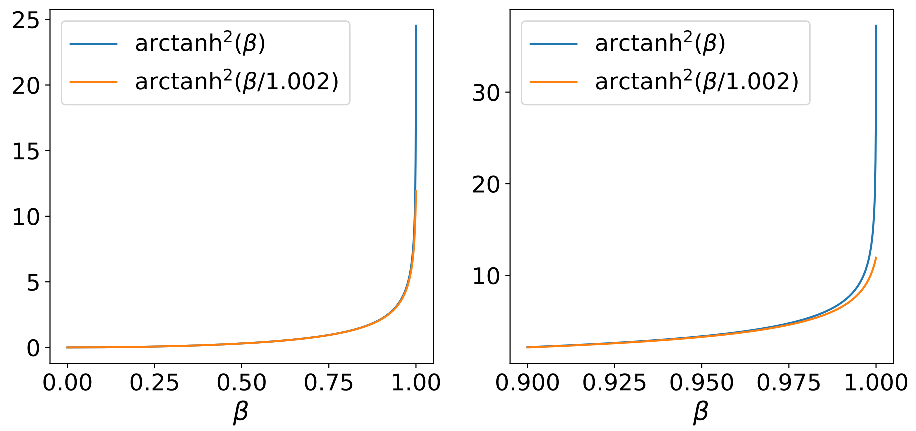


Figure 6.2: How calculation of q_h is affected ($q_{\min} = 0$ is affected with and without a safeguard over full β range (left) and zoomed in (right).

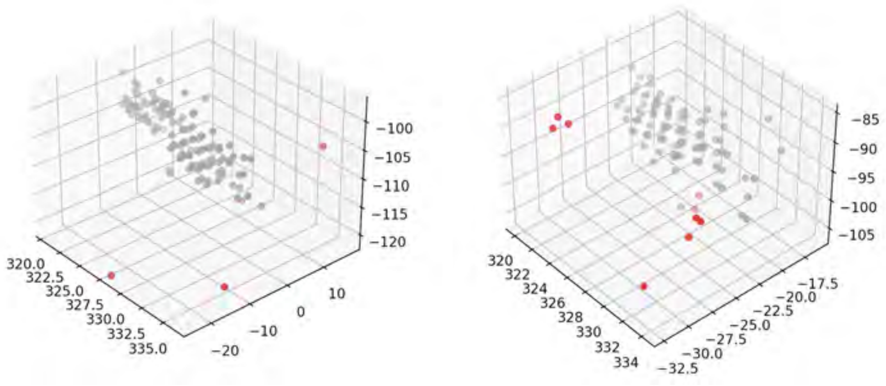


Figure 6.3: Two examples of the concept of spectators. The hits that are marked as spectators are shown in red and the rest of the hits are shown in grey.

adding a safeguard:

$$q_h = \text{arctanh}^2(\beta_h/1.002) + q_{\min}\nu_h \ . \tag{6.21}$$

This is also shown in Fig. 6.2. arctanh^2 approaches ∞ when β approaches 1 without the safeguard.

In addition, a new parameter ν_h that describes a spectator weight is introduced for each hit. Hits that are scattered far away from the shower core receive a smaller weight of $\nu_h = 0.1$, while all other hits receive a weight of $\nu_h = 1$. To define whether a given hit should be considered a spectator, a principal component analysis (PCA) is applied on the truth-assigned energy-weighted hit coordinates of the shower to identify the two principal components which act as the proxies for the shower axes. For this task, the shower axes are defined in two dimensions only, where the dimensionality is reduced by one because of the shower symmetry. The hits belonging to the shower are then projected onto the defined shower axes. Using the projected coordinates, the Mahalanobis distance (Mahalanobis, 1936) is computed for each hit. A hit is considered a spectator if its Mahalanobis distance is larger than 3. This is visually shown in Fig. 6.3. It can be easily observed there are some hits that are very far from the core of the shower and are hence marked as spectators (red). The spectator hits are assigned as such during the dataset preparation stage.

For the attractive and repulsive potential losses, the hit α_t with the highest β score for each truth shower t, also taking into account the spectator weights, plays a special role. It is defined as:

$$\alpha_t = \underset{h \in H_t}{\operatorname{argmax}}(\beta_h \nu_h), \tag{6.22}$$

where H_t is the set of hits belonging to truth shower t. Furthermore, the β-weighted average learned distance scale $\overline{\varphi}_t$ for a truth shower t is calculated as:

$$\overline{\varphi}_t = \frac{\sum_{h \in H_t} \beta_h * \varphi_h}{\sum_{h \in H_t} \beta_h}. \tag{6.23}$$

Here, ϕ_h is produced at the output of the neural network for every hit.

Taking the weighted average over the shower as opposed to considering only the hit α_t has the advantage that it creates a more consistent gradient for φ_h to be learned for every hit. A similar approach is taken for the reference point in the clustering space of the potentials that attracts or repulses other hits. Here, the reference point for each truth shower t is calculated as

$$\overline{x}(t) = \frac{1}{2}\left(x_{\alpha_t} + \frac{\sum_{h\in H_t}(\beta_h q_h x_h)}{\sum_{h\in H_t}\beta_h q_h} \right). \tag{6.24}$$

This represents another modification of the original object condensation loss, which takes x_{α_t} only. The new term in the sum serves to remove noise from the training, while keeping a large impact of the hit α_t, which helps to resolve the degeneracy at the beginning of the training. Based on these ingredients, the attractive potential loss, $\breve{V}_h$, is then re-defined as follows:

$$\breve{V}_t(h) = q_{\alpha_t} w_t \ln\left(e \cdot \left(\frac{\|x_h - \overline{x}(t)\|^2}{2\overline{\varphi}_t^2 + \epsilon} \right) + 1 \right), \tag{6.25}$$

where w_t is the shower weight. For $E_{\text{true}} > 10$, $w_t = 1$. From 10 to 0.5 GeV, it linearly decreases from 1 to 0. ϵ is a small number that is added for numerical stability. The repulsive loss is modified accordingly, as

$$\hat{V}_t(h) = q_{\alpha_t} w_t \cdot \exp\left(-\frac{\|x_h - \overline{x}(t)\|^2}{2\overline{\varphi}_t^2 + \epsilon} \right). \tag{6.26}$$

The full potential loss function takes the form:

$$L_V = \frac{1}{|T|}\sum_{t\in T}\left(\frac{1}{|H_t|}\sum_{h\in H_t} q_h \breve{V}_t(h) + \frac{1}{|H - H_t|}\sum_{h\in(H-H_t)} q_h \hat{V}_t(h) \right). \tag{6.27}$$

Here, $H - H_t$ represents the set difference, i.e., all hits that are not assigned to shower t. The payload loss L_P is also weighed by the object weight w_t to reduce the impact of low energy showers, that is

$$L_P = \sum_{t\in T} \frac{w_t}{\sum_{h\in H_t}\xi(h)} \sum_{h\in H_t} \xi(h) L_E, \tag{6.28}$$

with $\xi(h) = \operatorname{arctanh}^2(\beta_h/1.002)$. The energy loss contribution L_E is calculated as

$$L_E = \log\left(\left(\frac{E_{\text{true},t} - \psi_h E_{\text{dep},t}}{\sqrt{E_{\text{true},t}} + 0.003} \right)^2 + 1 \right), \tag{6.29}$$

where E_{dep} is the total energy collected in the calibrated calorimeter cells associated with the truth shower t.

The beta loss term consists of two parts and is identical to Kieseler (2020),

$$L_\beta = \frac{1}{|T|}\sum_{t \in T}(1 - \beta_{\alpha_t}) + s_B \frac{1}{|H_\circ|}\sum_{h \in H_\circ}\beta_h. \tag{6.30}$$

The first term ensures that at least one hit per truth shower is promoted to a condensation point. The second term suppresses noise. $H_\circ$ represents the set of all noise hits. The scaling factor $s_B = 1$ is chosen to be 1.

6.3 Training

Standard stochastic gradient descent and Adam optimiser (Kingma and Ba, 2014) were used for training the neural network only one event is used per training iteration. A learning rate of 10^-4 was used. The training is continued until the loss function converges to a stable value in expectation. This occurs in $\mathcal{O}(10^4)$ iterations and approximately takes a week on a high-end GPU.

Two models were trained for the results presented in this thesis. First, 60 multi-particle events are overlapped with 200 pileup. Due to computational constraints, it is impractical to train with full 200 pileup. To overcome this, for each pileup event, a point is randomly sampled in $\phi_0 \sim U(0, 2\pi)$ and afterwards $P_{\text{secondary}}$ with impact directions between ϕ_0 and $\phi_0 + 30°$ are selected. This is labelled the phase cut trick. This ensures that highly dense local conditions are present in the training set but reduces the computational complexity for training. The non-pileup particles are left intact. This is also visualised in Fig. 6.4.

The model is first trained for 68 epochs on a toy dataset using $5,000$ training samples. The toy dataset is described in Chapter 7, in Section 7.1.2. The same model is then copied and fine tuned on a multi-particle dataset without pileup for 3 epochs. This dataset is composed of HGCAL events and contains $33,000$ training samples (further described in Chapter 7, in

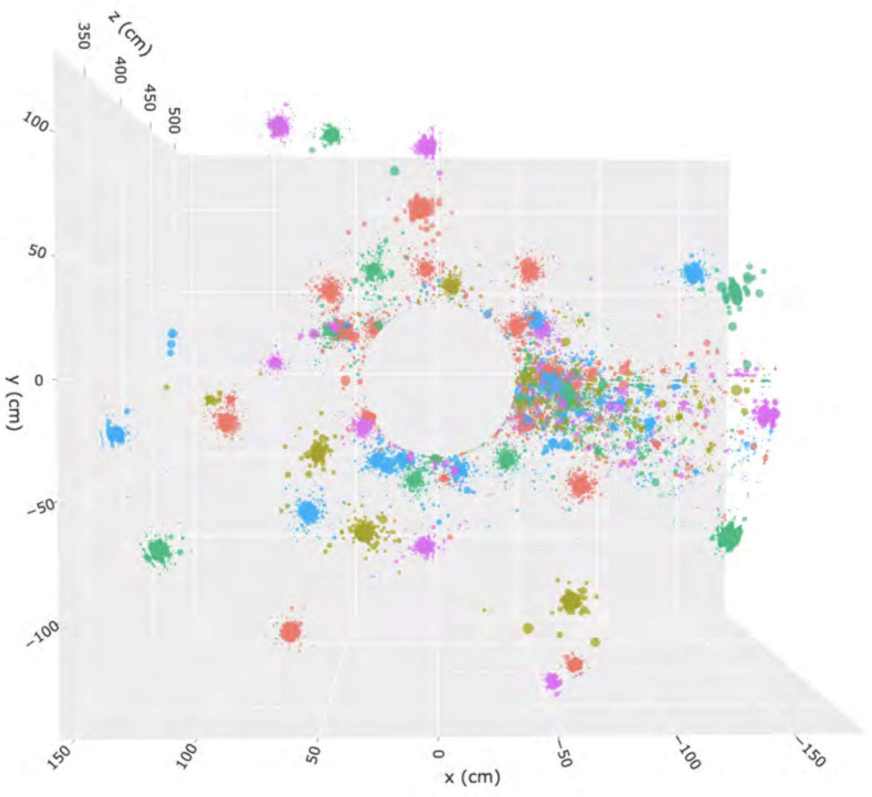


Figure 6.4: An example of a training event. Pileup particles are only present in a randomly selected ϕ region (on the right side in this figure).

Section 7.2). During the fine-tuning process, ν_h is set to 1 $\forall h \in H$; and therefore, the spectators have not been used for the multi-particle model. Since no pileup is present here, the training can be easily done over the full event without the need of the phase cut technique.

6.4 Inference

Once the training has finished, the predicted clusters are evaluated, using an inference clustering algorithm. The inference algorithm from Kieseler (2020) has also been extended to reflect the introduction of the local distance scale ϕ_h. The algorithm is outlined in Procedure 1 and is applied to the learned clustering space. The algorithm starts with the hit with the highest β-score β_α and assigns all hits within a certain radius $t_d \cdot \phi_\alpha$ to it, with $t_d = 1.0$ (Step 7). These hits are removed for the next iteration (Step 9 and Step 10). This procedure is repeated until the highest β-score is lower than the threshold t_β, set to 0.3 for the toy calorimeter and 0.5 for the HGCAL. The remaining unassigned hits are considered as noise.

To choose the t_d and t_β values, a parameter scan was performed. At every value pair, the efficiency and resolution curves were manually studied to find the best combination.[3] A lower value of t_β results in production of more showers and hence, more fakes. A lower value of t_d results also results in more showers as it would potentially over split the showers due to the lower radius.

To determine the energy of the reconstructed cluster, the energy is summed over all the hits assigned to the cluster collected around hit α and multiplied by the learned energy correction factor ψ_α (Step 8).

Procedure 1 Clustering Inference

```
    Input $H, \beta, x, \psi, \varphi, t_d, t_\beta$
    Output $P$
 1: $\mathbf{P} \leftarrow \{\}$
 2: $H_{\text{cand}} \leftarrow \{h \ni \beta_h > t_\beta \forall h \in H\}$
 3: $H_{\text{free}} \leftarrow H$
 4: while $|H_{\text{cand}}| > 0$ do
 5:     $\alpha \leftarrow \text{argmax}_h(\beta_h \forall h \in H_{\text{cand}})$
 6:     $p \leftarrow$ NEW_PARTICLE
 7:     $H_p \leftarrow \{h \in H_{\text{free}}, \|x_h - x_\alpha\| < t_d \varphi_\alpha\}$
 8:     $E_{\text{pred}}(p) \leftarrow \psi_\alpha \sum_{h \in H_p} e_h$
 9:     $H_{\text{free}} \leftarrow H_{\text{free}} - H_p$
10:     $H_{\text{cand}} \leftarrow H_{\text{cand}} - H_p$
11:     $P \leftarrow P \cup \{p\}$
12: end while
```

[3]This is same in spirit – albeit much more complex – to how the classification threshold is chosen by plotting the ROC curve.

6.5 Summary

In this chapter, the reconstruction method that is the main contribution of this thesis is presented. The data produced in complex detectors can be generically represented as a point cloud. Dynamic graph neural networks are then employed which learn the graph structure. These methods are based on k nearest neighbours in a learned latent space. Message passing is then performed on these nearest neighbours.

One such method is called GravNet where the hit features are transformed to independent spaces, the feature space and the coordinate space. The graph is defined in a lower dimensional coordinate space for performance reasons and feature information is carried in a higher dimensional space. Distance based attention is then applied on the coordinate to learn the coordinate space.

A generic clustering loss function – called object condensation – is paired with GravNet. In object condensation all the hits are mapped from the input space to a learned clustering space such that all the hits that belong to one shower are very close to each other and those that belong to different showers are far away. A simple clustering approach can then be used to find the desired particle showers. In object condensation, object properties can also be jointly inferred using a payload loss function. This capability is then employed to regress energy of the reconstructed particles. The adaptions to the original object condensation method – which allowed reconstruction in highly stochastic environment of the HGCAL – are also presented in this chapter.

Chapter 7

Physics Performance

Physics or reconstruction performance refers to the accuracy of the reconstructed events, i.e., both the clustering performance and the accuracy of the reconstructed particles. The accuracy of the reconstructed particles is defined by both identification (ID) accuracy and the energy resolution. Although, ID – which encompasses finding the type of the particle (i.e., whether it was an electron, proton etc) – is beyond the scope of this thesis, the clustering and energy/momentum reconstruction performance is studied in this chapter.

The events produced at the LHC are complex in nature and are defined by a range of effects that affect the reconstruction performance. The performance is a function of energy and also depends on the type of particles, i.e. hadronic particles are harder to reconstruct compared to the electromagnetic particles. Therefore, depending on the event type, the reconstruction performance analysis has to be carefully conducted.

First, reconstruction performance in pileup environments is studied in Section 7.1. A stand-alone calorimeter, with properties sufficiently close to the HGCAL but without the computational complexity of simulating the other detector components of CMS that are of no or little effect, is used for this purpose. It is presented in Section 7.1.1. The reconstruction performance is then studied independently for single particles and for jets in Section 7.1.3 and Section 7.1.4, respectively.

The HGCAL data does not contain one probe particle per event, unlike the toy calorimeter. This fact makes it harder to study reconstruction performance as now it has to be studied over all the particles in an event. This is done in a number of ways. First, in Section 7.2.1, the performance analysis is done on all the particles by one-to-one truth-predicted matching. It is then extended to a case where multiple matches are allowed in Section 7.2.2 to study over splitting and over merging. Like in the toy calorimeter, jets are also used to study the reconstruction performance in the HGCAL, in Section 7.2.3. Here, the performance of the neural network is compared to the classical reconstruction algorithm for the HGCAL. Finally, results are extrapolated to pileup conditions in Section 7.2.4 based on local energy density.

In all the cases, the reconstruction performance is presented as a function of p_T. While the neural network is regressing only the particles' energy, for the p_T computation, energy-weighted mean hit positions are used to estimate the particles' direction. For consistency, the same methodology has also been employed to compute the truth-level p_T.

Finally, Section 7.3 summarises the Chapter.

7.1 Reconstruction performance in the toy calorimeter

7.1.1 Calorimeter description

For studying the reconstruction performance in high pileup environments expected at the HL-LHC, a simplified model of the CMS HGCAL[1] in GEANT4 (Agostinelli et al., 2003) is presented in this thesis. While this toy calorimeter does not simulate electronics and has simpler geometry compared to the HGCAL, the average occupancy is approximately the same as in the HGCAL. Therefore, reconstruction studies done on the toy calorime-

[1]Use of CMSSW requires years in approval before the results can be published. This is not compatible with the timeline of this thesis.

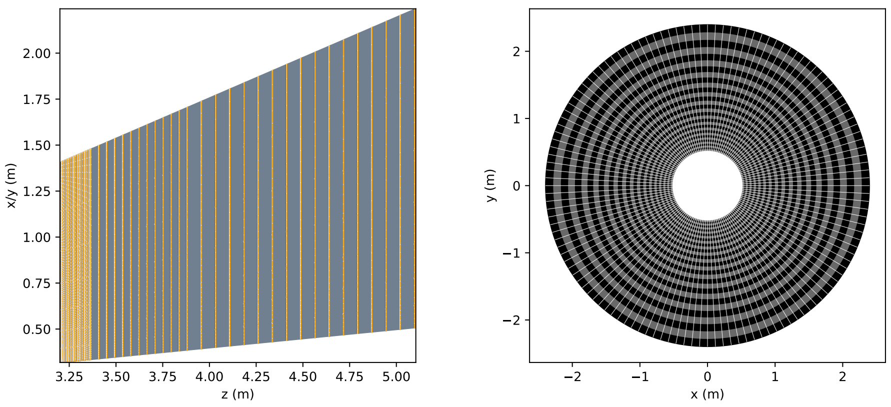


Figure 7.1: On the left, the longitudinal cross section of the detector is shown where different colours correspond to different materials - copper (orange), stainless steel and lead (gray), air (white) and silicon (black). Right: Transverse cross section of the last active layer of the detector, showing how the sensors are formed by slicing across r and ϕ.

ter are expected to fully generalise to the HGCAL.

A cross-sectional view of the detector is shown in Fig. 7.1. The detector is positioned 3.2 m away from the interaction point and covers the endcap region ($1.5 < |\eta| < 3$). Longitudinally, the detector is divided into three sections:

1. Electromagnetic part consisting of 14 layers
2. Hadronic Section 1 consisting of 12 layers
3. Hadronic Section 2 consisting of 16 layers

The thickness of the simulated silicon sensors is always 200 μm and there are two sensor segments in each of the electromagnetic layers. Copper, lead and steel are used as the absorber materials. The hadronic part is divided into two sections, and the first section has relatively thinner absorbers. There is always one active silicon segment in each of the hadronic layers. The materials and their thickness can be observed in Fig. 7.2. The electromagnetic section corresponds to $17 \cdot X_0$ and $1.3 \cdot \lambda_I$ while the hadronic section is $10 \cdot \lambda_I$.

Instead of hexagonal shaped sensors, which will increase the complexity of the simulation, square shaped sensors in (η,ϕ) plane were used in the toy calorimeter. This is achieved by segmenting the silicon layers uniformly first across η and then ϕ. The number of segments across the ϕ axis is always $4\times$ the number of segments across the η axis. This ensures that the sensors are approximately square shaped in the (η,ϕ) plane. The size of a cell decreases with the number of the active layer it is part of. In the first layer, 0.02×0.02 size sensors were used, and it drops to 0.07×0.07 in the last layer. Fig. 7.3 shows the number of segments and number of sensors as a function of z. The high slope on the left of the Fig. 7.3b represents the high number of sensors in the electromagnetic section of the calorimeter. In total, there are approximately 0.8 million sensors.

A calibration is performed on the raw energy deposits by re-scaling the hit energies. For this purpose, only one parameter (g, the global scaling factor) is learned. The thickness of the absorber preceding an active layer is then used to infer the calibration factor of that layer. w_s and $w_a(s)$ represent the width of the sensor s and the absorber preceding it.

$$c_s = 1 + \frac{w_a(s)}{w_s(s)} \ , \tag{7.1}$$

g is evaluated using only photons as the calibration set (T) via the RMS method, as follows:

$$\hat{g} = \underset{g}{\operatorname{argmin}} \sum_{t \in T} (E(t) - \sum_{s \in \mathbb{S}} g c_s \ d_{\text{raw}}(s,t))^2. \tag{7.2}$$

The quality of the calibration can be observed in Fig. 7.4.

Fig. 7.5 visualises examples of single-particle showers in the toy detector. As expected, the photon fully showers out in the electromagnetic section of the calorimeter whereas the π^+ crosses into the hadronic section. Laterally, the hadronic shower is also wider, as noted in Section 3.3.

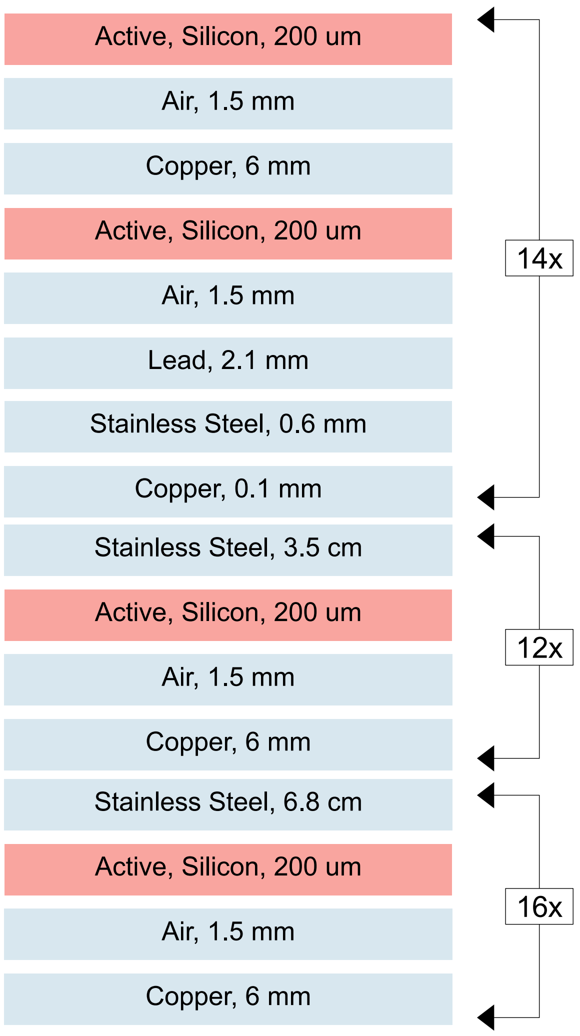
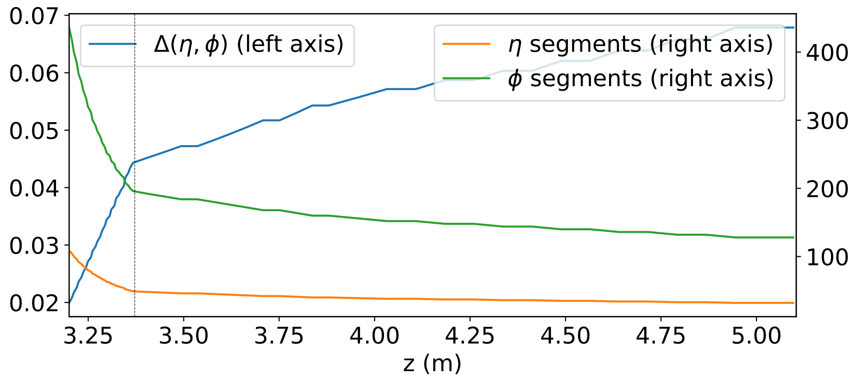


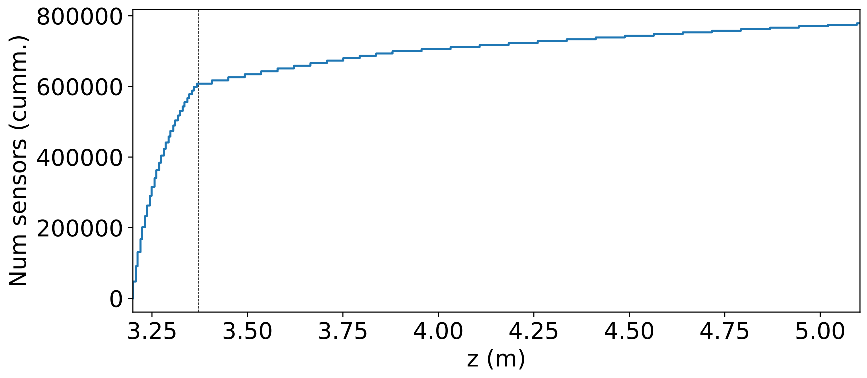
Figure 7.2: Block diagram showing different materials that form the detector. The arrows on the right of the diagram divide into three sections (one electromagnetic and two hadronic).

7.1.2 Dataset

For event generation in the toy calorimeter, first independent simulations are generated, which can either be the cascade resulting from proton-proton interactions or single particles. These are later joined to form events of desired complexity. The pileup collisions are independent random processes, as the probability of particle-particle interactions is close to zero. Therefore,



(a)



(b)

Figure 7.3: Fig 7.3a: $\Delta(\eta, \phi)$ as a function of z, spanning the length of the detector along the longitudinal axis (left axis) and the number of segments needed across η and ϕ (right axis). Fig 7.3b: Cumulative number of sensors as a function of z. The vertical line represents the location where the electromagnetic section ends.

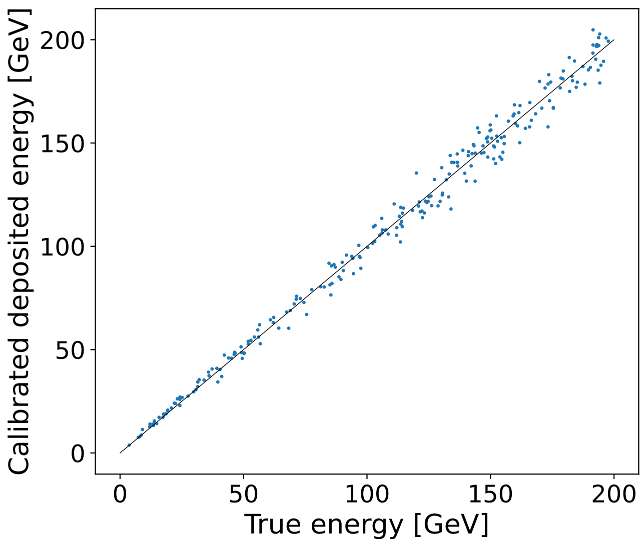


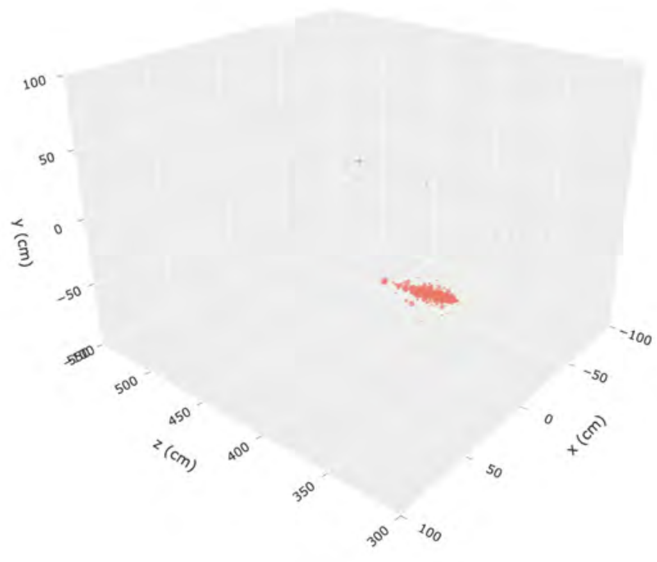
Figure 7.4: A scatter plot showing calibrated deposited energy versus true energy of photons shot between $1.8 < \eta < 2.8$. $E_{\text{dep}} = E_{\text{impact}}$ line is highlighted.

a set of pp collisions can be simulated and stored in a pileup library. Pythia (Sjöstrand et al., 2015), the most common software used for the production of primary particles, is employed for the production of primary particles in proton-proton interactions.

The location of a pp interaction at the LHC is called the vertex. The z coordinates of the vertex are randomly distributed ($z \sim N(0, 5$ cm) in a realistic collider; however, the interaction point is considerably far away that it can be taken as $(0, 0, 0)$ for the purposes of calorimetric reconstruction. A magnetic field is also present in the CMS detector that bends the particles along the ϕ axis. This does not impact the distribution of particles in the calorimeters and it can also be safely omitted without loss of complexity.

Four types of simulations are generated:

1. **Type A**: Single-particle simulations for training. The particles are randomly chosen as e^-, γ, π^+, π^0 or τ, with momentum coordinates uniformly distributed in $E \in [0.1, 200]$ GeV, $\eta \in [1.4, 3.1]$, and $\phi \in$



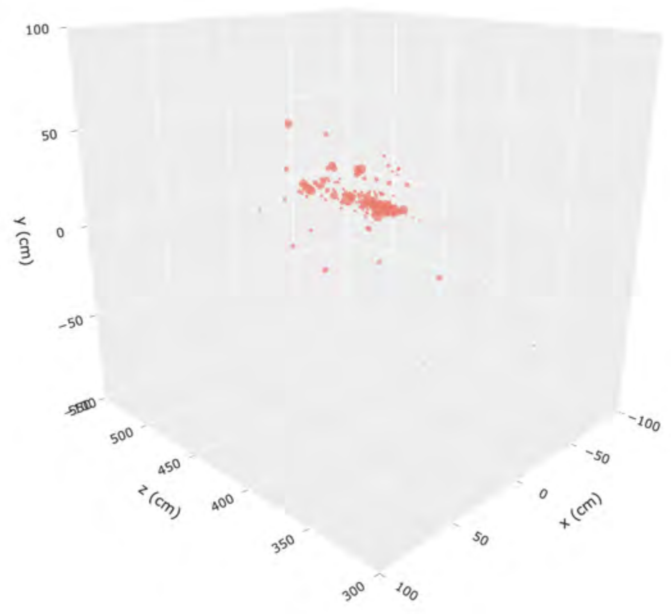


Figure 7.5: Top: 100 GeV γ shower. Bottom: 110 GeV π^+ shower.

$[0, 2\pi]$. $3.1 \cdot 10^5$ simulations are generated that are all used for training.

2. **Type B**: Stable single-particle simulations generated 1 mm away from the detector, as if they were coming from the interaction point in a straight line for testing performance of the models. The particles are randomly chosen as e^-, γ, or π^+, with momentum coordinates uniformly distributed in $E \in [0.1, 200]$ GeV, $\eta \in [1.6, 2.9]$, and $\phi \in [0, 2\pi]$. $80,000$ simulations are generated which are all used for testing.

3. **Minimum Bias**: Minimum bias proton-proton interactions generated at a center-of-mass energy of $\sqrt{s} = 13$ TeV. $3.1 \cdot 10^5$ and $2 \cdot 10^5$ simulations are generated for training and testing, respectively. These are stored as a pileup library.

4. **$\mathbf{t\bar{t}}$**: Synthetic $q\bar{q} \rightarrow t\bar{t}$ events generated at $\sqrt{s} = 13$ TeV using PYTHIA8. This sample is used to study the jet reconstruction accuracy (see Section 3.5). $40,000$ simulations are generated, which are all used for testing.

Procedure 2 Event Generation

```
    Input S
    Output $H, T$
 1: $T \leftarrow \bigcup_{\mathbb{S}\in\mathbf{S}} P_{\text{secondary}}(\mathbf{S})$
 2: $G_{\text{close}} \leftarrow (T, \{f_{\text{close}}(p_1, p_2)\forall(p_1, p_2) \in T \times T\})^2$
 3: $T \leftarrow$ merge (connected_components ($G_{\text{close}}$))
 4: for each $i \in \mathbb{S}$ do
 5:     $t(i) \leftarrow$ undef
 6:     $\bar{e} \leftarrow$ sample($N(0, 5 \times 10^{-6})$)
 7:     $e_{\max} \leftarrow \max_T(d_{\text{raw}}(i, p)\forall p \in T)$
 8:     if $e_{\max} > \bar{e}$ then
 9:         $t(i) \leftarrow \text{argmax}_T(d_{\text{raw}}(i, p)\forall p \in T)$
10:     end if
11:     $\bar{e} \leftarrow \bar{e} + \sum_{p\in T} d_{\text{raw}}(i, p)$
12:     $e_i \leftarrow \hat{g}c_i\overline{e_i}$
13: end for
14: $H \leftarrow \{i\forall i \in \mathbb{S} \ni \overline{e}_i/A_i > \rho\}$
15: $T \leftarrow \{p\forall p \in T \ni |h\forall h \in H \ni t(h) = p| > 0\}$
```

[2]Defines a graph as (nodes, edges)

To produce a 200 pileup event, for example, 200 minimum bias simulations (**S**) are joined together with Procedure 2. The raw deposits on the sensors from different simulations are added together. In order to emulate realistic detector conditions, detector noise is added to the deposited energy at generator-level, as specified in Step 6. The detector noise model consists of a generation of spurious energy measurements in the detector sensors, distributed according to a Gaussian probability density function centred at 0 and with a variance of $5 \cdot 10^{-6}$. All the sensors with uncalibrated deposited energy per sensor area ($\overline{e}/A$) greater than ρ are considered as the reconstructed hits (rechits) in the event (Step 14). The value of the constant is taken as $\rho = 1.3 \cdot 10^{-7}$ GeV/mm^2, which corresponds to an uncalibrated energy ranging from $5.5 \cdot 10^{-3}$ MeV to 3 MeV between the smallest and the largest sensor.

The $f_{\text{close}}(p_1, p_2)$ in Step 2 evaluates if two particles are too close. This is done based on the sensor size in η and ϕ space (w_η and w_ϕ), allowing more merging to occur in low η region where the sensors are thicker. Two showers are defined as too close if three conditions are satisfied: their $\Delta\eta$ is less than $1.5w_\eta$, $\Delta\phi$ is less than $1.5w_\phi$, and if the difference in their showering angles is less than 0.09, an angle taken from GEANT4 when particles start to shower.

As truth is defined on hit-level, i.e. every hit is assigned a truth shower, only the clusters have at least one hit assigned to them are taken as the reconstruction target (T). An example of 40 pileup event is shown in Fig. 7.6.

In total, six datasets are created:

1. Training set: 5,000 events, where each event is created from 200 Minimum Bias simulations and 60 Type B simulations.

2. Single-particle testing set: 20,000 events, where each event is created from Type B simulation only.

3. PU40+1 testing set: 6,800 events, where each event is created from 40 Minimum Bias simulations and 1 Type B simulation.

4. PU200+1 testing set: 6,800 events, where each event is created from 200 Minimum Bias simulations and 1 Type B simulation.

5. PU40+$t\bar{t}$ testing set: 6,800 events, where each event is created from 40 Minimum Bias simulations and 1 $t\bar{t}$ simulation.

6. PU200+$t\bar{t}$ testing set: 6,800 events, where each event is created from 200 Minimum Bias simulations and 1 $q\bar{q} \rightarrow t\bar{t}$ simulation.

The training set with 5,000 events would, for instance, require 10^6 minimum bias simulations, there are only 3.1×10^5. Therefore, these events are randomly sampled from the simulations set without replacement. This strategy ensures a minimum overlap of pileup between consecutive events and is used for generating all training and testing datasets.

In Fig. 7.7, various properties of the dataset in 200 pileup are shown. The number of hits can be as high as $200,000$. This is approximately the same as what is expected in 200 pileup in the HGCAL. There can be upwards of 3000 showers in an event. The distribution of the number of hits per shower is slightly, but not significantly, different from what is found in the multi-particle dataset in the HGCAL (Section 7.2). The difference arises from domination of low-energy pileup particles. Fig. 7.8 shows the properties of showers in the endcaps. Most of the showers have low energy, evident from the logarithmic y axis. The distribution is uniform on the η spectrum, signifying presence of a large number of particles close to the beam pipe.

7.1.3 Single particle reconstruction

Pileup events are dominated by low energy particles; and at higher energy levels, exponentially lower number of particles are present. This is shown in Fig. 7.8. It should be noted that while the number of low energy particles is on an exponentially higher scale, they are much less important from a physics point of view when compared to the high-energy particles. To compensate for the complex distribution, a controlled dataset was created where a probe particle, sampled from a uniform energy distribution, is added to the pileup. This is discussed in Section 7.1.2. While the neural network

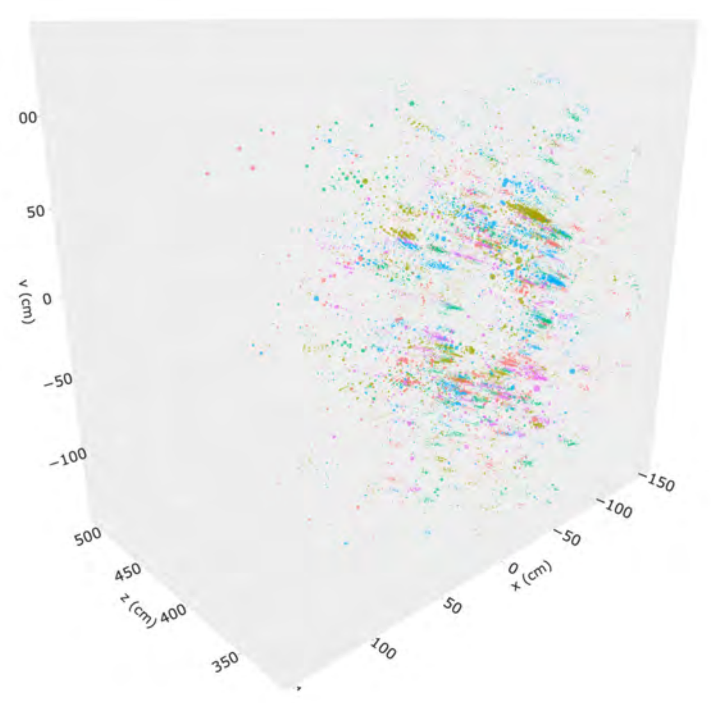


Figure 7.6: An example of a 40 PU event where different colours correspond to different true particles.

reconstructs all the particles, the reconstruction performance of only the probe particle is studied.

A hit-based matching procedure is first used to match the probe true particle to one of the predicted showers. The matching procedure is based on the metric called Jaccard Index or intersection over union (IOU) (Jaccard, 1901), commonly used in computer vision. The IOU is defined as the number of pixels in the intersection of two objects and divided by the number of pixels in the union. This definition assumes that all the pixels are of equal importance. This is not true for hits in, for instance, a calorimeter, as higher energy deposits carry greater importance. Therefore, this metric is extended to energy-weighted intersection-over-union (EIOU) (Qasim, Long,

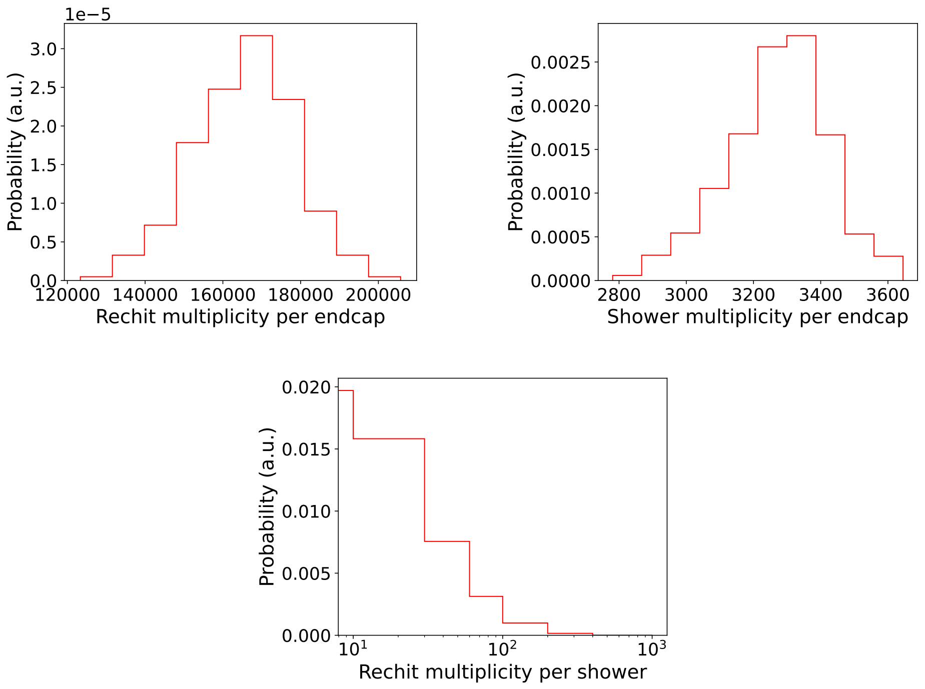


Figure 7.7: All figures are based on 200 PU events in the toy detector. Top left: distribution of number of rechits in an endcap. Top right: distribution of number of showers in an endcap. Bottom: Distribution of number of rechits in a shower.

et al., 2021), defined as:

$$\mathrm{EIOU}(t,p) = \frac{\sum_{h\in(H_t\cap H_p)} e_h}{\sum_{h\in(H_t\cup H_p)} e_h} \ . \tag{7.3}$$

An EIOU value of 1 implies a perfect match, whereas a value of 0 refers to no match at all, typically resulting from trying to match completely different true and predicted showers. H_t represents all the hits associated with the truth shower t and H_p represents all the hits associated with a predicted shower p.

The predicted shower that has the highest EIOU overlap with the probe particle is taken as the matched predicted shower ($\hat{p}$):

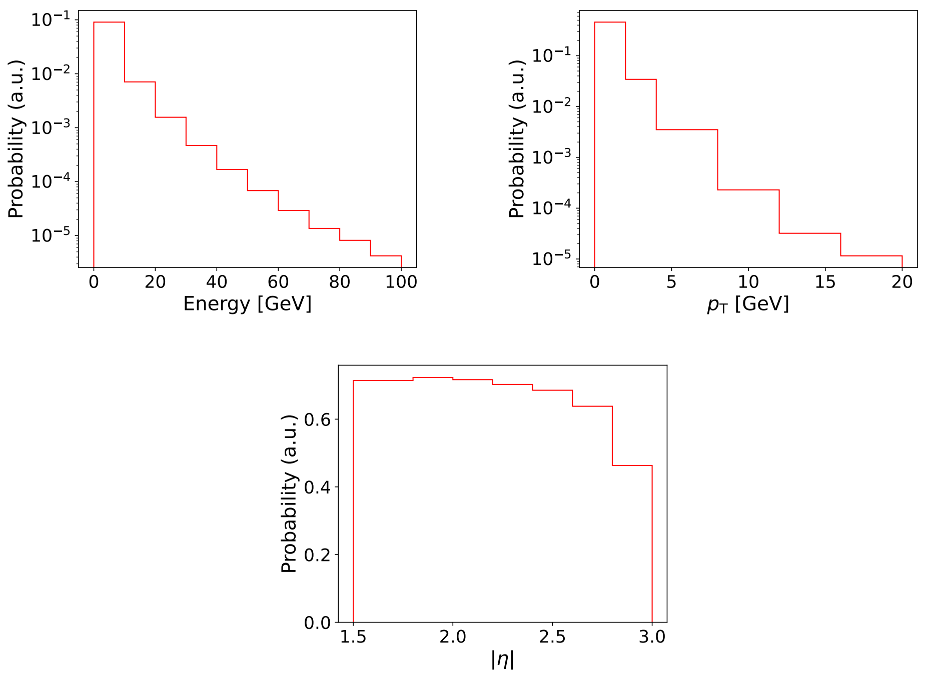


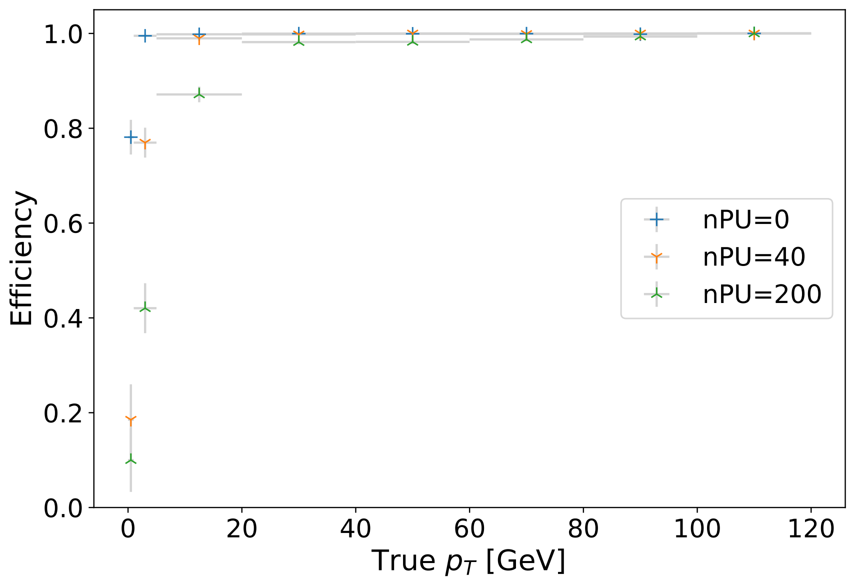
Figure 7.8: Properties of minimum bias showers in the endcap calorimeters. Top left: Distribution of energy of the particles. Top right: Distribution of p_T of the particles. Bottom: Distribution of η of the particles. These numbers are extracted from 200 PU events.

$$\hat{p} = \underset{p \in P}{\mathrm{argmax}}(\mathrm{EIOU}(\hat{t}, p)). \tag{7.4}$$

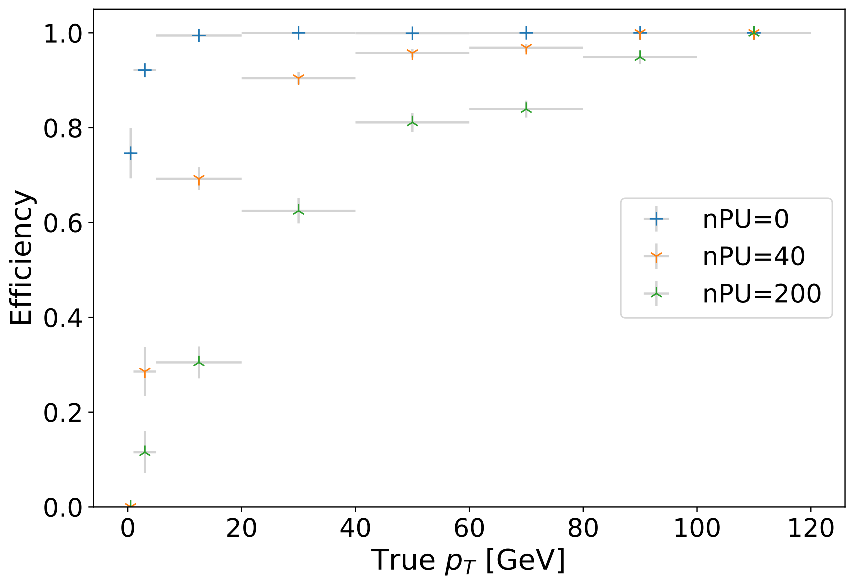
A threshold of 0.5 is applied and hence particles which have less overlap than this are considered not well reconstructed. Efficiency is computed based on this.

Fig. 7.16 shows an example where a true electromagnetic shower was matched to a corresponding reconstructed predicted shower in 200 pileup. Similarly, in Fig. 7.17, a hadron is demonstrated.

The reconstruction efficiency for electromagnetic particles in different pileup conditions is shown in Fig. 7.9a. In 0 pileup, the efficiency is 100%, except if $p_T < 1$ GeV where it is approximately 80%. The lower efficiency is because the low energy particles are indistinguishable from the electronic



(a)



(b)

Figure 7.9: Reconstruction efficiency of single particles in different pileup conditions. Fig. 7.9a: efficiency as a function of the true p_{T} for electromagnetic particles (photons and electrons). Fig. 7.9b: efficiency as a function of the true p_{T} for hadronic particles (charged pions)

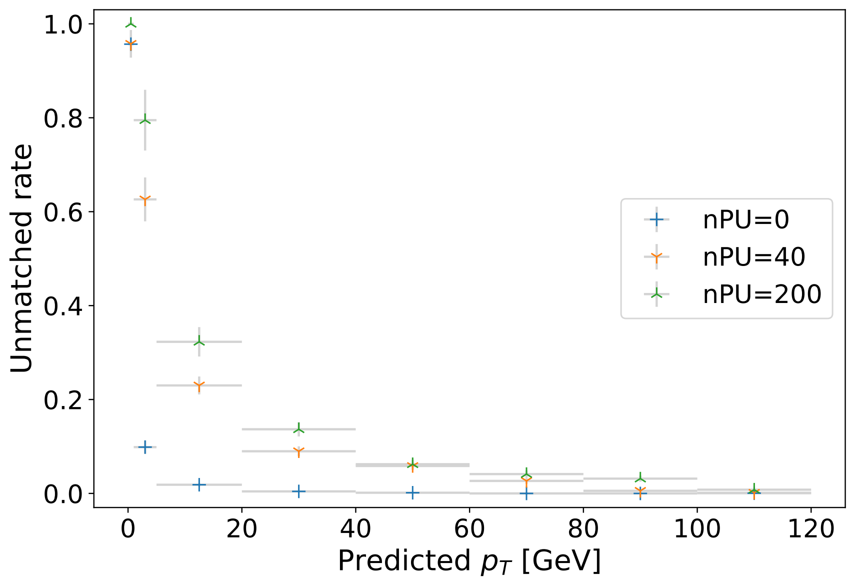


Figure 7.10: Unmatched rate as a function of predicted p_T.

noise and are considered as such by the neural network. In 40 and 200 pileup, the efficiency significantly drops at $p_T < 1$ GeV. This differs from 0 pileup because now, the reconstruction algorithm is likely unable to separate low energy showers from nearby high energy ones instead of considering them as noise. In $1 < p_T[\text{GeV}] < 20$, the efficiency in 40 PU is similar to that in 0 PU, whereas it drops slightly more to $\sim 90\%$ in 200 PU. At higher energy ranges, the difference is only minute in both 40 and 200 pileup.

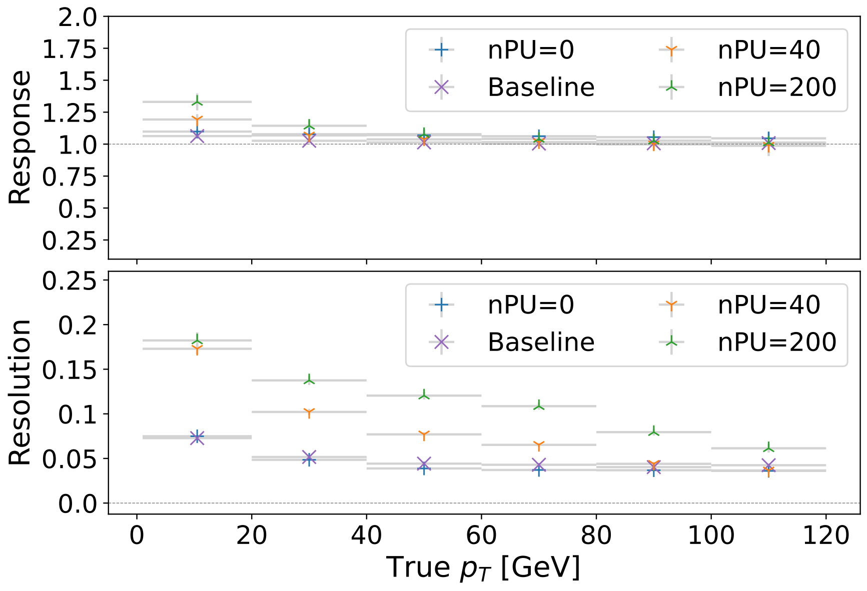
The reconstruction efficiency for hadronic particles in different pileup conditions is shown in Fig. 7.9b. In 0 pileup, the reconstruction efficiency is only slightly lower than for the electromagnetic particles. In $1 < p_T[\text{GeV}] < 20$, the efficiency is not 100% anymore. The efficiency sharply drops when pileup is increased, resulting in $\sim 70\%$ in $1-20$ GeV in 40 pileup and $\sim 60\%$ in $1-20$ GeV range. This is because of the inherent nature of the hadronic particles: they tend to create showers that are split from each other. Energy-Weighted Intersection Over Minimum (EIOM), defined below, can be used to study the over-splitting effect.

$$\mathrm{EIOM}(t,p) = \frac{\sum_{h\in(H_t\cap H_p)} e_h}{\min(\sum_{h\in H_t} e_h, \sum_{h\in H_p} e_h)} \tag{7.5}$$

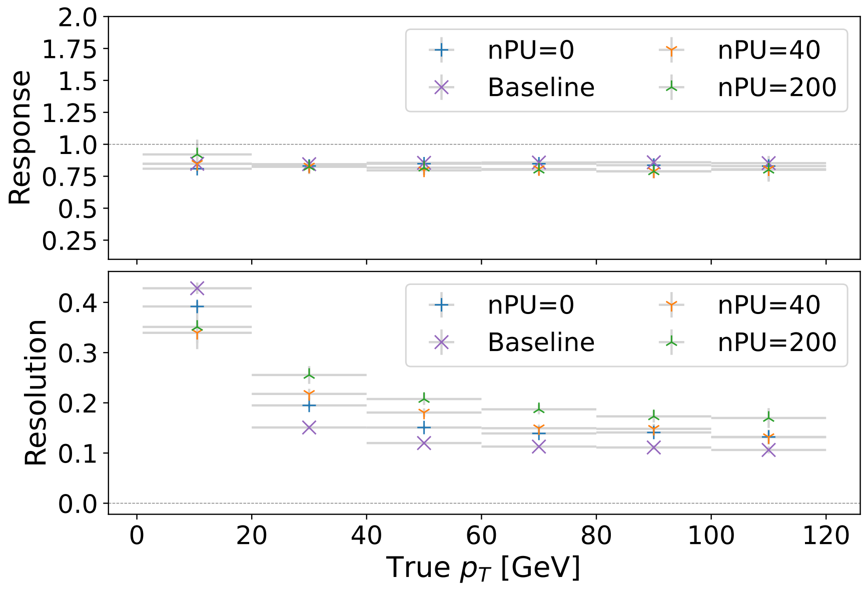
Unmatched showers can be defined as all the predicted clusters with EIOM > 0.9 with the truth-level probe particle but with EIOU less than 0.5, and these are shown in Fig. 7.10. The unmatched rate decreases steeply with the predicted p_T. This indicates that low p_T clusters are split off from higher-p_T showers, while most of the energy is reconstructed properly. By adding tracking information and employing a suitable particle flow algorithm, these splits could be re-merged, increasing the efficiency. In addition to over splitting, the neural network can also create fake showers from the noise hits only. However, the fake rate was observed to be close to zero for p_T above 1 GeV. Additionally, only 0.5% of the total noise energy on average is assigned to a predicted cluster.

The quality of the reconstructed showers can be studied by comparing the truth p_T with the predicted p_T. Response for a single particle is defined as ${p_T}_\mathrm{pred}/{p_T}_\mathrm{true}$. It is averaged over all the events to compute the mean response $\mu\left({p_T}_\mathrm{pred}/{p_T}_\mathrm{true}\right)$. The response can be corrected a posteriori, but it serves as an important metric to understand an algorithm's behaviour. The sum of calibrated energy deposits that the particle leave on the sensors is taken as the baseline. It is therefore truth assisted and sets an approximate upper limit for the resolution.

Response as a function of p_T is shown in Fig. 7.11a and Fig. 7.11b for electromagnetic and hadronic particles, respectively. Due to the nature of hadronic showers and because the cell energies are calibrated on electromagnetic showers, the baseline response for charged pions is below one while it is compatible with one for electromagnetic showers, in particular at high energies. The reconstructed p_T response provided by the network only differs mildly from the baseline response. The difference to unity response increases by about a factor of two for charged pions if the energy correction factor is not applied (not shown), indicating that the network is capable of



(a)



(b)

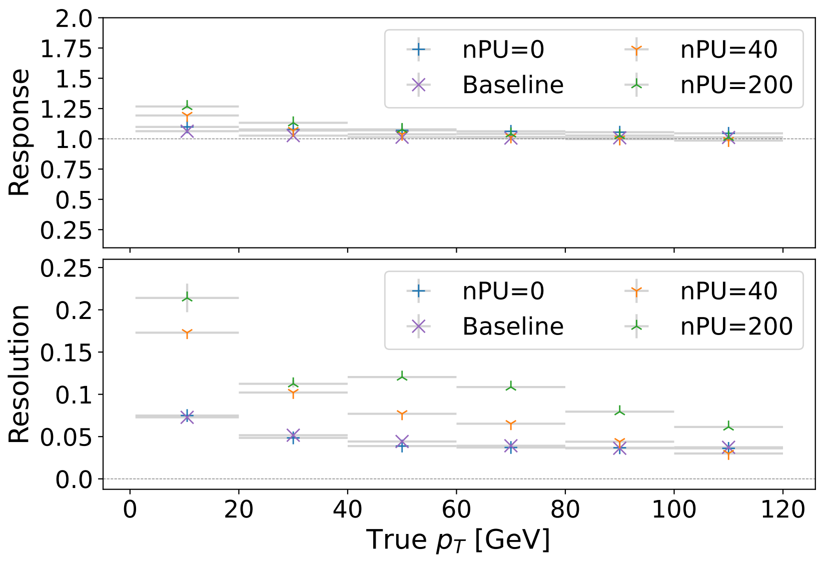
Figure 7.11: The response and the resolution of reconstructed single-particles in different pileup conditions as a function of true p_{T}. Fig. 7.11a shows the electromagnetic particles and Fig. 7.11b, the hadronic particles.

distinguishing different shower types.

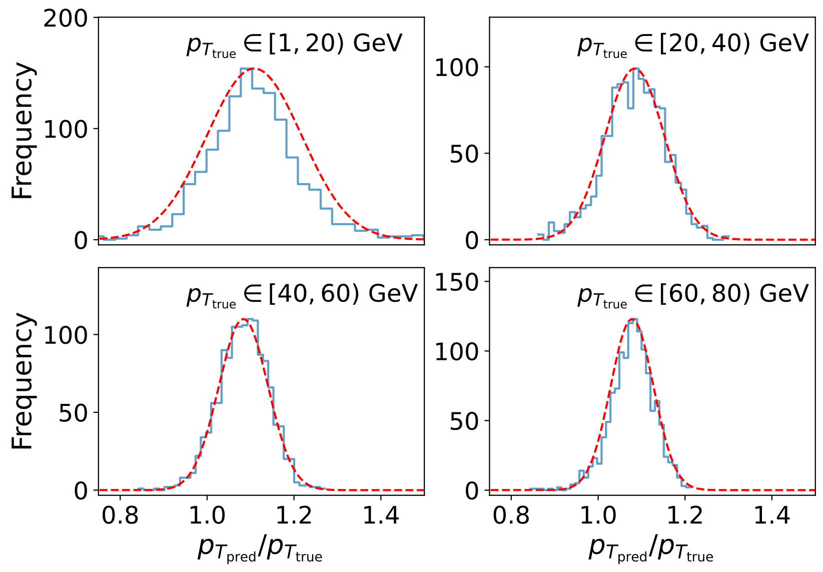
The resolution is also defined according to p_T and is mean-corrected as $\sigma\left(p_{T\text{pred}}/p_{T\text{true}}\right)/\mu\left(p_{T\text{pred}}/p_{T\text{true}}\right)$. It is also shown in Fig. 7.11a and Fig. 7.11b as a function of p_T for electromagnetic and hadronic particles, respectively. As expected, the resolution improves with an increase in the true p_T and degrades with an increase in pileup. Even the reconstructed hadronic shower resolution is close to the baseline reconstruction and converges to $\sim$ 15% above 60 GeV, even in high pileup. The reconstructed energy resolution in 0 pileup for electromagnetic showers is almost indistinguishable from the baseline and, therefore, close to the detector limitations. In 40 pileup, the electromagnetic resolution deviates from the baseline only at low p_T but approximates the baseline at high p_T; and in 200 pileup, it deviates slightly more.

The resolution and mean response can also be computed by fitting a Gaussian function on the response distribution. This method is more robust to outliers and is more commonly used in particle physics. In every p_T bin, a histogram is computed and a Gaussian function is fitted on the histogram values by the χ^2 method. The number of bins is heuristically chosen as $\sqrt{N}$, where N is the total number of particles in the corresponding p_T bin. The optimal fit parameters (μ, σ and A) can be discovered using the Levenberg–Marquardt algorithm (Levenberg, 1944). The initial estimates of μ, σ and A are $\mu\left(p_{T\text{pred}}/p_{T\text{true}}\right)$ and $\sigma\left(p_{T\text{pred}}/p_{T\text{true}}\right)$ and $\max(\{g(b)\ \forall b \in B\})$ for the corresponding p_T bin. After the fit is complete, the centre (μ) and the width (σ) of the fitted Gaussian function are taken as the response and resolution, respectively. The resolution is then mean-corrected, i.e. σ/μ.

Computed through the Gaussian fit method, Fig. 7.12 shows the response and the resolution for the electromagnetic particles, and Fig. 7.13 shows the same for the hadronic particles. The quality of the fit can be observed in Fig. 7.12b and Fig. 7.13b for electromagnetic and hadronic particles, respectively, where the fitted Gaussian functions are overlaid on the corresponding histograms. The values are comparable to those computed without fitting the Gaussian function. They also converge to 5% and 15% at high p_T range for the electromagnetic and hadronic particles, respectively.

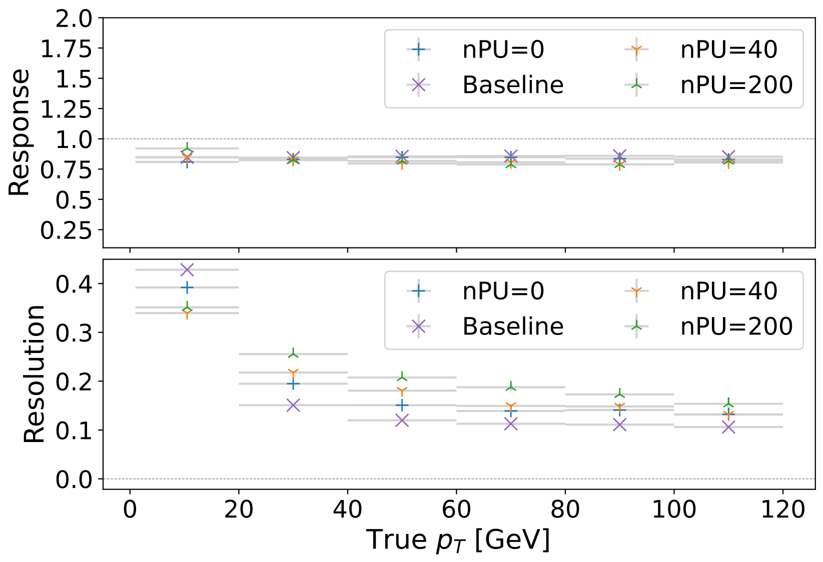


(a)

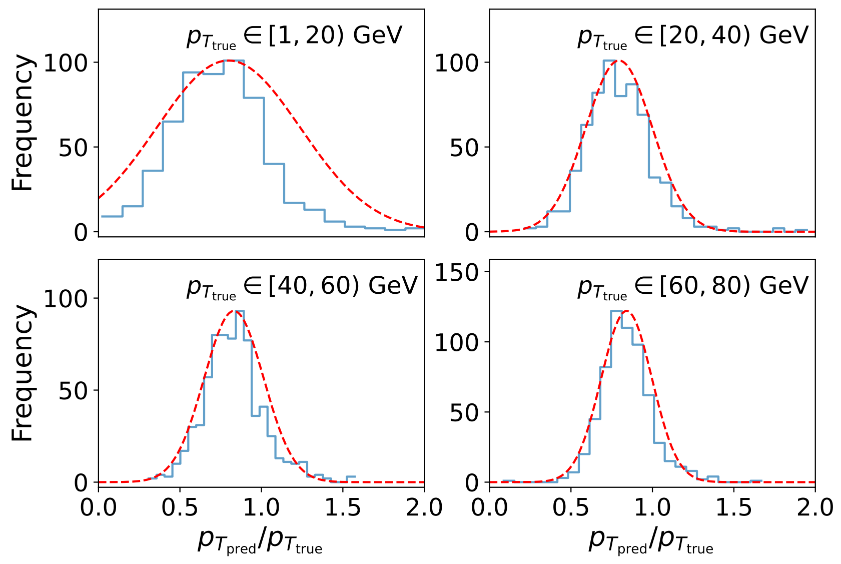


(b)

Figure 7.12: Fig. 7.12a: The response and resolution of electromagnetic particles computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T\text{pred}}/p_{T\text{true}}$ distribution in individual p_T bins. Fig. 7.12b: The distribution of p_T response in different p_T ranges corresponding to the first four bins in Fig. 7.12a in 0 pileup environment.



(a)



(b)

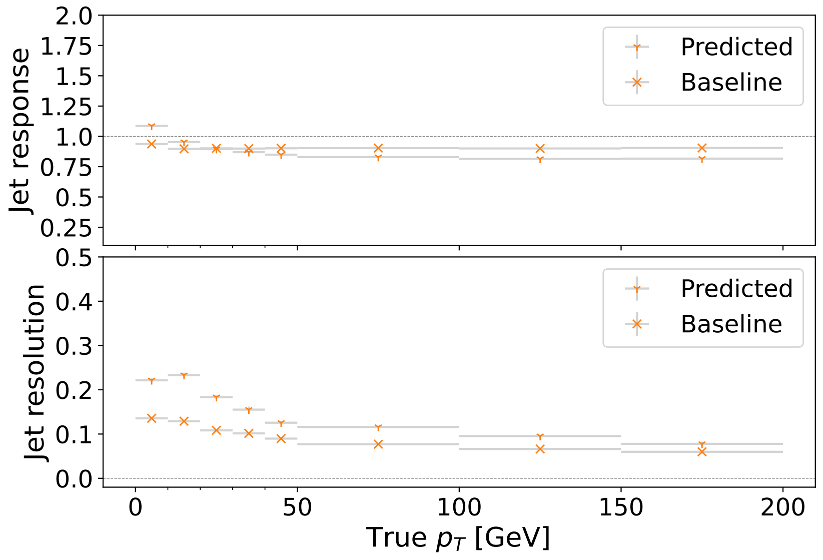
Figure 7.13: Fig. 7.13a: The response and resolution of hadronic particles computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T\text{pred}}/p_{T\text{true}}$ distribution in individual p_T bins. Fig. 7.13b: The distribution of p_T response in different p_T ranges corresponding to the first four bins in Fig. 7.13a in 0 pileup environment.

At low p_T ranges, the values are slightly different, a result of asymmetry in the response distribution.

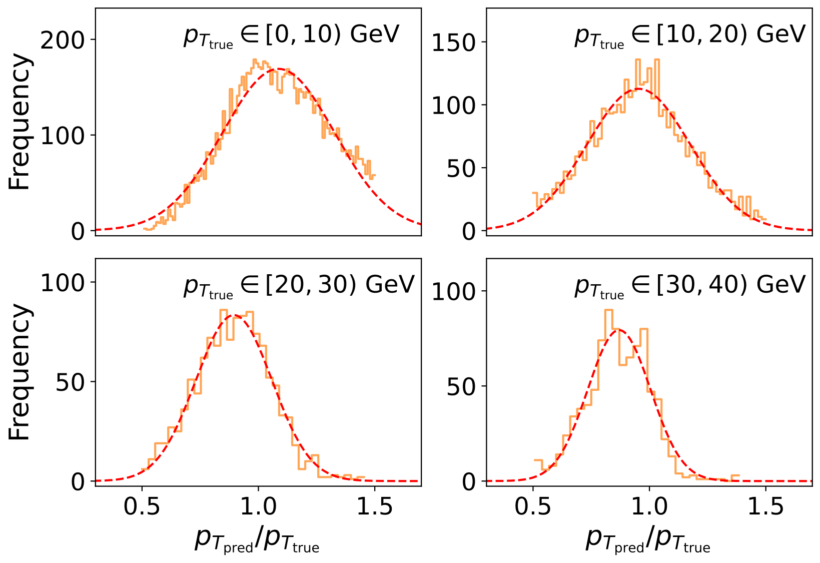
7.1.4 Jet reconstruction performance

As modern jet clustering algorithms are infrared and collinear safe, as observed in Section 3.5, jets also offer a way to gauge the performance of calorimeter clustering algorithms without strong dependencies of subtleties in the definition of the single-particle truth. Moreover, over splitting and over merging due to the reconstruction process have a less impact on the cumulative jet quantities. At the CMS experiment, a pileup removal algorithm is applied before jet clustering to remove contributions from particles not associated with the primary collision. As discussed in Section 3.5, $q\bar{q} \to t\bar{t}$ events are generated and added to 40 or 200 pileup. After reconstruction is performed, a pileup removal algorithm is simulated aided by truth information: all the showers that originate from pileup interactions are removed unless they share more than 10% of their energy-weighted hits with a reconstructed non-pileup shower. All remaining reconstructed showers are considered for clustering the reconstructed jets. To form the truth jets, only truth showers are considered that stem from the non-pileup interaction. The baseline is defined according to the deposited energy. Jets are then clustered using the anti-k_t algorithm (Cacciari et al., 2008) with a distance parameter of $R = 0.4$. Reconstructed and truth jets are matched based on a $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ matching. Among all jets with $|\Delta p_T|/p_{T\text{true}} < 0.5$ and $\Delta R < 0.3$, the best match by minimum ΔR is selected. An event display of jet reconstruction is shown in Fig. 7.18.

Following the procedure performed for individual particle reconstruction, Gaussian functions are fitted to the p_T response distribution in each p_T bin. The mean (μ) and the mean-corrected standard deviation (σ/μ) of the Gaussian function are taken as jet response and resolution. The distributions and the fitted functions are shown in Fig. 7.14a, and the response and resolution are shown in Fig. 7.14a and Fig. 7.15a for 40 and 200 pileup, respectively.

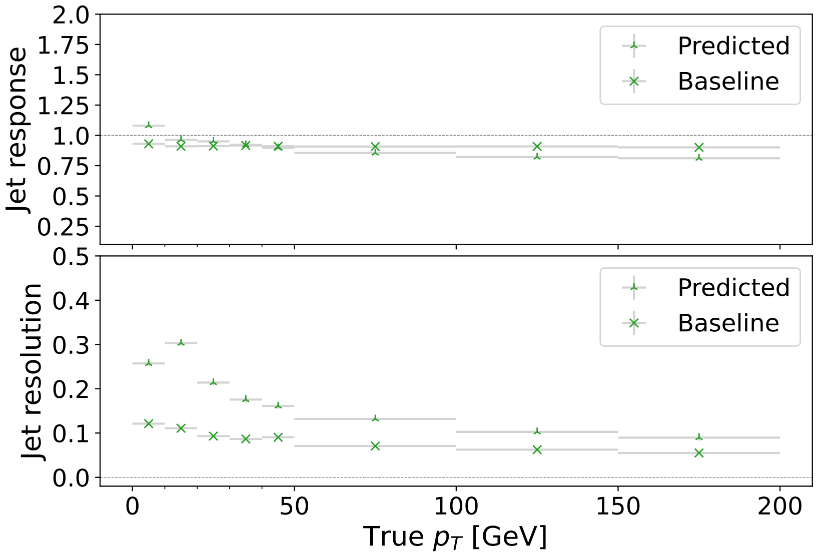


(a)

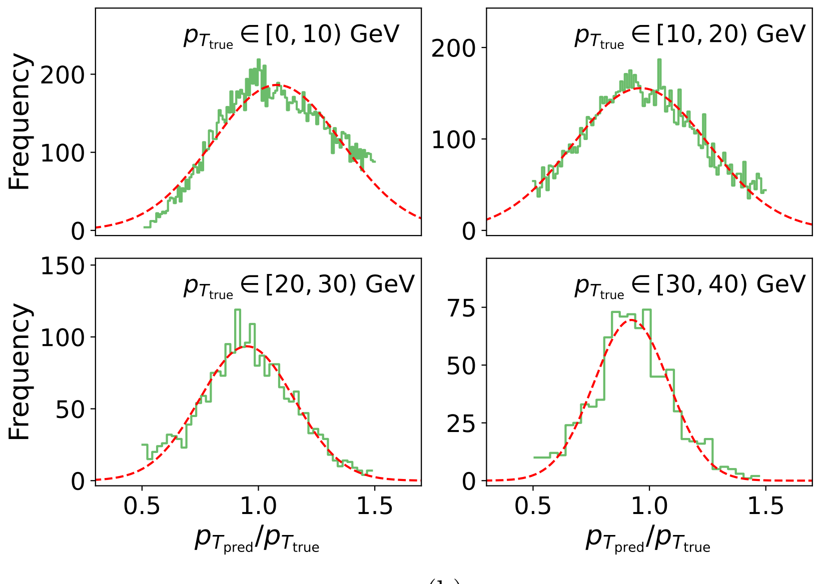


(b)

Figure 7.14: Jet reconstruction performance. Fig. 7.14a: Mean jet response (top) and resolution (bottom) in 40 pileup as functions of the true p_T. Fig. 7.14b: Response distributions and the fitted Gaussian functions in different p_T ranges corresponding to Fig. 7.14a. The response and resolution are computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T\,\mathrm{pred}}/p_{T\,\mathrm{true}}$ distribution in individual p_T bins.

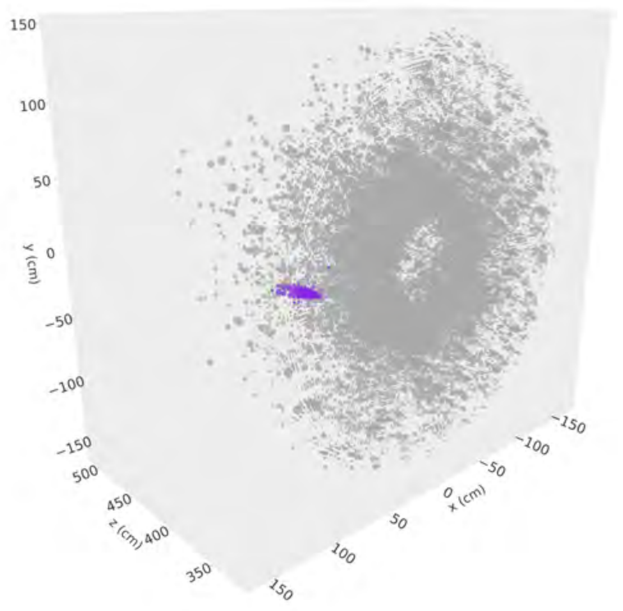


(a)



(b)

Figure 7.15: Fig. 7.15a: Mean jet response (top) and resolution (bottom) in 200 pileup as functions of the true p_T. Fig. 7.15b: Response distributions and the fitted Gaussian functions in different p_T ranges corresponding to Fig. 7.15a. The response and resolution are computed as the mean and mean-corrected standard deviation of the Gaussian fit to the $p_{T\text{pred}}/p_{T\text{true}}$ distribution in individual p_T bins.



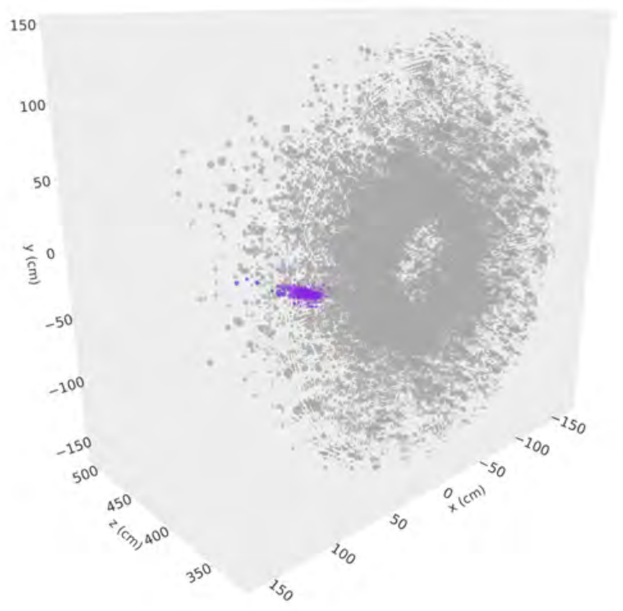
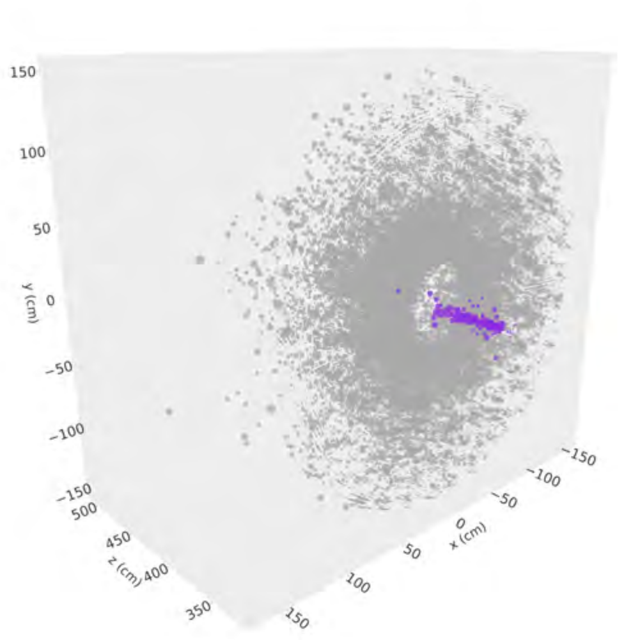


Figure 7.16: An example of a well-reconstructed electromagnetic shower in 200 pileup. The top figure shows the truth hits and the bottom figure shows the reconstructed hits. Grey represents the pileup and coloured points show hits associated with the test shower and the predicted shower matched to it.



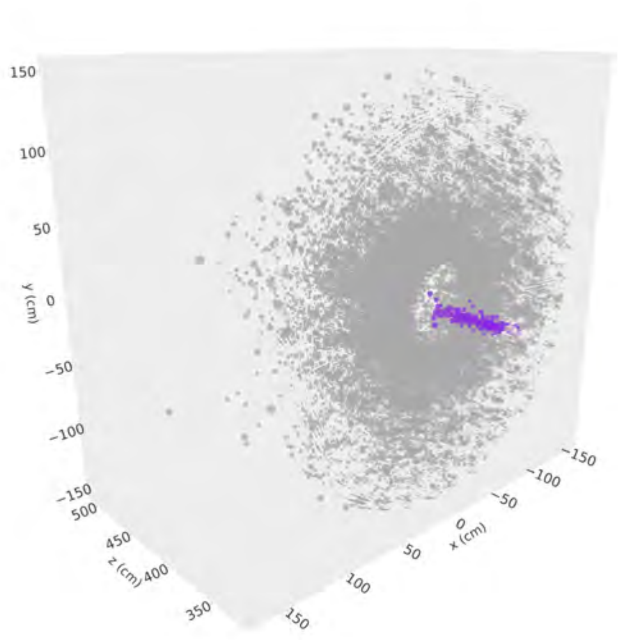
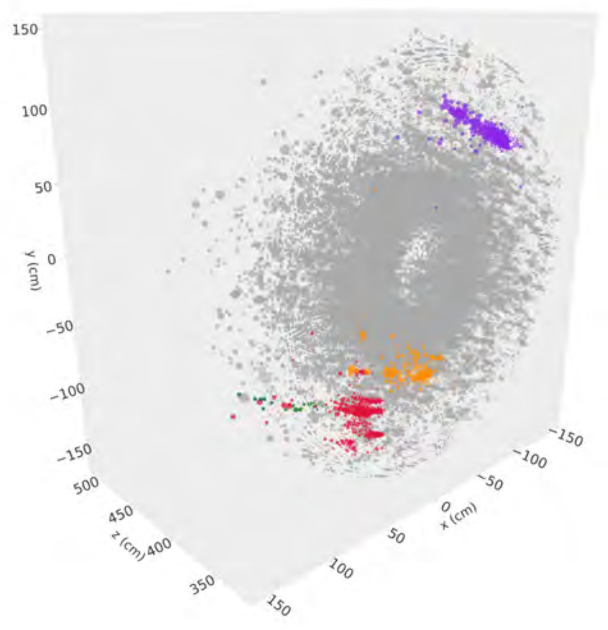


Figure 7.17: An example of a well-reconstructed hadronic shower in 200 pileup. The top figure shows the truth hits and the bottom figure shows the reconstructed hits. Grey represents the pileup and coloured points show hits associated with the test shower and the predicted shower matched to it.



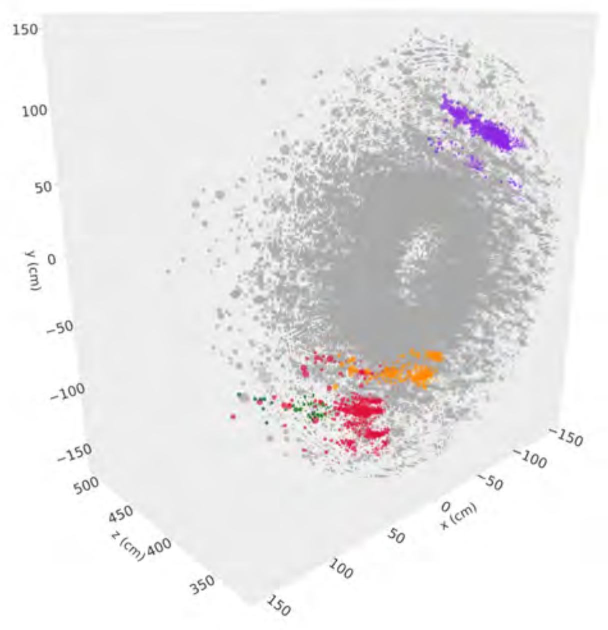


Figure 7.18: An example of jet reconstruction. The top figure shows the truth and the bottom figure shows the predicted jets. Only the matched jets are shown.

The response falling below one is a direct consequence of the single-particle responses shown in Fig. 7.11b. At higher energies, the resolution starts to approximate the baseline and converges to $\sim 10\%$. At lower energies, the presence of pileup degrades the performance slightly, however much less than in the case of single particles. As jets are less affected by truth matching and splitting effects, the assumption that the single-particle performance depends on the matching procedure and the splitting of showers is therefore verified. As shown in Fig. 7.15a, the performance degrades slightly in 200 pileup when compared to 40 pileup. The better resolution in the first p_T bin with the respect to the second bin is an artifact of the biased Gaussian fit due to the asymmetry of the response distribution, as shown in the top-left subplots of Fig. 7.14b and Fig. 7.15b.

7.1.5 Generalisation

The neural network was trained on events with hundreds of true showers. However, it offers an excellent generalisation performance on a variety of problems. There are no fakes observed in 0 pileup and p_T reconstruction performance does not suffer. It also works for jet reconstruction, where the energy of showers can be much higher than the showers found in the training set. This increases confidence in the extrapolation capabilities of the network and training method beyond training conditions in general, and is an excellent result on its own.

7.2 Reconstruction performance in the HGCAL

The HGCAL simulation is implemented as part of of the CMS Software (CMSSW) (CMS collaboration, 2022) and is based on GEANT4 (Agostinelli et al., 2003). The simulation software includes the complex geometry of the HGCAL and also simulates the electronics, including the addition of Gaussian noise. The CMSSW also simulates the other sub-detectors, including the tracker and the magnetic field. The detector response can be very accurately simulated using the simulation software, and the studies done on the simulation, including reconstruction, generalise to the real-world data.

The ground truth is defined at calorimeter level, i.e. all the particles that reach the endcap calorimeters $P_{\text{secondary}}$ are taken for the definition of truth. The ground truth is then defined as a realistic target for the reconstruction algorithm. When two particles (p_1 and p_2) are maximally overlapping, they are merged into one reconstruction target, as their separation will be a hopeless task. Once the truth merging has been done, every hit is assigned to one true particle. It is possible that multiple particles left an energy deposit on a single sensor. However, only the particle that left the most deposit is considered as the true particle associated with that hit.

The dataset that is produced for the studies that follow contains 40 primary particles that were generated at the interaction point with energy uniformly distributed between 5 and 200 GeV. The direction of the particles is also chosen such that the particles have ϕ uniformly distributed in $[0, 2\pi]$ and $|\eta|$ uniformly distributed in $[1.5, 3.0]$. The distributions of various properties of the true particles in this dataset are shown in Fig. 7.20. It can be observed, for instance, that there can be up to 200 true particles in these events. This is because some of the 20 primary particles are unstable and will decay into lower energy particles before they reach the HGCAL. Some of the particles also interact with the tracker material and, therefore, cross the HGCAL boundary as multiple particles. These particles are close to each other in (η, ϕ) space, and will frequently create locally dense conditions that are similar to those found in a high pileup environment. It can also be observed that in an event, there can be up to $40,000$ hits. In 200 pileup, up to $200,000$ hits are expected in the HGCAL. While it may seem that this environment is much simpler than the HGCAL, this is only strictly true when the computational time is considered. From the reconstruction performance point of view, the locally dense conditions can likely result in equally challenging conditions as those found in high pileup environment. A larger presence of high energy showers in this dataset when compared with pileup environments offers an added advantage as these high energy showers are much less common in pileup but are more important for physics studies. There can be as low as 10 hits associated with a particle shower but also as many as 10^3. This exponentially wide range represents of many challenges encountered while performing HGCAL reconstruction.

$33,000$ training samples and $2,000$ testing samples were generated in total for reconstruction studies conducted in this thesis. There are no methodological differences between the training and the test sets.

7.2.1 Multi-particle performance analysis

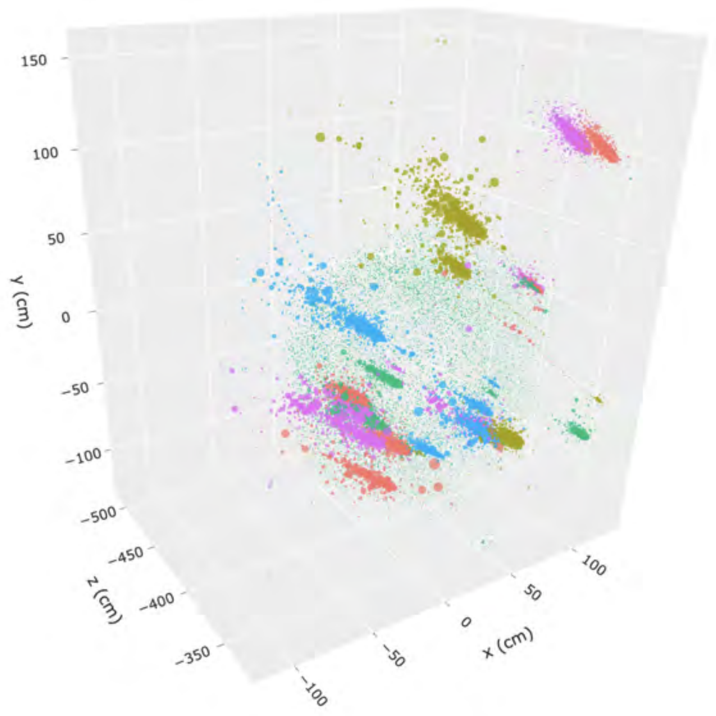
To study the reconstruction performance in a multi-particle environment, first, all the true particles in an event need to be associated with corresponding particles predicted by the neural network. A hit-based matching method is also employed here (Qasim, Long, et al., 2021). The matching is done as follows:

$$M = \text{argmax}(\sum EIOU(p,t)) \forall (p,t) \in \mathbb{C}(\mathbb{P}(\{P \times T\})) \tag{7.6}$$

P is the set of particles that were predicted by the neural network and T is the set of true particles. $P \times T$ takes the Cartesian products of two sets. $\mathbb{P}$ refers to the power set and $\mathbb{C}$ is a filter that defines two constraints: a) one true particle can only be matched to one predicted particle and vice versa, and b) each matching cluster should have a minimum EIOU of 0.5 (Equation 7.3) corresponding to a 50% hit overlap. M will therefore contain a set of truth-prediction matches. This defines a bipartite graph matching problem and can be solved in polynomial time by the Hungarian algorithm (Kuhn, 1955). The cost matrix for the Hungarian algorithm can be constructed in $\mathcal{O}(N)$, with N as the number of hits.

Fig. 7.24 visualises an excellently reconstructed event where predicted and true clusters are colour matched.

Once the particles are matched, the rate of true particles that haven't been matched is taken as efficiency and it is shown in Fig. 7.21. As expected, it increases with an increase in p_T. It reaches over 80% for elec-



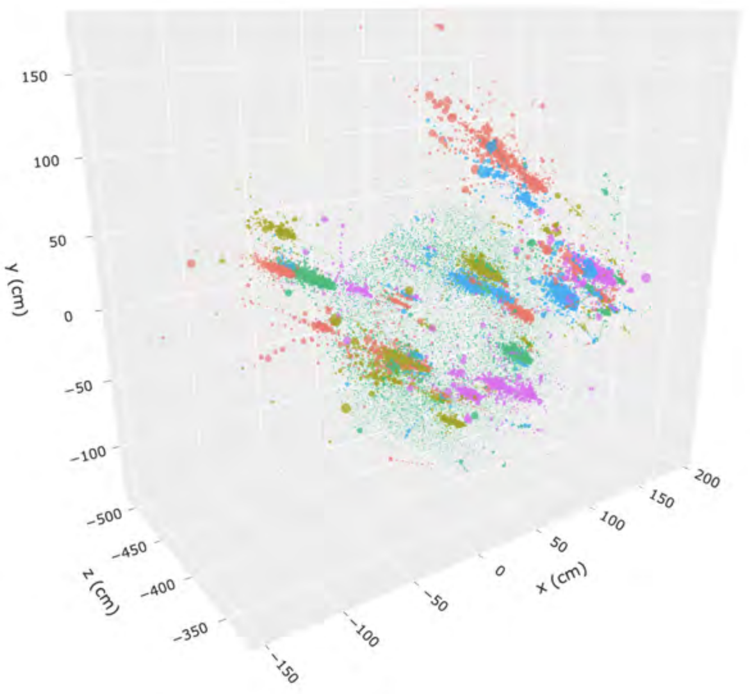


Figure 7.19: Two examples of HGCAL events resulting from 20 primary particles at the interaction point. Different colours correspond to different truth level particles.

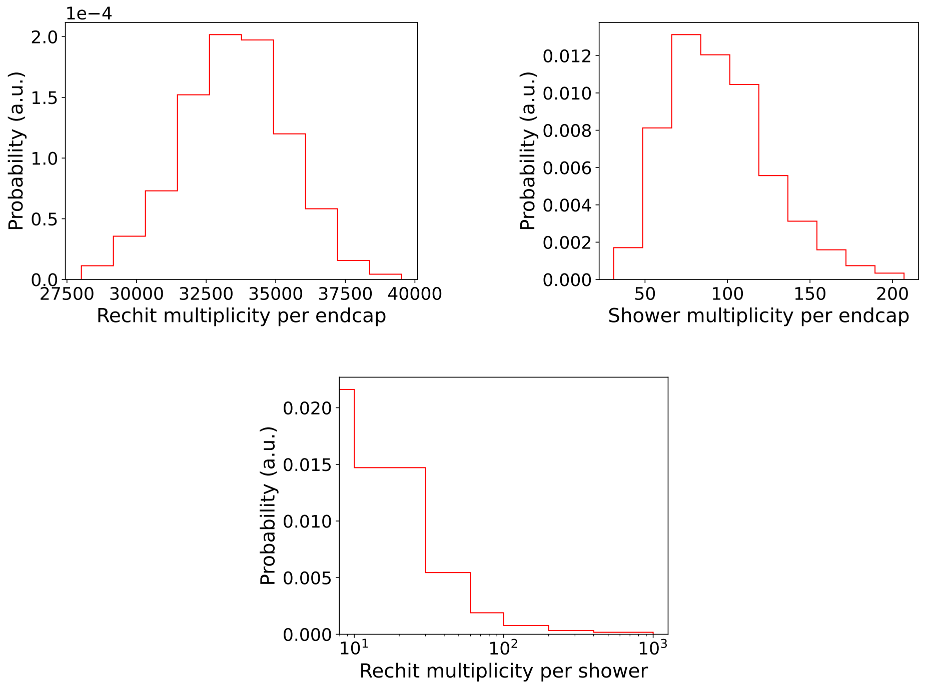


Figure 7.20: HGCAL dataset distributions. Top left: distribution of number of rechits in an endcap. Top right: distribution of number of showers in an endcap. Bottom: Distribution of number of rechits in a shower.

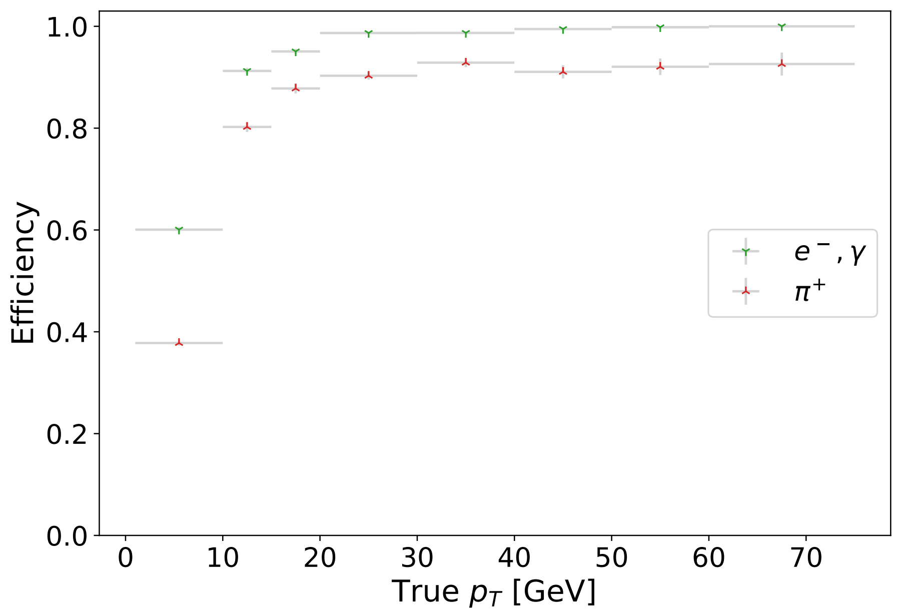


Figure 7.21: Efficiency as a function of the true p_T. Electromagnetic particles and hadrons are separated.

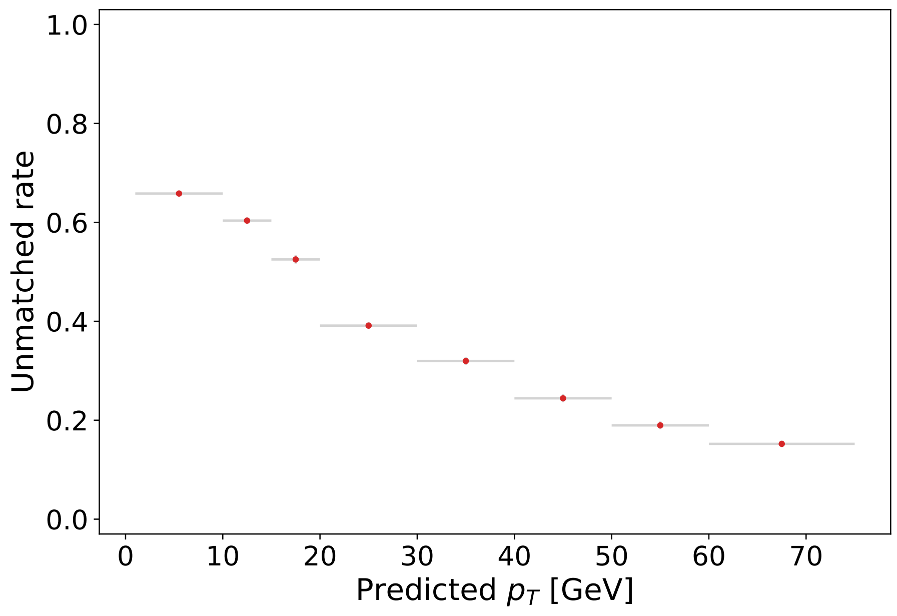
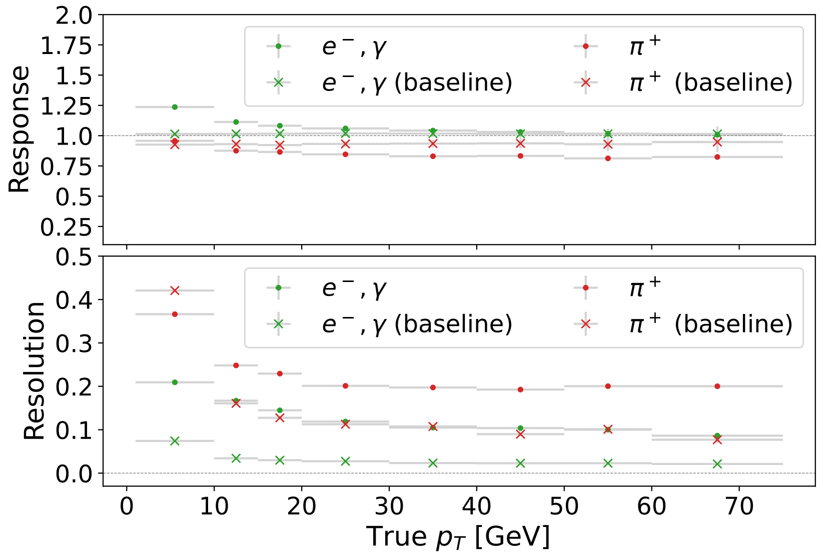
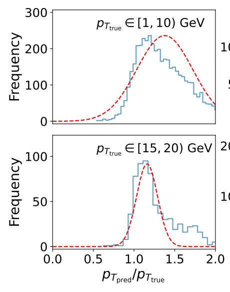


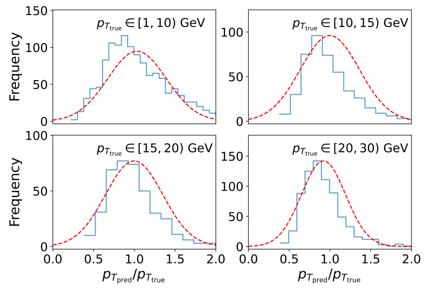
Figure 7.22: Unmatched rate as a function of the predicted p_T.



(a)

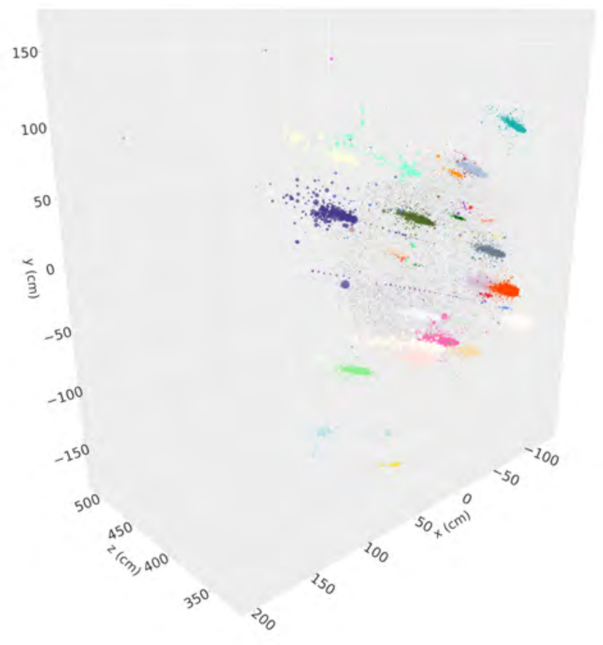


(b)



(c)

Figure 7.23: Fig. 7.23a shows resolution as a function of the true p_T for the CMS HGCAL data. The resolution is computed as mean corrected standard deviation and not by fitting a Gaussian function as the data is not Gaussian, evident in Fig. 7.23b and Fig. 7.23c, where Gaussian functions are fitted on the response distributions for the electromagnetic and hadronic particles, respectively.



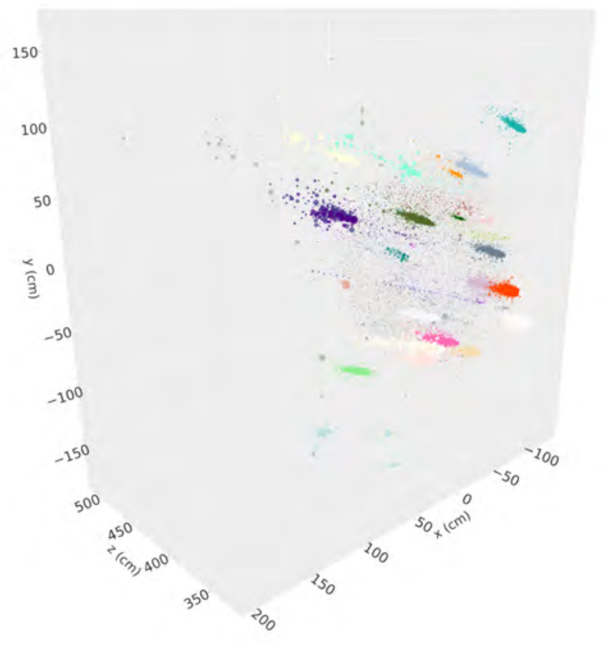


Figure 7.24: An example of event reconstruction in the CMS HGCAL. The top figure shows true showers and the bottom figure shows the predicted showers. The colours are the same if a predicted and a true shower is matched together.

tromagnetic particles at just over 10 GeV. Hadronic efficiency is slightly worse and crosses 80% at 15 GeV. The predicted particles that haven't been matched to corresponding true particles can be studied by observing their unmatched rate, shown in Fig. 7.22. This is different from how it is done in Section 7.1.3 for the toy calorimeter. Like the efficiency, the unmatched rate also improves with an increase in p_T.

The p_T reconstruction performance of the true particles that were matched to predicted particles is shown in Fig. 7.23. Electromagnetic and hadronic particles are separately studied. Unlike in the toy calorimeter, here, the response distribution is not Gaussian, as shown by Fig. 7.23b and Fig. 7.23c, where the fitted Gaussian functions do not approximate the data well. Therefore, the mean response and the mean response corrected standard deviation is used to approximate the resolution. The mean p_T response in the HGCAL follows a trend similar to that in the toy calorimeter (Fig. 7.11). The electromagnetic response is slightly over unity and hadronic response, below. However, the response deviates less from unity compared to the toy calorimeter. This is because the calibration for the HGCAL is more sophisticated and uses both electromagnetic and hadronic particles. The resolution can also be observed, and as expected, is better for the electromagnetic particles. When compared to the baseline, the electromagnetic resolution is only slightly off and falls below 10% at $p_T > 50$ GeV. The hadronic resolution is approximately 20% at $p_T > 50$ GeV.

7.2.2 Multi-matching

Over splitting and over merging can often occur because of ambiguities in the truth definition and isn't necessarily a weakness of the reconstruction algorithm. Furthermore, physics studies are often done on jets which are much less sensitive to these effects. Over splitting and over merging can be studied by a matching algorithm that allows matching of multiple predicted particles to a true particle or multiple true particles to a predicted particle. Here, either one predicted particle gets matched to multiple predicted particles or multiple predicted particles get matched to one true particle. It

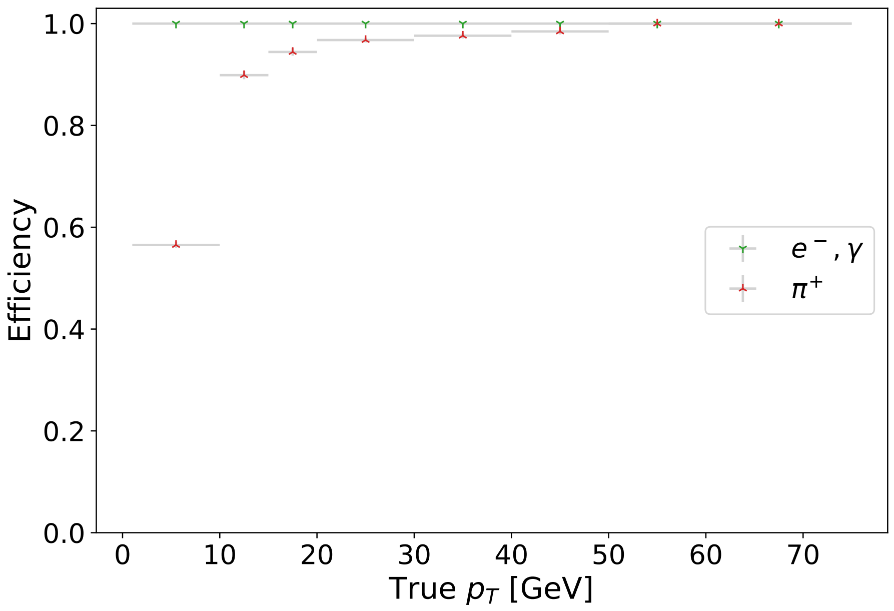


Figure 7.25: Efficiency as a function of predicted p_T with multi-matching allowed in the CMS HGCAL data.

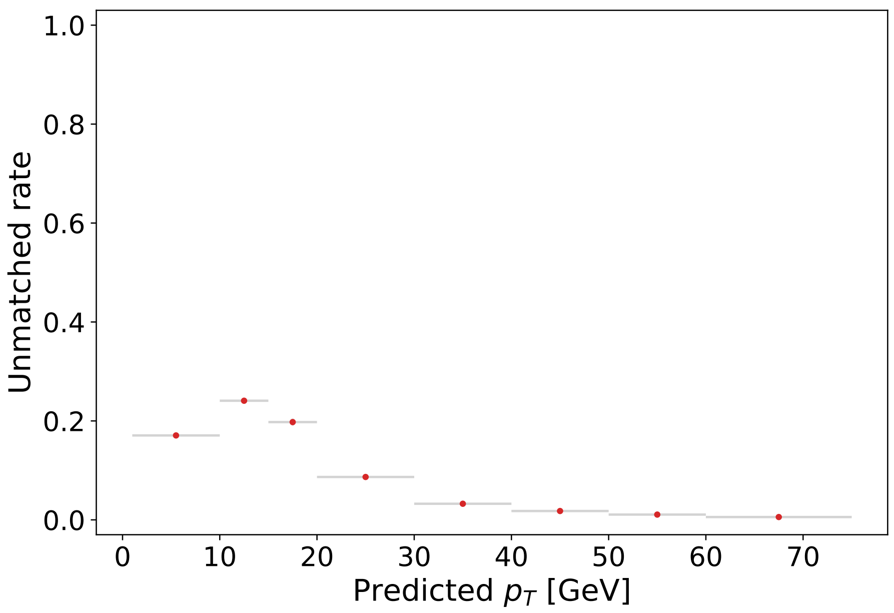


Figure 7.26: Unmatched rate as a function of predicted p_T with multi-matching allowed in the CMS HGCAL data.

cannot occur that multiple predicted particles are matched to multiple true particles, as that would defeat the purpose of clustering. The matching algorithm is iterative and the first step is the same as defined in Equation 7.6 i.e. matching is done on the basis of EIOU scores with a minimum threshold of 0.5. A graph is defined based on the resulting matching (Equation 7.7). The nodes of this graph are all the predicted and the true particles and the edges indicate if a predicted shower and a true shower are matched. In the second and the next iterations, the free nodes are matched to the connected nodes of the opposite types using the EIOM score:

$$G \leftarrow (T \cup P, M) \ , \tag{7.7}$$

$$F_T \leftarrow \{t \in T \ni |\mathcal{N}_G(t)| = 0\} \ , \tag{7.8}$$

$$F_P \leftarrow \{p \in P \ni |\mathcal{N}_G(t)| = 0\} \ , \tag{7.9}$$

$$C_T \leftarrow \{t \in T \ni |\mathcal{N}_G(v)| \neq 0\} \ , \tag{7.10}$$

$$C_P \leftarrow \{p \in P \ni |\mathcal{N}_G(v)| \neq 0\} \ , \tag{7.11}$$

$$\begin{aligned} M \leftarrow M \cup \operatorname{argmax}(\sum EIOM(p,t)) \forall (p,t) \\ \in \mathbb{C}^{'}(\mathbb{P}(\{C_P \times F_T\} \cup \{F_P \times C_T\})) \ . \end{aligned} \tag{7.12}$$

F_T represents the unmatched truth nodes in an iteration and F_P, the unmatched predicted nodes. C_T and C_P represent connected truth nodes and connected predicted nodes. $\mathbb{C}^{'}$ once again defines two constraints: a) one true particle can only be matched to one predicted particle and vice versa, and b) Each matching pair should have a minimum EIOM of 0.9. The EIOM threshold of 0.9 is strict and only allows for splits to be merged. The graph is then updated to add newly the connected nodes. This is repeated for a few iterations until no more nodes are linked. The algorithm converges in less than five iterations. After it has converged, a true particle might have been matched to multiple predicted particles, and a predicted particle might have been matched to multiple true particles. Fig. 7.25 shows the resulting efficiency. All the electromagnetic particles are matched and hadronic efficiency significantly increases. Similarly, the unmatched rate also substantially goes down, as shown in Fig. 7.26.

7.2.3 Jet reconstruction performance

Following the discussion in Section 7.1.4, as jet clustering is useful to study reconstruction performance in a fashion robust to over splitting, over merging and ambiguities in the truth definition, they have also been used to study the reconstruction performance on the HGCAL data. Clustering data on this data does not have strong physics meaning because jets are only defined on specific type of interactions. However, a significant number of high p_T particles are present in the forward region to allow studying of jet reconstruction performance.

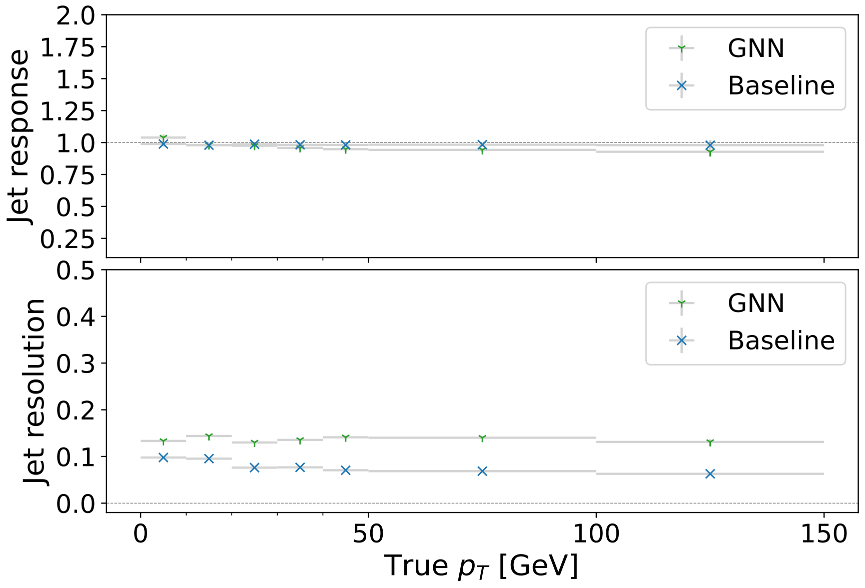
There is no pileup present in this dataset that can be filtered out by a pileup mitigation algorithm. Therefore, all the truth level particles are considered for truth-level jet clustering and all the predicted particles are considered for reco-level jet clustering. After the jets are defined, the same methodology is then used for matching the two sets of jets as the one used in Section 7.1.4. Fig. 7.29 shows an event and the corresponding true and predicted jets.

The jet resolution and response is shown in Fig. 7.27. The response is close to unity, unlike in the toy calorimeter. This results from better calibration and is also observed in single-particle reconstruction in Section 7.2.1. The resolution is approximately 12% and does not change significantly across the p_T spectrum. This shows an excellent reconstruction performance.

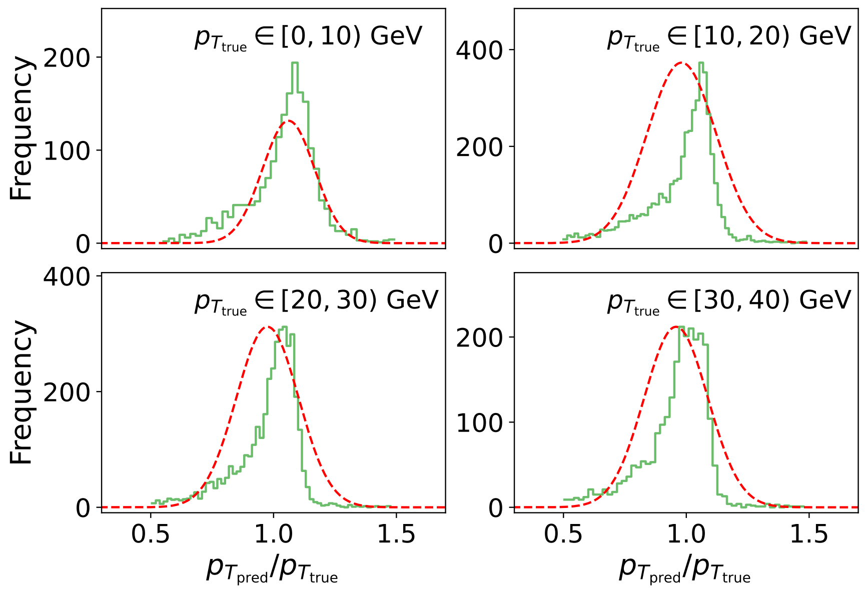
Comparison to TICL

In Fig. 7.28, the jet reconstruction performance of the graph neural network presented in this thesis is compared to the traditional reconstruction algorithm, TICL (Section 4.5)[3]. It can be observed that the neural network

[3]Here, a different dataset is used, as the TICL data was not present in the HGCAL test set used in the other sections. In this dataset the η is more restricted.



(a)



(b)

Figure 7.27: Fig. 7.27a shows mean response (top) and resolution (bottom) of the jets clustered on the HGCAL data. The resolution is computed as the mean corrected standard deviation, as the data is not Gaussian, as shown in Fig. 7.27b where the fitted Gaussian functions are visualised.

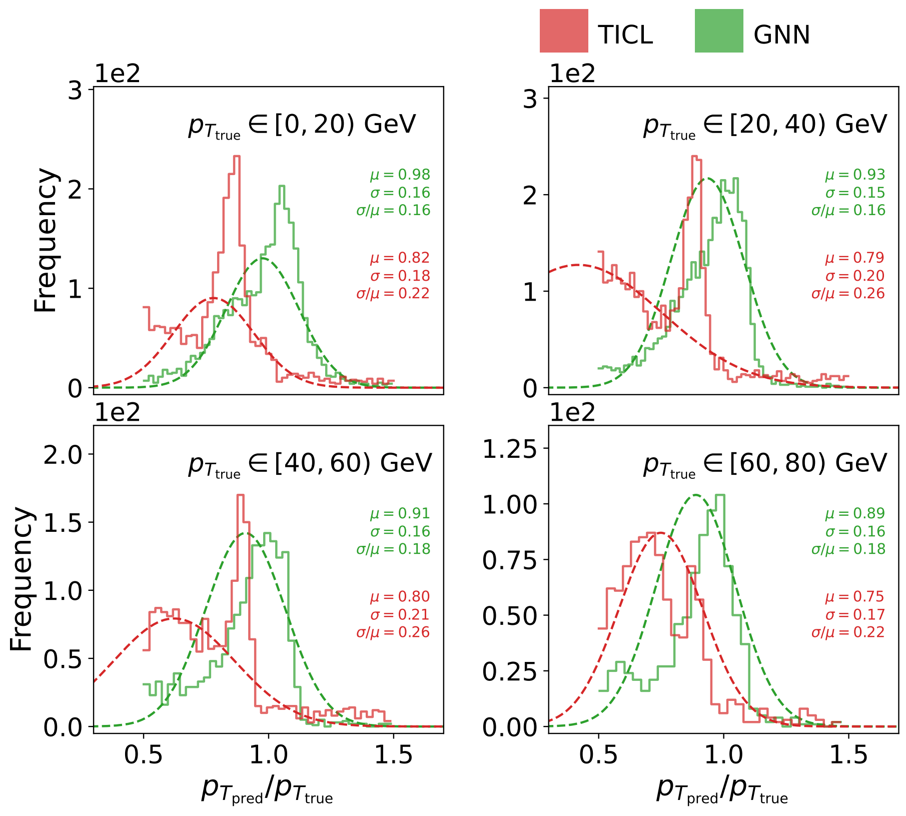
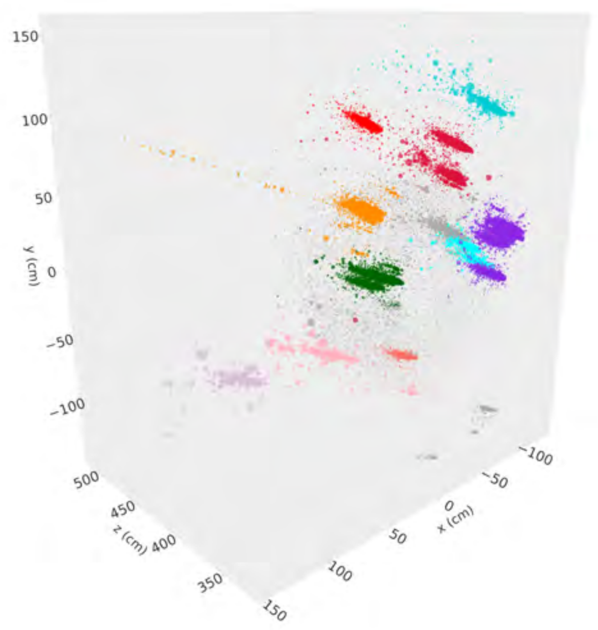


Figure 7.28: Jet reconstruction performance comparison of the graph neural network presented in this thesis with TICL in different $p_{T\text{true}}$ ranges in the CMS HGCAL data. The μ and σ estimates are those of the data, not of Gaussian functions, as the data is not Gaussian, as evident by the fitted Gaussian functions.



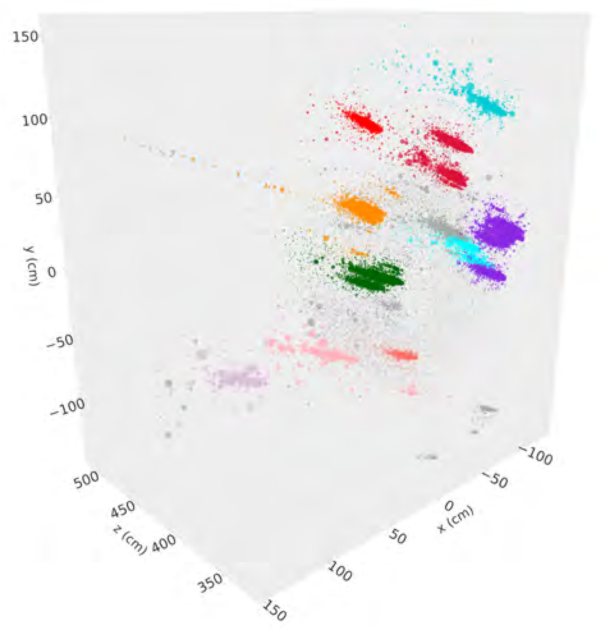


Figure 7.29: Examples of jets clustered on the HGCAL data. Top figure shows jets clustered on true particles and the bottom figure shows jet clustered on predicted particles. Jets that are matched together have the same colours.

substantially outperforms TICL across the p_{true} spectrum, yielding between up to 60% better resolution. Further, the response distribution of only the neural network is centred around 1, highlighting the challenges the classical approach faces in fully reconstructing jets. In particular, it is also noteworthy that the response of the GNN reconstruction shows a more Gaussian behaviour with fewer contributions from tails, making uncertainty estimates in the future more reliable.

7.2.4 Extrapolation to 200 pileup

There is no pileup in the HGCAL data; however, based on particles' local energy density, the reconstruction performance can be extrapolated, employing the technique from Iiyama et al. (2021). Multiple non-overlapping test points in ϕ are sampled in 200 pileup events at $\eta = 2$. The local energy density i.e. energy of all the particles in a cone of $\Delta(\eta, \phi) = 0.3$ around the samples test points is then calculated. The pileup simulations used here are lightweight because detector responses do not need to be considered. A Landau distribution (Landau, 1944; Zolotarev, 1986) is fitted on the resulting distribution. The probability density function of the Landau distribution is given below:

$$p(x; \mu, c) = \frac{1}{\pi c} \int_0^\infty \exp(-t) \cos\left(t\left(\frac{x-\mu}{c}\right) + \frac{2t}{\pi} \log\left(\frac{t}{c}\right)\right) dt \ . \tag{7.13}$$

The fitted $\mu = 109.15$ and $c = 57.22$. The original and the fitted distribution are shown Fig. 7.30.

The fitted distribution is used to infer the weight of every shower by evaluating Equation 7.13 with a shower's local energy density, i.e. energy sum of all the particles in a cone of $\Delta(\eta, \phi) = 0.3$ around it. The performance metrics are then redistributed according to this weight. The efficiency is shown in Fig. 7.31. It approximates the reconstruction efficiency for the toy calorimeter (Fig. 7.9a and Fig. 7.9b). The efficiency reaches 70% at 10 GeV for the electromagnetic particles and 15 GeV for hadronic particles and does not significantly worsen compared to a multi-particle environment.

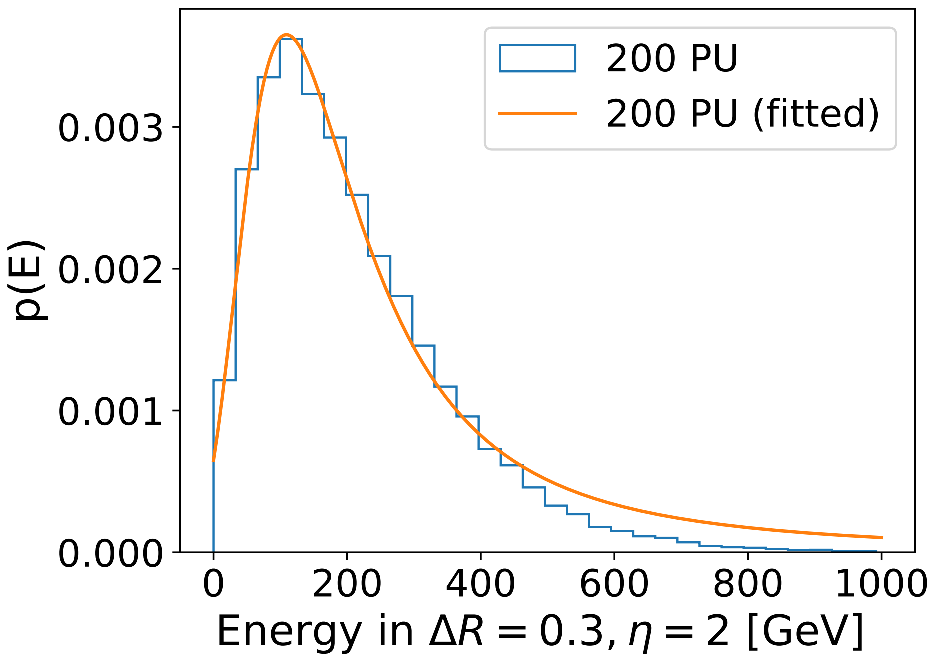


Figure 7.30: Expected local energy density in 200 PU at $\eta = 2$.

The resolution is shown in Fig. 7.32. It does not significantly decrease in the pileup environment and reaches as low as 10% at high p_T for electromagnetic particles and 20% for the hadronic particles.

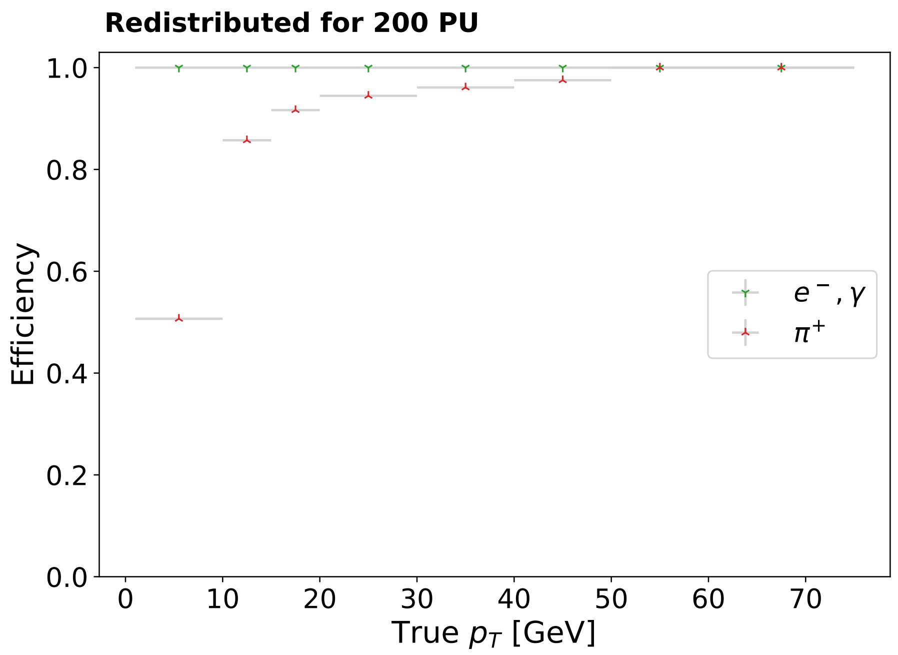


Figure 7.31: Efficiency as a function of truth p_T in the HGCAL. The data is redistributed to mimic 200 PU based on local energy density.

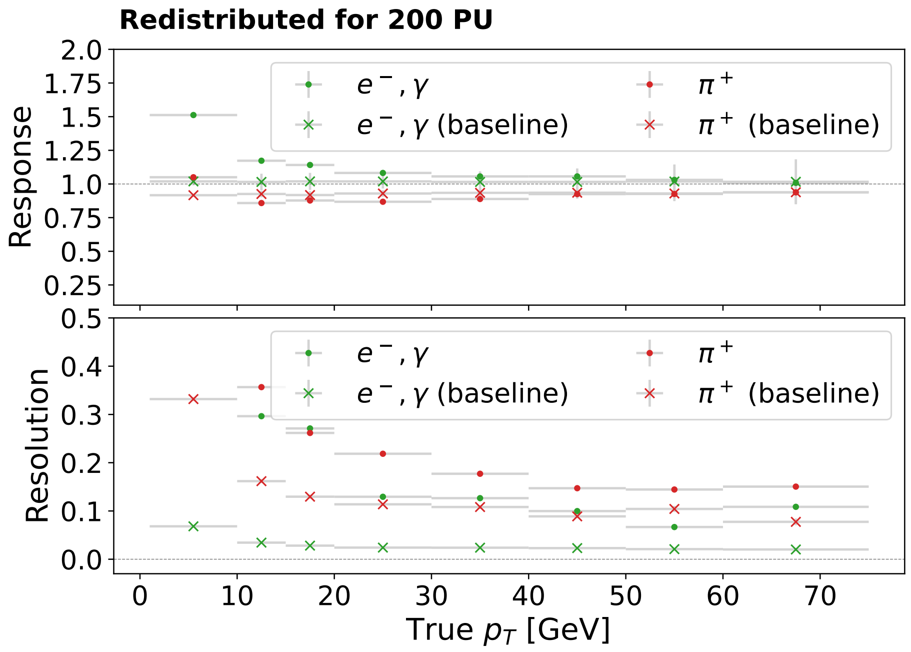


Figure 7.32: Resolution as a function of truth p_T in the HGCAL. The data is redistributed to mimic 200 PU based on the local energy density of particles.

7.3 Summary

In this chapter, the reconstruction results of the method discussed in this thesis are presented. First, the results are presented in LHC-like conditions in a toy calorimeter as major developments are still needed to generate truth for such events in the CMS Software. Although the toy calorimeter is based on the HGCAL and its events match the reconstruction complexity of the HGCAL. The physics performance is presented in different complexity conditions. Reconstruction efficiency is studied as well the energy reconstruction performance of the predicted particles. This is done in different complexity conditions, in 0 pileup i.e. presence of only the test shower, in 40 pileup – which corresponds to event complexity at the LHC currently – and in 200 pileup, which is expected in the HL-LHC conditions. Jet reconstruction performance is also presented on the toy calorimeter in different pileup conditions.

A synthetic multi-particle dataset is generated to study physics performance in the HGCAL itself as well. Multi-particle reconstruction performance is then presented. Over merging and over splitting is often a result of stochasticity in the truth definition and is not necessarily a weakness of the reconstruction method. A multi-matching algorithm is also presented to study the performance in a way which is less sensitive to such stochasticities. Jet reconstruction performance is also discussed and is used to compare to the traditional approach, TICL. Finally, the reconstruction results are extrapolated to 200 pileup conditions using local energy density.

Chapter 8

Computational Performance

While getting a good reconstruction performance is the ultimate goal, it is also important to perform reconstruction within the computational and memory specifications. Therefore, in this chapter, the computational performance of the model is studied. Section 8.1 explores the computational performance of the model as a whole and Section 8.2 explores the optimisations that were needed to get the aforementioned performance.

Finally, Section 8.3 concludes the chapter by presenting a summary.

8.1 Computational performance

Machine learning based approaches, such as the one proposed in this thesis, are mostly composed of code that is often referred to as "embarrassingly parallel". Such pieces of code require little to no effort to parallelise from the algorithmic point of view, and involve a set of instructions that are repeated for a large set of inputs. An example of this is a matrix-multiplication, which is the basis of all modern machine learning techniques. Modern GPUs have thousands of cores which can all run in parallel and, hence, massive performance gains can be harvested.

In Fig. 8.1, the compute specifications of the neural network presented in this thesis are shown for different GPUs. To achieve the presented performance, two optimisations were made, and they are presented in Section 8.2.

Table 8.1: Event complexity in different pileup conditions

Pileup	$\|H\|$	$\|T\|$
0	$2,600 \pm 240$	1.0 ± 0.0
40	$43,000 \pm 8,000$	$1,000 \pm 160$
80	$78,000 \pm 18,000$	$1,800 \pm 190$
140	$124,000 \pm 12,000$	$2,600 \pm 180$
200	$160,000 \pm 12,000$	$3,200 \pm 130$

On a Nvidia A100, the algorithm takes approximately 0.6 and 2 seconds in 40 and 200 pileup, respectively. Nvidia V100 and RTX 2080 Ti perform roughly the same and there, it takes 1.3 and 7 seconds in 40 and 200 pileup, respectively. The superior performance in the A100 GPU is likely because of the higher memory bandwidth of the Nvidia Ampere architecture. The number of vertices (hits) is the most important property when considering the computational complexity. This is shown in Table 8.1 and the numbers correspond to the testing pileup dataset used in Chapter 7. The time complexity of the neural network linearly scales with the number of hits.

The memory requirements can be judged by studying the peak memory allocated during inference. This is also depicted in Fig. 8.1. It takes less than 250 MiB in 40 pileup and less than 1400 MiB in 200 pileup. This shows that the algorithm does not need a high-end GPU to operate and can be deployed on machines with low-end devices. It is to be noted that memory requirements during the training phase are much higher.

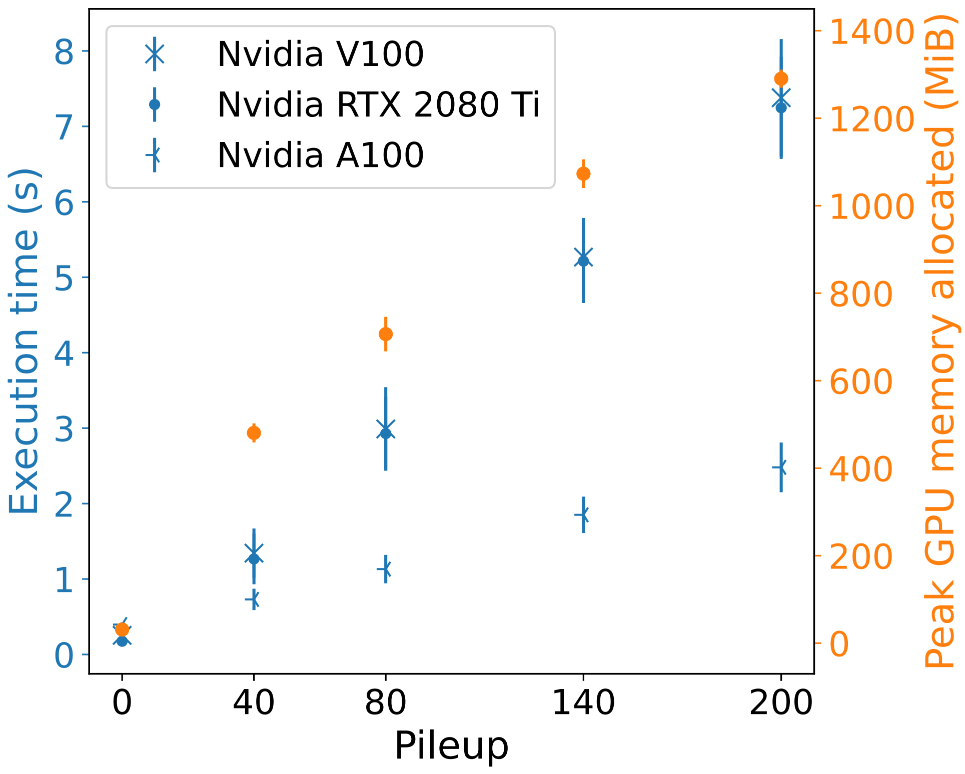


Figure 8.1: Compute specifications of the model as a function of the amount of pileup to run inference with one event. The left axis (blue) shows average execution time on different GPUs, and the right (orange) shows the average peak memory allocated on the GPU. The error bars highlight the standard deviation instead of the standard error to demonstrate event-by-event variance in resource consumption.

8.2 Similarity search

As noted in Section 6.1.1, GravNet layer involves a KNN computation during its execution. This step was optimised for the performance reported in the previous section. With the optimisation, approximately 40% execution time of the neural network in 200 pileup is elapsed in the computation of the KNNs. The optimisation is achieved by binning the search space. The inference clustering step (Section 6.4) has also been optimised using the same methodology. With the optimisation, around 15% of the time is taken in the inference clustering step.

8.2.1 Binning

Let the search space be d dimensional, and, therefore, vertices to be searched can be represented as a matrix $S \in \mathbb{R}^{N\times d}$. $\mathbb{R}$ refers to the set of real numbers. The bin number is first individually computed for every dimension for every vertex. The bins are then flattened in a row-major format to get a global bin number. The binning process is summarised as follows:

$$B, R, I = \text{bin}(S, w) \; . \tag{8.1}$$

Here, w is the bin width, which is the same for all dimensions. The global bin number ($B \in \mathbb{W}^N$) serves as an ordering function to sort the vertices into their associated bins. $\mathbb{W}$ refers to the set of whole numbers. However, instead of sorting all the vertex data, it is easier to store row indices pointing to S in every bin ($I \in \mathbb{W}^N$). It is possible that there will be a variable number of entities in each bin. Therefore, the indices where the bins are split are stored separately in another array, R, called bin splits. The bin splits $R \in \mathbb{W}^{M+1}$, with M as the number of bins. In a zero-indexed scheme, $(i-1)$th and ith element in R represent the index of the first and the last vertices in the ith bin, respectively. The 0th element is always 0.

The bin assignment operation is an embarrassingly parallel operation with a time complexity of $\mathcal{O}(N/T)$. T here is the number of parallel processors. $T = \mathcal{O}(1000)$ on the modern GPUs. Sorting can also be parallelised to

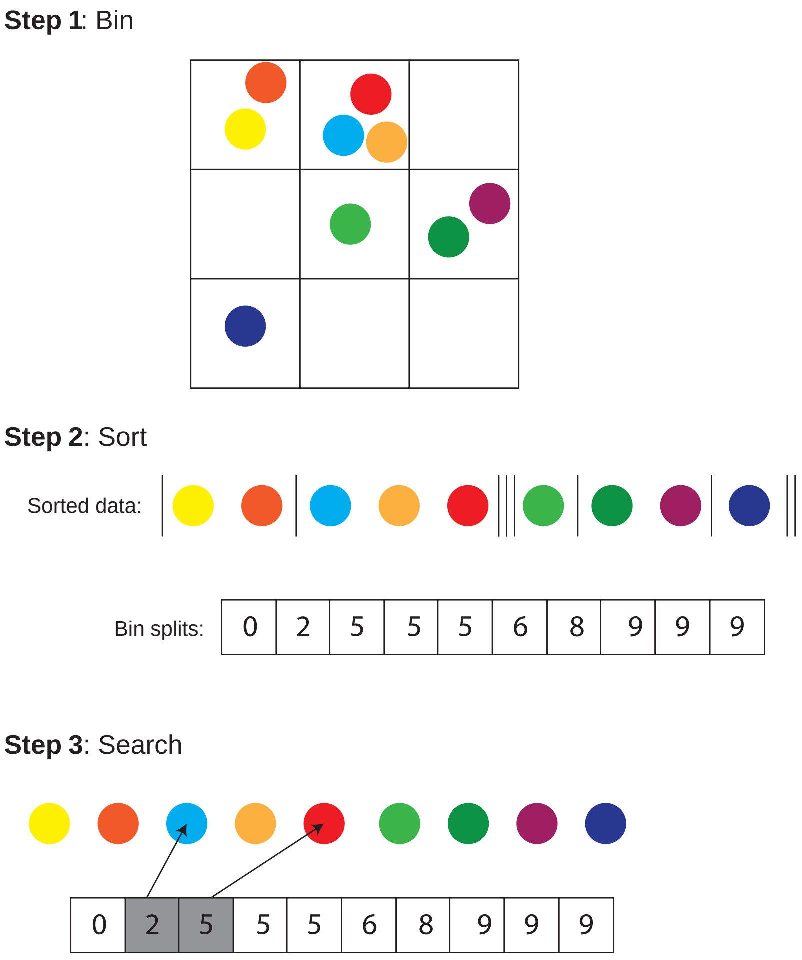


Figure 8.2: Visual representation of how bin search works. In the first step, the bin number are computed for all the bins. In the second step, the data is sorted and the bin splits are computed. In the third step, bin splits (R) can be used to search for data in a specific bin. The search in the second bin is highlighted.

get a time complexity of $\mathcal{O}(N\log^2(N)/T)$ using parallel merge sort (Cole, 1988) and bitonic sort (Nassimi and Sahni, 1979).

However, if, for instance, there are 10 dimensions and 8 bins are used in every dimension, the number of bins will equate to over a million. Therefore, the binning is only done in the first m dimensions. Generally, $m = 3$. Because binning is done in only a few dimensions, it is possible that, in a bin, there are some points that are very far from each other and a secondary condition will be required to filter undesired vertices during the search. A visual representation of the binning process is shown in Fig. 8.2.

8.2.2 Binned K Nearest Neighbours (KNN)

The adapted binned KNN algorithm's performance is shown in Fig. 8.4. The performance of the state-of-the-art approach, FAISS (Johnson et al., 2019), is also shown. For the searches required for operation of the neural network presented in this thesis, the binned algorithm significantly outperforms FAISS, as shown, yielding 2x performance improvement. Furthermore, the memory overhead of the binned KNN is close to zero, while FAISS takes $\mathcal{O}$(GB) GPU memory. The computational complexity of the algorithm is $\mathcal{O}(N^2M^{-1}T^{-1})$, where N and M are the number of vertices and the number of bins, respectively.

Algorithm 3 shows its operation. As the input, it takes a number k, representing how many neighbours to find for each vertex, w, the suggested bin width to use, and the set of vertices ($S \in \mathbb{R}^{N\times f}$), represented as a matrix. The variable d represents the dimensionality of the clustering space. As the output, the nearest neighbours indexes (K) are computed for every vertex as well as squared distance to every nearest neighbour (D).

In the first step, the vertices are divided into bins (Step 1) using the method discussed in Section 8.2.1. The computation is then parallelised over all the vertices.

For every bin, the search is first performed in the bin where a vertex is present ($B(v)$). Second, another layer of surrounding bins is added until and the search is repeated. The addition of one more layer continues until all the

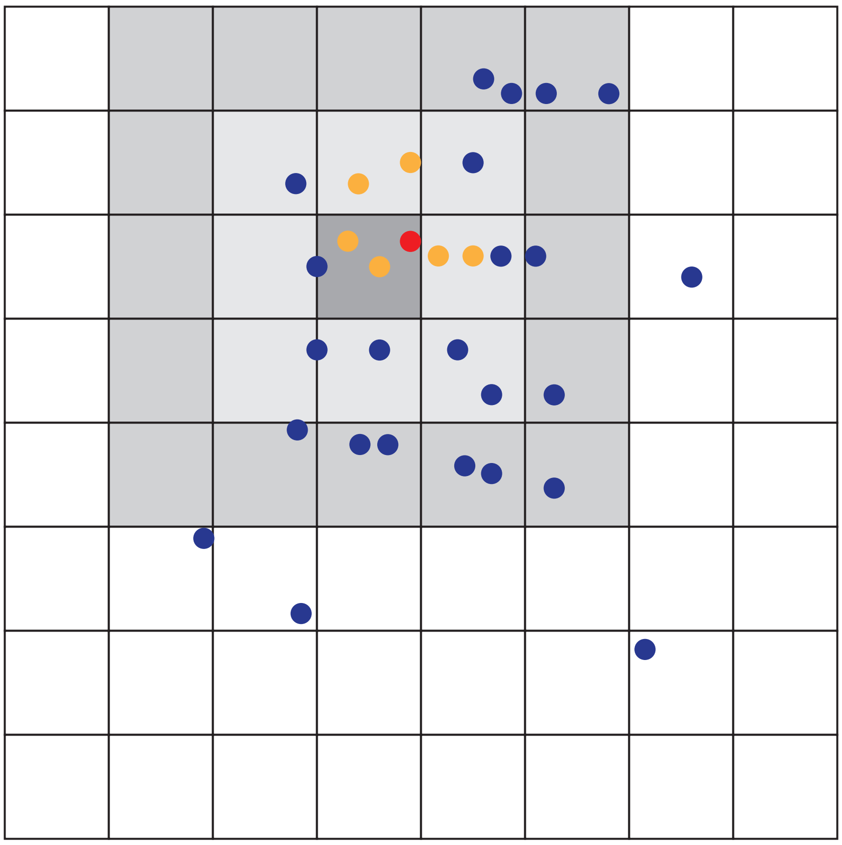


Figure 8.3: A visual representation of the binned KNN algorithm. The red vertex is in the center and the orange vertices are its six nearest neighbours. The nearest neighbours can be discovered by first looking in the bin the red vertex is in. In the second step, the next layer of bins is searched. Three bin layers are highlighted in different shades of grey. In the example shown here, all the 6 nearest neighbours can be discovered already in the second layer.

Algorithm 3 Parallel binned KNN

```
    Input $S, k, w$
    Output $K, D$
1:  $B, R, I \leftarrow \text{bin}(S, w)$
2:  $D \leftarrow 0^{N \times k}$
3:  $K \leftarrow 0^{N \times k}$
4:  for $v = 0$ to $|S|$ do in parallel
5:      $i \leftarrow 0$
6:      $n \leftarrow 0$
7:      $d'_{\max} \leftarrow 0$
8:      $m'_{\max} \leftarrow 0$
9:      while true do
10:         $L \leftarrow \text{stepper}(B[v], i)$
11:         $i \leftarrow i + 1$
12:         for all $l \in L$ do
13:             for $j = R[l-1]$ to $R[l]$ do
14:                 $d' \leftarrow \|S[I[j]] - S[I[v]]\|^2$
15:                 if $n < k$ then
16:                     $D[v, n] \leftarrow d'$
17:                     $K[v, n] \leftarrow I[j]$
18:                     $n \leftarrow n + 1$
19:                     if $d' > d'_{\max}$ then
20:                         $d'_{\max} \leftarrow d'$
21:                         $m'_{\max} \leftarrow I[j]$
22:                     end if
23:                 end if
24:                 if $d' < d'_{\max}$ then
25:                     $D[m'_{\max}, n] \leftarrow d'$
26:                     $K[m'_{\max}, n] \leftarrow I[j]$
27:                     $m'_{\max} \leftarrow \text{argmax}(K[v])$
28:                 end if
29:             end for
30:         end for
31:         if $n = k, iw^2 > d'_{\max}$ then
32:             break
33:         end if
34:     end while
35: end for
```

neighbours are gathered. This is also visually shown in Fig. 8.3. The index of bins layer being currently iterated is labeled as i. The stepper function defined in Step 10 gives as output a set of bins, L by stepping through all the bins in the ith layer. The stepper function can be parameterised and is fast to compute. In Step 13, the elements present in the lth bin are iterated. I can index to the original matrix S, as done in Step 14, where the distance from the vth vertex is computed. If there are less than k nodes yet gathered, the vertex is stored as one of the nearest neighbours and the number of the nearest neighbours gathered (n) is incremented (Step 15). The farthest neighbour (m'_{max}th) and distance to it (d'_{max}) is also re-calculated (Step 15), if needed. If the nearest neighbours' list is full, it is possible that a nearer neighbour was discovered. If that happens, the neighbour that is the farthest (m'_{max}th) is replaced (Step 24). The search ends for a vertex if k neighbours are gathered and if $w^2 > d'_{\text{max}} * i^2$, which ensures that the boundary condition is satisfied.

8.2.3 Inference clustering

In Fig. 8.5, the binned version of the inference clustering algorithm[1] is compared to the naïve version. In the naïve version, the high β hits are separated (Step 2) and iterated through until each one of them is assigned to a shower. Distance is computed from each not-separated vertex in a vectorised fashion and the assignment is done accordingly (Step 7). The naïve version is faster in 0 pileup because of the vectorisation. This step can, however, be optimised by limiting the search in the close-by bins. Here, instead of iterating through nearest bins layer-by-layer, the bins to be searched can be computed directly and can also be parameterised. This algorithm is implemented serially because one vertex can be in the radius of more than one high-β vertices; and has a time complexity of $\mathcal{O}(N^2M^{-1})$. A parallel version (not shown) was studied, but the overhead to run the parallel kernels is far too high to yield any benefits. The inference clustering step consumes only a fraction of the neural network's execution time and hence, further

[1]Presented in Section 6.4

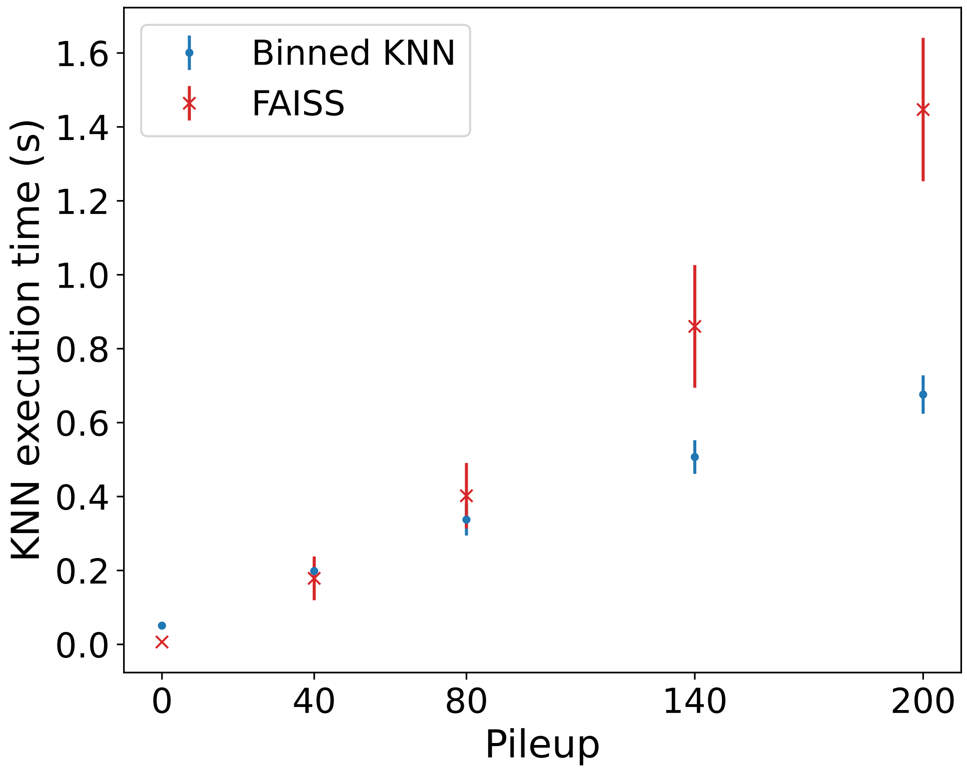


Figure 8.4: Total time taken by the binned KNN algorithm and FAISS (Johnson et al., 2019) to execute all the KNN graph-building operations during execution of the neural network as a function of pileup. The error bars highlight the standard deviation instead of the standard error to demonstrate event-by-event variance in the KNN execution.

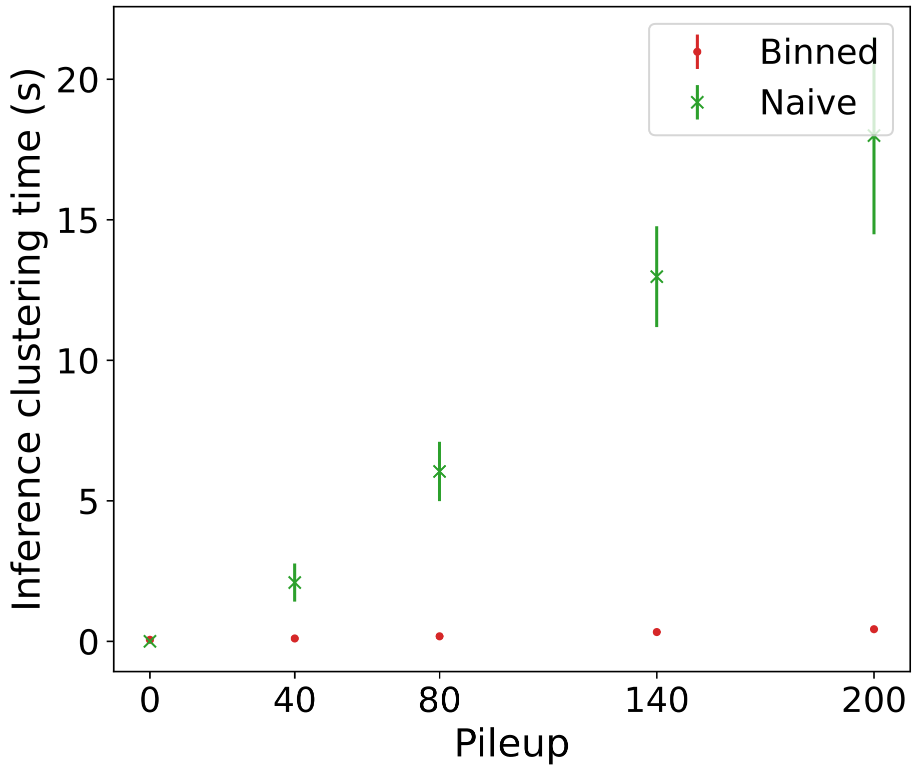


Figure 8.5: Inference clustering time in different pileup conditions. The optimised binned algorithm is compared to the naïve implementation. The error bars highlight the standard deviation instead of the standard error to demonstrate event-by-event variance in inference clustering time.

optimisations only yield diminishing returns.

8.3 Summary

The computational performance of the method presented in this thesis is discussed in this chapter. Both time requirements for inference and peak memory allocated on the GPU are discussed in different pileup conditions and on different GPUs. In 200 pileup, the presented method takes ~ 2 seconds on a high-end A100 GPU and ~ 7 seconds on a lower-end RTX

2080 Ti GPU.

In order to obtain the presented computational performance, two optimisations were performed. First, the KNN computation – which lies at the heart of dynamic GNNs including GravNet – was optimised by binning the search space. The presented KNN algorithm outperforms the state-of-the-art approach FAISS for the HGCAL data, giving up to 50% better performance. Second, the inference clustering algorithm was also optimised using a similar approach. This algorithm is employed at inference time (i.e. is not used during training) to build the final clusters. The naive algorithm takes over 15 seconds in 200 pileup while the optimised version consumes near-zero time.

Chapter 9

Summary and Conclusions

The CMS experiment will replace both the HCAL and the ECAL at the endcaps with the HGCAL for the High Luminosity upgrade of the LHC (HL-LHC). In this thesis, the need for a deep learning based event reconstruction algorithm for high-granularity calorimeters such as the HGCAL was addressed. A novel reconstruction algorithm based on GravNet, a dynamic GNN, was presented. The presented neural network eliminates the need for seeding and only takes as input a set of reconstructed calorimetric hits. As the output, the set of reconstructed particles, their associated hits and the energy are produced in an end-to-end fashion.

Several experiments were presented to study the reconstruction performance. The lack of a good truth-definition algorithm that works in a high pileup environment at the HGCAL was addressed by building a toy calorimeter based on the HGCAL in GEANT4. Single-particle reconstruction performance was studied on the toy calorimeter, first in Run 3 conditions with 40 pileup and then in 200 pileup expected at the HL-LHC. Electromagnetic and hadronic particles were separately studied as they exhibit different responses in the calorimeters. Jet reconstruction performance was also studied by enriching pileup data with $q\bar{q} \rightarrow t\bar{t}$ interactions.

An analysis framework, based on Energy-weighted Intersection Over Union (EIOU), was also presented in this thesis. This framework defines truth-prediction matching as an assignment problem. The EIOU was also extended to an algorithm that performs $K-1$ and $1-K$ matching using Energy-weighted Intersection over Minimum (EIOM), allowing analyses of over merging and over splitting of showers.

The presented analysis framework was used to study the multi-particle reconstruction performance directly on the HGCAL data. This data does not include pileup interactions explicitly, but provides sufficient local particle densities to approximate conditions similar to 200 PU around individual particles. Exploiting this property of the dataset, the results were also extrapolated to 200 PU by re-weighting the contribution of each particle to the evaluation metrics according to their local energy density. Based on this and the experiments conducted on the toy calorimeter, it can be concluded that this method will perform well in the HL-LHC conditions at the HGCAL.

The reconstruction performance of the neural network was compared to the classical approach, TICL. The neural network significantly outperforms the classical approach and yields up to 60% better jet reconstruction performance.

The method also offers excellent generalisation performance. It was shown that the same neural network can perform reconstruction in a wide range of environments, including single particles in presence of only noise, pileup with thousands of particles and pileup enriched with $q\bar{q} \rightarrow t\bar{t}$ interactions where highly energetic hadrons, with energy range never seen in the training set, are present.

The computational performance of the method was also optimised using a binned search for fast KNN computation. The algorithm was implemented using Nvidia CUDA. The same method was also used to optimise the inference clustering step significantly. These optimisations enabled fast inference in high pileup environments and it was shown that the neural network takes less than 3 seconds in 200 pileup on a high-end GPU. This is within computational constraints for offline computing. Furthermore, the GPU memory required for inference is reasonably low, which enables the method to work on low-end GPUs.

For future research, the neural network can be easily extended to perform particle ID. Moreover, currently, the algorithm does not use tracker information. The addition of track information is expected to yield significant performance improvements. One way to do so is by matching reconstructed particles to tracks and performing post-processing to improve the reconstruction performance, although it will require another matching algorithm that has the ability to re-split and re-merge showers. However, benefiting from the generalisability of the GNNs, the tracks can be naturally added to the input hits as another vertex type without the need for a dedicated matching algorithm a posteriori.

Both the computational and reconstruction performance of the model can be likely improved by adding a pre-clustering layer. A significant fraction of hits that belong to the same shower are very easy to identify and can be grouped together by a lightweight neural network. This will also increase the receptive field, resulting in a possible improvement in the reconstruction performance.

Another way to improve reconstruction performance is by building a graph at the output of the neural network. In this graph, each reconstructed shower will be taken as a node of type A, connected to its associated hits, represented as type B nodes. Once this is done, graph neural network approaches that are more expressive compared to dynamic graph neural networks can be used. It will result in a substantial improvement in the receptive field of the neural network and might improve reconstruction performance.

In summary, the work presented in this thesis resulted in the first-ever example of single-shot calorimetric reconstruction of $\mathcal{O}(1000)$ particles in HL conditions with 200 pileup, showing very promising physics and computational performance. Furthermore, the techniques presented here open up a large variety of applications beyond calorimetry, as well as paths for future research.

List of Figures

References

Aaboud, M., Aad, G., Abbott, B., Abdallah, J., Abeloos, B., Abidi, S., AbouZeid, O., Abraham, N., Abramowicz, H., Abreu, H., et al. (2017). 'Performance of the ATLAS track reconstruction algorithms in dense environments in LHC Run 2'. *The European Physical Journal C* 77 (10), pp. 1–30.

Aaboud, M., Aad, G., Abbott, B., Abdallah, J., Abeloos, B., Abidi, S. H., AbouZeid, O., Abraham, N., Abramowicz, H., Abreu, H., et al. (2017). 'Jet reconstruction and performance using particle flow with the ATLAS Detector'. *The European Physical Journal C* 77 (7), pp. 1–47.

Aad, G., Abbott, B., Abdallah, J., Aben, R., Abolins, M., AbouZeid, O., Abramowicz, H., Abreu, H., Abreu, R., Abulaiti, Y., et al. (2017). 'Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1'. *The European Physical Journal C* 77 (7), pp. 1–73.

Abdughani, M., Ren, J., Wu, L., and Yang, J. M. (2019). 'Probing stop pair production at the LHC with graph neural networks'. *Journal of High Energy Physics* 2019 (8), pp. 1–14.

Abdughani, M., Wang, D., Wu, L., Yang, J. M., and Zhao, J. (2021). 'Probing the triple Higgs boson coupling with machine learning at the LHC'. *Physical Review D* 104 (5), p. 056003.

Abreu, P., Adam, W., Adye, T., Agasi, E., Alekseev, G., Algeri, A., Allen, P., Almehed, S., Alvsvaag, S., Amaldi, U., et al. (1992). 'Classification of the hadronic decays of the Z0 into b and c quark pairs using a neural network'. *Physics Letters B* 295 (3-4), pp. 383–395.

Acciarri, R., Adams, C., An, R., Asaadi, J., Auger, M., Bagby, L., Baller, B., Barr, G., Bass, M., Bay, F., et al. (2017). 'Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber'. *Journal of instrumentation* 12 (03), P03011.

Agostinelli, S. et al. (2003). 'GEANT4—a simulation toolkit'. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506 (3), pp. 250–303.

Ai, P., Wang, D., Huang, G., and Sun, X. (2018). 'Three-dimensional convolutional neural networks for neutrinoless double-beta decay signal/background discrimination in high-pressure gaseous Time Projection Chamber'. *Journal of Instrumentation* 13 (08), P08015.

Akchurin, N., Cowden, C., Damgov, J., Hussain, A., and Kunori, S. (2021). 'On the use of neural networks for energy reconstruction in high-granularity calorimeters'. *Journal of Instrumentation* 16 (12), P12036.

Alice Collaboration (2006). 'ALICE: physics performance report, volume II'. *Journal of Physics G: Nuclear and Particle Physics* 32, pp. 1295–2040.

Almeida, L. G., Backović, M., Cliche, M., Lee, S. J., and Perelstein, M. (2015). 'Playing tag with ANN: boosted top identification with pattern recognition'. *Journal of High Energy Physics* 2015 (7), pp. 1–21.

Anderson, C. D. (1933). 'The positive electron'. *Physical review* 43 (6), p. 491.

Arnaudon, L., Magistris, M., Paoluzzi, M., Hori, M., Küchler, D., Bourquin, P., Hanke, K., Wegner, R., Rossi, C., Bellodi, G., et al. (2006). *Linac4 technical design report*. Tech. rep.

ATLAS Collaboration (2014). 'A neural network clustering algorithm for the ATLAS silicon pixel detector'. *Journal of Instrumentation* 9 (09), P09009.

ATLAS Collaboration (2012). ‘G. Aad et al., Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC’. *Phys. Lett. B* 716 (1), pp. 1–29.

ATLAS Collaboration (2017). ‘Quark versus gluon jet tagging using jet images with the ATLAS detector’. *ATLAS Public Note ATL-PHYS-PUB-2017-017.*

ATLAS Collaboration (2008). ‘The ATLAS experiment at the CERN large hadron collider’. *Journal of instrumentation* 3 (S08003).

ATLAS collaboration (2019). *Development of ATLAS primary vertex reconstruction for LHC Run 3.* Tech. rep. Tech. Rep. ATL-PHYS-PUB-2019-015, CERN, Geneva (Apr, 2019).

Atwood, J. and Towsley, D. (2016). ‘Diffusion-convolutional neural networks’. *Advances in neural information processing systems* 29.

Aurisano, A., Radovic, A., Rocco, D., Himmel, A., Messier, M., Niner, E., Pawloski, G., Psihas, F., Sousa, A., and Vahle, P. (2016). ‘A convolutional neural network neutrino event classifier’. *Journal of Instrumentation* 11 (09), P09001.

Ayres, D., Drake, G., Goodman, M., Grudzinski, J., Guarino, V., Talaga, R., Zhao, A., Stamoulis, P., Stiliaris, E., Tzanakos, G., et al. (2007). *The NOvA technical design report.* Tech. rep. Fermi National Accelerator Lab.(FNAL), Batavia, IL (United States).

Baldi, P., Bauer, K., Eng, C., Sadowski, P., and Whiteson, D. (2016). ‘Jet substructure classification in high-energy physics with deep neural networks’. *Physical Review D* 93 (9), p. 094034.

Baldi, P., Sadowski, P., and Whiteson, D. (2014). ‘Searching for exotic particles in high-energy physics with deep learning’. *Nature communications* 5 (1), pp. 1–9.

Barney, D. (2015). ‘CMS Slice’. URL: `https://cds.cern.ch/record/2628641`.

Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al. (2016). ‘Interaction networks for learning about objects, relations and physics’. *Advances in neural information processing systems* 29.

Belayneh, D., Carminati, F., Farbin, A., Hooberman, B., Khattak, G., Liu, M., Liu, J., Olivito, D., Pacela, V. B., Pierini, M., et al. (2020). ‘Calorimetry with deep learning: particle simulation and reconstruction for collider physics’. *The European Physical Journal C* 80 (7), pp. 1–31.

Bentley, J. L. (1975). ‘Multidimensional binary search trees used for associative searching’. *Communications of the ACM* 18 (9), pp. 509–517.

Bernreuther, E., Finke, T., Kahlhoefer, F., Krämer, M., and Mück, A. (2021). ‘Casting a graph net to catch dark showers’. *SciPost Physics* 10 (2), p. 046.

Bhattacharya, S., Chernyavskaya, N., Ghosh, S., Gray, L., Kieseler, J., Klijnsma, T., Long, K., Nawaz, R., Pedro, K., Pierini, M., et al. (2022). ‘GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter’. *arXiv preprint arXiv:2203.01189*.

Bottou, L. and Bousquet, O. (2007). ‘The tradeoffs of large scale learning’. *Advances in neural information processing systems* 20.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and

Zhang, Q. (2018). *JAX: composable transformations of Python+NumPy programs.* Version 0.3.13. URL: `http://github.com/google/jax`.

Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., and Shafait, F. (2013). 'High-performance OCR for printed English and Fraktur using LSTM networks'. *2013 12th international conference on document analysis and recognition.* IEEE, pp. 683–687.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). 'Spectral networks and locally connected networks on graphs'. *arXiv preprint arXiv:1312.6203.*

Buhmann, E., Diefenbacher, S., Eren, E., Gaede, F., Kasieczka, G., Korol, A., and Krüger, K. (2021). 'Getting high: high fidelity simulation of high granularity calorimeters with high speed'. *Computing and Software for Big Science* 5 (1), pp. 1–17.

Cacciari, M., Salam, G. P., and Soyez, G. (2012). 'FastJet user manual'. *The European Physical Journal C* 72 (3), pp. 1–54.

Cacciari, M., Salam, G. P., and Soyez, G. (2008). 'The anti-kt jet clustering algorithm'. *Journal of High Energy Physics* 2008 (04), p. 063.

Cauchy, A. et al. (1847). 'Méthode générale pour la résolution des systemes d'équations simultanées'. *Comp. Rend. Sci. Paris* 25 (1847), pp. 536–538.

Chabanat, E. and Estre, N. (2005). 'Deterministic annealing for vertex finding at CMS'.

Charpak, G., Bouclier, R., Bressani, T., Favier, J., and Zupančič, Č. (1968). 'The use of multiwire proportional counters to select and localize charged particles'. *Nuclear Instruments and Methods* 62 (3), pp. 262–268.

Chen, T. and He, T. (2015). 'Higgs boson discovery with boosted trees'. *NIPS 2014 workshop on high-energy physics and machine learning*. PMLR, pp. 69–80.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). 'Learning phrase representations using RNN encoder-decoder for statistical machine translation'. *arXiv preprint arXiv:1406.1078*.

Choi, S., Lee, S. J., and Perelstein, M. (2019). 'Infrared safety of a neural-net top tagging algorithm'. *Journal of High Energy Physics* 2019 (2), pp. 1–14.

Clarke, F. H. (1990). *Optimization and nonsmooth analysis*. SIAM.

CMS Collaboration (Oct. 2014). 'Description and performance of track and primary-vertex reconstruction with the CMS tracker'. *Journal of Instrumentation* 9 (10), P10009–P10009. DOI: 10.1088/1748-0221/9/10/p10009. URL: https://doi.org/10.1088/1748-0221/9/10/p10009.

CMS Collaboration (2012). 'Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC'. *Physics Letters B* 716 (1), pp. 30–61.

CMS Collaboration (2008). 'The CMS experiment at the CERN LHC'. *JInst* 3, S08004.

CMS Collaboration (Nov. 2017). *The Phase-2 Upgrade of the CMS Endcap Calorimeter*. Tech. rep. Geneva: CERN. URL: https://cds.cern.ch/record/2293646.

CMS collaboration (2022). *CMS Software*. URL: http://cms-sw.github.io (visited on 10/20/2022).

CMS collaboration (2017). ‘Particle-flow reconstruction and global event description with the CMS detector’. *JINST* 12 (10), P10003.

CMS collaboration (2010a). ‘Performance and operation of the CMS electromagnetic calorimeter’. *Journal of Instrumentation* 5 (03), T03010.

CMS collaboration (2010b). ‘Performance of CMS hadron calorimeter timing and synchronization using test beam, cosmic ray, and LHC beam data’. *Journal of Instrumentation* 5 (03), T03013.

CMS collaboration (2015a). ‘Performance of photon reconstruction and identification with the CMS detector in proton-proton collisions at $\sqrt{s}$= 8 TeV’. *Journal of Instrumentation* 10 (8), P08010.

CMS collaboration (2018). ‘Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s}$= 13 TeV’.

CMS collaboration (2015b). ‘Technical proposal for the Phase-II upgrade of the Compact Muon Solenoid’. *CMS Technical Proposal CERN-LHCC-2015-010, CMS-TDR-15-02.*

CMS collaboration (2014). ‘The CMS ECAL performance with examples’. *JINST* 9, p. C02008.

Cole, R. (1988). ‘Parallel merge sort’. *SIAM Journal on Computing* 17 (4), pp. 770–785.

Cristella, L. (2021). ‘A novel reconstruction framework for an imaging calorimeter for HL-LHC’. *EPJ Web of Conferences.* Vol. 251. EDP Sciences, p. 03013.

Dalal, N. and Triggs, B. (2005). ‘Histograms of oriented gradients for human detection’. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05).* Vol. 1. Ieee, pp. 886–893.

De Oliveira, L., Nachman, B., and Paganini, M. (2020). ‘Electromagnetic showers beyond shower shapes’. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 951, p. 162879.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). ‘Convolutional neural networks on graphs with fast localized spectral filtering’. *Advances in neural information processing systems* 29.

DELPHI collaboration (1991). ‘The DELPHI detector at LEP’. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 303 (2), pp. 233–276. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/0168-9002(91)90793-P`. URL: `https://www.sciencedirect.com/science/article/pii/016890029190793P`.

Denby, B. (1988). ‘Neural networks and cellular automata in experimental high energy physics’. *Computer Physics Communications* 49 (3), pp. 429–448.

Denby, B. (1999). ‘Neural networks in high energy physics: a ten year perspective’. *Computer Physics Communications* 119 (2-3), pp. 219–231.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ‘Imagenet: A large-scale hierarchical image database’. *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Di Sipio, R., Giannelli, M. F., Haghighat, S. K., and Palazzo, S. (2019). ‘DijetGAN: a generative-adversarial network approach for the simulation of QCD dijet events at the LHC’. *Journal of high energy physics* 2019 (8), pp. 1–17.

Djouadi, A. (1995). ‘Higgs particles at future hadron and electron-positron colliders’. *International Journal of Modern Physics A* 10 (01), pp. 1–63.

Dominguez, D. (2014). '3D cut of the LHC dipole. Coupe 3D du dipôle du LHC'. General Photo. URL: `https://cds.cern.ch/record/1741036`.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). 'Convolutional networks on graphs for learning molecular fingerprints'. *Advances in neural information processing systems* 28.

Englert, F. and Brout, R. (1964). 'Broken symmetry and the mass of gauge vector mesons'. *Physical review letters* 13 (9), p. 321.

Erdmann, M., Glombitza, J., and Quast, T. (2019). 'Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network'. *Computing and Software for Big Science* 3 (1), pp. 1–13.

Fabjan, C. W. and Gianotti, F. (2003). 'Calorimetry for particle physics'. *Reviews of Modern Physics* 75 (4), p. 1243.

FCC collaboration (2019a). 'Calorimeters for the FCC-hh'. *arXiv preprint arXiv:1912.09962.*

FCC collaboration (2019b). 'FCC physics opportunities'. *The European Physical Journal C* 79 (6), pp. 1–161.

Field, R. (2011). 'Min-bias and the underlying event at the LHC'. *arXiv preprint arXiv:1110.5530.*

Fraser, K. and Schwartz, M. D. (2018). 'Jet charge and machine learning'. *Journal of High Energy Physics* 2018 (10), pp. 1–18.

Gilani, A., Qasim, S. R., Malik, I., and Shafait, F. (2017). 'Table detection using deep learning'. *2017 14th IAPR international conference on document analysis and recognition (ICDAR).* Vol. 1. IEEE, pp. 771–776.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). ‘Neural message passing for quantum chemistry’. *International conference on machine learning*. PMLR, pp. 1263–1272.

Glorot, X. and Bengio, Y. (2010). ‘Understanding the difficulty of training deep feedforward neural networks’. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 249–256.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). ‘Generative Adversarial Nets’. *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger. Vol. 27. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). ‘Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks’. *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376.

Guo, J., Li, J., Li, T., Xu, F., and Zhang, W. (2018). ‘Deep learning for R-parity violating supersymmetry searches at the LHC’. *Physical Review D* 98 (7), p. 076017.

Guralnik, G. S., Hagen, C. R., and Kibble, T. W. (1964). ‘Global conservation laws and massless particles’. *Physical Review Letters* 13 (20), p. 585.

Hagiwara, K. et al. (July 2002). ‘Review of Particle Properties’. *Phys. Rev. D* 66 (1), p. 010001. DOI: `10.1103/PhysRevD.66.010001`. URL: `https://link.aps.org/doi/10.1103/PhysRevD.66.010001`.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). 'Inductive representation learning on large graphs'. *Advances in neural information processing systems* 30.

Hartmann, F. (2007). 'Construction of the CMS Tracker'. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 572 (1). Frontier Detectors for Frontier Physics, pp. 73–76. ISSN: 0168-9002. DOI: `https://doi.org/10.1016/j.nima.2006.10.349`. URL: `https://www.sciencedirect.com/science/article/pii/S0168900206019851`.

Henaff, M., Bruna, J., and LeCun, Y. (2015). 'Deep convolutional networks on graph-structured data'. *arXiv preprint arXiv:1506.05163*.

Henrion, I., Brehmer, J., Bruna, J., Cho, K., Cranmer, K., Louppe, G., and Rochette, G. (2017). 'Neural message passing for jet physics'.

Higgs, P. W. (1964). 'Broken symmetries and the masses of gauge bosons'. *Physical Review Letters* 13 (16), p. 508.

Hornik, K., Stinchcombe, M., and White, H. (1989). 'Multilayer feedforward networks are universal approximators'. *Neural networks* 2 (5), pp. 359–366.

Hubel, D. H. and Wiesel, T. N. (1959). 'Receptive fields of single neurones in the cat's striate cortex'. *The Journal of physiology* 148 (3), p. 574.

Iiyama, Y., Cerminara, G., Gupta, A., Kieseler, J., Loncar, V., Pierini, M., Qasim, S. R., Rieger, M., Summers, S., Van Onsem, G., et al. (2021). 'Distance-weighted graph neural networks on FPGAs for real-time particle reconstruction in high energy physics'. *Frontiers in big Data* 3, p. 598927.

Ioffe, S. and Szegedy, C. (2015). 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. *International conference on machine learning*. PMLR, pp. 448–456.

Jaccard, P. (1901). ‘Étude comparative de la distribution florale dans une portion des Alpes et des Jura’. *Bull Soc Vaudoise Sci Nat* 37, pp. 547–579.

Johnson, J., Douze, M., and Jégou, H. (2019). ‘Billion-scale similarity search with GPUs’. *IEEE Transactions on Big Data* 7 (3), pp. 535–547.

Kalman, R. E. (Mar. 1960). ‘A New Approach to Linear Filtering and Prediction Problems’. *Journal of Basic Engineering* 82 (1), pp. 35–45. ISSN: 0021-9223. DOI: `10.1115/1.3662552`. eprint: `https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35\_1.pdf`. URL: `https://doi.org/10.1115/1.3662552`.

Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). ‘Deep learning in agriculture: A survey’. *Computers and electronics in agriculture* 147, pp. 70–90.

Kasieczka, G., Plehn, T., Russell, M., and Schell, T. (2017). ‘Deep-learning top taggers or the end of QCD?’ *Journal of High Energy Physics* 2017 (5), pp. 1–22.

Kendall, A., Gal, Y., and Cipolla, R. (2018). ‘Multi-task learning using uncertainty to weigh losses for scene geometry and semantics’. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491.

Kieseler, J. (2020). ‘Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph, and image data’. *The European Physical Journal C* 80 (9), pp. 1–12.

Kingma, D. P. and Ba, J. (2014). ‘Adam: A method for stochastic optimization’. *arXiv preprint arXiv:1412.6980.*

Kingma, D. P. and Welling, M. (2013). ‘Auto-encoding variational bayes’. *arXiv preprint arXiv:1312.6114.*

Kipf, T. N. and Welling, M. (2016). 'Semi-supervised classification with graph convolutional networks'. *arXiv preprint arXiv:1609.02907*.

Komiske, P. T., Metodiev, E. M., Nachman, B., and Schwartz, M. D. (2018). 'Learning to classify from impure samples with high-dimensional data'. *Physical Review D* 98 (1), p. 011502.

Komiske, P. T., Metodiev, E. M., and Schwartz, M. D. (2017). 'Deep learning in color: towards automated quark/gluon jet discrimination'. *Journal of High Energy Physics* 2017 (1), pp. 1–23.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). 'ImageNet Classification with Deep Convolutional Neural Networks'. *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. Burges, L. Bottou, and K. Weinberger. Vol. 25. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

Kuhn, H. W. (1955). 'The Hungarian method for the assignment problem'. *Naval research logistics quarterly* 2 (1-2), pp. 83–97.

Landau, L. D. (1944). 'On the energy loss of fast particles by ionization'. *J. Phys.* 8, pp. 201–205.

Lattes, C. M., Muirhead, H., Occhialini, G. P., and Powell, C. F. (1947). 'Processes involving charged mesons'. *Nature* 159 (4047), pp. 694–697.

Levenberg, K. (1944). 'A method for the solution of certain non-linear problems in least squares'. *Quarterly of applied mathematics* 2 (2), pp. 164–168.

LHCb Collaboration (2008). 'The LHCb detector at the LHC'. *Journal of instrumentation* 3 (08), S08005.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). 'Gated graph sequence neural networks'. *arXiv preprint arXiv:1511.05493*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). 'Ssd: Single shot multibox detector'. *European conference on computer vision*. Springer, pp. 21–37.

Louppe, G., Cho, K., Becot, C., and Cranmer, K. (2019). 'QCD-aware recursive neural networks for jet physics'. *Journal of High Energy Physics* 2019 (1), pp. 1–23.

Lowe, D. G. (1999). 'Object recognition from local scale-invariant features'. *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, pp. 1150–1157.

Macaluso, S. and Shih, D. (2018). 'Pulling out all the tops with computer vision and deep learning'. *Journal of High Energy Physics* 2018 (10), pp. 1–27.

Mahalanobis, P. C. (1936). 'On the generalized distance in statistics'. National Institute of Science of India.

Martelli, A. (2017). *The CMS HGCAL detector for HL-LHC upgrade. The CMS HGCAL detector for HL-LHC upgrade.* Tech. rep. proceedings of the LHCP2017 conference, 6 pages, 8 figures. arXiv: 1708.08234. URL: https://cds.cern.ch/record/2281440.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. URL: `https://www.tensorflow.org/`.

Martínez, J. A., Cerri, O., Spiropulu, M., Vlimant, J., and Pierini, M. (2019). 'Pileup mitigation at the Large Hadron Collider with graph neural networks'. *The European Physical Journal Plus* 134 (7), p. 333.

Mikuni, V. and Canelli, F. (2020). 'ABCNet: An attention-based method for particle tagging'. *The European Physical Journal Plus* 135 (6), pp. 1–11.

MiniBooNE collaboration (2009). 'The miniboone detector'. *Nuclear instruments and methods in physics research section a: accelerators, spectrometers, detectors and associated equipment* 599 (1), pp. 28–46.

Mobs, E. (2019). 'The CERN accelerator complex in 2019. Complexe des accélérateurs du CERN en 2019'. General Photo. URL: `https://cds.cern.ch/record/2684277`.

Moreno, E. A., Cerri, O., Duarte, J. M., Newman, H. B., Nguyen, T. Q., Periwal, A., Pierini, M., Serikova, A., Spiropulu, M., and Vlimant, J.-R. (2020). 'JEDI-net: a jet identification algorithm based on interaction networks'. *The European Physical Journal C* 80 (1), pp. 1–15.

Moreno, E. A., Nguyen, T. Q., Vlimant, J.-R., Cerri, O., Newman, H. B., Periwal, A., Spiropulu, M., Duarte, J. M., and Pierini, M. (2020). 'Interaction networks for the identification of boosted h$\to$ b b decays'. *Physical Review D* 102 (1), p. 012010.

Motz, J., Olsen, H. A., and Koch, H. (1969). 'Pair production by photons'. *Reviews of Modern Physics* 41 (4), p. 581.

Müller, B., Schukraft, J., and Wysłouch, B. (2012). 'First results from Pb+ Pb collisions at the LHC'. *Annual Review of Nuclear and Particle Science* 62, pp. 361–386.

Musella, P. and Pandolfi, F. (2018). 'Fast and accurate simulation of particle detectors using generative adversarial networks'. *Computing and Software for Big Science* 2 (1), pp. 1–11.

Myers, S. (2012). 'Large Hadron Collider commissioning and first operation'. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370 (1961), pp. 859–875.

Nassimi, D. and Sahni, S. (1979). 'Bitonic sort on a mesh-connected parallel computer'. *IEEE Transactions on Computers* 28 (01), pp. 2–7.

Neubeck, A. and Van Gool, L. (2006). 'Efficient non-maximum suppression'. *18th International Conference on Pattern Recognition (ICPR'06).* Vol. 3. IEEE, pp. 850–855.

Neubüser, C., Kieseler, J., and Lujan, P. (2022). 'Optimising longitudinal and lateral calorimeter granularity for software compensation in hadronic showers using deep neural networks'. *The European Physical Journal C* 82 (1), pp. 1–9.

Neven, D., Brabandere, B. D., Proesmans, M., and Gool, L. V. (2019). 'Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth'. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8837–8845.

Oerter, R. (2006). *The theory of almost everything: The standard model, the unsung triumph of modern physics.* Penguin.

Oliveira, L. de, Kagan, M., Mackey, L., Nachman, B., and Schwartzman, A. (2016). ‘Jet-images—deep learning edition’. *Journal of High Energy Physics* 2016 (7), pp. 1–32.

Oliveira, L. de, Paganini, M., and Nachman, B. (2017). ‘Learning particle physics by example: location-aware generative adversarial networks for physics synthesis’. *Computing and Software for Big Science* 1 (1), pp. 1–24.

Paganini, M., Oliveira, L. de, and Nachman, B. (2018). ‘CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks’. *Physical Review D* 97 (1), p. 014021.

Pantaleo, F. and Rovere, M. (2022). *The Iterative Clustering framework for the CMS HGCAL Reconstruction.* Tech. rep.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). ‘PyTorch: An Imperative Style, High-Performance Deep Learning Library’. *Advances in Neural Information Processing Systems 32.* Curran Associates, Inc., pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Peterson, C. (1989). ‘Track finding with neural networks’. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 279 (3), pp. 537–545.

Piacquadio, G., Prokofiev, K., and Wildauer, A. (2008). ‘Primary vertex reconstruction in the ATLAS experiment at LHC’. *Journal of Physics: Conference Series.* Vol. 119. 3. IOP Publishing, p. 032033.

Pol, A. A., Aarrestad, T., Govorkova, E., Halily, R., Klempner, A., Kopetz, T., Loncar, V., Ngadiuba, J., Pierini, M., Sirkin, O., and Summers, S. (July

2022). ‘Lightweight jet reconstruction and identification as an object detection task’. *Machine Learning: Science and Technology* 3 (2), p. 025016. DOI: 10.1088/2632-2153/ac7a02. URL: https://dx.doi.org/10.1088/2632-2153/ac7a02.

Qasim, S. R., Chernyavskaya, N., Kieseler, J., Long, K., Viazlo, O., Pierini, M., and Nawaz, R. (2022). ‘End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks’. *Eur. Phys. J. C* 82 (8), p. 753. DOI: 10.1140/epjc/s10052-022-10665-7. arXiv: 2204.01681. URL: https://cds.cern.ch/record/2807254.

Qasim, S. R., Kieseler, J., Iiyama, Y., and Pierini, M. (2019). ‘Learning representations of irregular particle-detector geometry with distance-weighted graph networks’. *The European Physical Journal C* 79 (7), pp. 1–11.

Qasim, S. R., Long, K., Kieseler, J., Pierini, M., and Nawaz, R. (2021). ‘Multi-particle reconstruction in the High Granularity Calorimeter using object condensation and graph neural networks’. *EPJ Web Conf.* 251, p. 03072. DOI: 10.1051/epjconf/202125103072. arXiv: 2106.01832. URL: https://cds.cern.ch/record/2775923.

Qu, H. and Gouskos, L. (2020). ‘Jet tagging via particle clouds’. *Physical Review D* 101 (5), p. 056019.

Racah, E., Ko, S., Sadowski, P., Bhimji, W., Tull, C., Oh, S.-Y., Baldi, P., et al. (2016). ‘Revealing fundamental physics from the daya bay neutrino experiment using deep neural networks’. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 892–897.

Regler, M., Frühwirth, R., and Mitaroff, W. (1996). ‘Filter methods in track and vertex reconstruction’. *International Journal of Modern Physics C* 7 (04), pp. 521–542.

Ren, J., Wu, L., and Yang, J. M. (2020). 'Unveiling CP property of top-Higgs coupling with graph neural networks at the LHC'. *Physics Letters B* 802, p. 135198.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). 'Faster r-cnn: Towards real-time object detection with region proposal networks'. *Advances in neural information processing systems* 28.

Renner, J., Farbin, A., Vidal, J. M., Benlloch-Rodríguez, J., Botas, A., Ferrario, P., Gómez-Cadenas, J. J., Alvarez, V., Azevedo, C., Borges, F., et al. (2017). 'Background rejection in NEXT using deep neural networks'. *Journal of Instrumentation* 12 (01), T01004.

Rose, K. (1998). 'Deterministic annealing for clustering, compression, classification, regression, and related optimization problems'. *Proceedings of the IEEE* 86 (11), pp. 2210–2239.

Rosenblatt, F. (1958). 'The perceptron: a probabilistic model for information storage and organization in the brain.' *Psychological review* 65 (6), p. 386.

Rovere, M., Chen, Z., Di Pilato, A., Pantaleo, F., and Seez, C. (2020). 'Clue: A fast parallel clustering algorithm for high granularity calorimeters in high-energy physics'. *Frontiers in big Data* 3, p. 591315.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). 'Learning representations by back-propagating errors'. *nature* 323 (6088), pp. 533–536.

Rutherford, E. (1911). 'LXXIX. The scattering of α and β particles by matter and the structure of the atom'. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 21 (125), pp. 669–688.

Sakuma, T. and McCauley, T. (2014). ‘Detector and event visualization with SketchUp at the CMS experiment’. *Journal of Physics: Conference Series*. Vol. 513. 2. IOP Publishing, p. 022032.

Salamani, D., Gadatsch, S., Golling, T., Stewart, G. A., Ghosh, A., Rousseau, D., Hasib, A., and Schaarschmidt, J. (2018). ‘Deep generative models for fast shower simulation in ATLAS’. *2018 IEEE 14th International Conference on e-Science (e-Science)*. IEEE, pp. 348–348.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). ‘The graph neural network model’. *IEEE transactions on neural networks* 20 (1), pp. 61–80.

Schmidt, B. (2016). ‘The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments’. *Journal of Physics: Conference Series*. Vol. 706. 2. IOP Publishing, p. 022002.

Selvan, R. S. and Arutchelvan, K. (2021). ‘Abstractive Summarization Using Categorical Graph Network’. *REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS* 11 (4), pp. 1997–2007.

Sjöstrand, T. et al. (2015). ‘An introduction to PYTHIA 8.2’. *Comput. Phys. Commun.* 191, pp. 159–177. DOI: `10.1016/j.cpc.2015.01.024`. arXiv: `1410.3012 [hep-ph]`.

Suzuki, K. (2017). ‘Overview of deep learning in medical imaging’. *Radiological physics and technology* 10 (3), pp. 257–273.

Touranakou, M., Chernyavskaya, N., Duarte, J., Gunopulos, D., Kansal, R., Orzari, B., Pierini, M., Tomei, T., and Vlimant, J.-R. (July 2022). ‘Particle-based fast jet simulation at the LHC with variational autoencoders’. *Machine Learning: Science and Technology* 3 (3), p. 035003. DOI: `10.1088/2632-2153/ac7c56`. URL: `https://doi.org/10.1088/2632-2153/ac7c56`.

Uhrig, J., Rehder, E., Fröhlich, B., Franke, U., and Brox, T. (2018). 'Box2pix: Single-shot instance segmentation by assigning pixels to object boxes'. *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 292–299.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). 'Graph attention networks'. *arXiv preprint arXiv:1710.10903*.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). 'Dynamic graph cnn for learning on point clouds'. *Acm Transactions On Graphics (tog)* 38 (5), pp. 1–12.

Wikimedia Commons (2019). *Standard model of elementary particles.* URL: `https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg`.

Wong, C.-Y. (1994). *Introduction to high-energy heavy-ion collisions.* World scientific.

Yang, H.-J., Roe, B. P., and Zhu, J. (2005). 'Studies of boosted decision trees for MiniBooNE particle identification'. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 555 (1-2), pp. 370–385.

Zhang, B. and Wonka, P. (2021). 'Point cloud instance segmentation using probabilistic embeddings'. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8883–8892.

Zolotarev, V. M. (1986). *One-dimensional stable distributions.* Vol. 65. American Mathematical Soc.

Zou, Z., Shi, Z., Guo, Y., and Ye, J. (2019). 'Object detection in 20 years: A survey'. *arXiv preprint arXiv:1905.05055*.