

Graph Building for Graph Neural Networks for Photon Reconstruction in the Belle II Calorimeter

Johanna Matusche

Bachelorthesis

2nd August 2023

Institute of Experimental Particle Physics (ETP)

Advisor: Prof. Dr. Torben Ferber

Coadvisor: Dr. Pablo Goldenzweig

Editing time: 1st April 2023 – 2nd August 2023

Graph Bauen für Graph Neuronale Netzwerke für die Photon Rekonstruktion im Belle II Kalorimeter

Johanna Matusche

Bachelorarbeit

2. August 2023

Institut für Experimentelle Teilchenphysik (ETP)

Referent: Prof. Dr. Torben Ferber

Korreferent: Dr. Pablo Goldenzweig

Bearbeitungszeit: 1. April 2023 – 2. August 2023

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, 2. August 2023

.....
(Johanna Matusche)

Abstract

This thesis presents the implementation and evaluation of the GCN algorithm for photon energy reconstruction in the Belle II electromagnetic calorimeter. The GCN algorithm, a machine learning technique based on graph convolutional networks, uses fuzzy clustering. The model is employed to analyze and reconstruct photon energies of single photon events using Monte Carlo generated and simulated data. This study focuses on the graph-building process and on investigating the graph input features utilized in the GCN model to optimize the algorithm's performance. In particular, the node input features and the number of edges in a graph as well as the edge weights are studied. To assess the effectiveness of the GCN algorithm, a comparison is made with the algorithm of the Belle II Analysis Software Framework and the GravNet algorithm. The results show that the GCN model outperforms the Belle II Analysis Software Framework model by up to 15 % in the photon energy range of 0.01 GeV to 2.5 GeV while maintaining comparable performance to the GravNet model. For photon energies exceeding 2.5 GeV, the GCN model performs better than the GravNet model, with an improvement of up to 16 %.

Disclaimer

This thesis builds on the work of Florian Wemmer [1] studying the photon reconstruction in the Belle II calorimeter using graph neural networks. The production and selection of the Monte Carlo simulated single photon events was done by Florian Wemmer with the Belle II Analysis Software Framework by the Belle II Framework Software Group [2] and the generated data sets are property of the Belle II collaboration. I reimplemented Florian Wemmer's architecture of the GravNet model [3] with a GCN layer [4] and performed the training and evaluation of the GCN model. I built all graphs using the package `NetworkX` [5]. Florian Wemmer's proposed metrics are used in this work and my results of the GCN model are compared to Wemmer's results of the GravNet and the basf2 model. The fitting functions of the relative reconstruction errors were modified by Isabel Haide (KIT, ETP) and Jonas Eppelt (KIT, ETP). All plots shown in this thesis are created by me with the `Matplotlib` package [6] using adapted versions of the plotting functions of Florian Wemmer, unless otherwise noted. My supervisor Prof. Torben Ferber proposed the studies for this thesis.

Contents

Abstract	VII
Disclaimer	IX
1. Introduction	1
2. The Belle II Experiment	3
2.1. The SuperKEKB Accelerator and the Belle II Detector	3
2.2. The Belle II Electromagnetic Calorimeter	6
2.2.1. Geometry and Operation	6
2.2.2. Clusters and Leakage	8
2.2.3. Beam Background	8
2.3. Basf2 Reconstruction Algorithm	10
3. Graph Neural Networks	13
3.1. Fundamentals	13
3.1.1. Graph Features	13
3.1.2. Introduction to Fuzzy Clustering	13
3.2. GravNet Layer	14
3.3. GCN Layer	16
4. Event Generation and Selection	17
5. Training and Implementation	21
5.1. Representation of Events by Graphs	21
5.1.1. Node Input Features	22
5.1.2. Edge Input Features	22
5.2. Architecture	23
5.3. Implementation	24
5.3.1. Input Features	24
5.3.2. Loss Function and Optimizer	25
5.3.3. Training	26
6. Metrics	29
6.1. Fundamental Reconstruction Quantities	29
6.2. Performance Evaluation	30
6.2.1. Reconstruction Errors	30

6.2.2. Energy Resolution	31
7. Toy Studies for Single Photon Clusters	37
7.1. Motivation and general Settings	37
7.2. Node Input Features	38
7.2.1. Settings	38
7.2.2. Performance Evaluation	38
7.2.3. Overview and Conclusions	39
7.3. Edge Input Features	42
7.3.1. Number of Edges in the Graph	42
7.3.2. Edge Weights in the Graph	46
7.4. Energy dependence	50
7.4.1. Settings	50
7.4.2. Performance Evaluation	50
7.4.3. Overview and Conclusions	51
8. Summary and Outlook	55
Bibliography	59
A. Appendix	61
A.1. Bad Double-Sided Crystal Ball Fit	61
A.2. Performance Evaluation of the Deposited Energy for the Photon Energy 0.3 GeV for Different Node Input Features	62
B. List of Figures	65

1. Introduction

Particle physics revolves around the study of subatomic particles, delving into the intricate fabric of matter and energy of which the universe is built. The Standard Model (SM) has proven remarkably successful in explaining the interactions between known particles and the fundamental forces that govern them [7]. Yet it leaves significant gaps and unanswered questions, hinting at the existence of yet-to-be-discovered particles and phenomena.

The Belle II experiment aims to uncover new physics by analyzing rare decays utilizing the Belle II detector, located at the SuperKEKB electron-positron collider in Tsukuba, Japan. At the SuperKEKB accelerator, the $\Upsilon(4S)$ resonance is finely tuned, creating an ideal experimental environment for studying the decay of B-Meson pairs [8, 9]. To achieve high-precision measurements, a substantial amount of data is required which depends upon a high instantaneous luminosity. Ongoing efforts to enhance SuperKEKB's capabilities intend to increase the luminosity by an order of magnitude in the coming years. Excellent performance in photon energy reconstruction is crucial for numerous physics analyses at Belle II. However, the high instantaneous luminosity leads to increased beam background and presents a significant challenge to this process [8].

This thesis introduces the graph-building process for graph neural networks (GNNs) to improve the energy resolution of photon clusters. In particular, the implementation of graphs in a graph convolutional network (GCN) and the evaluation of the model's performance for the photon energy reconstruction in the Belle II electromagnetic calorimeter (ECL) for different graph input features is studied. By employing the GCN algorithm, all relevant input features are implemented in contrast to traditional clustering methods. This aids the GCN model, particularly when confronted with high levels of background interference. As a result, this thesis successfully enhances the energy resolution for photons in comparison to the current reconstruction algorithm used in Belle II and gives a better understanding of graphs and their features that are used here in GNNs.

Chapter 2 provides an introductory overview of the Belle II experimental setup, with a particular focus on the ECL and beam background processes. It also introduces the Belle II Analysis Software Framework. Chapter 3 presents the GNN models GCN and GravNet and their specific implementation for application in the Belle II ECL. In chapter 4, the event generation and selection processes are described, establishing the essential conditions for training the machine learning (ML) algorithm. This thesis exclusively utilizes Monte Carlo (MC) generated and simulated data. The training of the GCN model and the implementation of the graphs with various input features as well as the model's architecture are studied in

chapter 5. Metrics that effectively characterize and evaluate the reconstruction performance for the subsequent studies are introduced in chapter 6. Chapter 7 explores the behavior and performance of the GCN model for different graph input features. It also compares the performance of the GCN model to the GravNet and the Belle II Analysis Software Framework algorithm regarding the reconstruction of photon energies for different beam background scenarios. Finally, chapter 8 serves as a comprehensive summary, encompassing all the presented results throughout the thesis, and offers an outlook toward future work.

2. The Belle II Experiment

The Belle II experiment is a particle physics experiment located at the SuperKEKB particle accelerator in Tsukuba, Japan. Resulting from the SuperKEKB collider are electron-positron collisions of high intensity. As upgrades to their respective predecessors, the Belle II experiment and the SuperKEKB accelerator supersede the Belle experiment and the KEKB collider. With an anticipated integrated luminosity of about 50 ab^{-1} over a decade, the upgrade amplifies data-taking capabilities, yielding a data set 50 times larger than what the Belle experiment achieved in the same timeframe. Moreover, the upgrade aspires to increase the instantaneous luminosity by more than 40-fold compared to the previous Belle experiment conducted at the KEKB collider. Consequently, the enhanced instantaneous luminosity leads to an elevated background level within the Belle II experiment.

This chapter about the Belle II experiment describes the detector with its sub-detector parts in section 2.1. Furthermore, the electromagnetic calorimeter is introduced in detail in section 2.2. The information in this chapter is based on the Belle II technical report [9] and the Belle II physics book [8].

2.1. The SuperKEKB Accelerator and the Belle II Detector

The SuperKEKB accelerator comprises two storage rings, one dedicated to high-energy electron beams and the other to low-energy positron beams. With a circumference of 3016 m, the accelerator features four straight sections, each housing an experimental hall. Figure 2.1 illustrates a schematic view of the SuperKEKB accelerator. The e^+e^- -collision occurs exclusively at the Tsukuba experimental hall's interaction point (IP) within the Belle II detector. Operating at the energy of the $\Upsilon(4S)$ resonance, 7 GeV electrons collide with 4 GeV positrons. This resonance decays mostly into B meson pairs, which subsequently undergo further decays, yielding leptonic or hadronic particles.

The primary objective of the Belle II detector is to enhance our understanding of particle physics and explore phenomena beyond the SM. A crucial aspect of this pursuit is the precise identification of various particles generated during collisions. Particle identification involves combining information from different sub-detectors, each specializing in measuring specific particle properties and tracks. The collision between the electron and positron beams in Belle II is asymmetric due to their differing energies. Consequently, the produced B mesons experience a boosted state instead of decaying immediately.

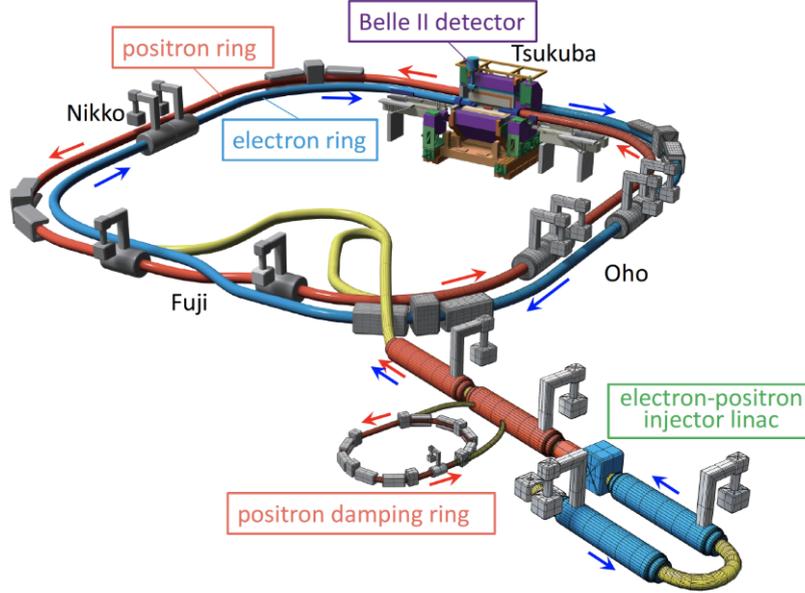


Figure 2.1.: A schematic representation of the SuperKEKB accelerator. The Belle II detector is located at the experimental hall Tsukuba. The other three experimental halls Oho, Fuji, and Nikko are as well on the straight sections of the accelerator. The figure is taken from [10].

To achieve comprehensive coverage, the detector is divided into three main components: the barrel, the forward endcap, and the backward endcap. The barrel assumes a cylindrical shape, with its parallel axis aligned to the electron and positron beams. The endcaps, circular in shape, are positioned at either end of the barrel. While the forward endcap is located at the incoming positron beam, the backward endcap is positioned at the incoming electron beam. Figure 2.2 provides a schematic representation of the Belle II detector, illustrating its sub-detectors and their arrangements.

The Vertex Detector, which surrounds the IP, is composed of two main components: the pixelated silicon sensor (PXD) and the silicon strip layer (SVD). The PXD is constructed from two all-silicon modules, each featuring sensor matrices of depleted P-channel field effect transistors (DEPFETs). On the other hand, the SVD comprises four layers of double-sided silicon strip detectors (DSSDs). The sensors in the Vertex Detector precisely measure the decay vertex of collision products, primarily focusing on B mesons while also capturing other particles generated in the e^+e^- -collision.

Adjacent to the Vertex Detector is the cylindrical central drift chamber (CDC). The CDC contains numerous wires with the purpose of tracking the trajectories of charged particles as well as their momenta and their dE/dx information.

Moving further away from the IP, the time-of-propagation (TOP) detector is located which is designed for particle identification. Positioned in the barrel region, the TOP detector features 16 quartz bars arranged circularly around the collision point. As particles traverse

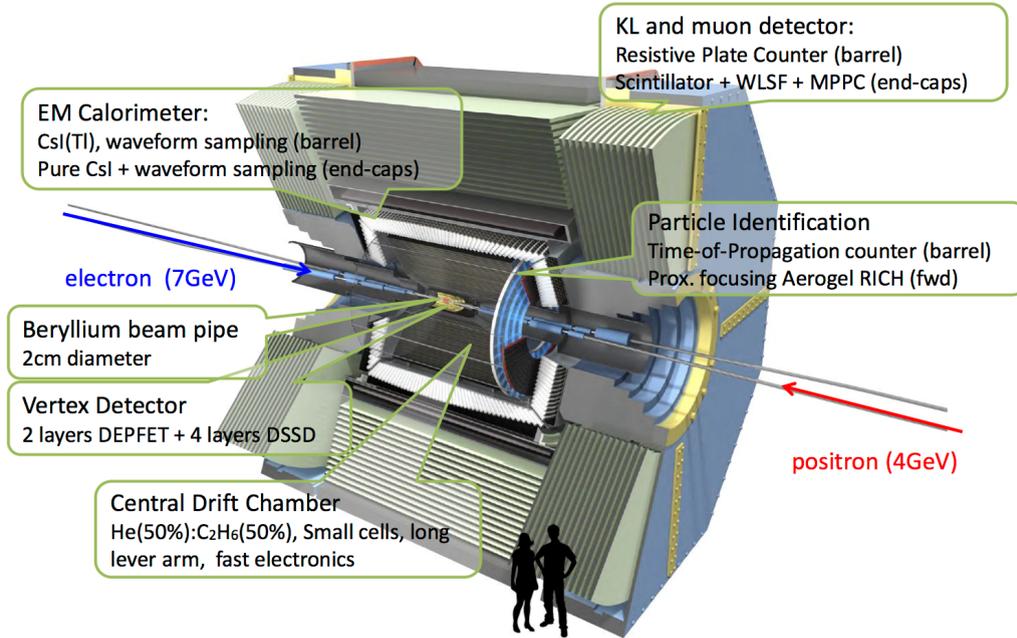


Figure 2.2.: A schematic overview of the Belle II detector with its sub-detector parts. The detector is centered around the interaction point. The figure is taken from [11].

through the quartz bars, they emit Cherenkov photons, and the TOP detector measures the propagation and position of these. The distribution of the Cherenkov photons provides valuable information for particle identification. In addition to the TOP detector in the barrel is the aerogel ring imaging Cherenkov (ARICH) counter in the forward region of the detector for particle identification.

Surrounding the CDC is the ECL whose primary function is to measure the total energy of charged particles and photons. It consists of scintillator crystals that are placed inside the superconducting solenoid coil. The purpose of the ECL is the determination of the total energy and the angular coordinates of the particles, with a specific emphasis on identifying electrons, hadrons, and photons. This thesis focuses on the photon detection and reconstruction of photon energy and therefore the ECL is described in more detail in section 2.2.

A constant magnetic field of 1.5 T is generated by the superconducting solenoid coil to curve the trajectories of charged particles in the CDC. The radius of the charged particle trajectory, influenced by the constant magnetic field, allows the determination of the particle's momentum.

The outermost part of the Belle II detector is the K_L^0 and muon detector (KLM). As the name says, its primary purpose is to identify and determine the properties of the long living K_L^0 and muons. Additionally, the KLM acts as a particle decelerator and serves as the magnetic flux return yoke for the magnet.

2.2. The Belle II Electromagnetic Calorimeter

A significant fraction of B decay products consists of neutral particles, a portion of which decay into photons within the energy range of 20 MeV to 4 GeV. The primary purpose of the Belle II electromagnetic calorimeter is to detect photons with high efficiency while accurately determining their energy and angular coordinates. Furthermore, the ECL aids in particle identification by analyzing the shower shape of particles, in conjunction with other sub-detector parts such as the KLM. In addition to its photon detection capabilities and particle identification role, the ECL is also used for measuring the luminosity. The luminosity is a parameter used to quantify the rate of collisions [9].

2.2.1. Geometry and Operation

The following section is based on [8,9]. The ECL is divided into three sections: the barrel region, the forward endcap, and the backward endcap. In the ECL's coordinate system, the IP serves as the origin for spherical coordinates θ and ϕ . θ represents the polar angle, while ϕ denotes the azimuthal angle with respect to the beam pipe. The z -axis aligns along the beam pipe in the direction of the electron beam.

Regarding the position from the IP, the forward endcap is located at $z = 1.96$ m and the backward endcap at $z = -1.02$ m. The barrel extends for a length of about 3 m and has an inner radius of 1.25 m. The ECL covers a polar angle range of $12.01^\circ < \theta < 155.03^\circ$, providing comprehensive coverage, except for two narrow gaps situated between the barrel and each endcap. These gaps have a width of approximately 1° . A schematic representation of the ECL and its components can be seen in figure 2.3.

The three ECL regions include in total 8736 thallium-doped cesium iodide CsI(Tl) crystals. Within the barrel region, there are 6624 crystals, each possessing one of 29 distinct shapes. These crystals take the form of truncated pyramids, with an average cross-section of 6×6 cm² and a length of 30 cm, which is equivalent to 16.1 radiation lengths. The crystals are positioned with their front facing the IP, and they are deliberately shifted relative to one another to prevent particles from escaping through gaps between the crystals.

The geometrically more complicated endcaps consist of 2112 crystals that also point towards the IP. The crystals have a similar length as the crystals in the barrel. Due to the irregular arrangement, there are 69 crystal shapes with a wider variation in cross section and size. Additionally, there are larger gaps between crystals in the endcaps compared to the barrel region. The forward endcap is densely packed with crystals compared to the backward endcap. This difference in packing density is made due to the asymmetric nature of the electron-positron collision.

Each crystal is wrapped with a thin layer of Teflon and aluminum to keep the emitted light of the scintillation within the crystal. At the two ends of each crystal are PIN photodiodes that measure the light yield generated by particles and their subsequent showers.

Crystal Measurements

In the ECL, the basic crystal measurements are the reconstructed energy, the reconstructed time, and the pulse shape discrimination (PSD). The pulses of the incoming particles at

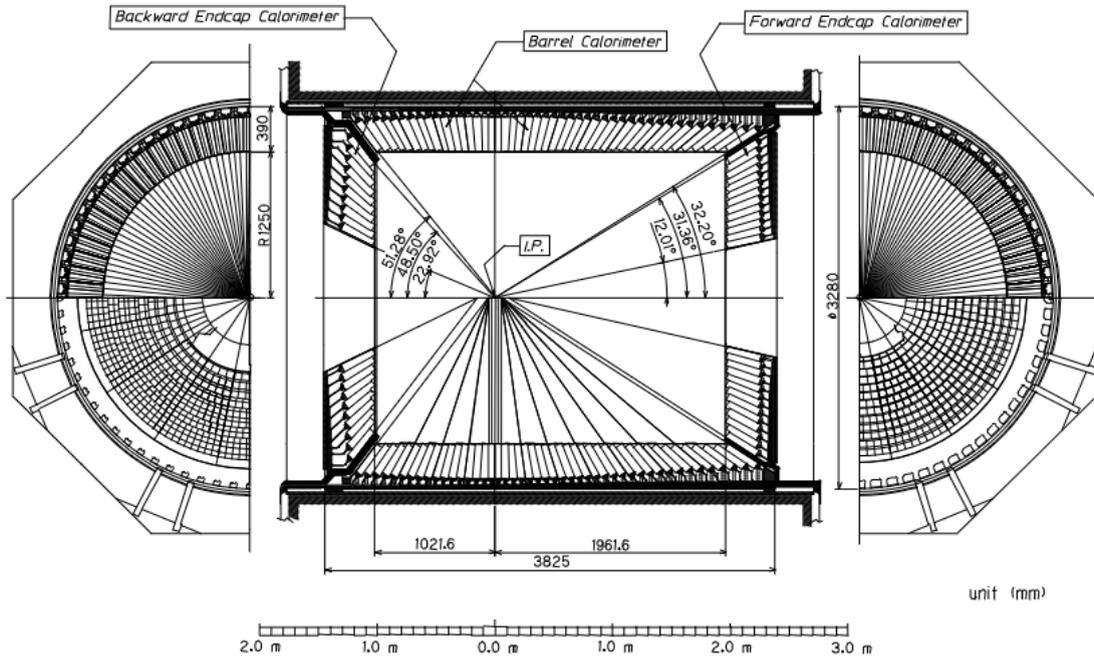


Figure 2.3.: A schematic representation of the Belle calorimeter. The mechanical structure and crystal positions are identical for Belle and Belle II. In the middle is a display of the cross-section of the overall electromagnetic calorimeter. The interaction point is in its center and marked with IP. On the left side is the backward endcap displayed from the perspective of the interaction point and similarly on the right side is the forward endcap. The figure is taken from [12].

the photodiodes are converted into voltage signals by a preamplifier and then shaped and digitized by the ShaperDSP module, resulting in a wavefunction [13].

For each crystal i , the reconstructed energy is denoted by E_i^{rec} , while the reconstructed time is denoted as t_i^{rec} . The reconstructed time is determined by comparing the arrival times of signals from different photodiodes within the calorimeter and is reported relative to the time of the collision. This reconstructed time aids in distinguishing prompt particles produced during the collision from particles originating from the decay of other particles. Both the reconstructed energy and time are independently obtained by fitting the digitized amplitudes to a known response function.

The PSD information in the ECL plays a crucial role in accurately identifying particle types and distinguishing between photons and charged particles within the detector. Different types of particles exhibit distinct pulse shapes in the ECL due to variations in energy deposition and subsequent light emission processes. The PSD information is obtained offline from the reconstructed wavefunction of the crystals. It includes the hadron intensity (HI), its corresponding fit type, and χ^2 for each crystal.

The HI is determined by the number of scintillations caused by hadrons in the crystal. To properly fit the reconstructed wavefunction, both a photon template fit and, if applicable, a hadron template fit are employed. The PSD represents the extent to which the hadron

template fit is necessary to adequately fit the reconstructed wavefunction. The fit type and χ^2 are determined during this process [14]. PSD information is only determined for crystals with reconstructed energy exceeding 30 MeV.

While the measurement of the recorded energy and time are online, the PSD is determined offline from the recorded data.

2.2.2. Clusters and Leakage

In the ECL, it is common for particles and their showers to distribute their energy across multiple crystals rather than depositing their total energy within a single crystal. As a result, a single crystal does not provide the complete reconstructed energy of a particle. Instead, the energy of a particle must be determined from a cluster. The cluster is defined by several crystals that all have deposited energy of the same particle. The shape of the cluster is influenced by the shape of the particle shower and the detected crystals form a connected region (CR) in the ECL. It is also possible that multiple particles deposit energy in the same crystal. Therefore, the challenges for the energy estimation are to determine by cluster algorithms which particle deposited what amount of energy in each crystal.

The simplest shower shape belongs to photons, as they typically deposit the majority of their energy in a single crystal. The energy distribution around this central crystal is typically radially symmetrical. Clusters from other particles are often more complicated due to the asymmetry of their shower shape. This thesis focuses on the reconstruction of photon showers, neglecting the reconstruction of other particle types.

The reconstruction of the energy is compromised by the leakage of the ECL due to small gaps between crystals or due to partial energy depositions of the particle after the ECL in different sub-detector parts. This leads to a lower cluster energy reconstructed by the cluster than the original particle energy. Leakage has a more pronounced impact in the endcaps of the ECL, as there are more gaps between the crystals in this region. Additionally, particles deposit energy in inactive material in other sub-detector parts before reaching the ECL. Therefore, the energy resolution of the ECL is limited by the leakage, resulting in a loss of a few percent of the initial particle energy.

2.2.3. Beam Background

This section is based on [8, 15]. A high fraction of the energy depositions in the ECL is not from collision product particles but from beam background. The beam background is generated by unstable beam pipe particles that collide on the inner side of the beam pipe. Their shower is usually low-energetic but in total the energy of the beam background ranges from 0.5 MeV to 1.0 MeV in the barrel and is up to 2.0 MeV in the endcaps per crystal. Most beam background particles deposit their total energy in a single crystal.

In the ECL, the most prominent sources of beam background are the Touschek effect, beam gas scattering, and radiative Bhabha scattering.

The Touschek effect accounts for 98% of the beam background in the ECL. It results from Coulomb scattering of beam particles that are part of the same bunch and its rate is proportional to the squared beam current and the inverse beam size.

Beam gas scattering accounts for around 2% of the beam background. It is the result of the Coulomb scattering of residual gas atoms with beam pipe particles. The scattering rate

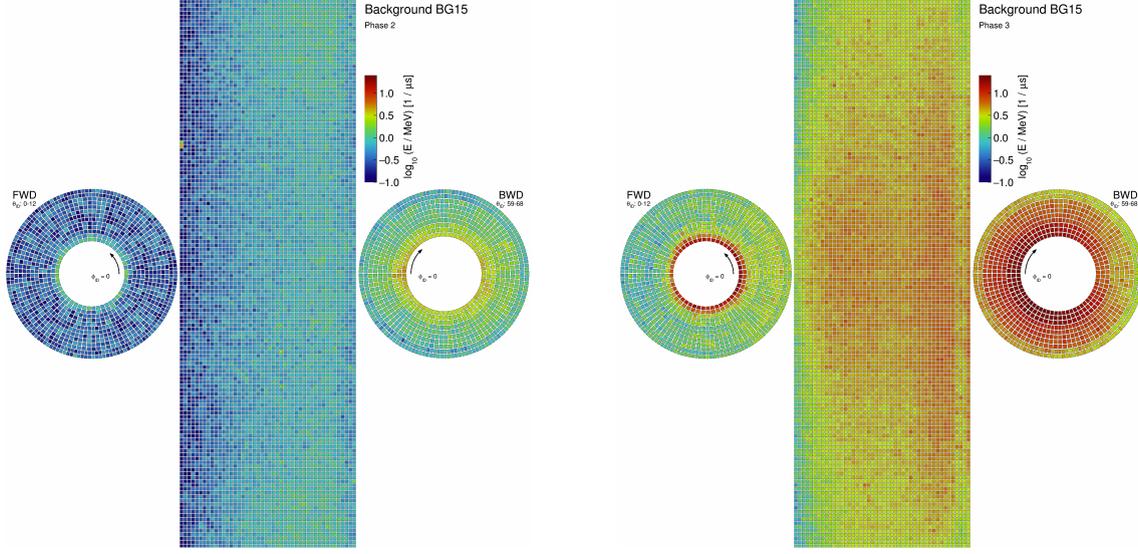


Figure 2.4.: Background energy depositions in the electromagnetic calorimeter for past low beam background (2018) and future high beam background (2027). The projection includes the rolled-out barrel and the endcaps. Both are displayed from the perspective of the interaction point. The energy in each crystal is represented by color. The figure is adapted from [16].

is proportional to the vacuum pressure and the beam current.

The $e^+e^- \rightarrow e^+e^-\gamma$ process where an additional photon hits the beam pipe is called radiative Bhabha scattering. The process is proportional to the luminosity and very likely to produce high-energetic beam background.

Most of the beam background particles just hit a single crystal but it also occurs that high-energetic beam background particles hit the ECL and create clusters that are indistinguishable from collision products and their decays. Therefore, the distinguishing of the cluster type needs to be done at a later stage.

In this thesis two beam background types are studied that are associated with different phases of the Belle II experiment :

- Low beam background (LBB) also called early background in this thesis is the current background simulation. The instantaneous luminosity is $\mathcal{L} = 1.06 \cdot 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$. In a singular event around 1500 crystals have beam background energy deposition.
- High beam background (HBB) also called nominal background in this thesis is the expected future background simulation in 2027. The instantaneous luminosity is expected to be $\mathcal{L} = 6 \cdot 10^{35} \text{ cm}^{-2} \cdot \text{s}^{-1}$. In one event around 3500 crystals have beam background energy deposition.

j

The occurrence rate of beam background processes increases with higher luminosity, smaller beam size, and higher beam currents. This phenomenon is shown in figure 2.4 as well as the dependency of the beam background energy on the ECL region.

In the backward endcap and in the inner crystals of the beam pipe in the forward endcap are the highest beam background depositions. Due to the fact that the positron beam directed towards the backward endcap is less focused than the electron beam the positrons collide more often with the beam pipe. Additionally, many wires are positioned in front of the backward endcap resulting in more non-physical collisions of the beam and detector material. The beam background energy decreases in the barrel from the crystals closest to the backward endcap to the crystals closest to the forward endcap.

2.3. Basf2 Reconstruction Algorithm

The Belle II Analysis Software Framework (`basf2`) [2] is the current software framework used at the Belle II experiment and includes all software-related activities. In this thesis, the term `basf2` is referring to the energy reconstruction algorithm in the ECL using the actual `basf2`.

The energy reconstruction in the ECL using the `basf2` algorithm determines clusters as follows [8]:

1. The clustering starts with finding seed crystals that have a reconstructed energy $E_{\text{seed}}^{\text{rec}} > 10 \text{ MeV}$.
2. The eight nearest neighbors of the seed crystals are examined and added to the CR if the crystals have an energy deposition above $E_i^{\text{rec}} > 0.5 \text{ MeV}$.
3. The next-to-nearest neighbors of the seed crystal are added to the CR if the crystals have an energy of at least $E_i^{\text{rec}} > 1.5 \text{ MeV}$. This procedure is repeated for every nearest neighbor of a crystal in the CR.
4. If another crystal in the CR of the seed crystal also has an energy deposition of $E_i^{\text{rec}} > 10 \text{ MeV}$, this crystal's eight nearest neighbors are also considered as in step two and the two CRs get merged.
5. All local maximums (LMs) in reconstructed energy in the CRs are determined and become the origin of a potential cluster. A LM crystal has the highest reconstructed energy of all its direct nearest neighbors.
6. A cluster is limited to a 5×5 crystal arrangement, excluding the corners, around the LMs.

If there are several LMs in one CR each crystal is assigned partially to each LM by using shared weights that are normalized to unity per crystal. The weight per crystal i is defined by

$$w_i = \frac{E_i e^{-Cd_i}}{\sum_k E_k e^{-Cd_k}}, \quad (2.1)$$

where $C = 0.7$ is an MC determined constant, d_i is the distance between the centroid of the cluster and crystal i and is determined by iteration. The denominator sums over all crystals k in the cluster. The cluster centroid is determined by

$$\vec{x} = \frac{\sum_i w_i' \vec{x}_i}{\sum_i w_i'}, \quad (2.2)$$

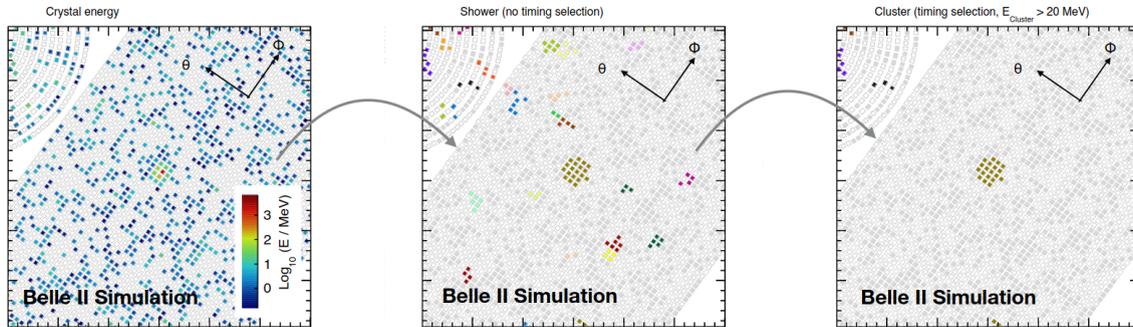


Figure 2.5.: Representation of the basf2 clustering algorithm showing a flat projection of the $\theta - \phi$ crystal plane in the Belle II electromagnetic calorimeter. The view is from the interaction point and the barrel is rolled out and visible on the right side and part of the forward endcap at the top left corner. The left plot shows the reconstructed energy of the crystals that are the input for the basf2 algorithm. The middle plot shows the determined clusters without timing selection. In the right plot, the final clusters are displayed where each cluster with $E_{\text{cluster}} < 20$ MeV is rejected. Each cluster is associated with one particle. The figure is taken from [16].

with $w'_i = 4.0 + \log(w_i \times E_i / E_{\text{all}})$. E_{all} is the sum of all weighted crystal with $w_i > 0$. Only crystals with $w'_i > 0$ are used for the calculation of \vec{x} . Starting with the centroid position

$$\vec{x}^{(0)} = \frac{\sum_i E_i \vec{x}_i}{\sum_i E_i}, \quad (2.3)$$

where \vec{x}_i is the geometric center and E_i the energy of crystal i , the iteration procedure is done until the average centroid positions of all clusters in one CR are stable within 1 mm. The energy of a cluster $E_{\text{cluster}} = \sum_n w_i \times E_i$ is determined by summation over the n highest weighted crystal energies of up to 21 nearest neighbors. n depends on the beam background level and the energy estimation of the eight nearest neighbors. The cluster time resolution is defined to include 99% of all signal clusters based on MC. Showers below 50 MeV with a reconstructed time t_{rec} exceeding this value are not retained. Afterwards, the cluster energy gets corrected as a function of position, energy, and expected background level. The clustering of one event in the ECL is visualized in figure 2.5. The final cluster energy determined by the basf2 algorithm is referred to as $E_{\text{pred}}^{\text{corr}}$ in this thesis.

In conclusion, the basf2 algorithm uses the reconstructed energies of each crystal as an input and forms CRs. The algorithm determines LMs within the CRs and assigns all crystals partially to every LM. Thereby clusters are formed that are associated with only one particle. A disadvantage of the basf2 algorithm is that it cannot assign partial membership of a crystal to beam background.

3. Graph Neural Networks

GNNs are neural models commonly utilized in deep-learning tasks to process graph data. GNNs employ a message passing mechanism, where nodes in a graph exchange information with their connected nodes. One of the notable advantages of GNNs is their ability to handle graphs of varying sizes and complex structures.

This chapter provides an overview of the fundamental concepts of GNNs, including graph features and fuzzy clustering, which are discussed in section 3.1. Additionally, this chapter explores GNN layers relevant to this thesis, such as the GravNet layer detailed in section 3.2, and the GCN layer explained in section 3.3.

3.1. Fundamentals

3.1.1. Graph Features

The following section is based on [17]. Graphs serve as the input for GNN models and are characterized by two crucial components: node features and edge features. Thus, a graph is represented as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $|\mathbf{V}|$ denotes the number of nodes and $|\mathbf{E}|$ represents the number of edges within the graph.

For the graphs examined in this thesis, undirected edges are utilized, which establish connections between pairs of nodes. The GNN model employing the GravNet layer takes dynamic graphs as input, where the graph's input features vary over time. On the other hand, the algorithm utilizing the GCN layer operates on static graphs, where the input features remain constant over time.

ML methods aim to classify objects, which can be node classification, edge classification, or graph classification for graphs. In this thesis, the focus is on node-level classification, where each node is assigned partial membership to different classes. This process is commonly referred to as fuzzy clustering.

3.1.2. Introduction to Fuzzy Clustering

Fuzzy clustering refers to the practice of partially assigning nodes to multiple classes, as opposed to the exclusive assignment of nodes to a single class, known as hard clustering [18]. The algorithms examined in this study, namely the GravNet and GCN models, both employ fuzzy clustering techniques. In this context, each crystal is represented as a node, and

the algorithm determines the fractional contributions of the photon signal and the beam background to the crystal’s energy. More details regarding this implementation can be found in chapter 5. The fuzzy clustering weights $w_i^{(u)}$, which indicate the fraction of energy assigned to either the photon signal or the beam background, must satisfy the following condition:

$$\sum_u w_i^{(u)} = 1 \quad \forall i, \quad (3.1)$$

where $w_i^{(u)} \in [0, 1]$. This equation ensures that the total energy assigned to each crystal i is equal to E_i^{rec} , indicating a complete allocation of energy between the photon signal and the beam background.

3.2. GravNet Layer

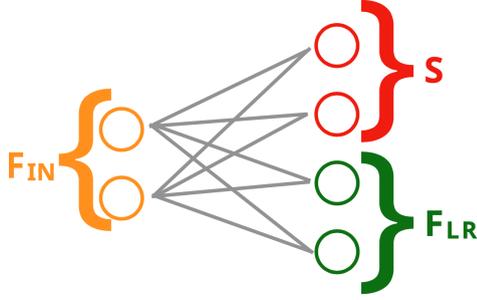
This section is based on [3]. The GravNet layer is the core of the GravNet architecture as the layer is responsible for the message passing among the nodes of the graph. The goal of the GravNet architecture is to maintain a trainable space representation while minimizing computational complexity.

The operation of the GravNet layer is visualized in figure 3.1. The layer takes as input a $B \times V \times F_{\text{IN}}$ data set, where B represents a batch of examples, each comprising a set of V detector hits, and F_{IN} denotes the initial features such as Cartesian coordinates or the reconstructed energy.

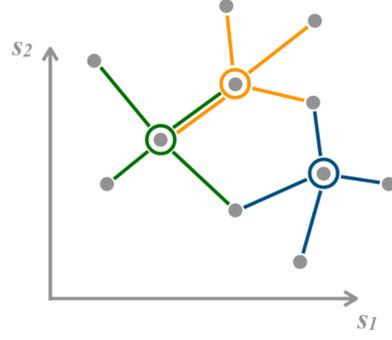
In the first step it processes the input features F_{IN} by a dense neural network. The resulting output of the dense neural network consists of two arrays: S , which represents the coordinates in a learned representation space, and F_{LR} , serving as the learned representation of node features. The k nearest neighbors are determined based on the Euclidean distance d_{ij} in S between two nodes v_i and v_j . In the third step, each node v_i gathers information f_j^i of its edges from the learned features F_{LR} . The features are weighted by a Gaussian potential $V(d_{ij}) = \exp(-10 d_{ij}^2)$ that depends on the distance between the connected nodes in S . This gravitational potential, which decreases with distance, is the inspiration behind the layer’s name: gravitational network (GravNet). The scaled features \tilde{f}_{jk}^i , all associated with node v_k , are combined to generate a new feature \tilde{f}_k^i by summation. This process is carried out for all nodes, leading to the formation of the newly learned feature representation \tilde{F}_{LR} . In the last step, another dense layer is used to concatenate the initial features F_{IN} with the newly learned feature representation \tilde{F}_{LR} , resulting in a set of output features F_{OUT} . This final data set has the dimensions $B \times V \times F_{\text{OUT}}$.

The GravNet layer involves several hyperparameters, including the number of initial features F_{IN} , the dimensions of the representation spaces S and F_{LR} , and the number of nearest neighbors k . The dimension of the output features F_{OUT} is set to the dimension of the initial features F_{IN} .

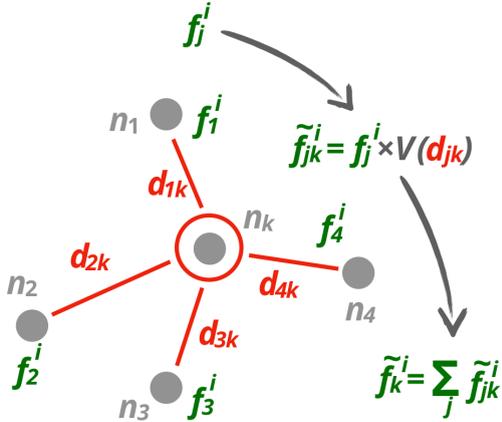
One advantage of the GravNet layer over other GNNs is that the output features F_{OUT} contain information about the nodes and their spatial surroundings. This is achieved through the use of two information layers, S and F_{LR} , which provide a low-dimensional representation of node connections. As a result, the GravNet layer offers computational efficiency without sacrificing important spatial information.



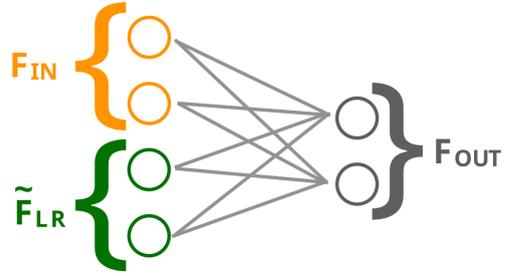
(a) The input features F_{IN} are processed by a fully connected layer. The spatial information S and a set of features F_{LR} are learned.



(b) Nodes are connected to their k nearest neighbors in the spatial information space S on the basis of the Euclidean distance d between neighboring nodes.



(c) The features f_j^i of node n_k are converted to the features \tilde{f}_{jk}^i by a Gaussian potential $V(d)$ and then aggregated by summation to \tilde{f}_k^i .



(d) The initial features F_{IN} and the aggregated learned features \tilde{F}_{LR} are processed by a fully connected layer to the output features F_{OUT} .

Figure 3.1.: Pictorial representation of the data processing in the GravNet layer. The plots display the learning of the spatial and feature space (a), the connection of nodes (b), the message passing (c), and the concatenation to one output (d) for one node. The process is done simultaneously for all nodes in one graph. The figures are adapted from [3].

3.3. GCN Layer

This section is based on [4, 19]. The goal of a GCN is to learn representations of nodes that capture both their local information (i.e., features of the node itself) and their global information (i.e., features derived from neighboring nodes). Unlike convolutional neural networks (CNNs), GCNs need to adapt the convolution operation to accommodate the non-grid-like nature of graphs. GCNs are commonly used for semi-supervised learning tasks, where graph-based regularization helps propagate information of the input features across the graph.

Graph-based regularization smooths label information over the graph. In comparison to the GravNet layer, the GCN layer does not build the graph but gets it as an input feature. The layer-wise propagation in a GCN is performed as follows:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)}), \quad (3.2)$$

where $H^{(l)}$ represents the activation matrix in the l^{th} layer, with $H^{(0)} = X$. $X \in \mathbb{R}^{N \times C}$ denotes the signal, where C represents the dimension of the node feature vector. $\tilde{A} = A + I_N$ denotes the adjacency matrix of the undirected graph \mathbf{G} , and I_N is the identity matrix. Adjacency matrices are able to represent the existence of edges that connect the node pairs through the value in the matrices. \tilde{A} is a $N \times N$ matrix, where N is the total number of nodes in the graph and A_{ij} is the edge weight of the edge connection between node i and j . $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ are layer specific trainable weight matrices and $\sigma(\cdot)$ is a non-linear activation function.

The GCN layer is derived from stacking multiple convolutional layers, which can be expressed as:

$$Z = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} X \Theta), \quad (3.3)$$

where $Z \in \mathbb{R}^{N \times F}$ represents the convolved matrix, and $\Theta \in \mathbb{R}^{C \times F}$ denotes the matrix of filter parameters. Here, F represents the number of feature maps. A visual representation of a multi-layer GCN is shown in figure 3.2.

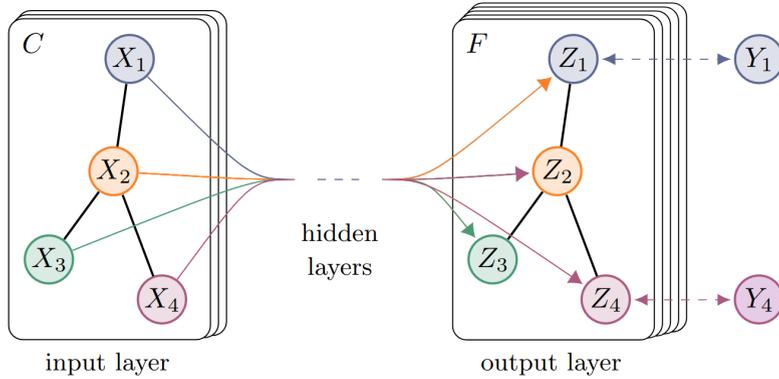


Figure 3.2.: Schematic representation of a multi-layer graph convolutional network. The input signals X_i in C input channels are transformed by the semi-supervised learning algorithm to F feature maps in the output layer. Graph edges are visualized as black lines and the labels are denoted by Y_i . The graph structure is shared over all layers. The figure is taken from [4].

4. Event Generation and Selection

All information in this chapter is taken from [20]. The event generation and selection were done for the master thesis [1] where more detailed information can be found. Therefore, only a short summary of the event generation and selection is given in this chapter.

In this thesis, the reconstruction algorithms rely on simulated events. The simulation process involves generating events that accurately represent the interactions between photons and the different detector components as explained in section 2.2.1, along with the specific response of the detector using the `GEANT4` software. Simulated MC data offers the advantage of providing knowledge about the true information of the events. This truth information is crucial for training the clustering algorithms used in this study.

In this thesis, the focus is on reconstructing single photon events. The photons used for training and testing the algorithms are simulated with energies ranging from 0.1 GeV to 5 GeV. The energy and direction of the photons are drawn randomly from independent uniform distributions, with the photon direction specified by angles $17^\circ < \theta < 150^\circ$ and $0^\circ < \phi < 360^\circ$. The generation vertex is fixed at $x = 0$, $y = 0$, and $z = 0$.

In addition to simulating the signal from single photons, the expected beam background (see section 2.2.3) also needs to be simulated. The simulated LBB has an instantaneous luminosity of $\mathcal{L}_{\text{LBB}} = 1.06 \cdot 10^{34} \text{ cm}^{-2} \cdot \text{s}^{-1}$, which corresponds to the conditions in 2021. The simulated HBB has an instantaneous luminosity of $\mathcal{L}_{\text{HBB}} = 8 \cdot 10^{35} \text{ cm}^{-2} \cdot \text{s}^{-1}$. The HBB luminosity that is used for this thesis is higher than described in section 2.2.3. This higher luminosity is chosen to create a more challenging simulation scenario [21], as higher beam background levels make it more difficult for ML algorithms to distinguish the signal from the background.

The single or also called isolated photon events are simulated and then selected by the following criteria:

- Any crystal must have the reconstructed energy $E_{\text{rec}} > 1 \text{ MeV}$ to be considered for the following steps.
- The MC particle is a primary photon and the only particle in the event. Consequently, the photon is the only particle that deposits energy.
- The LM crystal must have a reconstructed energy of at least 10 MeV.
- The region of interest (ROI) consists of a 9×9 array of crystals around the LM. Therefore the array is defined by the 80 nearest neighbors of the LM.

- The LM is the only LM in the ROI.
- All energy depositions of the photon are within the ROI.
- The matched truth photon must be responsible for at least 20% of the reconstructed energy in the LM crystal.
- The photon is the only particle that deposits energy in the ROI.

For each event, the truth energy deposition per photon, the reconstructed crystal energy $E_{\text{rec}}^{\text{crystal}}$, the crystal time $t_{\text{rec}}^{\text{crystal}}$, the crystal PSD information (see section 2.2.1), and the LM positions in the ROI are simulated and recorded. The ROI, which represents a localized coordinate area centered around the LM crystal, is used as input for the GNN clustering algorithms.

Figure 4.1 displays an example of a simulated single photon event in the full detector with LBB and figure 4.1 an example of a simulated single photon event with HBB.

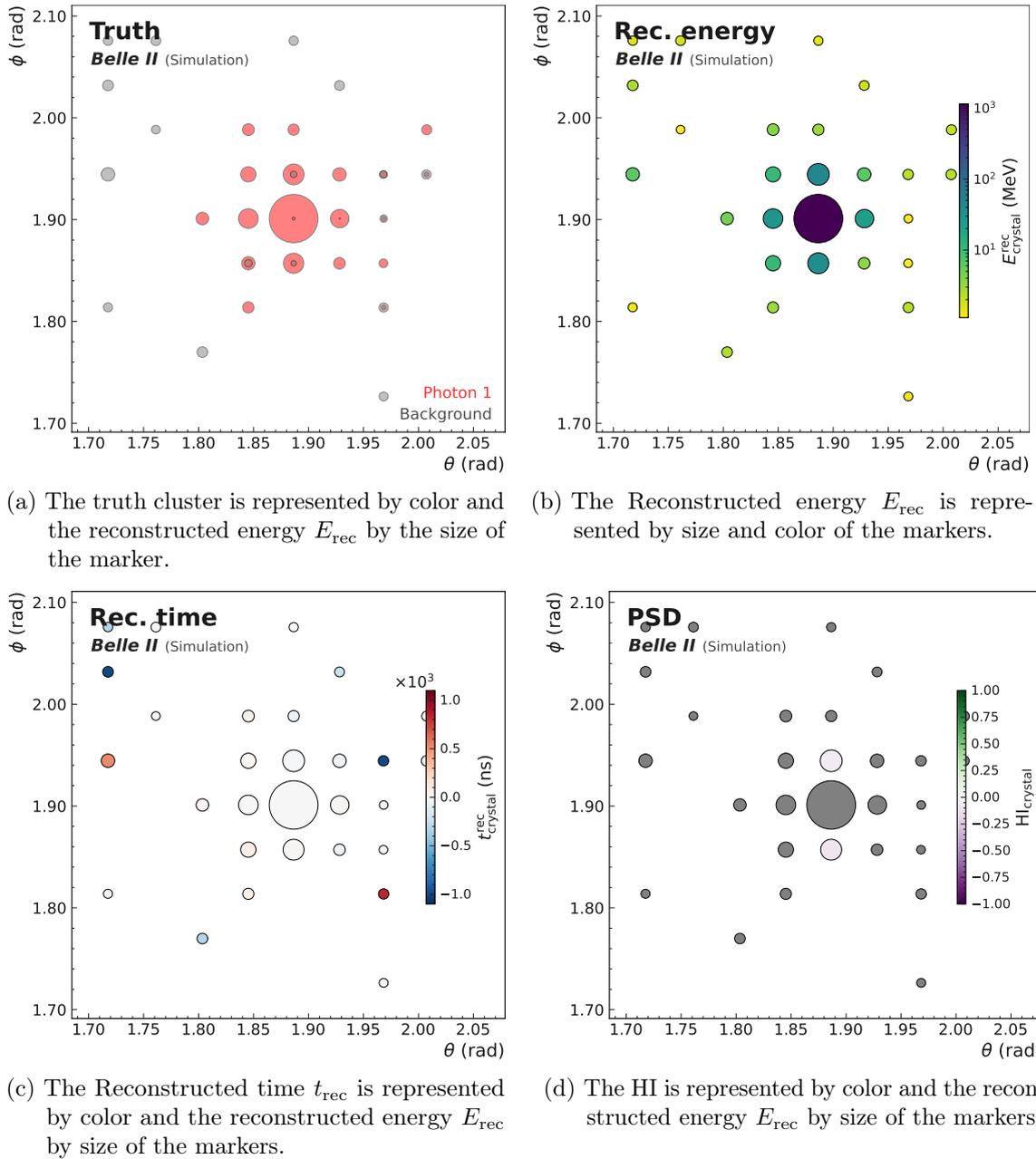
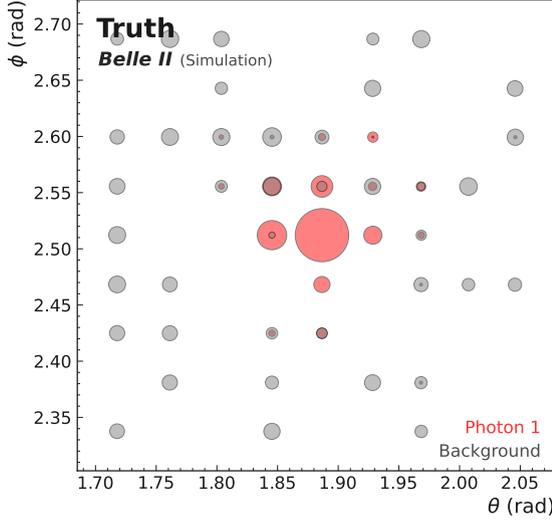
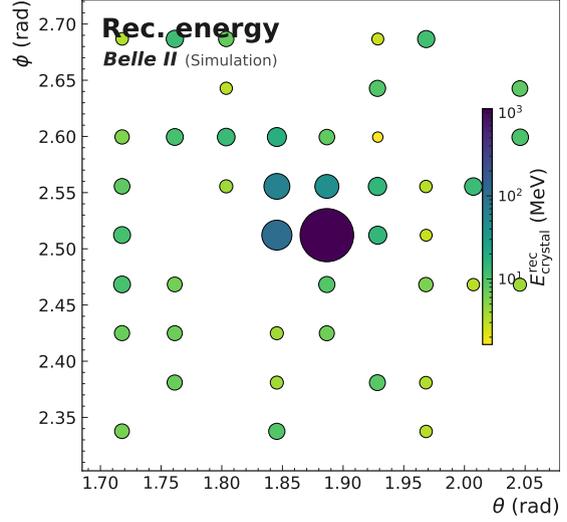


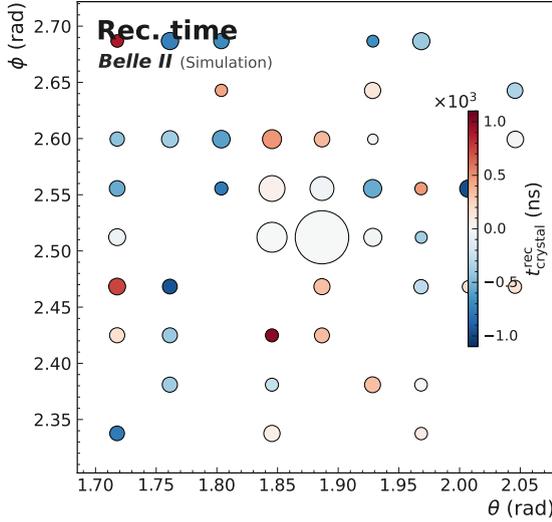
Figure 4.1.: Representation of a typical low beam background event. The crystals in the region of interest are displayed. ϕ and θ are the detector coordinates. The size of the circles for the reconstructed energy is scaled by $\sqrt{E_{\text{rec}}}$ in all plots to improve the visibility of low-energy crystals. The figures are adapted from [1].



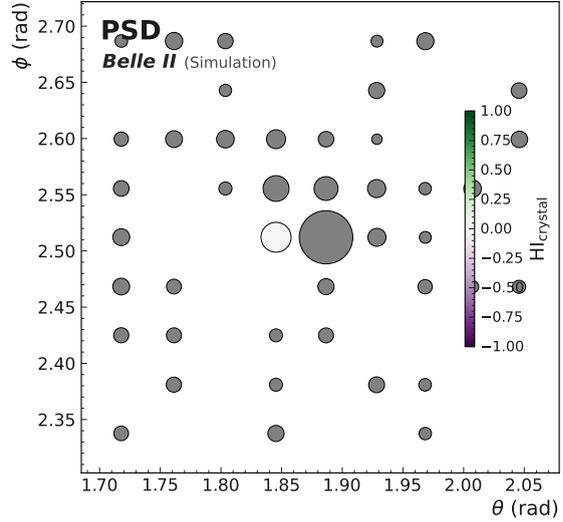
(a) The truth cluster is represented by color and the reconstructed energy E_{rec} by the size of the marker.



(b) The Reconstructed energy E_{rec} is represented by size and color of the markers.



(c) The Reconstructed time t_{rec} is represented by color and the reconstructed energy E_{rec} by size of the markers.



(d) The HI is represented by color and the reconstructed energy E_{rec} by size of the markers.

Figure 4.2.: Representation of a typical high beam background event. The crystals in the region of interest are displayed. ϕ and θ are the detector coordinates. The size of the circles for the reconstructed energy is scaled by $\sqrt{E_{\text{rec}}}$ in all plots to improve the visibility of low-energy crystals. The figures are adapted from [1].

5. Training and Implementation

This section focuses on the training and implementation of the GNN algorithms. The training and implementation of the GravNet algorithm were done in [1] and therefore are only summarized, while the training and implementation of the GCN with the GCN layer is done as part of this thesis and therefore described in detail.

Throughout the subsequent sections of this chapter, the term "GNN algorithms" refers to both the GravNet algorithm, comprising the GravNet layer, and the GCN algorithm, utilizing the GCN layer, as discussed in section 3.2 and section 3.3, respectively.

The graph representation with its node and edge input features is described in section 5.1. These graphs serve as an input for the GNN models, whose architecture is explained in section 5.2. Subsequently, the implementation of the graph representations and the full training process is described in section 5.3.

5.1. Representation of Events by Graphs

Both the GravNet and GCN algorithms operate on graphs as their input. These graphs are constructed using the event information discussed in chapter 4. Additionally, the location of the ROI within the ECL is considered, and only the 9×9 crystals within the ROI are included in the graph representation. Each event corresponds to a single graph, where each crystal within the ROI is represented as a node. The details of the node features can be found in section 5.1.1.

While the GravNet algorithm dynamically builds the edges during training within the GravNet layer (see section 3.2), the edges for the graphs used as input in the GCN algorithm are constructed statically. The process of edge building for the GCN algorithm is described in section 5.1.2.

One advantage of GNNs, in contrast to other neural network architectures like CNNs, is their ability to handle varying input sizes. As a result, only a subset of the 81 crystals in the ROI, specifically those with reconstructed energy greater than 1 MeV to filter out noise-related energy depositions, are included in the graph representation. Consequently, the size of the graphs varies from event to event, as different crystals are eliminated, resulting in a different number of nodes in each graph. On average, approximately 20 crystals are used as graph nodes, for LBB conditions and around 45 for HBB events.

5.1.1. Node Input Features

Each node in the graph represents a crystal within the ROI that has sufficiently high reconstructed energy. These nodes possess features containing information about each crystal. Prior to being used in training, the node input features undergo pre-processing to ensure their suitability and equal importance within the model. This is achieved by bringing features outside of $[0,1]$ within this range by normalizing them which was conducted as part of [1] and is summarized here.

The available node input features include the reconstructed energy E_{rec} , the reconstructed time t_{rec} , and the PSD information (HI, fit type, and χ^2) as described in section 2.2.1.

Since the reconstructed energy E_{rec} per crystal in GeV ranges naturally from 0 GeV to approximately 1 GeV and only a few outliers to the desired range are expected which are considered within tolerance, no pre-processing is needed.

The reconstructed time t_{rec} per crystal in μs ranges from $-1 \mu\text{s}$ to $1 \mu\text{s}$. Since the magnitude of the values is in the desired range and no issue during training and evaluation occurs, the reconstructed time is not pre-processed.

The PSD information consists of three features. The HI is capped at $[-1,1]$ where $\text{HI} > 0$ indicates a hadronic shower and $\text{HI} \leq 0$ indicates a fully electromagnetic shower. Due to the magnitude of the HI, no pre-processing is needed. The fit type is given by four distinct values $\{-1,0,1,2\}$, indicating the fitted template or failure of the fit. The values are mapped to $\{0,1/3,2/3,1\}$. The corresponding χ^2 is divided by 200 and capped at 1. The range $[0,1]$ then contains 90% of the χ^2 distribution.

Other possible node input features, describing the crystal properties, include the crystal mass m , crystal coordinates, and the LM indicator.

The crystal mass m , ranging from 4.03 kg to 5.94 kg, is globally normalized using min-max normalization to the range $[0,1]$ with the `scikit-learn` package [22].

To locate each crystal unambiguously, the two global coordinates θ and ϕ are sufficient. Global coordinates have the advantage over Cartesian coordinates by also representing the cylindrical form of the ECL. Both θ and ϕ naturally range from 0 to 2π . A min-max normalization is applied to θ resulting in the desired range $[0,1]$. For ϕ , the normalization is performed on the features $\phi_{\text{sin}} = \sin \phi$ and $\phi_{\text{cos}} = \cos \phi$ to include the edges of the interval.

Local coordinates θ' and ϕ' are useful as the crystal representing the LM is defined as the origin of the ROI (see chapter 4). Additionally, the local coordinates help the network so it does not have to learn the continuity in ϕ . θ' and ϕ' represent the respective angular separations in radian to the origin. Since the coordinates are local, the normalization to the range $[0,1]$ is done per event for the ROI.

Lastly, the LM indicator serves as a node input feature, taking the value 1 if the crystal is the LM within the ROI and 0 otherwise.

5.1.2. Edge Input Features

This section focuses on the edge input features used in the graphs for the GCN algorithm, as the GravNet algorithm dynamically builds and learns edge features in the GravNet layer (see section 3.2).

In this work, only undirected edges are used to connect two nodes in the graph. The edge weights are provided as input to the GCN layer and represent the inverse of the Euclidean distance

$$\frac{1}{d_{ij}} = \left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right)^{-1/2}, \quad (5.1)$$

between the connected nodes of the respective crystals i and j . The distance d_{ij} is used as the edge feature for the majority of this thesis. Additionally, the algorithm is examined with the edge weight computed from the Euclidean distance of Cartesian coordinates x_i , y_i , z_i , and the reconstructed time t_{rec} (see section 7.3.2)

$$\frac{1}{d'_{ij}} = \left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 + (t_i^{\text{rec}} - t_j^{\text{rec}})^2 \text{ m}^2 \cdot \mu\text{s}^{-2} \right)^{-1/2}. \quad (5.2)$$

Since the edge weight is the only feature used for the edges, no pre-processing or normalization is performed on the weights.

The implementation of the node, as well as the edge input features, is described in section 5.3.1.

5.2. Architecture

The GCN and GravNet algorithms share a similar architecture. To achieve deep learning the architectures are extended and the layers are included in so-called blocks. Each block contains three fully connected layers, the GCN or the GravNet layer, and a batch normalization layer. Multiple blocks are stacked to create a deep learning architecture. Figure 5.1 illustrates the architecture with four stacked blocks.

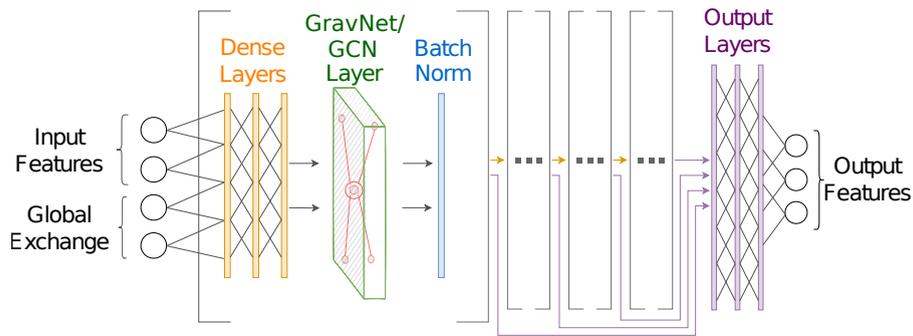


Figure 5.1.: A schematic view of the graph neural network architectures. In the beginning, a global exchange is performed. This architecture example consists of four stacked blocks. A block is visualized in black square brackets and each block includes three dense layers, the graph neural network layer (GravNet or GCN), and a batch norm layer. Three final fully collected layers collect all blocks' output and transform the information into the final output. The figure is adapted from [20].

At the initial step and before every block, a global exchange operation is performed. This operation gathers global information from all nodes and extends the input features by averaging each feature per graph. The output of each block is passed as an input to the subsequent block and additionally, the block output is input for the final three fully connected layers. These final layers concatenate the outputs from each block and reduce the dimension to the desired number of output classes. For each cluster and background, one output class is required. In the case of the single photon energy reconstruction, there are two output classes: one for the photon cluster and one for the background. The input dimension for the GCN algorithm is given by the number of node input features.

The activation functions used in the GCN and GravNet architectures differ. The GravNet algorithm employs exponential linear units (ELUs) [23] for the first two fully connected layers and a hyperbolic tangent (tanh) activation for the last fully connected layer of each block. On the other hand, the GCN algorithm uses rectified linear units (ReLUs) [24] for all three fully connected layers. A softmax function is applied to the final output of both GNN architectures to ensure that the output satisfies equation (3.1). This guarantees the interpretation of the output as fractions of the reconstructed energy.

Hyperparameters of the architecture include the number of stacked blocks, the dimensions of the fully connected layers, and the batch normalization momenta. For the GravNet layer, a hyperparameter optimization was performed using `Optuna` [25]. The hyperparameters of the GCN model are discussed in section 5.3.3.

5.3. Implementation

The GNN algorithms are implemented and trained using the ML library `PyTorch` [26]. The GravNet and GCN layer are implemented using the `GravNetConv` and `GCNConv` layers from `PyTorch Geometric` [27]. The graph-building process is done using the package `NetworkX` [5].

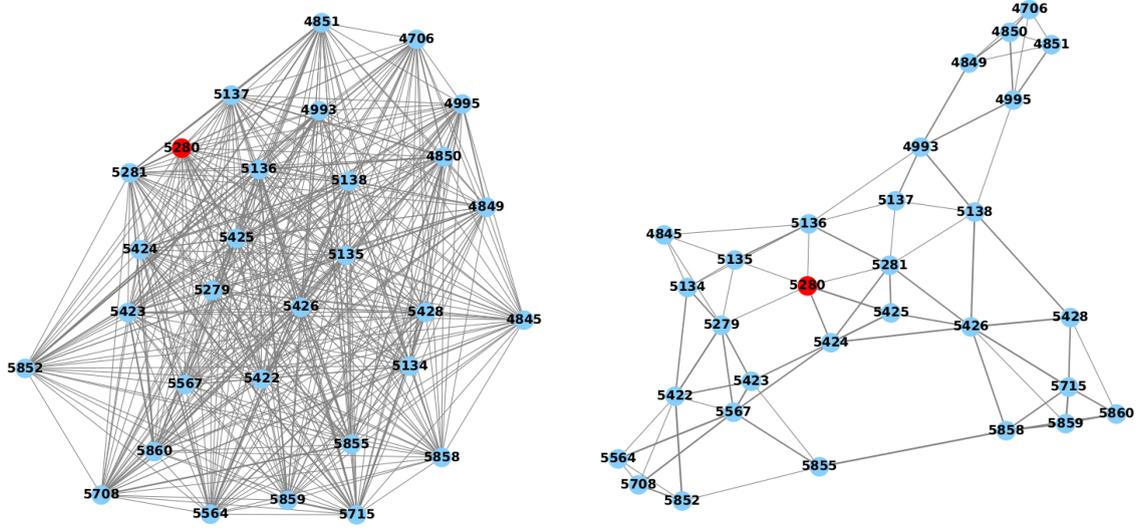
This work focuses on single photon clusters. Each model is trained and evaluated on events with LBB and on events with HBB. Because the training and evaluation as part of this thesis are done for the GCN model, the focus lies on the implementation of this particular model. The performance of the GCN model is compared later to that of the GravNet model and the `basf2` algorithm.

5.3.1. Input Features

The input features, described in section 5.1.1 and section 5.1.2, are implemented crystal-wise. Each node represents a crystal and includes up to ten node input features with the corresponding pre-processed values of the reconstructed energy E_{rec} , the reconstructed time t_{rec} , the HI, the crystal mass m , the three global coordinates θ , ϕ_{sin} , and ϕ_{cos} , and the two local coordinates θ' and ϕ' , and the LM indicator.

The impact of individual node input features is studied in section 7.2.

To connect the nodes to a graph edges are used. The number of edges and their edge weights are studied in section 7.3. One representation of the graph involves connecting each node in the ROI with all other nodes through edges. The edge weight is determined by the



(a) Single photon event graph where each node is connected by an edge to all other nodes.

(b) Single photon event graph where each node is connected by an edge to its $k = 4$ nearest neighbors.

Figure 5.2.: Pictorial representations of a single photon event graph. All nodes are shown in blue and labeled with their respective crystal id. The local maximum crystal is colored in red. The edges are visualized by gray lines and the edge weight by the thickness of the line.

inverse of the Euclidean distance based on Cartesian coordinates. Figure 5.2a provides a visualization of this graph structure. This way each node has direct access to information of every other node in the graph. Alternatively, in figure 5.2b, the number of edges in the example event graph is reduced to the k nearest neighbors of each node. The k nearest neighbors of each node are determined by the Euclidean distance in the ECL and then an edge is drawn with the inverse distance as the edge weight.

5.3.2. Loss Function and Optimizer

The GNN models are supervised ML algorithms with learnable model parameters. This means that the parameters are adjusted by an optimizer to minimize a loss function.

A comparison of the predicted fractions $p_i^{(u)}$ from the model and the true fraction $t_i^{(u)}$ of the MC training's data set defines the loss. The training of the GCN model deploys the mean squared error (MSE) loss function

$$\text{MSE} = \frac{1}{N_u} \frac{1}{N} \sum_{u,i} (p_i^{(u)} - t_i^{(u)})^2, \quad (5.3)$$

where the squared difference between the prediction and truth of each crystal i is summed over all output classes u and all crystals i and divided by the number of output classes N_u and the number of nodes N in the graph.

The optimizer Adam [28] optimizes the weights to reach the loss function's minimum based on gradient descent. Furthermore, the first and second moments of the gradients are used to adjust the learning rate adaptively. At the beginning of each training, the learning rate is set to 0.01 and decays by a factor of 0.25 if the validation loss does not decrease for five epochs.

5.3.3. Training

The GCN model is either trained on the full ECL, including the barrel and both endcaps or only the barrel region. Depending on the ECL region used for the training the number of events varies.

When training on the full ECL, a total of two million events were generated and selected according to the process described in chapter 4. Out of these events, 400 000 events are chosen as the validation data set, while the remaining 1.6 million events are used for training. If the training is performed on the barrel region alone, 20 000 events are used, with 4 000 events used for validation and the remaining 16 000 events used for training. The training is performed with mini-batches of the size 1 024. This applies to both beam background types.

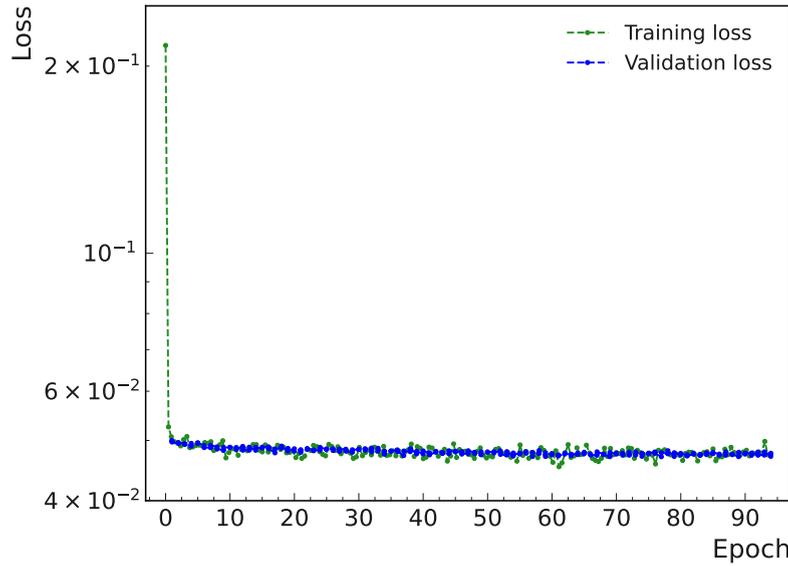
Every training is evaluated with events taking place in the barrel region. For the case of training on the full detector, 200 000 barrel events are used for testing. If the training is done on the barrel region, a separate data set of 20 000 barrel events is used for testing the GCN model. The details of the training and testing regions for each toy study can be found in chapter 7.

The hyperparameters of the GCN architecture include the dimensions of the fully connected layers, the dimensions of the GCN layer, the number of blocks, and the batch normalization momentum. The batch normalization momentum is set to 0.01. After evaluating the training on 20 000 barrel LBB events for different hyperparameter configurations, the number of blocks is set to 4, and the in- and output dimensions of the fully connected and GCN layers are set to 9.

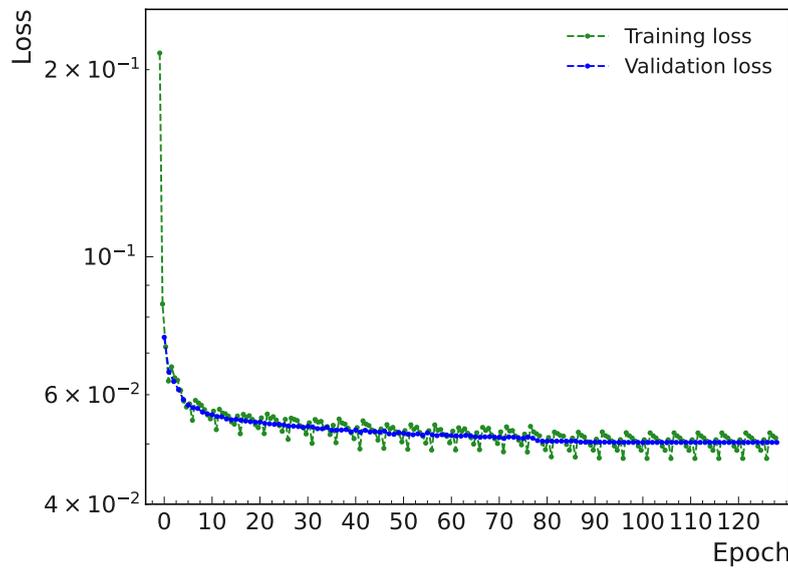
This hyperparameter configuration leads to 2 635 learnable parameters for the GCN model if the input dimension is ten and therefore all node input features are included in the graph. In comparison to the GravNet algorithm, the total number of model parameters for the GCN model is a sixth of the number for the GravNet architecture with 15 799 learnable parameters [1].

The training process stops and the last configuration is saved when the validation loss does not decrease for 15 epochs. Figure 5.3 provides two examples showing the training and validation loss curves.

The training duration depends on the number of events used. Training on 2 million events takes around 7 hours while the training on 20 000 events typically takes approximately 5 to 10 minutes. These duration are based on training using a single NVIDIA Titan X 12GB GPU. During training, the GPU memory is loaded with 6881 MiB.



(a) Training and validation loss of the training of 2 million full detector low beam background events.



(b) Training and validation loss of the training of 20 000 full low beam background events.

Figure 5.3.: Loss and validation loss during training in dependency of the training epoch. Plot (a) shows the loss of the training of 2 million full detector low beam background events. The plot (b) shows the loss of the training of 20 000 barrel low beam background events.

6. Metrics

This chapter presents the metrics used to assess the performance of the GNN models. First, the fundamental reconstruction quantities are introduced in section 6.1. Following that, the performance of the models is evaluated in section 6.2. The evaluation involves determining the relative reconstruction errors by comparing the predicted data generated by the algorithms with the true data obtained from the MC information. Additionally, the energy resolution of the GCN model is analyzed and compared to the performance of the GravNet and basf2 models. This analysis provides insights into the accuracy and precision of the energy reconstruction achieved by each model.

6.1. Fundamental Reconstruction Quantities

The trained algorithms have as an output the clustering weights $w_i^{(u)}$. The output classes u contain the photon signal and the background. Here, the weights describe the predicted fractions of the reconstructed energy per crystal i , i.e. $w_i^{(u)} = p_i^{(u)}$. Furthermore, equation (3.1) must hold true, which means for the predicted fraction of the reconstructed background energy

$$p_i^{\text{background}} = (1 - p_i^{\text{photon}}). \quad (6.1)$$

Because the goal of this thesis is to evaluate the performance of the different GNN algorithms regarding the accuracy of the single photon energy reconstruction as described in section 6.2, from now on only the single photon cluster is described and therefore the notation is reduced to $p_i^{\text{photon}} = p_i$. Since the fractions alone have no physical meaning, the predicted reconstructed energy per crystal i needs to be determined by combining the predicted fractions and the reconstructed energy E_i^{rec} . This leads to the fundamental reconstruction quantities per crystal:

$$E_i^{\text{dep}} = E_i^{\text{rec}} t_i \quad \text{and} \quad E_i^{\text{pred}} = E_i^{\text{rec}} p_i. \quad (6.2)$$

E_i^{dep} corresponds to the true amount of energy the MC photon deposited in crystal i , while the predicted energy per crystal i is denoted as E_i^{pred} . The total true (deposited) energy E_{dep} and the total predicted energy E_{pred} are defined by summation over the energies of all crystals i :

$$E_{\text{dep}} = \sum_i E_i^{\text{dep}} \quad \text{and} \quad E_{\text{pred}} = \sum_i E_i^{\text{pred}}. \quad (6.3)$$

Event property cross-checks for the GravNet algorithm can be seen in the thesis [1]. As part of this thesis, only the performance evaluation is conducted for the GCN algorithm.

6.2. Performance Evaluation

This work aims to assess the accuracy of photon energy reconstruction. To evaluate the GNN algorithms, the comparison of just the output weights and the MC information is not sufficient for this purpose. Instead, entire events are analyzed representing the performance of the algorithms in a more intuitive way. The improvement in the reconstruction of the photon energy is evaluated using the reconstruction errors discussed in section 6.2.1. Additionally, the energy resolution is determined based on the reconstruction errors, as described in section 6.2.2.

6.2.1. Reconstruction Errors

The primary objects for performance evaluation are the relative reconstruction errors, also called reconstruction errors. In this thesis, two types of reconstruction errors are studied. Either the reconstruction error on the deposited energy E_{dep} or the reconstruction error on the generated energy E_{gen} . While the deposited energy does not factor in any detector effects such as leakage, the generated energy includes these effects. The performance evaluation is conducted on all events of a test data set. The distribution of the reconstruction errors of all the tested events shows a peak. This peak is characterized by its full width half maximum (FWHM) and tails that are described in detail in section 6.2.2.

The **reconstruction error on the deposited energy** is determined by

$$\eta_{\text{dep}} = \frac{E_{\text{pred}} - E_{\text{dep}}}{E_{\text{dep}}}, \quad (6.4)$$

with the deposited energy E_{dep} and the predicted photon energy E_{pred} as defined in equation (6.3).

η_{dep} gives a direct evaluation of the clustering algorithm because it does not take detector effects such as leakage into account. The reconstruction errors are not absolute values, they rather distinguish whether the algorithm overestimates or underestimates the photon energy.

The **reconstruction error on the generated energy** is determined by

$$\eta_{\text{gen}} = \frac{E_{\text{pred}}^{\text{cor}} - E_{\text{gen}}}{E_{\text{gen}}}, \quad (6.5)$$

with the generated photon energy $E_{\text{gen}} = E_{\gamma}$ and the leakage-corrected, predicted energy

$$E_{\text{pred}}^{\text{cor}} = \begin{cases} \text{clusterE} & \text{basf2} \\ E_{\text{pred}} & \text{GravNet, GCN.} \end{cases} \quad (6.6)$$

The leakage-corrected total photon energy clusterE of the basf2 algorithm (see section 2.3) is used for comparison. The GNN algorithms are trained on the deposited energies as truth. Therefore, the algorithms do not factor in any detector effects and $E_{\text{pred}}^{\text{cor}} = E_{\text{pred}}$ as defined in equation (6.3).

η_{gen} takes into account detector effects and determines how much of the improvements in the underlying clustering carry over to actual physics reconstruction. With the perfect

clustering algorithm, the error on the deposited energy would technically be zero. In contrast, even with perfect clustering, the reconstruction error on the generated energy is never zero due to the limitations of the hardware. Consequently, η_{gen} is on average significantly larger than η_{dep} . This effect is shown in the form of a broader, spread-out peak when comparing the complete distributions of η_{gen} and η_{dep} in figure 6.1. The difference between the two distributions is particularly noticeable in the bigger left tail of the η_{gen} distribution.

6.2.2. Energy Resolution

The previous section 6.2.1 presents the distributions of the relative reconstruction errors η_{dep} and η_{gen} . Their characteristics are quantified by the FWHM. Most of the evaluated events are concentrated in the peak of these distributions. The FWHM measures the width of the peak and serves as a single metric to evaluate the performance of an algorithm on a test data set. It is also known as the resolution. A wider, more spread-out peak results in a larger FWHM, indicating a worse resolution or poorer performance. Apart from the peak itself, the tails of the distributions are significant for identifying outliers.

FWHM

The FWHM of a given distribution of relative reconstruction errors $f(\eta)$ is defined as

$$\text{FWHM} = |\eta_1 - \eta_2|, \quad (6.7)$$

with η_1 and η_2 being related to the maximum of the distribution $f(\eta_{\text{max}})$ by

$$f(\eta_1) = f(\eta_2) = \frac{1}{2}f(\eta_{\text{max}}). \quad (6.8)$$

To determine the FWHM a binned χ^2 -fit is performed with the `zfit` package [29]. Before the fit is performed the fit range

$$r_{\text{fit}} = [r_{\text{L}}^{\text{fit}}, r_{\text{R}}^{\text{fit}}], \quad (6.9)$$

is estimated with $r_{\text{L/R}}^{\text{fit}}$ being the left and right limits of the range.

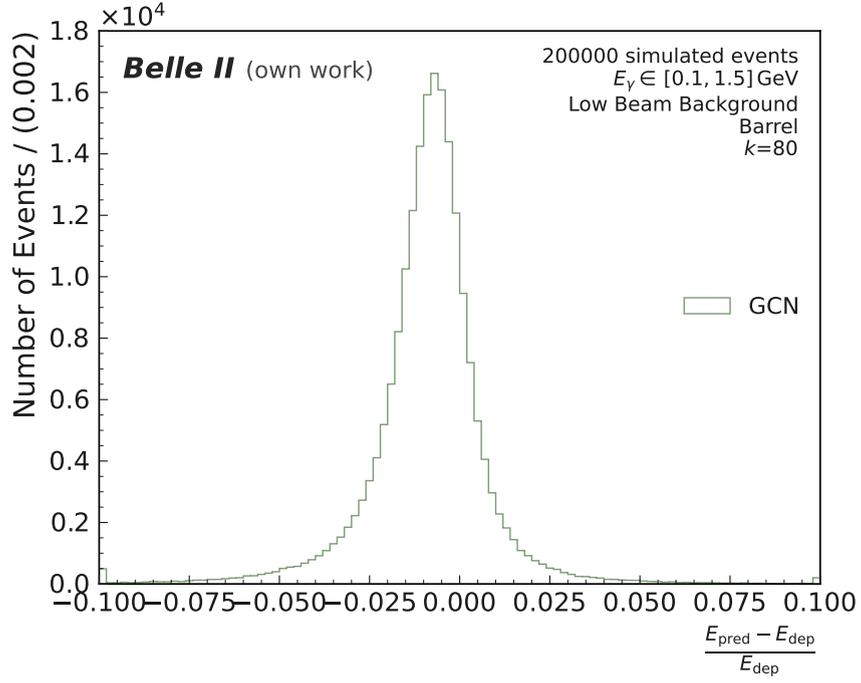
The distribution of the reconstruction errors is fitted with a double-sided crystal ball function (DCB) which is defined by

$$f(x; N, \mu, \sigma, \alpha_{\text{L}}, n_{\text{L}}, \alpha_{\text{R}}, n_{\text{R}}) = N \cdot \begin{cases} A_{\text{L}} \cdot (B_{\text{L}} - \frac{x-\mu}{\sigma})^{-n_{\text{L}}}, & \frac{x-\mu}{\sigma} < -\alpha_{\text{L}} \\ \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), & -\alpha_{\text{L}} \leq \frac{x-\mu}{\sigma} \leq \alpha_{\text{R}} \\ A_{\text{R}} \cdot (B_{\text{R}} - \frac{x-\mu}{\sigma})^{-n_{\text{R}}}, & \frac{x-\mu}{\sigma} > \alpha_{\text{R}} \end{cases} \quad (6.10)$$

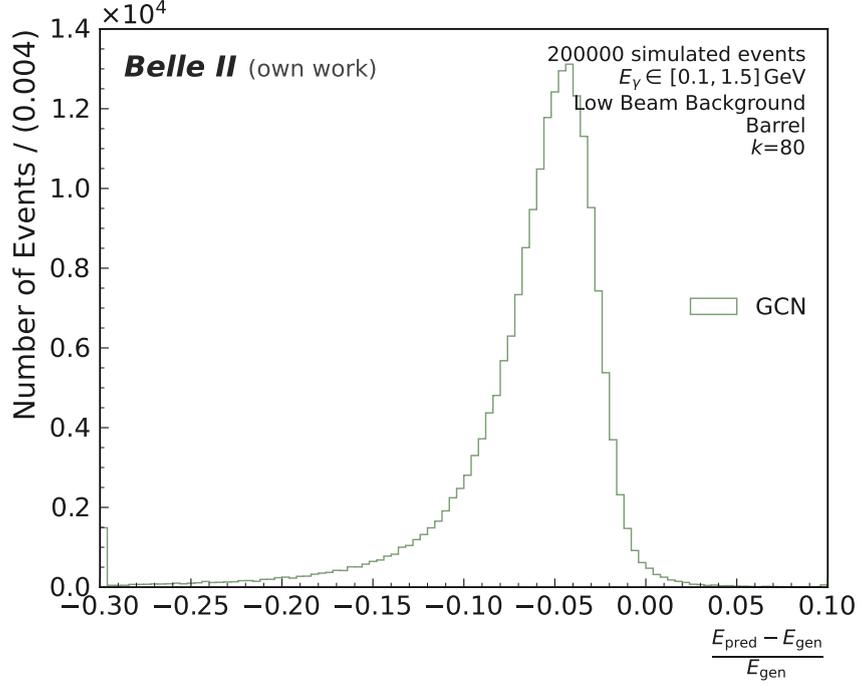
with

$$A_{\text{L/R}} = \left(\frac{n_{\text{L/R}}}{|\alpha_{\text{L/R}}|}\right)^{n_{\text{L/R}}} \cdot \exp\left(-\frac{|\alpha_{\text{L/R}}|^2}{2}\right), \quad B_{\text{L/R}} = \frac{n_{\text{L/R}}}{|\alpha_{\text{L/R}}|} - |\alpha_{\text{L/R}}|. \quad (6.11)$$

The height of the DCB is given by N . The Gaussian distribution with the mean μ and the standard deviation σ characterizes the peak while the transition factors $\alpha_{\text{L/R}}$ and the exponent $n_{\text{L/R}}$ describe the tails.



(a) Relative reconstruction error on the deposited energy.



(b) Relative reconstruction error on the generated energy.

Figure 6.1.: Distribution of the relative reconstruction errors $f(\eta_{\text{dep/gen}})$ of an example test data set of the graph convolutional network model trained on low beam background events. The first bin contains all underflow entries and the last bin contains all overflow entries. The results are from the toy study in section 7.4.

After the fit is performed, the FWHM is determined. Since the distribution and also the DCB is asymmetric, the left and right $\text{FWHM}_{L/R}$ are determined separately. $\text{FWHM}_{L/R}$ are obtained depending on the transition factors $\alpha_{L/R}$ and therefore depend if $\text{FWHM}_{L/R}$ are in the Gaussian part or the exponential tails:

$$\text{FWHM}_{L/R} = \begin{cases} \sqrt{2 \log 2} \cdot \sigma, & \alpha_{L/R} > \sqrt{\log 4} \\ \left| \mu + \left(\frac{\sigma}{\alpha_{L/R}} (|\alpha_{L/R}|^2 + S_{L/R} - n_{L/R} - \frac{\mu |\alpha_{L/R}|}{\sigma}) \right) \right|, & \alpha_{L/R} \leq \sqrt{\log 4} \end{cases} \quad (6.12)$$

with

$$S_{L/R} = \left(\frac{1}{2} n_{L/R}^{-n_{L/R}} \cdot \exp(\alpha_{L/R}^2/2) \right)^{-1/n_{L/R}}. \quad (6.13)$$

The FWHM is then determined by

$$\text{FWHM} = \text{FWHM}_L + \text{FWHM}_R. \quad (6.14)$$

The uncertainty of the FWHM δ_{FWHM} are determined from the correlated uncertainties of the fit parameters using the `uncertainties` package [30]. As a final step, the FWHM and δ_{FWHM} get corrected

$$\text{FWHM}_{\text{cor}} \pm \delta_{\text{FWHM}}^{\text{cor}} = (1 - \mu) \cdot (\text{FWHM} \pm \delta_{\text{FWHM}}). \quad (6.15)$$

This guarantees a fair comparison between different types of algorithms. It is equivalent to a shift of the peak and then centering the underlying distribution around zero. In the following sections FWHM_{cor} is denoted by $\text{FWHM}_{\text{dep/gen}}$ depending on the type of the relative reconstruction error.

The **energy resolution** is defined by the standard deviation of a Gaussian distribution

$$\sigma_{\text{dep/gen}} = \text{FWHM}_{\text{dep/gen}}/2.355. \quad (6.16)$$

Tail lengths

The FWHM characterizes the peak of the distribution of the relative reconstruction errors. It is necessary to additionally analyze the outliers. These outliers are characterized by the tail lengths $T_{L/R}$. Due to the potentially significant asymmetry of the tails, the lengths of the left tail T_L and the right tail T_R are studied separately. To determine $T_{L/R}$ the distribution is divided in half at the position of the peak μ . Then the total number of events on the respective side of the distribution are determined. The number of events, starting at μ , are counted until reaching 95% of the total number of events and the absolute distance between μ and the reconstruction error of the last event within the 95% range determines the tail length $T_{L/R}$.

To account for any bias introduced by the fitting process, the procedure is repeated twice, splitting at $\mu + \delta_\mu$ and $\mu - \delta_\mu$, respectively, to determine the uncertainties $\delta_T^{L/R}$ on the tail lengths. These uncertainties are corrected in a similar way to the FWHM as described in equation (6.15), taking into account potential biases induced by the fitting process:

$$T_{L/R}^{\text{cor}} \pm \delta_{T,\text{cor}}^{L/R} = (1 - \mu) \cdot (T_{L/R} \pm \delta_T^{L/R}). \quad (6.17)$$

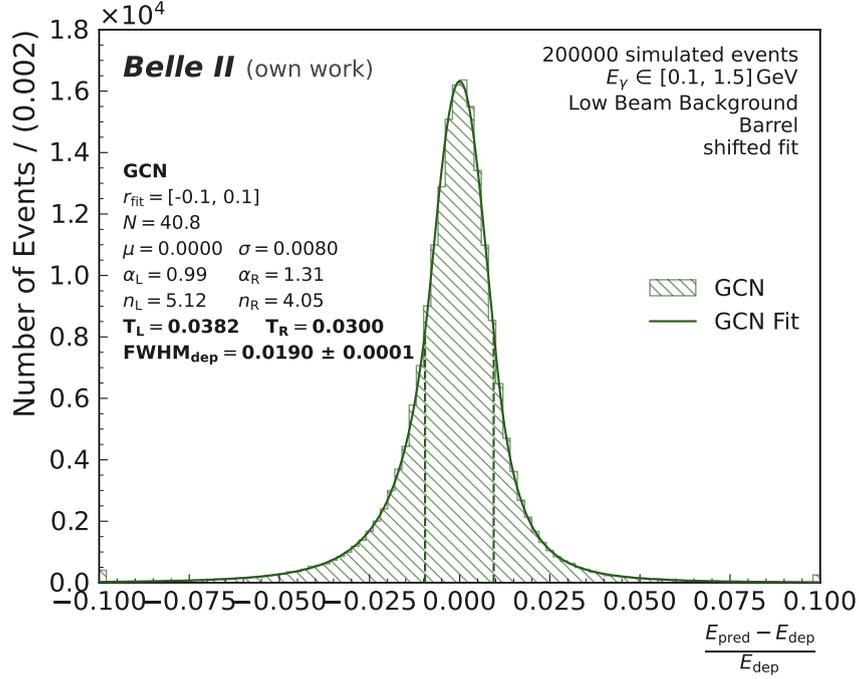


Figure 6.2.: Example of a double-sided crystal ball function fit for the distribution of the relative reconstruction errors on the deposited energy $f(\eta_{\text{dep}})$ of an example test data set of the GCN model trained on low beam background events. The fit range, the fit parameters, the tails, and the full width half maximum (FWHM) are shown in the figure. The uncertainties on the tail lengths, given by $\mathcal{O}(\delta_T^{\text{L/R}}) \approx 0.1\%$, are extremely small and hence not shown in the limited space of the plot. The first bin contains all underflow entries and the last bin contains all overflow entries. The results are from the toy study in section 7.4.

In the following sections $T_{\text{L/R}}^{\text{cor}}$ is replaced by $T_{\text{L/R}}$ for the sake of readability denoting the corrected tail lengths.

In figure 6.2, the distribution of η_{dep} is presented for an example data set. Additionally, the figure includes the quantities r_{fit} , the fit parameters, and the resulting FWHM and tail lengths $T_{\text{L/R}}$.

Energy dependence

Figure 6.3 shows the energy resolution of several different photon energies. The energy dependence of the resolution is determined by analyzing the resolutions of photons that are generated at different fixed energies. If the energy resolutions are plotted in dependency of the generated photon energy E_γ they follow the function

$$\sigma_{\text{dep/gen}}(E_\gamma) = \frac{a}{\sqrt{E_\gamma(\text{GeV})}} - b, \quad (6.18)$$

with a and b being constants. The full energy dependence shows the performance spectrum of the algorithms and is studied in section 7.4.

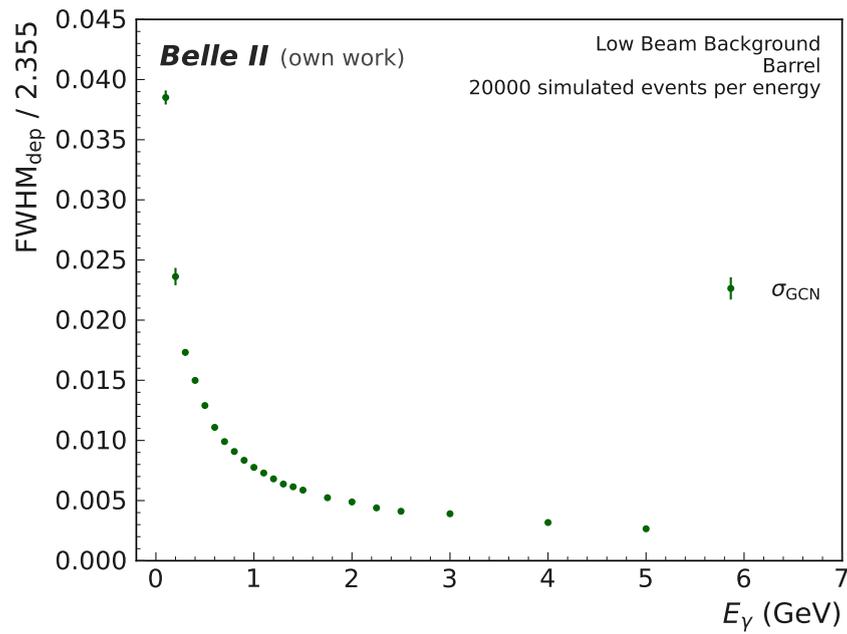


Figure 6.3.: Example of the energy resolution dependence $\sigma_{\text{gen}}(E_\gamma)$ of the relative reconstruction errors on the deposited energy $f(\eta_{\text{dep}})$ of example test data sets of the GCN model trained on low beam background events. The results are from the toy study in section 7.4.

7. Toy Studies for Single Photon Clusters

The goal of this section is to evaluate and understand the GCN algorithm's performance based on different graph input features. This evaluation is crucial since any physics application needs a good understanding of the model's performance.

The study focuses on analyzing single photon clusters using different graph input features in conjunction with LBB. The optimal configuration of the graph input feature is then trained on HBB events as well. While the training primarily utilizes full detector events, the evaluation is always done on a test data set consisting of events from the barrel region.

In the first section 7.1, a brief motivation for the toy studies is presented, and the settings for each of them are explained. Subsequently, the performance of the GCN algorithm is evaluated by examining various graph input features. First, the node input features are investigated in section 7.2. Then, the edge input features are explored by adjusting the number of edges in the graph in section 7.3.1, followed by evaluating the model's performance for different edge weights in section 7.3.2. Each section provides an explanation of the respective graph input configurations and assesses the performance of the GCN model. Finally, in section 7.4, the energy dependence of the GCN model is analyzed for both LBB and HBB, and compared to the performance of the GravNet and basf2 algorithms.

7.1. Motivation and general Settings

Single photon clusters have a very distinctive signature and are numerous in the ECL. Many of these photons originate from the decays of B^+B^- and $B^0\bar{B}^0$ particles at Belle II. These photons typically have energies within the range of $E_\gamma \in [0.1, 1.5]$ GeV. Hence, this energy interval is selected for the generation of the training events.

Photons with energy below this interval are considered too challenging in this study. Furthermore, they have less impact in physics analyses since they are mostly due to radiation processes and are therefore not used for training and testing in this thesis. Technically the photon energy range could be extended to higher energies which could be part of further studies. Nonetheless, the GCN model is able to generalize to energies outside the training interval, as demonstrated in section 7.4.

The angular distribution of ϕ is defined as $\phi \in [0.0^\circ, 360.0^\circ]$, while the range of θ depends on the detector region. For the full detector, θ lies within the interval $\theta \in [17.0^\circ, 150.0^\circ]$, whereas for the barrel region, it is confined to $\theta \in [37.2^\circ, 123.7^\circ]$. The event generation

process involves creating photons with uniform distributions in space and momentum while staying within the ranges of the ECL regions. Afterwards, the events are overlaid with either LBB or HBB files to model the beam background correctly. The ROI must then meet the requirements for one-cluster events, as outlined in chapter 4. Training and testing events are generated independently but share the same settings and criteria. This approach ensures comparability while also preventing the model from being evaluated on the same events it was trained on.

7.2. Node Input Features

In this section, different node input features of the graphs for the GCN model are discussed. Possible node input features as described in section 5.1.1 are combined and the performance of the GCN algorithm is evaluated.

7.2.1. Settings

The available node input features for the graph, as discussed in section 5.1.1, include the reconstructed energy E_{rec} , the reconstructed time t_{rec} , the hadron intensity HI from the PSD information, the global crystal coordinates θ , ϕ_{sin} , ϕ_{cos} , the local coordinates θ' , ϕ' , the crystal mass m , and the LM indicator.

The basic node input features used for every training are the reconstructed energy E_{rec} , the global crystal coordinates θ , ϕ_{sin} , ϕ_{cos} , and the crystal mass m . Different combinations of the remaining input features, such as the reconstructed time t_{rec} , the hadron intensity HI, the local coordinates θ' , ϕ' , and the LM indicator, are employed to train, test, and evaluate the performance of the GCN model. To enhance readability, the excluded node input features are specified.

In the graph, each node is connected to every other node through an edge. The weight of the edge between two nodes is determined by the inverse Euclidean distance, as defined in equation (5.1).

Each training data set, with different combinations of node input features, consists of a total of two million full detector events with LBB. To assess the GCN algorithm's performance, it is tested on 200 000 barrel events with LBB.

7.2.2. Performance Evaluation

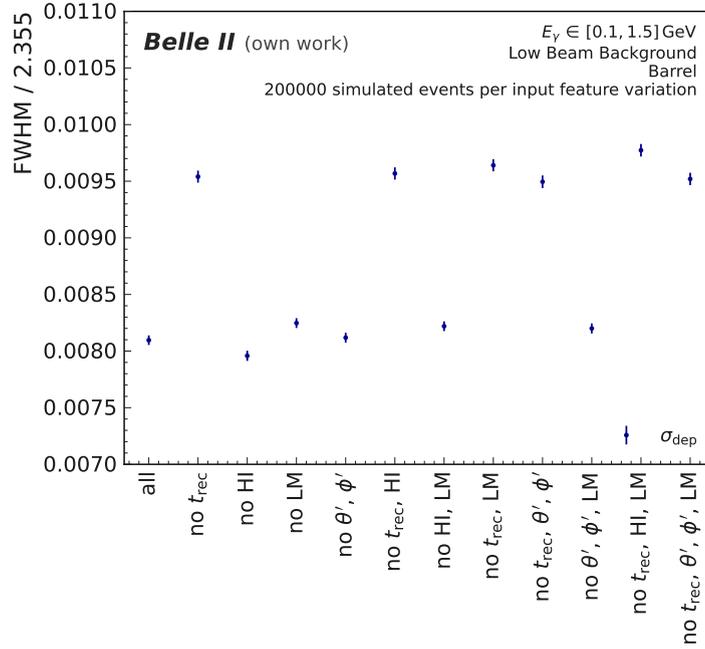
Figure 7.1 shows the energy resolution on the deposited energy σ_{dep} as well as the corresponding tail lengths T_{dep} for a test set of 200 000 single photon events with the energy $E_{\gamma} \in [0.1, 1.5]$ GeV. This analysis is valuable as it allows to verify the success of the training process and evaluate the clustering performance without considering leakage effects. It is shown in figure 7.1a that the absence of the reconstructed time t_{rec} in the node input features leads to poorer energy resolution. All other node input features have a negligible impact on the energy resolution, as they consistently yield a resolution of approximately $\sigma_{\text{dep}} \approx 0.0082$. This phenomenon is also observed in the tail lengths T_{dep} shown in figure 7.1b. When the reconstructed time is omitted from the node input features, the reconstruction error distribution becomes more dispersed, resulting in larger left and right tail lengths.

To assess the actual model performance and physical reconstruction accuracy, the energy resolution of the generated energy σ_{gen} is examined. However, displaying the energy resolution σ_{gen} for the entire test set of 200 000 single photon events with energy $E_\gamma \in [0.1, 1.5]$ GeV is challenging because the generated relative reconstruction error η_{gen} distribution is not conforming to a DCB, leading to inaccurate fit results. An example of such a distribution is depicted in figure A.1. Consequently, the energy resolution σ_{gen} is showcased for a test set of 20 000 single photon events with the fixed energy $E_\gamma = 0.3$ GeV in figure 7.2a, along with the corresponding tail lengths in figure 7.2b. Again, the energy resolution of the generated energy σ_{gen} is worse, and the left and right tail lengths $T_{\text{L/R,gen}}$ are larger when the node input features do not include the reconstructed time t_{rec} .

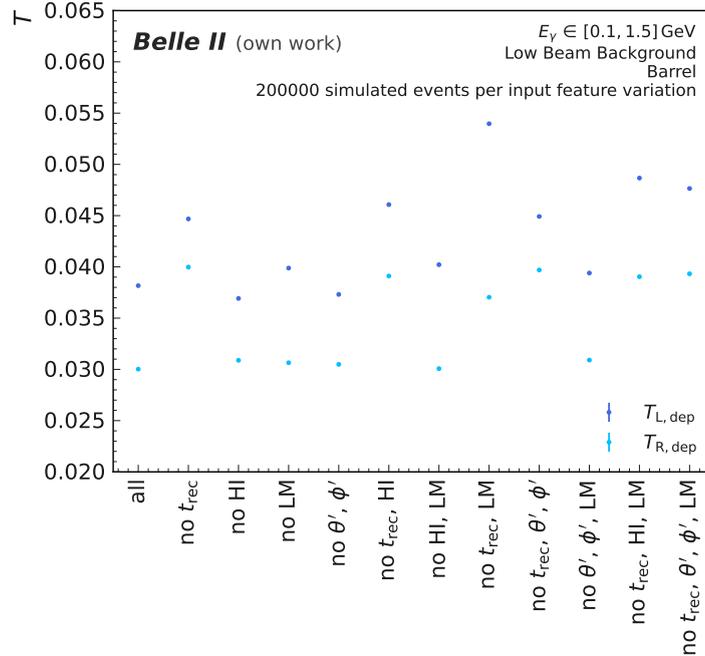
7.2.3. Overview and Conclusions

The reconstructed time t_{rec} , along with the basic node input features including the reconstructed energy E_{rec} , the global crystal coordinates θ , ϕ_{sin} , ϕ_{cos} , and the crystal mass m , are the only relevant node input features while working with single photon clusters. This observation aligns with expectations since beam background crystals typically have reconstructed times closer to $t_{\text{rec}} = \pm 1 \mu\text{s}$, while crystals with energy deposition from photons generally have reconstructed times around $t_{\text{rec}} \approx 0 \mu\text{s}$. The GCN model can effectively learn this correlation and performs better when provided with time information. Other node input features such as the HI, the LM indicator, and the local coordinates have almost no impact on the model's performance. This outcome is anticipated since the HI is consistently around 1 for all crystals in single photon events comprising purely electromagnetic showers. It is expected that the HI would have an impact in events containing hadronic particles, thus it is retained in the node input features for future studies. The LM indicator and local coordinates do not significantly affect the algorithm's performance, as the LM information can be learned from the reconstructed energy E_{rec} and from the ROI since the event is located by definition around the LM. In addition to the global coordinates, the local coordinates do not contribute relevant information.

In conclusion, only the reconstructed time t_{rec} , in addition to the basic node input features, is found to be relevant. Considering that the time and computational cost for training and testing the GCN model are similar across all node input feature combinations and further studies might benefit from these additional input features, all possible node input features are used in other toy studies for consistency.

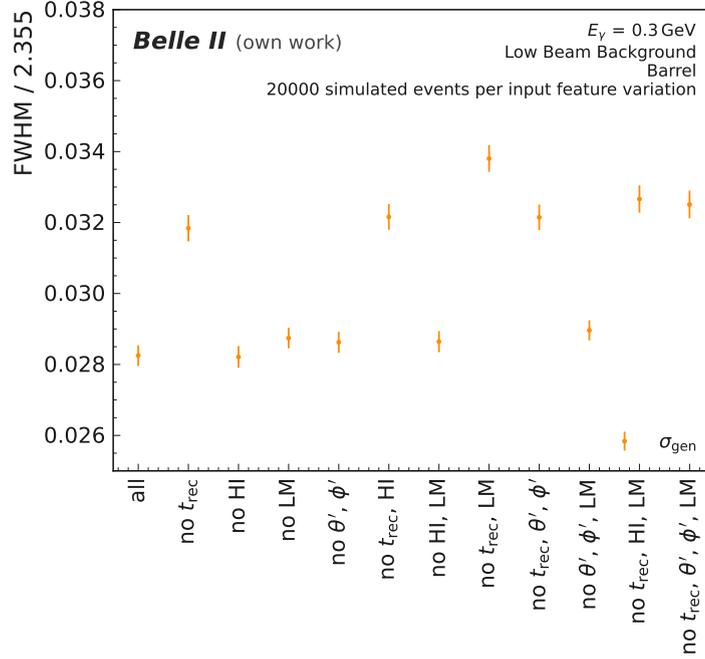


(a) Energy resolution of different node input feature combinations.

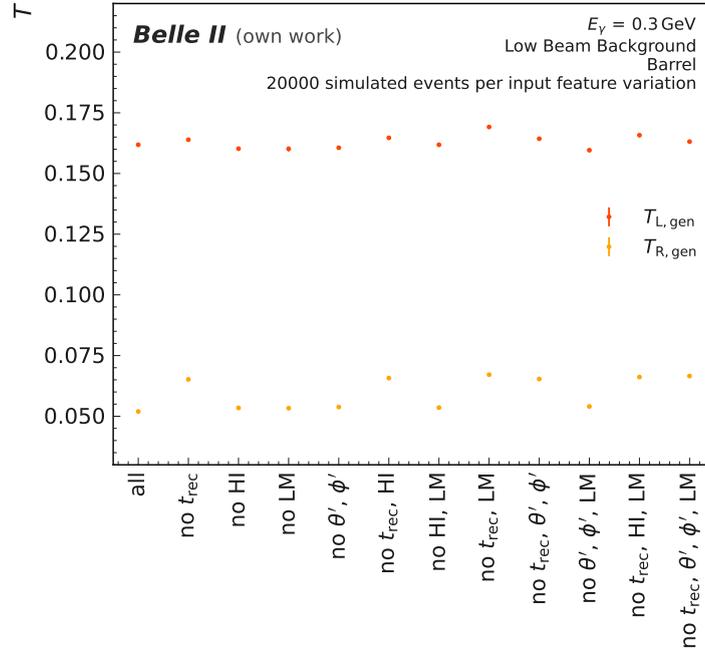


(b) Tail lengths of different node input feature combinations.

Figure 7.1.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of the deposited energy for different node input feature combinations of the reconstructed time t_{rec} , the hadron intensity HI, the local maximum indicator LM and the local coordinates θ', ϕ' . The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5] \text{ GeV}$ and tested on 200 000 barrel low beam background events with $E_\gamma \in [0.1, 1.5] \text{ GeV}$. Without the reconstructed time the energy resolution (a) is worse and both left and right tails (b) are longer compared to other node input feature combinations that include the reconstructed time.



(a) Energy resolution of different node input feature combinations.



(b) Tail lengths of different node input feature combinations.

Figure 7.2.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of the generated energy for different node input feature combinations of the reconstructed time t_{rec} , the hadron intensity HI, the local maximum indicator LM and the local coordinates θ', ϕ' . The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5] \text{ GeV}$ and tested on 20 000 barrel low beam background events with $E_\gamma = 0.3 \text{ GeV}$. Without the reconstructed time the energy resolution (a) is worse and both left and right tails (b) are longer compared to other node input feature combinations that include the reconstructed time.

7.3. Edge Input Features

In this section, different edge input features of the graphs for the GCN model are discussed. These edge input features are adjusted either by modifying the number of edges in the graph, as discussed in section 7.3.1, or by altering both the number of edges and the edge weights, as examined in section 7.3.2 to evaluate the performance of the GCN algorithm.

7.3.1. Number of Edges in the Graph

7.3.1.1. Settings

In this toy study, all possible node input features (see section 5.1.1) are employed as inputs for the graphs.

The number of edges in the graph is depending on the event and the number of edges each node is connected to. Each node gets an edge connection to its k nearest neighboring nodes. The distance between two nodes i and j is given by the Euclidean distance

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, \quad (7.1)$$

with the Cartesian coordinates $x_{i/j}$, $y_{i/j}$, and $z_{i/j}$ of the corresponding crystals. The edge weight is defined as the inverse of the Euclidean distance as described in equation (5.1).

Each training data set with a different number of edges consists of a total of two million full detector events with LBB. The performance of the GCN algorithm is then evaluated on a test set consisting of 200 000 barrel events with LBB.

7.3.1.2. Performance Evaluation

Figure 7.3 shows the energy resolution on the deposited energy σ_{dep} as well as the corresponding tail lengths T_{dep} for a test set of 200 000 single photon events with the energy $E_\gamma \in [0.1, 1.5]$ GeV. Similar to the previous analysis, this assessment serves as a check to verify the success of the training process and evaluate clustering performance without considering leakage effects, since the training is performed on the deposited energy. Figure 7.3a demonstrates that the energy resolution remains consistent across different numbers of k nearest neighbors. The training and testing for $k \in \{40, 50, 60, 70, 80\}$ is performed twice. The fluctuation of the training is by far bigger than the statistical error of the energy resolution and the tail lengths. This is particularly evident in the resolution of the deposited energy. In future work, it would be valuable to analyze the performance of the GCN model through multiple independent training and testing processes. The tail lengths T , depicted in figure 7.3b, further demonstrate the training-dependent fluctuations and indicate that the number of edges does not significantly impact the model's performance.

Again, it is important to examine the energy resolution and tail lengths of the relative reconstruction errors for the generated energy η_{gen} to assess the GCN model's performance in photon energy reconstruction. Figure 7.4 displays both the energy resolution and tail lengths of the reconstruction errors for the generated energy η_{gen} and the deposited energy η_{dep} for a test set of 20 000 single photon events with the energy $E_\gamma = 0.3$ GeV. The resolution of the generated energy depending on different training fluctuates within the statistical error range. Again, there is no significant influence observed from the number of edges.

7.3.1.3. Overview and Conclusions

The number of edges, determined by the number of k nearest neighbor connections between nodes in the graph, does not significantly impact the performance of the GCN algorithm. Surprisingly, even when each node is connected to all other nodes, the performance remains comparable to having $k = 10$ nearest neighbors. This observation can be attributed to the complex model architecture (see section 5.2), where information is gathered globally and locally and passed through multiple layers, facilitating high information exchange between nodes.

It is worth noting that the fluctuations observed during training are considerably larger than the statistical error in the energy resolution on the deposited energy and tail lengths. This aspect requires further investigation in future work.

Since the time and the computational cost for the training and testing of the GCN model is similar for all numbers of k nearest neighbors, $k = 80$ is used in the other toy studies. This implies that every node is connected to all other nodes in the graph. For actual real-time applications on the trigger level, the number of nearest neighbors must be reduced which could be studied in future work.

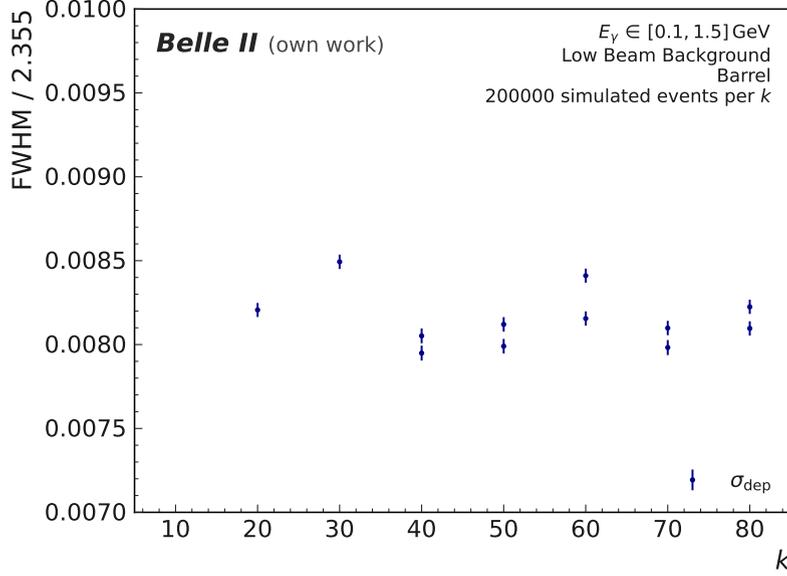
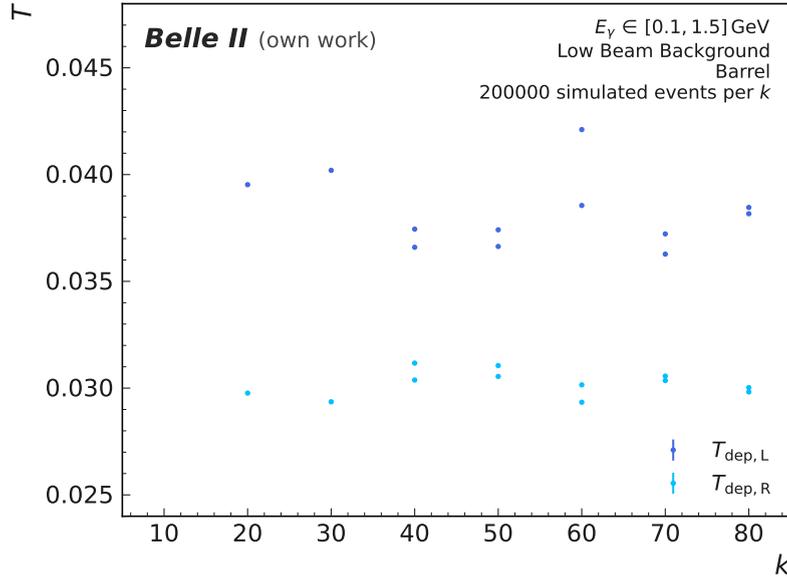
(a) Energy resolution of different k nearest neighbor edge connections.(b) Tail lengths of different k nearest neighbor edge connections.

Figure 7.3.: Energy resolution FWHM/2.355 (a) and tail lengths T (b) for different numbers of edges in the graph. The number of edges is determined from the k nearest neighbor edge connections in Euclidean distance between the crystal nodes. The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel events with low beam background and $E_\gamma \in [0.1, 1.5]$ GeV. The number of edges does not have a visible impact on the performance of the GCN algorithm.

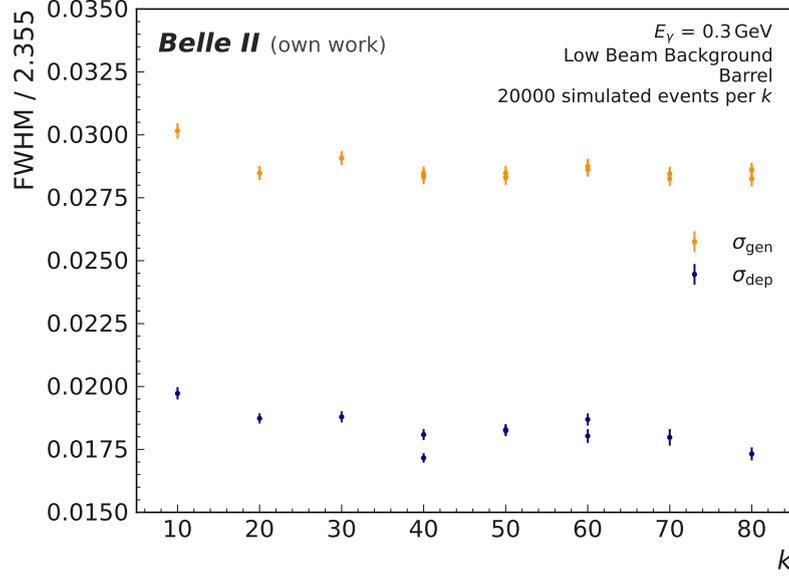
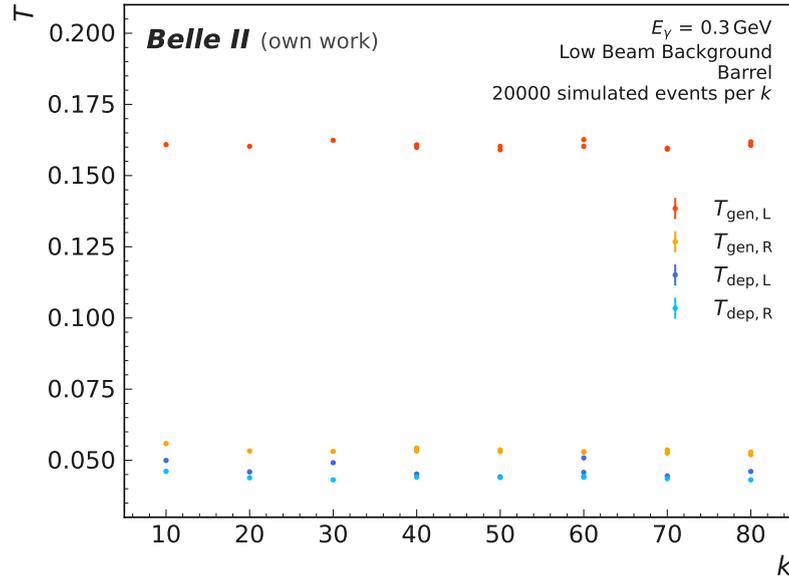
(a) Energy resolution of different k nearest neighbor edge connections.(b) Tail lengths of different k nearest neighbor edge connections.

Figure 7.4.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) for different numbers of edges in the graph. The number of edges is determined from the k nearest neighbor edge connections in Euclidean distance between the crystal nodes. The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5] \text{ GeV}$ and tested on 20 000 barrel events with low beam background and $E_\gamma = 0.3 \text{ GeV}$. The number of edges does not have a visible impact on the performance of the GCN algorithm.

7.3.2. Edge Weights in the Graph

7.3.2.1. Settings

In this toy study, all possible node input features (see section 5.1.1) are used as inputs for the graphs.

Each node is connected to its k nearest neighboring nodes by an edge. Since the reconstructed time t_{rec} is a necessary node input feature, the goal is to investigate whether including the reconstructed time t_{rec} additionally in the edge input features improves the performance. The distance between two nodes i and j is given by the Euclidean distance

$$d'_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 + (t_i^{\text{rec}} - t_j^{\text{rec}})^2} \text{ m}^2 \cdot \mu\text{s}^{-2}, \quad (7.2)$$

with the Cartesian coordinates $x_{i/j}$, $y_{i/j}$, $z_{i/j}$, and the reconstructed time $t_{i/j}^{\text{rec}}$ of the respective crystals. As the reconstructed time in μs has a similar magnitude as the Cartesian coordinates in m , the reconstructed time is added as a fourth dimension in Euclidean space. The edge weight is defined as the inverse of the Euclidean distance as described in equation (5.2).

The GCN algorithm is trained on a data set of 20 000 barrel events with LBB, and then tested on a separate data set of 20 000 barrel events with LBB.

7.3.2.2. Performance Evaluation

Figure 7.5 displays the energy resolution on the deposited energy σ_{dep} and on the generated energy σ_{gen} as well as the corresponding tail lengths $T_{\text{dep/gen}}$ for a test set of 20 000 single photon events with the energy $E_\gamma \in [0.1, 1.5]$ GeV. Interestingly, when the reconstructed time is included in the edge weight, the energy resolution on both deposited and generated energy is worse for all given values of k . This observation is consistent in figure 7.6, which presents the energy resolution on the deposited and generated energy, along with the corresponding tail lengths, for a test set of 20 000 single photon events with the photon energy $E_\gamma = 0.3$ GeV. Furthermore, a dependency of the energy resolution on the number of k nearest neighbors appears when the reconstructed time is included in the edge input features. Especially for $k = 12$, the energy resolution is a lot worse and the tails are significantly longer (see figure 7.6) in comparison to $k \in \{8, 10\}$ with the reconstructed time included in the edge weights.

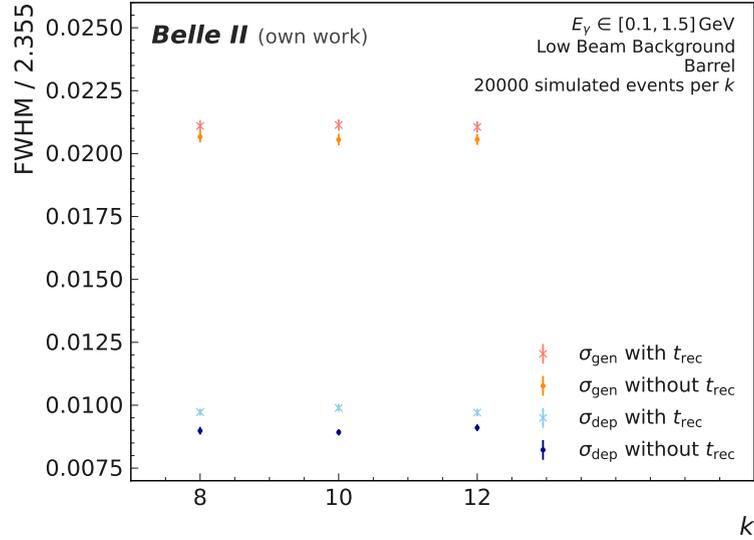
7.3.2.3. Overview and Conclusions

Including the reconstructed time t_{rec} in the edges as a fourth dimension in the Cartesian coordinates does not enhance the performance of the GCN model in comparison to the edges only depending on the Cartesian crystal coordinates.

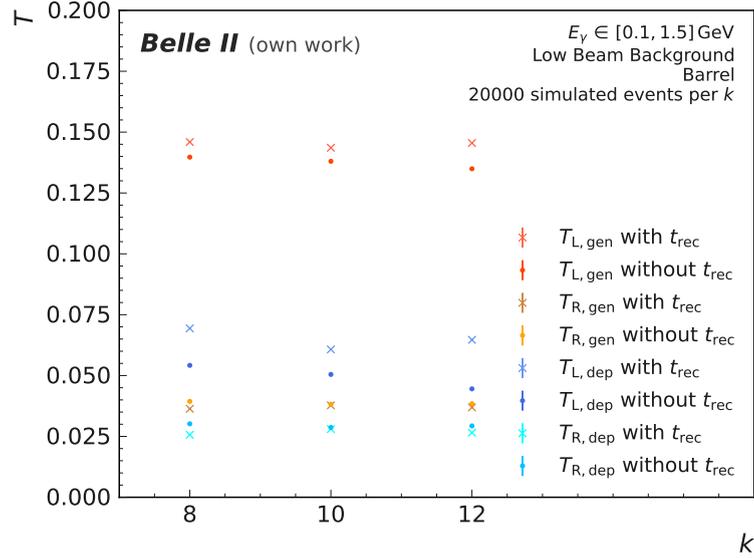
It is not intuitive that the performance for higher k gets worse since more nodes are connected via edges where their corresponding crystals are further away from each other in the ECL. Therefore, it is more likely that their major energy fraction belongs to different types, such as photon signal or beam background. Because crystals with mostly beam background energy tend to have a reconstructed time close to $t_{\text{rec}} = \pm 1 \mu\text{s}$ and crystals with

mostly energy of the photon have a reconstructed time around $t_{\text{rec}} \approx 0 \mu\text{s}$, the reconstructed time should help the algorithm to learn which nodes belong to crystals that are spatial close in the ECL. A reason for the counter-intuitive results could be that the training is very sensitive and therefore unstable. Consequently, the study should be repeated in future studies with training on more events to detect if the result for $k = 12$ is an outlier.

In conclusion, the reconstructed time should not be included in the edge input features as a fourth dimension in the Cartesian coordinates, as its incorporation leads to worse performance of the GCN model compared to when the reconstructed time is excluded.

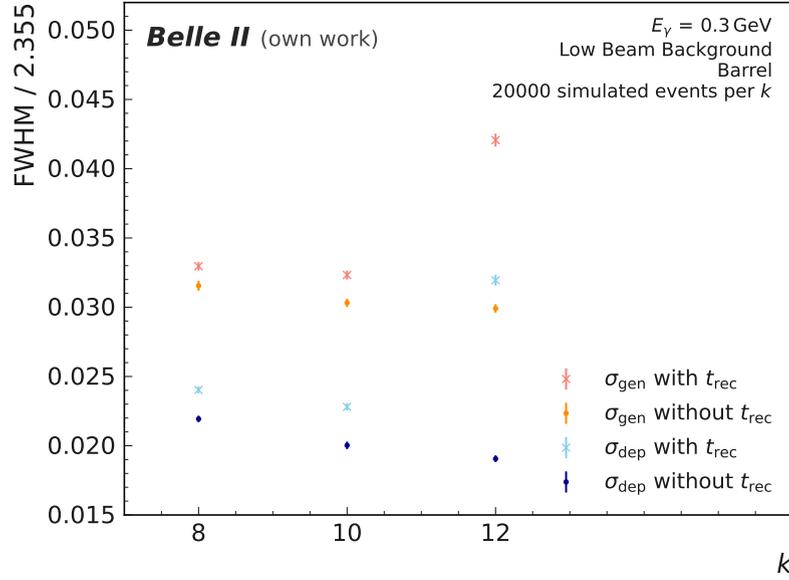


(a) Energy resolution of different k nearest neighbor edge connections with the reconstructed time t_{rec} in the edge input features.

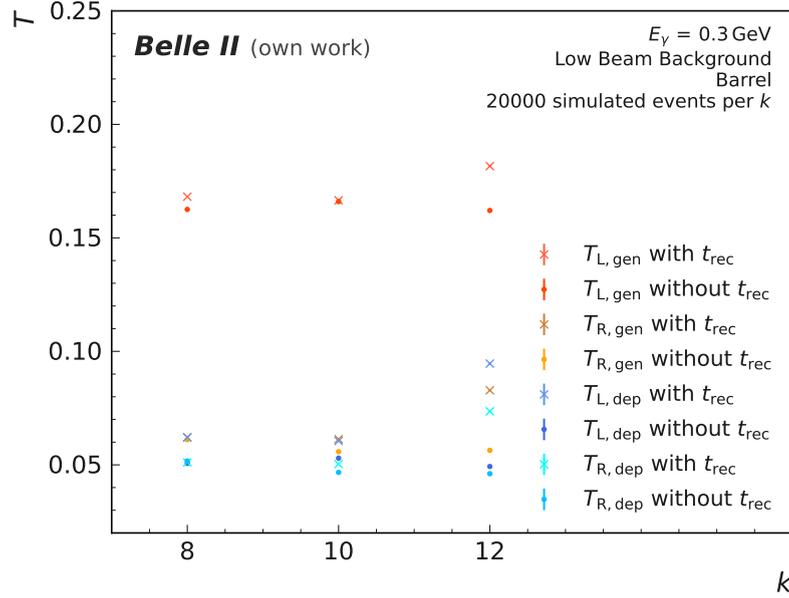


(b) Tail lengths of different k nearest neighbor edge connections with the reconstructed time t_{rec} in the edge input features.

Figure 7.5.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of different k nearest neighbors with the reconstructed time t_{rec} as an edge input feature as a fourth dimension in the Cartesian coordinates. The GCN model is trained on 20000 barrel detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5] \text{ GeV}$ and tested on 20000 barrel events with low beam background and $E_\gamma \in [0.1, 1.5] \text{ GeV}$. The performance of the GCN is worse if the reconstructed time is included in the edge weights as a fourth dimension in Euclidean distance.



(a) Energy resolution of different k nearest neighbor edge connections with the reconstructed time t_{rec} in the edge input features.



(b) Tail lengths of different k nearest neighbor edge connections with the reconstructed time t_{rec} in the edge input features.

Figure 7.6.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of different k nearest neighbors with the reconstructed time t_{rec} as an edge input feature as a fourth dimension in the Cartesian coordinates. The GCN model is trained on 20 000 barrel detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with low beam background and $E_\gamma = 0.3$ GeV. The performance of the GCN is worse if the reconstructed time is included in the edge weights as a fourth dimension in Euclidean distance and the model performance is dependent on the number of edges.

7.4. Energy dependence

This section focuses on analyzing the energy dependence of the GCN algorithm, considering both training and testing on LBB and HBB events. Assessing the energy dependence is an essential aspect of this study as it offers insights into how the algorithm performs across various energy ranges. Furthermore, a comparison is made between the performance of the GCN algorithm and that of the GravNet and basf2 algorithms. This comparative analysis provides valuable information regarding the relative strengths and weaknesses of the different algorithms.

7.4.1. Settings

In this toy study, all possible node input features (see section 5.1.1) are used as inputs for the graph and each node is connected to every other node in the graph. The edge weight is determined by equation (5.1).

The training of the GCN model is performed on two million full detector events with either LBB or HBB with the photon energies ranging in $E_\gamma \in [0.1, 1.5]$ GeV. Afterwards, the trained models are tested on 20 000 barrel events with LBB or respectively HBB for each fixed photon energy E_γ ranging from 0.1 GeV to 1.5 GeV in steps of 0.1 GeV and from 1.5 GeV to 2.5 GeV in steps of 0.25 GeV as well as the photon energies 3 GeV, 4 GeV, and 5 GeV. The energy range extends beyond the photon energies used to train the GCN and also the GravNet model, allowing for an evaluation of the model's ability to generalize to different photon energies.

7.4.2. Performance Evaluation

Figure 7.7 illustrates the energy resolution of the deposited energy σ_{dep} as well as the corresponding left and right tail lengths $T_{\text{L/R,dep}}$. Resolution results of the GravNet and the basf2 models are only given for the photon energy $E_\gamma = 0.3$ GeV. Notably, for a photon energy of $E_\gamma = 0.3$ GeV, the GNN algorithms outperform the basf2 model by 27%. Comparing the GCN model to the GravNet model, the GCN model demonstrates a slightly better performance by 4%, indicating that the fundamental clustering capability of the GCN model is similar to or even better than the GravNet model.

In figure 7.7b, the tail lengths of the photon energies up to $E_\gamma = 2$ GeV follow the same function as the energy resolution. However, for higher energies, the left tail becomes significantly longer, suggesting an underestimation of the photon energy by the GCN algorithm. This behavior arises from the fact that the GCN model is trained specifically for the photon energy range $E_\gamma \in [0.1, 1.5]$ GeV.

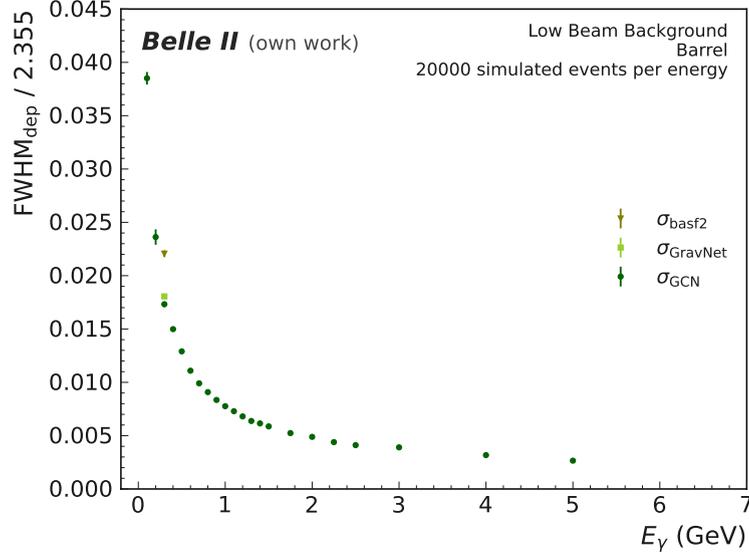
For both LBB and HBB, the GNN algorithms demonstrate significantly better performance for the energy reconstruction of the generated energy in the energy regime $E_\gamma \in [0.1, 2.5]$ GeV compared to the basf2 model with an improvement of up to 15% for $E_\gamma = 0.1$ GeV. This is demonstrated in figure 7.8 for LBB events and in figure 7.9 for HBB events. The energy dependence of the GCN model follows the expected function described in equation (6.18) since the GravNet and the basf2 model can be described by this function [1] and all algorithms have a very similar dependency on the photon energy. The GCN model is only

slightly worse performing than the GravNet model by 4% for energies up to 2.5 GeV in the case of the LBB. For the HBB study this is similar but only results for energies up to $E_\gamma = 1.5$ GeV are determined since the distribution is difficult to be described by a DCB for higher energies. The GCN model outperforms the basf2 model on average by 20% and is around 5% worse in performance compared to the GravNet model. For higher energies $E_\gamma > 2.5$ GeV, the GCN model trained on LBB events performs better with an improvement of up to 16% for $E_\gamma = 5$ GeV than the GravNet model but worse than the basf2 algorithm.

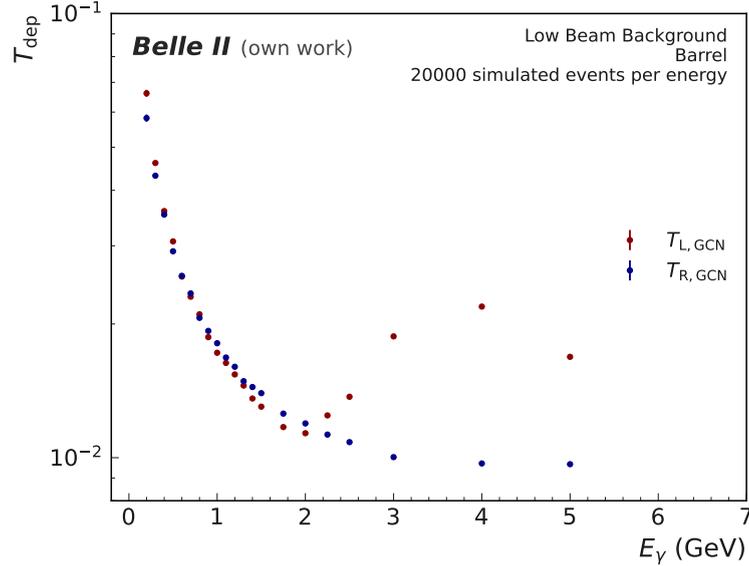
The tails of the GCN and the GravNet model for each photon energy are very similar in length as shown in figure 7.8b for LBB events and in figure 7.9b for HBB events. The GravNet model has slightly shorter tails than the GCN for the training's range $E_\gamma \in [0.1, 1.5]$ GeV.

7.4.3. Overview and Conclusions

In both cases of LBB and HBB, the GNN algorithms show significantly better performance than the basf2 model for photon energies up to $E_\gamma = 2.5$ GeV. The GCN model has a slightly worse performance for photons in this energy range than the GravNet model. This could arise from the fact, that the GravNet model has optimized hyperparameters and therefore a more stable architecture in comparison to the GCN that is not optimized in any way so far. As part of future work, the performance of the GCN model with optimized hyperparameters could be analyzed. However, for higher energies, the GCN model trained on LBB events performs better than the GravNet model but worse than the basf2 algorithm. One of the reasons is that photons with $E_\gamma \in [0.1, 2.5]$ GeV often deposit their energy in only a few crystals, which is a characteristic that the GNN models learn already with photons that have an energy of $E_\gamma = 0.1$ GeV. For photons with $E_\gamma > 2.5$ GeV, the number of crystals in a cluster increases, which exceeds the algorithm's training capacity and results in reduced performance.

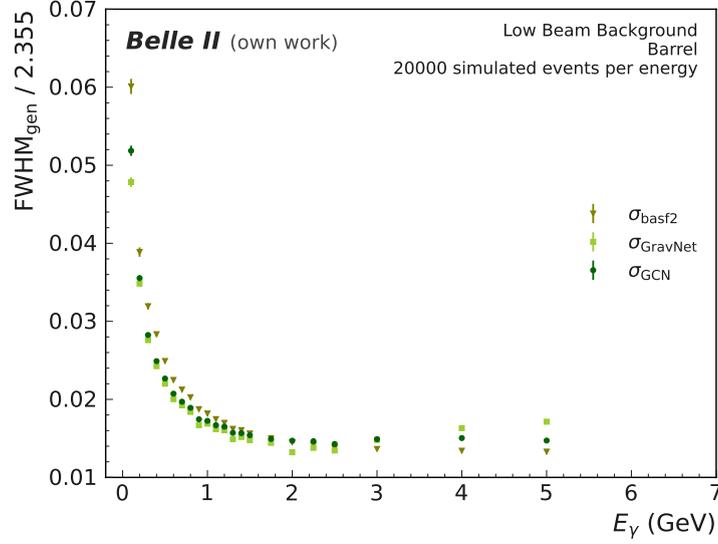


(a) Energy resolution of the deposited energy in dependency of the photon energy E_γ for the three different models GCN, GravNet, and basf2.

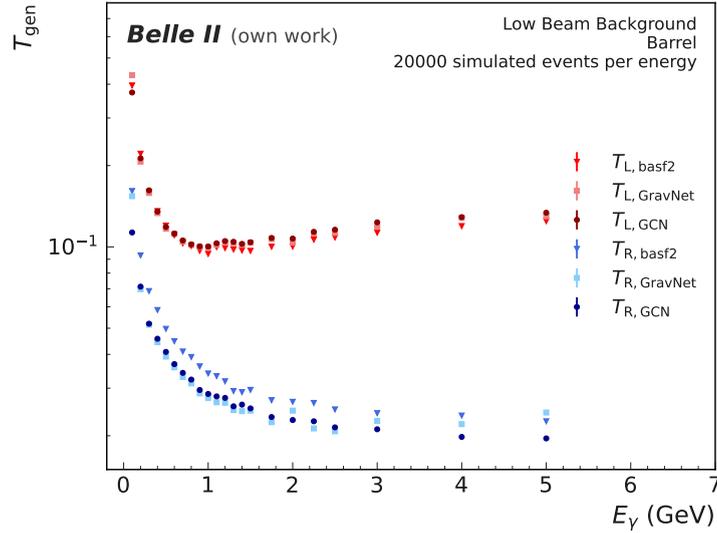


(b) Left and right tail lengths for the deposited energy in dependency of the photon energy E_γ for the GCN model.

Figure 7.7.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of the deposited energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20000 barrel events with low beam background per photon energy E_γ . The energy resolution of the GCN model is compared to the basf2 and the GravNet algorithm for $E_\gamma = 0.3$ GeV. The GCN model performs better than the basf2 algorithm.

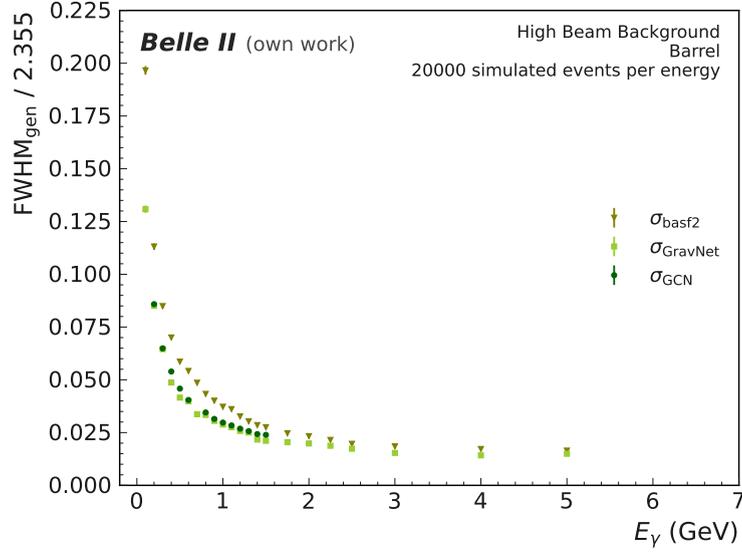


(a) Energy resolution of the generated energy in dependency of the photon energy E_γ for the three different models GCN, GravNet, and basf2.

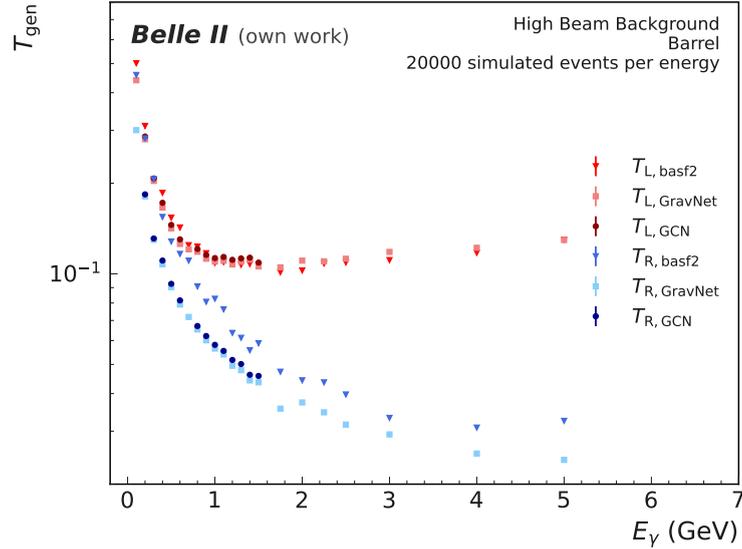


(b) Left and right tail lengths for the generated energy in dependency of the photon energy E_γ for the three different models GCN, GravNet, and basf2.

Figure 7.8.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of the generated energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with low beam background per photon energy E_γ . The energy resolution and the tail lengths are compared to the basf2 and the GravNet algorithm. The GCN model performs better than the basf2 algorithm for $E_\gamma \leq 2$ GeV and generalizes better than the GravNet model for $E_\gamma > 3$ GeV.



(a) Energy resolution of the generated energy in dependency of the photon energy E_γ for the three different models GCN, GravNet, and basf2.



(b) Left and right tail lengths for the generated energy in dependency of the photon energy E_γ for the three different models GCN, GravNet, and basf2.

Figure 7.9.: Energy resolution $\text{FWHM}/2.355$ (a) and tail lengths T (b) of the generated energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with high beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with high beam background per photon energy E_γ . The energy resolution and tail lengths are compared to the basf2 and the GravNet algorithm. The GCN model performs better than the basf2 algorithm for $E_\gamma \leq 1.5$ GeV.

8. Summary and Outlook

This work introduces and analyzes the energy reconstruction of photons in the Belle II ECL using statically build graphs as an input for GNNs. The ML based GCN model employs fuzzy clustering for the photon energy reconstruction. The GCN architecture includes stacked graph convolutional layers that use message passing between nodes in the graph. As previous studies have shown, the advantage of using GNNs due to varying input sizes and the models' ability not to rely on a regular calorimeter geometry can significantly improve the energy resolution in the ECL. The GravNet model employs message passing between nodes in a dynamic graph-building process in contrast to the GCN algorithm that uses static graphs for message passing. MC data is used for training and evaluating the model's performance.

The Belle II Analysis Software Framework is currently used for photon reconstruction. The reconstruction process is referred to as the basf2 model and together with the GravNet model, these algorithms serve as a comparison for the performance of the GCN model. The utilized metrics are introduced in chapter 6 for the evaluation of the performance for different graphs as an input of the GCN model and to ensure a fair comparison of the three algorithms.

Toy studies for different graph structures as input of the GCN model are analyzed by evaluating the performance of the energy reconstruction of single photon events. The graphs are studied with different node input features, different numbers of edges, and different edge weights. The relevant node input features for the GCN algorithm include the reconstructed time t_{rec} , along with the basic node input features such as the reconstructed energy E_{rec} , the global crystal coordinates θ , ϕ_{sin} , ϕ_{cos} , and the crystal mass m . Other node input features like the hadron intensity HI, the local maximum indicator LM, and the local coordinates θ' and ϕ' have minimal impact on the model's performance. The number of edges, represented by the parameter k for the k nearest neighbor connections between nodes, does not significantly affect the performance of the GCN algorithm above a certain number of edges. Even with all nodes connected, the performance remains comparable to having a limited number of k nearest neighbors. Including the reconstructed time t_{rec} in the edge input features as a fourth Cartesian coordinate does not improve the GCN model's performance. Instead, it leads to worse results compared to using only Cartesian crystal coordinates for the edge weights.

One important thing to note is that fluctuations during training studied in chapter 7 are significant and require further investigation. Therefore, the training process of the GCN

model should be further optimized regarding training hyperparameters. The uncertainty on the FWHM and the tail lengths depending on the training should be determined from statistics. Additionally, training on more events is expected to lead to more stable training results.

Furthermore, the energy dependence of the GCN model is compared to the GravNet and basf2 algorithms. The GNN algorithms including the GCN and the GravNet model demonstrate significantly better performance than the basf2 model for photons with an energy up to $E_\gamma = 2.5$ GeV. The GCN model performs slightly worse than the GravNet model for low photon energies. This applies to both LBB and HBB conditions. The reason could be that the GravNet model is optimized regarding its hyperparameters in comparison to the GCN that is not optimized in any way so far. The optimization of the GCN model could be done to see if the performance of the GNN algorithms is equal if each is optimized individually. This could include optimizing the existing GCN structure and extending the used hyperparameters, offering possibilities for fine-tuning its performance and addressing specific requirements. One additional area worth exploring further is the specific architecture of the GCN model. There are alternative approaches to implementing GNNs and performing message passing that can be considered instead of the GCN layer. Investigating these aspects can lead to further enhancements and refinements in the application of the GCN architecture for energy reconstruction tasks.

However, for higher energies, the GCN model performs better than the GravNet model but worse than the basf2 algorithm. This is due to the GNN models' limited capacity to handle clusters with a large number of crystals for higher-energy photons. Since the models are only trained on energies up to 1.5 GeV the evaluation of higher energies shows the ability of the GCN model to generalize better than the GravNet model.

In the next step, the GCN model could be studied and optimized for the energy reconstruction of overlapping photon clusters or other MC generated particles such as pions or electrons in a similar way as presented in this work. Furthermore, a validation on data has to be done to ensure the correct behavior of the network. Although there are still many challenges until the GCN will predict the photon energies in the Belle II ECL the model is promising since the input features available in real data could allow the model to distinguish various cluster signatures effectively.

In conclusion, the GCN algorithm demonstrates its effectiveness and adaptability as an algorithm for clustering energy depositions, making it a promising choice as an alternative to the GravNet model for a wide range of current and future applications. Especially the statically build graphs and the lower amount of free model parameters might make the GCN algorithm more useful for example in online applications, such as the Level 1 trigger at Belle II.

Bibliography

- [1] F. Wemmer, “Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks,” Master’s thesis, Karlsruhe Institute of Technology (KIT), 11, 2022. <https://publish.etp.kit.edu/record/22142>.
- [2] “Belle II Analysis Software Framework (basf2).” <https://software.belle2.org>.
- [3] S. R. Qasim, J. Kieseler, Y. Iiyama, and M. Pierini, “Learning representations of irregular particle-detector geometry with distance-weighted graph networks,” *The European Physical Journal C* **79** no. 7, (Jul, 2019) . <https://doi.org/10.1140%2Fepjc%2Fs10052-019-7113-9>.
- [4] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” 2017. <https://arxiv.org/abs/1609.02907>.
- [5] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using networkx,” in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, eds., pp. 11 – 15. Pasadena, CA USA, 2008. https://conference.scipy.org/proceedings/scipy2008/paper_2/.
- [6] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering* **9** no. 3, (2007) 90–95.
- [7] B. Povh, K. Rith, C. Scholz, F. Zersche, and W. Rodejohann, *Particles and Nuclei: An Introduction to the Physical Concepts*. Graduate Texts in Physics. Springer, 1995.
- [8] E. Kou *et al.*, “The Belle II Physics Book,” *Progress of Theoretical and Experimental Physics* **2020** no. 2, (Feb, 2020) . <https://doi.org/10.1093/ptep/ptaa008>. 029201.
- [9] T. Abe, *et al.*, “Belle II Technical Design Report,” 2010. <https://arxiv.org/abs/1011.0352>.
- [10] **SuperKEKB**, K. Akai, K. Furukawa, and H. Koiso, “SuperKEKB Collider,” *Nucl. Instrum. Meth. A* **907** (2018) 188–199, [arXiv:1809.01958](https://arxiv.org/abs/1809.01958) [physics.acc-ph].
- [11] D. Sanders, Aug, 2019. <http://www.phy.olemiss.edu/HEP/belle2/index.html>.
- [12] H. Ikeda, “Development of the CsI(Tl) Calorimeter for the Measurement of CP Violation at KEK B-Factory,” *Inst. Ital. Attuari, Giorn.* **4** (11, 2014) 83–91. <https://opac2.lib.nara-wu.ac.jp/webopac/TD00003756>.
- [13] V. Aulchenko, A. Bobrov, T. Ferber, A. Kuzmin, K. Miyabayshi, G. de Nardo, V. Shebalin, A. Sibidanov, Y. Usov, and V. Zhulanov, “Time and energy

- reconstruction at the electromagnetic calorimeter of the Belle-II detector,” *Journal of Instrumentation* **12** (2017) .
<https://iopscience.iop.org/article/10.1088/1748-0221/12/08/C08001>.
- [14] S. Longo, J. Roney, C. Cecchi, and et al., “CsI(Tl) Pulse Shape Discrimination With the Belle II Electromagnetic Calorimeter as a Novel Method to Improve Particle Identification at Electron-Positron Colliders,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **982** (2020) . 164562.
- [15] S. Sugihara and H. Aihara, *Design Study of Belle II Interaction Region*. PhD thesis, Tokyo, University of Tokyo, Tokyo, 2011. <https://docs.belle2.org/record/217>. Presented on 03 02 2011.
- [16] Ferber, T, “Electromagnetic Calorimeter Reconstruction in Belle II.”
URL: <https://indico.cern.ch/event/708041/contributions/3269704>, 2019.
- [17] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” *AI Open* **1** (2020) 57–81. <https://doi.org/10.1016/j.aiopen.2021.01.001>.
- [18] D. J. Bora and D. A. K. Gupta, “A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm,” *International Journal of Computer Trends and Technology* **10** no. 2, (Apr, 2014) 108–113.
<https://doi.org/10.14445/2F22312803%2Fijctt-v10p119>.
- [19] Z. Zhang, P. Cui, and W. Zhu, “Deep Learning on Graphs: A Survey,” 2020.
<https://arxiv.org/abs/1812.04202>.
- [20] F. Wemmer, *et al.*, “Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks,” 2023. <https://arxiv.org/abs/2306.04179>.
- [21] Z. Liptak, *et al.*, “Measurements of beam backgrounds in SuperKEKB Phase 2,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **1040** (Oct, 2022) 167168.
<https://doi.org/10.1016%2Fj.nima.2022.167168>.
- [22] F. Pedregosa, *et al.*, “Scikit-learn: Machine learning in python,” *J. Mach. Learn. Res.* **12** no. null, (Nov, 2011) 2825–2830.
<https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>.
- [23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” 2016.
<https://arxiv.org/abs/1511.07289>.
- [24] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” 2019.
<https://arxiv.org/abs/1803.08375>.
- [25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” 2019.
<https://arxiv.org/abs/1907.10902>.

- [26] A. Paszke, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” 2019. <https://arxiv.org/abs/1912.01703>.
- [27] M. Fey and J. E. Lenssen, “Fast graph representation learning with pytorch geometric,” 2019. <https://arxiv.org/abs/1903.02428>.
- [28] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 2017. <https://arxiv.org/abs/1412.6980>.
- [29] J. Eschle, A. Puig Navarro, R. Silva Coutinho, and N. Serra, “zfit: Scalable pythonic fitting,” *SoftwareX* **11** (2020) 100508. <https://www.sciencedirect.com/science/article/pii/S2352711019303851>.
- [30] E. O. Lebigot, “Uncertainties: a Python package for calculations with uncertainties.” <http://pythonhosted.org/uncertainties/>.

A. Appendix

A.1. Bad Double-Sided Crystal Ball Fit

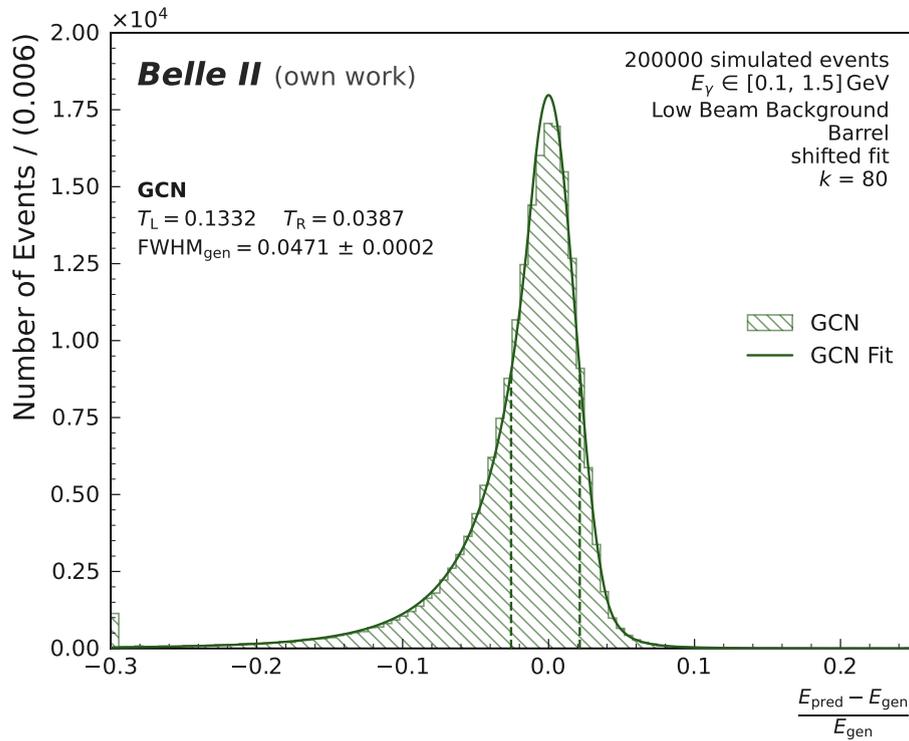


Figure A.1.: Example of the distribution of the relative reconstruction error on the generated energy η_{gen} . The distribution can not be described by a double-sided crystal ball function and consequently, the FWHM and tail lengths are not correctly determined.

A.2. Performance Evaluation of the Deposited Energy for the Photon Energy 0.3 GeV for Different Node Input Features

For the photon energy $E_\gamma = 0.3$ GeV, most DCB fits fail when applied to the distributions of the reconstruction errors on the deposited energy η_{dep} . An example of such distributions is given in figure A.3.

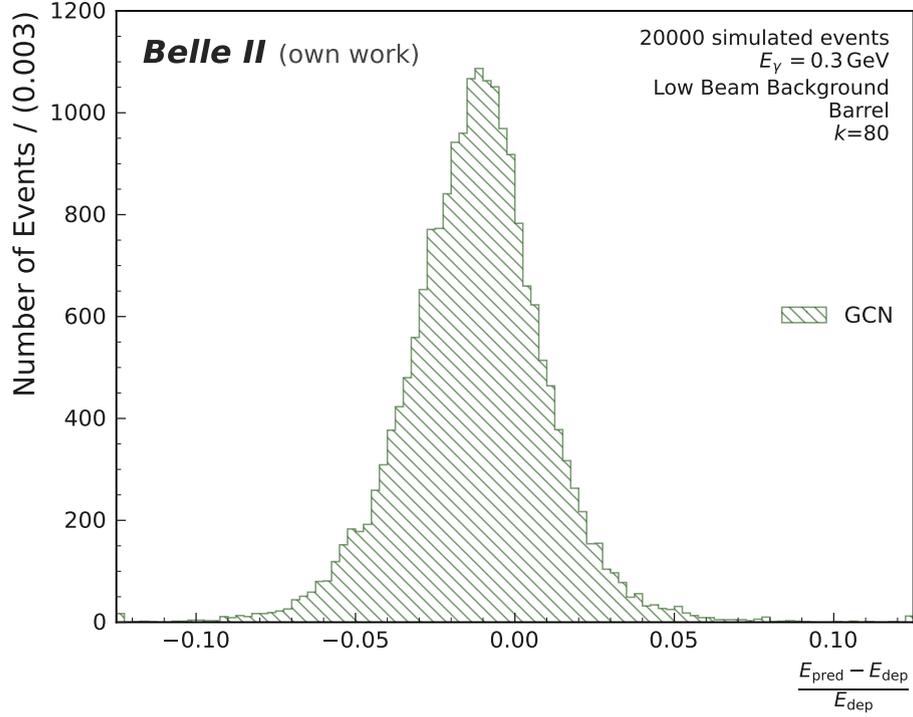
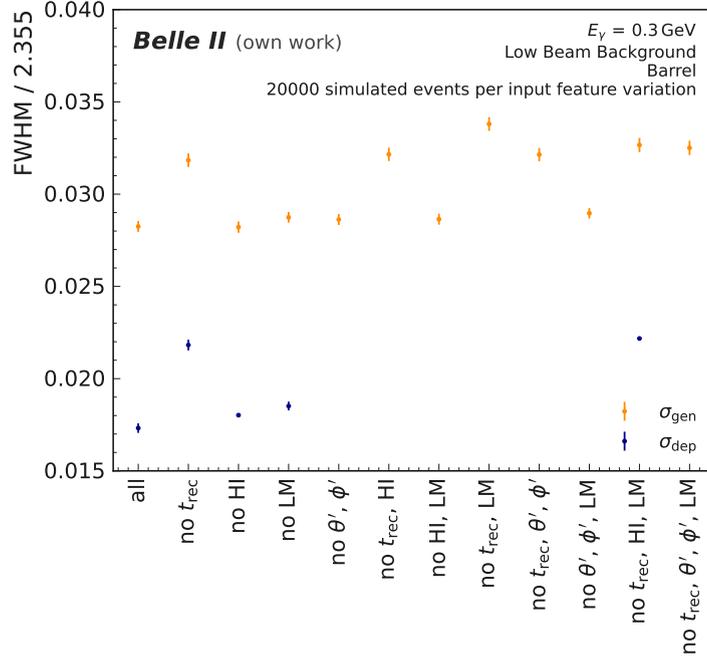
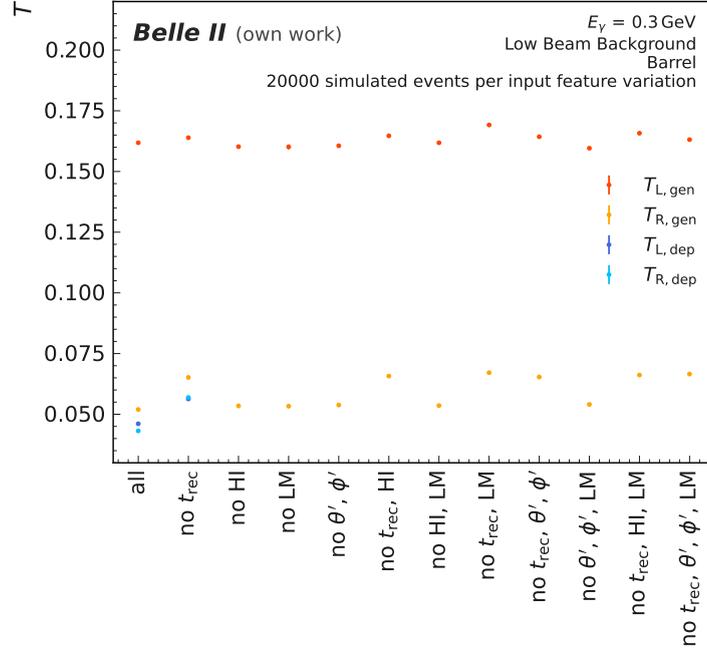


Figure A.3.: Example of the distribution of the relative reconstruction error on the deposited energy η_{dep} . The fitting procedure fails for this type of distribution.

A.2. Performance Evaluation of the Deposited Energy for the Photon Energy 0.3 GeV for Different Node Input Features 63



(a) Energy resolution of different node input feature combinations.



(b) Tail lengths of different node input feature combinations.

Figure A.2.: Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different node input feature combinations of the reconstructed time t_{rec} , the hadron intensity HI, the local maximum indicator LM and the local coordinates θ', ϕ' . The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5] \text{ GeV}$ and tested on 200 000 barrel low beam background events with $E_\gamma = 0.3 \text{ GeV}$. Without the reconstructed time the energy resolution (a) is worse and both left and right tails (b) are longer compared to other node input feature combinations that include the reconstructed time.

B. List of Figures

2.1.	A schematic representation of the SuperKEKB accelerator where the Belle II detector is located at the experimental hall Tsukuba [10].	4
2.2.	A schematic overview of the Belle II detector with its sub-detector parts that are centered around the interaction point [11].	5
2.3.	A schematic representation of the Belle calorimeter with the barrel region, the forward, and the backward endcap. The interaction point is in its center and marked with interaction point (IP) [12].	7
2.4.	Background energy depositions in the electromagnetic calorimeter for low beam background and high beam background [16].	9
2.5.	Representation of the basf2 clustering algorithm in the Belle II electromagnetic calorimeter [16].	11
3.1.	Pictorial representation of the data processing in the GravNet layer. The plots display the learning of the spatial and feature space (a), the connection of nodes (b), the message passing (c), and the concatenation to one output (d) for one node [3].	15
3.2.	Schematic representation of a multi-layer graph convolutional network [4].	16
4.1.	Representation of a typical low beam background event with the crystals in the region of interest. Displayed are the Monte Carlo truth energy deposition, the reconstructed energy E_{rec} , the reconstructed time t_{rec} and the hadron intensity as part of the pulse shape discrimination information [1].	19
4.2.	Representation of a typical high beam background event with the crystals in the region of interest. Displayed are the Monte Carlo truth energy deposition, the reconstructed energy E_{rec} , the reconstructed time t_{rec} and the hadron intensity as part of the pulse shape discrimination information [1].	20
5.1.	A schematic view of the graph neural network architectures [20].	23
5.2.	Pictorial representations of a single photon event graph where each node is connected by an edge to all other nodes (a) and where each node is connected to its $k = 4$ nearest neighbors (b).	25
5.3.	Loss and validation loss during training in dependency of the training epoch.	27

6.1.	Distribution of the relative reconstruction errors $f(\eta_{\text{dep/gen}})$ of an example test data set of the graph convolutional network model trained on low beam background events.	32
6.2.	Example of a double-sided crystal ball function fit for the distribution of the relative reconstruction errors on the deposited energy $f(\eta_{\text{dep}})$ of an example test data set of the GCN model trained on low beam background events. . .	34
6.3.	Example of the energy resolution dependence $\sigma_{\text{gen}}(E_\gamma)$ of the relative reconstruction errors on the deposited energy $f(\eta_{\text{dep}})$ of example test data sets of the GCN model trained on low beam background events.	35
7.1.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different node input feature combinations. The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel low beam background events with $E_\gamma \in [0.1, 1.5]$ GeV.	40
7.2.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different node input feature combinations. The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel low beam background events with $E_\gamma = 0.3$ GeV.	41
7.3.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) for different numbers of edges in the graph. The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel events with low beam background and $E_\gamma \in [0.1, 1.5]$ GeV.	44
7.4.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) for different numbers of edges in the graph. The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel events with low beam background and $E_\gamma = 0.3$ GeV.	45
7.5.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different k nearest neighbors with the reconstructed time t_{rec} as an edge input feature as a fourth dimension in the Cartesian coordinates. The GCN model is trained on 20 000 barrel detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel events with low beam background and $E_\gamma \in [0.1, 1.5]$ GeV.	48
7.6.	Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different k nearest neighbors with the reconstructed time t_{rec} as an edge input feature. The GCN model is trained on 20 000 barrel detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 200 000 barrel events with low beam background and $E_\gamma = 0.3$ GeV.	49

- 7.7. Energy resolution FWHM/2.355 (a) and tail lengths T (b) of the deposited energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with low beam background per photon energy E_γ . The energy resolution is compared to the basf2 and the GravNet algorithm for $E_\gamma = 0.3$ GeV. 52
- 7.8. Energy resolution FWHM/2.355 (a) and tail lengths T (b) of the generated energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with low beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with low beam background per photon energy E_γ . The energy resolution is compared to the basf2 and the GravNet algorithm. 53
- 7.9. Energy resolution FWHM/2.355 (a) and tail lengths T (b) of the generated energy in dependency of the photon energy E_γ . The GCN model is trained on two million full detector events with high beam background and the photon energy $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel events with high beam background per photon energy E_γ . The energy resolution is compared to the basf2 and the GravNet algorithm. 54
- A.1. Example of the distribution of the relative reconstruction error on the generated energy η_{gen} . The distribution can not be described by a double-sided crystal ball function and consequently, the FWHM and tail lengths are not correctly determined. 61
- A.3. Example of the distribution of the relative reconstruction error on the deposited energy η_{dep} . The fitting procedure fails for this type of distribution. 62
- A.2. Energy resolution FWHM/2.355 (a) and tail lengths T (b) of different node input feature combinations. The GCN model is trained on two million full detector low beam background events with the photon energies $E_\gamma \in [0.1, 1.5]$ GeV and tested on 20 000 barrel low beam background events with $E_\gamma = 0.3$ GeV. 63