Deep Learning based Energy Reconstruction for the CALICE AHCAL

Master Thesis

von
Erik Buhmann
geboren am 11. Oktober 1991

Universität Hamburg
Institut für Experimentalphysik

Hamburg
Juli 2019

Gutachter: Jun.-Prof. Dr. Gregor Kasieczka

Institut für Experimentalphysik
Universität Hamburg

Prof. Dr. Erika Garutti

Institut für Experimentalphysik
Universität Hamburg

Eidesstattliche Versicherung

Ich versichere, dass ich die beigefügte schriftliche Masterarbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für alle Informationen, die dem Internet oder anderer elektronischer Datensammlungen entnommen wurden. Ich erkläre ferner, dass die von mir angefertigte Masterarbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung im Rahmen meines Studiums war. Die von mir eingereichte schriftliche Fassung entspricht jener auf dem elektronischen Speichermedium.

Mit einer Veröffentlichung dieser Arbeit erkläre ich mich einverstanden.

Erik Buhmann
Hamburg, der 4. Juli 2019

Zusammenfassung

Zukünftige e^+e^--Linearbeschleuniger bieten die Möglichkeit für präzise Teilchenmessungen und benötigen eine bisher unerreichte Detektorauflösung. Dafür entwickelt die CALICE Kollaboration fein-segmentierte Kalorimeter zur Anwendung des Particle Flow Approach (PFA).

Ein technischer Prototyp für ein Analoges Hadronkalorimeter (AHCAL) basierend auf der SiPM-on-tile Technologie wurde von der CALICE Kollaboration in Betrieb genommen. Er besteht aus 22.000 Kanälen mit jeweils einer Größe von 30 x 30 mm^2. Mit dem AHCAL kann Information in 5 Dimensionen gemessen werden: Der Ort in 3D, die Energie und der Zeitpunkt jedes Kalorimeter-Ereignisses.

Der Prototyp wurde 2018 am Teststrahl des CERN SPS getestet. Die Testläufe mit Pionen und die darauf basierende Monte Carlo Simulation werden genutzt, um die Energieauflösung des AHCAL zu bestimmen. Eine Verbesserung der Energieauflösung gegenüber der Standard-Energierekonstuktion kann durch den Einsatz von Software-Kompensation oder andere Machine Learning Algorithmen erreicht werden.

Tiefe neuronale Netzwerkarchitekturen (DNN) können für die Energierekonstuktion eines kompletten 5D Kalorimeterbildes genutzt werden. Mehrere Architekturen werden vorgestellt und mit der Standardrekonstruktion und lokaler Software-Kompensation verglichen. Diese Netzwerke werden sowohl auf Teststrahl-Daten als auch auf Monte Carlo Ereignissen trainiert.

Ein auf Locally Connected Layern basierendes Netzwerk nutzt Gewichtungsfaktoren für jeden Kalorimeter-Kanal und kann Schauerverluste in der Energiesumme ausgleichen. Weiter wird eine Convolutional Neural Network (CNN) Architektur gezeigt, welche die Energieauflösung im Vergleich zu Software-Kompensation verbessert, aber mit dem Nachteil von Overfitting an den Grenzen des trainierten Energiebereiches. Außerdem wird eine aus beiden Ansätzen zusammengesetzte Netwerkachitektur vorgestellt, welche zusätzlich die Zeit-Information im Event nutzt.

Abstract

Future linear lepton colliders offer the possibility of precision physics studies and demand detectors with an unprecedented resolution. For this purpose the CALICE collaboration is developing highly-granular calorimeters for the application of the Particle Flow Approach (PFA).

An engineering prototype for an analogue hadron calorimeter (AHCAL) based on the SiPM-on-tile technology was assembled by the CALICE collaboration consisting of about 22,000 channels each with a size of 30 x 30 mm^2. Events measured by the AHCAL include 5-dimensional information: the 3D location, energy and timing of a each calorimeter hit.

The prototype underwent test beam at the CERN SPS in 2018. These pion test beam runs and a Monte Carlo simulation of the test beam setup is used to determine the energy resolution of the AHCAL. An improvement over the energy resolution with the standard event energy reconstruction can be achieved by employing a hit energy weighting based on software compensation or other supervised machine learning approaches.

Deep neural network (DNN) architectures can be used to reconstruct the event energy from the full 5D calorimeter image. Multiple architectures are presented and compared to the standard reconstruction and to local software compensation. The networks are trained on either test beam data or Monte Carlo samples.

Neural networks based on locally connected layers with cell-wise hit energy weighting show promising results for shower leakage compensation. A convolutional neural network (CNN) architecture is presented that improves the energy resolution over software compensation, but suffers from overfitting at the boundaries of the trained energy space. A merged architecture of both approaches also including the hit timing is discussed.

Contents

Chapter 1

Introduction

Over the past 50 years the *Standard Model of Particle Physics* has emerged as a successful theoretical framework to describe the fundamental structure of matter. The Standard Model is explored with particle collision experiments at very high energies. Currently the largest particle accelerator experiments are located at the Large Hadron Collider (LHC) at CERN which collides protons with a centre of mass energy of $\sqrt{s} = 13\,\text{TeV}$. The greatest success of the LHC experiments to date was the discovery of the Higgs boson [1, 2] which was predicted in 1964 [3, 4].

Despite the success of the Standard Model there is overwhelming observable proof that physics beyond the Standard Model exists. To explore and understand 'new physics' very precise measurements of particle properties are necessary. Such measurements could be performed at a future linear lepton collider such as the proposed *Compact Linear Collider (CLIC)*. With CLIC electrons and positrons could be collided with a centre of mass energy of up to $\sqrt{s} = 3\,\text{TeV}$. To achieve the best possible measurement precision, an unprecedented detector resolution is required, i.e. a jet energy resolution of $\sigma_E / E \approx 3.5\,\%$ above 100 GeV [5].

Such a resolution can be achieved with the *Particle Flow Approach (PFA)* to calorimetry [6]. For successful application of PFA a highly-granular calorimeter is necessary. These calorimeters are subject of studies in the *Calorimetry for Linear Collider Experiment (CALICE)* collaboration. A engineering prototype for an analogue hadron calorimeter (AHCAL) was assembled by CALICE in 2018. This prototype is based on the *SiPM-on-tile* technology and underwent multiple test beam campaigns in 2018 and 2019.

In this thesis the energy resolution of the AHCAL prototype for pions is evaluated and algorithms are compared to improve the energy resolution. In past studies by the CALICE collaboration with previous prototypes, *software compensation* algorithms were successfully applied to improve the detector resolution by utilizing a few event parameters [7].

Modern deep learning approaches use all available information in a given data set, i.e. convolutional neural networks are trained on images and are very successful in classifying objects [8]. In a highly-granular calorimeter all measured information of an event can be represented as a 5-dimensional image. These images include the location in 3D, energy and timing of a calorimeter hit. This thesis discusses the application of deep neural network architectures to event energy reconstruction to improve the energy resolution of the AHCAL. Multiple network architectures are explored and

applied to both test beam data and Monte Carlo simulation. A comparison with software compensation is made.

This thesis is structured as follows. In chapter 2 a basic introduction to high-energy physics is given. The Standard Model of Particle Physics as well as collider experiments and detectors are introduced. In chapter 3 the basics of the interaction between particles and matter in calorimeters are explained and particle shower development and their calorimeter response are outlined. The CALICE AHCAL engineering prototype is presented in chapter 4. This chapter focuses in particular on the data quality of the test beam campaign in May 2018. A basic introduction to machine learning and neural networks is given in chapter 5. The results of this thesis are presented in chapter 6. The results are divided into network trainings on data and on Monte Carlo samples, for which results are compared to software compensation. In chapter 7 the findings of this thesis are summarized.

Chapter 2

Introduction to High-Energy Physics

Elementary particle physics is the study of the fundamental structure of matter. Currently, the best theory for our understanding of matter and three of the four fundamental forces is the *Standard Model of Particle Physics*. Apart from experiments with cosmic rays, colliding particles in particle accelerators have emerged as a very important tool for physicists to explore and validate the predictions of the Standard Model over the past decades. These experiments are undertaken with large accelerators and modern particle detection systems.

This chapter is giving a brief introduction to the Standard Model as well as collider experiments and corresponding detectors.

2.1 The Standard Model of Particle Physics

The Standard Model of Particle Physics is a theoretical framework that classifies all known elementary particles and their interactions. It emerged in the 1960s and 70s from the theory of Quantum Electrodynamics (QED) [9], the Glashow-Weinberg-Salam (GWS) theory of electroweak processes [10, 11] and the theory of Quantum Chromodynamics (QCD) [12].

The Standard Model includes 12 elementary particles with spin $\frac{1}{2}$ that follow the Fermi-Dirac statistics. These particles form all known matter and are called *fermions*. The fermions are divided into two families: *leptons* and *quarks*. Each is categorized in three groups ordered by increasing particle mass. In addition to those 12 particles there is another set of 12 anti-particles each corresponding to its counterpart with the same mass, but opposite quantum numbers. This leads to a total of 24 fermions in the Standard Model.

The leptons are categorized by their electric charge (0, $\pm$1) and their lepton numbers (electron, muon, or tau number). The negatively charged leptons are the *electron* (e^-), the *muon* (μ^-) and the *tau* (τ^-) and their positive counterparts are called *positron* (e^+), *anti-muon* (μ^+) and *anti-tau* (τ^+). The electrically neutral leptons are the *neutrinos* (ν_e, ν_μ and ν_τ) and the *anti-neutrinos* ($\overline{\nu}_e$, $\overline{\nu}_\mu$ and $\overline{\nu}_\tau$) each classified to their respective lepton family. While the electron and the neutrinos (and their anti-particles) are stable, the muons and taus decay with a lifetime of 2.2 x 10^{-6} s and 2.9 x 10^{-13} s respectively.

The quarks are categorized into up-type and down-type. The up-type quarks are the *up (u)*, *charm (c)* and *top (t)* and carry an electrical charge of $+\frac{2}{3}$. The down-type quarks are the *down (d)*, *strange (s)* and *bottom (b)* with an electrical charge of $-\frac{1}{3}$. Their anti-quarks carry the charges $-\frac{2}{3}$ and $+\frac{1}{3}$. Each quarks comes in one of three *colours*: (anit-)red, (anti-)green and (anti-)blue. Only colourless bound states of two or three quarks have been discovered. These bound states form together particles called *hadrons*. Those bound hadrons are divided into two groups: *mesons* and *baryons*. A meson consists of two quarks with corresponding colour and anti-colour. The baryon consists of three quarks with each different colour or anti-colour.

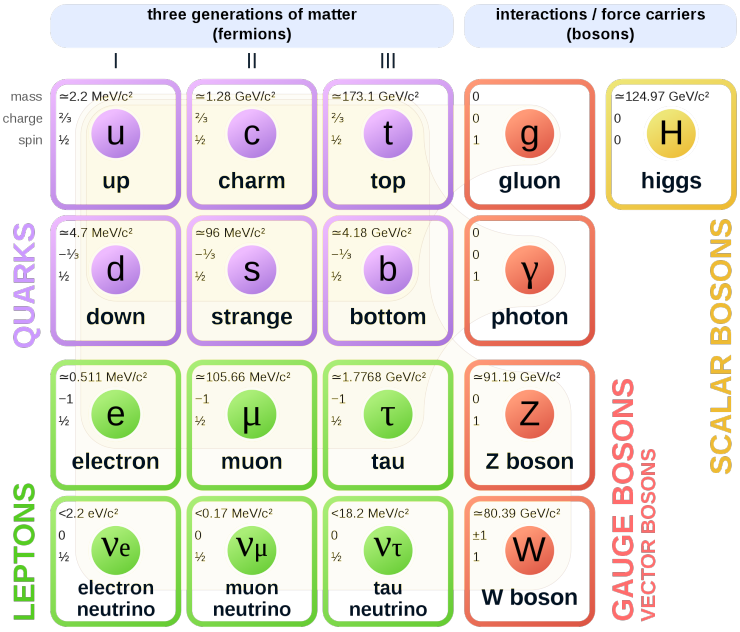


Figure 2.1: The elementary particles of the Standard Model, consisting of the 12 fundamental fermions and four fundamental gauge bosons and the Higgs boson. Brown loops describe which gauge bosons (red) couple to which fermions (purple and green). The Higgs boson couples to all massive particles shown. Adapted from [13].

Another set of elementary particles in the Standard Model are the *Gauge bosons* with a spin 1. These Gauge bosons are acting as force carriers for the three fundamental interactions considered in the Standard Model: the *electromagnetic interaction*, the *weak interaction* and the *strong interaction*. The fourth fundamental interaction, gravity, is not described by the Standard Model, which is one of the limitations of this theory.

The electromagnetic force acts on any particle with an electric charge and is mediated by the massless *photon* (γ). The weak interaction is transmitted by two charged bosons, the W^+ and W^-, and a neutral boson, the Z^0. The reach of this interaction is limited as these bosons are massive with $m_{W^\pm} \approx 80.4\,\text{GeV}$ and $m_{Z^0} \approx 91.2\,\text{GeV}$. The strong interaction acts on any particle with a colour charge and is mediated by eight gluons that carry both colour and anti-colour and are massless. All of these interactions obey several conservation law such as conservation of energy, momentum, angular momentum, charge, colour, quark number, flavour (except for the weak interaction) and lepton number.

The final elementary particle is the *Higgs boson (H)*, which is a spin 0 particle with no colour or electric charge and a mass of $m_H \approx 125.09\,\text{GeV}$. [14]. This particle is produced by the excitation of the Higgs field, introduced by the Higgs mechanism [3, 4], and gives mass to the particles of the Standard Model. An overview of the elementary particles and the interactions is shown in figure 2.1.

2.2 Future Collider Experiments

The experimental discovery of all the particles described above, including the Higgs boson in 2012 [1, 2], have made the Standard Model to one of the most successful theories in physics. The common way for physicists to produce and study elementary particles is with accelerator experiments in which particles are accelerated to very high energies and smashed at a target (fixed target experiment) or at another accelerated particle beam (collider experiments). The particles produced in these experiments are limited by the centre of mass energy of the collision as the produced particle has to have a lower rest mass energy, leading to the popularity of collider experiments and the development of more and more powerful accelerators in recent decades.

There are two types of particle accelerators: linear and circular ones. The most powerful circular accelerator to date is the *Large Hadron Collider (LHC)* at CERN colliding protons and heavy ions with a current centre of mass energy of $\sqrt{s} = 13\,\text{TeV}$ in a synchrotron ring with a circumference of 26.7 km. The LHC was designed to explore the Standard Model up to until previously unreached energies and search for physics beyond the Standard Model. The Higgs boson was discovered by the CMS and ATLAS experiments at the LHC [1, 2]. However, as protons consist of quarks and gluons there is a limitation to the precision of measurements in proton-collisions at the LHC. The proton constituents are the actual colliding objects and their exact centre of mass energy is uncertain. Another limitation arises from the strong interaction between theses particles that result in a large QCD background.

Without particles consisting of quarks, the electrons (and positrons) are very good candidates for precision collider experiments as they are stable and muons and taus have a short mean lifetime. In an e^+e^--collider the initial states of the particles are well defined and the precise energy and spin-orientation (in polarized beams) are known. With less background this allows for very precise measurements. The most powerful circular lepton collider was the *Large Electron-Positron Collider (LEP)* at CERN that was operated between 1989 and 2000 with a centre of mass energy of up to 209 GeV. It was not possible to reach a higher beam energy with the given accelerator cavities as the synchrotron radiation emitted by electrons is a limiting factor in the circular accelerators.

The energy loss through synchrotron radiation is proportional to

$$\Delta E \propto \frac{E^4}{r \cdot m_0^4} \tag{2.1}$$

with the E and m_0 the energy and rest mass of the accelerated particle and r as the radius of the circular accelerator. Hence for electrons as very light particles the radius of the collider would need to increase further to reach very high energies. For heavy particles such as protons at the LHC the energy loss trough synchrotron radiations is less of a limitation to the reachable centre of mass energy.

For practical reasons it is likely that the next large scale lepton collider will be a linear collider, as for linear colliders the synchrotron radiation is negligible and the reachable energy increases with its length. This way precise measurements could be done complementary to experiments at the LHC to explore the Standard Model further and physics beyond. A current study for such a high energy linear collider is the *Compact Linear Collider (CLIC)* possibly located at CERN with a centre of mass energy up to 3 TeV [5].

2.3 Detectors for Future Linear Colliders

The detectors for the experiments at high-energy colliders are designed to identify and measure the particles produced at the collision. Each individual collision produces an *event* at a specific *interaction point*. The identification of the particles produced in each event by measuring their four-momenta are the subject of studies in high-energy physics. Large particle detectors are covering most of the 4π area around the interaction point. These modern detectors consists of several layers of sub-detectors each designed to perform specific measurements. A schematic design of such a detector for CLIC can be found in figure 4.1 of chapter 4.

The innermost region of the detector closest to the interaction point is occupied by the *vertex detector*, which has a fine resolution to resolve short-lived secondary vertices originating from the primary vertex. A *tracking detector* is used to measure the charge and momentum of charged particles. For this purpose it is placed in a strong magnetic field as the paths, or *tracks*, of the produced charged particles are directed perpendicular to the magnetic field. The charge and momentum of the particle can be calculated from the direction and radius of the bent track.

The following sub-detectors are the *electromagnetic calorimeter (ECAL)* and the *hadronic calorimeter (HCAL)*, which are optimized to measure the energy of electrons and photons and hadrons respectively. The particles deposit energy in the calorimeters, hence this is a destructive measurement. Further detector parts are positioned as outer layers to detect muons and energy leaked though the calorimeter.

The energy of particles that leave no response in the detector can be calculated from *missing momentum* $\vec{p}_{miss}$. The $\vec{p}_{miss}$ can be calculated as the the total momentum of the colliding leptons is well defined (except for effects such as beamstrahlung).

To utilize the full potential of a future lepton collider all event structures need to be measured with great detail. This requires a calorimeter system that can measure energies with a very fine resolution. A concept for such a calorimeter and corresponding event reconstruction algorithms are presented in the following chapters.

Chapter 3

Calorimetry

In particle physics calorimetry refers to the destructive measurement of a particles' energy by its absorption in matter. When a particles interacts with matter, the particle deposits its energy via electromagnetic or (in case of hadrons) via hard-/hadronic interactions. This process is called shower development and is discussed in section 3.1 by introducing both electromagnetic and hadronic showers. The two different types of calorimeters, namely homogeneous and sampling calorimeter, are presented in section 3.2. The response to electromagnetic and hadronic showers of either calorimeter is discussed in section 3.3. The calorimeter response to electromagnetic and hadronic showers leads to different energy resolutions, explained in section 3.4. Lastly an introduction to the particle flow approach (PFA) to calorimetry is given in section 3.5.

3.1 Particle Showers Development

Particles interact with matter in the calorimeter. By depositing energy, the particles are creating cascades of more particles. Those cascades of particle multiplication are called *particle showers*. We differentiate showers by their kind of fundamental interaction. Electrons and photons interact with matter only via the electromagnetic force and therefore these particle develop *electromagnetic showers*. Showers that originate from particles interacting via the strong nuclear force, such as protons or pions, are called *hadronic showers*.

3.1.1 Electromagnetic Showers

Electromagnetic Showers are started by electrons (positrons), photons or neutral pions that interact with matter via the electromagnetic force. Photons interact due to the photoelectric effect, due to Compton scattering or due to pair production, the latter dominating at energies above 1 MeV.

Electrons (positrons) are interacting with matter mostly via ionization and radiation (Bremsstrahlung) although processes such as Møller scattering, Bhabha scattering and e+ annihilation contribute, too. At energies above 10 MeV radiation dominates, below ionization does. The contribution of the different interactions to the energy loss of an electron in matter are shown in figure 3.1.

At the *critical energy* both processes, ionization and radiation, are contributing

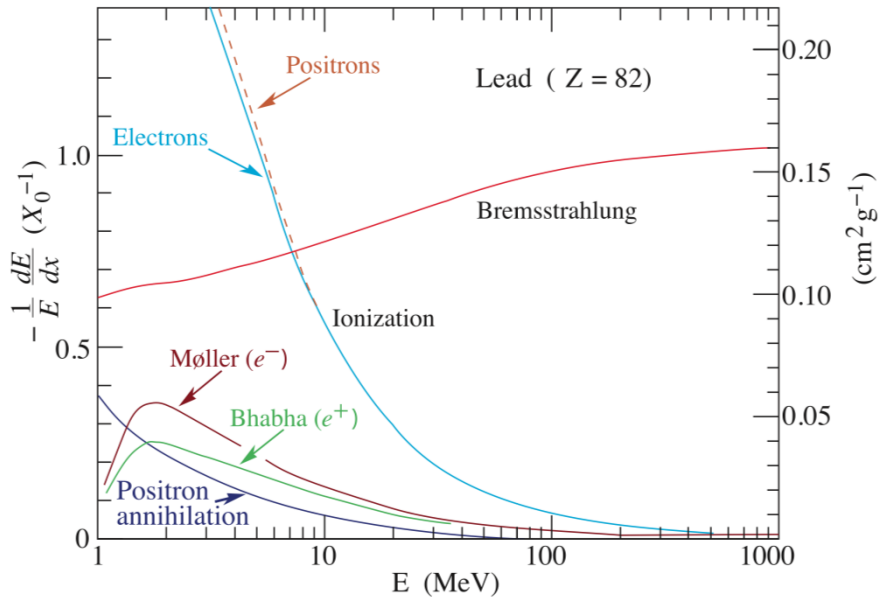


Figure 3.1: Fractional energy loss per radiation length in lead as a function of electron energy. The different interaction types of electrons with matter are shown. [15]

equally on average. This energy can be parametrized with [16]

$$E_{cirt} = \frac{610\ \text{MeV}}{Z + 1.24} \tag{3.1}$$

for material in the liquid or solid state. The critical energy for iron is around 21 MeV.

Electrons and photons with an energy above 1 GeV initiate electromagnetic showers. Electrons lose their energy by radiation emitting on average one photon per *radiation length* (see below). This radiated photon will undergo pair production starting a cascade of particle multiplication. This is shown in as a simplified illustration in figure 3.2. This cascade of e^-e^+ and γ production continues until the energy of the original particle is fully deposited inside the calorimeter material (or until parts of the shower are leaked outside).

The depth of the shower at which the number of particles newly produced in the cascade reaches a maximum is called *shower maximum*. It increases logarithmically with the energy of the electron triggering the particle multiplication cascade.

Two scaling factors are describing the electromagnetic shower development above 1 GeV, called *radiation length* and *Molière radius*. The *radiation length* (X_0) describes the longitudinal shower development. For electrons it is defined such that over 1 X_0 of material the energy loss equals on average $(1 - e^{-1})$ = 63.2 %. For photons the radiation length has a different meaning and is related to the *mean free path*. It is given by [16]

$$\lambda_\gamma = \frac{9}{7} X_0 \tag{3.2}$$

The different definitions of the radiation length is due to the fact that high-energetic electrons are starting to radiate immediately once they encounter the material. Photons however do not necessarily interact in the same amount of material, hence the relation to λ_γ.

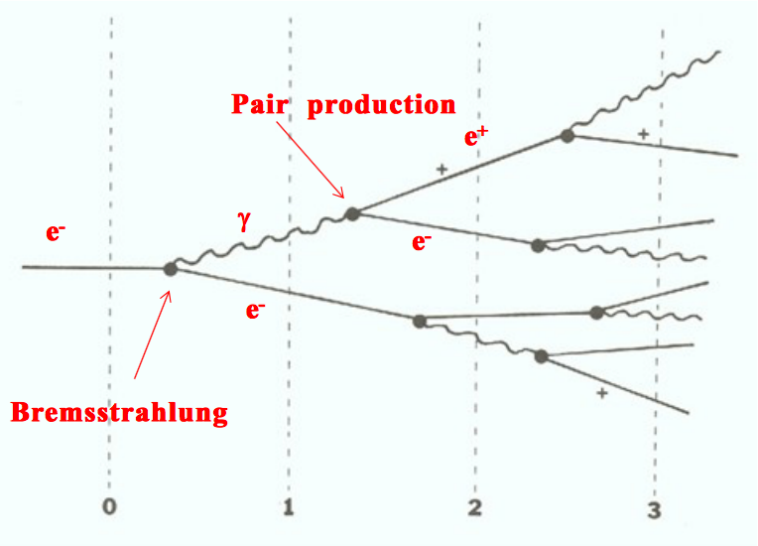


Figure 3.2: Illustration of a particle multiplication cascade initiated by an electron entering a block of matter. The electron interacts with the material and radiates photons due to Bremsstrahlung. The photons on the other hand can produce electron-positron due to pair production. The x-axis represents the shower depth in units of radiation lengths X_0. [17]

The *Molière radius* (ρ_M) describes the lateral / transverse shower development and is defined by the ratio between the radiation length and the critical energy:

$$\rho_M \propto \frac{X_0}{E_{crit}} \tag{3.3}$$

Therefore ρ_M is less material dependent than X_0 as one can express their scaling via

$$X_0 \propto \frac{A}{Z^2} \text{ and } \rho \propto \frac{A}{Z}. \tag{3.4}$$

This lateral development is on the one hand due to multiple scattering of electrons, and on the other hand due to the production of photons and electrons in isotropic processes such as Compton scattering or the photoelectric effect.

The longitudinal shower development differs in high-Z and low-Z materials as well as in different energy scales. The critical energy decreases for high-Z materials implying that the process of particle multiplication continues till lower energies. This leads to longer shower extensions in high-Z materials as demonstrated in the bottom plot of figure 3.3. The difference between shower containment of electron and photon induced showers in the same material can be understood by the relation between X_0 and the mean free path λ_γ. In addition, one should be aware that the scaling of X_0 and ρ_M is only applicable for energies above the critical energy and the part of the energy deposited by shower particles below the critical energy is not following the same depth dependence. [18]

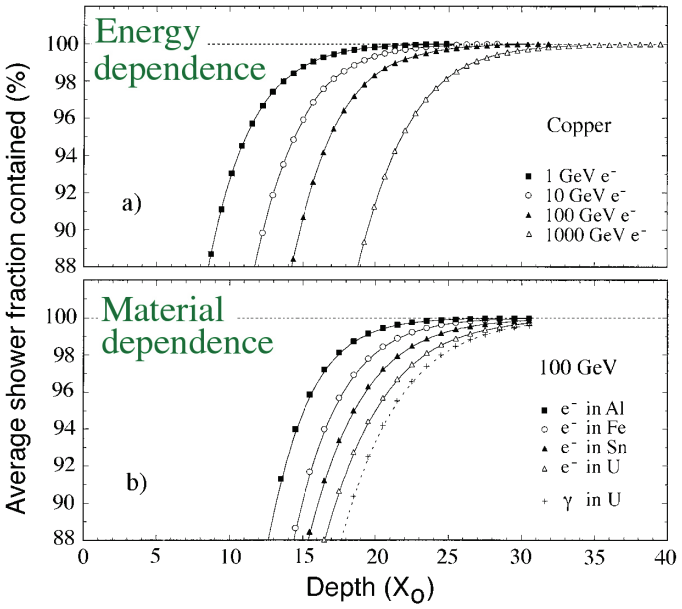


Figure 3.3: Energy (a) and material (b) dependence of the average shower fraction contained as a function of the radiation length X_0. Results of EGS4 calculations. [16]

Interactions of Charged Heavy Particles

The critical energy, at which the energy loss from radiation and ionization is equal, scales with $(m/m_e)^2$ with m as the incoming particles' mass. For a muon this gives a scaling factor of $(m_\mu/m_e)^2 = 40,000$. Hence for muons and other charged particles heavier than the electron the critical energy is much higher and up to larger energies ionization is dominant. For muons ionization dominates in the mean energy loss for energies below 100 GeV in all absorber materials. [18]

The mean rate of energy loss by charged heavy particles for an intermediate energy range (10 MeV - 100 GeV) in intermediate-Z materials is given by the *Bethe-Bloch equation*: [15]

$$\left\langle -\frac{dE}{dx} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 W_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right] \tag{3.5}$$

Here K equals a constant $4\pi N_A r_e^2 m_e c^2$, z is the charge number of the incident particle, W_{max} is the maximum energy transfer in a single collision, I is the mean excitation energy, $\delta(\beta\gamma)$ is a correction term for the *density effect*, and β and γ are the particle velocity and the Lorentz factor.

The equation applied for muons entering copper is shown in the 'Bethe' region of figure 3.1 which shows the mean energy loss as a function of $\beta\gamma = p/Mc$. Above 100 GeV radiative interactions dominate, a discussion on the low energy corrections to the Bethe-Block equation in the region below 10 MeV can be found in [15]. Particles with an energy close to the minimum of this curve are called *Minimum Ionizing Particles*, short *MIP's*. For muons this applies to an energy around 2 -3 GeV.

The probability distribution of the energy loss in thin absorber materials such as scintillators is described by a Landau(-Vavilov) distribution. Compared to a Gaussian

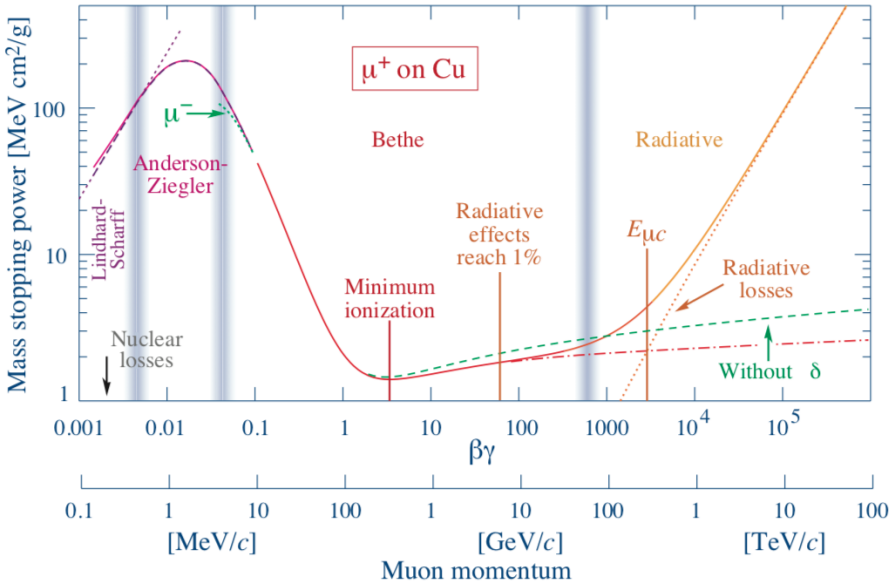


Figure 3.4: Mass stopping power for positive muons in copper as a function of $\beta\gamma = p/Mc$. The x-axis is separated into regions for different models. [15]

distribution the Landau distribution is highly-skewed with a long tail towards higher values. Therefore the most probable values (MPV) for the energy loss is quite different from the mean energy loss calculated with the Bethe-Bloch equation. The tail is due to rare collisions with atomic electrons that lead to a high-energy transfer. Due to this rare events, the mean fluctuates strongly and is experimentally hard to measure. Hence for energy deposition scaling the most probable value is used.

3.1.2 Hadronic Showers

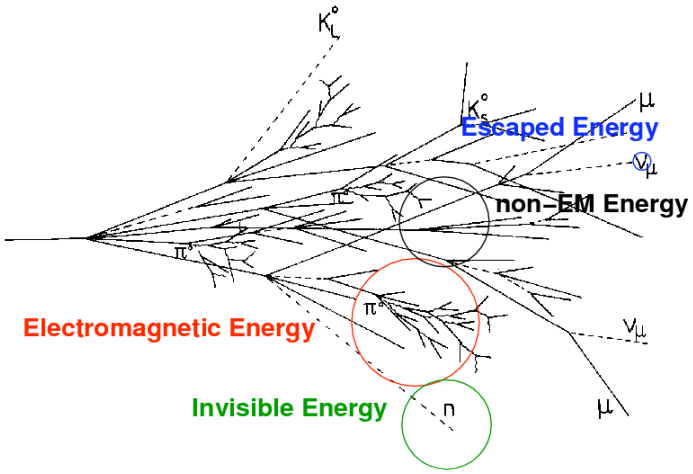


Figure 3.5: Illustration of a hadronic shower. Its different energy components are highlighted and labelled. Detailed explanations can be found in the text. [19]

Hadronic showers are initiated by charged or neutral hadrons interacting with matter. In addition to the electromagnetic force, the strong interaction plays an important role in hadronic showers. The energy deposition of the hadronic shower can be categorized into different energy components shown in figure 3.5. A rough difference can be made between the electromagnetic and the hadronic part of the shower.

The *electromagnetic energy* component behaves like the electromagnetic shower described previously. It is mainly due to neutral pions that decay into two photons producing an electromagnetic cascade. These neutral pions have a very short lifetime in the order of 10^{-7} s. The fraction of the electromagnetic energy on the whole shower energy increases on average with the incident particles' energy, but can vary significantly on an event by event basis. This energy dependence leads to a non-linear calorimeter response for the electromagnetic fraction in ratio to the hadronic fraction of the shower.

The *non-EM energy* corresponds to the energy deposited through the strong interaction and in nuclear interactions such as spallation and ionization. Such protons released by nuclear spallation carry about 50 - 100 MeV per particle.

The *invisible energy phenomenon* refers an energy fraction that is 'lost' trough nuclear binding energy. The energy needed for the nuclear interaction between a hadron and a nucleus releasing protons and neutrons is not contributing to the signal of the calorimeter and is hence 'invisible' in the detector response. The amount of invisible energy can fluctuate largely between showers of same energy which limits the resolution of hadronic calorimeters. Furthermore this phenomenon leads to hadronic calorimeters in general being non-compensating. Compensating calorimeters detect equal amounts of energy of the hadronic and electromagnetic shower fraction, which is not the case when a part of the hadronic energy cannot be detected as this phenomenon does not apply to the electromagnetic part.

The *escaped energy* is defined as energy deposited in neutrino production which cannot be detected either.

A hadronic shower is developing on a different length scale due to the difference between nuclear and electromagnetic cross-sections. In general a hadronic shower is spread more in longitudinal as well as in lateral direction. The scaling is given by the *nuclear interaction length*, short λ_{int}. It is defined by the mean distance or mean free path a hadron is travelling before having lost $(1 - e^{-1})$ of its energy due to nuclear interaction. λ_{int} is generally much larger than X_0 and is given by [20]

$$\lambda_{int} = \frac{A}{N_A \, \rho \, \sigma_{inel}} \approx 35 A^{1/3} \, \mathrm{g/cm^2} \tag{3.6}$$

with ρ begin the material density and σ_{inel} the inelastic cross section. Usually the inelastic proton cross section is used, neglecting energy and particle type dependence on incoming particle. For iron λ_{int} is about 132 g/cm^2 or 17 cm. The depth of the shower increases logarithmically with the energy just as it does for electromagnetic showers. However, shower leakage can vary significantly due to fluctuations in the shower development. The lateral shower development on the other hand has an inverse relationship with the energy. Since the electromagnetic fraction of the shower scales with energy, a shower at a higher energy is contained in less transverse material on a 99 % level as the electromagnetic shower is more compact.

Not all the components of the hadronic shower fraction develop on an relativistic time scale as the electromagnetic fraction does. The thermal neutrons have the most important impact on time structure inside the hadronic shower as they can travel several microseconds to seconds before generating a signal in the calorimeter.

3.2 Calorimeter Types

There are two categories of calorimeters: Sampling and homogeneous calorimeters. The terminology refers to its configuration. A *homogeneous calorimeter* consists of only *active* material in which particles can generate a measurable signal. The active materials are often scintillating crystals, lead loaded glass or noble gas. These calorimeters can achieve a very good single particle energy resolution as all energy deposited inside the calorimeter contributes to the signal.

A *sampling calorimeter* on the other hand is build with layers of active and *passive* material in alternating order. The passive material acts only as absorber and the energy deposited does not generate a signal response. Using a dense absorber with a high atomic number allows for a more compact design as the shower produced are shorter. Examples for absorber materials include steel, tungsten, copper, lead, or uranium. Various approaches can be made to the exact geometric configuration and the implementation of read-out electronics for the active material. As the energy deposited in the passive material does not contribute to the signal a large fraction of the shower energy is not detected which results in sampling fluctuations and a worse energy resolution. The frequent material transition of particles inside the calorimeter adds another layer of complexity. However, the segmentation allows for a layer-specific signal read-out which can be a big advantage depending on the experiment and reconstruction algorithm used.

As electromagnetic showers are governed by the radiation length and the hadronic one by the nuclear interaction length there is a difference in the length scale of their longitudinal development. Hence in most major experiments the calorimeter is roughly split into two parts each optimized to detect a respective particle type. The electromagnetic calorimeter, the *ECAL*, is used to measure electromagnetic showers while the hadronic calorimeter, the *HCAL* is optimized for measuring hadronic showers. Optimization can include the choice of active and passive material as well as the geometry and read-out electronic.

3.3 Calorimeter Response

Due to instrumental constraints only a part of the energy deposited in the calorimeter is actually measured as a signal. In this section a difference is made between the response to electromagnetic showers and hadronic showers. As mentioned before their interaction and energy deposition are governed by different principles which makes an individual discussion necessary.

In general, the calorimeter response is defined as the ratio of average calorimeter signal to a unit of deposited energy. [16] In the practical case of a scintillating crystal as active material this could be a number of photos per GeV. If the calorimeter response

is constant, one speaks of a *linear* calorimeter. This implies that the measured signal is proportional to the incident particles energy.

A useful metric to give the precision of the calorimeter is given by the ratio of the width σ and the mean energy $\overline{E}$ of a signal distribution. This ratio $\sigma/\overline{E}$ is called the *energy resolution*. A lower ratio is regarded as a better resolution. [21]

3.3.1 Electromagnetic Response

Electromagnetic showers in homogeneous calorimeters can be measured with a high resolution as the whole energy of the incident particle deposits energy with processes that create a signal in the active material. Therefore in general electromagnetic calorimeters are linear. However, non-linearities might be due to experimental constraints such as shower leakage or saturation effects in the photodetectors and readout electronics. Fluctuations in the shower development are described by Poissonian statistics. Hence the width of the signal distribution σ is proportional to $\sqrt{N_{visible}}$ for $N_{visible}$ particles measured in the shower. Since with increasing shower energy the number of particles in a shower N_{shower} increases and for homogeneous calorimeters $N_{visible} \approx N_{shower}$, a more precise measurement is possible for higher particle energies. This means for higher energy the resolution of a calorimeter $\sigma/\overline{E}$ decreases. The fluctuations are particle type dependent, so a calorimeter resolution has to be specifically given for a particle type such as electrons, pions (hadrons) or muons.

In sampling calorimeters $N_{visible}$ does not equal N_{shower} as the particles absorbed in the passive material do not contribute to the measured signal. Only a fraction $f_{sampling} = E_{visible}/E_{shower}$ of the total shower energy is observed and contribute to the calorimeter signal. This fraction is called the *sampling fraction* $f_{sampling}$ and the fluctuations in $N_{sampling}$ are refereed to as *sampling fluctuations*. In general, the sampling fluctuations should dominate other sources of statistical fluctuations such as signal quantum fluctuations (i.e. photo-electron statistics), fluctuations in the shower leakage or fluctuations from electronic noise and other instrumental effects. These sampling fluctuations are behaving in accordance with the Poissonian statistics and lead to an energy uncertainty that can be written as $\sigma_{sampling} = \sqrt{d/f_{sampling} \cdot E}$ with d as the thickness of the active material in mm. The $f_{sampling}$ is often defined by the calorimeter response to *minimum ionising particles (MIPs)*. [16]

3.3.2 Hadronic Response

The calorimeter response to hadronic particles is more complicate that the one to EM particles as the energy deposited in the calorimeter has multiple components, measurable ones, such as the EM component and the purely hadronic, and not measurable ones, like the invisible energy.

The energy components and their fractions measured are schematically illustrated in figure 3.6. The calorimeter response to hadronic showers π can be parametrized as

$$\pi = f_{em} \cdot e + (1 - f_{em}) \cdot h \tag{3.7}$$

with e as the calorimeter response to the electromagnetic fraction of the hadronic shower, with h as the response to the hadronic one and with f_{em} as the electromagnetic fraction of the shower. The calorimeter response π is non-linear. One reason for

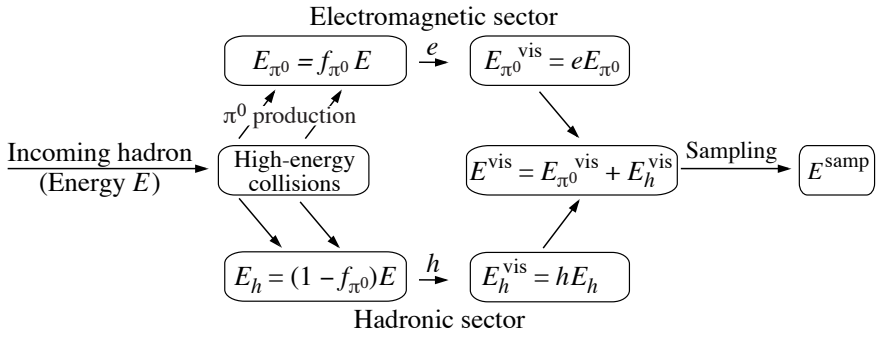


Figure 3.6: Schematic illustration of the energy flow in a sampling hadron calorimeter separating the hadronic and electromagnetic sector. Detailed explanation in the text. [22]

the non-linearity is that f_{em} increases with energy. It also arises due to the invisible energy phenomenon. In most hadronic calorimeters $h < e$ is true as a part of the hadronic energy is 'invisible' to the calorimeter. When $e = h$ a calorimeter is called *compensating*. Most hadronic calorimeters are *non-compensating* as the response to the electromagnetic and hadronic shower fraction is not the same. If $e/h > 1$ the calorimeter is called *undercompensating*, for $e/h < 1$ it is called *overcompensating*. Response distribution of the measurable electromagnetic and hadronic shower fraction for an undercompensating are schematically shown in figure 3.7. With a sampling calorimeter it is technically possible to design a compensating calorimeter by fine tuning the material and geometric setup.

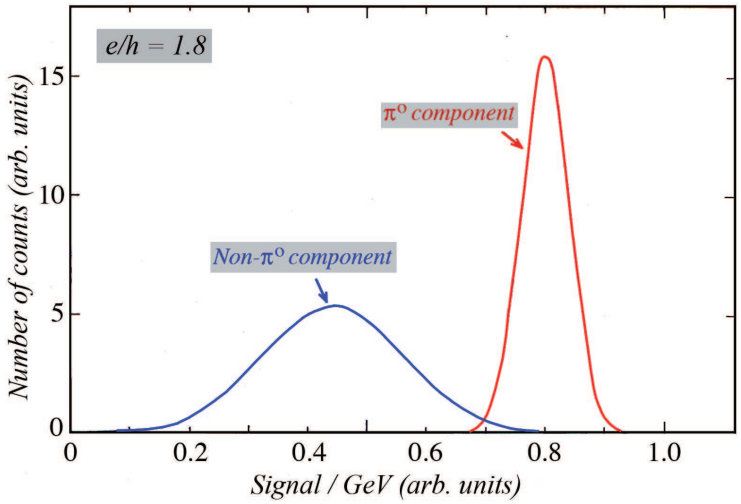


Figure 3.7: Response functions for the hadronic (non-π_0) and the electromagnetic shower fraction (π_0) in an undercompensating calorimeter. The ratio of the mean of the distributions give $e/h = 1.8$. [16]

3.4 Energy resolution

The energy resolution of a calorimeter can be parametrized in the following way:

$$\frac{\sigma}{E} = \sqrt{\left(\frac{a}{\sqrt{E}}\right)^2 + (b)^2 + \left(\frac{c}{E}\right)^2} = \frac{a}{\sqrt{E}} \oplus b \oplus \frac{c}{E} \tag{3.8}$$

The first term is called the *stochastic term*, the second one is the *constant term* and the third one is the *noise term*. The energy E is usually given in GeV and a, b and c are free parameters usually cited in percent. The terms are added in quadrature as their origins are uncorrelated. The terms origin will be explained in the following.

The *stochastic term* is due to intrinsic statistical shower fluctuations and sampling fluctuations that follow the Poissonian statistics and is therefore energy dependent. For electromagnetic calorimeters a is of the order of 10 %. For hadronic calorimeters fluctuations affect the resolution much stronger resulting in an a in the order of 60 %. A very good energy resolution was achieved in the ZEUS compensating hadron calorimeter with $a = 35\,\%$ [23].

The *constant term* originates from instrumental effects such as inhomogeneities in the detector layout, dead active material or calibration uncertainties. All these effects scale with energy and dominate at high energies with b in the order of a few percent.

The *noise term* describes the uncertainty of a measurement due to electronic noise in the read-out. The term dominates in the low energy region. For some calorimeters one assumes the noise to be neglectable and sets $c = 0\,\%$ which results in

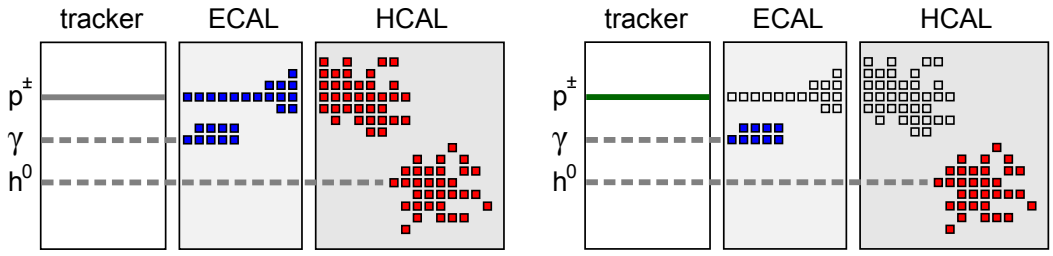
$$\frac{\sigma}{E} = \frac{a}{\sqrt{E}} \oplus b \tag{3.9}$$

3.5 Particle Flow Calorimetry

In detector experiments using traditional calorimetry a particles energy is usually obtained from summing up all energy depositions in the ECAL and the HCAL. The resolution of such a calorimeter can be described with equation 3.8. To improve the energy resolution the *particle flow approach (PFA) to calorimetry* can be applied. In a large detector experiment this algorithm combines information of the tracker with measurements of both the ECAL and the HCAL. The basic concept is that with pooling the measurements the algorithm can use for every particle the detector part it was best measured in. While with conventional calorimetry the informations are separately analysed, an analysis with PFA uses clustering algorithms to separate particles and analyse them particle-wise, not calorimeter-wise.

Measurements at LEP showed that a typical jet deposits energy as about 60 % charged particles (mainly hadrons), as about 30 % photons and about 10 % neutral hadrons [6]. This means approximately 70 % of the jet energy is measured by the hadron calorimeter which incidentally has a worse energy resolution compared the ECAL or the tracker (see section 3.4). With PFA the HCAL would only be used to measure the energy of neutral particles, so only of 10 % of the shower energy increasing the energy resolution significantly. The charged particles would be measured with the tracker and the photons in the ECAL. The PFA concept is schematically shown in figure 3.8.

A limitation of PFA is the *confusion term* that has to be added to the energy resolution equation 3.8. The confusion term increases with beam energy as particle showers increase in size and overlap. This makes it harder for a clustering algorithm



(a) Traditional Calorimetry **(b)** Particle Flow Calorimetry

Figure 3.8: The concept of Particle Flow Calorimetry. In traditional calorimetry 3.8a each calorimeter part is analysed individually while in particle flow calorimetry the particle showers are clustered and tracking information is used to reconstruct the energy of charged hadrons, the ECAL is photons and the HCAL for neutral hadrons. [24]

to determine which calorimeter hit originated from which particle or jet. Hence this uncertainty is introduced as confusion.

To achieve a efficient clustering a highly granular calorimeter is necessary in which small calorimeter cells can be read out individually. For this purpose the CALICE collaboration is developing highly granular calorimeter to be used in future detectors that are optimized to make use of PFA. The next chapter 4 introduces this calorimeter technology.

Chapter 4

The CALICE Analogue Hadron Calorimeter

For the detector of a future linear collider, such as the concept for the detector of the *Compact LInear Collider (CLIC)*, calorimeters are developed that are highly-granular to use the particle flow approach (see section 3.5). A model design for such a CLIC detector is shown in figure 4.1. The detector aims to achieve a jet energy resolution of $\sigma_E / E \approx 3.5\,\%$ above 100 GeV and a time resolution of about 1 ns allowing for the separation of W and Z boson candidates of about 2.5σ in hadronic decays. [5]

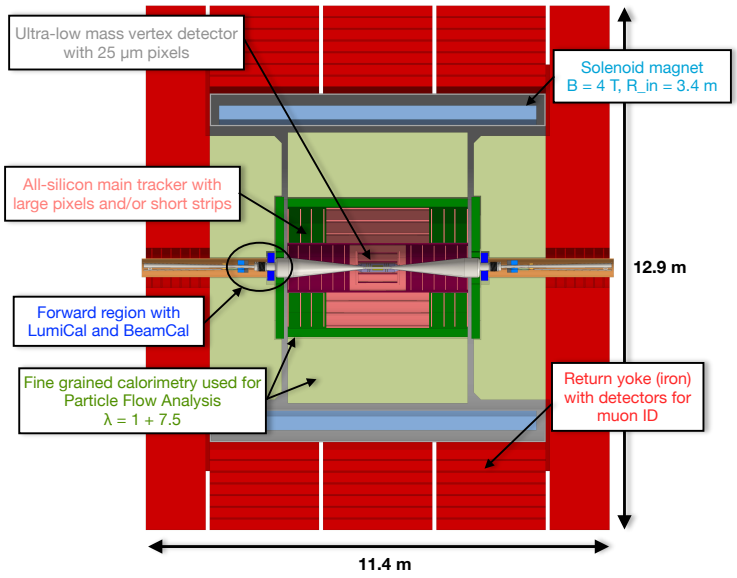


Figure 4.1: CLIC detector concept; top view. The hadron calorimeter is shown in light-green. [5]

A candidate for a HCAL of such a detector (in the figure in light-green) is presented in this chapter. First the CALICE AHCAL technology and the current prototype is introduced in section 4.1. Details of the calibration for such a detector are highlighted. In section 4.2 the test beam campaigns the prototype underwent so far are explained. A focus is lied on the test beam in May 2018 at CERNs' SPS. Furthermore the quality of

the data is checked and the Monte Carlo simulation of that test beam setup introduced. In section 4.3 different ways to calculate the energy resolution with the AHCAL data are discussed.

4.1 CALICE AHCAL Calorimeter Concept

CALICE (*CAlorimeter for LInear Collider Experiment*) is an international R&D collaboration whose members develop calorimeter technologies for detectors in future linear collider experiments. A focus of the collaboration are highly-granular calorimeters for application of the particle flow approach (see 3.5). The technologies explored include various approaches for electromagnetic and hadronic calorimeters such as Analogue Calorimeters and (Semi-)Digital Calorimeters. In the following the *Analogue Hadron CALorimeter (AHCAL)* implementation by CALICE will be explained in detail.

4.1.1 AHCAL Technology

The AHCAL is a highly-granular sampling calorimeter separated into active and passive layers in longitudinal direction, a so called *sandwich* structure. For the current design study most of the passive material consists of steel. The active layers are made of multiple plastic scintillator tiles of a few square centimetres surface area and a few millimetres thickness. The scintillator tiles are wrapped in reflective foil with one hole for scintillation photons to be read-out. The read-out is performed with a *silicon photomulitplier (SiPM)*. Those SiPMs, together with the read-out electronics including multiple ASIC (*application-specific integrated circuit*) chips, are soldered onto a *printed circuit board (PCB)*. The wrapped scintillator tiles are glued on top of the respective SiPM on the PCB with ideally no air gap in between tiles. This stack of tile and SiPM is called *SiPM-on-tile* technology. A picture of a wrapped and unwrapped SiPM-on-tile is shown in figure 4.2.

The ASIC reads out the SiPM charge and store basically two information: an integrated charge which corresponds to the energy deposited in the scintillator tile and a time stamp. This way for each scintillator tile, or *calorimeter cell*, energy and time of shower particle is recorded.

4.1.2 AHCAL Engineering Prototype

The *2018 CALICE AHCAL engineering prototype*, from now on only refereed to as *'AHCAL'*, is the most recent and largest AHCAL prototype developed by the CALICE collaboration to date. The AHCAL consists of 38 active SiPM-on-tile layers with approximately 20 mm stainless steel absorber plates in between. This leads to a depth of $\approx 4\,\lambda_{int}$.

Each active layer consists of 4 *hadronic base units (HBUs)* - PCBs integrated with four ASICs, 144 SiPMs, calibration LEDs, and additional read-out electronics. As ASIC the SPIROC2E chip was used. All HBUs are tiled with 12x12 = 144 plastic scintillator tiles of 30x30x3 mm^3 size. Hence each active layer consists of 4x12x12 = 576 calorimeter cells. The total surface area of each layer is 72x72 cm^2 and the calorimeter is about 75 cm deep resulting in a calorimeter volume of approximately 0.4 m^3. With its 38 layers the AHCAL has 21,888 channels in total. After completing the assembly

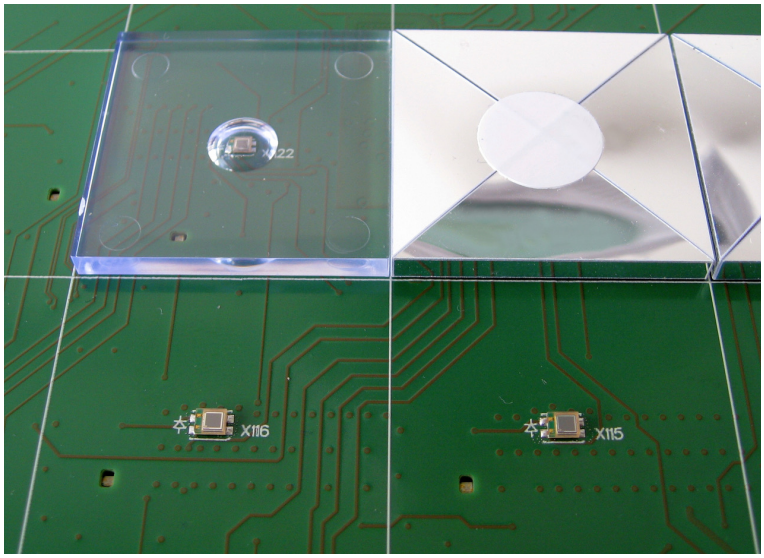


Figure 4.2: SiPM-on-tile technology: Two 30x30x3 mm^3 CALICE AHCAL plastic scintillator tiles (left: unwrapped, right: wrapped in reflective foil) glued on top of silicon photomultipliers (SiPMs) on a *hadronic base unit (HBU)*). [25]

99.96 % channels are tested as operational. Figure 4.3 shows one active layer consisting of four HBUs. In figure 4.4 the whole assembled AHCAL is shown.

For efficient operation in a linear collider the AHCAL can be operated in a *power-pulsing* mode. Unlike in normal operation mode, this mode allows to turn off most of the read-out electronics in between beam collisions which are much less frequent in linear colliders than in circular ones. The power-pulsing mode leads to extensive power saving and lower temperature operation.

During the manufacturing of the prototype a process was designed to create a scalable AHCAL assembly for a future detector. Most of the assembly including HBU manufacturing, the tile wrapping and glueing were (semi-)automated. A *International Large Detector (ILD)* study assumed approximately 8 million channels in the final HCAL making an efficient production process necessary.

4.1.3 Calibration

A comprehensive calibration chain is necessary to properly calibrate all 21,888 calorimeter channels. Every single channel has to be calibrated separately due to non-uniformities such as unequal tile wrapping and glueing or SiPM and ASIC specific features. Furthermore, a chip by chip calibration is necessary as there are non-uniformities in the electronics and memory cells of each individual chip.

Two calibration chains are needed, one for hit energy calibration and one for time calibration. The energy or SiPM charge is read-out as ADC (analogue to digital converter) and needs to be calibrated into units of minimum ionising particles (MIPs). The time is stored as a charge TDC (time to digital converter) and is calibrated into values of nanoseconds.

In addition to ADC and TDC a *gain bit* is stored as the SPIROC2E chip can store the ADC either in low gain or in high gain mode. For the low gain mode a different

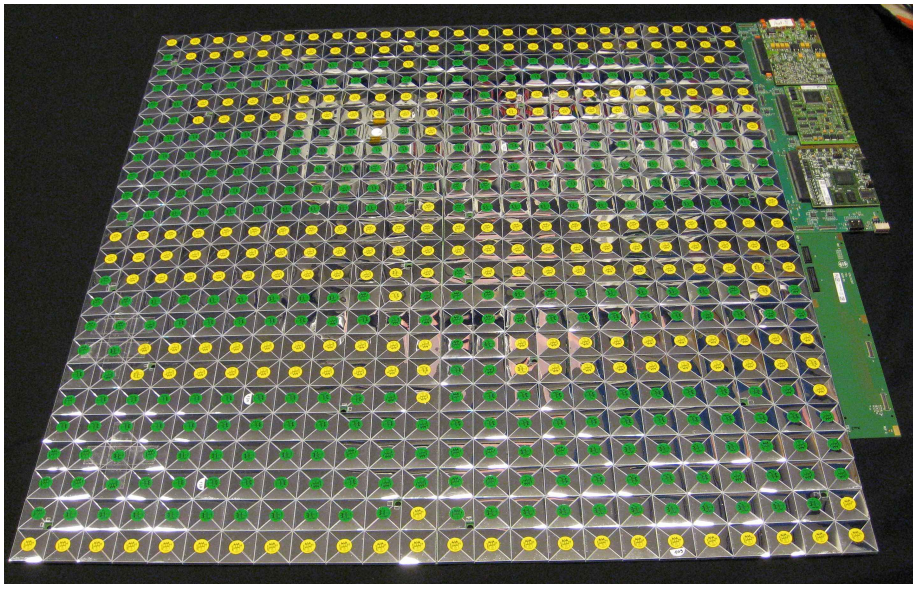


Figure 4.3: Whole active AHCAL layer with four hadronic base units (HBUs) of wrapped tiles and the interfaces for data acquisition, LEDs and power supply. The colour coding on the wrapped tiles corresponds to the LED location. [25]

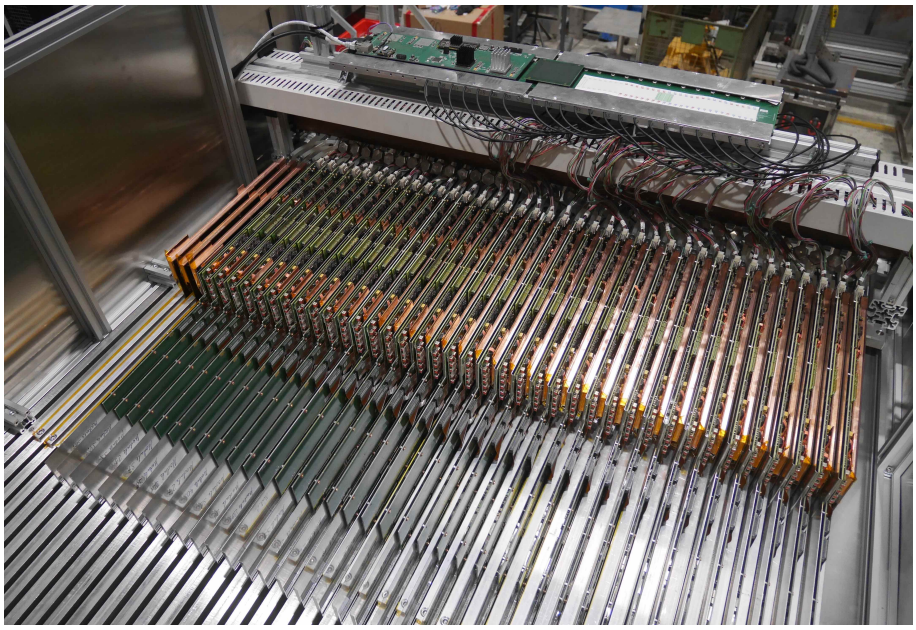


Figure 4.4: View on the top of the (opened) 2018 AHCAL engineering prototype fully assembled with 38 active layers. The picture was taken during test beam at SPS in May 2018.

capacitor is used that reduces the ADC by an *intercalibration factor* IC^{HGLG} (roughly factor 20). This way a wider dynamic range in which the capacitors have a linear response to the charge deposited is achieved. The intercalibration factor is ASIC specific and needs to be included in the calibration chain.

4.1.3.1 Energy Calibration

The following calibrations to the measured raw signal a_i [ADC] in channel i are applied:

- *SiPM gain G_i*

  The SiPM gain G_i is the response difference between one more or less SiPM pixel firing. It is extracted from a single photon spectrum recorded during a LED calibration run. The LED runs are runs in calibration mode in which a LED under each scintillator tile is emitting exactly enough photons for the SiPM to fire a couple of pixels and record the single photon spectrum. G_i is defined as the distance between the pedestal and the second photo-electron peak of the spectrum divided by two. The gain is temperature dependent.

- *saturation correction function $f_{sat,i}$*

  The SiPM response saturates with increasing photons emitted in the scintillator tile due to the dead time of the SiPM pixels in Geiger mode. Hence the recorded ADC charge a_i has to be corrected for this saturation effect. In calibration mode the SiPM response is measured until saturation. A function $f_{sat,i}$ is extracted from this response and its inverse can be used as the desaturation function $f^{-1}_{sat,i}$.

- *pedestal subtraction P_i*

  Specific test beam runs with muons are performed to extract the pedestal and the MIP constants. For each low gain and high gain mode a pedestal P_i is measured and subtracted from a_i.

- *pedestal memory cell offset $O_{i,m}$*

  A special feature of the SPIROC2E ASIC is that all of its 16 memory cells are slighly different. Depending on in which memory cell (m) a_i was stored a offset factor $O_{i,m}$ has to be subtracted from the channel wise pedestal P_i.

- *intercalibration factors IC_i^{phy} and IC_i^{HGLG}*

  As the saturation function $f_{sat,i}$ is measured in calibration mode but applied to data from the detector in physics mode an intercalibration factor IC_i^{phy} is applied to the gain value when applying it in $f_{sat,i}$. As mentioned above to account for low gain and high gain mode another intercalibration factor IC_i^{HGLG} is applied to a_i after pedestal subtraction in case a_i was recorded in low gain mode.

- *MIP constant $C_{i,p}^{MIP}$*

  To compare physics results, values of MIP were chosen as a common energy scale. For this MIP calibration constants $C_{i,p}^{MIP}$ are extracted from muon runs by calibrating the constants such that the most probable value (MPV) of muon response peaks at 1. This is done by fitting a Landau-Gaussian convolution function to the channel wise response to muons which is expected to be most probable at 1 MIP. A complication to the extraction of MIP constants arises from the two power modes in which the AHCAL was operated - either with or without *power-pulsing*. In power-pulsing mode the detector is cooler and MIP constants differ by about 2 %. Hence $C_{i,p}^{MIP}$ depend on the power mode p.

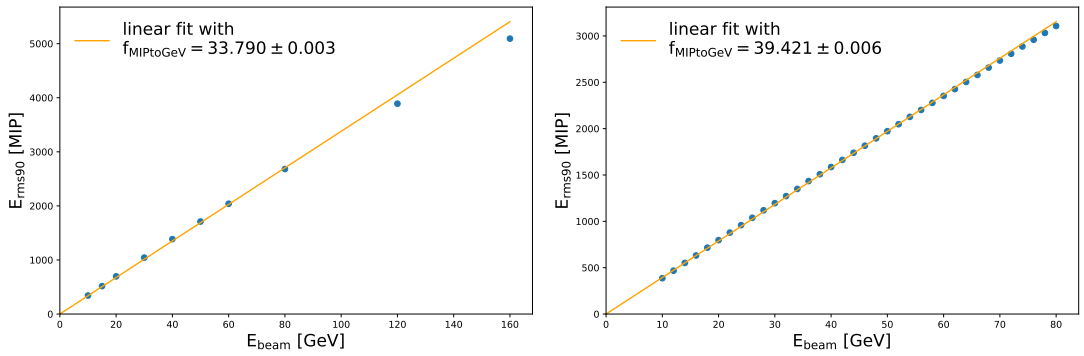
With all the above calibration factors determined the calibrated hit energy E_i in MIP can be calculated by applying the hit energy calibration as follows:

$$E_i = f_{sat,i}^{-1} \left(\frac{(a_i - P_i - O_{i,m}) \cdot IC_i^{HGLG}}{\frac{G_i}{IC_1^{phy}}} \right) \cdot \frac{G_i}{C_{i,p}^{MIP}} \tag{4.1}$$

If channel specific values could not be obtained, default values are estimated. All calibration factors are made available to members in CALICE via an internal database. The database is accessed by the CALICE AHCAL reconstruction software for event reconstruction and digitalization of Monte Carlo simulations. All events are stored uncalibrated in the LCIO file format (Linear Collider Input Output) and can be reconstructed into reconstructed LCIO files or ROOT files, the latter being a common standard in high-energy physics.

For the analysis in this thesis the energy scale is often given in GeV for easier readability and comparison with other experiments. This requires an additional MIP-to-GeV calibration factor $f_{MIPtoGeV}$. Assuming a linear response of the calorimeter this factor is derived from a linear fit without pedestal. However, neither for data nor for Monte Carlo the calorimeter response is exactly linear, as is shown in the upper plot of figure 6.3 in chapter 6. The factor is particle type dependent due to non-compensating nature of the AHCAL (see section 3.3.2).

To compute $f_{MIPtoGeV}$ for pion samples the beam energy was plotted against the mean reconstructed energy (using RMS90; see section 4.3). Before E_{rms90} was calculated, the sample selection to both data and simulation was applied as discussed in section 6.1. The data and the resulting fit parameters are shown in figure 4.5. The fit has to be performed for data and Monte Carlo simulation separately as there is a slight difference between data and Monte Carlo is observed, hence data and simulation specific $f_{MIPtoGeV}$ are used in this thesis (see section 4.2.3).



(a) Data **(b)** Simulation

Figure 4.5: Plot of beam energy against reconstructed mean energy E_{rms90} for both data and simulation for pion samples. Linear fit performed to extract MIP-to-GeV calibration factor $f_{MIPtoGeV}$.

4.1.3.2 Time Calibration

Just like most of the energy calibration, time calibration is performed with muon run data since interactions with muons as charged heavy particles happen instantaneous (see 3.1.1). The time information is stored in TDC values and needs to be calibrated to nanoseconds to have a common time reference in each event. For this purpose a time calibration chain is necessary as well as a global clock that references the hit time with the start of the bunch crossing (in a collider experiment) or in general the trigger that started the event recording.

For the time to be recorded the TDC voltage is ramped up and down in intervals of 4000 ns. For this purpose the SPIROC2E chip has two TDC ramps for even and odd bunch crossings (bx) that form together a combined up and down ramping. Additionally there is a global clock that is turned on and off every 2000 ns and triggers a new bunch crossing identification numer (bxID) to reference the global event time.

Once a hit occurs the bxID as well as the current TDC value is recorded in one of the SPIROC's 16 memory cells. From the slope of the TDC ramp and its pedestal the time in nanoseconds can be calculated so both has to be channel wise & memory cell wise calibrated. A detailed instruction on how the time calibration was performed for prior prototypes can be found in [17].

At the time this thesis was written the time calibration was not yet finished for AHCAL test beam data. Only in simulated Monte Carlo data the timing was consider, however no detector like digitalization was applied. Instead simply a 1 ns Gaussian smearing was applied to the accurate Monte Carlo timing. This 1 ns time resolution of the HCAL is one of the design goals for a future CLIC detector [5]. In practice the time resolution is limited by front-end electronic effects.

4.1.4 CALICE Software Framework

Event reconstruction and digitalization of simulation was performed by the CALICE software that is based on the ILCsoft framwork [26]. The latest version available at the time v4-12 of the CALICE software was used for all events analysed in this thesis. The CALICE software framework was developed over many years by several members of the CALICE collaboration including senior scientists and students. The software uses several *Marlin* processors to perform the energy reconstruction. During the reconstruction several hit-, event-, and layer-wise variables are calculated and written into root tree branches. Among those are number of hits ('nHits'), the sum $\sum E_i$ of all hits in one energy (called 'energy sum' or 'energySum'), hit energies, hit coordinates in X, Y, Z (named I, J, K coordinates), centre of gravity in X, Y, Z and layer-wise number of hits and energy sum.

During the reconstruction a cut on the hit energy values is applied. All reconstructed hit energies below 0.5 MIP are rejected because those hits cannot be distinguished from random noise.

An additional pre-release processor was used to find the active detector layer in which the hadronic shower starts, namely the layer in which the first hadronic interaction is detected. A detailed description of how the algorithm works can be found in [27]. For the data samples the shower start finder based on a *moving average window* has been applied, while for the Monte Carlo samples the *Monte Carlo shower start* was used.

4.2 Test Beam Campaigns

The AHCAL prototype was fully assembled in early 2018 and underwent multiple test beam campaigns since then. In April 2018 it underwent tests at DESY where it was tested with an electron test beam to acquire preliminary calibration constants. In May 2018 the AHCAL was shipped to CERN in Geneva to be used in three separate test beam campaigns over the following months. All three campaigns were carried out at the CERN North Area in beam line H2 of CERN's Super Proton Synchrotron (SPS). Campaigns took place for two weeks each in May 2018, June to July 2018 and in October 2018. The author of this thesis took test beam shifts for one week each during the May and June campaign.

For the May campaign only the AHCAL engineering prototype itself was tested. Multiple particle types and energies as well as the power pulsing mode were tested. A difficulty in the May test beam was a large electron contamination in the low energetic pion beams due to far from optimal beam line settings. This problem was resolved in the later campaigns. Further details will be given in the next section as the data analysis in this thesis mainly deals with data taken during the May campaign.

For the June campaign a former AHCAL prototype with only 12 layers with each 1 HBU and 7.4 cm steel absorbers was used as *tail catcher*. The purpose of the tail catcher is to capture a part of the shower that is not contained in the main AHCAL itself (*shower leakage*) and is therefore positioned centred behind the main AHCAL. Additionally a 'pre-shower' layer consisting of 1 HBU was glued in front of the detector to potentially reject showering particles that interact before they enter the main AHCAL.

Furthermore an additional active layer was added after the May campaign: The 'Tokyo layer' was positioned in layer 37 resulting in a total of 39 layers for the enhanced AHCAL engineering prototype. This layer was developed in Tokyo and offers a larger tile size with a surface area of 60x60 mm^2. With this layer a mixed calorimeter granularity is tested. Preliminary results from this study can be found in [28].

Having successfully tested the power pulsing mode in May, the later test beam studies were mostly performed in power pulsing mode.

The October campaign was undertaken together with the Compact Muon Solenoid (CMS) collaboration and their prototype for a *High Granularity Calorimeter (HGCAL)* as a replacement for the endcap calorimeters in the CMS detector during the High-Luminosity upgrade to the LHC (HL-LHC) [29]. The CMS HGCAL boasts a similar SiPM-on-tile technology as the CALICE AHCAL, but with a finer granularity. In this test beam the AHCAL was used as a tail catcher for the HGCAL and was placed centred behind the HGCAL. The data acquisition systems for both calorimeters were fully integrated into a common framework.

After the October campaign the AHCAL prototype was shipped back to DESY to undergo further test beams and studies in 2019 and onwards. The analysis efforts for all test beam campaigns are ongoing.

4.2.1 Test Beam Campaign in May 2018 at SPS

The studies in this thesis were performed with test beam data taken during the campaign at SPS between May 9-23, 2018. The photo 4.6 shows the AHCAL setup. The 38 layers are placed in a specially designed metal box of the size of a possible ILD

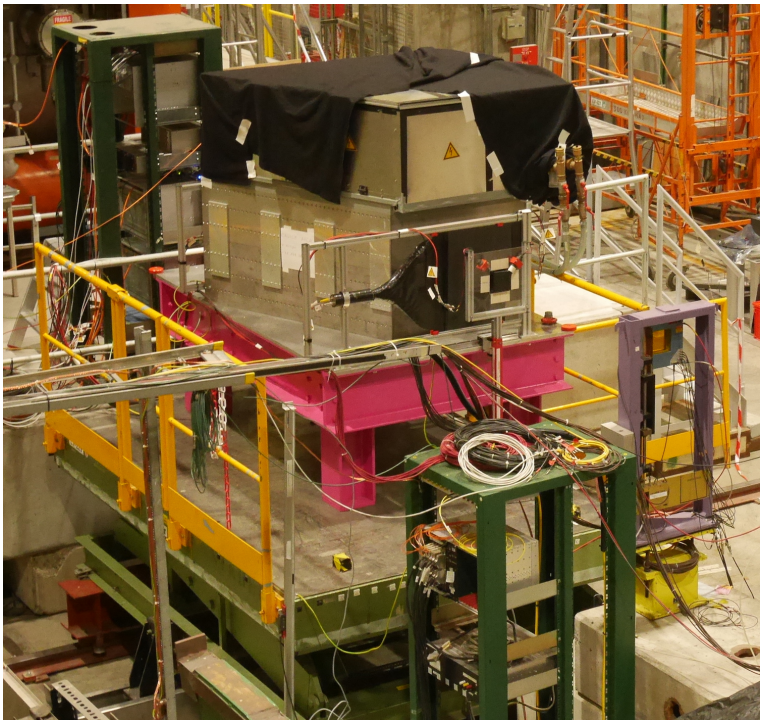


Figure 4.6: May 2018 test beam setup at SPS north area, beam line H2. The covered AHCAL prototype on a movable stage with DAQ system and cooling behind the detector. Beam incident is in the front of the detector from this point of view.

module. A water cooling system for the power interfaces and the data acquisition electronics are placed behind the calorimeter. The AHCAL sits on a movable stage that can be moved in x- and y-direction. With the movable stage a full detector scan could be performed. This was especially important for muon calibration runs as enough statistics for every single cell to be calibrated has to be recorded.

During the two weeks of test beam a great number of runs were performed, including LED calibration runs and standard test beam runs with negative muon, electron or negative pion beams of different beam energies, detector positions and powering mode. The total number of events recorded per particle type and energy can be found in the appendix under B. However, the numbers given do not account for different particle type contamination, noise triggered events and double particle events.

To choose the particle type several beam instruments such as absorbers, bending magnets and collimators can be adjusted upstream of the beam line. The SPS accelerates a primary proton beam to about 400 GeV, part of which is directed onto a target which results in the production of various particle types. From this target a secondary beam is channelled off that can be purified by usage of beam instruments. Muons are obtained by stopping a pion beam with a concrete absorber block as the muon cross section is very low. Hence all pion runs include muons, too. For pions an additional contamination through electrons is possible if they are not filtered correctly. This appears to have lead to additional electron contamination in low energy pion runs (< 20 GeV) as is shown in section 4.2.2.

4.2.2 Data Quality Analysis

Basic data quality checks for the pion test beam data for the May and June campaign were performed by the author of this thesis. The goal of the data quality analysis was to check the quality and correct labelling of all pion runs. The analysis of data quality and quantity was already started during the test beam campaigns itself including the continuous update of a spread sheet with all relevant run information in addition to the standard electronic log book. For the May and June campaigns the data quality analysis was finished during the *CALICE AHCAL Tokyo Analysis Workshop* in August 2018. In the following results for pion runs of the May campaign are presented.

For all runs the labelling was checked and they were sorted into three categories and flagged accordingly: *good*, *check*, and *bad*. The goal was to easily screen the runs for *good* runs that can be used for further analysis. Runs were marked as *check* if it was believed that further calibrations or an accurate particle selection could make the run usable. *Bad* runs were flagged if the runs were not usable for detailed analysis due to an error in the experimental setup.

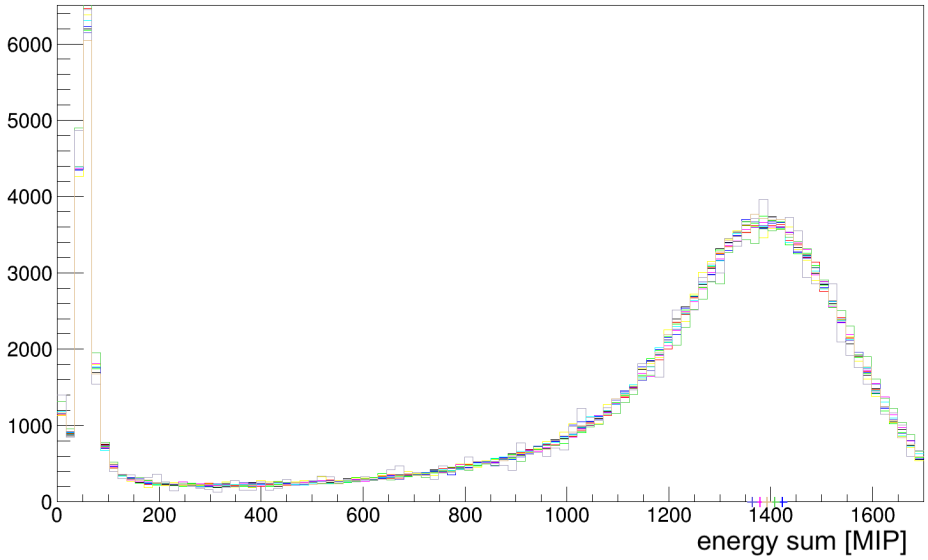


Figure 4.7: Energy sum ($\sum E_i$) distributions for all 40 GeV pion runs measured during the May test beam campaign. Power-pulsing mode specific calibration constants are applied. The peak bin position is marked on the x-axis for all runs.

The following variables were compared for all standard runs: energy sum ($\sum E_i$), number of hits (nHits) and centre of gravity in X and Y direction. All runs with common beam settings without any beam optimization or other detector tests are used. Those were marked in the log book accordingly. The comparison plot for 40 GeV pion runs is shown in figure 4.7. Further comparison plots can be found in the annex under C.

In the histograms a peak is observed around 60 MIP. This peak is due to muon contamination and 'punch-through' pions, which do not start a hadronic shower in the AHCAL. The most probable energy deposition for muons is calibrated to 1 MIP, but on average a muon deposits more energy per layer as the energy deposition follows a

Landau distribution. Therefore the 'muon peak' is above 38 MIP. Pions that shower in the calorimeter deposit all or a large part of their energy in the calorimeter. This leads to a second peak in the E_{sum} distribution at a larger energy than the 'muon peak'.

The position of the peak bin position of this second 'pion peak' is marked on the x-axis and compared with same beam energy runs. This way outliers could be easily spotted. Outliers were defined by a deviation of about 5 %. Additionally the peak bin positions were plotted against the run number which increased with every new run to be able to spot systematic differences that occurred over time.

During the quality analysis a few features were noticed:

- One 100 GeV run was marked as an outlier which is visible from the broader energy sum distribution. According to the log book this run was taken without one specific absorber upstream of the beam line. Therefore the broader distribution can be explained due to a larger electron contamination. It was additionally noticed that one specific collimator (XCHV.021.133) was fully opened for all 100 GeV runs recorded resulting in a smeared nHits distribution. This collimator is of importance to define the exact beam momentum. As this collimator was opened one cannot be confident that all events recored are actually 100 GeV events. The runs could be usable once a suitable particle and momentum identification algorithm is developed and are therefore all 12 runs labelled *check* to be reviewed in a later analysis. In this thesis the 100 GeV runs were not considered further. The number of hits histograms for all 100 GeV runs can be found in C.1.

- A systematic difference in the energy sum between runs with and without power pulsing mode of about 2 %. This difference is due to usage of the same MIP calibration constants for both modes although the temperature of the AHCAL is lower in power pulsing mode. In later calibrations this difference could be resolved by applying power-pulsing mode specific calibrations (see section 4.1.3.1). An example for the energy sum histograms of all 40 GeV runs with and without power mode specific calibrations can be found in C.2.

- A large electron contamination of the 10 and 15 GeV runs. This was already noticed during the test beam campaign and was resolved with optimized beam line settings in later campaigns. As this is a systematic issue for all low energy May runs, all those runs were flagged as *good* and were used in this thesis. A comparison between May and June pion runs of 10 GeV is shown in C.3. Electron rejection cuts are discussed in section 6.1.

- One 80 GeV run was flagged as a *check* run to be reviewed later as a shifted energy sum distribution was noticed. The outlier can be seen in plot C.4.

Overall the data quality of 137 out of 150 pion runs of the May test beam are marked as *good*. Therefore for the rest of the analysis in this thesis there is no differentiation between data of runs with the same beam energy made as only 'good' runs are considered as data. The same analysis has been performed for the pion data taken during the June campaign, but is not subject of this thesis.

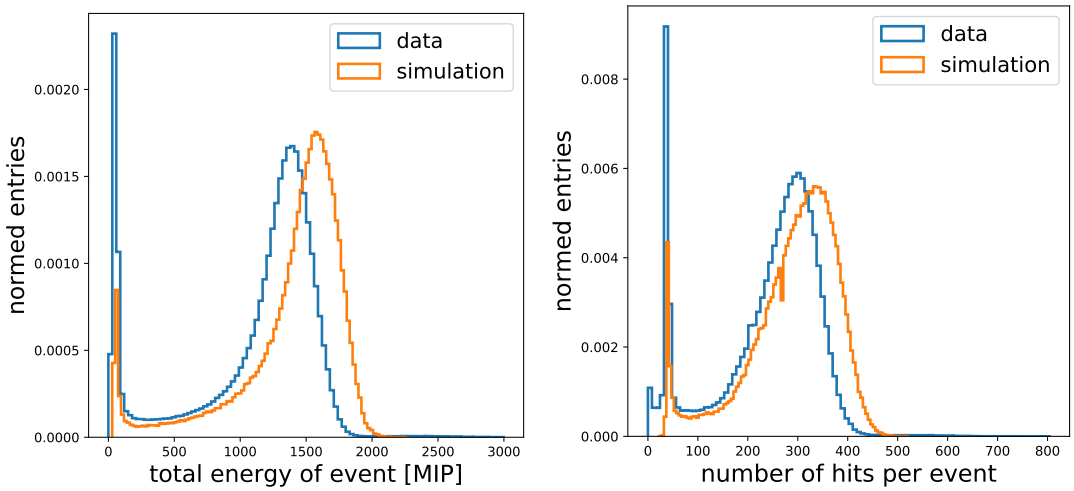
4.2.3 Monte Carlo Simulation

A Monte Carlo (MC) simulation of the AHCAL and the test beam setup has been developed by members of the CALICE AHCAL group using Geant4 with the simplified scripting interface DD4hep. Simulation parameters such as additional material in front of the calorimeter, detector position shifts, the influence of the choice of Birks' constant and exact material compositions are still under investigation and further improvement of the simulation accuracy is possible.

The simulation takes into account the geometry of the prototype and its materials. Realistic detector effects need to be applied hence a *digitalization* step is introduced. This step is performed on the simulated samples to bring them into the same uncalibrated state as the data samples with channel-wise irregularities and hit energies in values of ADC instead of GeV. Exactly like the real AHCAL, the simulation needs to be calibrated as well, i.e. with MC muon runs and the same Landau-Gaussian convolution fit to tune the MPV to 1 MIP. Further details can be found in [30] Afterwards the digitalized MC samples are reconstructed with the same energy calibration software as discussed in section 4.1.4. Detailed steps of the digitalization process can be found in [27].

For the production of the simulation samples in this thesis the most recent version as of March 2019 has been used. As simulation samples negative pions were simulated in the energy range from 10 - 80 GeV with a 1 GeV spacing. The simulation needs extensive computing resources, especially disk space, nevertheless 100k events per energy were produced. After cuts applied (see section 6.1) this leaves more than 32k events per energy.

Additional material is added in the front of the detector to tune the response to electrons of the AHCAL in comparison to data. The material is 8.785 mm steel which corresponds to $\approx \frac{1}{2} X_0$. The beam is positioned 1 m away from the centre of the AHCAL with an offset to the centre by 15 mm in x and y direction. This means the beam is aiming in the middle of tiles next to the AHCAL centre.



(a) total energy **(b)** number of hits

Figure 4.8: Normed histograms of 40 GeV pion samples for data and for Monte Carlo simulation.

However, the Monte Carlo simulation in its current implementation deviates from data. Figure 4.8 shows a comparison for the number of hits and the energy sum per event for both data and Monte Carlo. For both figures of merits the peak value in the simulation is approximately 15 % larger than in data. Hence also the 15 % larger $f_{MIPtoGeV}$ for MC (see section 4.1.3.1). Therefore a good comparison between data and simulated samples cannot be made in this thesis.

To improve the simulation, several effects are currently under investigation by members of the collaboration. This includes the influence of the exact constant used in Birk's law, a comparison of different physics lists in Geant4 and detector effects such as the SiPM saturation model in the digitization step.

4.3 Calculating the energy resolution

Ideally the energy distribution for one energy is of Gaussian shape. In that case one can determine a resolution σ/E by calculating E as the mean energy and σ as the standard deviation of the energy distribution. Approximately same values could be taken from the fitting parameters of a scaled Gaussian distribution

$$f(x;\mu,\sigma,a) = a \cdot e^{-\frac{(x-\mu)^2}{2*\sigma^2}} \tag{4.2}$$

with a as a scaling factor and the height of the distribution.

However, the energy distributions might not be Gaussian shaped, but rather have a tail towards lower and/or higher energies. A low energy tail can be caused by not fully containing all showers in the calorimeter which results in a lower total energy of the measured event (shower leakage). A high energy tail can be caused by multi particle events that occur during test beam, but are not modeled in the simulation.

For Monte Carlo samples, this tail is mainly due to shower leakage. If one cuts this distributions with requiring no hits in the first and last layer, fully in the AHCAL contained showers, this tail towards lower energies almost completely disappears. However, by requiring a particle shower fully contained in the calorimeter, one significantly reduces the number of available events as well as biasing the hit energy distribution. This bias is due to a higher electromagnetic content of the hadronic shower when one requires a shorter longitudinal shower expansion.

For quoting an 'energy resolution' in the following chapters we will not perform a cut on the shower end because of the bias on the electromagnetic shower fraction as well as the limited statistics that would be left from the available data and MC samples. Hence we want to determine the 'mean' and the 'width' of energy distributions with a tail towards lower energies. Multiple options to determine these parameters have been considered and will be outlined in the following. It is important to state that because of the non-Gaussian shape of the energy distributions in this thesis the resulting 'energy resolution' cannot be easily compared to other calorimeters. The metric chosen here should only be used to compare results for the same calorimeter and the same method as well as for comparing different reconstruction algorithms on the same event samples as is done in this thesis.

The following options were considered:

- *RMS*
  Using the mean and the standard deviation of the whole distribution. Here the

whole distribution including the low energy tail and potential fluctuations are taken into account. For consistency with the *root* software framework standard deviation and RMS (Root Mean Squared) are used in this thesis synonymously. The standard error of the mean is estimated with $\sigma/\sqrt{N}$ and of the RMS with $\sigma/\sqrt{2N}$ with σ as the standard deviation and N as the number of events.

- *RMS90*
  Using the mean and the standard deviation of 90 % of the distribution. Those 90 % are chosen by the smallest energy / x-axis window in which 90 % of all events are contained. The reasoning behind mean90 and RMS90 is that one would like to show parameters unaffected by high- or low-energy fluctuations that usually limit the resolution of any hadronic calorimeter. As no fitting procedure is involved quoting these numbers is very robust no matter how the distributions look. Furthermore for a Gaussian distribution the RMS can be estimated by approximately 1.26 x RMS90. [6]

  In practice this smallest window can be found by sorting the values of the distribution and calculating the RMS of all any continuous window in which 90 % of events are contained. The window with the smallest RMS is picked accordingly. The exact implementation as Python code can be found in A.

- *2-stage Gaussian fit in $\pm 2\sigma$*
  Here a Gaussian function (equation 4.2) is fitted to a part of a histogram of the distribution. This in done in two stages. In the first stage the fitting range of the Gaussian function is chosen in $\overline{x} - 2\sigma < x < \overline{x} + 2\sigma$. Afterwards the fitted parameters $\mu_{Gaussian}, \sigma_{Gaussian}$ of the first stage Gaussian are used to determine the fitting range for the second stage fit in $\mu_{Gaussian} - 2\sigma_{Gaussian} < x < \mu_{Gaussian} + 2\sigma_{Gaussian}$. The fitted parameters $\mu_{Gaussian,stage2}$ and $\sigma_{Gaussian,stage2}$ are used to quote the mean and width of the distribution. This way the low energy tail of the distribution is only partly taken into account and very high- or low-energy fluctuations are neglected. Furthermore $\mu_{Gaussian,stage2}$ represents very good the position of the peak of the distribution.

  This fitting procedure was used in previous CALICE studies such as [21]. The exact fitting range can be optimized i.e. by fitting in an asymmetric range around the mean. In general, for Gaussian distributions it is expected to have 95 % of all data within the $\pm 2\sigma$ interval.

- *GaussExp function*
  This function is a convolution of a Gaussian function and an exponential one. It is introduced in [31] and is a simplified version of the Crystal Ball function. The GaussExp function with the exponential tail towards lower values is given by

$$\begin{aligned} f(x;\mu,\sigma,k,a) &= a \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \text{ for} \frac{x-\mu}{\sigma} \geq -k \\ &= a \cdot e^{\frac{k^2}{2}+k\left(\frac{x-\mu}{\sigma}\right)}, \text{ for} \frac{x-\mu}{\sigma} < -k \end{aligned} \tag{4.3}$$

  with μ and σ as the mean and the standard deviation of the Gaussian, k describes the transition point between the Gaussian function and the exponential tail, and a is a scaling factor that gives the height of the distribution. More specifically the

transition point is given by $\mu - k\sigma$. This function can describe the distributions very well, but it might be misleading if the k parameter is low as the Gaussian width σ is here fitted in the range $\mu - k\sigma < x < \infty$. This way depending on the k parameter as large part of the distribution is neglected when quoting the σ.

The resulting fits and the fit parameters can be found in figure 4.9. Comparing the σ parameters it is apparent that the GaussExp function gives the lowest width due to the low k value. The mean of 90 % of the distribution is comparable to the μ from the Gaussian fit within $\pm 2\sigma$, but the width σ is a bit smaller. For this distribution the estimation 1.26 x RMS90 $\approx \sigma_{Gaussian,stage2}$ does not hold true.

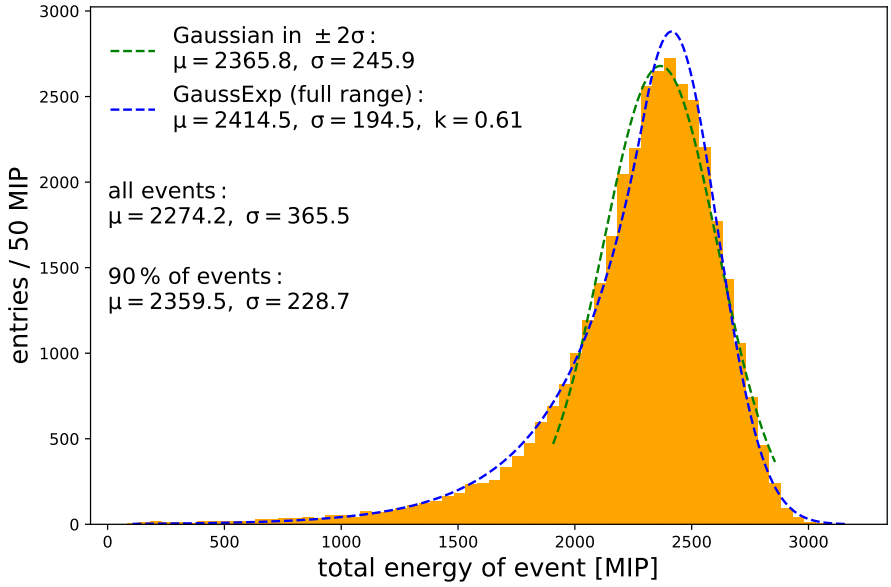


Figure 4.9: Histogram of a 60 GeV pion Monte Carlo sample. A Gaussian (equation 4.2) and a GaussExp function (equation4.3) is fitted to the histogram. Additionally the RMS and the RMS90 is stated, here μ and σ denote the mean and the standard deviation. A detailed explanation can be found in the text. The sample was prepared with shower start cut that requires the shower start in the first five layers and at least 60 hits per event. A binning width of 50 MIP was chosen.

The final choice of metric is arbitrary. *RMS90* has been chosen as it is easily to replicate and was used already in former studies of members of the CALICE collaboration and the wider PandoraPFA community such as [6]. It gives an estimate of the distributions width without overemphasizing the shower leakage or fluctuations that limit the resolution. Furthermore the method is very robust as no fit is involved and even can applied to predicted distributions of neural networks that exhibit non-physical features as can be found in chapter 6. Hence, in this thesis RMS90 is used to determine E and σ in the energy resolution to which henceforth is refereed to without quotation marks.

Chapter 5

Machine Learning

Physicist have been using machine learning algorithms already for the past two decades to analyse the large data sets emerging from high-energy physics experiments. These algorithms include boosted decision trees (BDTs) as well as neural networks (NNs). The improved data analysis capabilities of trainable algorithms made contributions to finding the Higgs boson in 2012 [1, 2]. The reliance on data analysis with machine learning algorithms in high-energy physics is likely to grow even further in the future [32].

In the following chapter, some general concepts of machine learning are introduced, as well as specific architectures of neural networks used for the physics analysis in this thesis. Section 5.1 introduces the basic ingredients one needs to train a algorithm. The concepts behind the minimization of the loss function with an optimizer as well as the terminology of Deep Learning are explained. Basic layers in a neural network, such as fully connected layers and convolutional layers are introduced in section 5.2. The software frameworks used for the implementation of the algorithms in this thesis are explained in section5.3.

5.1 Machine Learning Basics

Machine Learning (ML) referees in general to any kind of algorithm with variables that are optimized using data to make predictions. With increasing data storage capabilities and high performance computing infrastructures becoming widely available, ML is evolving as a common tool for statisticians and scientists in many industries and scientific fields. Often ML is refereed to as a subfield of *artificial intelligence (AI)* although the term is loosely defined.

In high-energy physics ML has been used to optimize (or *train*) algorithms for many years since large labelled datasets are available which improves their capability. ML is usually applied to complex, high-dimensional problems that cannot be solved by traditional statistical methods. However, conceptionally ML is very similar to 'fitting' a function to a dataset via a χ^2 minimization. The same is done in ML where a metric, the *loss*, is minimized during the training and the parameters are optimized with an algorithm referred to as *optimizer*.

With modern optimization algorithms and simple programming implementations (see section 5.3) it has become technically straightforward to train algorithms with a

very high number of parameters up to several millions.

To perform supervised ML one needs a couple of ingredients [33]:

- A *dataset* $D(X, y)$ with X as a matrix of independent variables and an observation or 'truth' label y per set entry. Generally the dataset is split into two statistically independent sets, a *training* $D_{train}(X, y)$ and a *testing* $D_{test}(X, y)$ set. Only $D_{train}(X, y)$ is used to optimize the model that predicts y from the X. After the training the model performance is evaluated based on its ability to predict y on $D_{test}(X, y)$. Often the $D(X, y)$ is split into a very large portion D_{train} ($\approx 80\,\%$) and a much smaller part D_{test} ($\approx 20\,\%$). This mutual exclusive splitting of the dataset is called *cross-validation*. Additionally one might choose to evaluate the performance of the model already during the training by splitting from D_{train} another *validation* set $D_{validation}$ that is used to validate the models performance after each training step.

- A *model* $f(x, w)$ which is a function $f : x \to \hat{y}$ of parameters or *weights* w. The model is a fitted to the training data to describe y by optimizing the weights w. The model f is defined by the user prior to training. Usually f is a non-linear model due to the introduction of non-linear functions, i.e. *activation functions* (see section 5.2.2). The exact layout of its functional relationship is called *model architecture* for neural networks and examples are given in section 5.2.

- A *loss function* $L(f(x, w), y)$ (also called *cost function*) is a metric to evaluate the performance of the model. This could be for example *mean squared error* $L_{MSE} = \frac{1}{N}\sum(f(x, w) - y)^2$, with the sample size N. The model is optimized by minimizing the loss function with the $D_{train}(X, y) = D(X_{train}, y_{train})$ training set. The optimized weights $\hat{w}$ are determined by $\hat{w} = arg\,min_w\{L(f(x_{train}, w), y_{train})\}$. An overview on different loss functions is given in section 5.1.1.

- An *initializer* w_0 that sets the initial values of the weights w before the model training is started. The values are randomly chosen based on the type of initializer. A recommendable initializer depends on the layer type and the activation function used in a neural network. A popular choice is a weights initialization with values randomly chosen from a Gaussian distribution with a mean zero [34].

- An *optimizer* that is used to find the weights w that minimize the loss function. This is basically done by adjusting w in the direction where the gradient of loss $\nabla_w L(w)$ is large and negative ('gradient decent'). Due to this constant adjustment of w by the optimizing algorithm the training is an iterative process. For a more detailed discussion see section 5.1.2.

Current uses of ML in particle physics can roughly be separated into three different task: *classification*, *regression* and *generation*. *Classification* tasks involve the model to separate the data into different classes. A popular example is an image classification algorithm that can distinguish between pictures of cats and dogs. A more physics relevant tasks could be top-tagging [35] or separation of particles in a calorimeter (see section **??**). *Regression* tasks involve the prediction of a continuous number such as the age of a person or the energy of a high-energetic particle. Most of the results of this thesis were produced with a regression ML algorithm and a discussed in

detail in chapter 6. *Generation* involves the training of an algorithm that can produce similar, but not identical examples of the training set. A fun example is the automatic generation of internet memes [36]. In physics generation networks can be used to create a fast alternative for Monte Carlo simulations [37].

5.1.1 Loss Functions

The loss function $L(f(x,w),y)$ gives a metric to evaluate the performance of the model $f(x,w)$ in predicting the observation y. For regression problems a widely used loss function is *mean squared error (MSE)*:

$$L_{MSE} = \frac{1}{N}\sum(f(x_i,w)-y_i)^2, \tag{5.1}$$

with the sample size N. Alternatively variations of this error estimator, such as the relative error, can be used for the performance estimation. The right loss function used depends highly on the task the model is supposed to fit.

In this case these loss functions apply to *supervised learning* which implies that for each data entry the truth observable y is given to the model optimizer.

5.1.2 Optimizers

The optimizing algorithm (called *optimizer*) is used to find the weight values $\hat{w}$ which minimize the loss function. This is basically done by adjusting w in the direction where the gradient of the loss function is large and negative. Such an optimizer ensures that a local minima of the loss function is found.

The optimizer algorithm and its settings have to be set before the training begins. During the training the loss is evaluated in specific intervals. Usually these intervals are set by the *batch size* which defines how many data samples are computed in the loss function before the optimizer changes the weights. This optimizing of weights to find the minimum loss is an iterative process that ends once an end condition is met. Usually these end conditions are either a certain amount of iterations over the whole training set (called *epochs*) or when the loss of the validation sample reaches a plateau; this is called *convergence* of the model.

The difficulty in optimizing the model lies in the high-dimensional parameter space that is possible. Modern models have often a very high amount of weights (sometimes tens of millions) and the function the model is supposed to approximate is learned from data samples with hundreds of thousands or even millions of entries. Furthermore, many local minima are possible to reach for the loss function in such a high dimensional space. Therefore it needs a sophisticated optimizer to achieve a good model performance.

The simplest *gradient decent (GD)* algorithm updates the weights w according to the equation

$$\begin{aligned} v_t &= \eta_t \nabla_w L(w_t) \\ w_{t+1} &= w_t - v_t. \end{aligned} \tag{5.2}$$

This equation introduces the *learning rate* η_t that determines the step size in the direction of the gradient with which w_t is updated at time step t. For a sufficiently

small η_t $L(w_t)$ *converges* to a local minima. For ML the choice of the learning rate is an important consideration as a small η_t likely reaches a local minima, but at a great computational cost, while a large η_t might not be able of find a local minimum. Therefore one might start with a large η_t and reduces it after a certain amount of time t. This process could be automated like in *Newton's method*.

The GD optimizer implies that the weights are updated based on the whole dataset. One can improve the algorithm by stochastically choose a small subset and apply GD to many of those data subsets, called *minibatches*. This leads to an optimizer called *Stochastic Gradient Descent (SGD)*. With SGD it is less likely that the optimizer gets stuck in a local minima (due to introduction of the stochastic selection of the minibatch) and it is computationally much less heavy.

A popular optimizer that adapts the learning rate without calculating the exact Hessian (like in Newton's methode) is called *Adam* [38]. Adam has been used in this thesis as optimizer for all models. The default starting parameters suggested in [38] were employed including a learning rate $\eta_{Adam} = 0.001$.

To calculate the gradients the optimizer uses another set of algorithms called *backpropagation*. Basically the backpropagation is an implementation of the chain rule for partial differentiation. A nice and simple explanation of backpropagation can be found in [39]. A more detailed explanation of the optimizers as well as backpropagation can be found in [33].

5.1.3 Deep Learning

With the development of modern optimizers such as Adam and high performance computers and GPUs becoming cost effective is became possible to effectively train *deep neural networks (DNN)*. These machine learning algorithms consists of many layers stacked onto each other, hence the term *deep*. These are very successful ML algorithms that are becoming popular in many fields. This lead to *deep learning (DL)* as a subfield of machine learning.

In DL the algorithmic approach is moving towards high dimensional data sets with little preprocessing and sophisticated deep machine learning architectures. The basic idea is that the algorithm shall receive all the available information to solve a given problem. The statistician is thinking more about how the architecture can be designed from a tool box of layers and functions. In the following section ML architectures are introduced.

5.2 Machine Learning Architectures

Today's advanced machine learning algorithms are often variations of *artificial neural networks (ANNs)*. A basic ANN consists of one or multiple layers of 'artificial neurons' stacked onto each other. A basic distinction is made between *visible* and *hidden* layers. The visible layers are the input and output layers, basically the first and the last layer of the model. All layers that are in between are 'hidden' inside the model. This hierarchical structure of network layers is called *model architecture*.

With complex problems to predict it is often not known a-priori which model architecture leads to the best performance result. Therefore in many studies (as well

as in this thesis) multiple models are compared. Usually the figure of merit for this comparison is the loss on the test sample.

In the following sections commonly used machine learning layers are introduced.

5.2.1 Fully Connected Layer

A basic representation of an *artificial neural network (ANN)* is shown in figure 5.1. The ANN is inspired by biology in the sense that it consists of artificial neurons and connections (synapses) in between those. The neurons are connected to all other neurons of the neighbouring layers. The strength of these connections are trainable parameters called *weights*. The neurons itself perform a very simple computation as each neuron (sometimes called *node*) computes a weighted sum of the neurons in the previous layer it is connected to. An additional trainable parameter, called *bias*, is added to to each neuron.

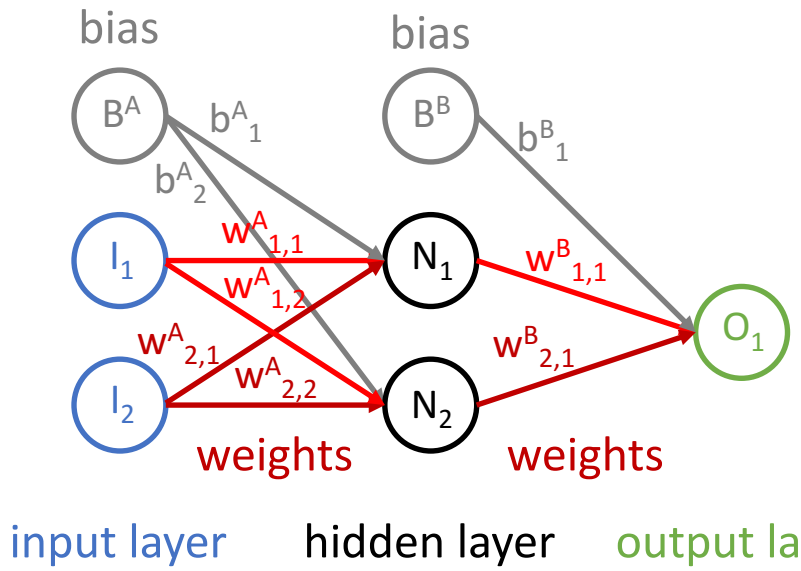


Figure 5.1: Illustration of a simple artificial neural network with two input values, one hidden layer with two neurons, one output node and bias nodes. The information flow from the input to the output is depicted with arrows.

The calculation performed in each neuron is defined by the equation

$$f(x_i, w_i, b) = b + w_i \cdot x_i \tag{5.3}$$

with x_i as the values of the neurons in the last layer, w_i as the trainable weights connected to x_i and b as the trainable bias node. The operation $\cdot$ implies the dot product in vector multiplication.

For the practical example in figure 5.1 this implies that the following computations are performed in neuron N_1 and N_2 of the hidden layer and in O_1 as the output value

of the ANN:

$$\begin{aligned} N_1 &= b_1^A + (I_1 w_{1,1}^A + I_2 w_{2,1}^A) \\ N_2 &= b_2^A + (I_1 w_{1,2}^A + I_2 w_{2,2}^A) \\ O_1 &= b_1^B + (N_1 w_{1,1}^B + N_2 w_{2,1}^B) \end{aligned} \tag{5.4}$$

The example here is given with one hidden layer of two neurons. This one hidden layer is called a *fully connected layer*. A larger ANN might consists out of multiple fully connected layers stacked behind each other in information flow direction with each layer consisting out of several or even hundreds of neurons. If many layers are stacked onto each other we speak of a *deep neural network (DNN)*.

Following the linear equation 5.3 a model consisting of only these layers could but model a linear function. To introduce non-linearities into the model *activation functions* are used. These functions are applied to each neuron and are therefore affecting the next layer. One can write the operation of each neuron with an activation as $f_{act}(f(x_i, w_i, b))$. Commonly used activation functions are introduced in the next section.

Furthermore, the layers can be connected to layers with different computations performed, such as convolutional layers or locally-connected layers introduced below.

5.2.2 Activation Functions

With an activation function (AF) in between layers a non-linearity can be introduced into the model. This way the model can be fitted as a non-linear function to a dataset. Usually activation functions are simple non-linear functions without any trainable parameters. A few popular activation functions are presented in figure 5.2.

For many machine learning tasks the activation function of choice is the *Rectified Linear Unit (ReLU)* function (figure 5.2a) as it is very easy to compute. With a enough neurons in a layer and a ReLU activation function any (non-)linear function can be modelled. A visual proof of this statement can be found in [39]. An alternative is the *Leaky ReLU* function (figure 5.2b) which has non-vanishing gradients over the full space of possible inputs. Whenever Leaky ReLU was used in this thesis, the parameter is set to $\alpha = 0.01$.

The *Sigmoid* function (figure 5.2c) can be used in the last network layer for a binary classification. For completion a *linear activation function* (figure 5.2d) is mentioned which basically corresponds to no activation function use at all. For regression problems the last layer often should be a linear function.

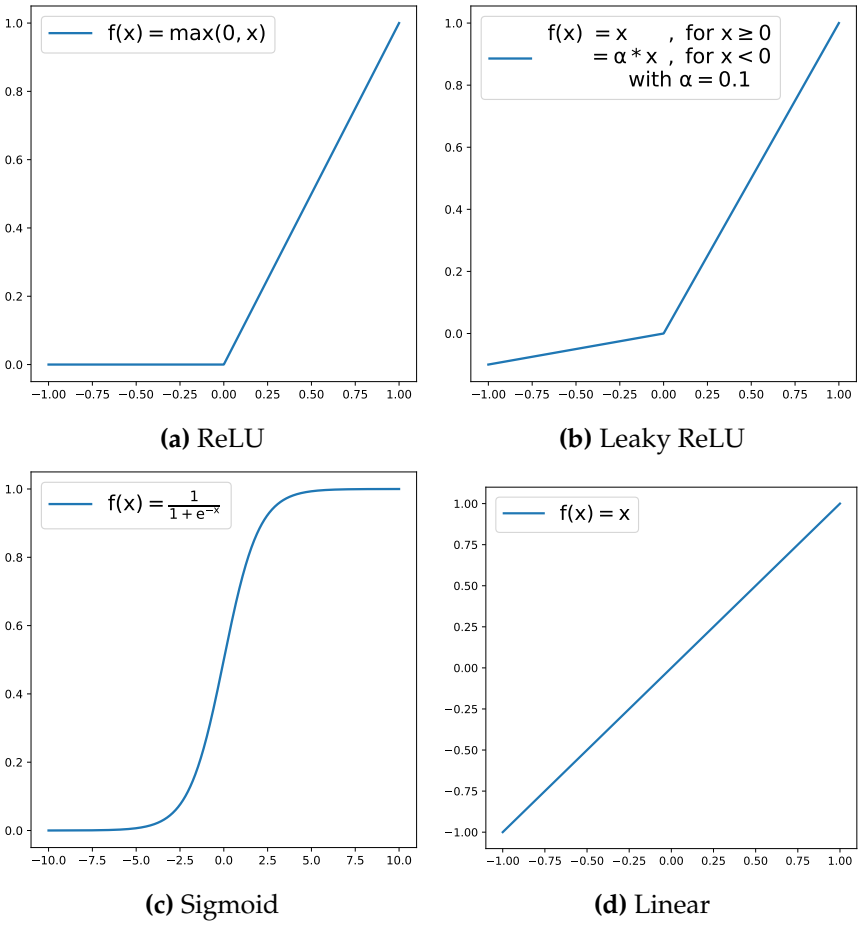
For classification problems one additional function is worth mentioning: The *Softmax* function, that is defined by

$$f(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \tag{5.5}$$

for i categories. This function computes a normalized probability distribution for each input value x_i.

5.2.3 Convolutional Layer

Many datasets, such as images of real world objects, posses symmetries and inherent structure. These symmetries can be exploited by specific layers structures. In com-



(a) ReLU **(b)** Leaky ReLU

(c) Sigmoid **(d)** Linear

Figure 5.2: Four activation functions used in neural network architectures.

parison to fully-connected layers, the number of weights in a layer can be reduced by using translational invariances. This way a neural network is easier to train. One layer type that uses spatial information is a *convolutional* layer. These layers can learn special features in the data. They are especially useful for image recognition. Networks employing convolutional layers are called *Convolutional Neural Nets (CNNs)*.

In CNNs the locality of the input data is preserved, in opposition to fully connected networks that see all input data as 1D vectors. A CNN is built basically from two kinds of layers: Convolutional layers which calculate a convolution of a spatial part of the data set with a set of *filters* (also called *kernels*) and *Pooling* layers which reduce the information in the data set while keeping its spatial structure.

For 2D data, such as images, a layer l is expressed with three dimensional information: hight H_l, width W_l and depth D_l. The height and width correspond to the spatial dimensions of the filter in the 2D data plane. The depth gives the number of filters used in layer l. The depth is also called *channel* or *colour* dimension as the layer can be thought of as multiple filters of different colours.

In a convolutional layer every filter consists of trainable weights giving one layer a total number of weights equal to H_l x W_l x D_l. Additionally one *bias* node is added to each filter. In each filter the weights are the same at each spatial position, hence

they are *shared weights and bias*. The convolution of filters with the input data implies running the filter over the data on every possible position and the output neuron is calculating the dot product between input data and (H_l, W_l)-neuron plane. Afterwards a non-linearity in the form of activation functions is usually applied to the output neuron. The convolution leads to a reduction in the spacial dimension of the image which can be countered by enhancing its dimensions with zeros around the image before the convolution is applied. This method is called *zero padding*.

Pooling layers are used to reduce the spatial dimensions of a data set and filter the important parts. In CNNs pooling layers are often used behind convolutional layers. A popular pooling layer is consists just of 2x2 neurons and output the maximum value in the applied area. This is called *maximum pooling (max pool)*. A modern CNN consists of multiple convolutional and pooling layers often in alternating order. These layers might then be followed by multiple fully connected layers ending in the output.

Here the examples for convolutional layers were given for a set of 2D data, but the convolutional filters can be expanded to cover any dimension, i.e. 3D images. A more detailed discussion on CNNs and the corresponding equations can be found in [33] and [39].

5.2.4 Locally Connected Layer

Convolutional layers have been proven to be very powerful when it comes to detection of objects in images. CNNs might be trained to find cats and dogs in images. In this case it does not matter where in the image the animal is located and it makes senses to use a convolutional filter for this task as the cats pixels are localized together. However, if the task requires that the exact location of an object in the image is known, a generalization of convolutional layers can be used: *locally connected layers*. With locally connected layers an independent set of weights are trained for each filter position. This means that a specific set of weights would be trained to detect an cat in the upper left corner of an image and another set of weights to find one in the lower right corner. This is called *unshared* weights and bias. As in physical problems often the location of a feature is equally important as the feature itself, it might be useful to employ locally connected layers instead of convolutional ones. However, as a new set of weights are trained at each filter position, the amount of weights per layer are much higher for locally connected layers than for convolutional ones of the same filter size.

Multiple layers of locally connected layers with a 1D filter size of 1, but with a depth of several filters can be used to virtually create independent fully connected networks for every data point. Each output neuron $O_{i,b}$ follows a very similar function as equation 5.3, but includes a channel dimension:

$$O_{i,b} = b_i^i + I_{i,a} \cdot w_{i,a}^b \tag{5.6}$$

with $I_{i,a}$ as the input neurons in a input channels and weights $w_{i,a}^b$ and bias b_i^i with b output channels.

5.3 Software Implementations

There are several convenient software implementations of the concepts discussed above. A popular scripting language for neural network implementations is Python. For Python several libraries have been developed to make model creations and its input and output very easy. Examples are *Tensorflow* developed by Google[40], *Theano* developed by the University of Montreal [41] and *PyTorch* by Facebook [42]. For the model implementations in this thesis the Python high-level API *Keras* [43] was used with a backend of Tensorflow.

Chapter 6

Energy Reconstruction with Neural Networks

In this chapter the main results of this thesis are presented. The goal of this analysis is the improvement of the energy resolution of the AHCAL by employing offline software algorithms for the energy reconstruction of an event. The basic energy reconstruction of an event in the AHCAL is the summation over all hit energies $E_{sum} = \sum_i E_i$. Additionally a linear fit is performed to calibrate the E_{sum} to a scale in GeV (see section 4.1.3.1). This reconstructed energy will be referred to as E_{sum} or *standard energy reconstruction* in the following chapter. From a large number of events with the reconstructed E_{sum} for one beam energy the energy resolution for this beam energy can be calculated. For this the metric RMS90 is chosen and the energy resolution is defined as $\sigma_{90}/\overline{E}_{sum,90}$ (see section 4.3).

To enhance the resolution, the energy reconstruction needs to be improved meaning the reconstructed energy E_{reco} should be as close as possible to the beam energy E_{beam}. This could be done i.e. by introducing energy dependent weighting factors that weight the electromagnetic content of the shower lower and the hadronic component higher. This method is called *software compensation* or *offline compensation* and was successfully applied to former CALICE prototype data. Details can be found in [21] and [7]. There are two kinds of software compensation methods: *global* and *local* software compensation. The terms refer to the way the weighting factors are chosen: either on an event basis (global), or on a hit energy basis (local). The software compensation can be seen as another calibration step that should lead to a compensating calorimeter with $e/h \approx 1$ (see section 3.3.2) although the actual AHCAL is non-compensating.

In this thesis machine learning learning algorithms including deep neural network architectures are employed to study how a better performance than software compensation can be reached. For this purpose two approaches have been explored: cell-wise weighing factors and a deep neural network (DNN) architecture. The cell-wise weighting is essentially a recalibration of the calorimeter and is able to compensate for shower leakage. For a DNN architecture a generic fully connected network (FCN) was tested as well as a second architecture with a convolutional layer in front of the FCN.

The training of the network was done with data samples and with Monte Carlo simulated samples. In most high-energy physics ML tasks the labelled data can only be acquired through simulation. With the CALICE test beam data this labelling can be done for data, too - in this case the known beam energy is to assumed to be exact.

Table 6.1: Selection criteria applied to data samples.

E_{beam} [GeV]	nHits	E_{sum} [GeV]	Shower start
10	60 - 200	< 20	4 - 7
15	60 - 200	< 25	4 - 7
20	60 - 250	< 35	4 - 7
30	60 - 350	< 50	4 - 7
40	60 - 450	< 70	4 - 7
50	60 - 500	< 70	4 - 7
60	60 - 600	< 80	4 - 7
80	60 - 700	< 100	4 - 7
120	60 - 900	< 150	4 - 7
160	60 - 1100	< 200	4 - 7

Hence a fully supervised training on experimental data is possible with the test beam data. However, the data taken has limitations due to the limited amount of beam energies recorded. Hence the studies presented here were also performed with MC samples with narrow beam energy steps.

In this chapter first the sample preprocessing is introduced in section 6.1. Afterwards the loss function and the network architectures tested are explained in section 6.2 and 6.4. In section 6.5 the results for data samples can be found and in section 6.6 the results for MC samples. In section 6.6.5 a comparison is made to local software compensation. This comparison summarizes the main results of this thesis and are outlined in section 6.7.

6.1 Sample Preprocessing

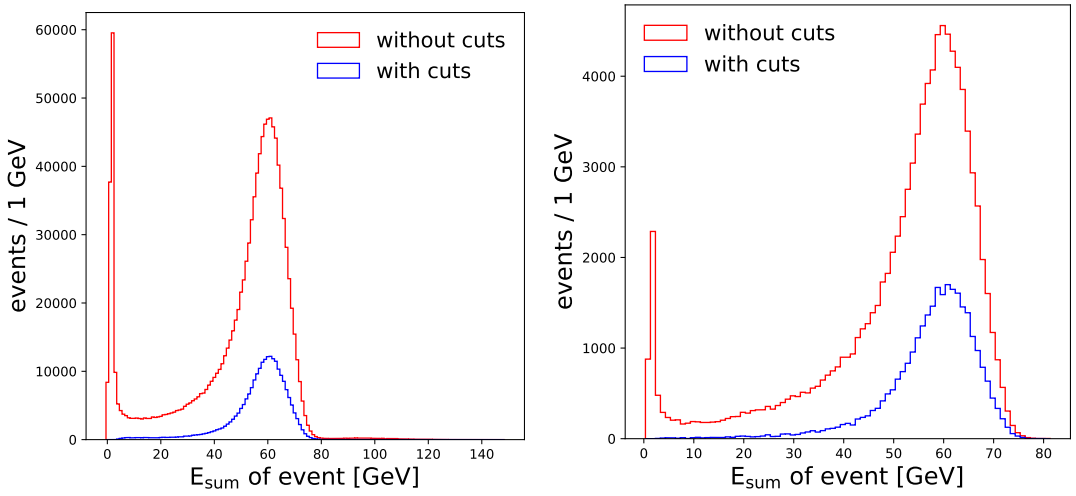
Following the event reconstruction (see section 4.1.4) both data and Monte Carlo samples are processed with certain selection criteria applied and images are created for each event. The selection applied to the data samples are energy dependent and listed in table 6.1. The number of hits (nHits) cut > 60 hits per event is applied to reject muons. The maximum nHits and maximum E_{sum} cuts are applied to reject double particle events. The shower start, namely the first hadronic interaction, is required to happen in layer 4 to 7, this way electrons are rejected since their shower starts mainly in the first three layers.

The following cuts were applied on an event basis to the Monte Carlo samples regardless of energy:

- shower start in the first five layers
- number of hits $\geq$ 60 (to reject 'muon-like' events)

Figure 6.1 shows histograms of the energy sum with and without these cuts applied to the 60 GeV pion data and MC sample. 2D histograms of the number of hits against the centre of gravity in z direction for the 60 GeV data sample can be found in the annex in figure D.1.

After cuts are performed an image is created for each individual event. These 'images' are python arrays with the dimension (24,24,38,2). The first three dimension



(a) Data **(b)** Simulation

Figure 6.1: Histograms of E_{sum} for the 60 GeV pion data and MC sample. The histograms are shown with cuts applied and without.

are the geometric position of a given hit in the event; the i, j, k coordinates. The last dimensions are the energy and time dimension. For each hit at a given (i, j, k) position in the python array the hit energy and the hit time were written. The energy and time for all position where a hit was not recorded in the given event was set to zero. Such an image for a 60 GeV pion event is shown in figure 6.2.

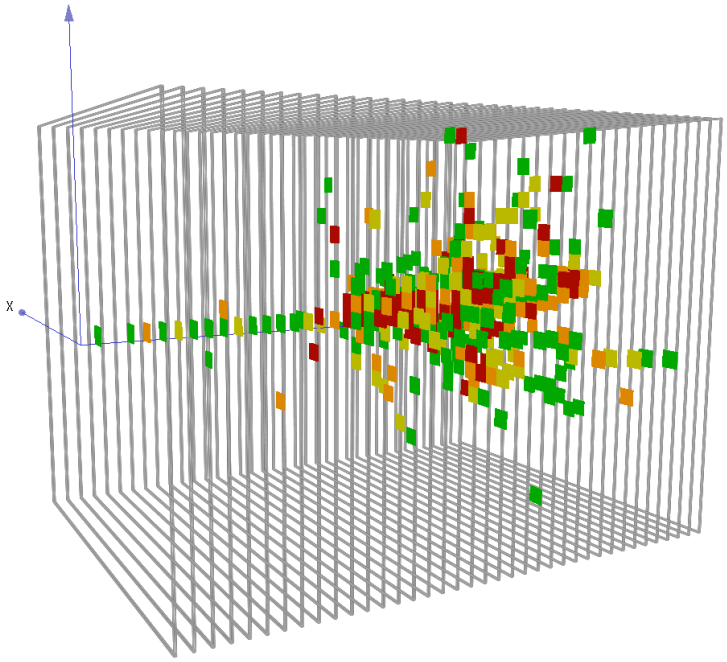


Figure 6.2: Eventdisplay image of a 60 GeV pion event measured during the test beam in May 2018. The colour scale corresponds to the hit energy.

Furthermore, the training was performed in the TeV energy range. The label of the true beam energy is given in TeV and the hit energies were divided by a factor 40,000 ($\approx f_{MIPtoGeV}$ x 1,000). This was done because neural networks perform best in the range $x_{out} \in (-1, 1)$ and because the locally connected network (see below) with weights $= 1$ should approximately reproduce E_{sum}.

The time dimension has only been utilized for studies with the Monte Carlos samples, because the timing calibration was not concluded by the time this thesis was finished. A scaling is applied to the time values as well by converting the values simulated in nanoseconds to microseconds.

To test if the proposed algorithms can reconstruct energies they are not trained on, the samples were split into two sets according to their beam energy labelling. Every second beam energy is stored in a different data set. Afterwards each set is shuffled and split again to create the three data sets used in training and evaluation. The samples were split into three statistically independent data sets according to this splitting: 50 % training set, 20 % validation set, and 30 % testing set. The testing set is larger than the validation set as it is used to determine the performance and the energy resolution for the given beam energy. The performance evaluation is performed with both data sets: the set with the beam energies used for training as well as the set with beam energies the algorithm was not trained on (interpolation set).

For *data* 120k events were chosen for each beam energy. The 10 usable beam energies recorded during the May 2018 test beam campaign were: 10 GeV, 15 GeV, 20 GeV, 30 GeV, 40 GeV, 50 GeV, 60 GeV, 80 GeV, 120 GeV, and 160 GeV. Hence the training data set consists of 6 beam energies: 10 GeV, 20 GeV, 40 GeV, 60 GeV, 120 GeV, and 160 GeV. And the interpolation set consists of the remaining 4 beam energies: 15 Gev, 30 GeV, 50 GeV, and 80 GeV. For the training set this is a total number of events of 6 x 120k x 0.5= 360k.

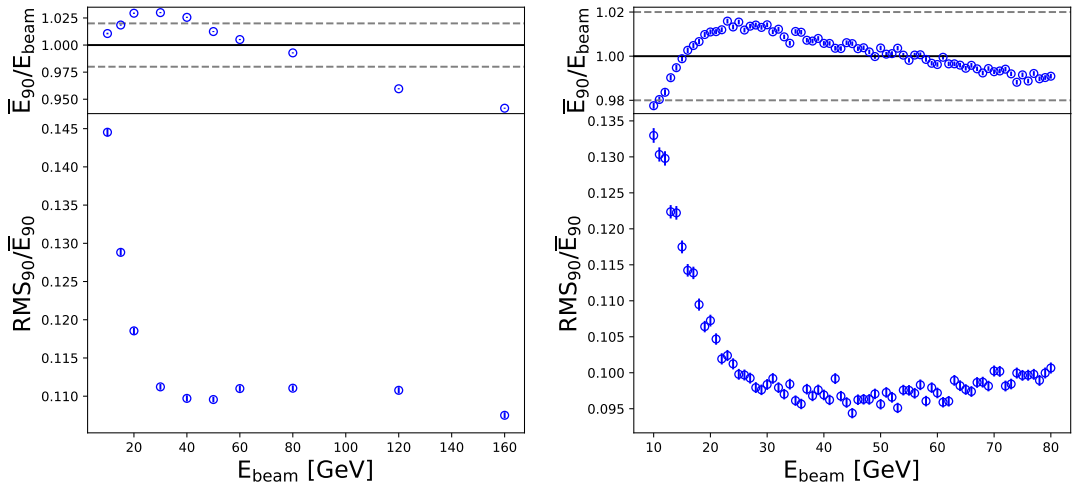
For the *MC sample* a total of 32k events remain after cuts applied for each beam energy. The beam energies simulated are 10 to 80 GeV in 1 GeV steps (see section 4.2.3); a total of 71 beam energies. Following the same procedure as with data the sample was split into two sets of beam energies: one training set with every even value beam energy and a interpolation set (only for testing) with every odd value beam energy. Hence the training set consists of 36 beam energies with a total number of events of 36 x 32k x 0.5 = 576k.

Histograms of the training samples for both data and MC are presented in the annex in figure G.1. For the supervised learning approach followed in this thesis the individual events are labelled with the true beam energy E_{beam}.

The technical implementation is done in Python. The reconstructed *root* files are first converted into *pandas* dataframes stored in the *HDF* file format to be easily imported into Python. In a second step the *HDF* files are loaded into another pandas dataframe and the cuts discussed above are applied. The whole data set is than split into the three sets (training, validation, testing) and stored as another HDF file. These files are finally loaded into the RAM for the training of the algorithms. The image creation is done batch-wise in a generator on the CPU before loading them into the GPU to save RAM.

6.2 Energy Resolution with Standard Energy Reconstruction

As mentioned above the standard energy reconstruction of an event is $E_{sum} = \sum E_i$, the summation over all hit energies E_i in one event. Additionally a linear fit to reach the GeV scale needs to be performed (see section 4.1.3.1). Using the metric RMS90 the energy resolution with E_{sum} can be calculated for each beam energy. The energy resolution and the linearity plotted against the beam energy E_{beam} for the events in both test samples are shown in figure 6.3 for data and Monte Carlos samples. The cuts discussed above were applied and the statistical error on E_{sum} is propagated. In addition to the energy resolution the linearity of the calorimeter response $\overline{E}_{90}/E_{beam}$ needs to be observed and should be around 1 for a linear response. A strong variation from this ratio would lead to a different energy resolution $RMS_{90}/\overline{E}_{90}$.



(a) Data **(b)** Simulation

Figure 6.3: The energy resolution and linearity of the AHCAL test samples calculated with RMS90 for both data and MC simulation.

For the MC samples the the resolution is degrading above 50 GeV. This behaviour is due to longitudinal shower leakage. Furthermore the linearity for MC simulation is observed to be within 2 %. For data an increasing resolution due to shower leakage can be seen for the 60 and 80 GeV runs. However, the beam energies 120 and 160 GeV exhibit a lower energy resolution as well as an underestimated mean energy. This might be due to the desaturation function in the energy calibration not appropriately correcting the non-linear response of the SiPM for high energetic hits.

6.3 Loss Function and Performance Evaluation

The loss function chosen for all network types in this chapter is *mean absolute relative error (MARE)*:

$$L(E_{reco,i}, E_{beam,i}) = \frac{1}{N} \sum_i \left| \frac{E_{reco,i} - E_{beam,i}}{E_{beam,i}} \right| \tag{6.1}$$

with $E_{reco,i}$ as reconstructed energy and the networks' output, $E_{beam,i}$ the true beam energy label and N as the sample size or batch size. In many regression problems

the loss function chosen is the mean squared error (MSE) $\frac{1}{N}\sum_i (E_{reco,i} - E_{beam,i})^2$. However, for the distributions in this thesis MSE is not optimal as thy are get wider with increasing energy. Therefore the relative error is used to roughly weight the error in each beam energy similar in the overall loss. Additionally events with a large invisible energy due to hadronic interactions should not have a much higher impact on the loss than events with visible energy closer to the true beam energy. Therefore the absolute relative error is chosen over the squared relative error. A similar case for MARE is made in [44].

The performance of each network was evaluated based on the test sample (see section 6.1). However the test loss is not necessarily the only figure of merit to judge a 'good' network performance. It is important for an energy reconstruction algorithm to conserve or optimize the linearity of the calorimeter response. Additionally a low energy resolution is important especially for low energies. Therefore for each tested network architecture a plot equivalent to figure 6.3 and a comparison to the standard energy reconstruction is made (see section 6.5 and 6.6).

6.4 Network Architectures

In the following the network architectures evaluated in this thesis are presented. Designing suitable neural network architecture does involve a fair amount of experimenting with different layer structure. Additionally, hyperparameters (see section 6.4.4) can be optimized for each investigated architecture.

Two different types of neural network architectures have been evaluated. They are categorized as *locally connected networks (LCN)* and *deep neural networks (DNN)*. Although a deep LCN can be considered as a DNN, this terminology is used in this thesis to distinguish the two architecture approaches. The *LCN* architecture is based on a channel-wise energy calibration of the calorimeter. The LCN is built from one or multiple layers of 1-dimensional locally connected layers with a filter size of 1 and a depth of one or multiple filters per layer. This is equivalent to training independent fully connected networks for each calorimeter cell to perform a MIP to GeV calibration (see section 5.2.4). The *DNN* architectures investigated include a standard fully connected network (FCN) with multiple layers and a second architecture with an added convolutional layer in front of the fully connected layers (CNN).

Multiple experiments were performed with each architecture type. For each architecture many layer configurations were tested before deciding on the ones presented. Two configurations for each type were evaluated to be representative for the architecture type. Those configurations performed best than similar types tested. This means that the list given in the following is not extensive, but can be seen as a guideline for the choice of an architecture for further energy reconstruction studies.

Additionally, a merged architecture is presented that combines the LCN and the DNN approach into a single architecture.

6.4.1 Locally Connected Networks Architectures

The Locally Connected Network (LCN) approaches are based on a cell-wise recalibration of the calorimeter to improve the energy reconstruction. The network should learn a (non-)linear MIP to GeV conversion function for each individual

Table 6.2: Network architecture of *LC1*

Layer Type	Size	Parameters	Initializer	Comment
LC layer	1x(1)	21,888	ones	no bias
Summation layer	-	-	-	
Total parameters:		21,888		

channel based on how important the channels' hit energy is in the overall energy reconstruction. The basis are one or multiple layers of 1-dimensional locally connected (LC) layers with a filter size of 1 and a depth of one or multiple filters per layer. The last layer of this stack of LC layers is always a layer with a depth of just one filter to generate one output per calorimeter cell value. This way the LCN can be thought of as a individually trained fully connected network for every calorimeter cell with one input and one output node (see section 5.2.4). The reconstructed energy as the output is given by the summation over all output nodes of the last LC layer.

Two types of LCNs are discussed in the following. A LCN with only one LC layer with one filter and no bias that can train exactly one weight per calorimeter cell. This way the network can learn a linear cell-wise calibration function for every cell. The amount of weights equals the number of channels in the AHCAL. This way the network resembles exactly the standard reconstruction if all weights are set to one and the appropriate $f_{MIPtoGeV}$ factor is applied. Therefore, the weights in the LC layer are initialized with the exact value one. A *kernel constraint* was added such that the weights cannot be negative which would imply an unphysical subtraction of a hit energy from the event energy. This network is henceforth called *LC1*. An overview of the LC1 network can be found in table 6.2.

The second LCN type consists of four LC layers with 16 filters in the first three. In between the layers a leaky ReLU activation function is added, therefore this network can learn a non-linear cell-wise calibration function. However, the LC layers do not include a bias as in the experiments run it was not possible to make the network converge with a bias. The training without bias is already a bit sophisticated. To make the network converge the layers need to be trained first with convolutional layers (shared weights) in the same configuration until it reaches the same performance as the standard reconstruction. The calibration hence equals one for each cell. The weight initialization is performed according to [34] ('he_normal). The LCN (unshared weights) is afterwards initialized with the trained weights from the convolutional layers. This way the network can be optimized from the baseline standard reconstruction. This network is henceforth called *LC4*. An overview of the LC4 network can be found in table 6.3.

A third LCN type was applied only to the MC sample as the time dimension was added. The basis is the LC1 architecture, but timing information was added in the form of three convolutional layers with kernel size of 1 that are only attached to the time dimension of the input images. This way, similar to the first training stage of LC4, a cell-wise time calibration function can be learned that is the same for every channel. The cell-wise LC1 output is multiplied with the time calibration factor from the timing convolutional network. The overall output, the reconstructed energy, is again the summation over all calibrated channels. This network was named *LC1+time*. An overview of the LC1+time structure can be found in table 6.4.

Table 6.3: Network architecture of *LC4*

Layer Type	Size	Parameters	Initializer	Comment
LC layer	16x(1)	350,208	he_local_normal	no bias
Leaky ReLU	-	-	-	$\alpha = 0.01$
LC layer	16x(1)	5,603,328	he_local_normal	no bias
Leaky ReLU	-	-	-	$\alpha = 0.01$
LC layer	16x(1)	5,603,328	he_local_normal	no bias
Leaky ReLU	-	-	-	$\alpha = 0.01$
LC layer	16x(1)	350,208	he_local_normal	no bias
Summation layer	-	-	-	
Total parameters:		11,907,072		

Table 6.4: Network architecture of *LC1+time*

Layer Type	Size	Parameters	Initializer	Comment
For energy channel:				
LC layer	1x(1)	21,888	ones	no bias
For time channel:				
Convolutional layer	128x(1)	256	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
Convolutional layer	64x(1)	8,256	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
Convolutional layer	1x(1)	64	he_normal	no bias
Merge energy & time:				
Multiplication layer	-	-	-	
Summation layer	-	-	-	
Total parameters:		30,464		

Table 6.5: Network architecture of *FCN*

Layer Type	Size	Parameters	Initializer	Comment
FC layer	32	700,448	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	33	he_normal	no bias
Total parameters:		703,649		

The standard implementation in Keras for LC layers was deemed to be insufficient as it is comparably slow and does not allow sufficient flexibility. Therefore a custom layer for a 1-dimensional LC layer with a kernel size of 1 was written to be used in Keras. Apart from running much faster than the standard Keras layer, the layer allows to set weights and bias to be (non-)trainable parameters and to create shared weights making the layer applicable as a convolutional layer. Furthermore the Keras implementation of the weight initialization according to [34] (in Keras called 'he_normal') was modified to allow proper application in case of a kernel size of 1 and multiple filters. This modified initializer is given the name *he_local_normal*.

6.4.2 Deep Neural Network Architectures

Results for two deep neural network (DNN) architectures are presented in this thesis. The first architecture is a fully connected network (FCN) with multiple layers and the second one a convolutional neural network (CNN) as an extension of the first architecture. Other similar architectures were experimented with, too, but these two are representative for the possible performance found with an FCN or CNN architecture.

The FCN consists of five fully connected layers (also called *dense* layers) with the first four made up of 32 neurons each and the last layer with one output neuron. In between the layers a non-linearity is introduced with leaky ReLU activation function. Experiments with more or less layers as well as wider layers have not resulted in significant performance difference. The exact network architecture is presented in table 6.5.

The CNN evaluated adds one convolutional layer before the FCN. This way it consists of six layers in total. An overview of the network can be found in table 6.6. The convolutional layer consists of 3-dimensional kernels with a size of (7,7,38). This way the kernels x- and y-dimension mirror those of a typical hadronic shower core, while the depth equals the number of calorimeter layers. As no zero-padding is used, the output of the filter is a 2D representation of the 3D calorimeter image - excluding the channel dimension. Trainings were run with different kernel sizes, but less wide as well as shorter kernels show a worse performance. The same is true for multiple of these layers as well as 2D convolutional layers in between the 3D layer and the FCN.

Table 6.6: Network architecture of *CNN*

Layer Type	Size	Parameters	Initializer	Comment
Convolutional layer	128x(7,7,38)	238,464	he_normal	no padding
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,327,136	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	33	he_normal	no bias
Total parameters:		1,568,801		

6.4.3 Merged Architecture

Finally both approaches, the LCN and the DNN approach were merged into a single architecture by stacking them and removing the last summation layer in LC1+time. This way the timing information is utilized by the CNN architecture and the CNN can improve upon the channel wise energy calibration trained in the LC1 architecture. To make this improvement directly possible the weights in the LC1+time part of the network are initialized with already trained weights; only the weights in the CNN part are random initialized. The fully merged architecture is named *LC1+time+CNN* and its structure is presented in table 6.7.

6.4.4 Hyperparameters

As *hyperparameters* one describes the non-trainable parameters in the neural network setup. These parameters need to be set before the training is performed. For all networks in this thesis the following hyperparameters have been applied if not otherwise stated:

- batch size = 128
- learning rate in Adam = 0.001
- maximum number of epochs = 200
- early stopping epochs = 5

  (if in a window of five epochs the validation loss is not decreasing, the training is stopped)

6.5 Results for Data Samples

In the following the results from the energy reconstruction for data samples with the described neural network architectures are presented. The test loss for both test

Table 6.7: Network architecture of *LC1+time+CNN*

Layer Type	**Size**	**Parameters**	**Initializer**	**Comment**
For energy channel:				
LC layer	1x(1)	21,888	pre-trained	no bias
For time channel:				
Convolutional layer	128x(1)	256	pre-trained	
Leaky ReLU	-	-	-	$\alpha = 0.01$
Convolutional layer	64x(1)	8,256	pre-trained	
Leaky ReLU	-	-	-	$\alpha = 0.01$
Convolutional layer	1x(1)	64	pre-trained	no bias
Merge energy & time:				
Multiplication layer	-	-	-	
Convolutional layer	128x(7,7,38)	238,464	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,327,136	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	1,057	he_normal	
Leaky ReLU	-	-	-	$\alpha = 0.01$
FC layer	32	33	he_normal	no bias
Total parameters:		1,599,265		

Table 6.8: Test loss for all network architectures for the data samples.

Architecture	Loss (trained)	Loss (not trained)
LC1	0.116	0.114
LC4	0.118	0.115
FCN	0.051	0.243

samples for the given network architectures can be found in table 6.8. The test loss for LC1 and LC4 are comparable with the loss for LC1 being slightly lower. For both architectures there is but a slight difference for the two test samples. The test loss for the FCN architecture however is very different for the two test samples, much lower for the trained on sample and much higher for the not trained on sample in comparison to the LCN architectures. This difference is explained below.

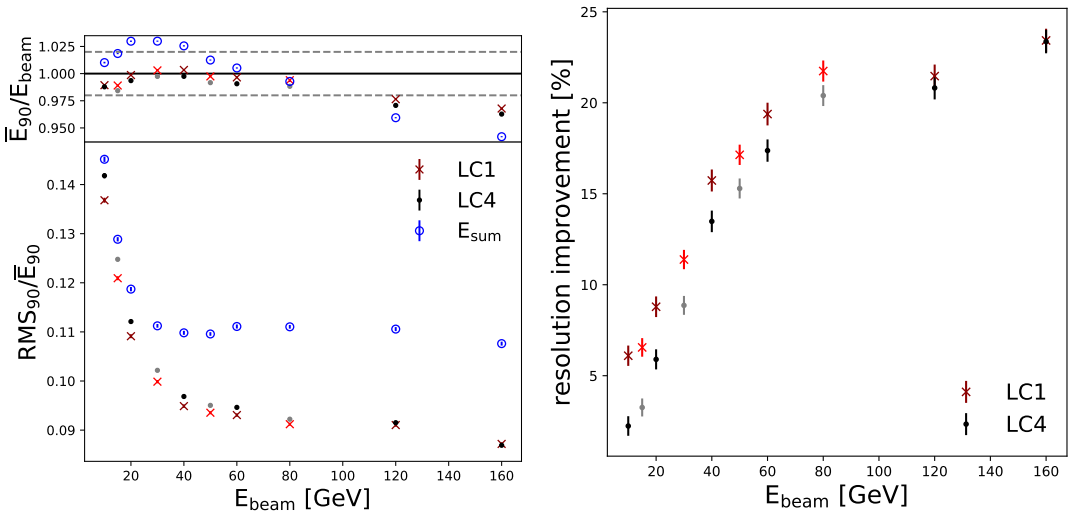
6.5.1 Locally Connected Architectures

The energy linearity and resolution for the energy reconstruction with both LCN architectures, LC1 and LC4, are shown together with the standard reconstruction E_{sum} in figure 6.4a. It is apparent that both architectures show no to systematic differences between the 'trained on' and 'not trained on' test samples. Furthermore, both networks lead to an improved energy linearity and resolution over the whole energy range. This improved resolution is compared with E_{sum} and shown in figure 6.4b. The LC1 architecture shows a better resolution with an improvement over E_{sum} of about 6 % for low energies and up to 24 % for 160 GeV (where the performance of LC1 and LC4 is the same). The histograms of the reconstructed energy by network LC1 for all energies in either test sample is shown in the annex in figure G.2. No systematic difference in the histogram shape is visible for the two test samples.

The slightly worse performance for LC4 is likely due to the non-linear cell-wise calibration functions the network is able to learn. It appears that the simple linear cell-wise calibration in LC1 is easier to optimize resulting in a lower test loss. In general the improvement over E_{sum} can be explained by the cell-wise hit energy weighting that is performed by the networks. Some of the weights for LC1 are plotted in figure E.1 in the annex. The weights are comparatively high for cells in the centre of the detector in the last two layers. This is understandable as all beams and showers are centred in the AHCAL's (i,j)-plane and shower leakage effects can be offset by weighting the last layers higher than the rest in the overall weighted hit energy summation. This explains the up to 24 % improved resolution for high energies in comparison to the standard reconstruction as well as the general trend that the energy resolution of the reconstructed energies improves with the beam energy E_{beam}.

6.5.2 Deep Neural Network Architecture

The fully connected network (FCN) architecture cannot be used to reconstruct the event energy properly. Figure 6.5 shows the histograms for the reconstructed energy for both test samples. It shows that the deep network architecture with many weights leads to overfitting on the limited amount of data beam energies. The 'trained on' true beam energies are precisely learned while the 'not trained on' energies cannot



(a) Energy linearity & resolution

(b) Resolutions in comparison to E_{sum}

Figure 6.4: Linearity and resolution for the reconstructed energy with both locally connected networks for data samples. **(a)** shows the linearity and resolution for E_{reco} with LC1 and LC4 for both test samples as well as the standard reconstruction E_{sum}. **(b)** shows the resolution improvement in comparison to E_{sum}. The colour coding per network architecture differentiates the two test samples.

be reconstructed properly. This reconstructed energy structure is the same for other tested deep neural network architectures as well as the CNN architecture.

This result lead to experiments with training on Monte Carlo simulated samples with a tight beam energy spacing. With many more and closer spaced beam energies to train it is possible to utilize the DNN architectures.

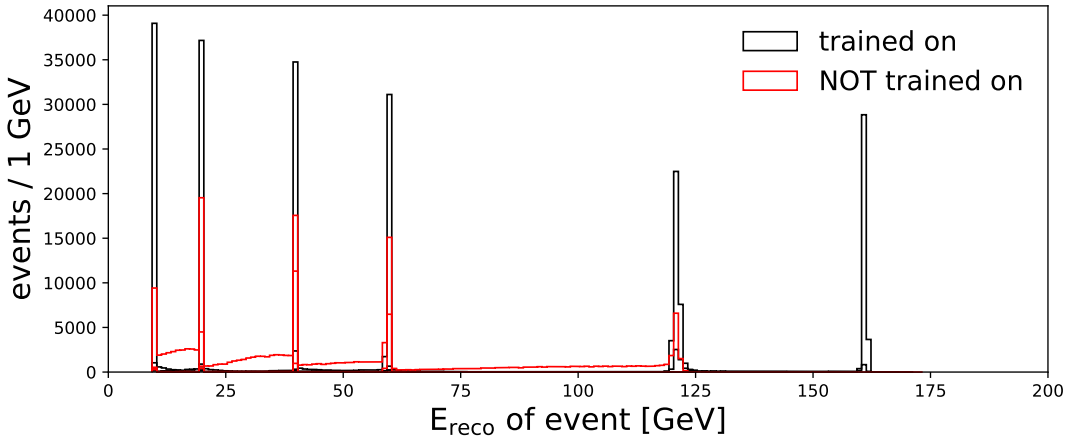


Figure 6.5: Histograms of the reconstructed energy E_{reco} by the FCN network for all energies in either data test sample.

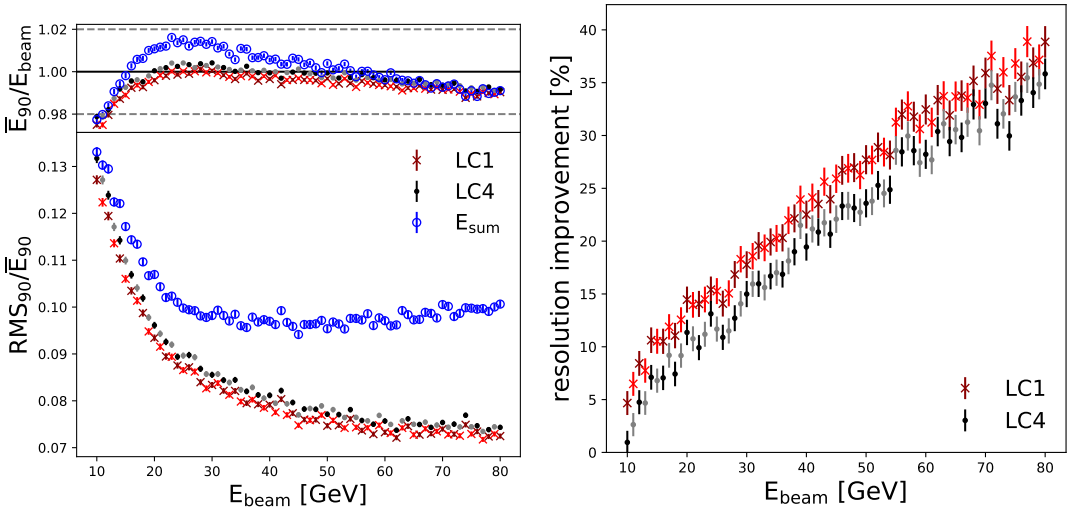
Table 6.9: Test loss for all network architectures for the MC samples.

Architecture	Loss (trained)	Loss (not trained)
LC1	0.092	0.092
LC4	0.094	0.094
FCN	0.087	0.087
CNN	0.078	0.079
LC1+time	0.089	0.088
LC1+time+CNN	0.076	0.075

6.6 Results for MC Samples

In the following the results from the energy reconstruction for Monte Carlo samples with the described neural network architectures are presented. The test loss for both test samples for the given network architectures can be found in table 6.9. The DNN architectures exhibit lower losses than the LCN ones, with the lowest loss achieved with the CNN architecture. Adding the timing information improves both the LC1 approach as well as the CNN. In general there is almost no difference between the two test samples.

6.6.1 Locally Connected Architectures



(a) Energy linearity & resolution

(b) Resolutions in comparison to E_{sum}

Figure 6.6: Linearity and resolution for the reconstructed energy with both locally connected networks for MC samples. **(a)** shows the linearity and resolution for E_{reco} with LC1 and LC4 for both test samples as well as the standard reconstruction E_{sum}. **(b)** shows the resolution improvement in comparison to E_{sum}. The colour coding per network architecture differentiates the two test samples.

The energy linearity and resolution for the energy reconstruction with both LCN architectures, LC1 and LC4, are shown together with the standard reconstruction E_{sum} in figure 6.6a. The resolution improvement in comparison to the standard

reconstruction is plotted in figure 6.6b. The same behaviour as for data is apparent for the MC sample training: There is no systematic difference between the test samples, linearity for both networks is within -2 % - 0 %, and the LC1 architecture performs slightly better than LC4 with an improvement of the resolution between 5 % for 10 GeV and about 40 % for 80 GeV. The same explanation for this network performance as for the data results applies (see section 6.5.1).

6.6.2 Deep Neural Network Architectures

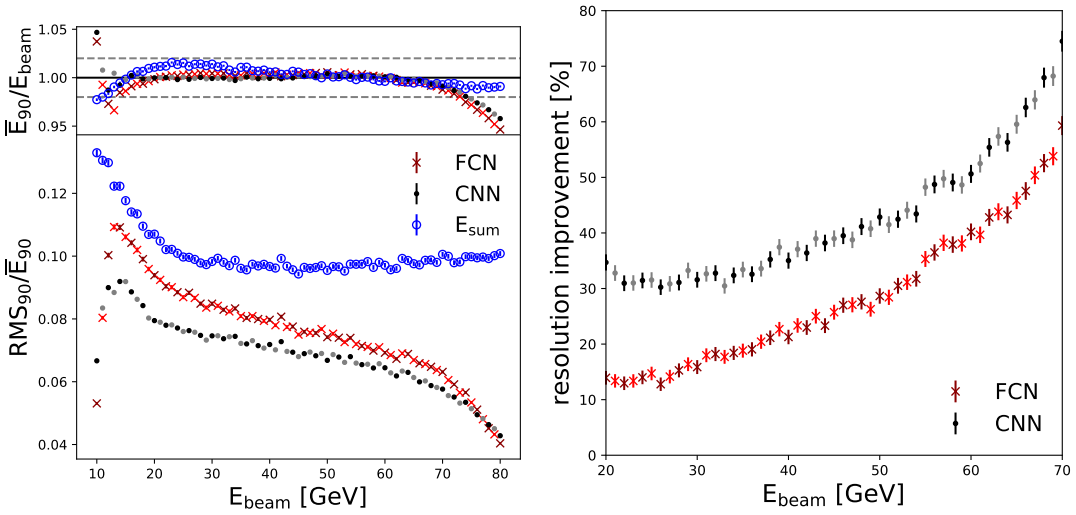
Histograms of the reconstructed energy with the CNN architecture are shown in the annex in figure G.3. In this plot the histograms for each beam energy in the 'trained on' test sample are shown by plotting $E_{reco} - E_{beam}$ to overlay all histograms. Five histograms are specifically labelled to show the noticeable feature of the DNN architectures: overfitting at the edges of the parameter space. The deep networks learn the beginning and the end of the trained E_{beam} space and therefore the events for the beam energies around 10 GeV and 80 GeV are largely reconstructed at precisely this energy. Hence for those energies the RMS90 is very low which cannot be considered a 'good' reconstruction performance but rather is an artefact of the training. The histograms between 22 to 68 GeV appear not to show this skewed behaviour. Therefore the networks reconstruction performance can be evaluated roughly between 20 - 70 GeV. This overfitting feature is the same for both evaluated DNN architectures.

The energy resolution plots for the two DNN architectures, the fully connected network (FCN) and the convolutional neural network (CNN), can be found in figure 6.7. In the beam energy region between 20 and 70 GeV the calorimeter response is linear. There is no systematic difference between the two test samples. The DNN architectures are reconstructing the event energy for 'trained on' and 'not trained on' beam energies which is largely due to the low energy spacing of 2 GeV in the training sample. Especially for low energies, the CNN architecture is clearly superior to the FCN. While for the FCN the resolution improvement over the standard reconstruction is between about 15 % for 20 GeV and 50 % for 69 GeV, it is for the CNN between 30 % and 70 % in the same energy region.

Both DNNs show a better performance than the locally connected networks. This is understandable as the DNNs have access to global event information as well as spacial information and in general many more parameters to optimize. The CNN is superior to the FCN which is due to the simplified input to the fully connected network that is created by the 3D convolutional layer. The layer produces many 2D images from the 3D event image which appears to support an optimized energy reconstruction. Of the four network architectures compared with only hit energy images as the networks input, the CNN shows the best energy resolution and the lowest test loss.

6.6.3 Application to Data Samples

The LC1 and the CNN architecture were trained on MC samples and show an improved energy reconstruction in comparison to E_{sum}. Here the on MC trained networks are applied to a data test sample, too. The data test sample consists of 50k events per energy with all available energies between 10 and 80 GeV. There is a systematic difference between the MC samples and the data (see section 4.2.3) and hence a correction



(a) Energy linearity & resolution

(b) Resolutions in comparison to E_{sum}

Figure 6.7: Linearity and resolution for the reconstructed energy with both deep neural networks for MC samples. **(a)** shows the linearity and resolution for E_{reco} with FCN and the CNN for both test samples as well as the standard reconstruction E_{sum}. **(b)** shows the resolution improvement in comparison to E_{sum}. The colour coding per network architecture differentiates the two test samples.

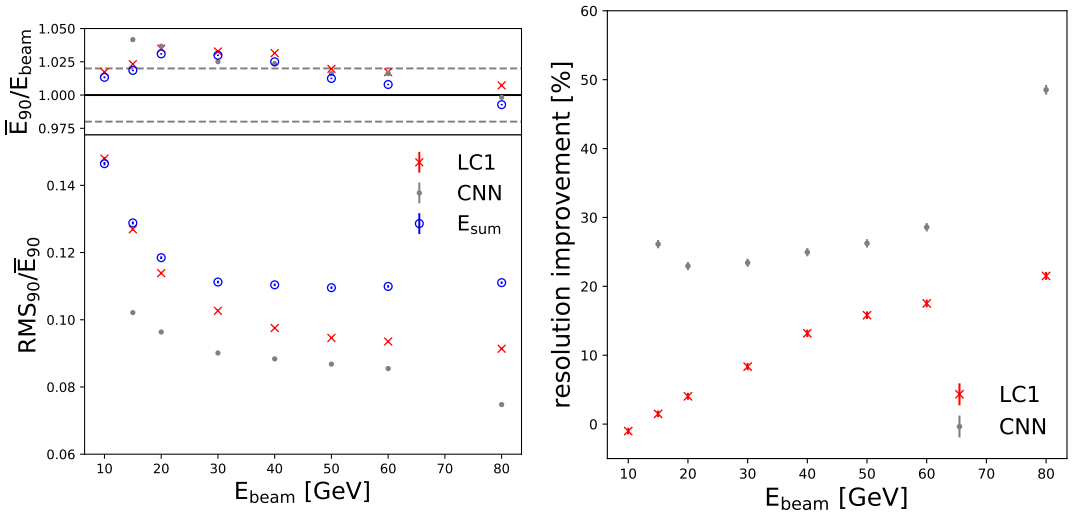
factor $f_{MIPtoGeV,MC}/f_{MIPtoGeV,data} = 1.17$ is applied to the networks' reconstructed energy. The resulting energy resolution plots are shown in figure 6.8.

For all energies, except for 10 GeV, the energy resolution by the LC1 network is better than the standard reconstruction. As seen above the resolution improvement increases to up 20 % at 80 GeV. However, this is a smaller resolution improvement than for the LC1 network trained and tested on data (see section 6.5.1).

The CNN on the other hand is not straightforward applicable to data. The overfitting on the edges of the parameter space is visible in the E_{reco} histogram for every energy. The histograms of $E_{reco} - E_{beam}$ for all energies is shown in the annex in figure G.4. The skewed distributions are especially noticeable for low energies such as 10 and 15 GeV, but a E_{reco} feature at 10 GeV is visible for all energies. For this reason the 10 GeV values were not plotted in figure 6.8. Interestingly for 80 GeV the distribution is not as skewed as for the MC sample. In fact the mean energy is reconstructed very close to the beam energy. However, whether the distributions is affected by the training overfitting will only be possible to evaluate if the CNN is trained up to higher energies than 80 GeV.

6.6.4 Including Timing Information

In addition to the hit energy, the MC samples include the hit time information. This information is used in the network architectures LC1+time and LC1+time+CNN. The resolution plots for both architectures are shown in figure 6.9. Adding the time information improves the energy resolution in comparison to the architectures without time by about 5 % for all energies. The non-linear time calibration function the convolutional part of the LC1+time network learns is shown in the annex in figure F.1.



(a) Energy linearity & resolution

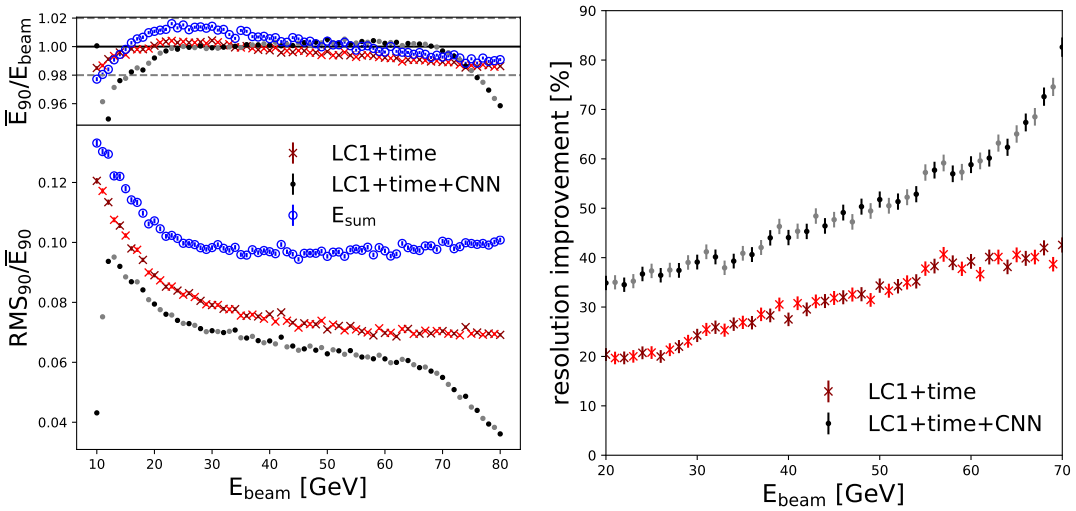
(b) Resolutions in comparison to E_{sum}

Figure 6.8: Linearity and resolution for the reconstructed energy with the LC1 and CNN architecture trained on MC, but applied to the data. **(a)** shows the linearity and resolution for E_{reco} with LC1 and the CNN as well as the standard reconstruction E_{sum}. **(b)** shows the resolution improvement in comparison to E_{sum}.

For each training run the calibration function looks different, but for all three trainings shown the factor increases for later hits and the slope changes at about 8 ns and 20 ns. The increasing factor makes physically sense as late hits indicate a hadronic part of the shower, i.e. late neutron energy depositions (see section 3.1.2). To compensate this less measured hadronic part in the under-compensating AHCAL, late hits are weighted with a higher factor (see section 3.3.2). The slope changes are due to the amount of statistics available in these timing regions. The overfitting at the beam energy edges is again apparent for the CNN.

6.6.5 Comparison to Local Software Compensation

The best resolutions are reached with the LC1 and CNN architectures (with timing). In the following those reconstruction methods are compared with *local software compensation*. The local software compensation algorithm was developed by the CALICE collaboration to improve the energy resolution of various prototypes in the past. The compensation algorithm applies a number of energy-dependent weights to hit energies in a binned energy spectrum. This way low energetic (and more likely hadronic) hits are weighted stronger than high energetic (and more likely electromagnetic) hits in the total energy summation of one event. The implementation here relies on eight hit energy regions with each weight parametrized by three energy dependent variables. This results in a total of 24 trainable weights. Their energy dependence is determined by the standard reconstructed energy E_{sum}. The local software compensation is explained in more detail in [7] and [21]. The algorithm here was optimized on the whole MC sample with cuts applied for the beam energies 10, 15, 20, 25, 30, 40, 50, 60, 70, and 80 GeV and 36k events per beam energy. No differentiation between a



(a) Energy linearity & resolution

(b) Resolutions in comparison to E_{sum}

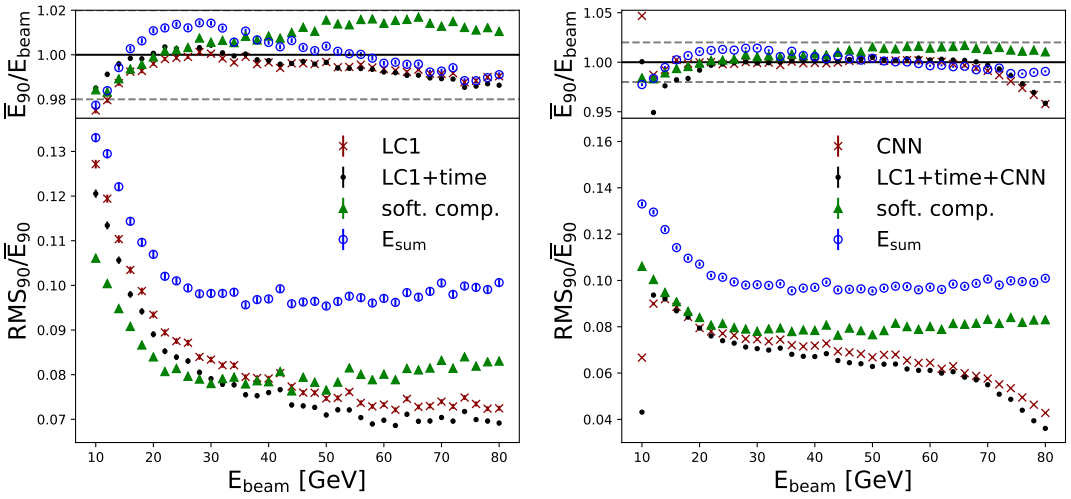
Figure 6.9: Linearity and resolution for the reconstructed energy with the LC1+time and LC1+time+CNN architecture for both MC test samples. For these networks the hit timing information is used in addition to the hit energy. **(a)** shows the linearity and resolution for E_{reco} as well as the standard reconstruction E_{sum}. **(b)** shows the resolution improvement in comparison to E_{sum}. The colour coding per network architecture differentiates the two test samples.

training sample and a testing sample is made [1].

The resulting energy linearity and resolution of the reconstructed energy with local software compensation for the test sample is shown in figure 6.10 plotted together with both locally connected architecture (LC1 and LC1+time) as well as the convolutional architectures (CNN and LC1+time+CNN). As was described before, the neural network algorithms utilizing timing reach a better energy resolution than without timing. Nonetheless, local software compensation reaches a better energy resolution than both LCN architectures for low energies below 35 GeV. The LCN architectures reach a lower energy resolution above this energy as those utilize spacial information and compensate for the shower leakage that limits the standard reconstruction as well as the local software compensation. While the LCN architectures use spatial shower information, but with energy independent weights, the local software compensation with its energy dependent weights is utilizing the event energy to reach a better energy resolution for low energies. The exact improvement of the resolution for each algorithm over the standard reconstruction can be found in the annex in figure G.5. The local software compensation improves the energy resolution by to 25 % over the E_{sum}, while it improves with LC1 by 5 % (10 % with timing).

A fair comparison between the CNNs and software compensation can only be made in the beam energy region between 20 and 70 GeV because of the overfitting outside this range. For these energies the networks' resolution is improved by a few % for low energies up to 60 % for high energies in comparison to software compensation. This improvement is due to the fact that the CNNs utilize global event information as

[1] private communication with Jack Rolph, University Hamburg; work based on [7] and [21]



(a) LCN architectures **(b)** CNN architectures

Figure 6.10: Linearity and resolution for the reconstructed energy with different networks in comparison to local software compensation for the 'trained on' test sample. LC1+time and LC1+time+CNN architecture for both MC test samples. **(a)** shows the linearity and resolution for the locally connected architectures (with and without timing). **(b)** shows the linearity and resolution for the convolutional architectures (with and without timing).

well as spatial shower information to compensate for leakage. This way the CNNs reach a similar performance to software compensation for low energies (showers largely contained in the AHCAL) and a much better resolution for higher energies (where shower leakage limits the resolution).

6.7 Summary and Outlook

Four different network architectures for energy reconstruction were examined and trained on data and Monte Carlo samples. Only the locally connected architectures are successfully trained on data, as the beam energies for training are too sparse for deep neural network architectures not to overfit. Most of the results have been obtained for the training on MC samples. The training on many beam energies with a spacing of 2 GeV seems to be successful for interpolating energies in between. In future test beam campaigns events could be recorded with the same beam energies to perform a similar study with real data. The sample size of 16k events per beam energies are sufficient for the training with about 40 beam energies.

In conclusion, a similar or better energy resolution than local software compensation can be reached by using the CNN architecture. A reconstruction algorithm that includes a shower leakage compensation improves the energy resolution for high energies, as shown with the locally connected architectures. Furthermore, energy dependent weights in an algorithm appear to be better than energy independent ones, as both software compensation and the DNN architectures reach good results with using event-level information. It was shown that the hit timing information can be used in neural network architectures to improve the energy resolution. The exact performance of the CNN approach should be investigated further by extending

the training energy range as well as the Monte Carlo accuracy. Reconstruction of data samples was performed with MC trained networks, but the performance might improve with a closer similarity between MC and data. An uncertainty here is that the network trained on MC was not trained on exactly the same beam energies as it was tested on with data.

Systematic uncertainties in the calorimeter samples have not been taken into account. Only the statistical errors were propagated in the results shown above. Uncertainties that result from the energy and time calibration chain will be subject to further studies in the collaboration. Another cause of error in the AHCAL samples presented here are fluctuations in the hadronic content of the shower as well as sampling fluctuation. With the limited sample sizes used in this study these fluctuations are the cause of small variations in the energy resolution for the MC samples for close beam energies.

When the neural networks are trained, the weights are random initialized. Therefore slight variations in the network performance and the test loss can occur over multiple trainings as the optimizer finds one of multiple local minima of the loss function. However, the algorithms presented here are deterministic once trained and applied to reconstruct the energy. Meaning that uncertainties in the sample data is directly propagated through the network.

Chapter 7

Conclusions & Outlook

An engineering prototype for an fine-granular analogue hadron calorimeter (AHCAL) was assembled by the CALICE collaboration in 2018. This AHCAL prototype consists of 38 active layers with each 576 SiPM-on-tile cells of 30 x 30 mm^2 size resulting in a total of 21,888 calorimeter channels. As passive material 2 cm stainless steel absorber were used in the sandwich structured calorimeter. It is the largest AHCAL prototype developed to date and was designed for a scalable mass production. Each event recorded includes 5-dimensional information: the 3D location, energy and timing of a each calorimeter hit.

The prototype underwent several test beam campaigns at DESY and CERN in 2018 and 2019. In this thesis a focus is on the negative pion test beam data recorded in May 2018 at SPS. The data includes runs at 11 beam energies between 10 and 160 GeV. A data quality analysis was performed evaluating the overall quality of runs to be very good, although the 100 GeV runs were not usable for studies of the energy resolution due to an open collimator in the beam line and the low energy runs suffer from electron contamination. In addition to the data, a Monte Carlo simulation of the test beam setup was developed by the collaboration. For both data and simulation samples a hadronic shower start was required to occur in the early calorimeter layers. Nonetheless for increasing beam energies the limitation of the energy resolution through shower leakage is apparent.

The energy resolution of the AHCAL for pion events is under investigation in this thesis. Several event energy reconstruction algorithms based on neural network architectures are compared with the standard reconstruction and an established local software compensation algorithm. The standard energy reconstruction denotes to the summation over all calibrated cell energies of one event ($E_{sum} = \sum E_i$). Local software compensation is a n algorithm that reconstructs the energy by applying event energy dependent weighting factors to a binned hit energy spectrum.

The supervised machine learning algorithms explored in this thesis are based on deep neural network architectures. Four architectures were studied, two with locally connected layers, one multi layer fully connected network and a convolutional neural network (CNN). In addition architectures are implemented that apply a cell-wise weighting factor based on the hit timing. Training of the neural networks is done on test beam data as well as on Monte Carlo samples. As the data suffers from muon, electron and multi-particle contamination, basic cuts are applied to reject those events. For training and testing of the networks two sets of data are used: six beam energies

are used for training and testing, and in addition four different energies are sorted in a sole testing set. This way it was tested whether the trained networks can reconstruct energies they were not trained on. The same procedure was undertaken for training on Monte Carlo samples with a tight energy spacing. The training was performed on a set of 36 beam energies between 10 and 80 GeV with a 2 GeV spacing, the testing on 71 beam energies with a 1 GeV spacing.

With the locally connected network architecture, which introduces cell-wise energy independent weighting factors, a better energy resolution than with the standard reconstruction is achieved. The resolution improvement is between 5 % for 10 GeV and 40 % for 80 GeV for the Monte Carlo samples. The increasing performance results from the inherent shower leakage compensation as cells in the last detector layers are weighted stronger. Above 40 GeV this architecture shows an improved resolution over local software compensation. Hence in general, event energy dependent weights, like in software compensation, as well as a shower leakage correction, like in the locally connected network, are powerful algorithmic tools to improve the energy reconstruction.

Of the deep neural network architectures explored, the CNN with a large kernel size offers the most promising results. However, it is not possible to properly train on the limited beam energies of the data sample due to overfitting. The overfitting occurs for training on Monte Carlo simulation, too, outside of the 20 to 70 GeV window. Inside this window the accuracy of the reconstruction is very good. For 20 GeV the CNN improves the resolution by a few percent in comparison to software compensation, for 70 GeV by 60 % due to additional shower leakage correction. The CNN has access to spatial and global event information resulting in this large resolution improvement.

Adding the timing information with a 1 ns smearing as a cell-wise calibration factor to the locally connected architecture increases its resolution performance by 5 - 8 % for all beam energies in the Monte Carlo sample. The CNN can be improved by merging it with the cell-wise output of the locally connected network including time. This merged architecture utilizing the whole 5-dimensional event information achieves the best resolution of all compared reconstruction algorithms.

The exact performance of the CNN approach should be investigated further by extending the training energy range as well as the Monte Carlo accuracy. The local software compensation could be improved by combining it with a shower leakage correction and usage of hit timing information. In future test beam campaigns pion runs with a tight beam energy spacing could be recorded to perform trainings of DNNs with data similar to the training performed here with Monte Carlo samples.

In conclusion, a deep learning based energy reconstruction offers a promising approach to improve the achievable energy resolution in future linear collider facilities.

Appendix A

Code for RMS90

Code A.1: Python code that calculates the mean and the standard deviation/rms and their errors of the smallest window in which 90 % of the distirbution x is contained.

```
def calc90(x):
        x = numpy.sort(x)
        n10percent = int(round(len(x)*0.1))
        n90percent = len(x) - n10percent
        for i in range(n10percent):
        rms90_i = numpy.std(x[i:i+n90percent])
        if i == 0:
        rms90 = rms90_i
        mean90 = numpy.mean(x[i:i+n90percent])
        mean90_err = rms90/numpy.sqrt(n90percent)
        rms90_err = rms90/numpy.sqrt(2*n90percent)
        elif i > 0 and rms90_i < rms90:
        rms90 = rms90_i
        mean90 = numpy.mean(x[i:i+n90percent])
        mean90_err = rms90/numpy.sqrt(n90percent)
        rms90_err = rms90/numpy.sqrt(2*n90percent)
        return mean90, rms90, mean90_err, rms90_err
```

Appendix B

May 2018 test beam events

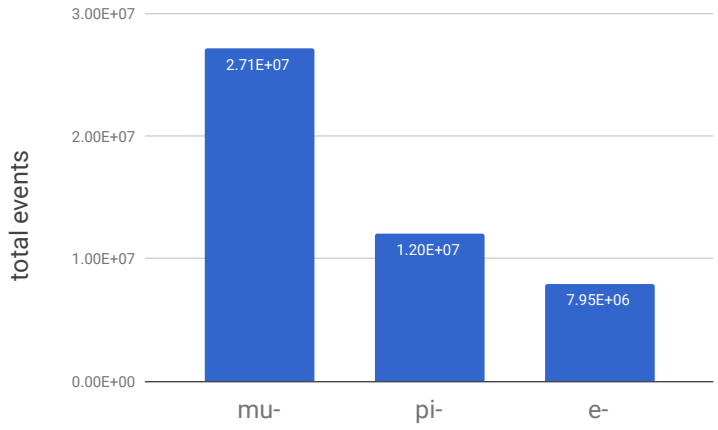


Figure B.1: Total number of events of muons, pions and electrons recorded during the May 2018 test beam campaign per particle. Different particle type contamination, noise triggered events and double particle events are not taken account for.

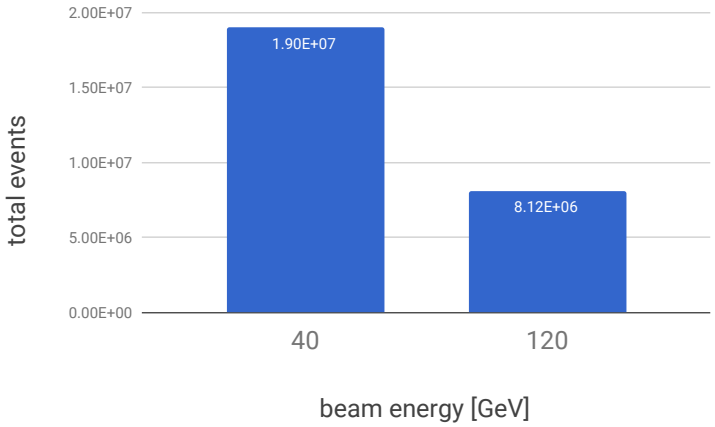


Figure B.2: Total number of muon events of recorded during the May 2018 test beam campaign. Including position scan runs. Different particle type contamination, noise triggered events and double particle events are not taken account for.

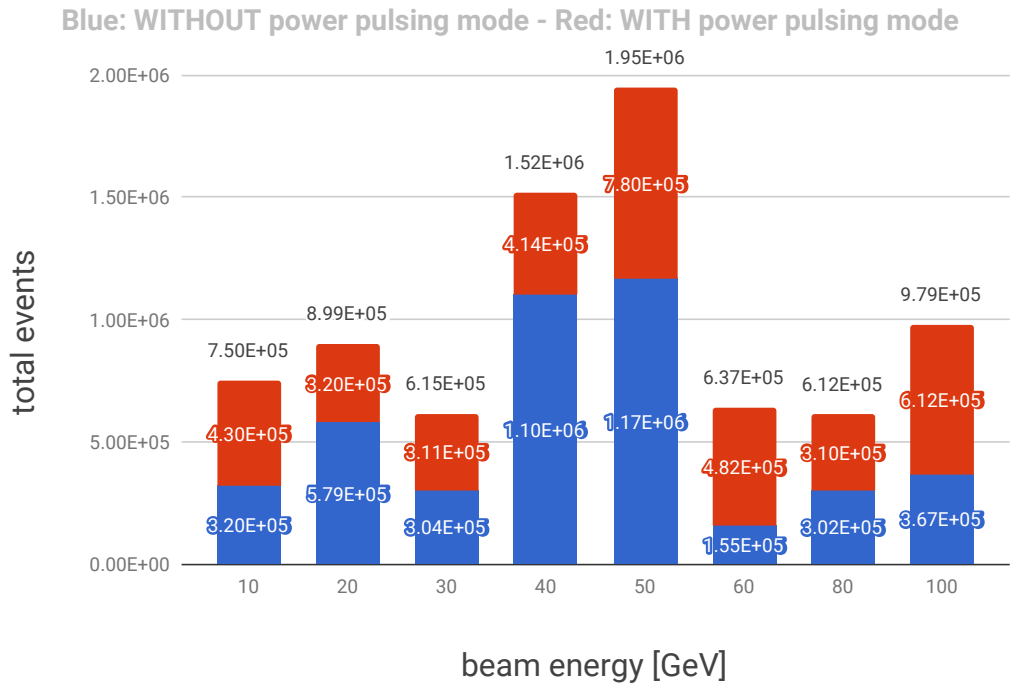


Figure B.3: Total number of electron events of recorded during the May 2018 test beam campaign. In red the runs in power pulsing mode are shown. Different particle type contamination, noise triggered events and double particle events are not taken account for.

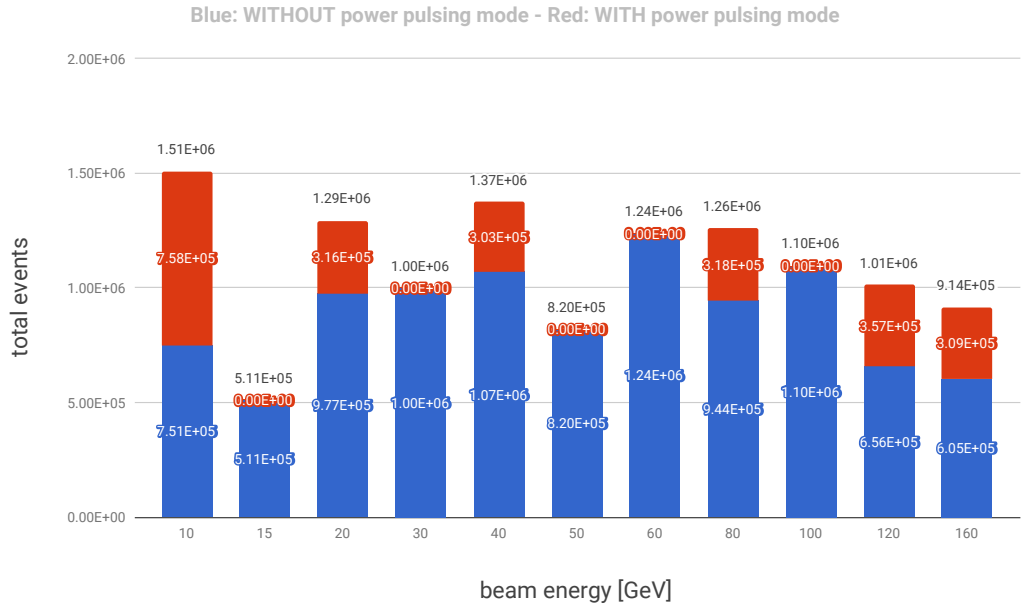


Figure B.4: Total number of pion events of recorded during the May 2018 test beam campaign. In red the runs in power pulsing mode are shown. Different particle type contamination, noise triggered events and double particle events are not taken account for.

Appendix C

Data Quality Analysis for May 2018 Test Beam Data

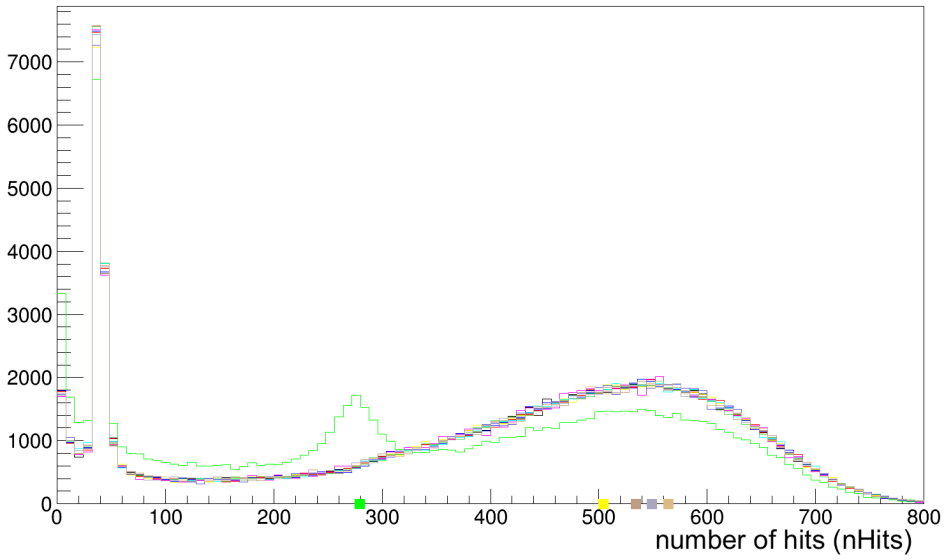
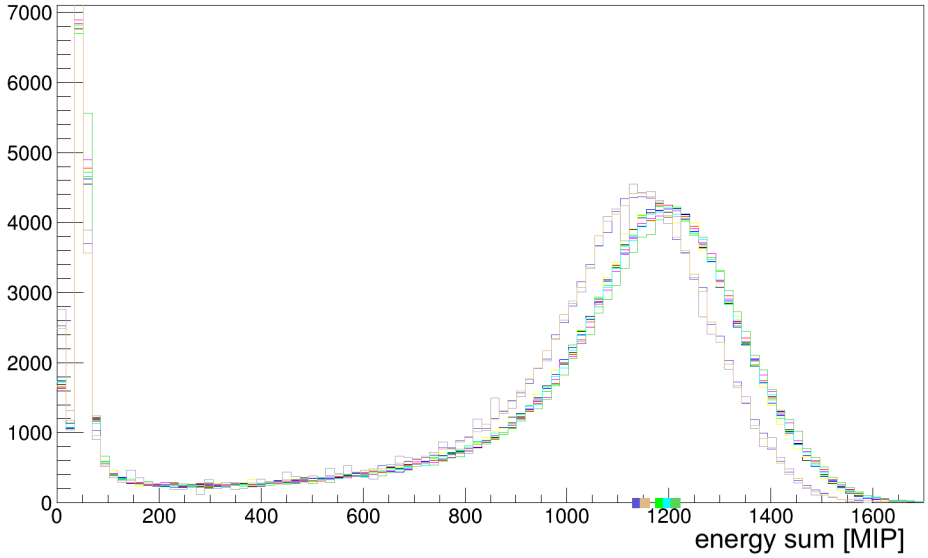
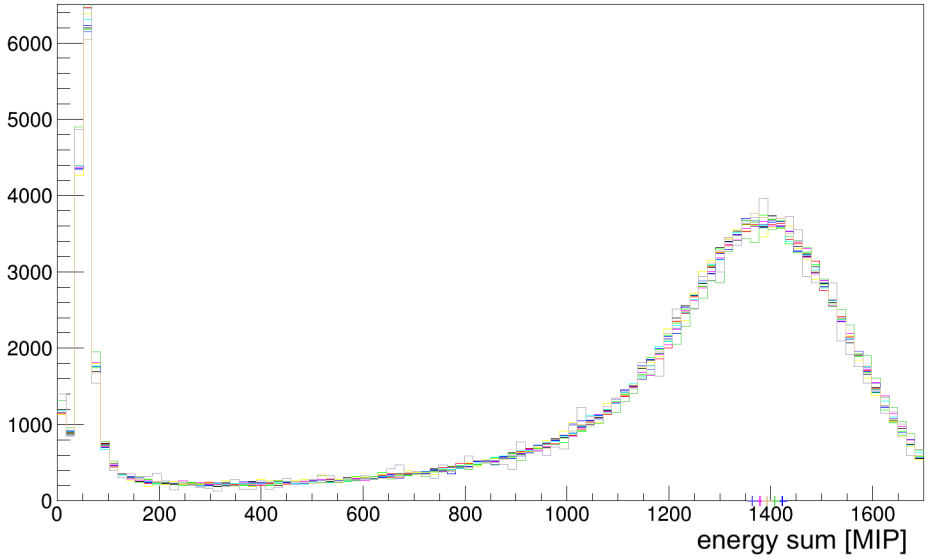


Figure C.1: All histograms of the number of hits per event for all standard 100 GeV pion runs recorded. The outlier (run 60766 in light green) is clearly noticeable.



(a) same calibration constants for both power modes



(b) power mode specific calibration constants

Figure C.2: Energy sum ($\sum E_i$) distributions for all 40 GeV pion runs measured during the May test beam campaign. Figure (a) shows a shift in the energy sum to lower energies for runs with power pulsing when the same calibrations are applied to all runs. Figure (b) shows the same runs but with power mode specific calibrations constants applied. The peak bin position is marked on the x-axis for all runs.

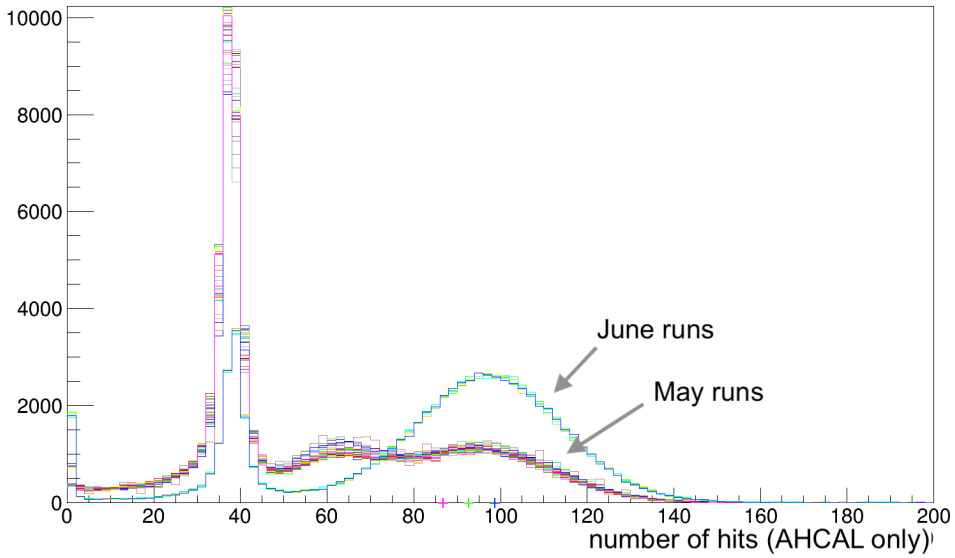


Figure C.3: All histograms of the number of hits per event for all standard 10 GeV pion runs recorded in May and June 2018. May runs and June runs and marked accordingly. The more pronounced peak of the June distributions result from optimized beam line configurations that minimized electron contamination.

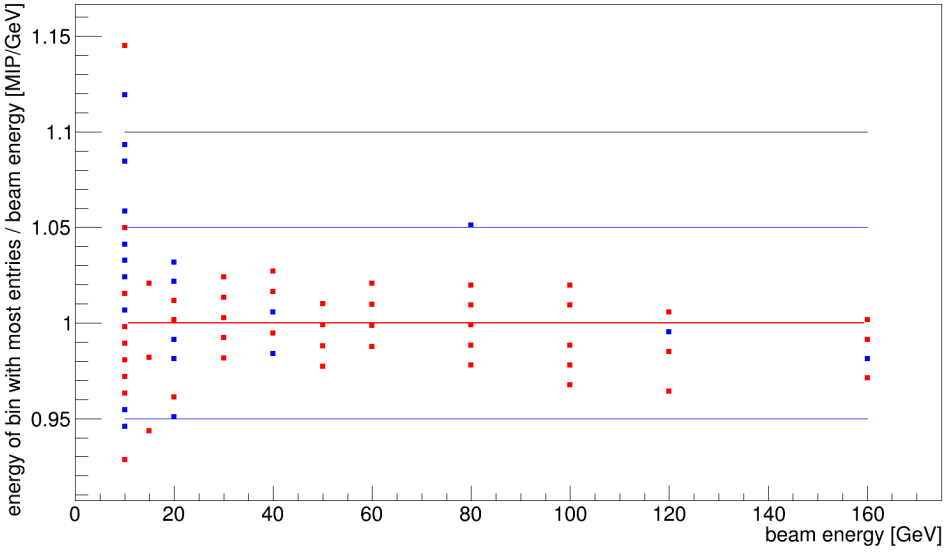
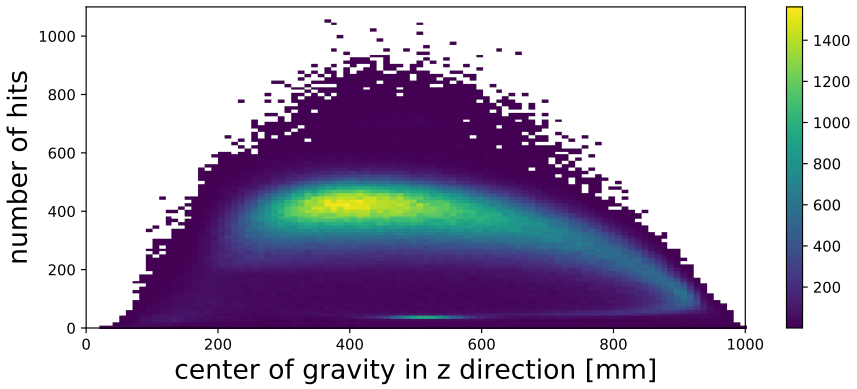


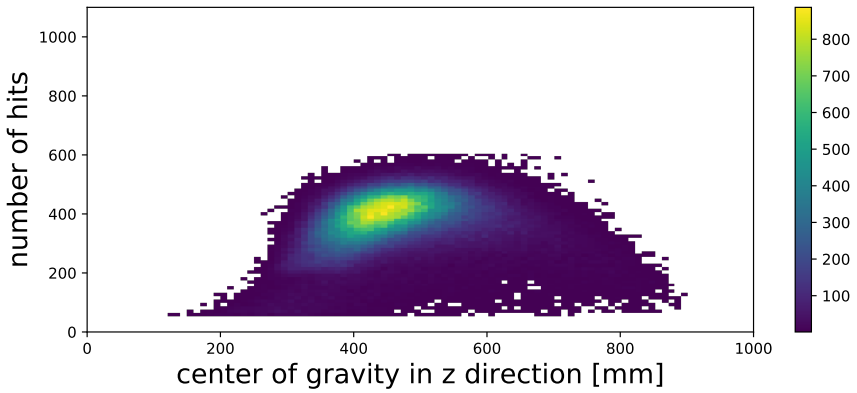
Figure C.4: Overview plot for the data quality analysis. Peak bin position of energy sum histogram divided by the corresponding beam energy plotted against the beam enery for all 150 runs taken during the May 2018 test beam. Due to binning effects several data points overlay in this plot.

Appendix D

Data Cuts for 60 GeV Pions



(a) no cuts

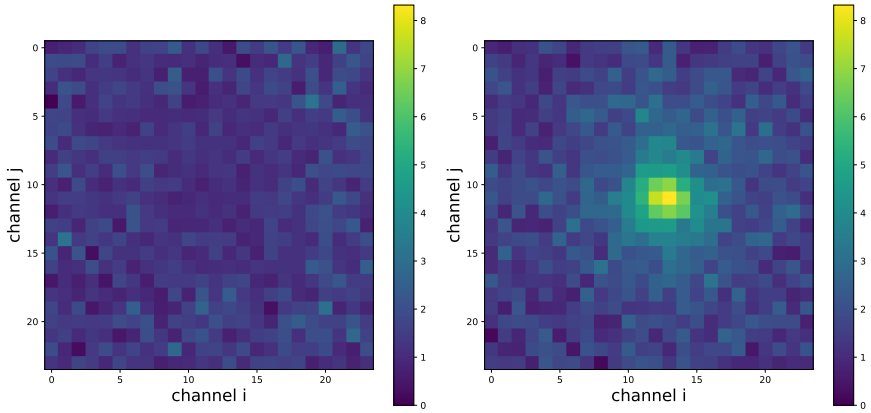


(b) cuts applied

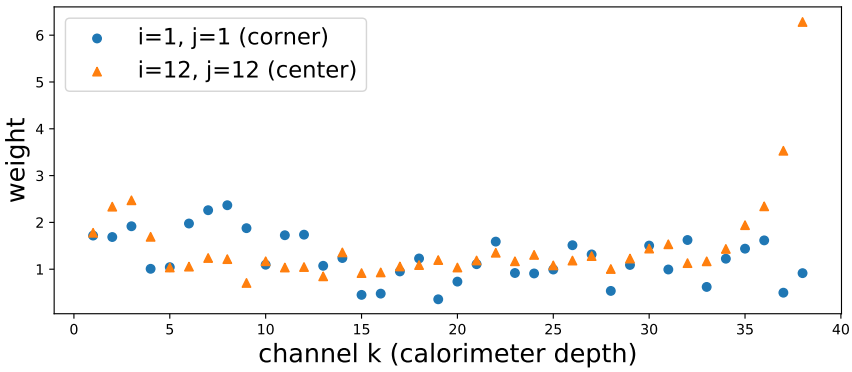
Figure D.1: 2D histograms for number of hits plotted against the centre of gravity in z direction for 60 GeV pion data samples. **(a)** shows the distribution of all 60 GeV runs without any cuts applied, figure **(b)** shows the samples with the cuts applied (see 6.1).

Appendix E

LC1 weights for data



(a) Weights of layer 19 **(b)** Weights of layer 38



(c) Depth-wise weights

Figure E.1: Visualizations of the trained LC1 weights. **(a)** shows the weights in the (i,j)-plane for detector layer 19 (middle layer) and **(b)** shows the weights for layer 38 (last layer). **(c)** shows weights for all cells in one corner and for the centre of the AHCAL plotted against the calorimeter depth k.

Appendix F

Time Calibration Factor

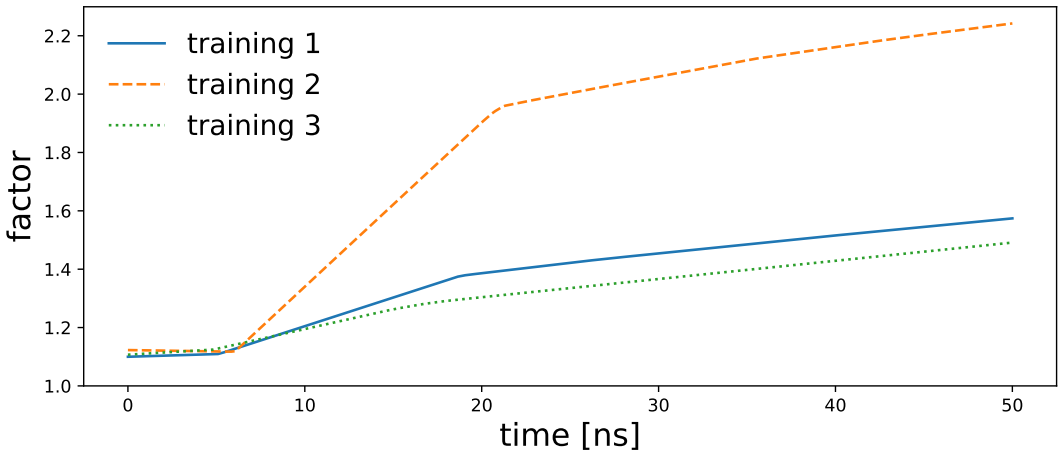
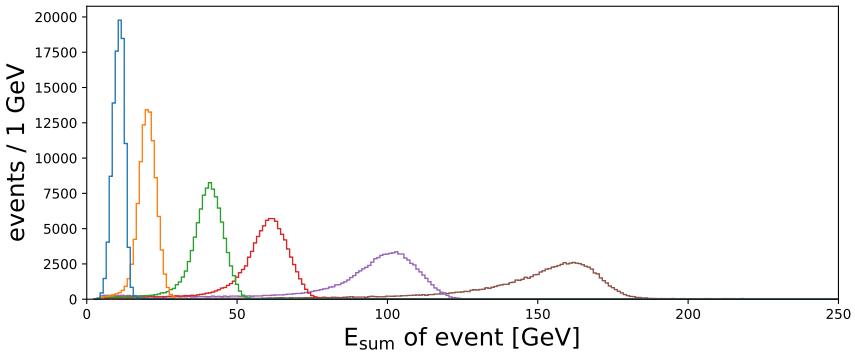


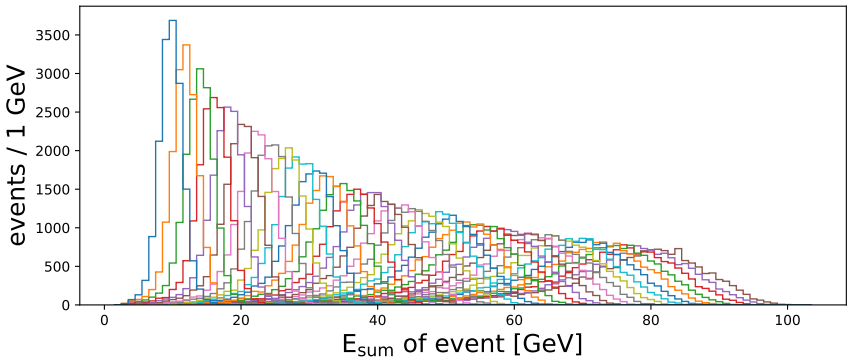
Figure F.1: Calibration factor based on the hit time learned by the convolutional network part in LC1+time. The timing calibration function is slightly different with each training.

Appendix G

Additional Energy Reconstruction Results



(a) Data



(b) Simulation

Figure G.1: Histograms of E_{sum} of the whole training samples for both data and MC simulation. The bin width was fixed to 1 GeV. The colour coding corresponds to the beam energy labels of each event (in the text).

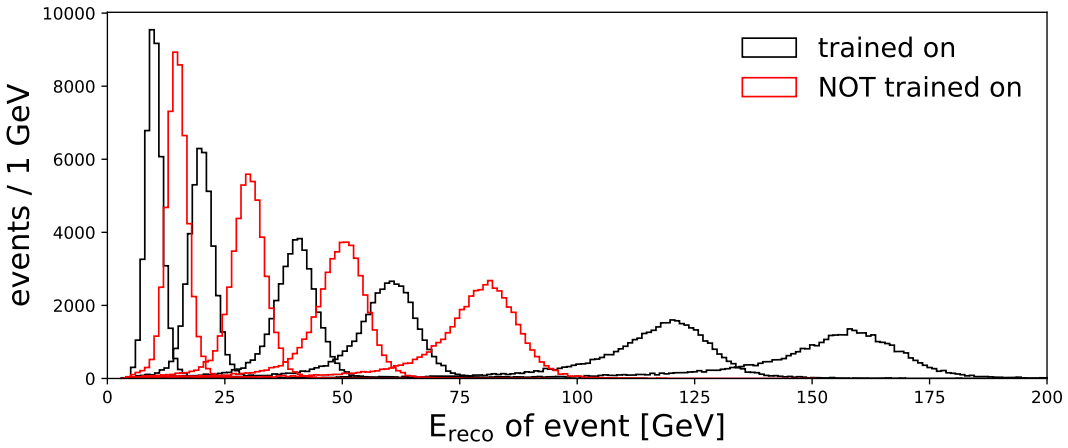


Figure G.2: Histograms of the reconstructed energy E_{reco} by network LC1 for all energies in either test sample of data.

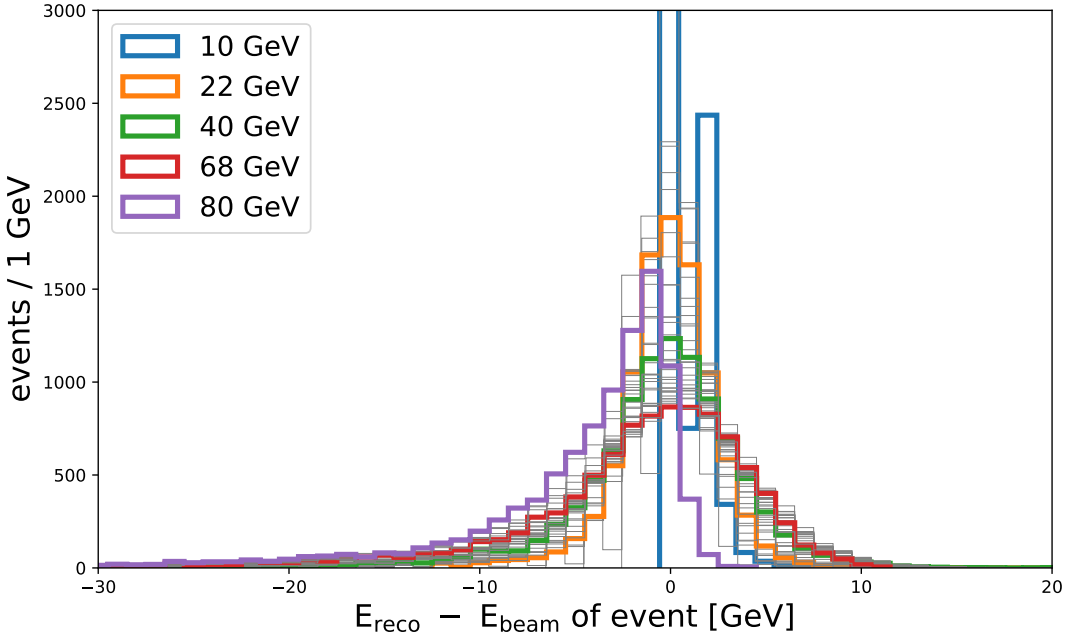


Figure G.3: Histograms of the reconstructed energy E_{reco} by the CNN network for all energies in the 'trained on' test sample for MC.

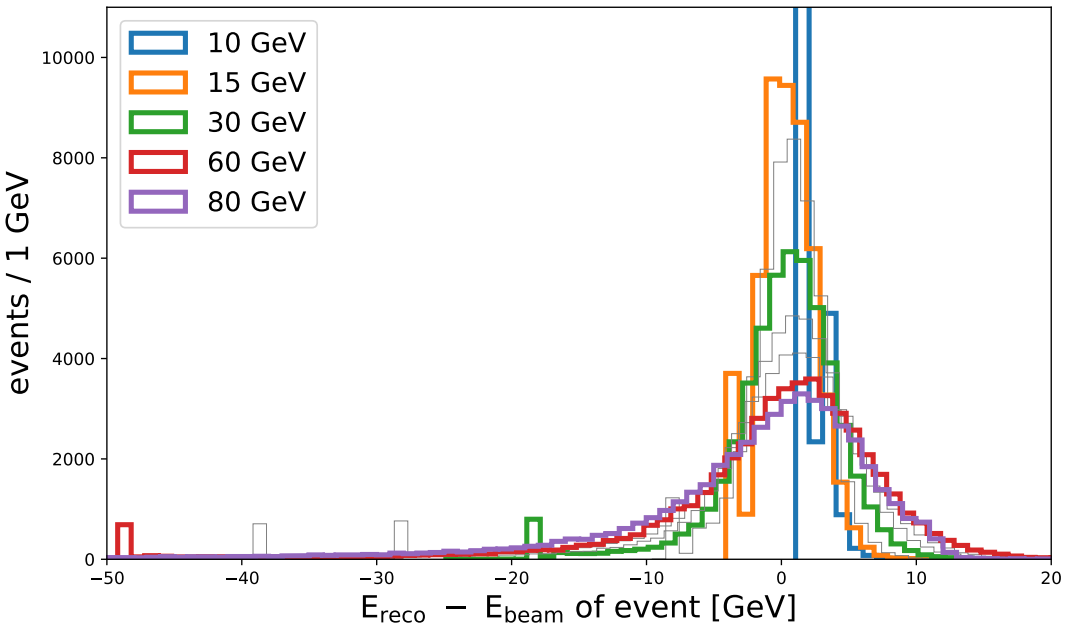
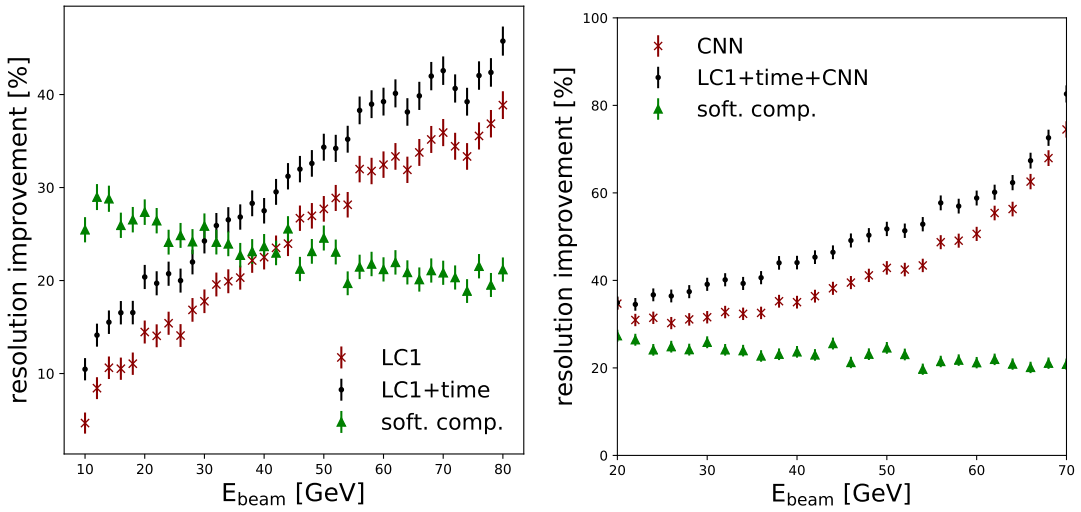


Figure G.4: Histograms of the reconstructed energy $E_{reco} - E_{beam}$ by the CNN network trained on MC but applied to data.



(a) LCN architectures

(b) CNN architectures

Figure G.5: Resolution improvement for the reconstructed energy with different networks and local software compensation for the 'trained on' MC test sample in comparison to the standard reconstruction E_{sum}. **(a)** shows the resolution difference for the locally connected architectures (with and without timing). **(b)** shows the resolution difference for the convolutional architectures (with and without timing).

Acknowledgements

I'd like to thank a few people who have supported me throughout the year working on my master project. Although I clearly have forgotten some who should to be mentioned here.

I'd like to thank Gregor Kasieczka and Erika Garutti for their exemplary supervision of this thesis and for deepening my interest in particle physics and machine learning. I am very grateful that they always took the time to discuss in detail aspects of my work during this year and supported my attendance at test beams, conferences and workshops in Germany and abroad.

I'd like to thank Katja Krüger and Felix Sefkow for warmly welcoming me into the CALICE collaboration and for encouraging me to join the test beams and the Tokyo workshop.

I'd like to thank Saiva Huck for being a great motivating office partner and a fun travel buddy.

I'd like to thank Eldwan Brianne, Simon Leiß and David Lomize for helpful technical support, when it comes to the CALICE Software, Keras, Python or Linux in general.

I'd like to thank Christian Graf, Daniel Heuchel, Lorenz Emberger, Olin Pinto, Yuji Sudo, Naoki Tsuji, Linghui Liu, Vladimir Bocharnikov, Amine Elkhalii, and Anna Rosmanitz for their great team spirit and enjoyable times, work related or not.

I'd like to thank Robert Klanner for always insightful questions, answers and discussions.

I'd like to thank Jan Schütte-Engel, Ole Brandt, Christoph Krieger and Jack Rolph for all those enjoyable lunch times and interesting discussions.

I'd like to thank Melanie Eich, Karla Pena Rodriguez and Lisa Benato for helpful advice whenever needed.

I'd like to thank Simon Schnake for stimulating machine learning discussions.

And I'd like to thank my parents for their unconditional support throughout the years of my studies.

Thank you.

Bibliography

[1] G. Aad *et al.* for the CMS collaboration, "A Particle Consistent with the Higgs Boson Observed with the ATLAS Detector at the Large Hadron Collider," *Science*, vol. 338, no. 6114, pp. 1576–1582, 2012.

[2] S. Chatrchyan *et al.* for the CMS collaboration, "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012.

[3] P. W. Higgs, "Broken Symmetries and the Masses of Gauge Bosons," *Physical Review Letters*, vol. 13, no. 16, pp. 508–509, 1964.

[4] F. Englert and R. Brout, "Broken Symmetry and the Mass of Gauge Vector Mesons," *Physical Review Letters*, vol. 13, no. 9, pp. 321–323, 1964.

[5] F. Pitters, "The CLIC Detector Concept," *arXiv:1802.06008 [physics.ins-det]*, 2018.

[6] M. Thomson, "Particle Flow Calorimetry and the PandoraPFA Algorithm," *arXiv:0907.3577 [physics.ins-det]*, 2009.

[7] The CALICE Collaboration, "Hadronic energy resolution of a highly granular scintillator-steel hadron calorimeter using software compensation techniques," *Journal of Instrumentation*, 2012.

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs.CV]*, 2014.

[9] R. Feynman and A. Zee, "QED: The Strange Theory of Light and Matter," *Princeton University Press*, 2014.

[10] S. L. Glashow, "The renormalizability of vector meson interactions," *Nuclear Physics*, vol. 10, pp. 107–117, 1959.

[11] S. Weinberg, "A Model of Leptons," *Physical Review Letters*, vol. 19, no. 21, pp. 1264–1266, 1967.

[12] H. Fritzsch, M. Gell-Mann, and H. Leutwyler, "Advantages of the color octet gluon picture," *Physics Letters B*, vol. 47, no. 4, pp. 365–368, 1973.

[13] MissMJ, "Standard Model of Elementary Particles," 2019. avaliable at Wikimedia Commons under the Creative Commons Attribution 3.0 Unported license.

[14] G. Aad *et al.* for the CMS and ATLAS collaborations, "Combined Measurement of the Higgs Boson Mass inppCollisions ats=7and 8 TeV with the ATLAS and CMS Experiments," *Physical Review Letters*, vol. 114, no. 19, 2015.

[15] M. Tanabashi *et al.*, "2018 Review of Particle Physics," *Phys. Rev. D 98, 030001*, 2018.

[16] R. Wigmans, "Calorimetry," *Scientifica Acta 2, No. 1, 18 - 55*, 2008.

[17] E. Brianne, *Time Development of Hadronic Showers in a Highly Granular Analog Hadron Calorimeter*. PhD thesis, University of Hamburg, 2018.

[18] R. Wigmans, *Calorimetry Energy Measurments in Particle Physics*. Oxford Science Publications, 2008.

[19] K.-J. Grahn *et al.*, "A Layer Correlation Technique for Pion Energy Calibration at the 2004 ATLAS Combined Beam Test," *Proceedings of the 2009 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC 2009)*, pp. 751–757, 2009.

[20] R. Kogler, *Measurments of Jet Production in Deep-Inelastic ep Scattering at HERA*. PhD thesis, Universitaet Hamburg, 2011.

[21] Y. Israeli, *Energy Reconstruction in Highly Granular Calorimeters for Future Electron-Positron Colliders*. PhD thesis, Technische Universitaet Muenchen, 2018.

[22] D. E. Groom, "Energy flow in a hadronic cascade: Application to hadron calorimetry," *Nucl.Instrum.Meth.A572:633-653*, 2007.

[23] A. Bernstein *et al.*, "Beam tests of the ZEUS barrel calorimeter," *Nuclear Instruments and Methods m Physics Research A 336 (1993) 23-52*, 1993.

[24] N. Feege, *Low-energetic Hadron Interactions in a Highly Granular Calorimeter*. PhD thesis, Universitaet Hamburg, 2011. available at http://www-library.desy.de/preparch/desy/thesis/desy-thesis-11-048.pdf.

[25] F. Sefkow and F. Simon for the CALICE collaboration, "A highly granular SiPM-on-tile calorimeter prototype," *Journal of Physics: Conference Series*, vol. 1162, p. 012012, 2019.

[26] ILCsoft developers, "ILCsoft framework," 2019. documentation under: https://ilcsoft.desy.de/portal/.

[27] O. Hartbrich, *Scintillator Calorimeters for a Future Linear Collider Experiment*. PhD thesis, Bergische Universiaet Wuppertal, 2016.

[28] N. Tsuji, "Performance of alternative scintillator tile geometry for AHCAL," *arXiv:1902.05266 [physics.ins-det]*, 2019.

[29] A. Martelli for the CMS collaboration, "The CMS HGCAL detector for HL-LHC upgrade," *arXiv:1708.08234 [physics.ins-det]*, 2017.

[30] S. Huck, "Investigation of Muon Detection with the CALICE Analogue Hadron Calorimeter." Master thesis, Universitaet Hamburg, 2019.

[31] S. Das, "A simple alternative to the Crystal Ball function," *arXiv:1603.08591 [hep-ex]*, 2016.

[32] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, "Machine learning at the energy and intensity frontiers of particle physics," *Nature*, vol. 560, no. 7716, pp. 41–48, 2018.

[33] P. Mehta *et al.*, "A high-bias, low-variance introduction to Machine Learning for physicists," *arXiv:1803.08823 [physics.comp-ph]*, 2019.

[34] K. He *et al.*, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv:1502.01852 [cs.CV]*, 2015.

[35] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, "Deep-learning top taggers or the end of QCD?," *Journal of High Energy Physics*, vol. 2017, no. 5, 2017.

[36] Peirson V, Abel L and Tolunaz, E Meltem, "Dank Learning: Generating Memes Using Deep Neural Networks," *arXiv:1806.04510 [cs.CL]*, 2018.

[37] M. Paganini, L. de Oliveira, and B. Nachman, "Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters," *Physical Review Letters*, vol. 120, jan 2018.

[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs.LG]*, 2014.

[39] M. A. Nielsen, *Neural networks and deep learning*. Determination Press, 2015. available at: http://neuralnetworksanddeeplearning.com.

[40] Martín Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from tensorflow.org.

[41] The Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," 2016.

[42] The PyTorch Developers, "PyTorch Deep Learning Library," 2019. available at pytorch.org.

[43] Keras developers, "Keras: The Python Deep Learning library," 2019. available at https://keras.io.

[44] P. Baldi, J. Bian, L. Hertel, and L. Li, "Improved energy reconstruction in NOvA with regression convolutional neural networks," *Physical Review D*, vol. 99, no. 1, 2019. available at https://arxiv.org/pdf/1811.04557.pdf.

[45] The SciPy community, "SciPy python library," 2019. available at https://docs.scipy.org/doc/scipy/.

[46] The CALICE collaboration, "Hadron selection using Boosted Decision Trees in the semi-digital hadronic calorimeter," *CALICE analysis note CALICE-CAN-2019-001*, 2019.