Karlsruhe Institute of Technology

ETP-KA/2025-7

A Real-Time Graph Neural Network Trigger Algorithm for the Belle II Electromagnetic Calorimeter

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc. Isabel Haide

aus Stuttgart

Tag der mündlichen Prüfung:	11. Juli 2025
Referent:	Prof. Dr. Torben Ferber
Korreferent:	Prof. Dr. Markus Klute

Disclaimer

Algorithm development and data analyses in high-energy physics such as the work presented in this doctoral thesis are a collaborative effort. The SuperKEKB particle accelerator which provides the particle beams essential for all studies at Belle II was built and is operated and maintained by the SuperKEKB accelerator group. The Belle II detector was built and is maintained and operated by the Belle II collaboration. The Belle II collaboration also creates the centrally provided simulated and recorded datasets and maintains the computing infrastructure necessary to process them. The software environment necessary for studies with Belle II data plays an important role and was created and is maintained by the collaboration. I have been a part of the Belle II collaboration since 2020 and performed all studies detailed in this thesis except for the following:

- The implementation of the GravNet layer in `QKeras` described in Section 6.2.1, which was done by Marc Neu;
- The hardware implementation of the preprocessing, the GNN-ETM network, the condensation point selection algorithm and the postprocessing of the GNN-ETM module, including the development of the C Simulation, described in Section 6.5. This was done by Marc Neu and Valdrin Dajaku.

This thesis employs the use of Artificial Intelligence (AI) tools to help with grammatical or stylistic improvement of text, and program code creation.

Grammarly[*] is used throughout the thesis for spell and grammar checks, as well as for paraphrasing individual, selected sentences to improve clarity and precision in academic writing. I have approved all suggested changes.

ChatGPT[†] is used to aid the development of C++ and Python code, in particular code restructuring and optimisation that do not constitute the core scientific work of this thesis. I have approved and tested all suggestions to provide robust and reliable results.

[*] Grammarly: An AI writing assistant. See https://app.grammarly.com/ (Access Date: 2025-06-03).

[†] ChatGPT: A virtual AI assistant based on large language models. See https://openai.com/chatgpt/ (Access Date: 2025-06-03).

Contents

1 Introduction **1**

2 SuperKEKB and the Belle II Experiment **3**

2.1 The SuperKEKB Accelerator . . . 3

2.2 The Belle II Detector . . . 4

2.3 The Electromagnetic Calorimeter . . . 8

3 The Existing First Level Trigger Setup at Belle II **11**

3.1 The Level 1 Trigger . . . 11

3.2 The Calorimeter Trigger Setup at Belle II . . . 15

3.2.1 ICN and Clustering Logic . . . 18

3.2.2 Trigger Window Decision and Trigger Timing . . . 20

3.2.3 Trigger Bit Determination on ICN-ETM . . . 22

3.2.4 Output Data . . . 22

4 Monte Carlo Data Simulation and Preparation for Training **25**

4.1 Simulation and Reconstruction . . . 25

4.1.1 ECL Trigger Simulation . . . 26

4.1.2 Reconstruction of Offline ECL Clusters . . . 27

4.1.3 Beam Backgrounds at Belle II . . . 29

4.2 Training Inputs and Targets . . . 37

4.2.1 Inputs . . . 37

4.2.2 Training Targets . . . 38

4.3 Training Datasets . . . 44

4.3.1 Category-One Sample . . . 45

4.3.2 Category-Two Sample . . . 48

4.4 Evaluation Datasets . . . 49

5 Metrics **52**

5.1 Matching to Offline ECL Clusters . . . 52

5.2 Evaluation Metrics . . . 55

5.2.1 Efficiency and Purity . . . 55

5.2.2 Energy and Position Resolution . . . 57
5.2.3 Signal Retention and Background Rejection . . . 58
5.2.4 Trigger Bits and Trigger Rate . . . 58

6 Network Design 60
6.1 Software/Hardware Co-Design . . . 60
6.2 Network Architecture . . . 61
6.2.1 Graph Neural Networks and the GravNet layer . . . 61
6.2.2 The Object Condensation Algorithm . . . 64
6.2.3 Overall Network Design . . . 66
6.2.4 Basic Training Setup . . . 71
6.3 Quantization Optimization . . . 71
6.3.1 Input and Output Quantization . . . 74
6.3.2 Quantization of Weights, Biases and Activation Functions . . . 77
6.3.3 Post-Training Quantizations . . . 80
6.4 Architecture and Input Optimization . . . 85
6.4.1 Low-Magnitude Pruning . . . 86
6.4.2 Input Scaling and Distance Optimization . . . 87
6.4.3 Energy Prediction Optimization and Network Stabilization . . . 89
6.4.4 Hyperparameter Optimization . . . 93
6.5 Implementation on Hardware . . . 94
6.5.1 Setup and Requirements for Integration to the Level 1 Trigger System 94
6.5.2 Implementation Design and Results . . . 96
6.5.3 Evaluation Setup for Hardware Performance . . . 100
6.5.4 Data Taken with GNN-ETM . . . 103

7 Evaluation on Technical Datasets 107
7.1 Comparison of Different Trainings . . . 107
7.2 Signal/Background Classification Performance . . . 112
7.2.1 Category-One Dataset . . . 115
7.2.2 Evaluation of Trigger Rate on Pure Beam Background . . . 117
7.3 Performance on Overlapping Clusters . . . 119

8 Evaluation on Physics Processes 123
8.1 Data Evaluation Setup . . . 123
8.1.1 Selection of ECL Clusters . . . 123
8.1.2 Data Samples . . . 125
8.2 Analysis of the GNN-ETM performance for $e^+e^- \rightarrow e^+e^-(\gamma)$ events . . . 126
8.2.1 Selection Procedure and High-Level Data/MC Agreement . . . 127
8.2.2 Trigger Data Sample Selection . . . 134
8.2.3 Trigger Cluster Efficiencies and Resolutions . . . 135

8.2.4 Trigger Bit Efficiency 147
8.3 Analysis of the GNN-ETM performance for $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$ events 151
8.3.1 Selection Procedure and General Data/MC Agreement 151
8.3.2 Trigger Data Sample Selection 157
8.3.3 Trigger Cluster Efficiencies and Resolutions 158
8.4 Signal Classifier Performance on Data 171

9 Conclusion 176

Bibliography 181

Appendix A Angle ID Mapping 191

Appendix B Input quantizations 192

Appendix C Resolutions of five Final Configuration Trainings 194

Appendix D Resolution Distributions and Trigger Efficiencies 207
D.1 GNN-ETM results for $e^+e^- \rightarrow e^+e^-(\gamma)$ 207
D.1.1 Energy and Position Resolutions 207
D.2 GNN-ETM results for $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$ 225
D.2.1 Energy and Position Resolutions 225
D.3 Trigger Bit Efficiency for Random Triggered Events 243

Glossary 247

Acronyms 248

Chapter 1

Introduction

The Belle II experiment has a unique position in the measurement of the Standard Model of particle physics (SM) and deviations from it, as it is sensitive to heavy new physics appearing in loops and feebly coupled light new physics [1] due to its very high luminosity provided by the SuperKEKB electron-positron collider. With its rather clean environment and precisely known Center-of-Mass (CM) energy, Belle II is ideal for probing rare decays that might show discrepancies to SM predictions. Measurements of rare processes such as $\mathrm{B}^+ \to \mathrm{K}^+ \nu\bar{\nu}$ [2] are sensitive to a wide array of physics beyond the SM phenomena, such as leptoquarks [3], axions [4], or dark-sector mediators [5].

With a bunch crossing rate of 250 MHz at Belle II, storing all detector hits at each bunch crossing, while leaving the filtering of interesting events to later stages in analyses, is unfeasible due to the limitations of both bandwidth and storage. A simple solution is the storage of each bunch crossing if the overall number of detector hits or the amount of deposited energy in different subdetector systems crosses a threshold. This is not possible in Belle II, as the high instantaneous luminosity of SuperKEKB comes at a cost: As most bunch crossings do not result in electron-positron collisions, the detector response is dominated by beam-induced background, originating from particles of the beam interacting with material or other particles outside the interaction point (IP). Higher beam background levels call for more sophisticated filtering algorithms to avoid the storage of events without collision data. While SuperKEKB has reached new world records with an instantaneous luminosity of $5.1 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ [6] and has seen already unexpectedly high beam-induced backgrounds, the increase to the design luminosity by more than yet another order of magnitude will magnify this problem.

The event filtering at Belle II employs a two-step filtering procedure, which has been used at many High Energy Physics (HEP) experiments: a hardware-based Level 1 Trigger (L1 trigger) system [7] on Field-Programmable Gate Arrays (FPGAs), which reconstructs the event with hard latency constraints and high throughput, and a subsequent software-based High Level Trigger (HLT) [8]. At Belle II, the L1 trigger system has a latency constraint of 5 µs and a throughput requirement of 8 up to 32 MHz, which is challenging for complex reconstruction and event interpretation algorithms.

For the current L1 trigger at Belle II, the Electromagnetic Calorimeter (ECL) subdetector trigger system is one of the main trigger systems, alongside the Central Drift Chamber (CDC)-based track trigger system and the K_L^0 and Muon Detector (KLM) trigger for muon detection. In combination with the track trigger, it provides a full event trigger for both neutral and charged particles. The ECL trigger is especially important for processes with a low number of tracks, where the track trigger is less efficient. Additionally, the identification of $\mathrm{e}^+\mathrm{e}^- \rightarrow \mathrm{e}^+\mathrm{e}^-(\gamma)$ and $\mathrm{e}^+\mathrm{e}^- \rightarrow \gamma\gamma(\gamma)$ is one of the main purposes of the ECL trigger, as these processes are used to provide exact measurements of the integrated luminosity of Belle II. The current ECL trigger relies on a simple, but very fast and efficient isolated cluster detection algorithm [9].

The development of trigger algorithms, especially for hardware-based trigger systems, will be increasingly important with each step towards higher luminosities and higher energies in HEP. The usage of Machine Learning (ML) techniques for such trigger algorithms widens the possibilities but increases the challenges for implementations on FPGAs due to the high number of multiplications in the matrix-vector calculations in the forward passes of neural networks. Standard ML architectures, such as fully connected networks, are already in use in the Belle II L1 trigger [10] and in development for other HEP experiments [11], [12], [13]. More complex architectures such as Graph Neural Networks (GNNs) have been implemented on firmware [14], but so far not reached the requirements necessary for deployment in a realistic environment.

In this work, I will show the development and implementation of a GNN based on dynamic graph building for cluster reconstruction for the Belle II ECL L1 trigger. The network, in this thesis called GNN-ETM, is based on the GravNet [15] dynamic graph building and predicts and reconstructs an unknown number of clusters with the Object Condensation (OC) algorithm [16]. The GNN-ETM is designed in a hardware-software codesign process to optimize performance while keeping the design within the hardware requirements. In Chapter 2, I will first give an overview over the experimental setup for data taking and describe the ECL detector in detail. Chapter 3 explains the Belle II L1 trigger setup, with a focus on the current ECL trigger algorithm, the ICN-ETM. The training data, including the training target design, and the evaluation datasets will be shown in Chapter 4, while Chapter 5 focuses on the evaluation metrics to test the performance of the GNN-ETM, especially in comparison to the ICN-ETM. Chapter 6 shows the overall network design process, including architecture, network optimizations for implementation, and the actual implementation on hardware. In Chapter 7, the evaluation on the technical datasets is shown. The evaluation of the performance on physics processes for both Monte Carlo (MC) data and collision data is described in Chapter 8. A summary and an outlook for future work will be given in Chapter 9.

Chapter 2

SuperKEKB and the Belle II Experiment

2.1 The SuperKEKB Accelerator

The SuperKEKB accelerator [17] is a double-ring electron-positron collider with asymmetric beam energies, namely 7 GeV for the electron beam, also called high-energy ring (HER), and 4 GeV for the positron beam, also called low-energy ring (LER). SuperKEKB operates at a CM energy of 10.58 GeV, which is the energy of the $\Upsilon(4S)$ resonance. This resonance is chosen as it decays in over 96 % of cases to a pair of B mesons, making the accelerator a so-called B-factory. The asymmetric beam energies result in a Lorentz boost $\beta\gamma = 0.28$, which applies a boost to all produced particles in the direction of the electron beam and allows for precise measurements of B decays.

The accelerator is designed to achieve an instantaneous luminosity of up to $6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, which is 40 times higher than its predecessor KEKB. An overview of the accelerator complex including Belle II can be seen in Fig. 2.1.

To reach this luminosity, higher beam currents, larger vertical beam-beam tune-shift parameters, and smaller vertical beta functions at the IP of Belle II are needed. The final luminosity $\mathcal{L}$ can be calculated via

$$\mathcal{L} \propto \left(1 + \frac{\sigma_y^*}{\sigma_x^*}\right) \left(\frac{I_\pm \xi_{y\pm}}{\beta_x^*}\right) \left(\frac{R_L}{R_{\xi_y}}\right), \tag{2.1}$$

with $I_\pm$ being the beam currents, $I_\pm \xi_{y\pm}$ the tune-shift parameters and β_x^* the beta functions at the IP. $\sigma_{x,y}^*$ are the beam sizes at the IP. As an increase of the beam currents of more than a factor of two to reach the target luminosity is very difficult due to hardware feasibility and operating costs, SuperKEKB employs the so-called nanobeam collision scheme [18]. The beam bunches with small $\sigma_{x,y}^*$ collide at a large horizontal crossing angle and β_x^* can be squeezed to much smaller values.

Additional changes from KEKB include the decrease of the beam asymmetry, changing

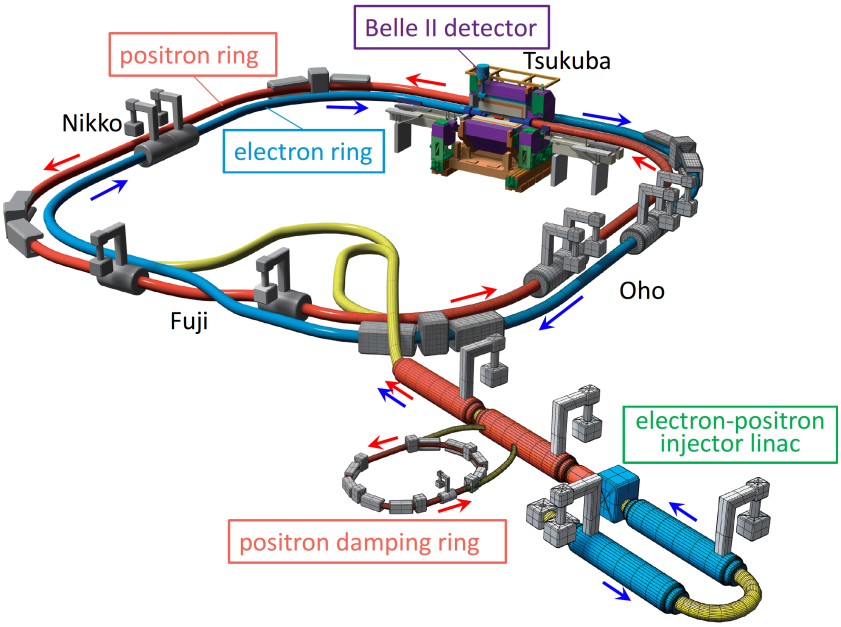


Figure 2.1: Overview of the SuperKEKB accelerator complex, including linear accelerator and the collision point with the Belle II detector. The ring has a circumference of about 3 km. The figure is taken from [17].

the energy of the electron beam from 8 GeV to 7 GeV and respectively increasing the energy of the positron beam from 3.5 to 4 GeV. The increase of beam energy for the positron beam was to improve the Touschek beam lifetime and suppress emittance growth due to intrabeam scattering, while the decrease of energy for the electron beam reduces synchrotron radiation and horizontal emittance. All four effects are also effects that lead to beam background hits at Belle II, which will be explained in detail in Section 4.1.3. Mitigating beam background at Belle II is one of the main challenges in comparison to Belle with the KEKB accelerator.

So far, the SuperKEKB accelerator was able to reach a world-record instantaneous luminosity of $\mathcal{L} = 5.1 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in December 2024 [6].

The main processes, their corresponding cross sections and the process rate at an instantaneous luminosity of $6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ are shown in Table 2.1. The most probable process is Bhabha scattering, at a cross section of 74.4 nb, if at least one of the particles is in the acceptance of the Belle II detector.

2.2 The Belle II Detector

The Belle II detector [20] is the upgrade of the Belle detector [21] taking data at the SuperKEKB accelerator since 2018. Due to the higher background levels at the upgraded SuperKEKB accelerator, which come with the higher instantaneous luminosity, the Belle II detector was designed to maintain the physics performance of the Belle detector while mitigating the difficulties of the additional background detector hits.

Table 2.1: Total production cross sections from collisions at $\sqrt{s} = 10.58\,\mathrm{GeV}$ and production rates at $6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ luminosity at Belle II. W_{ll} is the minimum invariant secondary fermion pair mass. Processes and cross sections are taken from [19].

Process	Selection	Cross Section (nb)	Rate (kHz
$\mathrm{e}^+\ \mathrm{e}^-\ (\gamma)$	e^+ or e^- in ECL	74.4	44.6
$\mathrm{e}^+\ \mathrm{e}^-\ \mathrm{e}^+\ \mathrm{e}^-$	$W_{ll} > 0.5\,\mathrm{GeV/c}^2$	39.7 ± 0.1	23.8
$\mathrm{e}^+\ \mathrm{e}^-\ \mu^-\ \mu^+$	$W_{ll} > 0.5\,\mathrm{GeV/c}^2$	18.9 ± 0.1	11.3
$\gamma\ \gamma\ (\gamma)$	1 γ in ECL, E_γ>0.5 GeV	3.30	1.98
$\mu^-\ \mu^+\ (\gamma)$	-	1.148	0.69
Υ(4S)	-	1.110 ± 0.008	0.67
$\tau^-\ \tau^+\ (\gamma)$	-	0.919	0.55

The detector is a 4π detector centered around the IP in a cylinder shape. The design is slightly asymmetric, with the IP not being in the center of the detector in the z direction, but with a larger detector region towards the forward direction. This is due to the asymmetry of the beam energies, where the particles created in the collision are boosted in the direction of the electron beam. In x and y direction the IP sits at (0, 0) with the different subdetector systems positioned in layers around the beam pipe.

The coordinate system of Belle II is a right-hand coordinate system with the origin at the nominal IP, with the z-axis pointing approximately in the direction of the electron beam. The x-axis points horizontally outwards from the center of the storage ring and the y-axis points vertically upwards. The polar angle θ is defined from 0 to π, with $\theta = 0$ pointing parallel to the z-axis in the forward direction. The azimuthal angle ϕ is defined from $-\pi$ to π, with $\phi = 0$ pointing parallel to the x-axis at $y = 0$.

As can be seen in Fig. 2.2, Belle II consists of seven subdetectors for different purposes: the Pixel Detector (PXD), Silicon Vertex Detector (SVD) and CDC, which make up the tracking detectors of Belle II; the Time-Of-Propagation counter (TOP) in the barrel and the Aerogel Ring-Imaging Cherenkov detector (ARICH) in the endcaps for particle identification (PID); the ECL for neutral particle identification and energy reconstruction; and the KLM for long-lived particles. In the following, I will give a quick overview of the different subdetectors, besides the ECL, which will be described in detail in Section 2.3.

The PXD Detector

The PXD detector [23] at Belle II is the innermost detector and consists of two layers with a silicon-based Depleted Field Effect Transistor (DEPFET) technology [24]. The positioning of this detector directly around the beam pipe leads to a very high occupancy in the detector, where silicon strip detectors cannot be utilized any more, as the number of hits per channel would make a reconstruction of B vertices impossible. The PXD is therefore made up out of 250 x 768 sensors on each module, with each sensor being approximately 50 by 50 µm, with 16 modules in the inner and 24 modules in the outer layer. The layers are

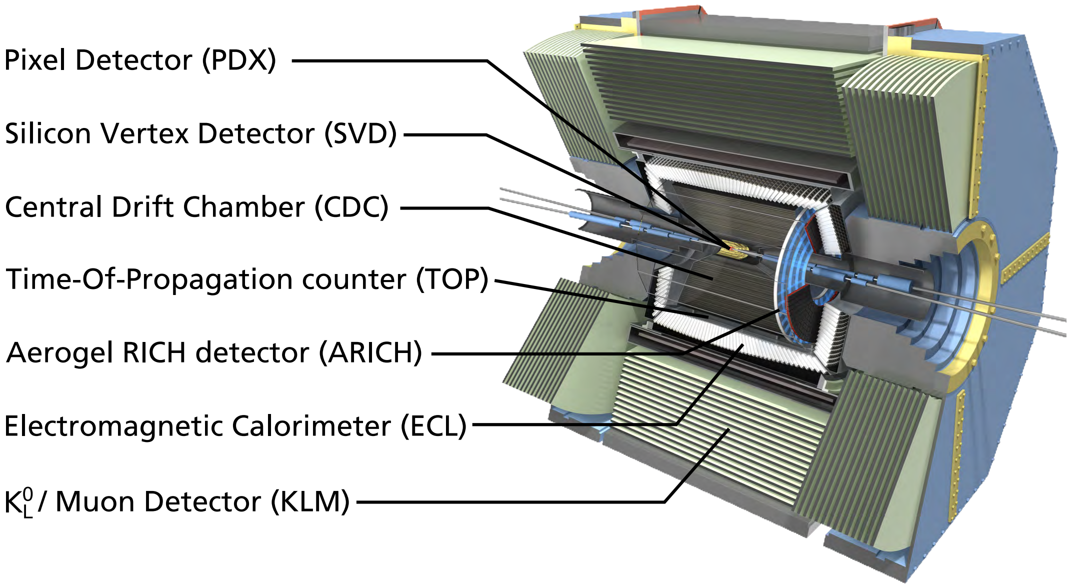

Pixel Detector (PDX)
Silicon Vertex Detector (SVD)
Central Drift Chamber (CDC)
Time-Of-Propagation counter (TOP)
Aerogel RICH detector (ARICH)
Electromagnetic Calorimeter (ECL)
K_L^0/ Muon Detector (KLM)

Figure 2.2: Overview of the Belle II detector with the marked subdetector systems. The IP is in the center of the PXD detector. The figure is taken from [22].

very thin with a material budget of 0.21 % X_0 per layer to minimize multiple scattering.

PXD 1 was deployed in Belle II from the start of data taking in 2018 until the start of long shutdown 1 in 2022. For PXD 1, the outer layer was non-functional besides two modules, which were installed to cover a non-functioning inner layer module. During the shutdown 2022 - 2023, PXD 2 was installed with two functioning layers.

The SVD Detector

The SVD detector [25] is a silicon strip detector positioned around the PXD detector and consisting of 172 strip sensors in four layers. It covers an angular acceptance of $17^\circ < \theta < 150^\circ$, which is achieved by slanted sensors in the forward region of the detector.

The SVD can perform standalone tracking, which is especially important for low-momentum particles that do not reach the CDC, and reconstruct vertices from K_S^0 decaying outside the PXD. Additionally, tracks from the CDC can be extrapolated to the PXD with the help of the SVD.

The CDC Detector

The CDC [26] is the main tracking detector of Belle II. It is designed as a cylindrical wire chamber with approximately 15000 sense wires to track charged particles and measure their momenta in the magnetic field. Additionally, particle identification can be done by using the energy loss within the CDC volume. It is also one of the main detectors used for the L1 trigger. To decrease multiple scattering, the CDC is filled with a mixture of 50 % helium and 50 % ethane, which both have low atomic numbers to reduce the scattering probability.

The design of the CDC splits the sense wires into layers along the z-axis. 32 axial layers have wires that are parallel to the z-axis, while the sense wires in the 24 stereo layers are tilted by up to 74 mrad to provide both transversal and longitudinal tracking information.

The TOP Detector

The TOP detector [27] is the main PID detector in the barrel region of Belle II and is designed to improve the K/π particle separation capabilities. Its design is based on the principle of total reflection of Cherenkov light within a quartz bar. 16 quartz bars with a length of 125 cm are positioned around the detector parallel to the beam pipe. Charged particles traverse a quartz bar and emit Cherenkov light within that bar. The emitted photons are propagated within that bar due to total reflection on the inner sides until they reach one of the ends. A mirror is attached to one end of the bar, reflecting the photons again and a photomultiplier at the other, which measures both the position and the time of the arriving photon. These hit patterns can then be transformed into a likelihood value for the different particle types.

The TOP detector can additionally provide timing information for the L1 trigger.

The ARICH Detector

The ARICH detector [28] provides PID information in the forward endcap region to distinguish kaons and pions, as the particles in Belle II are boosted in the forward region because of the beam energy asymmetry. It is built out of a silica aerogel radiator, in which charged particles, when traversing this radiator, emit Cherenkov photons. These photons are then detected by a photo detector. Through this measurement, the emission angle of the Cherenkov light can be reconstructed, which is proportional to the mass of the charged particle and can therefore be used to identify the particle type. The detector is composed out of 248 aerogel tiles, with each being 17 x 17 cm^2 and 2 cm thick. Two different aerogels with different refractive indices are used to improve the angle resolution.

The KLM Detector

The KLM detector [29] is the outermost detector at Belle II designed to detect K^0_L mesons and muons, that did not get stopped in the inner detector regions. It consists of 14 layers of 4.7 cm-thick steel plates which make up the flux return for the solenoidal magnetic field. These plates are interspersed with active particle detection modules, made up out of plastic scintillators for the endcaps and the first two layers in the barrel, and resistive plate counters for the remaining 13 barrel layers. The steel plates of the flux return add additional stopping power for hadrons, increasing the material budget to 3.9 interaction lengths in addition to the 0.8 interactions lengths of the ECL.

When charged particles pass the KLM, they generate scintillation light in the scintillator strips, which can then be used for the reconstruction of K^0_L decays by clustering 2D points and muons, which are reconstructed by extrapolating charged tracks in the CDC to the KLM [30].

`Belle II Analysis Software Framework`

For the simulation, reconstruction, and analysis of the Belle II data, the collaboration provides the core software framework `Belle II Analysis Software Framework` (`basf2`) ([31] [32]). This framework enables the interface to event generators for MC simulation, simulates the detector response with `GEANT4` [33], unpacks raw data, and reconstructs objects such as particle tracks and clusters from detector hits. For analysts, these reconstruction objects can be combined to particle hypothesis with custom cuts to dynamically adapt to every analysis setup.

2.3 The Electromagnetic Calorimeter

The Belle II ECL is a scintillation crystal based, single layer electromagnetic calorimeter used mainly for the detection of photons, identification of electrons and detection of K_L^0 together with the KLM [20]. Photons are one of the most common final-state particles in B decays, with about a third of the decays having at least one photon originating from π^0 or other particles. These photons cover a wide energy range between 0.02 and 4 GeV, while processes such as $e^+e^- \to \gamma\gamma(\gamma)$ commonly result in photons up to 7 GeV. The ECL is also one of the main detectors to issue L1 trigger signals, with the detection and veto of $e^+e^- \to e^+e^-(\gamma)$ processes being solely the responsibility of the ECL. The calorimeter hardware was taken from the Belle experiment [21], with a major electronics upgrade to improve performance for the higher backgrounds at Belle II.

The ECL consists of 8736 Thallium-doped Cesium-Iodide (CsI(Ti)) crystals for the forward endcap, the barrel region and the backward endcap, covering the angular region between 12.4° and 155.1° and weighting 43 tons in total. There are two 1° gaps between the barrel region and the respective endcaps. The crystals are arranged such that they point almost to the IP. A slight tilt of approximately 1.3° in θ and ϕ avoids the escape of photons between the gaps of two crystals. In Fig. 2.3, a technical overview of the crystals in the barrel, forward and backward endcap can be seen with their positioning in relation to the IP at the Belle experiment. The hardware was taken as is for the Belle II experiment.

For the barrel, 6624 crystals are positioned in a cylindrical shape at an inner radius of 1.25 m. The forward endcap consists of 1152 crystals at $z = 1.96$ m, while the backward endcap has 960 crystals at $z = -1.02$ m. The crystals in the barrel have in total 29 distinct shapes, but are of an average size of 6 x 6 cm^2 for the cross-section and 30 cm long, which corresponds to 16.1 X_0. The crystals in the endcap have a high variation in geometry, with a total of 69 different shapes, in order to form a stable structure without gaps, with the surface dimensions facing the IP varying between 44.5 to 70.8 mm. The length of approximately 30 cm is the same for all crystals to reduce the effect of the longitudinal leakage. Longitudinal

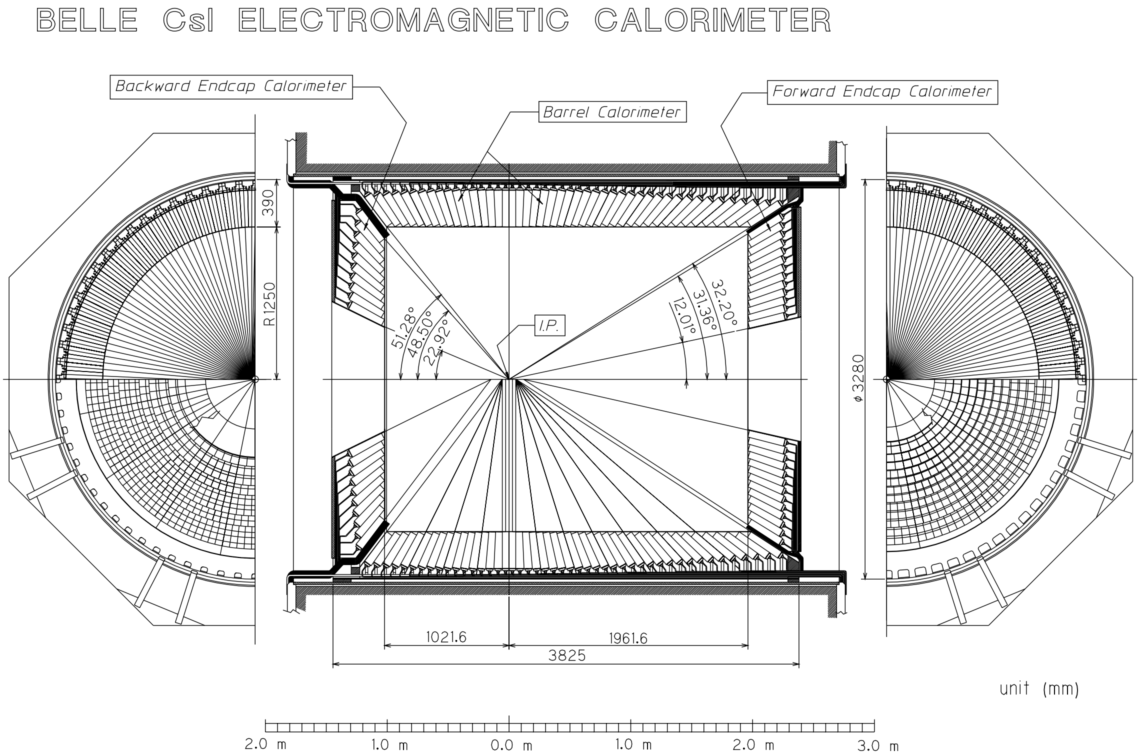


Figure 2.3: Schematic overview of the Belle ECL detector, with all crystals located in the barrel, forward and backward endcap. The center part depicts the ECL in a cut through the y-z axis, while the left and right parts show a cut through the x-y plane with the barrel view in the upper half and the backward and forward endcap in the lower half, respectively. For Belle II, the Belle ECL was taken without changes to the crystals or their positioning. The figure is taken from [21].

leakage is the term for the longitudinal energy loss when the electromagnetic shower of high-energetic particles is not fully contained in the length of the crystal but exits the back of the calorimeter. The size of the crystals is additionally chosen such that a photon entering at the center of the front-facing side creates a shower, where approximately 80 % of the photon energy is contained in this crystal. The remaining 20 % are contained in the surrounding crystals and have to be reconstructed as one cluster. Smaller crystals would improve the position resolution but worsen the energy resolution due to an increase in gaps and inactive material.

Each crystal has two attached photodiodes glued to the rear surface with an attached preamplifier. These two outputs provide independent measurements of the light yield within that crystal and are summed up in the later positioned shaper board. The photodiode sends the signal to a Shaper-Digital Signal Processor, which digitizes the signal and fits 16 points of the signal to a signal shape function $F(t) = A_0 \times f(t-t_0)$, with A_0 being the pulse height and t_0 the event time [34]. One improvement over the Belle readout electronics is the higher sampling frequency at 0.5 µs. This allows a much more efficient beam background cluster rejection through the better determination of t_0. Additionally, the digitized waveforms of crystals with an energy deposit above 30 MeV are recorded in the raw data to enable an offline pulse-shape analysis for the discrimination between hadronic and electromagnetic showers [35]. The fast shaper for the trigger signal, explained in Chapter 3, is located on the same board.

The amplitude, time, quality of the fit, and for higher-energetic crystals the digitized waveform are stored in raw data for later reconstruction. The reconstruction algorithm of the ECL implemented in the `Belle II Analysis Software Framework` is explained in Section 4.1.2.

Chapter 3

The Existing First Level Trigger Setup at Belle II

The current trigger setup at the Belle II Experiment consists of two steps, the L1 trigger and the HLT. The L1 trigger uses optimized reconstruction algorithms for event reconstruction on FPGAs to make a first decision to keep the event with very high throughput and latency requirements. The HLT then runs the full reconstruction with Belle II Analysis Software Framework (basf2) on CPUs and makes a second decision if an event, that already passed the L1 trigger, will be kept. Only if both the L1 trigger and the HLT decide to keep the event, the data is written to disk and permanently stored.

As this thesis shows an improved algorithm for the ECL L1 trigger, in the following I will provide an overview of the current L1 trigger setup and a detailed explanation of the ECL trigger.

3.1 The Level 1 Trigger

The L1 trigger at Belle II is designed for fast decisions with a high throughput rate and fixed latency. For the current trigger setup the CDC, ECL, KLM, and TOP are used to make the final trigger decision. The L1 trigger system is designed for a maximum output rate of 30 kHz. Additionally, due to the reduced amount of buffer available, the decision has to be made within a timeframe of 5 µs. The restricting factor for the latency budget for the L1 trigger system is the size of the front-end buffers of the subdetector systems. The smallest latency of these buffers determines the maximum L1 trigger latency, as the data is otherwise lost before the L1 trigger has made its decision. In Table 3.1, an approximation of the different maximum subdetector latencies is shown. The current L1 trigger latency is determined by the SVD. This latency requirement is a requirement for the whole trigger decision chain from data taking to Global Decision Logic (GDL) decision, which puts even more stringent limits on each subdetector reconstruction algorithm.

At Belle II the bunch crossing rate, as explained in Chapter 2, is 250 MHz. For the most

Table 3.1: Maximum allowed L1 trigger latency for the different subdetector systems due to the buffer of the detector front-ends. Values are based on internal communication with Taichiro Koga.

Subdetector	Max. Latency (µs)
PXD	$\approx$ 10 µs
SVD	5 µs
CDC	15 µs
TOP	9 µs
ARICH	up to 1015 µs
ECL	100 µs
KLM	5.2 µs

probable process $e^+e^- \to e^+e^-(\gamma)$ at an instantaneous luminosity of $6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, if at least one of the particles is emitted at an angle that is in the acceptance of the ECL, the process rate is 45 kHz, while the process $e^+e^- \to \Upsilon(4S)$ happens at a rate of $\approx$ 1 kHz. More high-rate processes and their respective rates can be found in Chapter 2, Table 2.1. This means that for the requirement of 30 kHz, a reduction of at least 50 % of all collision events is necessary, not counting false triggers due to beam background (see more in Section 4.1.3). High-rate processes such as $e^+e^- \to e^+e^-(\gamma)$ have to additionally be suppressed with high efficiency to achieve the required $\approx$ 100 % trigger efficiency for $e^+e^- \to B\overline{B}$ and $e^+e^- \to q\bar{q}(\gamma)$ events. Furthermore, Belle II is an ideal experiment to measure low multiplicity final states such as $e^+e^- \to \pi^+\pi^-\pi^0$ [36], which can help measure the e^+ e^- hadronic cross section, and lepton flavour violating τ decays such as $\tau^- \to \mu^-\mu^+\mu^-$ [37]. High trigger efficiencies for these and similar processes improve the measurements drastically. Additionally, ideally unbiased trigger decisions based on single tracks or energy sums without clustering improve searches for new physics such as dark matter searches [38] [39].

The throughput and latency requirements make an implementation on FPGA necessary. FPGAs are equipped with logic gates and RAM blocks for complex digital computations and can implement the same logical functions as ASICs. In comparison to ASICs, FPGAs can be reprogrammed to serve several functions while an ASIC is designed and optimized for one specific use case. Due to this, ASICs have high development costs and cannot be exchanged or adapted easily. The generic processing units, such as CPUs or GPUs for fast processing of parallelizable tasks, do not achieve the needed latency and throughput requirements necessary for the Belle II L1 trigger system. FPGAs are significantly faster than CPUs or GPUs for specific tasks due to their high ability to parallelize and custom optimizations for each application.

To achieve both reasonable cost requirements and the latency and throughput required by the trigger setup, FPGA implementations have been chosen for the design of the L1 trigger system. Besides Frontend and Merger chips, the current modules either use the Universal Trigger Board 3 (UT3) for the implementation of their algorithms, which features a Xilinx Virtex-6 FPGA, or the Universal Trigger Board 4 (UT4) board featuring a Xilinx Ultrascale XCVU080/160. For each different subdetector trigger system, one or several FPGAs are used in parallel to process and reconstruct the data of the different subdetectors. The overall L1 trigger system is without deadtime and processes the data in a first-in first-out pipeline [7].

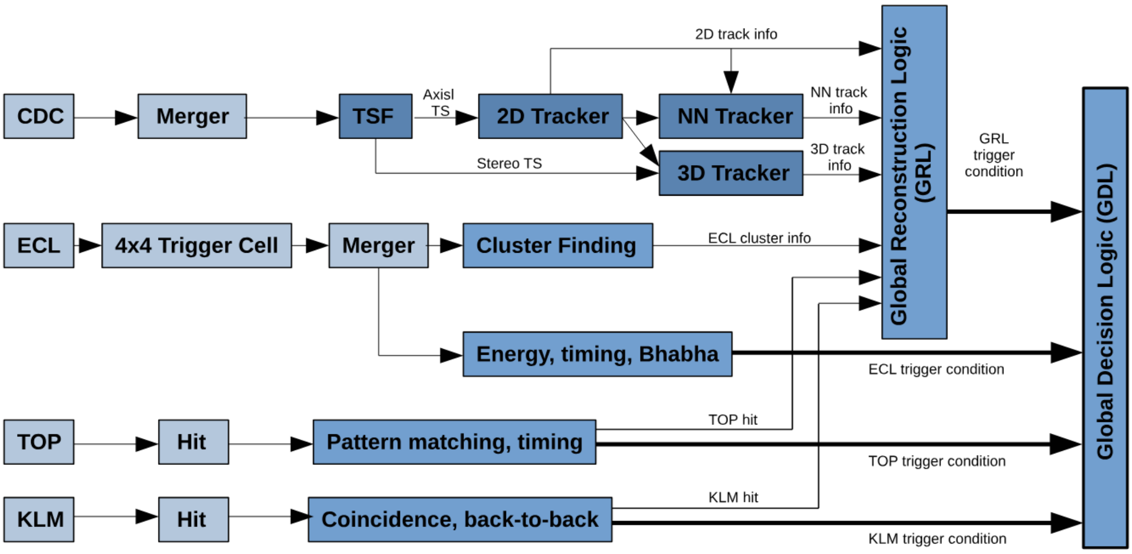


Figure 3.1: An overview of the Belle II L1 trigger system. The data from the CDC, ECL, KLM and TOP subdetectors is taken, reconstructed and sent to both the Global Reconstruction Logic (GRL) and GDL which then outputs the final trigger decision. The figure is taken from [40].

In Fig. 3.1 an overview of the L1 trigger dataflow can be seen. Data is taken by each subdetector separately with different rates and detector deadtimes. The entire L1 trigger system operates on a common clock with a frequency of 127.216 MHz, which is the global SuperKEKB clock of 508.9 MHz divided by 4. Additionally, a revolution signal with a period of 10 µs is distributed among the different subdetector systems. Both clocks assure a synchronization of all detector systems and the L1 trigger.

The CDC trigger provides information of charged particles by reconstructing charged tracks in the CDC and reconstructing their momenta, $z0$, which is the distance between the start of the track and the IP in z-direction, and the θ angle. The CDC trigger currently consists of three different track reconstruction algorithms, the 2D, 3D and NeuroZ tracker.

In the ECL trigger system, clusters for energy reconstruction are found by identifying connected energy depositions in the ECL. A more in-depth explanation is given in Section 3.2.

The KLM trigger finds clusters in the KLM system to improve muon identification and give approximate direction estimates. It additionally calculates back-to-back information for clusters in the KLM.

For the TOP trigger system, event timing is calculated from hit signals given by the 16

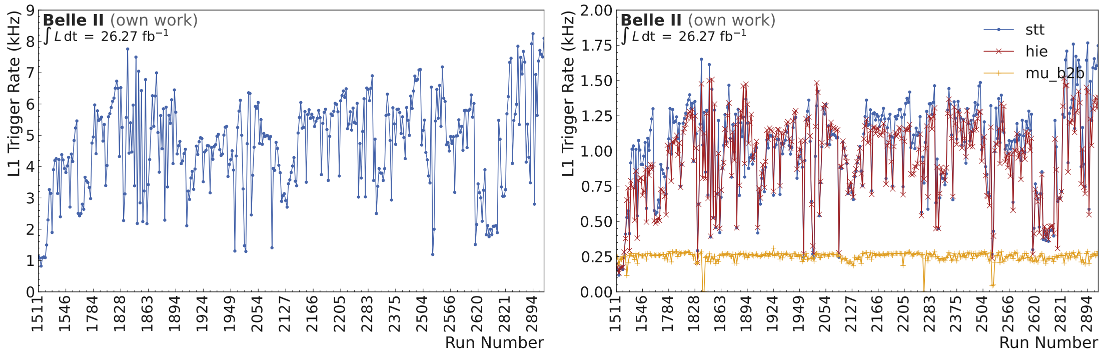


Figure 3.2: Total and selected output bit trigger rate for the last month of data-taking, December 2024, in Exp. 35. On the left the total trigger rate in kHz is shown, on the right the rate for the *hie*, *stt* and *mu_ b2b* trigger bits. This three trigger bits are output bits for the ECL, CDC, and KLM respectively, with no requirement of other subdetector information, each being the bit with the highest output rate of all bits of the respective subdetector.

quartz bars of the TOP detector. The TOP trigger is currently not involved in the trigger decision to keep the event.

The inputs from the different subdetector systems are combined on the global trigger boards, the GRL and GDL. The subdetectors send reconstructed objects such as tracks and clusters to the GRL, where they can be matched and combined to improve event interpretation and reduce false triggers due to beam background. The information is condensed into so-called trigger input bits, which require set conditions to be true and are then set to 1. The GDL receives already calculated trigger input bits from the GRL and the subdetector systems and combines them into trigger output bits by using AND, NOT, or OR operations. A trigger signal is issued if at least one output trigger bit equals 1. In Fig. 3.2, the full L1 trigger output rate for the last month of data-taking (Dec. 2024) and the output rates for the *hie*, *stt* and *mu_ b2b* output bit are shown. *hie*, *stt* and *mu_ b2b* are output bits with information only taken from the ECL, CDC, and KLM respectively. These three bits are the bits with the highest trigger rate of the all bits of the respective detector. The *hie* bit is the sum of energy in the inner calorimeter (more explanation in Section 3.2.1). The *stt* bit is the single track trigger, which requires one track given by neural network track reconstruction ([10]). For the KLM bit, the *mu_ b2b* bit requires two hits in the KLM that are on opposite sides of the detector.

To reduce the rates of trigger output bits, a separate prescale for each output bit can be set. An exemplary prescale of 100 reduces the trigger rate of one bit by 100. This works by setting a counter of the prescale value and increasing the counter by one every time the output bit is 1. If the prescale value is reached, the counter is set to 0 again. The output bit only contributes to the trigger decision if the counter is at 0. These prescale values can be tuned individually and changed for different runs. The raw output trigger bit information is called the Final Trigger Decision Line (FTDL) value, while the value after the prescale is called the Pre-Scale And Mask (PSNM) value.

The event timing can be determined by the ECL, CDC and TOP trigger systems. In practice, the ECL trigger provides the trigger timing in over 99 % of cases with the CDC trigger as a fallback option in past data runs.

3.2 The Calorimeter Trigger Setup at Belle II

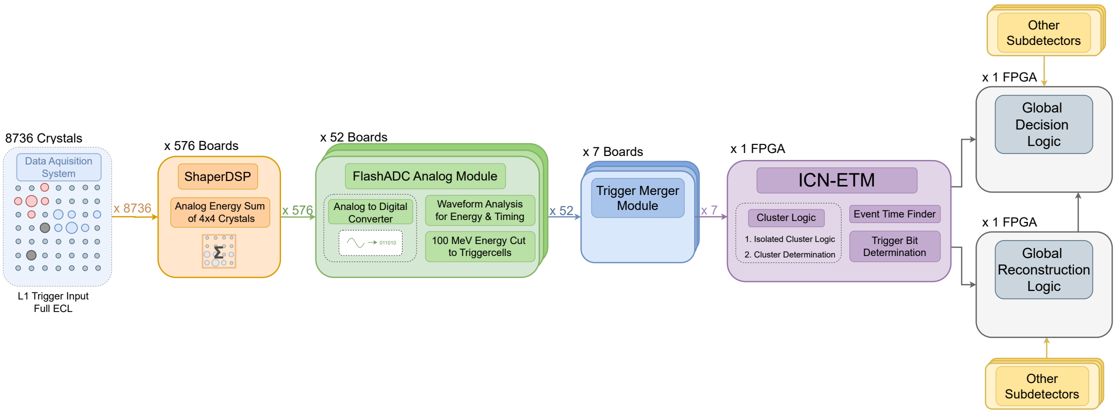


Figure 3.3: An overview of the current ECL Trigger Chain. All crystals are read out, analogously summed into Trigger Cells (TCs), and an energy cut of 100 MeV is applied. All incoming TCs are merged on the Trigger Merger Module and sent to the ICN-ETM, where we perform clustering, event timing, and trigger bit determination. This decision is sent both to the GRL and GDL, where the trigger decision is made.

The ECL L1 trigger consists of four different stages with a total of 636 modules, where each module consists of a board including an FPGA and a large Input-Output unit with optical transceivers. An overview of the different stages with the number of boards per stage is shown in Fig. 3.3. The data of all 8736 crystals in the ECL are read out and then sent to 576 ShaperDSP modules located around the detector. These modules contain both a slow shaper for offline data (see Section 2.3) and a fast shaper for trigger information. On each ShaperDSP, usually 16 crystals are merged into one TC and this data is transmitted to the FlashADC Analog Module (FAM). On this module, a digitization of the data is performed and the energy and timing of the TCs are analysed. This data is then sent to the Trigger Merger Module (TMM) which forwards the merged data to the ECL Trigger Master Module (ETM), where the main clustering logic and trigger bit determination is implemented. To avoid confusion, in this thesis the default ETM is called the ICN-ETM based on the implemented logic. The following sections will explain each step in more detail.

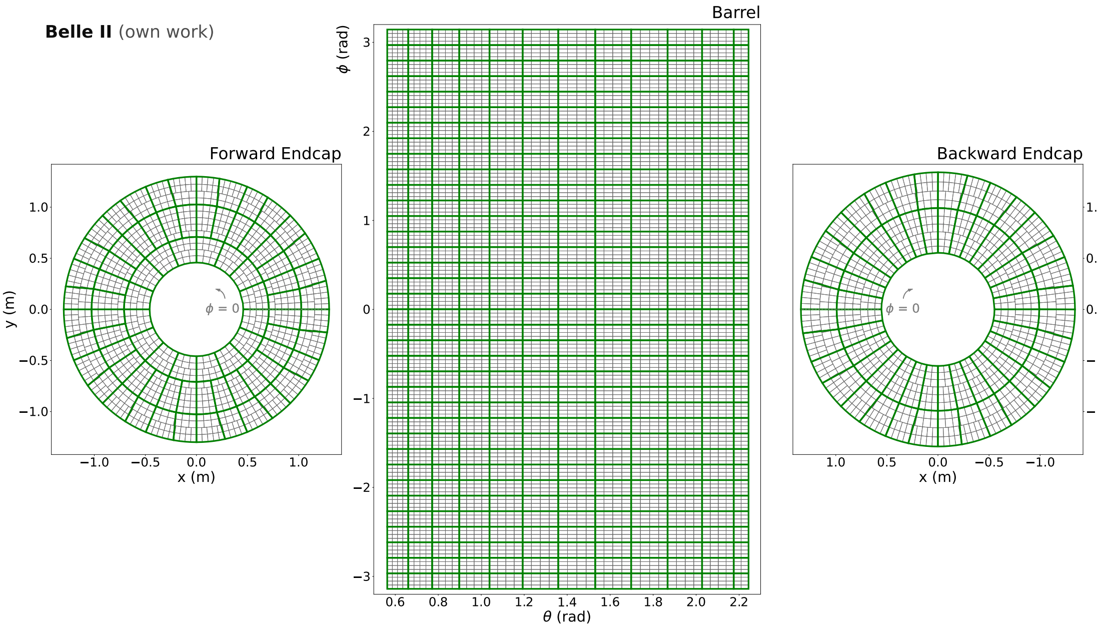
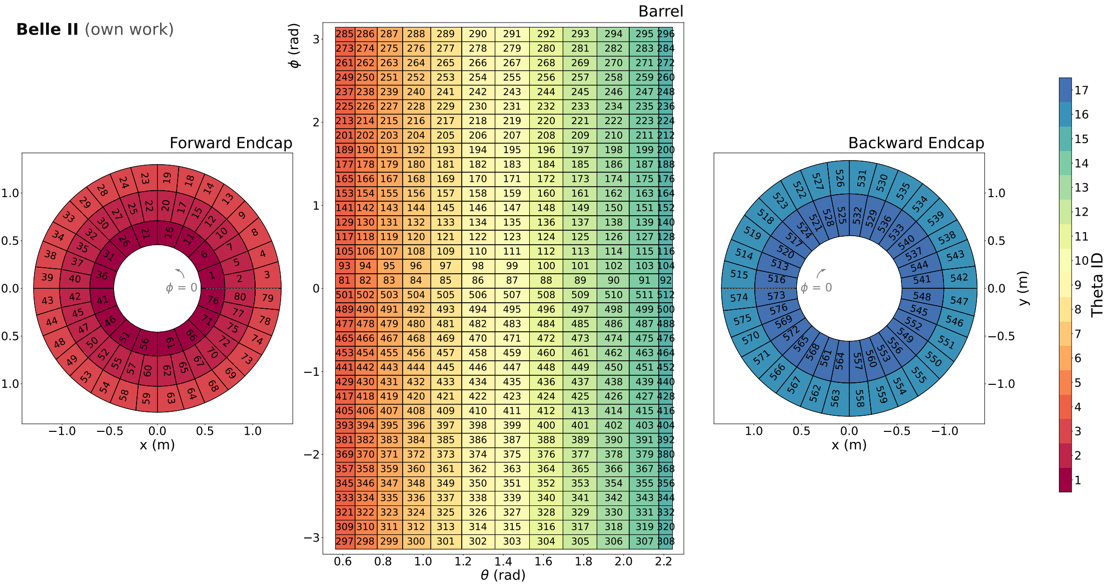


Figure 3.4: Overview of the Belle II ECL with the single crystal outlines of the 8736 CsI(Ti) crystals in black and the TC outlines in green.

Data Acquisition and Merging into Trigger Cells

As explained in Section 2.3, the ECL consists of 8736 CsI(Ti) crystals divided into the forward endcap, the barrel and the backward endcap. Each crystal is equipped with a PIN photodiode and a pre-amplifier to collect the light yield of the electromagnetic shower within the crystal. The crystal data is sent to 576 ShaperDSP modules, DSP meaning Digital Signal Processor in this case.

Each ShaperDSP module consists of two shapers, a slow shaper for more exact information used in offline data (more details in Section 2.3) and a fast shaper for the trigger data. In the fast-shaper circuit, up to 16 analog crystal signals given by the preamplifier are merged into one TC with a shaping time of 200 ns [9]. This is done to reduce the number of input signals to the cluster finding algorithm on the ICN-ETM. In Fig. 3.5, all TCs within the ECL are shown. The 8736 crystals result in 576 TCs. Each TC receives a θ ID ranging from 1 to 17 and a ϕ ID ranging from 1 to 36, both proportional to the θ and ϕ angle. The mapping of the IDs to the corresponding angles can be found in Appendix A. In most cases, one TC consists of 16 crystals. Exceptions are several TCs in the endcaps due to the irregular crystal geometry, where the TCs then consist of 14 or 15 crystals. Additionally, the TCs in the barrel region closest to the backward endcap only have 8 crystals. Fig. 3.4 shows the CsI(Ti)s with their corresponding TC outline. The pulse height of each crystal signal is corrected before merging. This correction is needed due to the different light yields depending on the crystal properties. The correction factors are determined using cosmic data [41]. After merging, each ShaperDSP module sends the TC signal to the FAM.



285	286	287	288	289	290	291	292	293	294	295	296
273	274	275	276	277	278	279	280	281	282	283	284
261	262	263	264	265	266	267	268	269	270	271	272
249	250	251	252	253	254	255	256	257	258	259	260
237	238	239	240	241	242	243	244	245	246	247	248
225	226	227	228	229	230	231	232	233	234	235	236
213	214	215	216	217	218	219	220	221	222	223	224
201	202	203	204	205	206	207	208	209	210	211	212
189	190	191	192	193	194	195	196	197	198	199	200
177	178	179	180	181	182	183	184	185	186	187	188
165	166	167	168	169	170	171	172	173	174	175	176
153	154	155	156	157	158	159	160	161	162	163	164
141	142	143	144	145	146	147	148	149	150	151	152
129	130	131	132	133	134	135	136	137	138	139	140
117	118	119	120	121	122	123	124	125	126	127	128
105	106	107	108	109	110	111	112	113	114	115	116
93	94	95	96	97	98	99	100	101	102	103	104
81	82	83	84	85	86	87	88	89	90	91	92
501	502	503	504	505	506	507	508	509	510	511	512
489	490	491	492	493	494	495	496	497	498	499	500
477	478	479	480	481	482	483	484	485	486	487	488
465	466	467	468	469	470	471	472	473	474	475	476
453	454	455	456	457	458	459	460	461	462	463	464
441	442	443	444	445	446	447	448	449	450	451	452
429	430	431	432	433	434	435	436	437	438	439	440
417	418	419	420	421	422	423	424	425	426	427	428
405	406	407	408	409	410	411	412	413	414	415	416
393	394	395	396	397	398	399	400	401	402	403	404
381	382	383	384	385	386	387	388	389	390	391	392
369	370	371	372	373	374	375	376	377	378	379	380
357	358	359	360	361	362	363	364	365	366	367	368
345	346	347	348	349	350	351	352	353	354	355	356
333	334	335	336	337	338	339	340	341	342	343	344
321	322	323	324	325	326	327	328	329	330	331	332
309	310	311	312	313	314	315	316	317	318	319	320
297	298	299	300	301	302	303	304	305	306	307	308

Figure 3.5: TC placement in the ECL. The current TC setup consists of 576 TC, shown here with their corresponding TC ID. The color of the TC denotes the θ ID, going from 1 in the innermost forward endcap to 17 in the innermost backward endcap. Each TC also has a ϕ ID, which ranges from 1 to 36 with 1 starting at $\phi = 0^\circ$ (not depicted in this figure).

Each FAM receives 12 TC input signals. The analog data is digitized by a Fast Analog-to-Digital Converter (FADC) and the digitized data is passed to an FPGA to evaluate it [42]. To measure energy and timing of the TC, a waveform analysis is applied on the FPGA. Every 125 ns, 12 data points are used as input to the following χ^2 fit:

$$\chi^2 = \sum_{i,j} \left(y_i - Af(t_i - \delta t - t_0) - P\right) S_{ij}^{-1} \left(y_j - Af(t_j - \delta t - t_0) - P\right) \qquad (3.1)$$

with y_i being the data points and S_{ij} the noise covariance matrix for the sampled data. The fit function consists of a pedestal function for the first four points and a signal function for the following 8 data points. P denotes the pedestal. The fit is performed to measure A and t_0, which correspond to the signal amplitude and the starting time of the signal. In Fig. 3.6 an example of the data and the fit window is shown.

On the FAM, an additional 100 MeV energy threshold is applied to each digitized signal of the incoming TCs. This reduces the amount of inputs to the clustering algorithm further and, more importantly, minimizes the amount of beam background energy depositions being forwarded to the ICN-ETM. The threshold value is set on the hardware and could be adjusted to lower or higher values. However, an analysis of the noise level of the pedestal variation evaluated the noise to be 11.5 MeV for one standard deviation, which is significantly lower than the 100 MeV energy threshold [20]. Without any or with a substantially lower threshold, this pedestal variation would introduce more noise in the calculation of the amplitude of the TC signal.

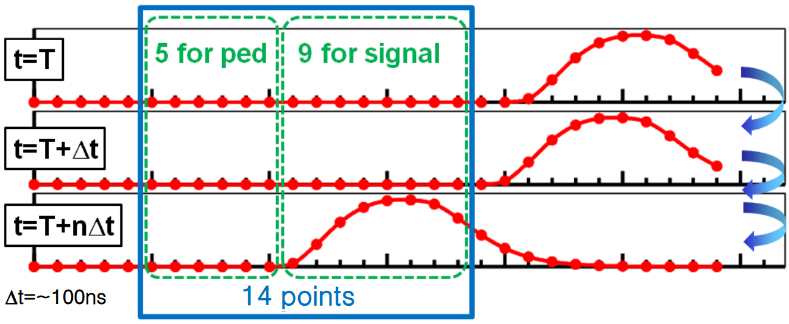


Figure 3.6: Example for data points used for the waveform analysis on the FAM. The fit function consists of a pedestal function for the first 4 points and a signal function for the following 8 data points. This is done for timing intervals of every 125 ns. The figure is taken from [43].

All 576 TCs are then sent to the TMMs. Each TC signal consists of a hit bit, which is 1 if the energy is above 100 MeV, 7 bit timing information, and 12 bit energy information. The timing is in 1 ns steps, representing the timing of a TC within one fit window (125 ns). The energy information is transmitted as ADC counts, where 1 ADC count corresponds to 5.25 MeV.

3.2.1 ICN and Clustering Logic

The cluster finding logic, called ICN logic, on the ICN-ETM is implemented based on the well-tested logic at the Belle Experiment ([44]). The ICN logic is based on the detection of isolated clusters by finding regions of connected TCs and combining those regions into one ICN hit. ICN stands for number of isolated clusters. After finding all ICN hits, these hits are then input to the actual clustering algorithm to find the correct energy and position of the clusters.

In Fig. 3.7 the full ICN logic of the ICN-ETM is shown. The ICN-ETM receives all TCs with their corresponding hit flag, energy, and timing from the TMMs modules. The logic then scans the entire ECL in 3x3 TC windows. In each window, the center column and the lower two TC of the left column are input to a decision algorithm (see Fig. 3.7 (b)). The ICN logic decides that a window of TCs with energy depositions are a cluster when three conditions are fulfilled:

- The center TC (TC 0) has to be hit.
- Neither the top center TC (TC 1) nor the middle left TC (TC 2) are hit.
- Not both of the left TC of the lower row, TC 3 and TC 4, are hit.

The ICN logic returns an ICN hit if all three conditions are true.

Due to limited resources available on the ICN-ETM, the clustering algorithm following the ICN logic cannot be performed on all ICN hits. Therefore, a reduced number of ICN

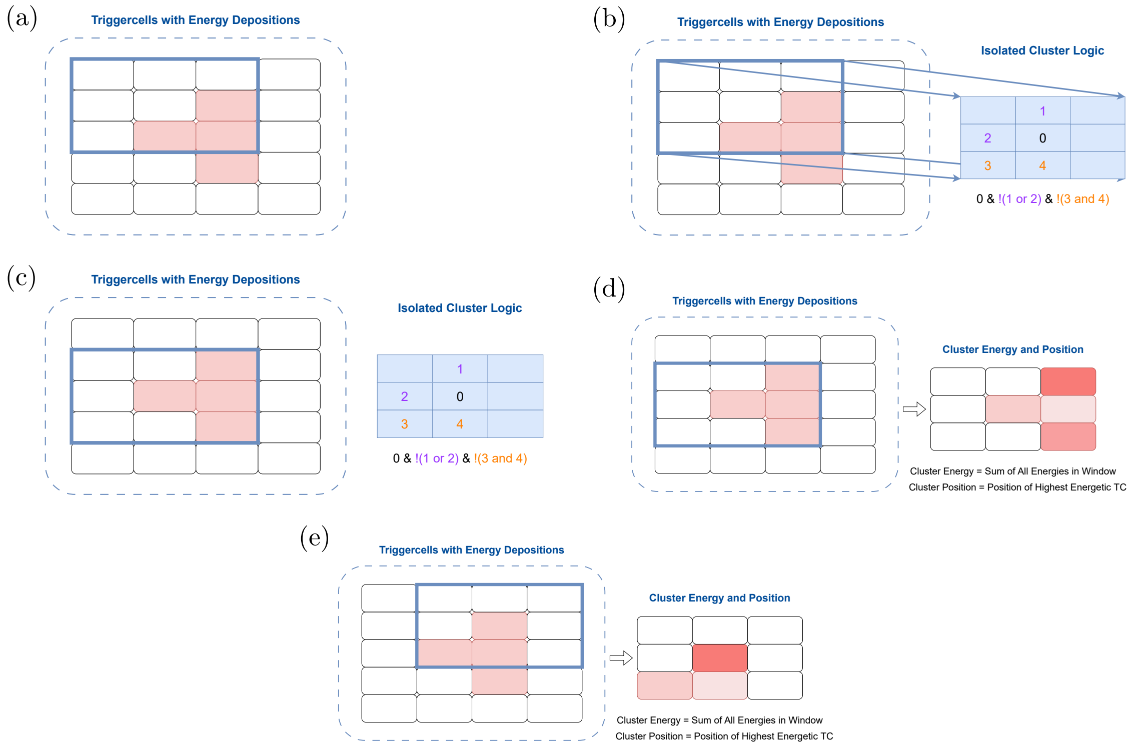

(a)
Triggercells with Energy Depositions
(b)
Triggercells with Energy Depositions
Isolated Cluster Logic
1
2
0
3
4
0 & !(1 or 2) & !(3 and 4)
(c)
Triggercells with Energy Depositions
Isolated Cluster Logic
1
2
0
3
4
0 & !(1 or 2) & !(3 and 4)
(d)
Triggercells with Energy Depositions
Cluster Energy and Position
Cluster Energy = Sum of All Energies in Window
Cluster Position = Position of Highest Energetic TC
(e)
Triggercells with Energy Depositions
Cluster Energy and Position
Cluster Energy = Sum of All Energies in Window
Cluster Position = Position of Highest Energetic TC

Figure 3.7: Isolated Cluster Logic. The algorithm evaluates a field of three by three TCs at once (as shown in (a)) and checks all TCs that have an energy above threshold. If the condition written in (b) is fulfilled, the algorithm returns an ICN hit (case (c)). For this ICN hit, in the clustering algorithm the energies of all TCs is checked (see (d)). If the highest-energetic TC is not the center, the window is moved once to fulfill this condition as seen in (e). The cluster energy is then calculated as the energy sum of all TCs in the window and the cluster position is the center position of the highest-energetic TCs.

hits is chosen, in the case of Belle II only six ICN hits. Early tests within the collaboration on simulated samples showed a negligible reduction of the trigger efficiencies if only six instead of all ICN hits were selected, this is another limitation of the ICN-ETM.

The selection of ICN hits to perform the clustering algorithm on is given by the TC ID. As the barrel region of the ECL contains less background hits on average, and it is the biggest region, ICN hits are first taken from the barrel (TC IDs 81-512). If there are less than six ICN hits within the barrel region, the selection algorithm moves to the forward endcap (TC IDs 1-80), which has significantly less background energy depositions than the backwards endcap. If the number of ICN hits is still below six, then ICN hits from the backwards endcap (TC IDs 513-576) are added to proceed to the clustering algorithm.

The input to the selection decision is 576 x 1 bit (1 if the ICN algorithm fired on this TC, 0 if not). The output is 12 bits per cluster, which contains 1 bit for cluster existence and 11 bits for the position of the cluster, which is a mapping to the different detector regions and θ and ϕ IDs. Due to the restriction to six clusters, 6 x 12 = 72 bits are returned by this decision. This is then used as input to the clustering algorithm, which finds energy

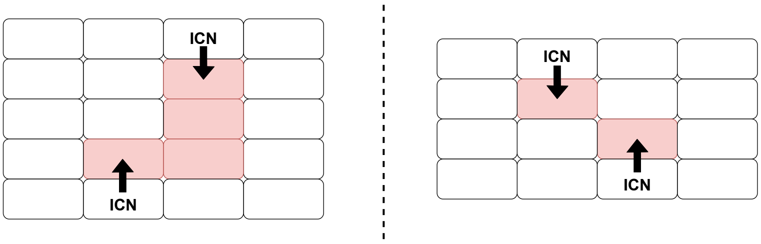


Figure 3.8: Bugs for the ICN cluster finding logic. In both cases, the ICN logic returns two hits even though there should be only one. In the case on the left, the two results might have different energies and positions, while in the right case both results are identical.

and position of the cluster.

For the energy and position of the cluster, the position of the highest-energetic TC within the evaluation window is checked. If this TC is already at the center of the window, the cluster energy is determined by summing up the energies of all hit TCs within the window. The cluster position is then the position of the highest-energetic TC, which is defined as the center of the TC front facing the interaction point. If the highest-energetic TC is not located at the center of the evaluation window, the window is moved, so that this TC is at the center. The energy is then again calculated as the sum of all TC energies within the window, while the position is the position of the TC that now is at the center of the evaluation window. In the case of a higher-energetic TC which is part of the evaluation window after it has been moved once, this procedure is not repeated and the position is still set as the position of the center of the evaluation window.

Due to the very fast, but simple design of the algorithm and the rather large 3×3 window, multiple cases arise where two clusters with the exact same information are returned. In Fig. 3.8, both cases yield two ICN hits, with the right case always returning the same cluster energy and timing information. In the left case, if either of the two TCs, which are not the ICN centers, have the highest energy deposit, the algorithm also returns two duplicate clusters. This is a known limitation of the current logic and leads to a double-counting of energy depositions.

The precision of the output values is 12 bits for the cluster energy in ADC counts from 0 to 4096, where 1 ADC count corresponds to 5 MeV, 7 bits for the cluster θ angle (0 to 180°) and 8 bits for the cluster ϕ angle (0 to 360°), which results in a 1.406° precision.

3.2.2 Trigger Window Decision and Trigger Timing

The sampling of TCs happens at an 8 MHz rate, corresponding to new TC information every 125 ns. This means that the full clustering logic of the ECL L1 trigger also has to run at this frequency, resulting in a high throughput requirement on the clustering algorithm. One 125 ns window is called an ECL-TRG data window. The input to the ICN algorithm and then subsequently to the clustering and trigger bit determination is done in

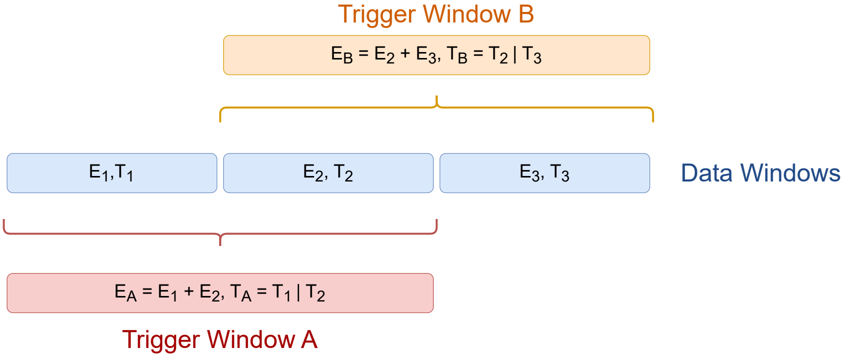


Figure 3.9: Comparing two adjacent trigger windows. The trigger windows A and B are comprised of the energy sum of the two corresponding data windows and have as timing the timing of the highest-energetic TC in the higher-energetic data window.

250 ns windows, using two adjacent ECL-TRG data windows as input. The 250 ns window is called the ECL-TRG trigger window.

The ECL-TRG event timing within the trigger window is given by the highest-energetic TC. For both data windows within the trigger window, data window 0 and data window 1, the TC with the highest energy within this window is determined, TC_0 and TC_1, respective. If $\mathrm{E}(\mathrm{TC}_0) \geq \mathrm{TC}_1$, the ECL-TRG event timing is the timing of TC_0. If $\mathrm{TC}_0 < \mathrm{TC}_1$, the ECL-TRG event timing is $\mathrm{T}(\mathrm{TC}_1)$. If two or more TCs in one data window have the same energy, then the ordering is given by the ordering of the FAM crates. Due to this hardware configuration, the order by TC number is: 81-512, 76-80, 1-75, 573-576, 513-572.

Additionally, due to requirements determined by the GDL, no two consecutive ECL-TRG trigger windows should be triggered. This is achieved by looking at three adjacent data windows, window 1, 2, and 3. Each window has the energy $\mathrm{E}_{1/2/3}$, which is the energy sum of all TCs within that window, and a timing $\mathrm{T}_{1/2/3}$, which is the timing of the highest-energetic TC. With those three data windows, two trigger windows could be chosen, trigger window A, corresponding to data windows 1+2, and trigger window B, corresponding to data windows 2+3. This can be seen in Fig. 3.9. To decide, the event timing is determined for trigger window A, following the algorithm described in the preceding paragraph. If the event timing is given by T_1, trigger window A is chosen as the actual trigger window and no trigger signal can be sent out for trigger window B. If the event timing is given by T_2, the energy sum of trigger windows A and B is compared. If $\mathrm{E}_\mathrm{A} = \mathrm{E}_1 + \mathrm{E}_2 \geq \mathrm{E}_\mathrm{B} = \mathrm{E}_2 + \mathrm{E}_3$, then trigger window A is chosen as the actual trigger window. Whereas, if $\mathrm{E}_\mathrm{A} = \mathrm{E}_1 + \mathrm{E}_2 < \mathrm{E}_\mathrm{B} = \mathrm{E}_2 + \mathrm{E}_3$, then trigger window B is chosen as the actual trigger window.

3.2.3 Trigger Bit Determination on ICN-ETM

Most trigger input bits are calculated on the GRL as they contain information given by several subdetectors. These input bits are then sent to the GDL where they are combined into the trigger output bits and the final trigger decision is made. Additionally, pure ECL trigger input bits, which only need information coming from the ECL trigger, are calculated directly on the ICN-ETM and then sent to the GDL for the trigger decision. Those input bits that are at the time of this work actively used in trigger decisions and calculated on the ICN-ETM are listed in Table 3.2.

For correct selections of relevant physics processes, such as $e^+e^- \rightarrow e^+e^-(\gamma)$ and $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$, which, for particles with an angular direction outside the barrel region, are mainly triggered by the ECL trigger, the cluster energies and positions have to be converted from the laboratory frame to the CM frame. This is done by a Lookup Table (LUT), which, given the input TC ID, returns the CM values for θ and ϕ and a conversion factor for the energy. The values for the LUT are set by running a simulation for a 1 GeV γ in the lab frame and calculating the correct conversion factors for this cluster.

To reduce higher trigger rates given by beam background energy depositions, a lot of the current input bits only take TCs or clusters into account that are within a certain θ range. This is due to the fact that the TCs that lie closest to the beam pipe and therefore have a very low or very high θ angle, have the highest rate of beam background energy depositions due to their positioning. In Fig. 3.5 the corresponding θ IDs of each TC is shown. Input trigger bits like *ehigh* or *clst_X*, for example, remove the innermost ring of the forward endcap with θ ID $= 1$ and the backward endcap entirely, with θ ID 16 or 17.

3.2.4 Output Data

The output data of the ICN-ETM consists of general event information, clustering information, and calculated trigger bits. The ICN-ETM sends information both to the GDL and a subset of this information to the GRL. The information sent to both boards consists of the following:

- Active bit: if there is any TC hit in the ECL,
- Event timing information given by the highest-energetic TC timing,
- The number of clusters,
- The *bha_veto* input bit,
- All cluster information.

The GRL additionally receives information on which clusters have a CM energy above 1 and 2 GeV, and which clusters are part of the *bha_veto* determination.

Input Bit	Definition
ehigh	$E_{tot} > 1.0$ GeV, E_{tot} in θ ID = 2-15
elum	$E_{tot} > 3.0$ GeV, E_{tot} in θ ID = 2-15
clst_X	bit X ($X \in [0, 1, 2, 3]$) of the number of clusters in θ ID = 2-15
ecl_oflo	ICN hits > 6
bha_veto	$165^\circ < \sum\theta_{CM} < 190^\circ$, $160^\circ < \Delta\phi_{CM} < 200^\circ$, E(CL1, CM) > 3 GeV & E(CL2, CM) > 3 GeV, E(CL1, CM) > 4.5 GeV | E(CL2, CM) > 4.5 GeV
ecl_bhapur	$160^\circ < \sum\theta_{CM} < 200^\circ$, $140^\circ < \Delta\phi_{CM} < 220^\circ$, E(CL1, CM) > 2.5 GeV & E(CL2, CM) > 2.5 GeV, E(CL1, CM) > 4 GeV | E(CL2, CM) > 4 GeV
ecl_mumu	$165^\circ < \sum\theta_{CM} < 190^\circ$, $160^\circ < \Delta\phi_{CM} < 200^\circ$, (E(CL1, CM) < 2 GeV & E(CL2, CM) < 2 GeV)
ecltaub2b3	$140^\circ < \sum\theta_{CM} < 220^\circ$ & $120^\circ < \Delta\phi_{CM} < 240^\circ$ & (E(CL1, Lab) > 0.14 GeV | E(CL2, Lab) > 0.14 GeV) for 2 CL in θ ID = 2-16 with E(CL, Lab) > 0.12 GeV, no CL with E(CL, Lab) > 4.5 GeV, E_{tot}(Lab) < 7 GeV in θ ID = 1-17

Table 3.2: General Trigger Input Bits calculated on ICN-ETM. These bits are calculated using the clusters returned by the clustering algorithm or the energy information of the TCs. CM stands for Center-of-Mass frame, while Lab is using the laboratory frame.

The GDL receives additional timing information, specifying which detector part is the source of the event timing information. Furthermore, the input bits specified in Table 3.2 and Table 3.3 are sent to the GDL to calculate output bits and generate the trigger decision. The full number of TCs hit within one data window is also sent to the GDL to be used in the injection veto (see Section 4.1.3), combined with the number of wire hits in the CDC.

Input Bit	Definition
ecl_lml_0	N(CL) ≥ 3, at least 1 CL with E(Lab) $\geq$ 300 MeV with θ ID = 1-17, not bha_veto
ecl_lml_1	At least 1 CL with E(CM) $\geq$ 2 GeV with θ ID = 4-14
ecl_lml_2	At least 1 CL with E(CM) $\geq$ 2 GeV with θ ID = 2, 3, 15 or 16, not bha_veto
ecl_lml_3	At least 1 CL with E(CM) $\geq$ 2 GeV with θ ID = 2, 3, 15 or 16, bha_veto
ecl_lml_4	At least 1 CL with E(CM) $\geq$ 2 GeV with θ ID = 1 or 17, not bha_veto
ecl_lml_5	At least 1 CL with E(CM) $\geq$ 2 GeV with θ ID = 1 or 17, bha_veto
ecl_lml_6	Exact 1 CL with E(CM) $\geq$ 1 GeV with θ ID = 4-15, no other CL with E(Lab) $\geq$ 300 MeV with θ ID = 1-17
ecl_lml_7	Exact 1 CL with E(CM) $\geq$ 1 GeV with θ ID = 2, 3 or 16, no other CL with E(Lab) $\geq$ 300 MeV with θ ID = 1-17
ecl_lml_8	2 CL with $170^\circ < \Delta\phi_{\mathrm{CM}} < 190^\circ$, E(CL1, Lab) $>$ 250 MeV & E(CL2, Lab) $>$ 250 MeV, no CL with E(CM) $\geq$ 2 GeV with θ ID = 1-17
ecl_lml_9	2 CL with $170^\circ < \Delta\phi_{\mathrm{CM}} < 190^\circ$, E(CL1, Lab) $<$ 250 MeV & E(CL2, Lab) $>$ 250 MeV, no CL with E(CM) $\geq$ 2 GeV with θ ID = 1-17
ecl_lml_10	2 CL with $170^\circ < \Delta\phi_{\mathrm{CM}} < 190^\circ$, $160^\circ < \sum\theta_{\mathrm{CM}} < 200^\circ$, no CL with E(CM) $\geq$ 2 GeV with θ ID = 1-17
ecl_lml_12	(N(CL) ≥ 3, at least 1 CL with E(Lab) $\geq$ 500 MeV (with θ ID = 2-16), not bha_veto
ecl_lml_13	Exact 1 CL with E(CM) $\geq$ 0.5 GeV with θ ID = 6-11, no other CL with E(Lab) $\geq$ 300 MeV with θ ID = 1-17

Table 3.3: Low Multiplicity Trigger Input Bits calculated on ICN-ETM. Low multiplicity trigger bits are used to find events that do not result from the $e^+e^- \rightarrow \Upsilon(4S)$ decay. These bits are calculated using the clusters returned by the clustering algorithm. CM stands for Center-of-Mass frame, while Lab is using the laboratory frame.

Chapter 4

Monte Carlo Data Simulation and Preparation for Training

The training and evaluation of supervised machine learning algorithms require labeled data which ideally contains all possible signatures later present in collision data. For this, several technical datasets are simulated using the MC method. The overall simulation of the ECL and the L1 trigger in `basf2` is explained, along with the use of beam background samples to create a realistic scenario. The design of the technical datasets for training of the GNN-ETM, alongside the definition of training targets, is shown. Finally, the datasets used to evaluate the GNN-ETM performance are explained.

4.1 Simulation and Reconstruction

HEP generally has access to very good and fast simulations of both the particles' decays and interactions, as well as the detector response [45]. This means that the standard way of developing new tools and analyses is done using simulated MC data, while verification is then done on collision data. In this thesis, simulated datasets are used to train the network, provide a definition for the signal/background classifier, and evaluate the network's general performance and its performance on different processes.

The particle generation is simulated with different event generators (see Section 4.2), the interactions of the particles with material are simulated with `GEANT4`, and the detector response is simulated with `basf2`.

After a full simulation of a process in Belle II, the simulated samples are then reconstructed. Reconstruction is done in the same way for both collision data and MC data and transforms detector hits in the different subdetectors into reconstructed particle candidates. A few steps differ between collision and MC data, for example, the calibration of the detector response for collision data. These reconstructed particle candidates and their properties are then available for analysts, whereas the raw detector hits are not stored in the user-available format at Belle II. This means that for this thesis, no centrally-produced samples were available, as they do not contain the data objects necessary to build the trigger objects necessary for this study.

4.1.1 ECL Trigger Simulation

The simulation package for the Belle II trigger in `basf2` is called Trigger Simulation (TSIM) and it creates the trigger objects from the simulated detector responses, in the ECL case the ECL hits. These hits include the detector response for both the particles generated by the event generator and the energy depositions due to beam background. A further explanation of beam backgrounds at Belle II will be given in Section 4.1.3.

The ECL crystal detector response is simulated for a time of ± 4 µs with the simulated collision happening at t = 0 s. For the correct trigger simulation, the TCs have to be built out of the ECL hits. For this, all detector hits are first binned into bins of 200 ns. The timing of the detector hits includes the time of flight of the particle generating that hit, as well as the time needed to measure the electromagnetic shower at the photodiode. For each TC and for all timing bins, the energy of crystals belonging to that TC and that are within the respective timing bin is summed up:

$$E^{\mathrm{TC}}(T) = \sum_i E_i^{\mathrm{Crystal}}(T), \tag{4.1}$$

with $E^{\mathrm{TC}}(T)$ and $E_i^{\mathrm{Crystal}}(T)$ being the reconstructed energy of a TC and the i-th ECL crystal in that TC in the respective timing bin. The timing of the TC is derived by an energy-weighted average of the timing of the crystals within that TC:

$$T^{\mathrm{TC}}(T) = \frac{\sum_i T_i^{\mathrm{Crystal}}(T) \cdot E_i^{\mathrm{Crystal}}(T)}{\sum_j E_j^{\mathrm{Crystal}}(T)}, \tag{4.2}$$

with $T^{\mathrm{TC}}(T)$ and $T_i^{\mathrm{Crystal}}(T)$ being the reconstructed time of a TC and the i-th ECL crystal in that TC in the respective timing bin. For each TC in each timing bin, an approximation of the digitization process is performed and the fit of the digitized signal on the FAM is simulated. After this fit, the energy threshold of currently 100 MeV is applied. Because the energy cumulation and the digitization process are performed for each timing step separately, simulated TCs can appear multiple times in the resulting data array, if the crystals within one TC fall in different timing steps.

After this simulation, a timing window has to be chosen to emulate the 250 ns decision window of the real ICN-ETM. For this, the entire event time window of −4000 ns to 4000 ns is separated into 125 ns bins. Only three center bins around 0 are taken into account, with the middle bin starting at t = 0 ns. These three bins are then shifted to the left by a random time between 0 and 125 ns to simulate the offset of the timing window to the true event timing. The energy of all TCs within 250 ns from the start of the respective bins is then summed up and compared for the three bins. The starting time of the bin with the

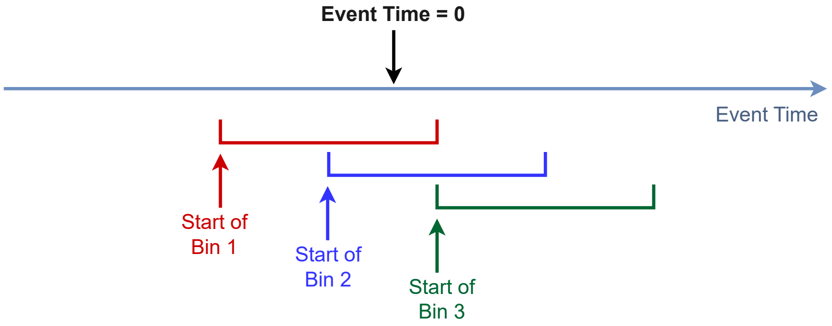


Figure 4.1: Schematic description of the trigger window selection in simulation. The start positions of the three bins are each 125 ns apart. The start of bin 2 is chosen to be the same timing as the event time = 0 and all three windows are then shifted to the left by a random time between 0 and 125 ns. Each window is 250 ns wide and the window with the highest energy sum of all TCs within that window is chosen.

highest energy sum is then the starting time of the simulated trigger window. A graphical description of this process can be seen in Fig. 4.1. Only the TCs with a timing within this timing window are then used for clustering and calculation of the trigger bits.

The ICN calculation, as well as the clustering and the trigger bit determination, is done with the same algorithms as described in Section 3.2. The same restrictions in terms of the number of clusters and the order in which they are taken from the detector hold for the simulation as well. For the conversion to the CM frame, the same LUT as the one used on the hardware is applied to mimic the behavior as closely as possible.

4.1.2 Reconstruction of Offline ECL Clusters

To be able to run the same evaluations on collision data and MC data and to reduce MC effects, the GNN-ETM is trained on offline ECL clusters. The reconstruction algorithm for the ECL is run on the HLT and in the processing of the full data and returns offline ECL clusters with their corresponding energy, timing [46], pulse-shape discrimination [35], if applicable, and various shower shapes to help the particle identification. The description of the reconstruction algorithm is based on [47] and [31].

The ECL reconstruction algorithm starts by building the so-called Connected Regions (CRs). For these CRs, all crystals with a reconstructed energy above an energy threshold, called seed threshold, are used as seeds for a CR. Starting from these seed crystals, the energy of the neighboring crystals of each seed crystal is checked. If the energy is higher than the growth energy threshold, this crystal is added to the CR. When encountering a crystal already belonging to another CR during that process, the two CRs are merged into one.

The purpose of building these CRs is that offline ECL clusters should always be contained in only one CR and not span several CRs, which simplifies the clustering algorithm in the next step. For the clustering, seeds for offline ECL clusters, so-called

Local Maximums (LMs), are identified in each CR. A LM is a crystal with a reconstructed energy above the LM threshold and with no direct neighboring crystal with a higher reconstructed energy. Each LM will become one offline ECL cluster with only rare exceptions. If a CR does not contain at least one LM, the highest-energetic crystal within that CR is set as an additional LM. If a CR contains more than 10 LMs, then only the 10 highest-energetic LMs are chosen as offline ECL cluster candidates and the remaining are removed from the list.

If a CR contains only one LM, the energy of the offline ECL cluster corresponding to this LM is determined by the sum of energies of the LM and a set number of neighbors around it. The optimal number of neighbors is determined by the region of the LM (forward endcap, barrel, backward endcap), the energy of the LM, and the background level in the event, choosing the number of crystals for the offline ECL cluster within the 5x5 grid around the LM. In most cases, all crystals in the 5x5 grid excluding the outermost corners, are used for the clustering.

If a CR contains more than one LM, the energy in each crystal can be split between several offline ECL cluster. The algorithm tries to split the offline ECL clusters by iteratively finding a stable center position for each offline ECL cluster, called a centroid (CD). As a first step, the centroids are assumed to be the LMs. All crystals within the same CR are assigned a weight for each centroid in this CR according to their energy and their Euclidean distance d to the centroid:

$$w_{\mathrm{i,CD}} = E(\mathrm{CD}) \cdot \frac{\exp(-C \cdot d(P(\mathrm{CD}), P_{\mathrm{i}})/R_{\mathrm{M}})}{\sum_{\mathrm{m}} E_{\mathrm{m}} \cdot \exp(-C \cdot d(P(\mathrm{CD}), P_{\mathrm{m}}))} \quad . \tag{4.3}$$

In this case, i denotes the crystal for which the weight is calculated, E and P are the energy and the position, respectively, for either the crystal or the centroid. C is a constant factor set to 2.5 and R_{M} is the Molière radius of CsI(Ti) crystals [48], which describes the transversal expansion of an electromagnetic shower within a material. The sum in the denominator is over all centroids within the CR.

For each offline ECL cluster, the new centroid position of the assumed offline ECL cluster, is calculated using the weights and positions of all crystals with a non-zero weight within the CR. This procedure is repeated until the positions of all centroids within the CR are stable within 1 mm.

There are two reasons a LM cannot become a offline ECL cluster in this algorithm. The first is if the position of the centroid has moved so much that it is now outside the original LM crystal. The second reason is that by weight splitting, a crystal, which is not the LM, has now more energy assigned to the hypothetical offline ECL cluster than the LM itself. In both cases, the LM is removed from the list of LMs. In this case, the algorithm repeats the full iterative centroid calculation again until no further LMs will be removed.

After this, the offline ECL cluster's energy and position are calculated by using the optimized number of crystals. The same algorithm as in the single LM case is used for determining the optimal number of crystals. The energy of the offline ECL cluster is then the sum of the products between the weight and the reconstructed energy of all crystals belonging to this LM and the position is the centroid position of the reduced number of crystals for this offline ECL cluster. The timing of an offline ECL cluster is the timing of the highest energetic crystal within that offline ECL cluster. The energy of the offline ECL cluster is then additionally corrected to account for longitudinal energy leakage (see Section 2.3).

In basf2, the offline ECL clusters found by the reconstruction algorithm so far are called ECL showers. For the actual offline ECL clusters available after reconstruction, two thresholds are applied. No shower below 20 MeV is kept. Additionally, if a shower is below 50 MeV, a timing threshold is applied. If the shower has timing outside the 99 % interval, that all showers within that energy range fall into, it is also not converted into offline ECL clusters. As for the ECL L1 trigger, we have an energy threshold of 100 MeV per TC, these cuts do not make an impact.

For the current standard Belle II reconstruction, the energy thresholds used are 10 MeV for the LM, seed and growth threshold. The current reconstruction is optimized for high efficiency with moderate beam backgrounds and has performed very well for past data-taking periods. However, with very high beam background levels, these thresholds lead to very large CRs spanning up to 3000 crystals. Due to this, the iterative centroid calculation algorithm becomes unstable, as every crystal within one CR is used for the calculation of every LM in that CR. In Fig. 4.2, the same event reconstructed with 10 MeV and 20 MeV for all three thresholds is seen. Especially in the backward endcap for the 10 MeV reconstruction, many LMs are not converted into showers due to the algorithmic instability in finding a stable centroid position. Even high-energetic LMs are removed in this case. For future uses, based on findings during this thesis, the official ECL reconstruction will be adapted to include a distance threshold, increase the energy thresholds, and allow the centroid positions to move outside the original LM. For this thesis, I increased the energy for all three thresholds to 20 MeV. This greatly reduces the size of the CRs in high beam background events, while not removing any offline ECL clusters of interest for the trigger as we apply the energy threshold of 100 MeV for each TC on trigger level anyway.

4.1.3 Beam Backgrounds at Belle II

Currently, one of the main challenges at the Belle II experiment is the increasing beam background rate when going towards higher instantaneous luminosity at the SuperKEKB accelerator. Beam background originates from beam particles that do not follow the nominal orbit and can therefore hit the beam pipe or other structures. In doing so, they generate shower particles which, if the original loss happened close to the Belle II detector, can travel into the active detector region and induce hits within the different subdetectors. These hits can mimic signatures coming from the actual collision at the interaction point and lead to

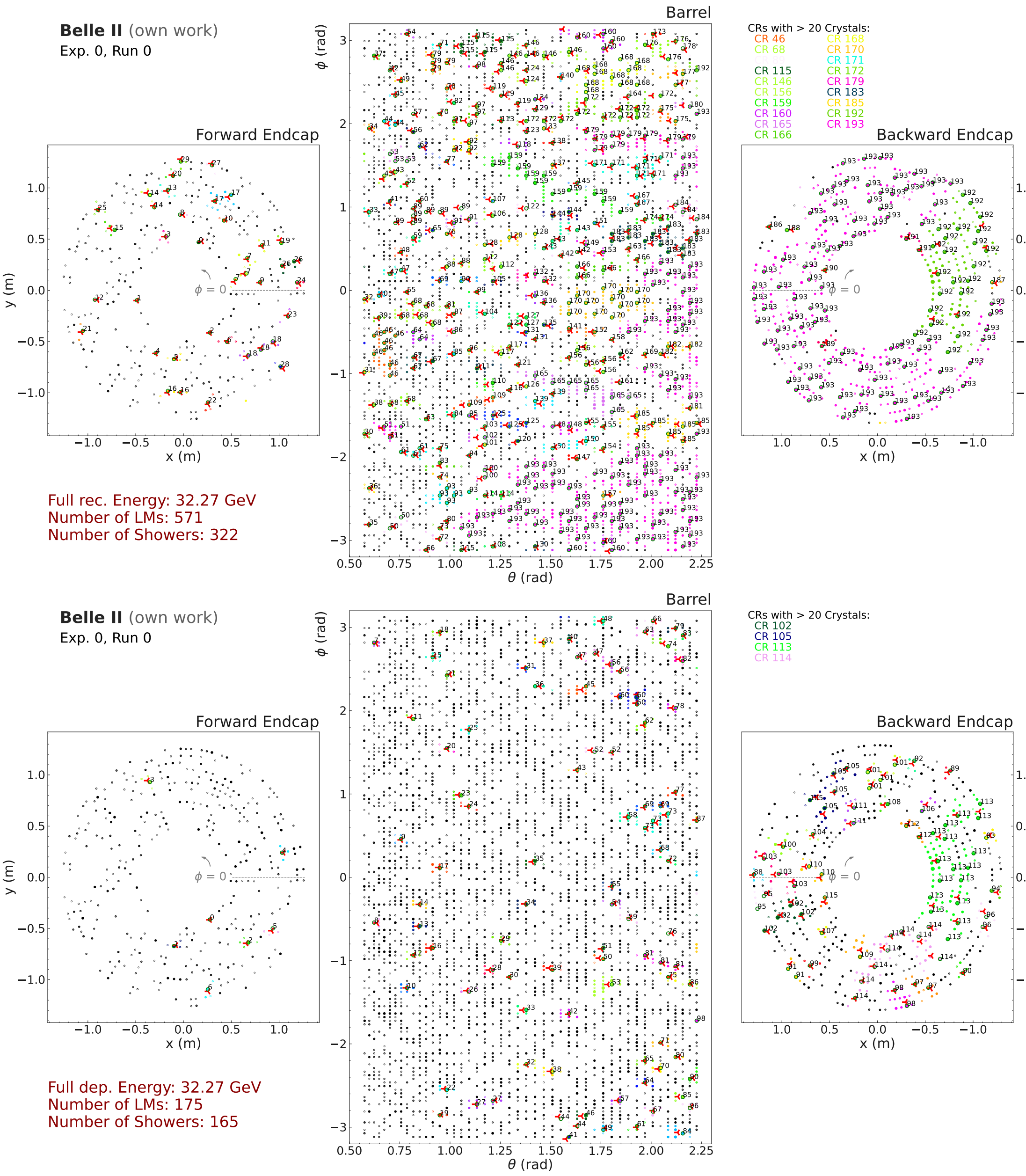


Figure 4.2: Crystal energy depositions with reconstruction objects for one event with Exp. 0 beam background. All crystals with a reconstructed energy > 1 MeV are shown with their size proportional to the reconstructed energy. The colors of the crystals show their CR. Crystals with a green circle are LMs, the red triangular shapes show the positions of reconstructed ECL showers. Each LM has the ID of its CR annotated as well. The upper plot is reconstructed with 10 MeV energy thresholds, the lower plot with 20 MeV.

either a wrong interpretation of the event or a worsening of the correct reconstruction due to smearing of the energy or tracking resolution, for example.

The measured beam background in the detector is a result of several background components, where some, but not all, are proportional to the luminosity. The main sources of beam background in Belle II [49] are explained in the following.

Touschek Background The Touschek effect is the effect of an exchange between transverse and longitudinal momentum of a particle due to the Coulomb scattering of two particles within the same bunch. When two particles scatter, one of them loses energy while the other gains energy, making both of them deviate from the nominal beam energy. The nano-beam scheme at SuperKEKB reduces the beam size and therefore increases the rate of Touschek scattering, which is inversely proportional to the beam size.

Beam-gas Background As the vacuum within the beam pipe is not perfect, beam particles can either interact via Coulomb scattering with Beam-gas particles, thereby changing direction, or emit Bremsstrahlung in the presence of Beam-gas particles, which results in an energy loss of the beam particles.

Luminosity Background This background source results from beam collisions at the IP and is therefore proportional to the luminosity. One major source is radiative Bhabha scattering, $e^+e^- \rightarrow e^+e^-\gamma$, where the beam particles lose energy by emitting photons. These beam particles can then be lost inside the detector. Additionally, photons emitted along the beam pipe can travel further outside Belle II and hit the accelerator magnets downstream of the IP. This interaction can then result in neutrons via the giant photonuclear resonance, which can be scattered back towards the detector. These neutrons then hit the outer KLM and increase the hit occupancy there. Another source can be $e^+e^- \rightarrow e^+e^-e^+e^-$, where the beam particles lose energy by emitting an electron-positron pair. In addition to the beam particles becoming a source of background, the emitted charged particles can curl within the magnetic field and leave hits in the tracking detectors.

Synchrotron Background The synchrotron radiation, especially emitted by the HER electron beam due to its higher energy, is a background source especially for the inner tracking detectors. The synchrotron photons can leave hits in both PXD and SVD.

Injection Background So-called top-up injections are performed for both the LER and HER beams in a continuous injection scheme. This injection is done at a maximum of 25 MHz. Bunches that have been newly injected oscillate around the main beam for a short time after injection. This increases the beam background rate by a high factor. Therefore, a special injection veto is applied by the trigger to veto all events that would pass the trigger for several milliseconds after injection. This avoids Data Acquisition (DAQ) dead time and read-out saturation. While for past data-taking periods, a so-called passive injection veto has been used, which vetoed a set time after injection, in the most recent data-taking period an active injection veto was tested. This relies on measurements of the CDC and ECL trigger to gauge the amount of beam background and reduce the veto time.

Beam Background Overlays for MC Simulation

For simulated samples, beam background hits have to be included to correctly model the experimental conditions at the Belle II detector. For this, basf2 can be provided with so-called beam background overlays. These are either simulated beam background overlays, called run-independent (ri) background samples, or overlays taken from data, called run-dependent (rd) background samples.

For samples taken from data, a dedicated trigger is employed to get the rd background overlays. As this trigger should be proportional to luminosity, a delayed Bhabha trigger bit is used. This trigger bit fires after a set time after the Bhabha trigger bit has triggered. The delay time is set to the time it takes the bunch, which has been at the IP at the time of the Bhabha trigger, exactly five revolutions around the accelerator ring. These triggered events are then stored with full detector information, including full waveform data for all ECL crystals. This information is needed for their usage as background overlays. Due to this, these events are much bigger in terms of data size and the total output rate of the delayed Bhabha trigger has to be limited to about 5 Hz.

The beam background files for rd overlays are available for every run in Belle II. For analyses, the MC datasets are usually simulated for every run with the corresponding rd beam background overlays.

For ri samples, the beam background simulation is done using both the Strategic Accelerator Design (SAD) framework [50] to simulate the Beam-gas interactions, both Coulomb scattering and Bremsstrahlung in the presence of Beam-gas particles, and Touschek scattering, while the luminosity background is simulated using event generators within basf2. For the luminosity background, radiative Bhabha events using the event generators BBBREM [51] and BHWIDE [52] and di-photon processes using AAFH [53] are simulated. In the case of the SAD simulation, the framework is initialized with beam optics parameters and machine apertures and tracks beam particles around the ring for 1000 machine turns. The beam currents and the beam-gas pressure are also input to the framework. For the simulation for Belle II, a refined description of the collimators is added as well as a new simulation of the beam particle interaction with the collimator materials. Additionally, the SAD to `GEANT4` interface has been improved to correctly model the curvature of beam pipes and the collimator shapes. The simulated region, in which lost particles are passed to `GEANT4`, is extended up to 30 m outside the IP along the beampipe.

All simulated components are then mixed to produce the ri background overlays. The mixing factors, which determine the relative strength of each background component in the overall beam background, are tuned to optimize data/MC agreement [49]. Belle II provides different ri background samples for different detector and accelerator conditions. The sample for Exp. 1003 uses the detector and accelerator conditions in the data-taking period between 2019–2022, before LS1. These conditions are also called low beam background, as the average occupancy of the sub-detectors was significantly lower than after LS1 due to the lower instantaneous luminosity.Exp. 0 is used for the extrapolated accelerator and detector conditions after LS2, and uses a much higher instantaneous luminosity at $6 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ and therefore significantly higher beam background than current conditions and especially Exp. 1003.

Effects of Beam Background and Measurements of Beam Background Levels

Most reconstruction algorithms are dependent on the amount of beam background visible in the detector, as for example tracking efficiency, photon energy resolutions and trigger efficiencies suffer under high beam background conditions. For this, beam background can be monitored with several variables, the two most common within Belle II are the number of out-of-time crystals in the ECL and the number of offline ECL clusters in the ECL. Out-of-time crystals are all crystals that have a timing $|t| > 110\,\mathrm{ns}$ relative to the event timing and an energy above $7\,\mathrm{MeV}$.

In Fig. 4.3 a comparison for out-of-time crystals and number of offline ECL clusters for five different simulated samples with 10000 events each can be seen. All detector hits can be attributed to the different beam background overlays used. Four samples with rd overlays from different runs are shown, and one sample with ri overlays from Exp. 1003. The Exp. 22, Run 26 data was taken at the beginning of Belle II data-taking in 2021, Exp. 26, Runs 898 and 1485 were taken at the end of the first full data-taking period in 2022, and Exp. 35, Run 2902 was taken in December 2024. The beam background conditions rise significantly with later runs, due to higher instantaneous luminosities.

The simulated beam backgrounds of Exp. 1003 have similar hit occupancies in the ECL as Exp. 26, which has medium to high beam background conditions.

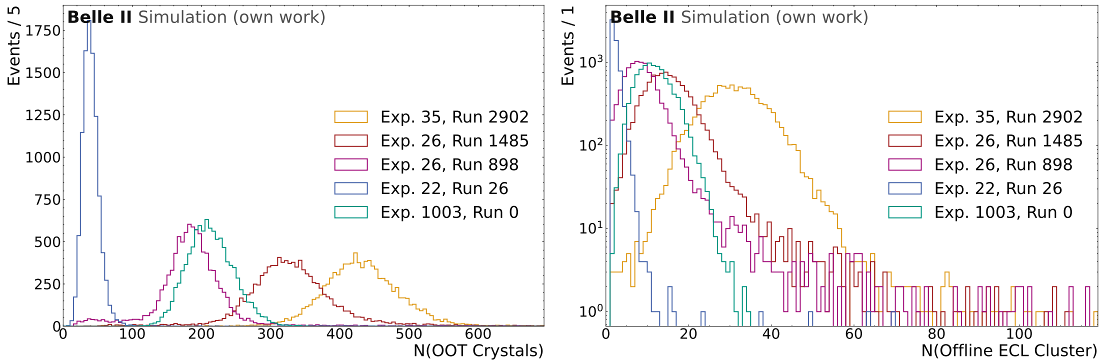


Figure 4.3: Number of out-of-time crystals and offline ECL clusters for simulated samples with different background overlays. The amount of beam background rises with later runs due to higher luminosities.

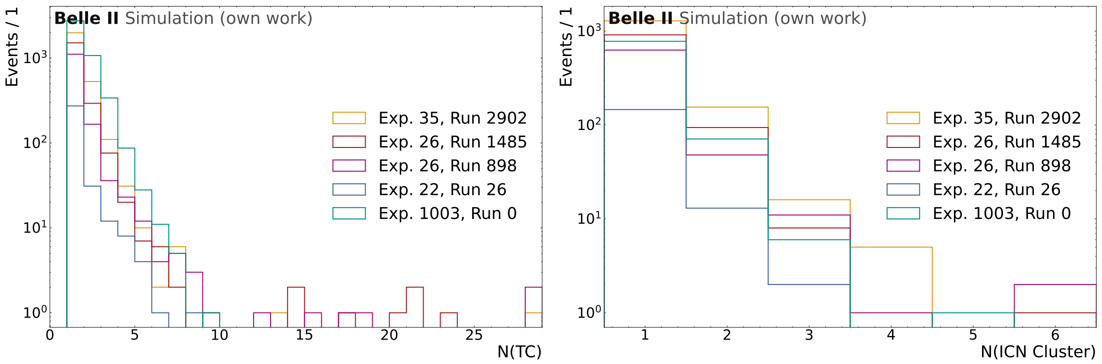


Figure 4.4: Number of TCs and ICN-ETM clusters for simulated samples with different background overlays. The amount of beam background increases in later runs due to higher instantaneous luminosities.

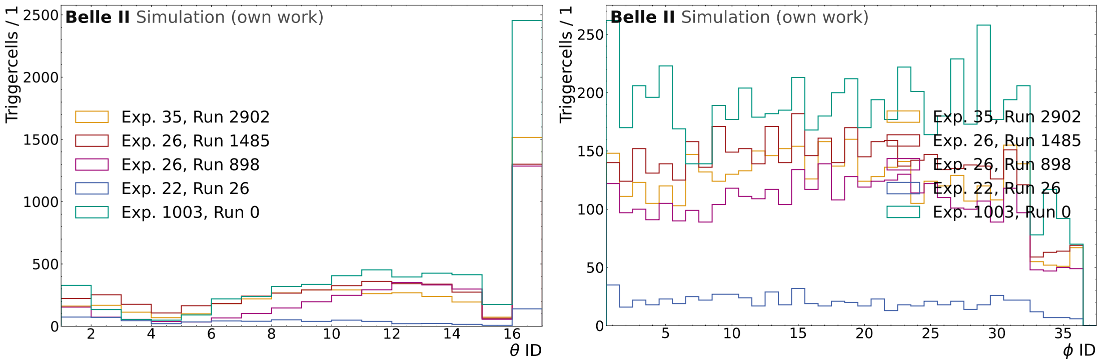


Figure 4.5: Number of TCs per θ ID (left) and ϕ ID (right). The rightmost ϕ IDs are only present in the barrel region.

Additionally, the number of TCs and ICN-ETM clusters can be seen in Fig. 4.4 for the same runs. These "fake" clusters can mimic depositions made by particles originating from the collision and worsen purities or increase trigger rates. As the ICN-ETM does not distinguish between signal or background clusters, rising rates of clusters due to beam background increase the probability of events being triggered.

To see the impact of beam background in the different detector regions, Fig. 4.5 shows the number of TCs per θ and ϕ ID. The θ ID distribution shows the large percentage of beam background energy depositions in the backward endcap, specifically in the inner ring, which corresponds to θ ID $= 16$. In general, the number of background energy depositions rises towards the backward endcap. The distribution in ϕ is more uniform. The significantly lower contributions in ϕ IDs above 33 is due to the fact, that this ϕ IDs only appear in the barrel region as the endcaps have less TCs. This is also the reason for the visible repeating peaks, which correspond to the ϕ IDs existing in both barrel and the endcaps.

To classify the impact of beam background for the overall trigger, trigger rates are estimated for different beam background conditions. For an approximate calculation of trigger rates due to beam background, the presence of beam background is assumed to be

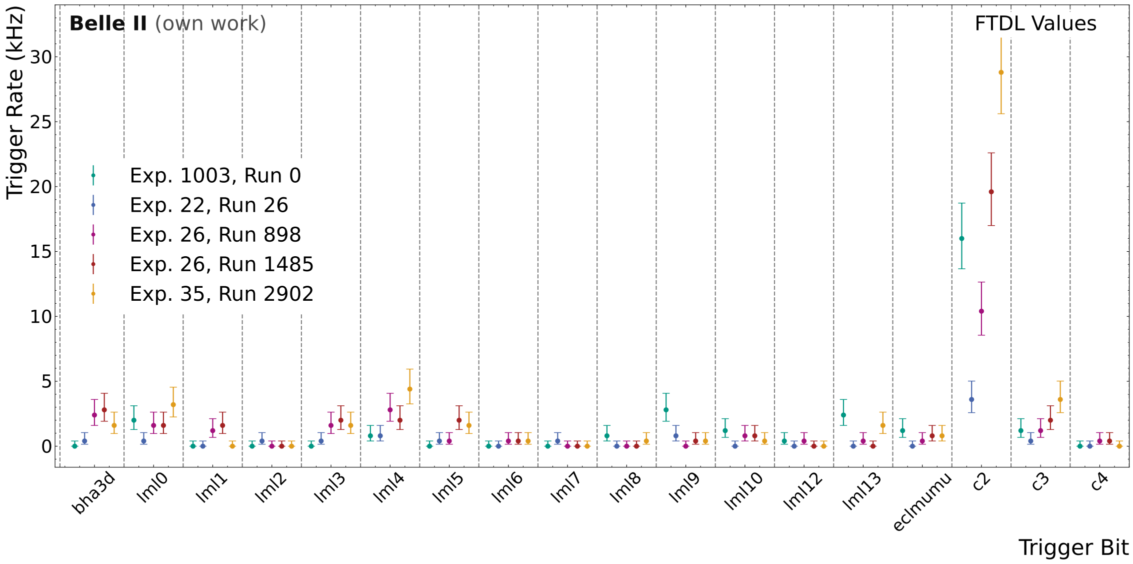


Figure 4.6: Estimated trigger rates for simulated samples with different background overlays for pure ECL trigger bits. The plot shows the FTDL trigger rate, which is the non-prescaled rate for the corresponding trigger bits.

constant at all times instead of using a cross-section as in the case of a specific process. The rate of a trigger bit for beam background can therefore be calculated by

$$R_{\mathrm{bit}} = \frac{\mathrm{N(bit = 1)}}{\mathrm{N(all)}} \cdot \frac{1}{250\,\mathrm{ns}}, \tag{4.4}$$

where N is the number of simulated events, bit is the chosen trigger bit and 1 / 250 ns is the time frame of one trigger window. As described in Section 3.2, in the actual implementation two adjacent windows cannot both generate a trigger signal, but this can still be taken as an estimate. In Fig. 4.6, an estimation of the trigger rate due to the presence of only beam background is shown for the different samples. This figure visualizes two things: first, that the Exp. 1003 ri background simulation is a good approximation for past background conditions in mid-2022, outside of runs with very high backgrounds. Two, that the trigger rates rise significantly already with beam background levels in the last runs of Belle II. If increasing the luminosity by an order of magnitude more, the overall rate will quickly exceed the maximum allowed rate of 30 kHz. Due to this, the development of adjustable or robust reconstruction algorithms in changing beam background conditions is necessary for the Belle II experiment.

Beam Background Overlays for Training and Evaluation Datasets

For MC simulation, the detector hits originating from the simulated signal particles are overlaid with hits coming from either ri or rd background overlays for all subdetectors. The overlay strategy differs for each subdetector.

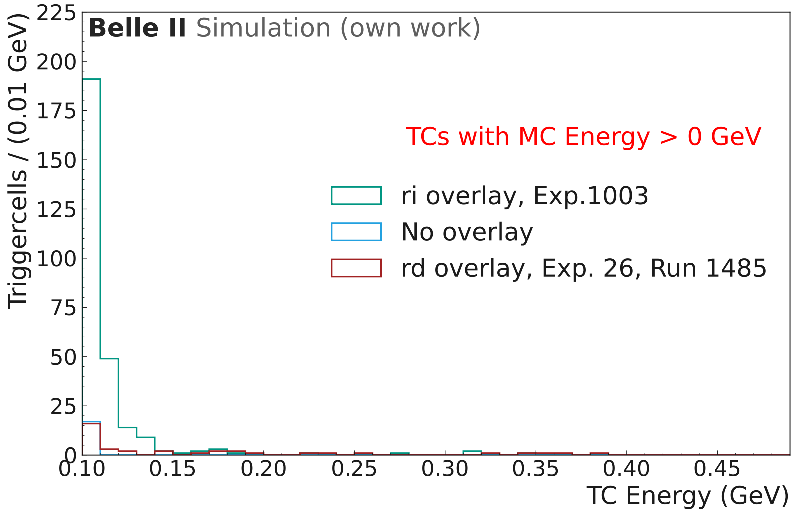


Figure 4.7: Number of TCs with MC energy deposition from an 80 MeV photon in 30000 events. The number of TCs with MC energy deposition is shown for events with ri overlays, rd overlays from Exp. 26, Run 1485, and no overlays.

For rd background, the information is directly taken from the measured TCs in data and not reconstructed from ECL crystals. As the energy threshold for the TCs is set as a hardware threshold, this information already has the 100 MeV threshold applied. While this is, of course, an accurate modeling of the detector response for the randomly triggered events, this underestimates the actual amount of background energy in the calorimeter, if used as background overlays. This is due to the fact that for rd background overlay the energy threshold of 100 MeV is applied separately on signal and on background energy depositions due to the usage of the measured TCs from data. In Fig. 4.7 an example comparison of the number of TCs with MC energy depositions for 30 000 MC events with a single 80 MeV photon with ri background overlay, rd background overlay and no background overlay is shown. For ri background overlays, if the MC photon deposits its entire energy with 80 MeV in one TC, 21 MeV beam background energy deposition is enough to surpass the 100 MeV cut. In comparison, for rd background overlays either the MC energy deposition or the background energy deposition separately have to exceed 100 MeV.

As the presence and possible increase of beam background energy depositions in the ECL are key reasons for developing this algorithm, an underestimation of beam background in simulation can significantly impact the algorithm's performance in real data-taking. The background overlays are centrally produced from events triggered by the delayed Bhabha trigger within Belle II, which means that a reproduction of rd beam backgrounds to reconstruct the TCs from ECL crystals instead of taking them from data was not feasible within the scope of this thesis. Additionally, as can be seen in Section 4.1.3, the Exp. 1003 ri background overlays approximate beam background conditions of the data-taking period in 2022 well and can therefore be used in the context of this study.

4.2 Training Inputs and Targets

Training the network and performance evaluation of the different trainings is done on both "technical" simulated datasets, meaning that they do not obey any conservation laws or follow decay chains, and on datasets consisting of simulated processes $e^+e^- \rightarrow e^+e^-(\gamma)$ or $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$. This section describes the inputs to the network, the definition of truth labels and the selection of events for training. The technical datasets are then described in Section 4.3, while the physics evaluation datasets are described in Section 4.4.

4.2.1 Inputs

Input nodes for the network are all TCs that have an energy deposition above 100 MeV and are within the timing window of 250 ns of the trigger decision. These TCs are taken from the basf2 TSIM as well as the start of the timing window, which is determined by basf2 as described in Section 4.1.1. For each TC, I read out the TC ID, the energy in GeV and the timing in µs. As the absolute timing is arbitrary and not comparable for MC and data, I use the difference of the TC time to the highest energetic TC within the timing window as an input. Using a set LUT for the center position of each TC, I map each TC with its TC ID to its Cartesian position (x,y,z) in m. The input variables and their approximate range can be seen in Table 4.1. In general, the input variables are not normalized or scaled to any range.

As explained in Chapter 6, due to hardware constraints, the network is restricted to 32 input TCs. If a timing window contains more than 32 TCs, the input is truncated at 32. In Fig. 4.8, the number of TCs per trigger decision window (see Section 3.2) is shown for Exp. 35, Runs 2882 - 2896, for events with at least one TC above 100 MeV, corresponding to 2596050 events in total. The threshold of 32 TCs is shown. Very few events are above this threshold, even though these runs have high beam background levels.

The ordering of the TCs for the GNN-ETM is arbitrary, which results in an arbitrary cut-off in the case of too many TCs per timing window. An ordering based on more relevant detector regions is a possibility for future work, but has to be additionally feasible for the hardware implementation.

Table 4.1: Input variables for each TC to the network. These variables are available in both MC data and collision data.

Input variable	approx. Range	Unit
Reconstructed Energy	[0, 12]	GeV
Relative Time	[-0.25, 0.25]	m s
x Position	[-1.27, 1.27]	m
y Position	[-1.27, 1.27]	m
z Position	[-1.05, 1.97]	m

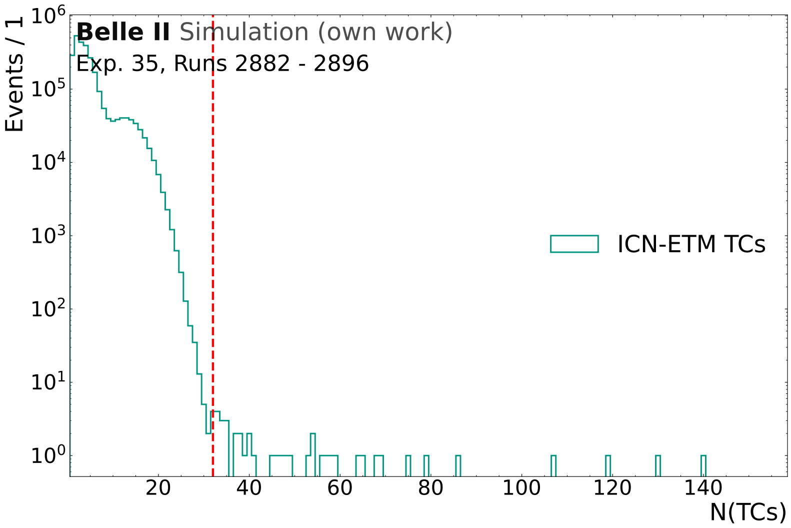


Figure 4.8: Number of TCs in the trigger decision window per event for collision data taken in December 2025. Only events are shown that have at least 1 TC over 100 MeV. The red line corresponds to the cutoff of 32 TCs.

4.2.2 Training Targets

As I use a supervised learning strategy for this work, correct training targets (also called labels) are required. The GNN-ETM is a node regression network, where each input TC is a node in a graph and will have several prediction values given by the network. Each node receives one truth value for all variables which should be predicted by the network. To reduce the dependency on MC truth values, the network is trained on the offline ECL clusters given by the offline reconstruction (see Section 4.1.2). For this thesis, the network output consists of the object condensation targets β and the latent coordinates, explained in Section 6.2.1, and the offline ECL cluster targets. The offline ECL cluster targets are the offline ECL cluster energy in GeV, the position in m and a signal/background label. offline ECL cluster energy and position are taken from the offline ECL clusters.

Signal/Background Label

One network output value is a signal/background classifier label between 0, meaning background, and 1, meaning signal. This can help in trigger decisions to determine if an event should be kept or if the clusters are mainly caused by beam background. Therefore, every offline ECL cluster target has to have a signal or background label, which the network can then attempt to learn. offline ECL clusters should be classified as signal if they are caused by a particle that was produced in the original collision or decay products of these particles and as background if they originate from beam background.

For this study, the offline ECL showers are used to avoid the cut for low-energy offline ECL clusters. This has no impact on the results and will be used interchangeably.

In MC data, for each offline ECL cluster, the percentage of signal energy deposition is calculated as well, to determine the true signal/background label. Two values are relevant in this context, the `MC Ratio`, which is the ratio of MC energy deposited in the shower by a given MC particle over the full energy of the MC particle and the `Cluster Ratio`, which is the ratio of the deposited MC energy in the offline ECL cluster over the full offline ECL cluster energy:

$$\text{MC Ratio} = \frac{E^{\text{Cluster}}_{\text{MCParticle}}}{E_{\text{MC Particle}}} \quad \text{and} \quad \text{Cluster Ratio} = \frac{E^{\text{Cluster}}_{\text{MCParticler}}}{E^{\text{Cluster}}} \quad , \tag{4.5}$$

with $E^{\text{Cluster}}_{\text{MCParticle}}$ being the energy of a MC particle deposited in a offline ECL cluster, E^{Cluster} the offline ECL cluster energy and $E_{\text{MCParticle}}$ the generated MC particle energy. These values can be calculated for each MC particle and for each offline ECL cluster. In the standard MC matching of basf2, a offline ECL cluster is then matched to an MC particle if the `MC Ratio` exceeds 0.3 and the `Cluster Ratio` exceeds 0.2.

In Fig. 4.9, the distribution of the `Cluster Ratio` over the `MC Ratio` for 11000 events from the technical category-one dataset described in Section 4.3.1 is shown. Each offline ECL cluster is only shown once, even if it has energy depositions from different MC particles. The `Cluster Ratio` and `MC Ratio` is then shown for the MC particle with the highest `MC Ratio`. The four plots depict the different detector regions, where the offline ECL cluster are categorized in by their reconstructed θ angle. The red dotted lines show the basf2 MC matching cuts of 0.3 for the `MC Ratio` and 0.2 for the `Cluster Ratio`. The offline ECL clusters that would be MC matched in basf2 are located in Region II. As shown in Fig. 4.9, offline ECL clusters located in the barrel are primarily in Region II, with over 99.6 % of all offline ECL clusters. These offline ECL clusters are usually signatures where a single photon has hit the ECL directly and deposited most of its energy directly at its hit location. These are classified as signal.

For all three detector regions, a higher amount of entries can also be found in Region IV, at (0.0, 0.0). These offline ECL clusters are "true" background, without any MC energy deposition, and are therefore classified as background. A rather small number of entries are in Region III. These usually occur, when a low-energetic MC particle has deposited a high percentage but a small absolute amount of its MC energy in a offline ECL cluster. This offline ECL cluster is then either a higher-energetic true background offline ECL cluster or a offline ECL cluster with a higher amount of absolute MC energy from another MC particle, which has not deposited a high percentage of its total energy in this cluster. In the second case, these offline ECL clusters are in Region III due to the sorting by highest `MC Ratio`. offline ECL clusters in either Region III or Region IV are classified as background.

A non-negligible percentage of offline ECL clusters, especially located in the forward or backward endcap, are sorted into Region I. offline ECL clusters in these regions have almost always a `Cluster Ratio` close to 1. In the standard definition of basf2, these would not be classified as signal, even though the energy in these offline ECL clusters is almost exclusively deposited by MC particles.

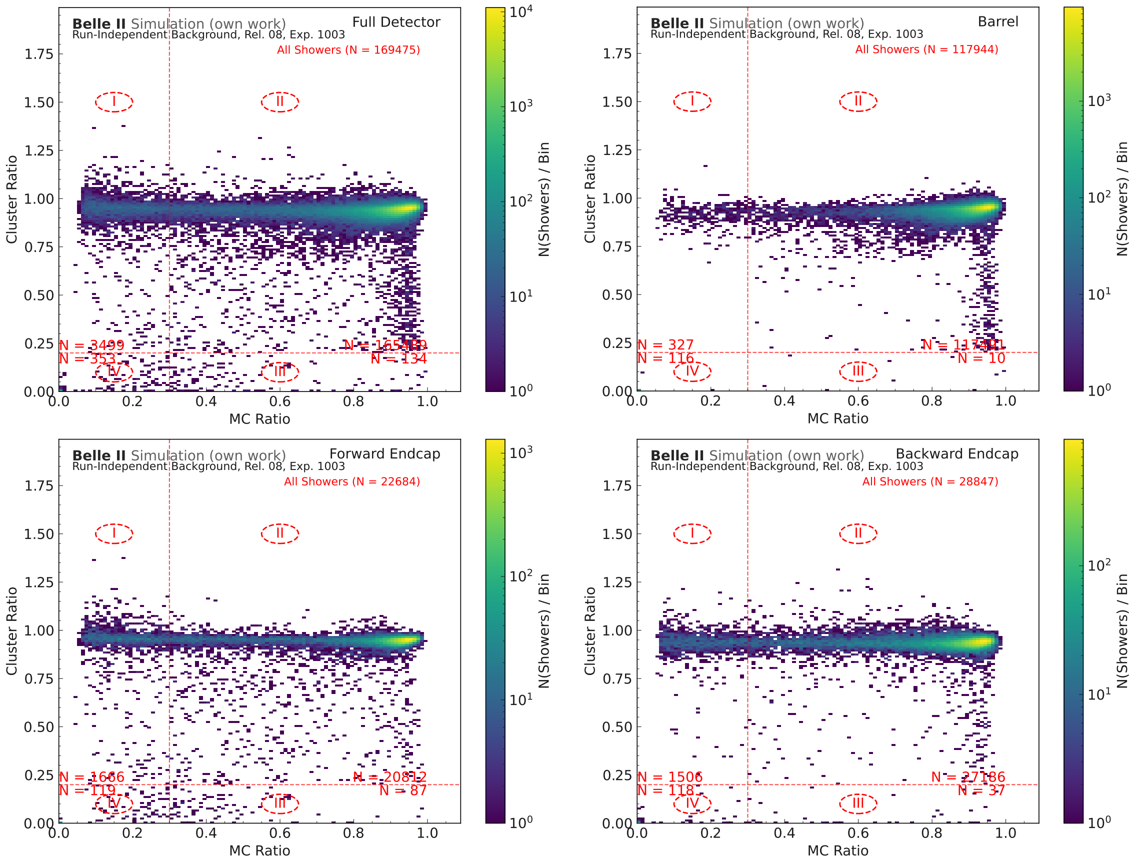


Figure 4.9: `Cluster Ratio` over `MC Ratio` for offline ECL showers with energy depositions from particles originating from 11000 events of the technical category-one sample (see Section 4.3.1). The top left plot shows the distribution for offline ECL cluster ECL showers in the full ECL detector, top right for the barrel part, lower left for the forward endcap, and lower right for the backward endcap. The red lines indicate the basf2 standard cuts of 0.2 and 0.3 to label a offline ECL shower as MC matched.

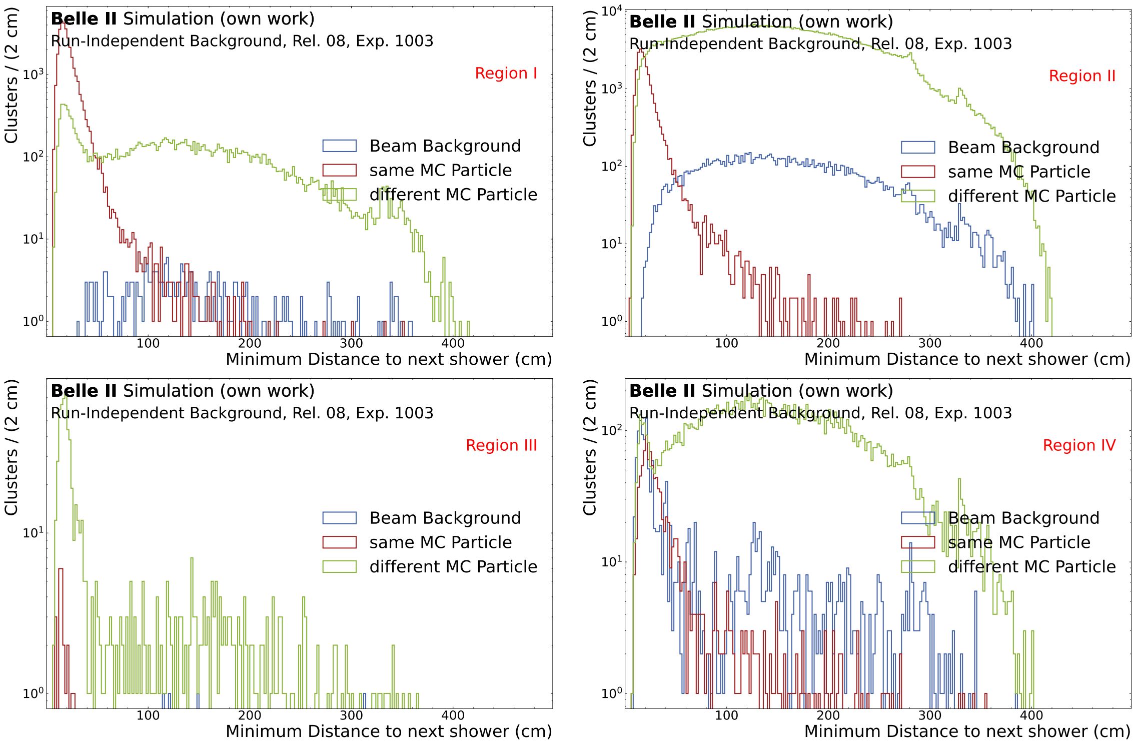


Figure 4.10: Euclidean distance of all offline ECL clusters to the closest other offline ECL cluster for all four regions of cluster MC matching for the full ECL detector. The colors indicate if the closest other offline ECL cluster has the highest `MC Ratio` to the same MC particle as the tested offline ECL cluster, to a different MC particle or to no MC particle, which tags it as a beam background offline ECL cluster. On the top left the tested offline ECL clusters are in Region I, top right in Region II, in lower left in Region III and lower right in Region IV.

The reason for the low `MC Ratio` of these clusters can be seen in Fig. 4.10. In these plots, the Euclidean distance for all offline ECL clusters to its nearest neighbor is shown. The tested offline ECL clusters are separated into the four regions for the four plots, while the closest offline ECL cluster can be in any region. The distributions are separated into three categories: 1, if the MC particle with the highest `MC Ratio` for the closest other offline ECL cluster is the same MC particle as in the original offline ECL cluster; 2, if the MC particle with the highest `MC Ratio` for the closest other offline ECL cluster is a different MC particle as in the original offline ECL cluster; and 3, if the closest other offline ECL cluster does not contain any MC particle energy depositions. As can be seen in the top left plot in Fig. 4.10, the offline ECL clusters in Region I are predominantly close to a offline ECL cluster assigned to the same MC particle. This is not visible in other regions, indicating a splitting of one particle's energy into several offline ECL clusters. Due to this, each of the offline ECL clusters from the same particle is not reaching the 0.3 threshold of the `MC Ratio`, while still having a majority of the total energy in that offline ECL cluster originating from an MC particle.

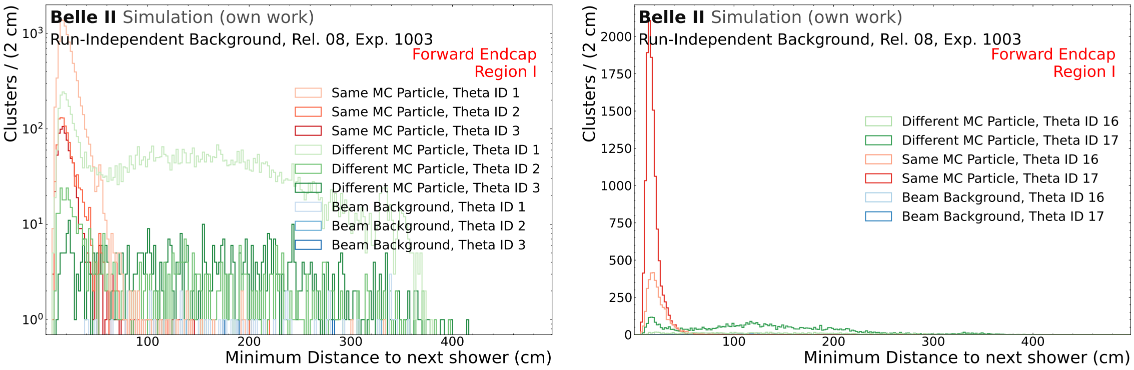


Figure 4.11: Euclidean distance of all offline ECL clusters to the closest other offline ECL cluster for Region I of cluster MC matching for the forward endcap (left) and the backward endcap (right). The colors indicate if the closest other offline ECL cluster has the highest `MC Ratio` to the same MC particle as the tested offline ECL cluster, to a different MC particle or to no MC particle, which tags it as beam background. Additionally, the distributions are split up into the trigger θ ID of the position of the tested offline ECL cluster.

One main factor of this effect is the interaction of photons with detector material at the edges of the detector. In Fig. 4.11, the distance to the next offline ECL cluster for all offline ECL clusters in the forward endcap (left plot) and the backward endcap (right plot) is shown. The distributions are again split up by the different categories of the closest offline ECL cluster. Additionally, a split into the TC θ IDs of the position of the tested offline ECL cluster is done (see θ IDs in Fig. 3.5). Most of the offline ECL clusters are located in the innermost θ ring in both endcaps, which corresponds to TC θ ID 1 in the forward endcap and TC θ ID 17 in the backward endcap. This indicates that photons interact with the material between the beam pipe and the inner rings of the ECL. Due to this material interaction, the electromagnetic shower started by the photons can already begin outside the ECL instead of in the crystals. This leads to a much wider shower spread and to a reconstruction of multiple offline ECL clusters instead of a single offline ECL cluster for the original photon.

Nevertheless, these offline ECL clusters are products of MC particles originating from the primary interaction. For the network training, these should be tagged as signal. A further argument is the analysis of offline ECL clusters from other MC particles than photons. In Fig. 4.12, the distribution of `MC Ratio` and `Cluster Ratio` in 10000 $e^+e^- \to B^0\overline{B}^0$ decays for different primary MC particles is shown. The particles, which deposited energy in the offline ECL cluster, are again sorted by `MC Ratio`, and only the particle with the highest `MC Ratio` is kept. If this particle is a primary MC photon, electron, muon, or pion, the offline ECL cluster is shown in one of the plots. On the top left plot, the distribution for primary photons is shown. As primary photons in $e^+e^- \to B^0\overline{B}^0$ decays mainly originate from neutral pion decays, the energy and position distribution is very different in comparison to the technical sample. Additionally, the offline ECL clusters

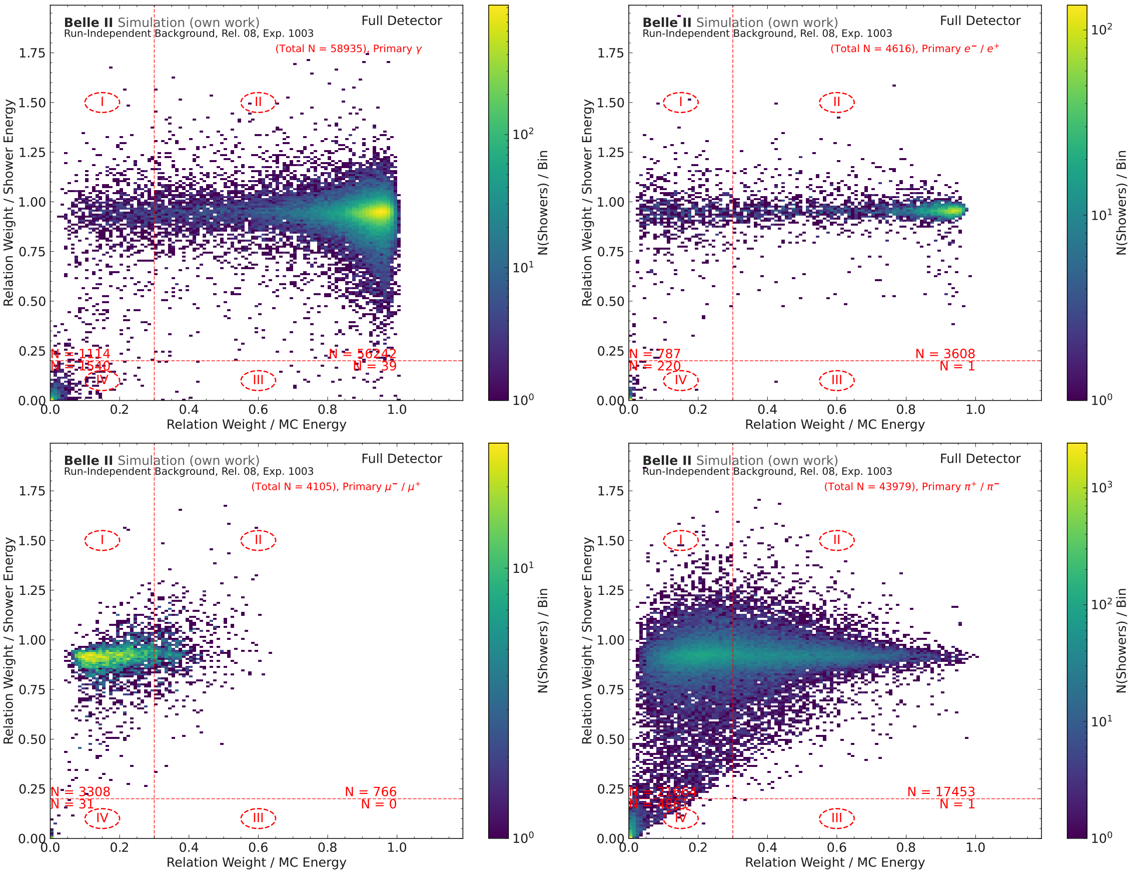


Figure 4.12: `Cluster Ratio` (y-Axis) over `MC Ratio` (x-Axis) for offline ECL clusters with energy depositions from primary particles originating from 10000 $e^+e^- \rightarrow B^0\overline{B}^0$ decays. One offline ECL cluster only appears in a single plot, as it is assigned to the particle with the highest `MC Ratio`. The top left plot shows the distribution for primary photons, top right for primary electrons, lower left for primary muons and lower right for primary pions. The red lines indicate the basf2 standard cuts of 0.2 and 0.3 to label a offline ECL cluster as MC matched.

of these photons often overlap, which can explain the wide spread of the distribution in y for values close to `MC Ratio` == 1. The offline ECL clusters assigned to primary electrons (upper right plot) behave very similar to the photon distributions of the technical sample. In comparison to that, offline ECL clusters assigned to primary muons and pions are very often located in Region I. This is due to the fact that muons as minimum ionizing particles in most cases only deposit 200 MeV in the ECL, regardless of their total energy. Pions as hadronic particles interact hadronically with the ECL and also only deposit a fraction of their total energy in the ECL. In both cases, offline ECL clusters in Region I should still be classified as signal, as in the ECL L1 trigger we do not have access to tracking or KLM information at the time of clustering.

As a result, for this work, offline ECL clusters in Region I and Region II are classified as signal, regardless of their `MC Ratio`.

Building of Target Clusters

For each TC, the corresponding target cluster is the cluster, whose energy is the highest percentage of energy within the TC. As described in Section 4.1.2, the energy of an ECL cluster is calculated by summing up the weights of the crystals belonging to that cluster. To now determine the percentage of energy of a given cluster within a TC, I build `Pseudo TCs` out of the ECL crystal energy depositions. This is necessary as the TSIM TCs do not contain the reconstruction information of the underlying ECL crystals.

Each ECL crystal has several energy values, the reconstructed energy $\mathrm{E}_{\mathrm{rec}}$ in this crystal, the partial energy E_{i} belonging to the i-th offline ECL cluster in this crystal, and the background energy $\mathrm{E}_{\mathrm{bkg}}$, which is calculated by

$$\mathrm{E}_{\mathrm{bkg}} = \mathrm{E}_{\mathrm{rec}} - \sum_i \mathrm{E}_i \quad . \tag{4.6}$$

To construct the `Pseudo TCs`, I add up the energies of all crystals that belong to one TC. Each TC then also has the energy values $\mathrm{E}_{\mathrm{rec}}$[*], E_i, and $\mathrm{E}_{\mathrm{bkg}}$. For each `Pseudo TC`, I check which E_i has the greatest absolute value in this TC and assign this Pseudo TC the corresponding offline ECL cluster i as a truth value. If $\mathrm{E}_{\mathrm{bkg}}$ is greater than every E_i, the `Pseudo TC` does not have a truth label. For every input TC given by TSIM, I now find the corresponding `Pseudo TC` and the assigned offline ECL cluster. The reconstructed energy, position and signal/background label of this assigned offline ECL cluster is then the truth label for this TC.

4.3 Training Datasets

To train the network, the datasets should represent different cluster signatures equally and contain all structures that the network should learn and be able to distinguish. For this network, which should improve the ECL L1 trigger, two focus points are set:

- **Signal/Background Classification:** With increasing beam backgrounds, a higher number of offline ECL clusters, even above the 100 MeV but not originating from the primary interaction, will be reconstructed in the ECL. The network should be able to distinguish between clusters from beam background and clusters from particles generated by the collision. Beam background clusters are predominantly low-energetic, but many low-multiplicity or dark sector processes also have only low-energetic clusters as their signature. To avoid the network learning a simple energy or timing cut, the so-called Category-One sample is constructed and explained in Section 4.3.1.
- **Overlapping Clusters:** The ICN-ETM can by design not separate clusters, which

[*] Ideally, $\mathrm{E}_{\mathrm{rec}}$ is the same value as the energy for that TC as given by TSIM, but due to the modeling of the hardware in TSIM, the values might differ.

hit adjacent TCs due to the ICN algorithm. This is one major point of improvement to increase trigger efficiencies for signatures that can be, when two adjacent particles are not resolved, mistaken for $e^+e^- \rightarrow e^+e^-(\gamma)$ or simply suffer from efficiency decreases. This problem is addressed by the construction of the so-called Category-Two Sample in Section 4.3.2.

In general, the network should not learn biases that are introduced by the construction of training samples. To avoid this, the training samples are technical samples, disregarding any conservation laws or physical constraints given by the original collision. For both the Category-One and the Category-Two sample, 1–6 photons are generated by using the `ParticleGun` module of `basf2`. For simplifications, and as the ECL is the only detector reliable for photons, the generated MC particles are solely photons. For the training samples, the photons are generated with a starting position at (0,0,0), with a timing of 0 ns. The energy is sampled from a uniform distribution between 0.05 and 7 GeV. The ϕ angle is sampled uniformly between 0 and 360°, covering the full ϕ angle of the detector. The θ angle is sampled uniformly between 5° and 175°. This covers a larger region than the ECL acceptance, which is between 12.4° and 155° to also include signatures, where particles from the collision are emitted very close to the beam pipe.

For the training of the networks deployed during data-taking at Belle II (see Section 6.5.2), ri Exp. 1003 beam background overlays were used.

4.3.1 Category-One Sample

For the Category-One sample, the goal is the mitigation of the bias for low-energetic beam background offline ECL clusters. As the energy distribution of the base dataset is uniform over the entire energy range, the beam background offline ECL clusters heavily dominate in the low energy region. Therefore, the sample is enriched with additional, low-energetic MC photons to model the energy distribution of the beam background offline ECL clusters.

Fig. 4.13 shows the number of offline ECL clusters per event and the distribution of offline ECL cluster energies for events with only beam background energies (as described in Section 4.1.3). The number of offline ECL clusters is modeled with a Poisson distribution, whereas the energy distribution is approximately modeled via an exponential fit, as shown in the plots.

While it is difficult to know the original energy of particles that are the cause of the beam background offline ECL clusters, they can be modeled via photons with the same energy as the offline ECL cluster. Low-energetic photons deposit in most cases their entire energy in the ECL and have only a small amount of leakage. Therefore, I generate additional `ParticleGun` photons following the distributions obtained from the fits in Fig. 4.13:

$$\text{Exp. 1003}: \quad P(N_\gamma) = \frac{3^{N_\gamma}}{N_\gamma!}e^{-3}, \quad P(E_\gamma) = e^{5.046-32.621E_\gamma} \quad . \tag{4.7}$$

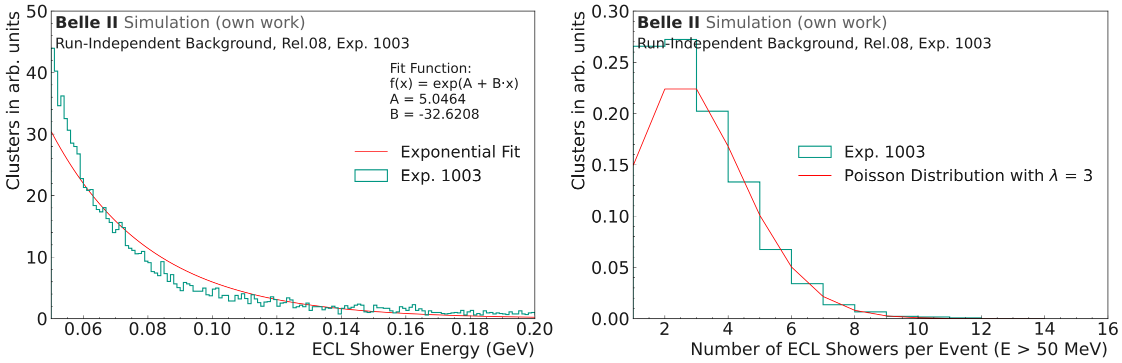


Figure 4.13: The number of offline ECL clusters (left) and the energy distribution of these offline ECL clusters (right) in events with only beam background energy depositions. The energy distribution is approximately modeled via an exponential fit. The beam background overlays used are ri Exp. 1003 overlays. The fit parameters are indicated on the plot.

If a generated photon energy is below 50 MeV, I redraw from the distribution until the generated energy is above 50 MeV. The θ and ϕ angles are chosen as explained before in a uniform way over the whole detector. This of course does not model beam background distributions perfectly, as they are more prevalent in the backward direction. One possible task for future work is to study the impact of an improved modeling of the angular distribution on the signal/background classifier.

In Fig. 4.14, the number of offline ECL clusters and the energy distribution of offline ECL clusters is shown. In both plots, one distribution is from events only consisting of energy depositions from beam background overlays. They are compared to events where no beam background overlays are used, but the energy depositions come from `ParticleGun` photons following the Poisson distribution and the exponential fit in terms of the number and energy of the photons. The plots show a good agreement between distributions. As this should only avoid an imbalance between true signal and true background offline ECL clusters, only similar distributions are necessary. For the energy distribution, offline ECL clusters below approximately 80 MeV will not be visible on trigger level due to the 100 MeV threshold. Therefore, the larger disagreement for low energies is acceptable.

For the Category-One sample, the 1–6 photons are simulated as explained in the previous section, and additional photons, following the energy and number distribution depending on the beam background overlay used, are added. This leads to the distributions in terms of generated MC particle energies and angles shown in Fig. 4.15 and offline ECL cluster energies and angles shown in Fig. 4.16. While the θ distribution for the generated particles is uniform, an excess in reconstructed offline ECL clusters in the backward endcap is visible. This is due to beam background, which is more prevalent in the backward endcap.

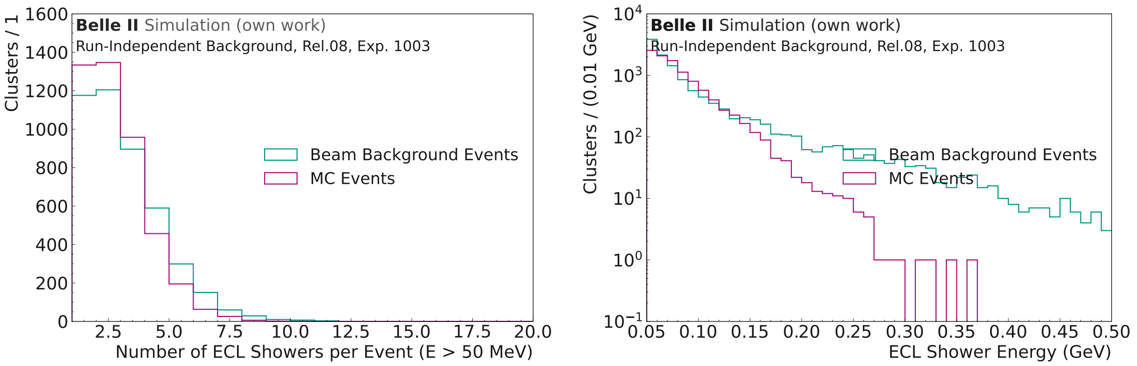


Figure 4.14: The number of offline ECL clusters (left) and the energy distribution of the offline ECL clusters (right) in events with only beam background energy depositions compared to the number and energy of offline ECL clusters in events with only MC particles generated following the approximated distributions. The beam background overlays are ri Exp. 1003 overlays.

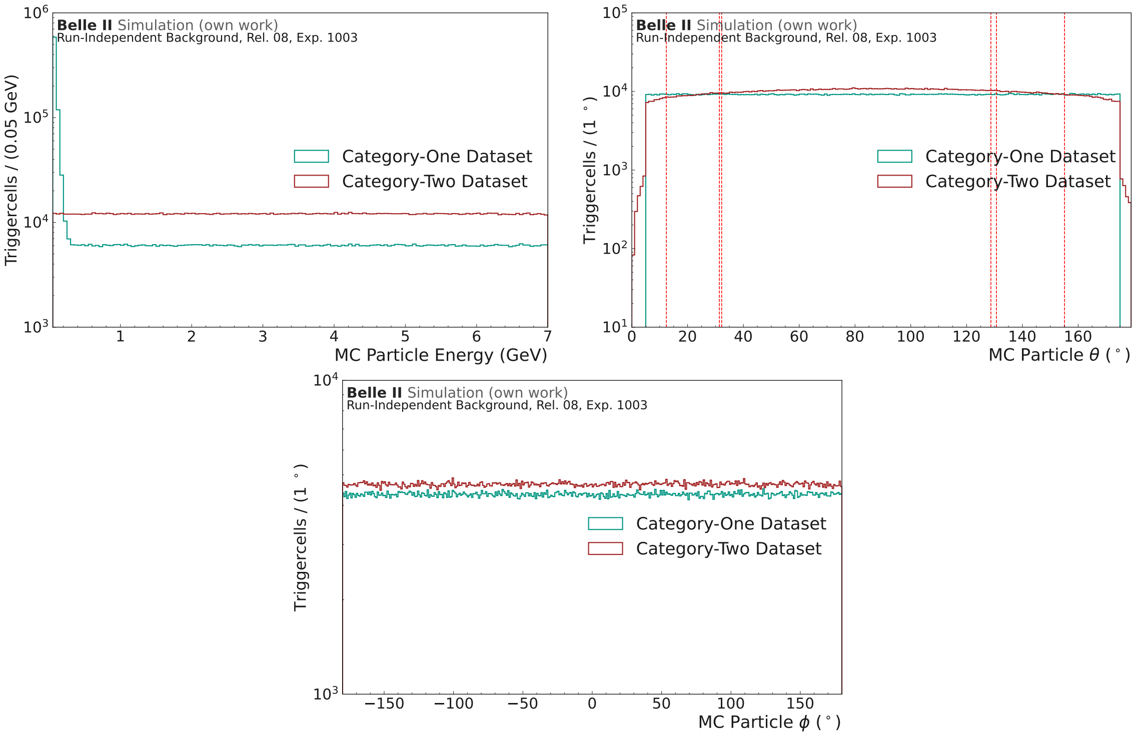


Figure 4.15: Energy (top left), θ (top right) and ϕ (bottom) distribution of the generated MC photons for the Category-One and Category-Two training sample. The red lines in the θ plot indicate the ECL acceptance, including the gaps between the barrel region and the endcaps.

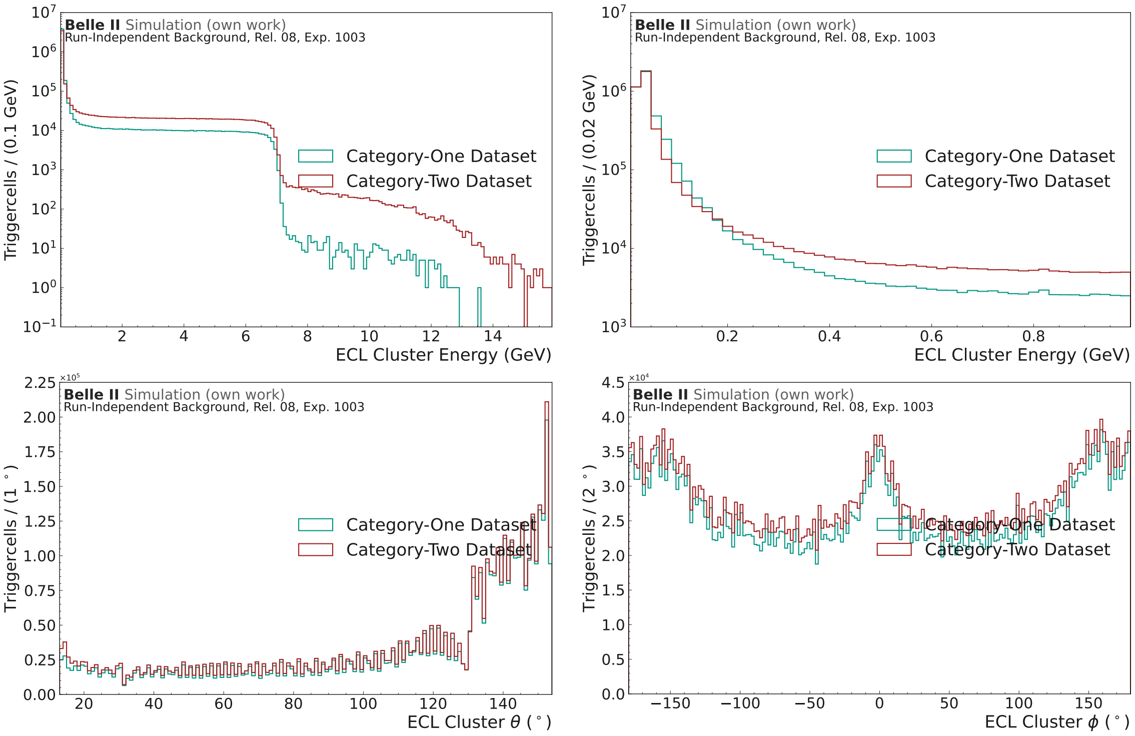


Figure 4.16: Reconstructed Energy (top left), reconstructed energy for the low-energy region between 0.01 and 1 GeV, θ (bottom left) and ϕ (bottom right) distribution of all reconstructed ECL clusters for the Category-One and Category-Two training sample.

4.3.2 Category-Two Sample

For the Category-Two sample, the network should learn to distinguish overlapping clusters. While there is no requirement for the original 1-6 simulated photons to have a minimum angular separation, on average the photons will hit the ECL with a wide enough distance to have clearly separated clusters.

To mitigate this, for this sample two additional photons are generated with the ParticleGun module with an opening angle between the generated direction of the two particles between 0.05 and 0.2 rad ($\approx 2.8^\circ$ - 11.2°). The generated energy values for photons are drawn from the same distribution as for the 1–6 photons. For θ and ϕ, the first photon values are drawn from the original distributions. For the second photon, the opening angle between both particles is drawn from a uniform distribution between 0.05 and 0.2 rad, and the momentum vector of the first particle is first rotated by the opening angle and then by a randomly drawn value between 0 and 2π. The distributions for energy, θ and ϕ of the generated MC particles can be seen in Fig. 4.15. The energy distribution is flat over the whole energy region in comparison to the Category-One energy distribution. The tails of the θ distribution outside the original generated range in comparison to the Category-One sample can be explained by the creation of the second, close-by particle with the rotation of the momentum vector.

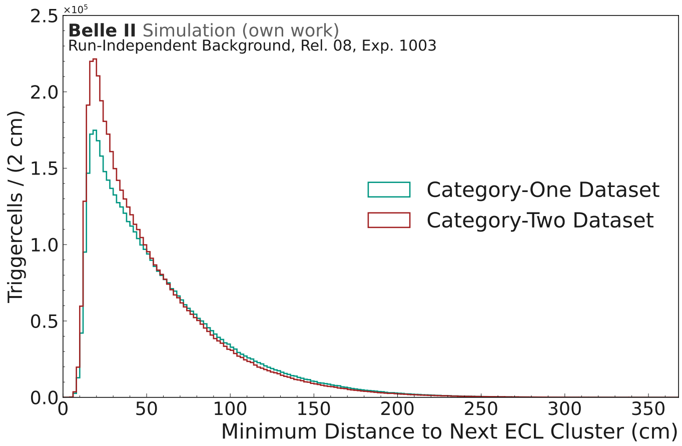


Figure 4.17: Euclidean Distance to the nearest cluster in the Category-One and the Category-Two dataset. For each reconstructed ECL cluster, the nearest neighbour in Euclidean distance is found and plotted here. The number of clusters with a small distance to their next neighbouring clusters dominate in the Category-Two dataset.

The overall distribution in energy, θ and ϕ for the reconstructed showers in the Category-Two sample can be seen in Fig. 4.16. Due to the two additional particles, the average number of higher-energetic offline ECL clusters is higher for the Category-Two sample, while the low-energy offline ECL clusters dominate in the Category-One sample.

In Fig. 4.17, the minimum distance for each cluster to the nearest other cluster in the baseline 1-6 photon dataset and the Category-Two sample is shown. The peak for close-together clusters is noticeably increased.

4.4 Evaluation Datasets

The evaluation of the network is done on test datasets for both the Category-One and Category-Two simulations. To further test the performance of the network on overlapping clusters and to reduce the uncertainty due to other effects, a fixed-energy overlap sample is simulated as well. In this sample, two photons with an opening angle between 0.05 and 0.2 rad are simulated with the `ParticleGun` module. Both photons have the same energy to simplify the evaluation. This does not necessarily cause two offline ECL clusters with the same energy but reduces the effect of energy imbalance.

Additionally, several physics samples are simulated to evaluate the network's performance and calculate trigger efficiencies. Collision data, not simulated samples, with which the algorithm is evaluated, will be explained in Chapter 8.

For the physics evaluations, the following datasets are simulated:

$\mathbf{e^+e^- \rightarrow e^+e^-(\gamma)}$ As this process is the process with the highest production rate at Belle II, the trigger rates have to always be monitored to avoid strongly increasing rates. Due to the favored low-angle emittance of the outgoing electrons and positrons, 10 Million

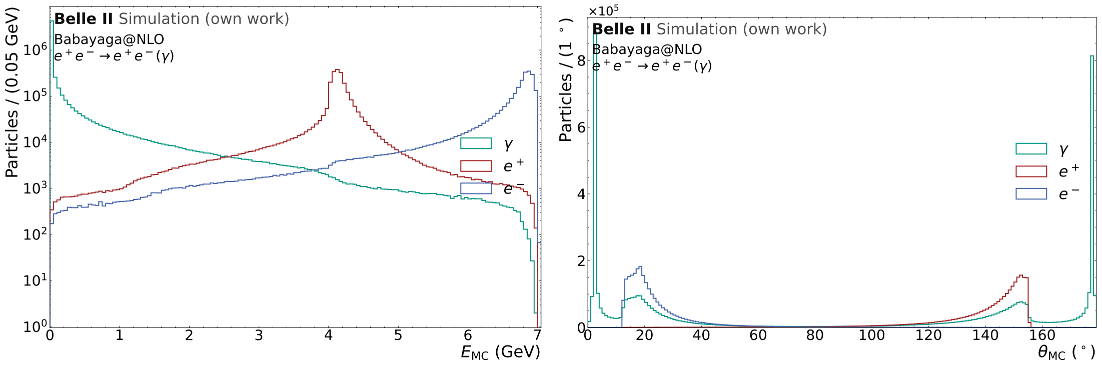


Figure 4.18: Generated energy and θ angle for the electron, positron and any radiative photons from the $e^+e^- \rightarrow e^+e^-(\gamma)$ process.

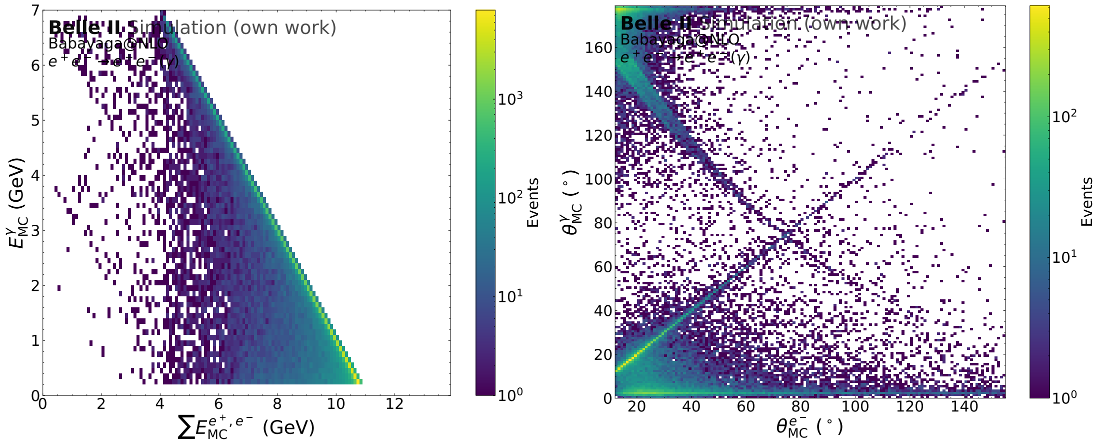


Figure 4.19: Sum of the electron and positron energy over the photon energy (left) and the electron θ angle over the photon θ angle for particles from the $e^+e^- \rightarrow e^+e^-(\gamma)$ process.

samples with a requirement of the θ angle for the outgoing e^- / e^+ between 10° and 170° in the generator are simulated with BABAYAGA@NLO ([54]). This results in a cross-section of 295.4 nb. After the event generator, further cuts on the generated particles are added, with θ for the electron and positron between 12.4° and 155.1° and at least two charged particles in the generator. This reduces the 10 Million events to 2.5 Million events.

Fig. 4.18 shows the generated distribution of particle energy and θ angle for the electron, positron and any additional, radiative photons. The two leptons have very high momentum and are peaked into the forward and backward direction, for the electron and positron, respectively. The photons are preferably emitted alongside the beam axis. Fig. 4.19 shows the sum of the lepton energies over the photon energy in the left plot, which is clearly anti-correlated. To select events without high-energy photons, requiring the two tracks to have high momentum suffices. Additionally, the right plot shows the θ angle of the photon over the θ angle of the electron. The angles are either positively correlated, so that the photon is emitted alongside the electron, or negatively correlated, so that the photon is parallel to the positron.

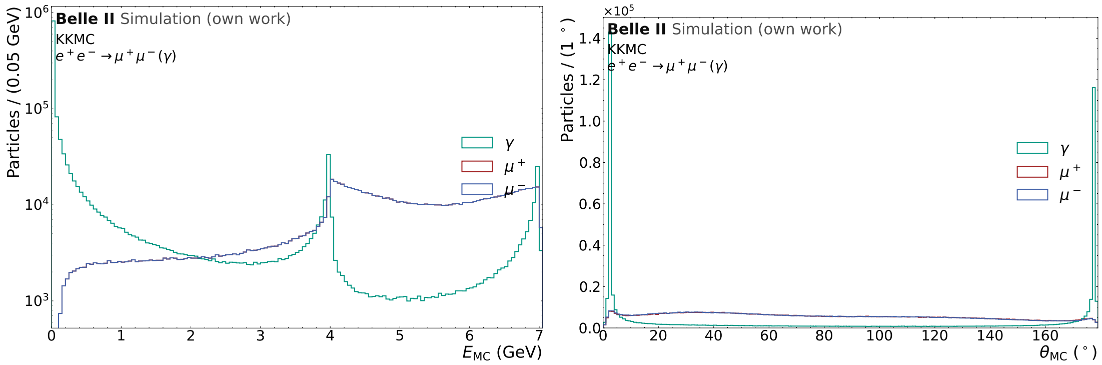


Figure 4.20: Generated energy and θ angle for the μ^-, μ^+ and any radiative photons from the $e^+e^- \to \mu^+\mu^-(\gamma)$ process.

$e^+e^- \to \mu^+\mu^-(\gamma)$ This process has a rather high cross-section at 1.15 nb and can be selected as a very clean sample for data- MC comparisons. As muons in Belle II are minimum ionizing particles, they deposit only an average of 6 MeV cm^{-1} in CsI(Ti) crystals [55]. As the crystals in the ECL, as explained in Section 2.3, are approximately 30 cm long, muons deposit around 200 MeV in the ECL with the energy deposition being contained in one crystal usually. This leads to clean selections and lower-energetic ECL clusters, which pose a challenge for the trigger. For this sample, 1 Million events are simulated with the KKMC generator ([56]) without any generator cuts.

Fig. 4.20 shows the generated energy and θ angle for the μ^-, μ^+ and any radiative photons. In contrast to $e^+e^- \to e^+e^-(\gamma)$, there is no clear energy or θ distinction between the μ^- and μ^+, but they are both nearly uniformly distributed in θ over the entire range. For the generated energy, the two peaks at 4 GeV and 7 GeV are visible, corresponding to the HER and LER beam energy. Any radiative photons are preferably emitted along the beam axis.

Pure Beam Background As the maximum trigger rate cannot get above 30 kHz even in high beam background conditions, the performance of the network has to be checked on samples only containing beam background to ensure a small false trigger rate. This is of course only an approximation of the real conditions during Belle II data-taking. For this, 500000 events containing only hits from ri Exp. 1003 beam background overlays are produced.

Chapter 5

Metrics

To evaluate the performance of the different network trainings and make them easily comparable, several metrics have to be implemented. While the network's loss is a good indicator if a network will perform reasonable or not, due to the high complexity of the OC loss, explained in Section 6.2.2, the overall loss value does not necessarily translate to good performance in all metrics. One of the important performance metrics is the Receiver Operating Characteristic (ROC) curve, explained in Section 5.2.3, to measure the signal classifier performance of the model. In Fig. 5.1, shows the comparison between five trainings with an identical training setup for the validation loss as well as the ROC curves. Even though the loss is equivalent, for three out of five networks, the signal/classifier performance is not better than random guessing.

There are several, very important tasks for the network to accomplish. It should find up to 100 % of all offline ECL clusters, not overpredict the number of offline ECL clusters heavily to not increase the trigger rate unreasonably, achieve good resolutions for the energy and position predictions and have a high background rejection for a good signal retention rate. Additionally, the trigger rate has to be evaluated, which is the end result and a combination of the performances of all other metrics. How the performance of the network is evaluated for these different tasks will be described in the next section.

5.1 Matching to Offline ECL Clusters

For the overall trigger evaluation, the total trigger rate and the trigger rate per bit are the main performance metrics. While this gives an upper bound to the overall performance, using only the rate as a metric makes it difficult to evaluate the network's performance in detail for different cluster signatures or varying physics processes and compare different trainings. As the network is trained on offline ECL clusters given by the offline reconstruction (see Section 4.2), the predicted clusters of the network are matched to the offline ECL clusters for a better evaluation. The same matching is applied to the ICN-ETM clusters to correctly compare the network to the existing trigger algorithm.

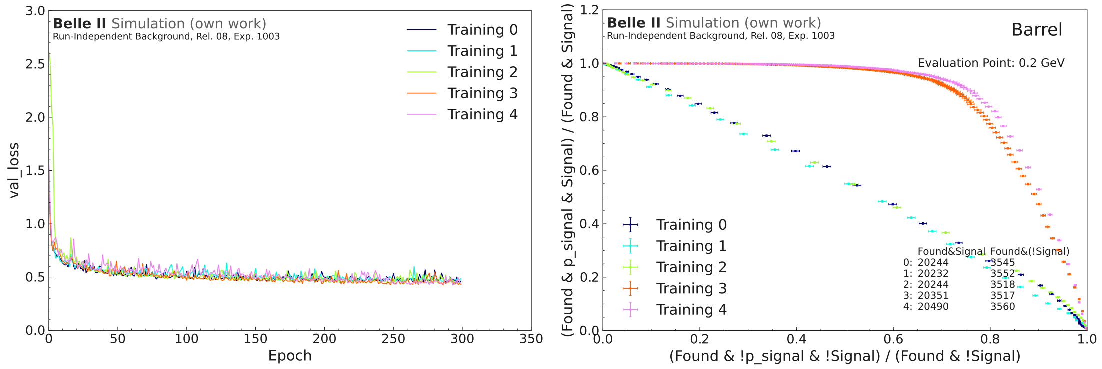


Figure 5.1: Comparison of the overall validation loss (left) and the output of the signal/background classifier as a ROC curve (right) for five trainings with the identical training setup. A ROC curve with a straight slope of 45° is the same signal/background performance as random guessing. As can be seen in the left plot, the overall validation loss between trainings does not differ significantly.

Matching trigger clusters to offline ECL clusters is based on the position difference and the energy ratio between the two clusters. While the network predicts the position of a cluster directly, the position of an ICN-ETM cluster is the center position of the highest energetic TC of that cluster (see Section 3.2). This position can be taken from a mapping of TC ID to cartesian coordinates in basf2, which uses a fixed LUT for this mapping. The positions have been calculated from the crystal positions, using the center of the side facing the IP, and then calculating the center of all crystals in one TC. In comparison to that, the offline position reconstruction uses the center in all three dimensions for a crystal to calculate the position of an offline ECL clusters. Additionally to this difference, there is a mistake in mapping the positions of the crystals to the TCs in basf2, where the positions have an offset in the ϕ direction. In Fig. 5.2, the outlines of the TCs can be seen with their center position according to basf2 in green, their center positions at the side facing the IP in purple and their center positions when taking the center of all three dimensions in black. As the outlines are drawn using the center of the crystals in all three dimensions, the black markers align with the visual center. The offset in ϕ between the TSIM TC positions and the calculated positions can be seen in all three detector regions.

For a correct comparison between ICN-ETM cluster and the network's cluster, the positions at the center of the crystals is used for the position of the highest energetic TC in an ICN-ETM cluster.

For the matching between trigger and offline ECL clusters, two conditions have to be fulfilled. The energy ratio between the trigger clusters and the offline ECL clusters has to be between 1 % and 200 %, whereas the Euclidean distance between both has to be equal or less than 40 cm. The two conditions are calculated with

$$\Delta\mathrm{pos} = \sqrt{(x_{\mathrm{TRG}} - x_{\mathrm{Offline}})^2 + (y_{\mathrm{TRG}} - y_{\mathrm{Offline}})^2 + (z_{\mathrm{TRG}} - z_{\mathrm{Offline}})^2} \quad \text{and} \quad R = \frac{E_{\mathrm{TRG}}}{E_{\mathrm{Offline}}} \quad , \tag{5.1}$$

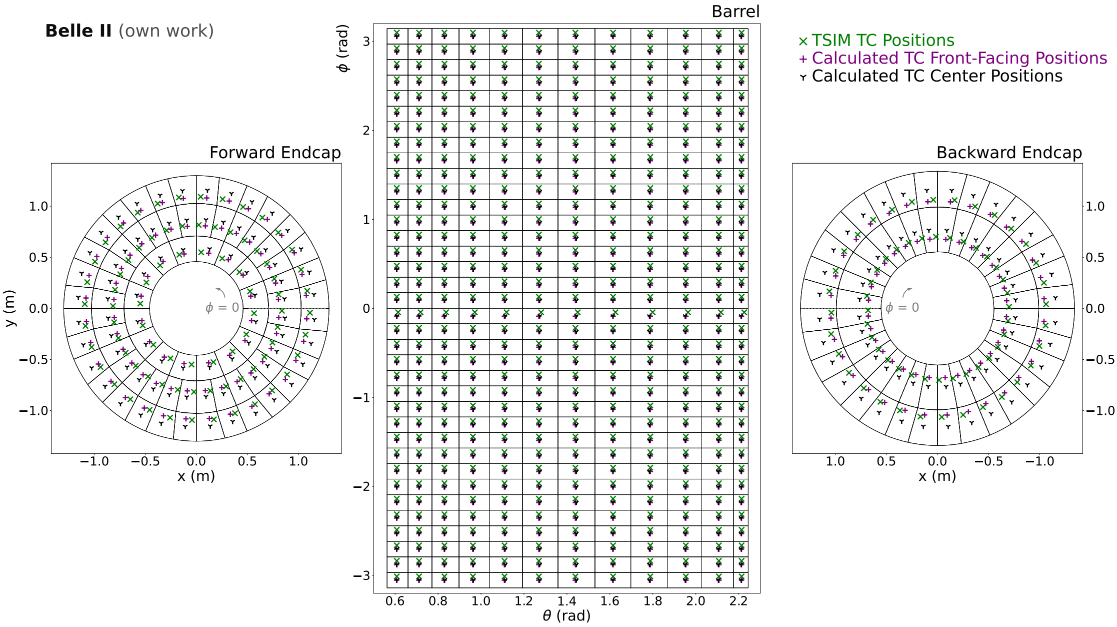


Figure 5.2: Comparison of the differently calculated center positions for the TCs in the ECL. The green x marks the center positions of the TCs as given by basf2 while the purple cross marks the positions if calculated from the center of the crystals. Both of these positions are taken at the front-facing side of the crystals. A shift in ϕ between purple and green can be observed over the full detector. The black marker shows the position when calculated in the center instead of the front face of the crystals. The θ position agrees with the purple markers, as the crystals are pointing towards the IP.

and the requirements are then Δpos $\leq$ 40 cm and $0.01 \leq \mathrm{R} \leq 2.0$.

For a justification of these matching boundaries, the minimum distance between offline ECL clusters and trigger clusters is tested in the left column of Fig. 5.3, while the best energy ratio is crosschecked in the right column. Every offline ECL cluster is compared to every ICN-ETM or network cluster in the same event and for the left column, the distance for the closest trigger cluster is shown. In the right column, I again compare every offline ECL cluster to every ICN-ETM or network cluster and take the energy ratio, where |1 - $E_{\mathrm{TRG}}/E_{\mathrm{Offline}}$| is the smallest. The trigger clusters for the same offline shower are not necessarily the same in the left and right column. The figure shows offline ECL clusters in the three detector parts, barrel, forward endcap and backward endcap, from top to bottom in that order. The thresholds for the matching between offline ECL clusters and trigger clusters are shown in red dotted lines. As can be seen, the thresholds do not cut into the distributions apart from singular outliers. This guarantees that the matching does not affect the resolutions and treats both ICN-ETM clusters and network clusters the same.

For the case of a trigger cluster being matched to several offline ECL clusters, I take the match with the lowest distance.

After this matching step, there can be several trigger clusters matched to the same offline ECL cluster. This does not play a role in the efficiency calculation but for all other metrics, the match with the best energy ratio (closest to 1) is taken.

5.2 Evaluation Metrics

Most of the evaluation metrics are based on the matching between trigger and offline ECL clusters to check the performance of the network and to validate the training. All evaluation metrics are calculated in bins of the true or predicted energy and separated into detector parts to correctly gauge the performance of the algorithms for all regions. Additionally, trigger bits and trigger rates are also calculated.

5.2.1 Efficiency and Purity

Efficiency and purity are defined by using the offline ECL clusters, that can be used as a target (see Section 4.2), as a baseline. The offline ECL clusters which have very low energy or are outside the trigger timing window can never be found and would arbitrarily lower the overall efficiency. Every offline ECL cluster that is assigned to one node as a target is a offline ECL clusters that both the ICN-ETM algorithm and the network have to find. These offline ECL clusters are denoted as true offline ECL clusters. Efficiency and purity are then defined as

$$\text{Efficiency} = \frac{N(matched)}{N(true)} \qquad \text{Purity} = \frac{N(matched)}{N(TRG)} \quad , \tag{5.2}$$

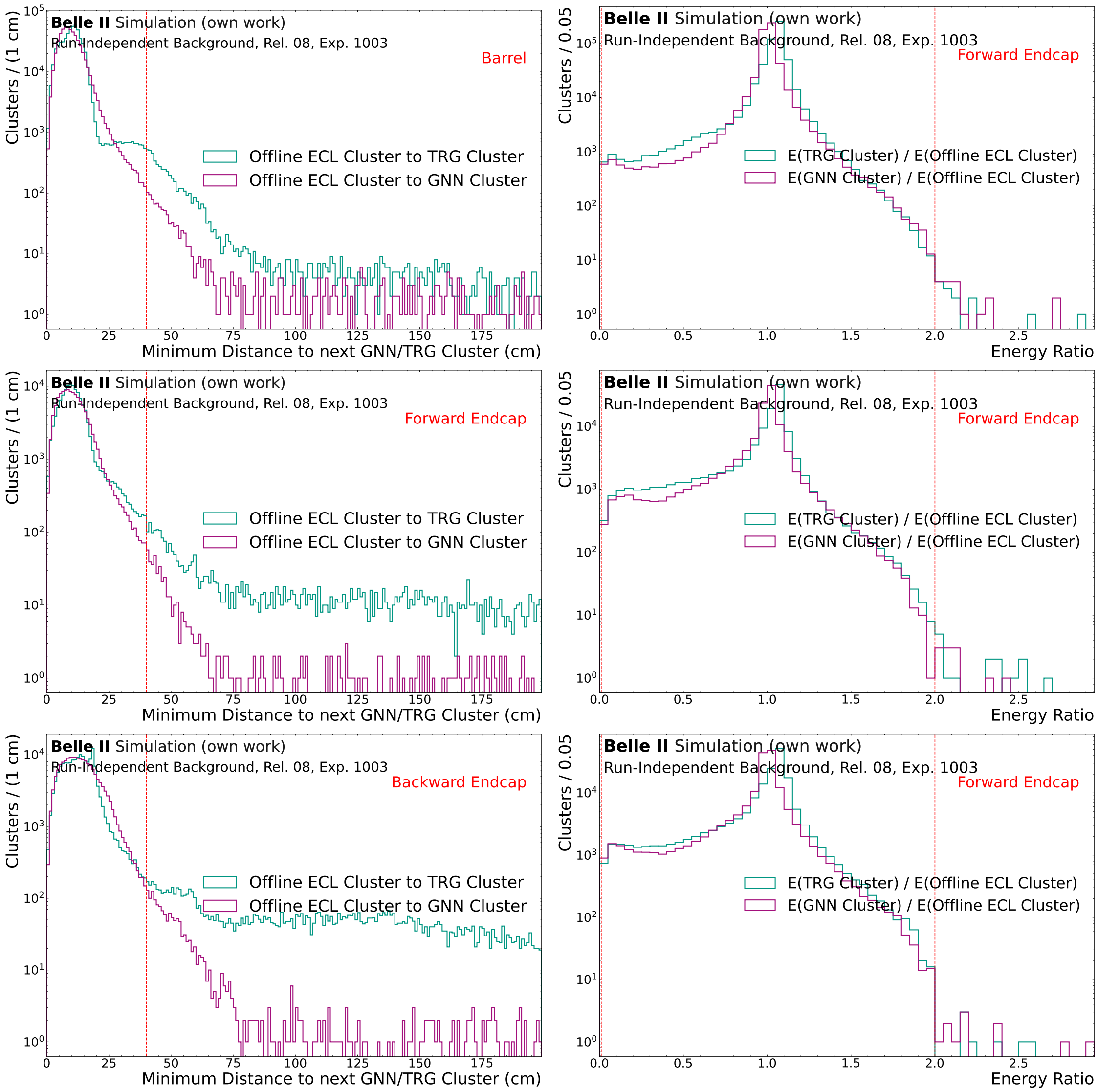


Figure 5.3: Minimum distance and closest energy ratio to 1 of each target offline ECL cluster to any ICN-ETM or network cluster. The network was trained on the combined training dataset described in Section 4.3 and evaluated on a test Category-One sample. The evaluation of the ICN-ETM performance was done on the same sample. The top row shows target offline ECL clusters located in the barrel part, the center row depicts the forward endcap and the bottom row the backward endcap. The red dotted lines show the thresholds of the matching requirements.

where matched clusters are ICN-ETM or network clusters which have been matched as described in the previous section to an offline ECL cluster. TRG clusters are all clusters predicted by the network or given by ICN-ETM regardless if they are matched to an offline ECL cluster or not.

Efficiency is a measure to check the algorithms' performance in returning the correct cluster, while a good purity ensures that the overall trigger rate will not rise massively by overpredicting the amount of clusters.

5.2.2 Energy and Position Resolution

To estimate the performance of the prediction for energy and position, the width of the resolution distribution is used as a metric. The resolution for energy is taken as the ratio between the absolute difference between the trigger energy and the offline ECL cluster energy over the offline ECL cluster energy. This results in a percentage value for the resolution to ensure that higher energy clusters don't dominate the resolution. For the position resolution, the difference between trigger position and offline ECL cluster position is taken. The resolutions can then be calculated with

$$\eta_E = \frac{E_{\mathrm{TRG}} - E_{\mathrm{offline}}}{E_{\mathrm{offline}}}; \qquad \eta_{\mathrm{x}} = x_{\mathrm{TRG}} - x_{\mathrm{offline}} \quad \text{for x, y, z} \quad . \tag{5.3}$$

This results in a positive value if the trigger algorithm, both ICN-ETM and the network, have a higher position or energy prediction than the offline ECL cluster value and in a negative value if the trigger prediction is lower than the offline prediction. The resolution is therefore more easily interpretable.

For the width of the resolution, one standard way is to fit the resolution with a Gaussian or a Crystal Ball function [57] to then estimate the Full Width Half Maximum of the distribution. While this has been done in studies of energy resolution improvements for ECL offline reconstruction such as [47], the restrictions of the trigger energy and position reconstruction due to energy thresholds and very coarse inherent resolution due to the size of the TCs result in very unstable fits and make a correct parametrization difficult.

A Gaussian fit is applied nevertheless to correct for biases in the resolutions. The fit range is estimated by using the interval of one standard deviation in both directions around the mean. The mean μ of the fit is then used as a multiplicative correction factor $f_{\mathrm{corr}} = 1$ - μ for the predicted energy or position:

$$\eta_E = \frac{E_{\mathrm{TRG}} - E_{\mathrm{offline}}}{E_{\mathrm{offline}}} \quad \rightarrow \quad \eta_{E,bc} = \frac{E_{\mathrm{TRG}} \cdot f_{\mathrm{corr}} - E_{\mathrm{offline}}}{E_{\mathrm{offline}}} \quad . \tag{5.4}$$

The mean μ can also be used as a measure for the bias of the resolution.

Then, the width of the resolution, denoted by r, is calculated analogously to [58]:

$$r_{\mathrm{E/pos}} = P_{68\%}(|\eta_{\mathrm{E/pos,bc}} - P_{50\%}(\eta_{\mathrm{E/pos,bc}})|) \quad , \tag{5.5}$$

where P_q stands for the q-th percentile of the distribution. This can be used for comparisons, but is not very sensitive to outliers or unwanted shapes of the resolution. Therefore, the overall resolution distributions have to be checked in tandem.

5.2.3 Signal Retention and Background Rejection

For the background rejection evaluation, all predicted clusters that are matched to an offline ECL cluster are taken into consideration, whereas unmatched clusters are discarded in this evaluation. The network predicts a floating point value between 0 and 1 for each trigger cluster, with 1 being signal and 0 being background. For the ICN-ETM, no such value is given, so there is no background rejection calculation for the ICN-ETM clusters.

First, a ROC curve can be calculated. For the ROC curve, the cut for the predicted signal classifier value is varied. For each cut value, where each cluster with $p_{\text{signal}} > \text{cut}$ will be classified as signal, a signal retention rate R_S and a background rejection rate R_B can be determined:

$$R_S = \frac{N(\text{matched}\,\&\,\text{signal}\,\&\,(p_{\text{signal}} > \text{cut}))}{N(\text{matched}\,\&\,\text{signal})} \quad R_B = \frac{N(\text{matched}\,\&\,\text{bckg}\,\&\,(p_{\text{signal}} \leq \text{cut}))}{N(\text{matched}\,\&\,\text{bckg})} \quad . \tag{5.6}$$

An exemplary ROC curve can be seen in Fig. 5.4. The points in the plot are for different cut values for the signal classifier output, with R_B shown on the x-axis and R_S on the y-axis. A perfect classifier output would be in the top right corner with 1.0 for both rates. A classifier, which would randomly guess a signal or background value, would have a ROC curve at 45° from top left to bottom right.

For an easy comparison between different trainings, I calculate the background rejection rate for a fixed signal retention rate of 95 % for each of the different detector parts. As the amount of background offline ECL clusters in the different detector parts varies, with the backward endcap having the highest occupancy of background hits, the cut value for each detector part can be determined separately. For simplicity, for all evaluations in Chapter 8 a single cut value is set for all detector regions.

5.2.4 Trigger Bits and Trigger Rate

To compare the final trigger rates between ICN-ETM and the network, the current used trigger bits as explained in Section 3.2.3 are also calculated for the network. For the transition from laboratory frame into the CM frame, the CM energies and positions are calculated by using the Lorentz boost from `ROOT` [59]. The trigger bits for the ICN-ETM are taken from basf2 TSIM.

These trigger bits can be calculated for each evaluation sample separately. Additionally, the expected trigger rate R_{TRG} for physics processes can be calculated by using the cross

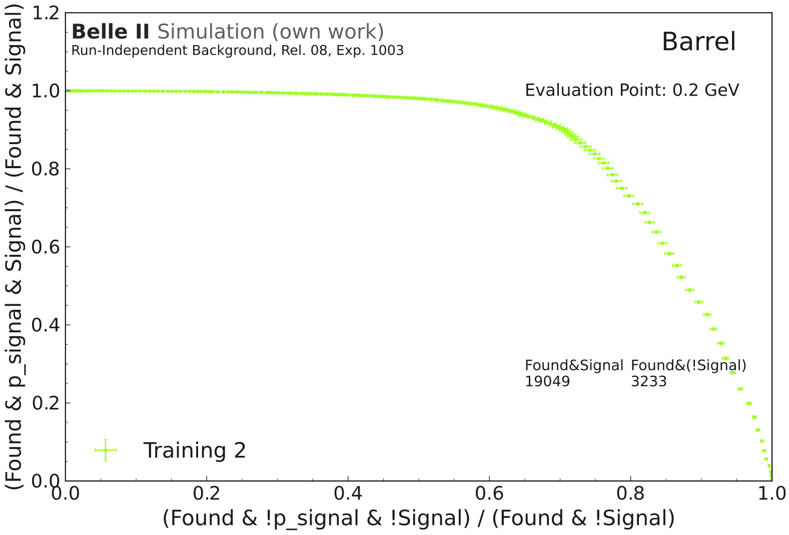


Figure 5.4: ROC curve for one network training for offline ECL clusters between 150 and 250 MeV. Each point in the plot is a different cut value for the signal classifier output. On the x-axis, the background rejection rate R_B is shown, while the y-axis shows the signal retention rate R_S. The ideal point of performance would be in the top right corner with 100 % signal retention and 100 % background rejection.

section and the trigger efficiency:

$$R_{TRG} = \mathcal{L} \cdot \sigma_{\text{process}} \cdot \frac{N(bit = 1)}{N(all)} \quad , \tag{5.7}$$

with the instantaneous luminosity $\mathcal{L}$, the process cross section σ_{process}, the number of events triggered by the respective trigger bit $N(bit = 1)$ and the total number of events $N(all)$. For the beam background rate, the calculation explained in Section 4.1.3, where the trigger rate is estimated by the size of the trigger window, is used.

Chapter 6

Network Design

The overall strategy of this work was to develop a Neural Network (NN)-based trigger algorithm to test and demonstrate the development, implementation and usage of ML algorithms in a realistic environment with high throughput and latency requirements. Improvements to the current ECL L1 trigger algorithm will be necessary when increasing the instantaneous luminosity towards the design luminosity of SuperKEKB and in the case of an update to the hardware which would result in finer granularity for the TCs. In this chapter, I will explain the requirements to the network imposed by the hardware requirements, the overall architecture of the network and the algorithm itself, and the improvements towards implementation.

6.1 Software/Hardware Co-Design

The implementation of the network is done on a UT4 board, which is used in the current L1 trigger system and features a Virtex Ultrascale XCVU080/160 FPGA. Due to the TCs geometry and 100 MeV energy threshold (described in Section 3.2), the input for the network is very sparse, with less than 5 % of all available TCs being active per trigger decision window on average. Additionally, the TCs are not given in a fixed order into the network without manually sorting, which is often required in machine learning algorithms.

The optimization of the network design is a high-dimensional problem. The implementation platform, in this case the UT4 board, gives hard limits on the size of the network and the number of multiplications possible due to the limited amount of resources. Additionally, since the here-used *GravNet* layer is based on a k-Nearest-Neighbour (kNN) clustering algorithm, the network input from different data windows has to be conglomerated at one point and cannot be processed separately. This leads to additional limits on the network design to reach a reasonable latency of $\mathcal{O}(1\,\mu\text{s})$ and fulfill the throughput requirement of 8 MHz. After choosing a network design with a set number of parameters, choosing the bit size of the weights and biases of the respective layers is a separate optimization step to reduce resource usage on the FPGA. This step is called quantization and is done with

QKeras, where the quantization values can be set separately for each layer. Additionally, a method often employed to further reduce the number of parameters is called pruning. With pruning, a set percentage of nodes with the smallest absolute weights in each layer of the network is set to 0 after the backpropagation step. This is done during training and the pruned nodes then do not have to be included in the calculation during the inference step.

With a set quantization and pruning setup, the network design can again be optimized within the given restrictions. This is an iterative process and has to be done in very close connection to the actual hardware implementation. In Section 6.3, the quantization technique, pruning and the actual network quantization optimization is explained with the final network architecture chosen for the implementation used for the results in Chapter 8. In Section 6.4, the optimization of the network design within the quantization set in Section 6.3 is shown.

6.2 Network Architecture

Due to the sparse, not-ordered input of the TCs and the variable geometry of the ECL, especially in the endcaps, a GNN architecture was used for this work. In earlier works for improving clustering for the offline ECL reconstruction[47], the GravNet layer [15] was applied and achieved improvements of up to 20 % in respect to the current clustering. The learnable graph-building, as explained in the next section, can improve the network's performance by removing preconceived information given by the programmer. Other GNN techniques have also been tested on offline clustering, but have not shown improved performance to the GravNet algorithm [60]. Therefore, GravNet was also used for the GNN implementation here. Testing different network architectures to improve performance and implementation results is currently a work in progress.

6.2.1 Graph Neural Networks and the GravNet layer

The use of a GNN as an algorithm necessitates the fact that the data has to be encoded in graphs. Graphs consists of a set of objects, here called nodes, and their relationship to each other, here called edges. Graphs are a non-Euclidean data structure, opposed to for example Convolutional Neural Networks (CNNs), and can be applied to a widespread number of use cases, as they are solely based on some kind of relational information between the objects. GNNs are widely used in many different applications outside HEP, such as antibiotic discovery [61], recommendation systems based on past interests [62], stock market predictions [63] or preventing adversary attacks on online systems [64]. In HEP, GNNs can be used for reconstructing tracks originating from charged particles [65], full decay chain reconstruction [66], [67], anomaly detection [68] or flavour tagging [69]. For the reconstruction of clusters in electromagnetic or hadronic calorimeters, GNNs are often employed due to the sparsity and irregularity of the inputs. Early works used static graph building using a kNN algorithm in Euclidean space and applied an EdgeConv [70] based GNN to predict edge labels to correctly cluster particles of different types [71]. In the Compact Muon Solenoid (CMS) experiment, the GravNet layer was introduced in the context of calorimeter clustering, as well as the OC algorithm [16]. The GarNet layer, which was introduced in the same work as the GravNet layer [15], was used in an algorithm for clustering in a toy calorimeter, which was then implemented in `hls4ml` [11] for a proof-of-concept implementation for a Xilinx Kintex UltraScale FPGA. The resource usage estimations for this work were taken from the HLS synthesis, which has been shown to lack accuracy in estimating the number of LUTs, time per clock cycle, and number of Digital Signal Processors (DSPs) [72].

For a comprehensive overview of GNNs, [73] gives a detailed explanation into the methodology of GNNs, while [74] gives an overview of GNN-based methods used in different areas at the Large Hadron Collider (LHC). I will give a short summary of the most important GNN features related to this thesis.

The GravNet layer

The graph construction, which is defined by connecting nodes by an edge with relational information between these nodes as features, is the underlying baseline in the use of GNNs. The number of connections each node can have to other nodes in the graph as well as the edge features directly influences the learning through feature aggregation and message passing. The GravNet layer learns the graph representation within a latent space, omitting the need to build the graphs beforehand.

To do this, the input F$_{in}$ to the GravNet layer, which is usually a point cloud input after one or several dense layers, is put through two parallel dense layers, which can be seen in Fig. 6.1 on the top left. The first dense layer learns a representation space for the input points. The number of output nodes of this layer equals to the number of dimensions in this representation space. The second dense layer transforms the input features into features within the feature space F_{LR}, which are then the features used for the message passing.

In step 2, the graphs are built using the learned representation space S. In this space, for each node, the k nearest neighbors are found by employing a kNN algorithm. The number of nearest neighbors is a hyperparameter, which can be optimized. For each of these neighbors, an edge is defined with a strictly decreasing weight with a larger distance.

In step 3, the features for each node are aggregated from its neighbors. For each neighbor, the feature of this node in the F_{LR} is multiplied by its weighted distance. The features from all neighbors are then aggregated with different methods, such as the sum or the average value of the features from all neighbors. This is the message passing step.

For each node, all aggregated features and the original input features are concatenated in step 4 and put through an output dense layer. This dense layer transforms the features into the output features F_{out}.

The number of dimensions in the representation space S, the number of features per node in the feature space F_{LR}, and the number of nearest neighbors for the graph building

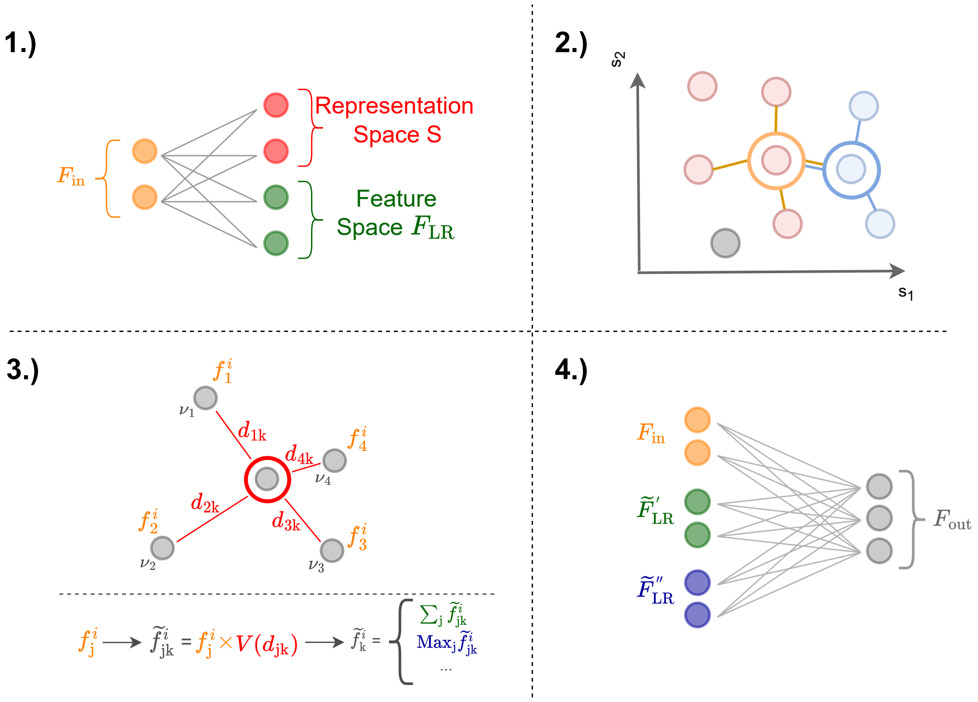

1.)
F_{in}
Representation Space S
Feature Space F_{LR}
2.)
s_2
s_1
3.)
f_1^i
ν_1
d_{1k}
d_{4k}
f_4^i
ν_4
d_{2k}
d_{3k}
f_2^i
ν_2
f_3^i
ν_3
$f_j^i \longrightarrow \tilde{f}_{jk}^i = f_j^i \times V(d_{jk}) \longrightarrow \tilde{f}_k^i = \begin{cases} \sum_j \tilde{f}_{jk}^i \\ \text{Max}_j \tilde{f}_{jk}^i \\ \dots \end{cases}$
4.)
F_{in}
$\tilde{F}'_{\text{LR}}$
$\tilde{F}''_{\text{LR}}$
F_{out}

Figure 6.1: Overview of the GravNet layer algorithm. The input F$_{in}$ is transformed via a dense layer into two latent spaces (1): The representation space S and the feature space F_{LR}. In the next step, the graphs are built via a kNN algorithm with the distance in the learned representation space S (2). Each node is connected to its N closest neighbors in this space. For each node, a message passing step (3) with the features of the feature F_{LR} is applied, using predefined aggregation metrics such as the sum or the mean of all features. The features are weighted by the distance in the latent space, with higher weights given to those nodes that are closer in the latent space. All features per node are then concatenated and put through another dense layer to be converted into the output features F$_{out}$ (4). The figure is reproduced from [15].

are hyperparameters for the overall network design.

6.2.2 The Object Condensation Algorithm

Conventional clustering algorithms in particle physics either start with a seed-finding approach and then apply the clustering solely on the area around this seed or impose boundaries on the number of objects in the calorimeter or the area of interest ([1], [75], [76]). While the two-step approach of seed finding works very efficiently, in a low-latency environment a one-shot approach to tackle both detection and reconstruction is beneficial to reduce the overall runtime and resource usage. However, restricting the number of clusters in total additionally reduces the generalization of the reconstruction algorithm.

While recognizing an unknown number of objects has been a task in computer vision, where many solutions have already been found [77] [78], the algorithms usually work on finding anchors within a picture and applying bounding boxes around those anchors. Overlapping objects are then solved by analysing the union of two bounding boxes and assigning pixels to different objects. In the case of clustering within a calorimeter, in difference to computer vision, in many cases one hit is fractionally assigned to different particles instead of a clear assignment to one.

OC was proposed as a one-shot approach specifically for calorimeter clustering in sparse environments, where the possibility of overlapping, but still well-separable clusters is high [16]. Each node in the input, which corresponds to a TC in this thesis, is assigned a label to belong to one object. This is described in Section 4.2. Nodes, which do not belong to one object, are labelled as background.

The algorithm is based on the introduction of a learned potential in a latent space, which draws nodes belonging to the same object close together and pushes points not belonging to the same object further away. One representative point per object then carries the prediction values, such as energy and prediction. To evaluate the algorithm, the representative points have to be selected, where the number of nodes then corresponds to the number of reconstructed objects, and their predictive values read out, which are then the properties of those reconstructed objects.

The notation in the following is close to the original notation in [16]. The potential is induced by a charge for each node. This charge is calculated by a so-called β-value, where one representative node of each object is assigned a high β-value. This β-value is learned by the network and bound between 0 and 1. The charge q is then defined by a function of β, which has zero gradient at 0 and a pole at 1. This function is chosen to be

$$q = \text{artanh}(\beta) + q_{min} \tag{6.1}$$

with β being clipped between 0 and 1 - ϵ ($\epsilon = 10^{-7}$ to avoid numerical instabilities. q_{min} is a value > 0 to set a minimum charge per node. This avoids the local minimum of $\beta = 0$

everywhere. With this charge, the repulsive and the attractive potential can be defined:

$$V_{\text{rep,k}}(x) = ||x - x_\alpha||^2 q_{\alpha k}, \quad V_{\text{att,k}}(x) = \max(0, 1 - ||x - x_\alpha||) q_{\alpha k} \quad \text{with} \quad q_{\alpha k} = \max_i q_i M_{ik}. \tag{6.2}$$

In theory, the potentials have to be calculated from every point to every other point. Since this can be computationally expensive, the potentials are approximated by the potential of the point α with the highest charge for the object k, which is ensured by M_{ik} which is 1 if the point i belongs to the object k and 0 otherwise. $|| \cdot ||$ is the L2-norm, while x are the coordinates of each point in an n-dimensional learned representation space.

The overall potential loss can then be written as:

$$L_V = \frac{1}{N} \sum_{j=1}^{N} q_j \sum_{i=1}^{K} \left(M_{jk} V_{\text{att,k}}(x_j) + (1 - M_{jk}) V_{\text{rep,k}}(x_j) \right) \tag{6.3}$$

with N being the total number of points and K the total number of objects in that sample.

To avoid the local minimum $\beta_i = \epsilon \ \forall i$, a β and noise suppression loss term is introduced. This loss additionally enforces one condensation point per object and none for background vertices by requiring

$$L_\beta = \frac{1}{K} \sum_k (1 - \beta_{k\alpha}) + \frac{1}{N_B} \sum_i^N n_i \beta_i \tag{6.4}$$

with N$_B$ being the number of background points. The sum of the potential loss and the β and noise suppression loss makes up the OC loss to determine condensation points. Additional loss terms are introduced for each feature which should be predicted. For this work, three feature loss terms are added, the signal loss, the energy loss, and the position loss. For the signal loss, the loss term is calculated by

$$L_{\text{signal}} = -t_{\text{signal}} \cdot \ln(p_{\text{signal}}) - (1 - t_{\text{signal}}) \cdot \ln(1 - p_{\text{signal}}), \tag{6.5}$$

with t_{signal} being the true signal/background flag with a value of 1/0, respectively, and p_{signal} the predicted value between 0 and 1. The position loss is computed as the mean absolute error between the predicted and true position.

For the energy loss, several loss terms have been tested. A mean absolute error loss has been used for most experiments, also in the deployed model explained in Chapter 8. To improve the energy prediction for low-energetic clusters, a logarithmic difference has been tested with the loss term as

$$L_E = |\ln(p_E + \epsilon) - \ln(t_E + \epsilon)| \quad . \tag{6.6}$$

Additionally, instead of predicting the energy of the cluster, the prediction of a scaling factor on the energy of the TC has also been tested.

The feature loss terms are weighted with the charge term so that they scale similarly with β. Due to this, the importance for the representation points per object, those with high β value, to carry the right prediction values is much higher than for all other points per object. The total feature loss term is then

$$L_F = \frac{1}{\sum_i^N \xi_i} \sum_i^N \left(L_{E,i} + L_{Pos,i} + L_{\text{signal},i} \right) \xi_i \tag{6.7}$$

with the scaling term

$$\xi_i = (1 - n_i) \operatorname{artanh}(\beta_i) + q_{\min} \tag{6.8}$$

where n_i is 1, if the point is noise and 0 if it belongs to an object.

All loss terms are then added to the overall loss function for the network. Each loss term can in theory be scaled with an additional factor to improve the performance of the network in a specific area. For this work, all loss terms have equal weight.

Inference

The inference for the OC is different than in usual network designs, where it is just a read-out of the network's prediction. For OC, the correct condensation points have to be selected and their predictive values have to be read out. In Fig. 6.2, an overview of the selection algorithm, called condensation point selection algorithm, is shown. In the first step, a cut on the β-value is applied. All points below this cut value, which is a hyperparameter, are removed in the inference step. As a next step, the remaining points are sorted by their β-value. The first point with the highest β-value in this list is chosen and the distance to all other remaining points in the learned latent space is calculated. Every point below a distance threshold is removed as well. The distance threshold is the second tunable hyperparameter. This process is then repeated with the next, still remaining point in the sorted list until the end of the list is reached. All remaining points are then the condensation points. The number of condensation points corresponds to the number of predicted clusters in this sample. Each condensation point also has predictive values for the energy, position and signal/background flag. These are then the values for the respective predicted clusters.

Both the cut on the β-value and the cut on the distance in the latent space are hyperparameters. They are not part of the training process, which means that fully trained networks can be optimized in terms of efficiency and purity afterward by changing the cut values. Smaller cut values translate into more predicted clusters, which in most cases increases efficiency but might lower the purity.

6.2.3 Overall Network Design

The implementation of the network has been done both in `PyTorch Geometric` [79], which is a library built on `PyTorch` [80], as well as in `Keras` [81], which uses `Tensorflow` [82] as a

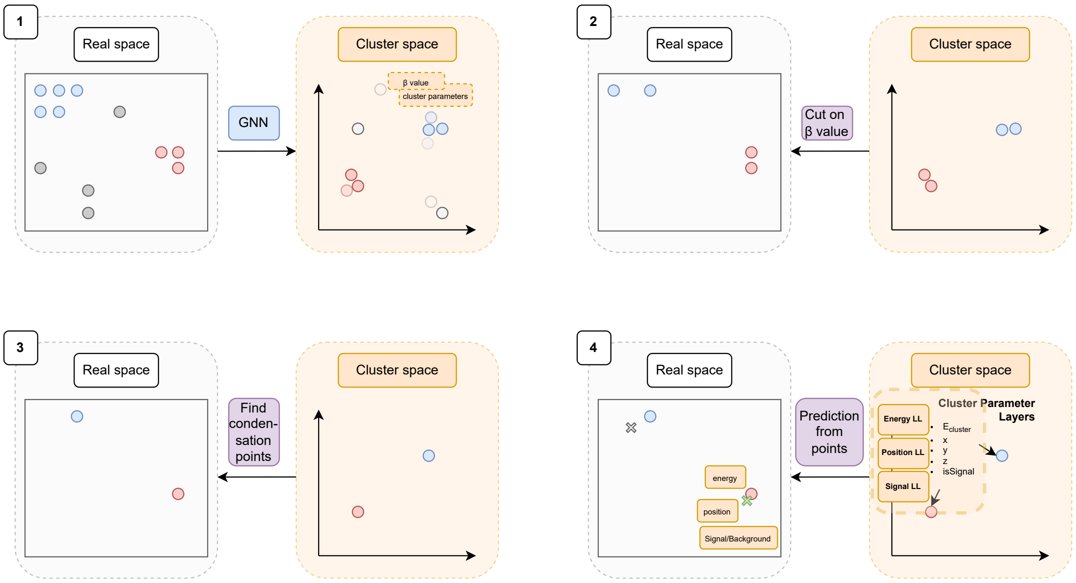


Figure 6.2: Overview of the inference step, also called condensation point selection algorithm. In step 1, a cut on β is applied to only select those points with a high probability of being a condensation point. In step 2, the remaining points are sorted by their β-value, and the first point is chosen. Every other point in this list, which is within a set distance in the latent space to this other point, is removed. Then the next remaining point is chosen and the same process is repeated. All points left are then condensation points, as shown in step 3. In step 4, the predicted values on the condensation points are then the energy, position, and signal/background flag for each predicted cluster. This figure is adapted from [65].

backend. The Keras implementation is then adaptable to be implemented in QKeras [12], a framework for quantization-aware training (see Section 6.3). QKeras is an extension to Keras and offers drop-in replacements for many common layers and activation functions to create quantized networks and perform quantization-aware trainings. Besides the quantization specifications, the implementation between Keras and QKeras is identical and will be treated as such in this text. Any differences will be specifically stated.

Initial tests and studies for this work have been done in Pytorch Geometric, whereas the quantization studies and network trainings for implementation have been done in Keras. One main difference is in the input format: Pytorch Geometric offers a data handling that has been optimized for graph structures so that differently sized inputs can directly be given to the dataloader of Pytorch Geometric. Internally, Pytorch Geometric builds one sparse block diagonal adjacency matrix per batch, which encodes the edges between the nodes, and concatenates feature and target matrices in the node dimension [83]. In comparison to that, Keras can only work with same-sized inputs. To achieve this, for each input event, the list of TCs is always padded to 32 or 64 nodes, with all features set to 0. For everything else, the effective implementation between Keras and Pytorch Geometric is the same.

The overall input to output chain in the inference can be seen in Fig. 6.3. The input is transformed into a 32 by 5 dimensional matrix and sent through the network. The input to the network has been described in detail in Section 4.2 and consists of all TCs above the set energy threshold, which corresponds to 100 MeV for the current ECL trigger setup. For each TC, the reconstructed energy, its position extracted from a LUT through the TC ID, and the time relative to the highest energetic TC in this trigger decision window is given to the network as input. The TCs do not follow any specific ordering as this can not be guaranteed on the hardware. This means that the model cannot use any information given by the ordering of the TCs.

For the overall network design, the network has three distinct layer types, dense layers, batch normalization layers [84] and GravNet layers. In Fig. 6.4, a detailed view of the model architecture is shown. The input nodes are passed into two layers, the first dense layer of the first of the repeating blocks in blue, and a scaling layer. This scaling layer is necessary due to the usage of skip connections [85]. Instead of a straight feed-forward pass for the data through the network, the input as well as the output of each repeating block additionally bypasses the rest of the network and is concatenated to the output of the last repeating blocks. These concatenated values are then input to the last dense layer before the output layers. These skip connections can help improve the network's performance by ensuring a more direct gradient calculation when performing the backpropagation step. Very deep networks often create vanishing gradients in early layers due to the high number of parameters beforehand, which can be mitigated with the help of skip connections. In a standard neural network, the scaling layer would not be necessary, as the input could directly be used as input to the last dense layer. In standard Keras, everything is calculated in

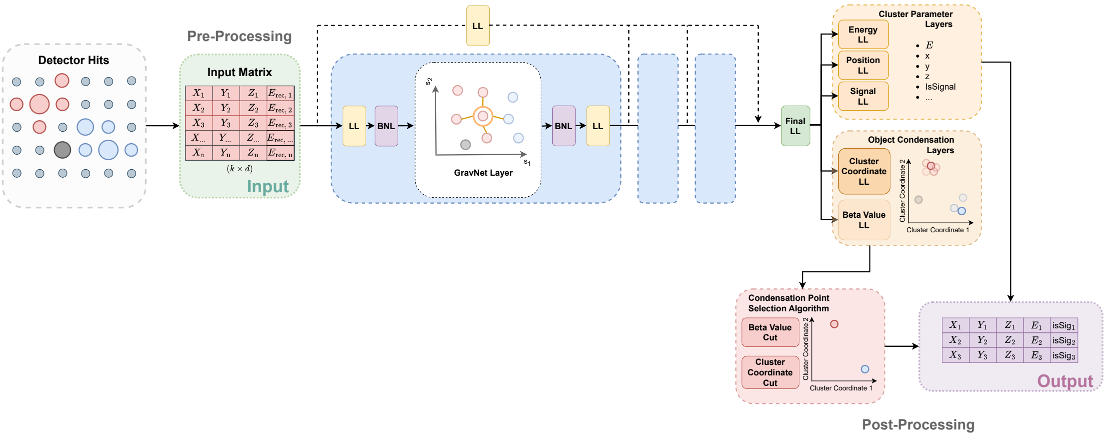


Figure 6.3: Overview of the inference chain from detector hits to cluster output values, including an overview of the model design. The input is provided in a matrix form and put through the network in a feed-forward pass. The output is predicted per node and is then input into the inference step, described in Section 6.2.2. The nodes remaining after the inference step are then the predicted clusters.

32-bit floating point precision, which poses no problem. However, in the case of a quantized network, the input does not necessarily have the same precision as the outputs of the repeat blocks, which could result in unwanted behavior. In Section 6.3, I will show different quantization implementations, where the scaling layer is used as an internal translation from one quantized value to another.

For the repeating block, the architecture is always the same. A first dense layer is used with an optional batch normalization layer afterward and a ReLu activation layer.

The batch normalization layer is a layer, which is used to normalize each batch by recentering the batch to zero and rescaling it to a standard deviation of 1. Batch normalization layers ensure more stable training and can help mitigate vanishing or exploding gradients, especially when high learning rates are used. During training, the batch normalization layer calculates the mean and the variance of each batch and calculates

$$Y = \gamma \frac{(X - mean(X))}{\sqrt{(var(X) + \epsilon)} + \beta}, \tag{6.9}$$

with X being the input to the layer and Y the output. γ and β are a learned scaling factor and a learned offset factor, respectively.

In inference, instead of calculating the mean and variance, a learned $moving\_mean$ and $moving\_var$ is used. These learned parameters are updated in every training step by

$$\begin{aligned} moving\_mean &= moving\_mean \cdot M + mean(X) \cdot (1 - momentum) \\ moving\_var &= moving\_var \cdot M + var(X) \cdot (1 - momentum) \end{aligned} \tag{6.10}$$

with M being a hyperparameter for the batch normalization layer, here set to 0.99. A more

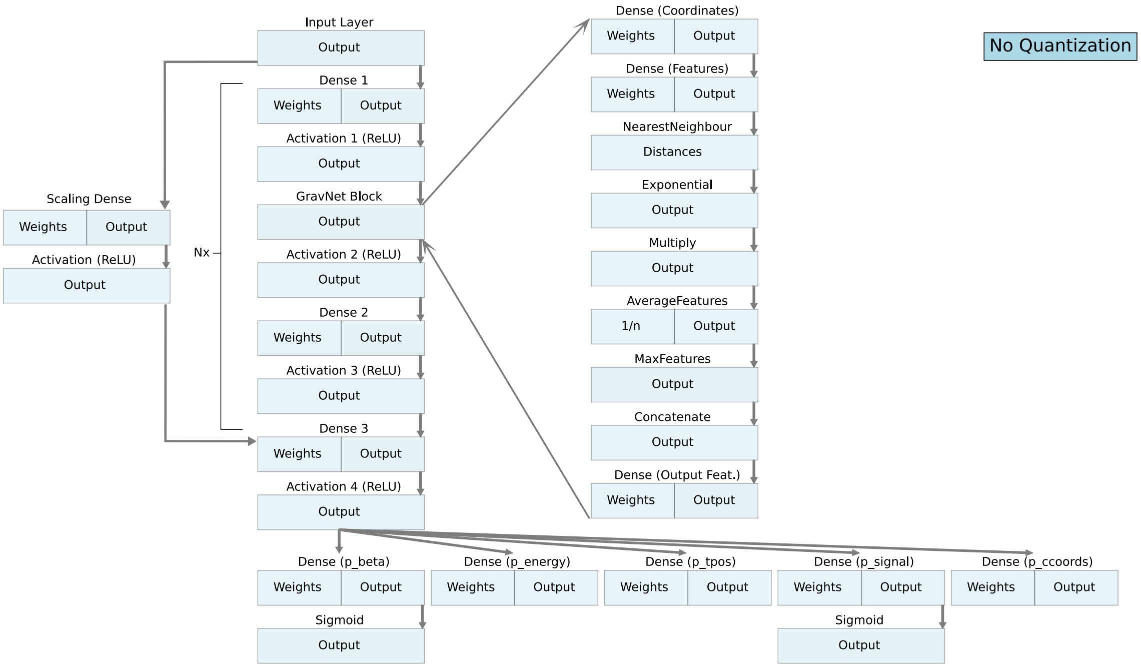


Figure 6.4: Full layer overview of the network design used in this thesis. The GravNet layer block is repeated N times. Each time, the output of one GravNet block is both input to the next GravNet block as well as concatenated to an output list, which is used as input to the last dense layer to employ skip connections.

typical network design is the ordering dense layer, activation function, batch normalization layer, whereas in this work the activation function is placed after the batch normalization layer. This is due to the necessary batch normalization fusing, explained in Section 6.5.

For all activation function layers besides the output layers, a ReLu activation function is used. This function is more easily implementable on hardware than other choices of activation functions, such as the tangens hyperbolicus, and has shown very good performance in a wide variety of machine learning models.

While the GravNet layer has a direct implementation in `Pytorch Geometric`, for `Keras` and `QKeras`, the layer has to be implemented separately. In Fig. 6.4, the layers needed for the full GravNet layer are shown in the right inset. A in-depth explanation to GravNet is given in Section 6.2.1.

The output of the full GravNet layer is then put through another batch normalization layer followed by a ReLu activation function. The repeat block finishes with a last dense layer and a ReLu activation function. The output of the repeat blocks is concatenated to the output list, together with the output of the scaling layer.

The number of times the repeat block is used within the network is a hyperparameter to be tuned. Every repetition introduces a message-passing step, which can help in aggregating knowledge per node. Nevertheless, for a hardware implementation, the overall size of the network has to be considered.

The output list is then fed into the last output dense layer, which can gather all information used in the network in one layer. This is passed through a ReLu activation function and then distributed into the five output layers. The output layers are the outputs for the β-value, the latent space of the OC algorithm and the prediction values for the energy, position, and signal classifier, respectively. The output values for β and the signal classifier have an additional sigmoid activation function to constrain the output values between 0 and 1.

6.2.4 Basic Training Setup

For training, an equal-size mixture of Category-One events (see Section 4.3.1) and Category-Two events (see Section 4.3.2) are used. For each sample, 40 000 events are simulated for 1-6 baseline photons, resulting in 240 000 events. Events without a TC above threshold or no reconstructed ECL offline clusters are discarded, resulting in a total of 460 000 events. The events are separated into a training and validation set, consisting of 90 % and 10 % of the total dataset, respectively.

The network is trained on the sum of all loss terms of the object condensation loss. For the main training, no weighting of the different loss terms is done. The training is logged and monitored via `Weights&Biases` [86], which is also used for the hyperparameter optimization in Section 6.4.4. For the training, both a learning rate scheduler to reduce the learning rate when reaching a loss plateau and early stopping to avoid overtraining are employed. For the learning rate scheduler, a decay rate of 2 is used if the total loss on the validation set did not improve over a set number of epochs, called patience. The minimum value for the learning rate, lr_{min}, is set to 10^{-7}. To make use of the full strength of the learning rate scheduler, early stopping is not activated before the minimum value is reached. This number of epochs N_{ES} is calculated by

$$N_{\text{ES}} = \lfloor \text{patience} \cdot \frac{\ln(lr_{\text{min}})/\ln(lr_{\text{start}})}{\ln 0.5} \rfloor, \tag{6.11}$$

derived from $lr_{\text{min}} = lr_{\text{start}}/2^n$, where n is the number of reduction steps to reach the minimum learning rate. After reaching this epoch, early stopping is triggered, when the validation loss has not improved over a further 50 epochs.

6.3 Quantization Optimization

Calculations on CPUs and GPUs are usually done in either 32 or 64-bit floating-point precision. This means a floating point number is represented by 32 or 64 bits in total, separated into a sign bit, a *significand* and an *exponent* [87]. An exemplary representation of 0.1 as a 32-bit floating point number in python is

$$0.1 = \underbrace{(-1)^0}_{\text{sign}} \cdot 1\underbrace{.10011001100110011001101}_{\text{significand}} \cdot \underbrace{2^{-4}}_{\text{exponent}} = 0.100000001490116119384765625. \tag{6.12}$$

In single precision, 8 bits are allocated for the exponent and 23 bits are allocated for the fraction, leading to a value range between $\pm\ 1.175 \times 10^{-38}$ and $\pm\ 3.4 \times 10^{38}$.

While calculations on CPUs most of the time use double-precision floating point numbers, which use a total bit width of 64 bits with 52 bits for the fraction and 11 bits for the exponent, calculations on GPUs are done with single precision to maximize multiplication speed. Therefore, frameworks such as `Keras` use single-precision numbers as default for implementations of machine learning algorithms to optimize training and inference speed on GPUs. For the multiplication of two floating point numbers, the two fractions have to be multiplied, while the exponents have to be added. Afterwards, the result has to be rounded and normalized to again be representable as a single-precision value.

For an implementation on FPGAs, the usage of floating point numbers leads to a very high resource usage, both due to the high bit width of 32 bits per number and to the flexible representation format of floating point numbers. For a multiplication of two single-precision numbers on an FPGA, the multiplication of the fractions is a 24bit x 24bit multiplication. This necessitates the usage of one or often even several DSPs slices. Additionally, the two exponents have to be added, the sign bits have to be subjected to a XOR operation, the result has to be normalized and rounded and overflows have to be dealt with. In general, this leads to a high resource usage of DSPs and LUTs and a reduction in processing speed due to the number of steps needed per multiplication, which is often 3-6 pipelining stages. In ML applications, where the number of multiplications is of the order of 10^5 or higher, this is often prohibitively costly.

To mitigate this, implementations on FPGAs often use fixed point representations for fractional numbers instead of floating point. In fixed point representation, the total bit width of the number is separated into a fixed number of bits for the value before the decimal point and a fixed number of bits for the value after the decimal point. As an example, in 32 bit fixed point representation with 10 bits before the decimal point including the sign bit, often written as Q10.22, the value 1.1 can be represented by

$$1.1 = \underbrace{0000000001}_{\text{integer (10 bits)}}.\underbrace{0001100110011001100110}_{\text{fraction (22 bits)}}. \tag{6.13}$$

This is calculated by separating 1.1 into the value before the decimal point and after the decimal point, which is 1 and 0.1 respectively. The first value, 1, is then represented in a

10 bit wide bit string. For the value after the decimal point, 0.1, this is transformed via

$$0.1 \rightarrow 0.1 \times 2^{22} = 0.1 \times 4194304 = \lfloor 419430.4 \rfloor = 419430 = 0001100110011001100110_2, \tag{6.14}$$

where 22 is the bit width allocated for the values after the decimal point. A representation in this format leads to a value range of [-512, 512) due to the leading bit necessary as a sign bit, with a precision of $1/2^{22} \approx 2.38 \cdot 10^{-7}$. Multiplications of two fixed point numbers with the same integer and fraction bit length on an FPGA is an integer multiplication of the entire bit length. Depending on the bit length, this either needs one DSP slice or a LUT and can be done in one clock cycle.

While some FPGAs models include specific support for floating-point arithmetic operations [88], algorithms deployed with fixed-point arithmetic still outperform floating-point calculations even on those platforms [89]. As an implementation of a GNN for the ECL L1 trigger has very stringent throughput and latency requirements, moving from floating point representations of weights, biases, inputs and outputs of neural networks to fixed point representations on the FPGA is a necessary step to enable implementation. Additionally, a reduction in number of bits per value is needed due to the limited availability of DSPs and LUTs on the hardware. This means that the values of the network's parameters are **quantized** and have a limited precision and range in comparison to the original single floating point precision used in offline reconstruction.

One method for quantization is post-training quantization, where the network is trained with floating point precision values and evaluated with the quantized value ranges used on the hardware. This is by definition lossy and leads to unwanted behaviour, especially if the parameters exceed the quantization range. An evaluation of this loss in performance is shown in Section 6.3.3.

An alternative is quantization-aware training, available through the `QKeras` framework [12]. In quantization-aware training, the behaviour of limited fixed point precision values is emulated. The network can then learn to stay in the boundaries of the quantization ranges, as well as mitigate the loss in precision. `QKeras` offers a wide range of quantization methods for the parameters which can optimize the performance of the network while staying within the given bit width. Each layer or operation can be quantized using a different quantization algorithm or different bit widths or value ranges for the parameters, making it highly flexible and optimizable. Each parameter is quantized during the forward pass of the network, while the quantization function is treated as the identity function in the backpropagation step to ensure differentiability. Additionally, `QKeras` provides quantized functions for standard activation functions such as the sigmoid function or the ReLu function.

While `QKeras` emulates the fixed point arithmetic as closely as possible as it is done in actual hardware calculations, two main differences remain. The first one is the behaviour when the result of a calculation exceeds the value range. In `QKeras`, the value is then clipped to the boundaries of the quantization range, which is necessary especially in training to keep differentiability and to help the network learn the boundaries of the value range. On hardware, as these are bit-wise operations, an overflow of the value range results in a wrap-around as only the least significant bits of the result are kept. This can result in undefined and unwanted behaviour and is not simulated in `QKeras`. The second difference, which is less likely to result in great differences, is the treatment of value rounding. While `QKeras` uses a straight-through estimator [90] to round to the nearest value in the specified precision range to ensure backpropagation, in hardware every value is automatically rounded towards negative infinity due to the cut-off of the least significant bits. This leads to slight differences in values, which in the worst case can propagate through the network. While for the first difference, a cross-check to keep all computed values away from the quantization ranges can help avoid this problem, the second difference cannot be fixed within the framework of `QKeras`. Due to this, the hardware behaviour cannot perfectly be simulated on CPUs and has to be checked separately as well.

6.3.1 Input and Output Quantization

As both the input and output values on the FPGA are quantized, the network has to be trained and evaluated with quantized inputs and outputs as well.

Input Quantization

The GNN-ETM board, on which the algorithm is implemented, receives the TC data from the ICN-ETM in a two-dimensional array with the first dimension being the TC IDs and the second dimension the TC features in the order (hit, energy, timing). The ID is the TC ID shown in Fig. 3.5, the hit flag shows if the energy is over 100 MeV, the energy is the energy after the fit in ADC counts and the timing is the absolute timing of the overall L1 trigger clock. During preprocessing, the x,y,z position is retrieved from a LUT using the TC ID, the energy is converted from ADC counts to GeV and the timing is calculated relative to the timing of the highest-energetic TC in that trigger data window. Due to this preprocessing, the quantization of the inputs can be set from outside and implemented within the given LUTs. This also ensures that no overflow can happen when exceeding the quantization range due to clipping.

To simplify input value handling, all inputs are quantized with the same quantization ranges. The network is trained with the quantized values. The quantization value is chosen to be Q4.12, using 16 bits in total with 3 bits for the value range before the decimal point and an additional sign bit. This leads to a value range of -8 to 7.9 with steps of $1/2^{-12} = 0.00024414$. As can be seen in Fig. 6.5, this quantization has nearly no effect on the input values. For the energy, a clipping of all energies above 8 GeV is visible. The ICN-ETM provides the energy in ADC counts between 0 and 4096. One ADC count corresponds to 2.25 MeV, leading to a range between 0 and 21.5 GeV. In practice, clusters above 8 GeV are nearly never seen due to the total available energy of 10.58 GeV and should

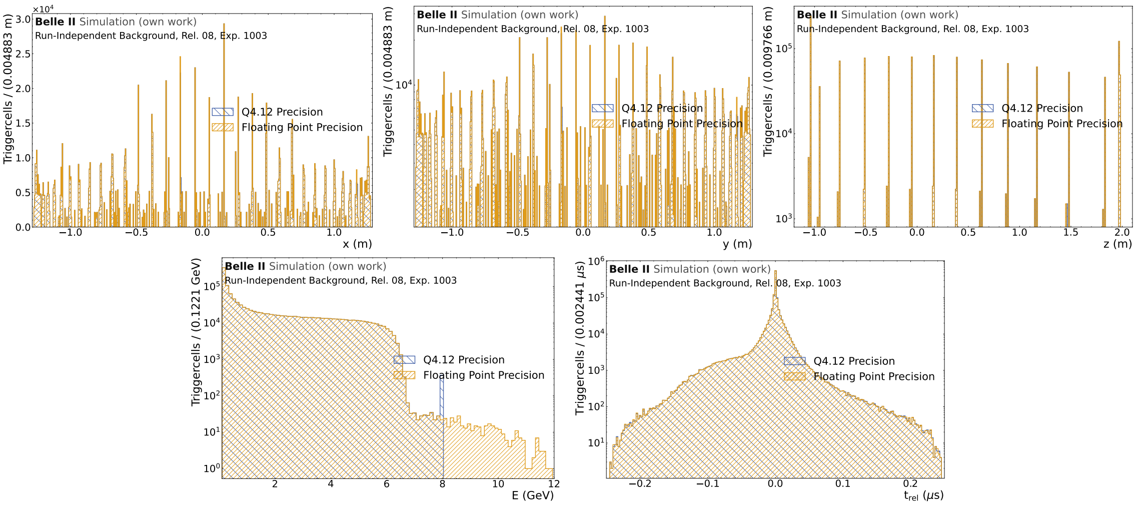


Figure 6.5: Distributions of input values with floating point single precision in comparison to the same values with Q4.12 fixed point precision. All values are quantized with the same fixed point quantization. The dataset is the Category-One test sample using 15000 events. The distributions are hatched to show the overlap. The top row shows the x, y, and z distributions while the bottom row shows the energy and the relative timing. The bins are chosen as multiples of the quantization step to make differences more visible.

result in a trigger decision regardless of their exact energy.

For future studies, the tuning of the input quantization is a point of optimization for the overall implementation performance of the network. Decreasing the number of bits needed for the inputs can result in lower latency and lower resource usage, but this has to be studied in detail.

Output Quantization

The quantization of the output values can be optimized for the range of the output values as well as for the needed precision. As the overall number of calculations with the output values is small in comparison to the total number of calculations within the network, it is feasible to allow large bit widths for the output values to guarantee no loss of information. A different restriction for the output quantization is the maximum bandwidth allowed by the overall L1 trigger system. Higher bit lengths of the output values increase the maximum data size needed to be transferred via the Belle2Link, which should be taken into consideration. A more detailed description of the bandwidth requirements can be found in Section 6.5. Additionally, the output values are, especially in the development phase, not solely used on the trigger system but also read out via `basf2`. Programming languages such as `C++`, which is the main programming language of `basf2`, handle data in bytes instead of bits. Handling output values in `C++` that are not multiples of 8 bits is then error-prone and leads to more complicated implementations of the unpacking code. Therefore, the output values should be quantized in multiples of 8 bits as the total bit length. The same argument

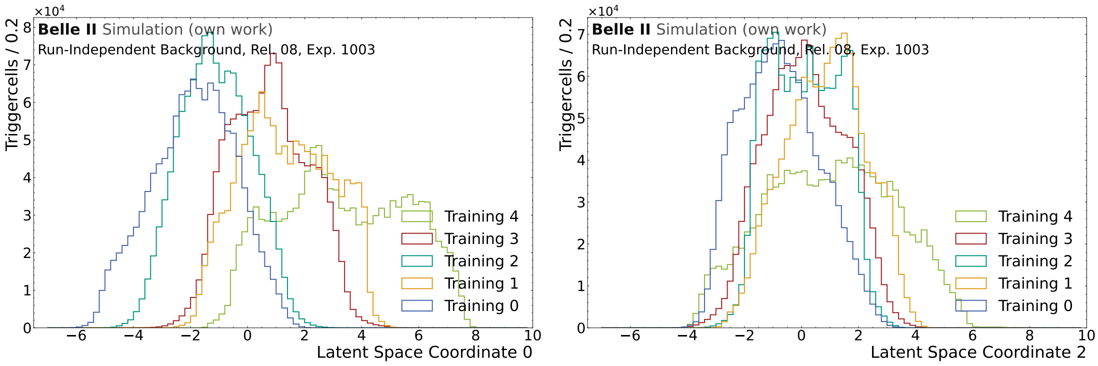


Figure 6.6: Distributions of the output values for the first and third dimension of the object condensation latent space. Five networks with the same configuration and training setup were trained and their output features on the Category-One test dataset evaluated. The value ranges of the three dimensions vary significantly between different trainings. All output values are shown here, regardless of the nodes being selected as condensation points by the condensation point selection algorithm.

holds for the input values, as they are also stored in the raw data and read out by `basf2`. For the input values, a downcast to lower bit lengths would be possible, whereas an upcast to higher bit lengths for the output values would result in empty information.

Predicting value ranges for the output features is slightly more difficult than for the input features. The output feature range for the energy and position predictions can be determined by the range of the energy and position values of the offline ECL target clusters (see Section 4.2.2). For the β value and the signal background classifier, the additional sigmoid activation function after these output layers restricts the total value range between 0 and 1. The most difficult output value range to gauge is the coordinate ranges of the latent space. While the network could in theory learn to stay within the given boundaries for those features, the range should be chosen so that no overflow can happen to reduce the risk of unwanted behaviour. Therefore, several networks without any quantization are trained and their maximum output feature range is determined. In Fig. 6.6, the distributions for all three latent space dimension outputs can be seen. The trainings were done with the exact same configurations and datasets. As visible in the figure, the output distributions vary significantly between different trainings. In addition, the value ranges for the latent space are comparable to the value ranges of all other output features, which results in the quantization range being determined by the latent space.

As the overall bit width should be in multiples of 8 bits and the same quantization should be chosen for all output features, a total bit width of 8 is very difficult to achieve. To safely encompass the latent space feature range, the number of bits before the decimal point should be either 4 or 5, which results in a range of [-16, 16) and [-32, 32), respectively. As one bit is reserved as the sign bit, this would lead to either 3 or 2 bits left for the value after the decimal point, which results in steps of 0.125 or 0.25. Values like the β value, the signal/background classifier output or the energy scaling factor are either bound between 0 and 1 or very close to 1, this would result in a loss of information.

Therefore, the overall bit width is chosen to be 16 bits. To ensure containment of the latent space feature range, 5 bits are used before the decimal point, leading to an overall quantization of Q6.10. While 4 bits would suffice for the ranges shown in Fig. 6.6, there is no set restriction on these value ranges. Therefore, an extra bit is allocated to ensure no overflow.

6.3.2 Quantization of Weights, Biases and Activation Functions

Deciding the quantization precisions for each layer and step of the network can be done in two ways: first, through a hyperparameter optimization. This optimization has to take the available hardware resources into account, making each layer's quantization widths highly dependent on the widths of the other layers and on other aspects, such as the pruning percentage (see Section 6.4.1) and the exact network design. This creates a very high-dimensional optimization problem, in which upper bounds for the overall parameter space are challenging to calculate. Each configuration therefore has to be tested in implementation. For this work, as the network is still in active development, the second design path was taken, which is a bottom-up approach. The quantization is first set to small bit widths and then increased or adapted, where necessary. For further development of this algorithm, the quantization procedure can be optimized.

The final network design can be seen in Fig. 6.7. This contains all layers with their respective quantization values for the model design that was used for implementation.

To reach this final design, the quantization is first defined as Q3.5 for all layers and then increased for strategically important points to improve the network's performance. For all dense layers, the weights, biases, and outputs for one layer are quantized with the same quantization width. In general, the output of the layers is not quantized itself if they have a ReLu activation function following, which will then be quantized.

As the quantization precision of the output values has been set to Q6.10, as described in Section 6.3.1, the weights and biases of the output dense layers are also set to a total length of 16 bit. However, as both the signal and β output values are to be between 0 and 1, the range of the corresponding layer weights is reduced to Q3.13 to allow for more precision in the calculation of these values. For all other output layers, the weights and biases are set to Q11.5.

Simplifications for the graph-building step, such as implementing the Manhattan distance instead of the Euclidean distance (see Section 6.4.2), can decrease the precision of the overall calculation. Therefore, more bits are allocated for the layers necessary for the GravNet step. The dense layers for the coordinate space S and the feature space F_{LR}, as well as the aggregated features used for message passing have a quantization precision of Q2.14 applied. In Fig. 6.8, the weights of the final training for these two layers, the coordinate space layer (left) and the feature space layer (right) are shown. The values are all very small and clustered around 0, for which a high number of bits after the decimal point can

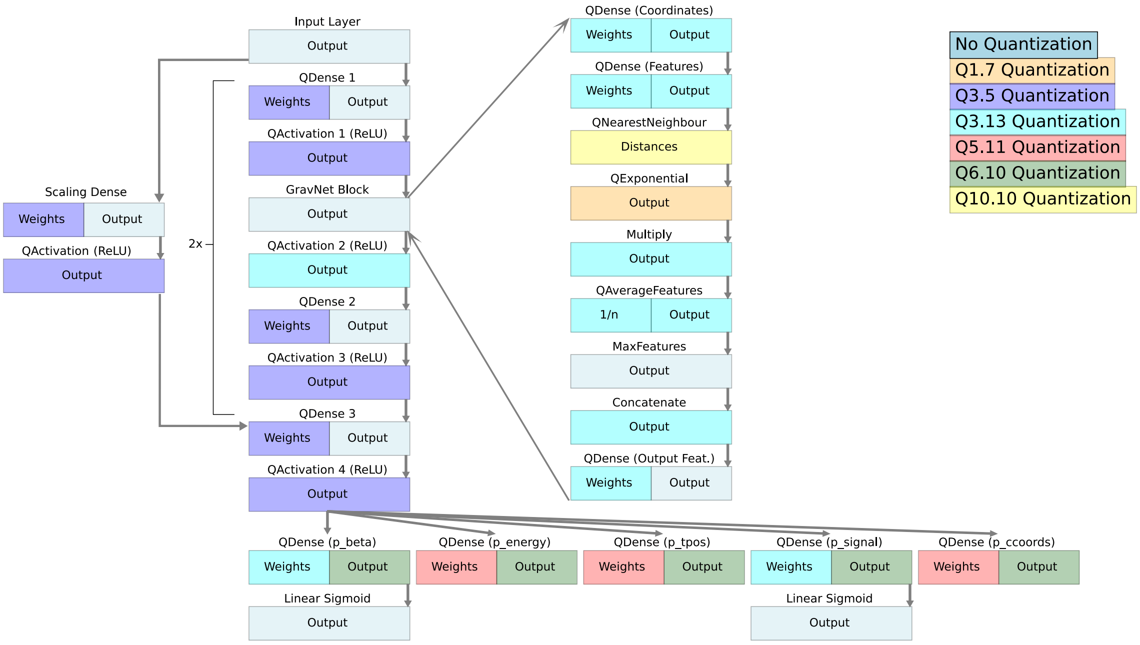


Figure 6.7: Network design including quantization for the overall model. The forward pass of the network follows along the arrows with the core part repeated twice. The inset describes the GravNet block. The coloring corresponds to the different quantizations.

help in increasing the calculation precision.

The distance calculation between the nodes in the coordinate space S is done via several difference calculations. This can, as the coordinate values can be both negative and positive, lead to up to double the value range for the output than for the input. The quantization for the distance calculation is therefore increased drastically. This is a safety measure to avoid overflow on the actual FPGA implementation, as this is not modeled in QKeras and can lead to unpredictable behavior.

In practice, the distances calculated in the graph-building steps are very small. In Fig. 6.9, the distances for the first and the second GravNet block can be seen. Distances to padding nodes, which are the 0-values appended to the input vector to have a fixed 32-length input vector, are set to 100. All valid distances are very close to 0. For next steps, a reduction in range for the distance output is possible.

In general, the weights, biases, and outputs of each layer and calculation of the network have to be checked for each new training. If a large number of values are saturated, the quantization ranges have to be optimized.

Additionally, the difference between the behavior of QKeras and the actual implementation on an FPGA has to be taken into account. In Fig. 6.10, the output of the final layer of the second GravNet block can be seen on the left, with the output of its ReLu activation function on the right. The ReLu output is quantized with Q3.13, having a total value range of [-4, 4]. The output of the ReLu is contained in this range, the output of the dense layer, which is not explicitly quantized, exceeds this range in negative direction. This

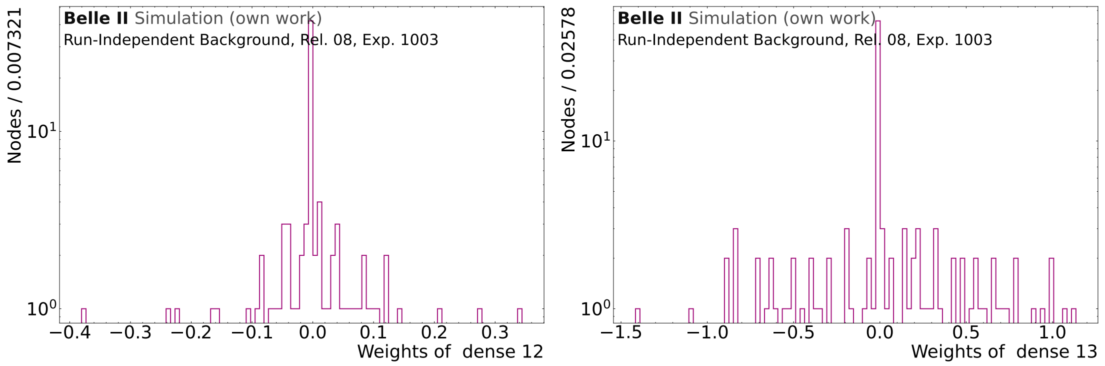


Figure 6.8: Weight values for the coordinate space layer (left) and the feature space layer (right) for the first GravNet block in the final, quantized training. The weight values are close to 0, where a high number of bits after the decimal point can help improving the precision.

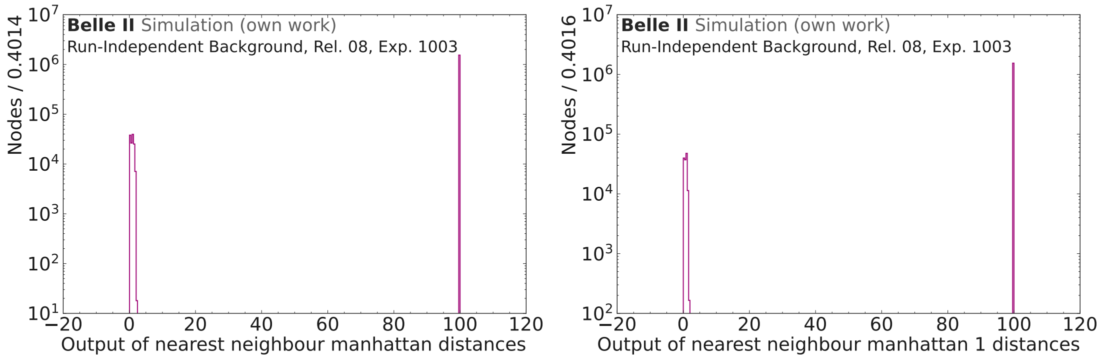


Figure 6.9: Distance outputs of the kNN step of the GravNet block for the first GravNet block (left) and the second Gravnet block (right) in the final, quantized training. The distance to invalid nodes, which are the nodes added to the input vector to pad the input to 32 nodes, is set to 100.

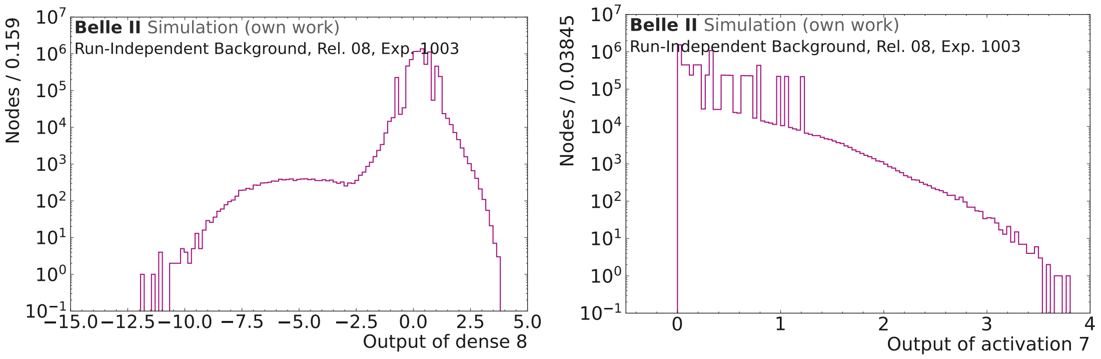


Figure 6.10: Output of the output feature dense layer of the second GravNet block (left) and the output of the following ReLu activation function (right) of the trained, quantized network. The dense layer exceeds the quantization range of [-4, 4] in negative direction, which is absorbed by the ReLu function.

does not pose a problem in `QKeras`, but will lead to an overflow and undefined values on the hardware. A solution used in this work is the implementation of dense-saturate layers, where the output of each layer on the hardware is clipped to its maximum values instead of allowing overflows.

6.3.3 Post-Training Quantizations

Training quantization-aware is necessary to ensure good training performance, as shown in Section 6.3.2. Nevertheless, not all operations can or need to be quantized during training. During training, the differentiability of all steps of the network has to be ensured to correctly apply the backpropagation algorithm. Additionally, some quantized functions introduce an instability into the training procedure, which can be mitigated by quantizing them only during inference.

Output Quantization of the Exponential Function

The GravNet algorithm, as described in Section 6.2.1, dynamically builds the graphs in a learned latent space. The message passing between the connected nodes is done by aggregating the features of a node and weighting these features by the exponential of $-A \cdot \Delta x$, where A is a tunable factor. The quantization of the exponential function is a quantization of the output value. Due to the nature of the exponential function, this leads to very small values very quickly. Floating point numbers are able to represent very small numbers up to 10^{-38}, whereas in comparison 32-bit fixed point numbers with all bits after the decimal point can only represent 10^{-10}. While in most practical use cases numbers smaller than 10^{-10} can be set to 0, during backpropagation this leads to a loss of information when calculating the derivative at this point. Due to this, the quantization of the exponential function in training leads to a complete efficiency loss for all values below a varying energy threshold, often around 1 GeV and a very bad energy and position

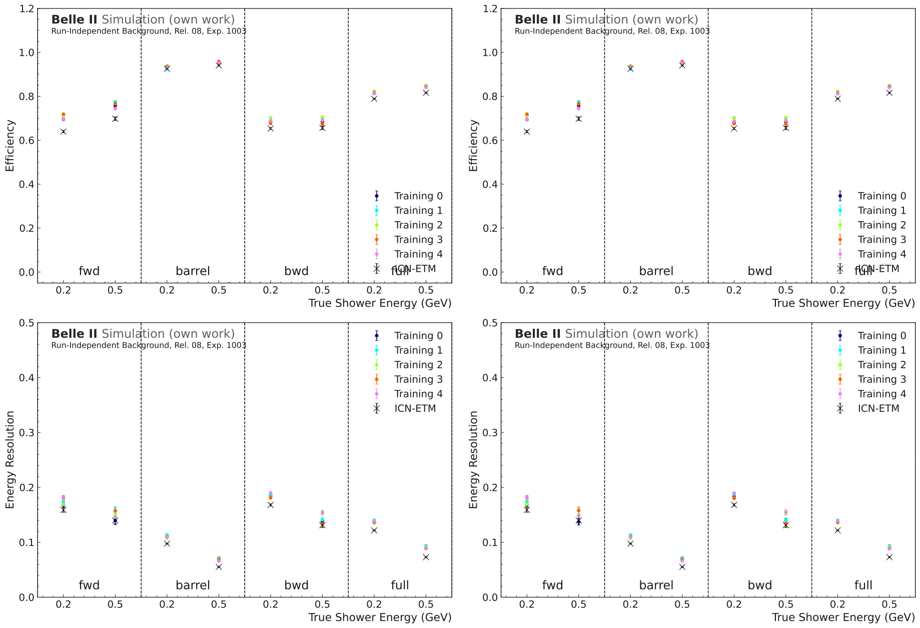


Figure 6.11: Comparison of the performance of five trained networks without quantization of the exponential function (left column) and with Q0.8 quantization of the exponential function. Both evaluations are done with the Category-One test dataset. The top row exemplary shows the efficiency while the bottom row shows the energy resolution. All plots are separated into the different detector regions and two energy points, 200 MeV and 500 MeV, are evaluated.

resolution for the entire energy range.

To mitigate this behaviour, the output of the exponential function is not quantized during training but kept as a floating point value. This ensures a correct training performance. In inference, the exponential function is then quantized with an unsigned quantization of Q0.8, leading to a precision of 0.003906.

In Fig. 6.11, the evaluation of several networks without quantization of the exponential function output versus the evaluation of the same networks with quantization of the exponential function is shown. Exemplary, the efficiency and the energy resolution for all different networks is shown, for all detector regions and for two energy bins, 150 - 250 MeV and 450 - 550 MeV. The performance is identical, which shows the negligible effect of the quantization of the exponential function in inference.

Substituting the Sigmoid Function

The sigmoid function defined as $(1+e^{-x})^{-1}$ is a commonly used activation function for output layers where the output is restricted to be in the range between 0 and 1. For this work, the β and signal classifier output layers both use a sigmoid activation function to enforce the value range. To calculate the output of the sigmoid function, an exponentiation and a division is necessary. While an exponentiation in a restricted input range is possible to describe on an FPGA via a LUT, this is a very resource-intensive operation for the

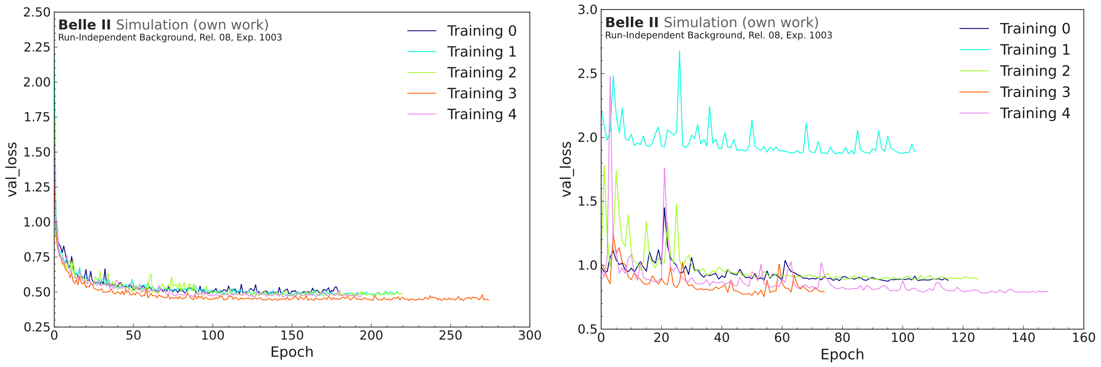


Figure 6.12: Comparison of the loss improvement for five trainings with a standard sigmoid activation function (left) and a `smooth sigmoid` activation function (right). In the right plot, several spikes and a higher overall value loss function can be seen, which hints at unstable training due to non-ideal weight adaption.

entire value range. Divisions, especially with higher bitwidths such as the 16 bits for the output values, are difficult to implement with smaller resource usage.

To improve this implementation difficulty, the sigmoid function can be replaced with different linear approximations. In this case, I choose the by `QKeras` provided `smooth sigmoid` approximation. This function is described by the linear function $0.1875x + 0.5$ in the center part and a clipping of the values to 0 as a lower bound and 1 as a higher bound.

Using the `smooth sigmoid` in training results in higher training instabilities. In Fig. 6.12, trainings with a standard sigmoid activation function for β and signal output layers are compared with trainings with a `smooth sigmoid` activation function. Both training procedures are otherwise identical. The loss function of the training with the linear approximation shows several high spikes and overall more erratic behavior. This is due to the fact that the gradient vanishes in the flat regions that are below -2.67 or above 2.67, which can lead to dying neurons [91]. If the gradient of a neuron vanishes, no updates will be given to the weights of this neuron and no improvements to its prediction performance are made. The network has to learn to stay within the range of the linear function, which is an added difficulty during training.

To mitigate this behaviour, the network is trained with the standard sigmoid function and only during inference the `smooth sigmoid` is used. This has no effect on the energy and position prediction, but affects especially the signal classifier and in a lesser extent the β value.

The effect on the signal/background classification can be seen in Fig. 6.13. The `smooth sigmoid` activation function has a very high probability of mapping values to exactly 0 or 1 in comparison to the standard sigmoid function. This leads to the high peaks at the edges of the distributions for both true signal and true background clusters. In Fig. 6.14, the ROC curves for five different trainings with the standard sigmoid on the left and the `smooth sigmoid` on the right can be seen. The signal classifier distribution in Fig. 6.13 is that of

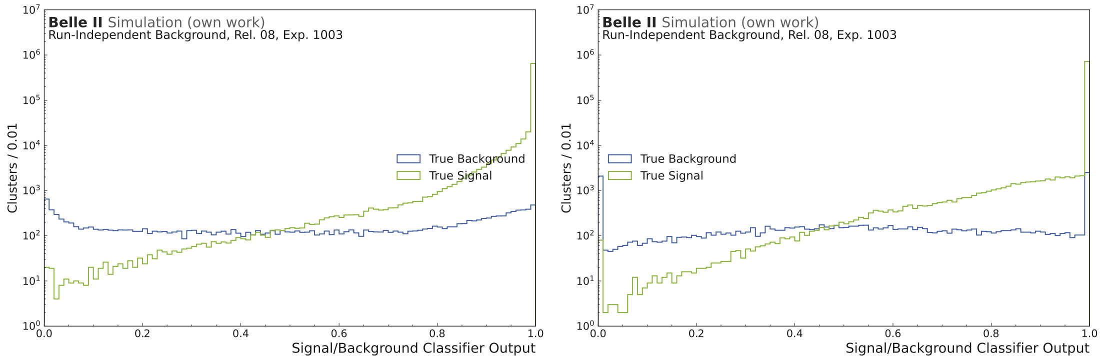


Figure 6.13: Comparison of the signal classifier output value for predicted clusters matched to target clusters. The distributions are separated into true signal clusters, where the target clusters are defined as signal, and true background clusters. The dataset is the Category-One test dataset. The left plot shows the distributions with a standard sigmoid activation function and the right plot with the `smooth sigmoid` activation function. The pronounced peaks at 0 and 1 in the right plot are due to the clamping of the linear approximation.

training 3. As can be seen, while the effect of the approximation of the sigmoid function is visible towards the left and right edges of the ROC curves, a high signal efficiency with a high background rejection can still be achieved. Working points at 99 % or 95 % signal efficiency have the same background rejection rate. This can of course change when the signal classifier distribution changes and has therefore be monitored carefully.

A second implication is the loss of tuning probabilities for the OC algorithm. For the OC inference, two hyperparameters have to be set, the threshold for the β value for each point to be considered a condensation point and the isolation criterion distance threshold. In Fig. 6.15, the network prediction of the β value for all nodes is shown in the case of the standard sigmoid activation function versus the `smooth sigmoid`. As the β output of the network is also subjected to the sigmoid activation function, the same shift towards exactly 0 or 1 as in the case of the signal classifier can be seen here as well. In the case of the `smooth sigmoid` inference, 36 % of all nodes have a value of exactly 0, which automatically declassifies them as condensation points when setting any threshold cut besides 0. For the standard sigmoid function, no value is exactly 0, but 27 % of nodes have a value below 10^{-6}, which can be regarded as 0 in the context of setting a value for the β threshold. This leaves a difference of up to 10 % of all nodes to potentially classify as clusters.

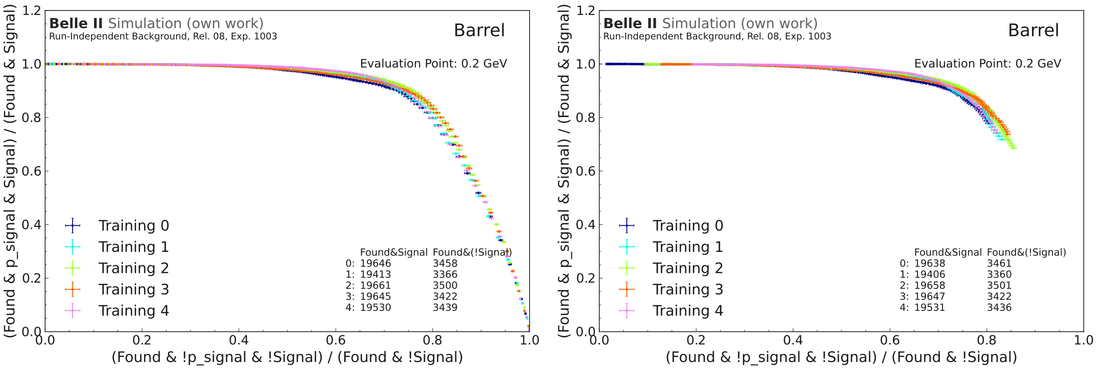


Figure 6.14: ROC curves for five trainings for the signal classifier for clusters in the Category-One test dataset with a true reconstructed energy between 150 and 250 MeV in the barrel region of the ECL. The left plot is evaluated with the standard sigmoid activation function and the right plot with the `smooth sigmoid` function. As only predicted clusters that are matched to target clusters are shown here per training, the number of matched true signal and true background clusters per training is indicated on the plot. The cutoff of the ROC curves in the right plot towards the very right and very left is due to the amassment of the values at 0 and 1.

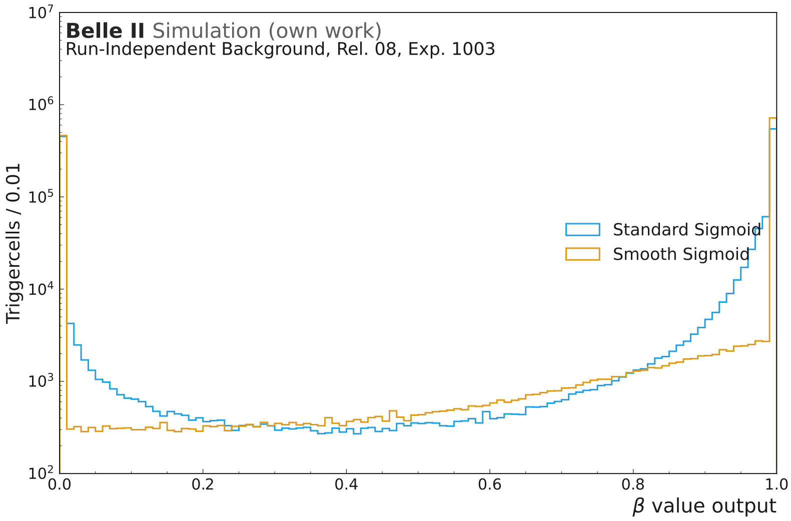


Figure 6.15: Comparison of the β value output when using the standard sigmoid activation function and the `smooth sigmoid` activation function. The β value is taken from all nodes in the Category-One test dataset before the condensation point selection algorithm. The `smooth sigmoid` output pushes most points towards exactly 0 or 1, removing the steady increase towards the edges of the distributions.

Batch Normalization Layer Fusing

When using batch normalization layers for networks on FPGAs, for inference the calculation of these layers should be fused into the dense layers beforehand to avoid extra multiplications. To fuse the four parameters of a batch normalization layer, $mean$, β, Var and ϵ, as explained in Section 6.2.3, into the dense layer beforehand, the following calculation has to be done to store the new weights:

$$w' = \gamma \frac{1}{\sqrt{\mathrm{Var} + \epsilon}} w \quad b' = \gamma \frac{b - mean}{\sqrt{\mathrm{Var} + \epsilon}} + \beta, \tag{6.15}$$

with w and b being the original weights and biases of the dense layer and w' and b' the updated weights and biases.

As this fusing happens after the quantization-aware training, the weight values can now exceed the original quantization range. This is very difficult to mitigate beforehand and can lead to undefined behavior on the hardware. For the final GNN-ETM network, batch normalization layers have been omitted to avoid this behavior.

6.4 Architecture and Input Optimization

While quantization of all parameters used in the network is necessary for usage on hardware, the overall size of the network has to be additionally restricted to fit within the hardware constraints. As mentioned before, the depth of the entire network, the size of the layers, the parameter bit widths and the percentage of pruning all have to be optimized in tandem with the maximum upper bound being difficult to determine in absolute values. When staying with the design of one GravNet block shown in Fig. 6.3 to limit the size of the design space, an absolute limit of the number of blocks is two due to the routing and latency constraints on the hardware.

In this section, the optimization on the overall number of parameters and on the input and output scaling is shown. Low-magnitude pruning is explained in Section 6.4.1, the impact of input scaling and the use of Manhattan distance instead of Euclidean distance is shown in Section 6.4.2, and the optimization of the energy prediction is shown in Section 6.4.3. For these sections, the final network design with the parameters shown in Table 6.1 is used. A hyperparameter optimization including all quantizations, the final pruning procedure, and the input and output scaling is shown in Section 6.4.4.

While every dense layer within the network, including the dense layers for the representation space S and the feature space $\mathrm{F_{LR}}$ shown in Fig. 6.1, can be optimized separately, for simplification, several hyperparameters are combined into one to make optimization easier.

Table 6.1: Summarized model parameters, their descriptions, and final values used for the evaluation. The final number of free model parameters is 4803.

Parameter	**Definition**	**Final Value**
Dim1	Size of the dense layer before the GravNet layer and the layer before the final output layers	16
Dim2	Size of the dense layer after the GravNet layer	16
momentum	Momentum for the batch norm layers	0.6
$\mathbf{N_S}$	Size of representation space S	6
$\mathbf{N_{F,LR}}$	Size of the feature space $\mathrm{F_{LR}}$	8
k	Number of nearest neighbours in the GravNet graph building	8
$\mathbf{f_{exp}}$	Scaling factor for the exponential weighting function	10.0
$\mathbf{N_{out}}$	Size of the dense output layer of the GravNet algorithm	32
$\mathbf{N(coords_{LS})}$	Number of coordinates for the OC latent space	3

6.4.1 Low-Magnitude Pruning

Low-magnitude pruning [92] is a model compression technique designed to remove less salient connections from the network to increase model sparsity. The goal of pruning is to reduce the overall number of parameters by setting the weights of a set percentage of nodes to 0 during training. The model can learn to be robust against these missing connections and have negligible loss in performance. Nodes with their weight set to 0 can be skipped during inference and are not needed to be stored on the hardware, reducing both the number of overall multiplications and the amount of needed storage.

In low-magnitude pruning, a binary mask is added for all layers subjected to the pruning algorithm. In each pruning step, the nodes of all layers are sorted by absolute weight and the mask value for each node is set to 0 from the bottom up until the required percentage is reached. In the backpropagation step, the backpropagation is subjected to these binary masks and the weights of the nodes are not updated in this step but kept at 0. For this work, the `Keras Constant Sparsity` pruning scheduler was used. This scheduler sets a given percentage of weights per layer to 0 at steps in a given interval during training. Additionally, the start and end epoch of the pruning can be given to the scheduler.

The overall goal of pruning is to reduce the amount of parameters as much as possible without a loss in network performance. For the given architecture and parameter settings given in Table 6.1, pruning percentages above 40 % turned out to be difficult to achieve by the pruning scheduler. Therefore, 40 % was used throughout this work.

The start of the pruning schedule is a hyperparameter and depends on the stability of the model and on the current learning rate. When the learning rate is already at a very small value, removing a high percentage of weights might mean that the updates due to the

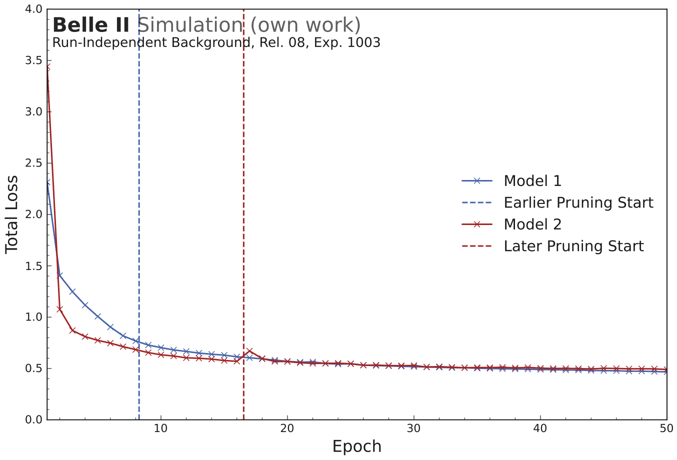


Figure 6.16: Comparison of the loss evolution of two networks trained with the same configuration besides the start of the pruning scheduler. The start of the pruning is indicated with the dashed vertical lines, with the same colours corresponding to the related loss curve. The later pruning start results in a temporary increase of the loss function, but the network still has enough learning capability to achieve the same loss after a recovery period than the network where the pruning started at the earlier point. The markers are connected by straight lines to guide the eye.

learning rate are too small for the model to recover from that loss of information. On the other hand, when starting too early, the weights might not be optimized already and the pruning removes necessary information for the further learning process. In Fig. 6.16, two pruning start points are compared for the same network design and training setup. The later start induces a temporary rise in the loss function, but the network can recover and stabilize the training again. Both networks still have a learning rate of 0.0005 at this point.

A drawback with the usage of the pruning algorithm implemented in `Keras` is the combination with the model checkpoint callback. When saving a model not at the end of training, but every time a given metric, e.g. the validation loss, reaches a new minimum, `Keras` does not guarantee the model weights being saved in their pruned state. This means that every training has to be checked in terms of overtraining and terminated early enough to avoid an increase in the loss function.

6.4.2 Input Scaling and Distance Optimization

Further optimizations can be tested to either improve the model performance or to facilitate the implementation on hardware.

Input Scaling

ML algorithms usually converge better or faster when the inputs and outputs are normalized, either to a range of [0, 1] or [-1, 1], to avoid large gradients. Additionally, all input features should have the same scale to facilitate the learning process [93]. In Table 4.1, the input

Table 6.2: Scale, offset and final value range for the normalization of the input variables for each TC to the network. This is done for the entire dataset.

Input variable	Offset	Scale	approx. Range
Reconstructed Energy	0	8	[0, 1.5]
Relative Time	0	0.25	[-1, 1]
x Position	0	1.27	[-1, 1]
y Position	0	1.27	[-1, 1]
z Position	0.46	1.52	[-1, 1]

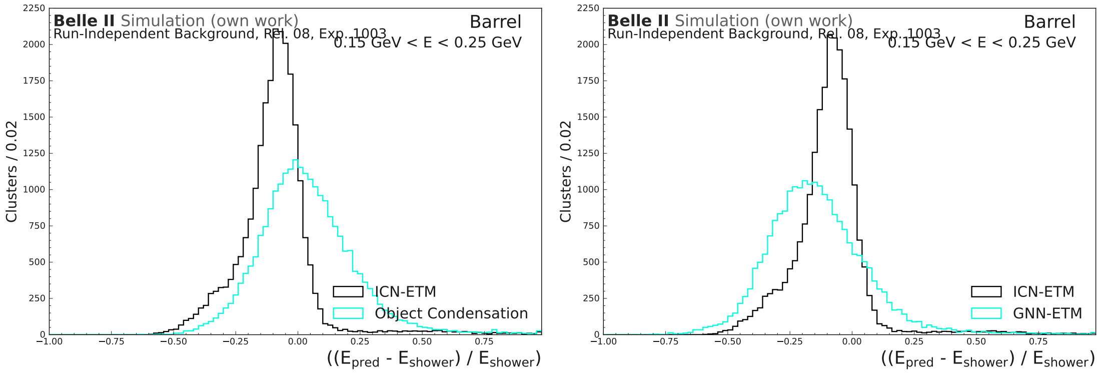


Figure 6.17: Comparison of the energy resolution for offline ECL clusters in the barrel region with $0.15 < E_{\text{offline cluster}} < 0.25\,\text{MeV}$. The left plot is a training trained with unscaled inputs, the right plot with scaled inputs.

features with their expected range is shown. While all features are already in a similar scale, encompassing an entire value range of -1.5 and 12, timing and energy can for example be different by almost two orders of magnitude. Therefore, a comparison between network performance for unscaled input features versus scaled input features is done.

The features are scaled by applying the operation $(x - \text{offset})/\text{scale}$. In Table 6.2, the offset, scale, and final value range for each feature are listed. While the energy can in theory go up to 21 GeV as the maximum value for each TC, which would exceed the normalized range, the network should focus on the lower energies. When setting the scale to much higher values, the lower energies would lack resolution, which is an unwanted effect.

Fig. 6.17 shows an exemplary energy resolution for one training trained with nonscaled inputs and one training trained with scaled inputs. There is no improvement seen by scaling the inputs.

Manhattan Distance

The standard way to calculate the distance between two points in a multi-dimensional space is by using the Euclidean distance or L2-Norm

$$d_E(p,q) = ||p - q||_E = \sqrt{\sum_i (p_i - q_i)^2}. \tag{6.16}$$

For the here used network, a distance calculation is necessary at two points. In the GravNet layer (see Section 6.2.1), the distance between two points in the representation space S is used for the kNN algorithm to build the graphs, as well as a weight factor for the message passing between two connected nodes in the graphs. Furthermore, the OC inference calculates the distances between points in the latent space to select condensation points.

While the calculation of the square root is not necessary for these use cases, the calculation of the square is quite resource-intensive due to the multiplication and the necessary bit width increase in the accumulator. Therefore, instead of using the Euclidean distance, I use the Manhattan distance to simplify the distance calculation on the FPGA. The Manhattan distance or L1-Norm is defined as

$$d_M(p,q) = ||p-q||_M = \sum_i |p_i - q_i| \tag{6.17}$$

and does not include any multiplications for the calculation.

All distances in the network are substituted by the Manhattan distance and the network is trained with these distances.

Training a network with one distance and solely evaluating with the other does not work, as especially the GravNet layer is finetuned on the absolute sizes of the calculated distance. The steep slope of the exponential weighting function leads to large differences when substituting the Euclidean distance with the Manhattan distance or vice versa and results in unsuable models. Therefore, the used distance has to be consistent in both training and inference.

6.4.3 Energy Prediction Optimization and Network Stabilization

While the OC prediction values, the β value and the latent space coordinates, are set due to the algorithm design, which and how the cluster features are to be predicted is a design choice. In the original design, the network predicts the absolute value of energy and position and the loss is designed as the Mean Absolute Error (MAE) loss. In Fig. 6.18, the energy resolution for 150 - 250 MeV and 450 - 550 MeV and the θ and ϕ resolution for 150 - 250 MeV for a quantized network in comparison to the resolution of ICN-ETM is shown. Both position and energy is predicted as an absolute value with the MAE loss. As the figure shows, while the position resolution is the same for both ICN-ETM and the GNN-ETM network, the energy resolution of the network is significantly worse. As explained in Section 3.2, the ICN-ETM algorithm uses an energy sum for the prediction of the energy and the center position of the highest-energetic TC as the prediction of the position. It is to be expected, that the network cannot improve especially the position prediction in these low energy regimes, as the shower is entirely contained within one TC and further position information is lost before the network receives the input. But while the same argument holds also for the energy prediction, a worsening of the energy resolution cannot be accepted if the network is to be used in the real trigger system.

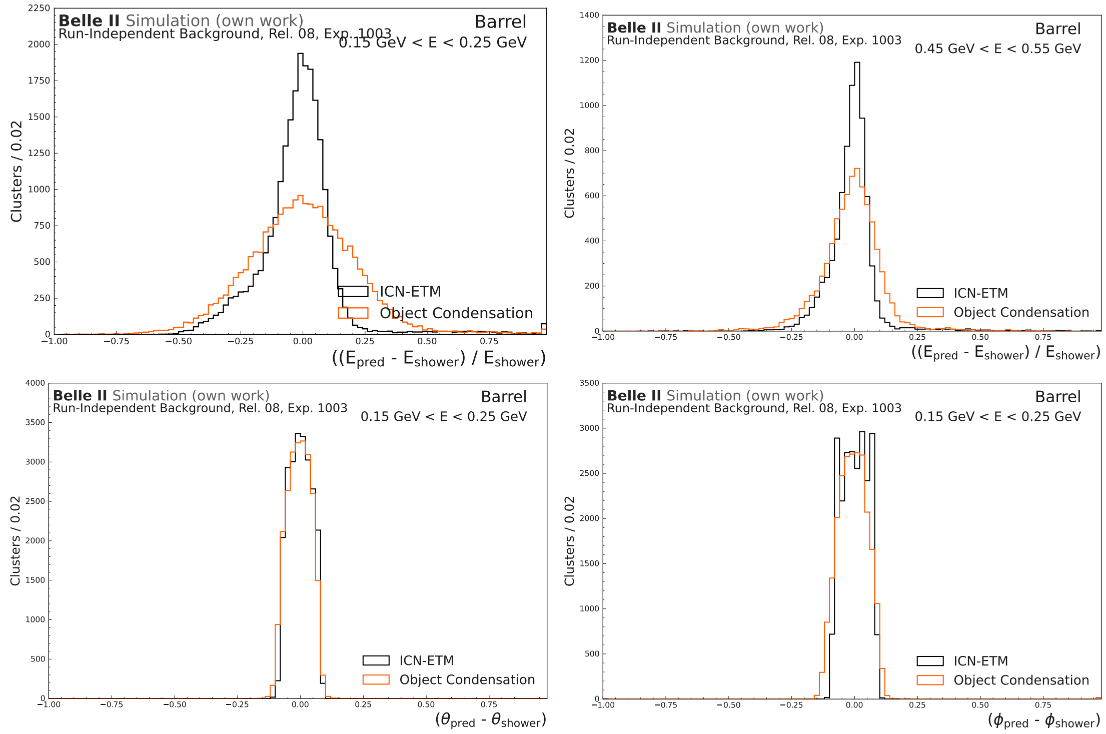


Figure 6.18: Energy resolution for clusters in the barrel region with 150 - 250 MeV (top left) and 450 - 550 MeV (top right) and θ (bottom left) and ϕ (bottom right) resolution for 150 - 250 MeV for one training with the network design and quantization explained in Section 6.3.2 and with an absolute energy prediction, compared to the resolution of ICN-ETM. The resolutions are bias-corrected as explained in Section 5.2. The dataset used is the Category-One test dataset.

An additional drawback is the instability of the network training and the respective network performance. In Fig. 6.19, the ROC curves for the signal/background classifier of five quantized networks trained with the identical setup described in Section 6.2.4 and evaluated with the Category-One test dataset are shown. The ROC curves are drawn for matched offline ECL clusters with a true energy between 150 - 250 MeV. As can be seen, three networks do not learn anything in regards to the signal/background classifier, while their overall efficiency indicated by the number of found offline ECL clusters in the lower right corner is similar to the other two networks. This training instability means that always several networks have to be trained with the same setup to increase the chances of good training results.

To improve the energy resolution for low energies, the loss was changed from a MAE loss to a logarithmic loss to heighten the impact of lower energies. The logarithmic loss is calculated as an MAE loss of the logarithm of both the true and the predicted energy. Additionally, a ReLu activation function for the energy output is introduced, since the energy prediction has to be strictly positive. As can be seen, the improvements to the network are negligible.

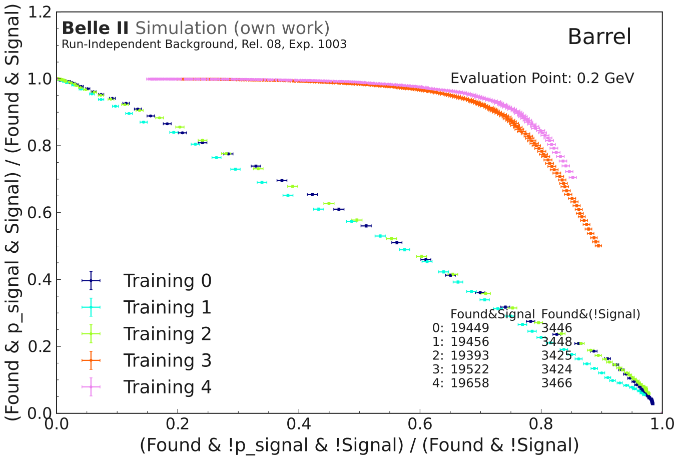


Figure 6.19: ROC curves for the signal/background classifier for the energy region between 150 - 250 MeV and the barrel region for five trainings with the identical training setup and network design as in Fig. 6.18. As can be seen, only two out of five networks learn a non-random signal/background classification.

A second improvement strategy is to predict the energy as a factor of the TC energy instead of an absolute value. Especially for lower energies, the TC energy is the best energy prediction possible, as the entire shower is contained within one TC, the predicted factor should always be close to one and therefore in a well-maintainable range for the network. The loss is calculated as the MAE loss between the true energy and the prediction multiplied with the TC energy of the prediction node.

In Fig. 6.20, the same energy and position resolutions as in Fig. 6.18 are shown. As can be seen, the energy resolution of the network is now nearly identical to the one of ICN-ETM, while the position resolutions have not changed. Additionally, in Fig. 6.21, again five trainings with the same setup are shown, all trained with the prediction of a correction factor for the energy instead of the absolute value. Now all five trainings show a good signal/background separation ability.

The same study was repeated for with a correction factor on the TC position for the position prediction, as well, but showed no improvement over the full position prediction.

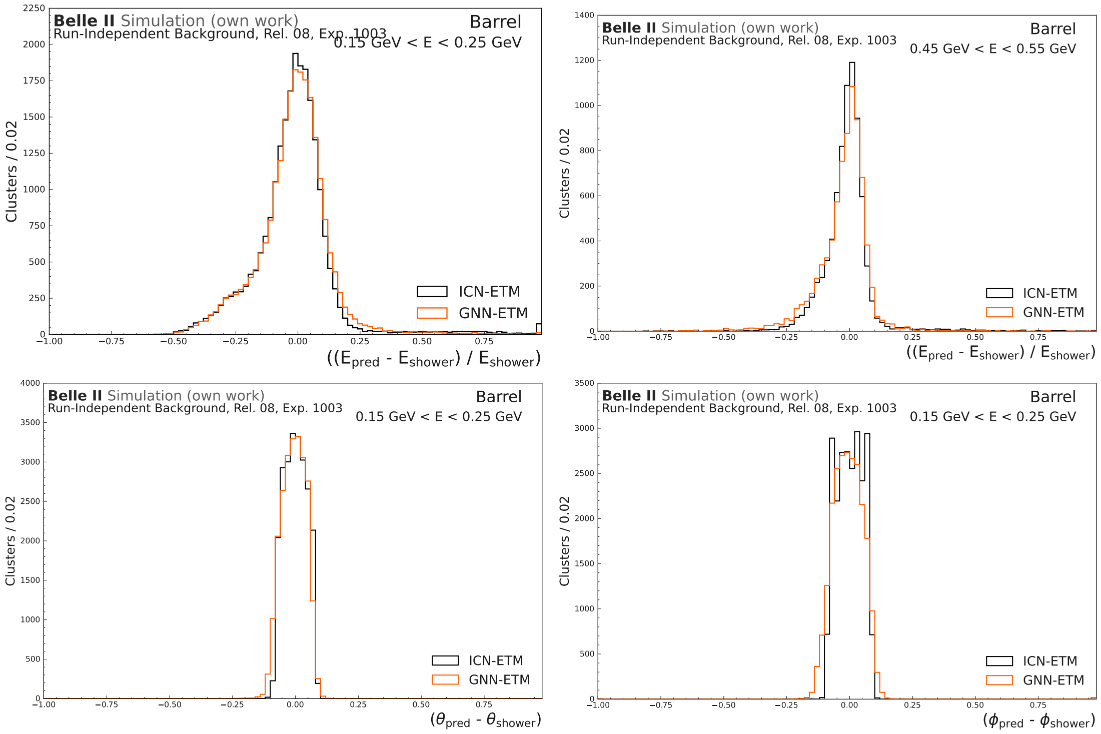


Figure 6.20: Energy resolution for clusters in the barrel region with 150 - 250 MeV (top left) and 450 - 550 MeV (top right) and θ (bottom left) and ϕ (bottom right) resolution for 150 - 250 MeV for one training with the network design and quantization explained in Section 6.3.2, but with the prediction of a correction factor for the TC energy, compared to the resolution of ICN-ETM. The resolutions are bias-corrected as explained in Section 5.2. The dataset used is the Category-One test dataset.

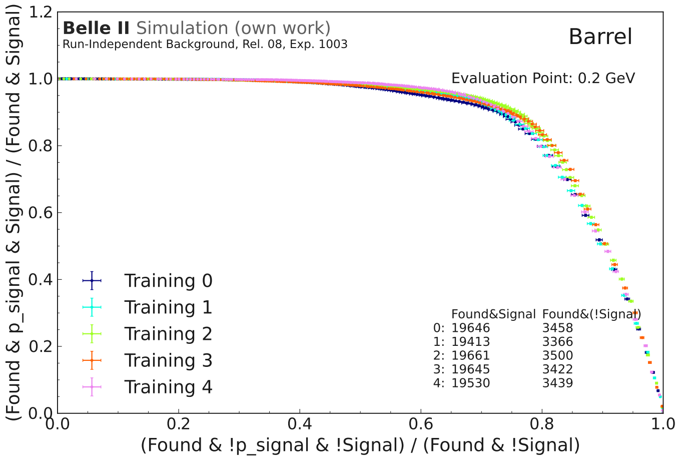


Figure 6.21: ROC curves for the signal/background classifier for the energy region between 150 - 250 MeV and the barrel region for five trainings with the identical training setup and network design as in Fig. 6.20. All networks now have a good signal/background separation.

6.4.4 Hyperparameter Optimization

To optimize the hyperparameters of a network, typically a grid or an optimized search procedure is done by training the network with a number of different configurations of hyperparameters to find the best-performing option. For this work, this has been done with the quantized network shown in Section 6.3 and includes low-magnitude pruning, input scaling, and the prediction of the energy as a scaling factor.

For the hyperparameter optimization, the `Weights&Biases` [86] sweep method was used. This employs a Bayesian hyperband optimization [94] to find the best configuration for the model based on a validation metric. For this work, the validation loss was used as a minimizing target.

The hyperparameters varied for the search and their corresponding value ranges are shown in Table 6.3. As the final network design utilizes approximately 100 % of the FPGA when implemented (more details in Section 6.5.2), this parameter set is regarded as the upper limit.

Table 6.3: Summarized model parameters and the range for the hyperparameter values for the optimization.

Parameter	Search Range
Dim1	4 - 16
Dim2	4 - 16
$\mathrm{N_S}$	2 - 6
$\mathrm{N_{F,LR}}$	2 - 8
k	2 - 8
$\mathrm{f_{exp}}$	1 - 10
$\mathrm{N_{out}}$	4 - 32
$\mathrm{N(coords_{LS})}$	2 - 4
Learning Rate	$[10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}]$
Batch Size	[256, 512, 1024, 2048]

400 trainings were run and the 100 best-performing trainings with the best validation loss below 1 were used for the calculation of the importance metric and the correlation between the hyperparameter and the validation loss. Both metrics are provided by `Weights&Biases`. The result can be seen in Fig. 6.22. While no hyperparameter has a very high importance feature or correlation since both values can go up to 1, a high number of hyperparameters are negatively correlated with the validation loss. This means a higher number of these parameters should improve the performance. This strengthens the design choices presented in Table 6.1, as these are the largest hyperparameter values possible while still staying within the implementation requirements.

The positive correlation between the number of nearest neighbors for the kNN and for the exponential factor is a step to test in further development.

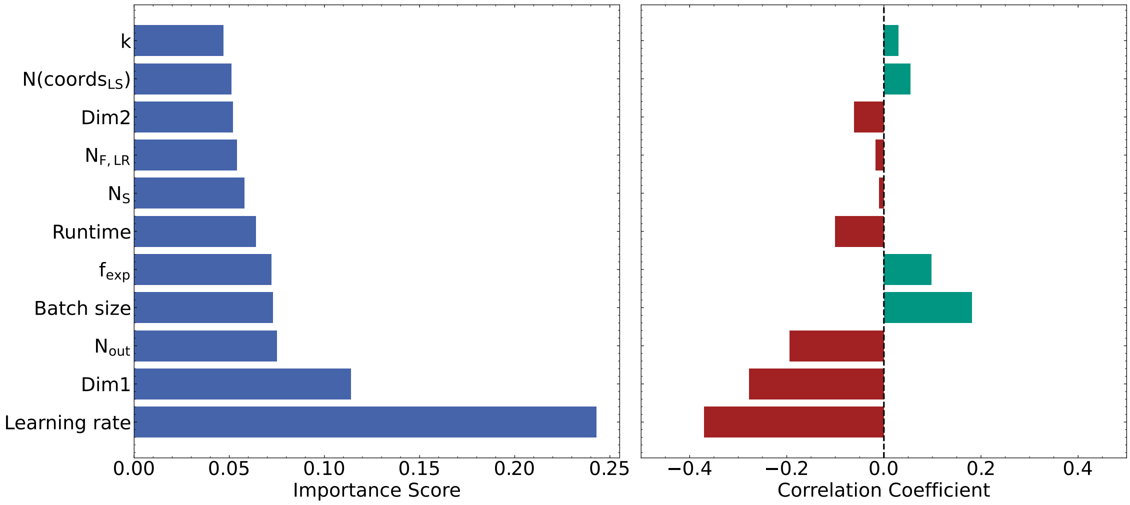


Figure 6.22: Importance metric and correlation between the hyperparameters and the validation loss. Both values are given by the `Weights&Biases` sweep framework and can have a maximum value of 1. Most hyperparameters have a negative correlation to the validation loss.

6.5 Implementation on Hardware

The final GNN-ETM network design was implemented and deployed on a UT4 board and included in the data taking of Belle II at the end of 2024 and operated successfully in a realistic environment. The GNN-ETM is not yet involved in any active trigger decisions, as the current implementation does not yet meet the full latency requirements to do so. However is able to operate with an 8 MHz throughput and perform the full clustering inference online. The work of this chapter has been done in very close collaboration with team members from the Institute für Technik der Informationsverarbeitung (ITIV). Therefore, I will describe the implementation as a collaborative effort, as this has not been done exclusively by me.

6.5.1 Setup and Requirements for Integration to the Level 1 Trigger System

The setup of the GNN-ETM board as part of the overall ECL trigger chain is shown in Fig. 6.23. The GNN-ETM board receives a sparse two-dimensional array containing all 576 TCs from the ICN-ETM. For each TC, the energy, if it is above 100 MeV, the timing, and a hit flag is transmitted. On the GNN-ETM board, preprocessing, network inference, cluster condensation selection, and postprocessing are run and the output is written to a ring buffer. When a trigger signal is being issued by the GDL, the output of the triggered event is sent to the output stream of Belle II via the `Belle2Link` [95], [96].

The full integration of the GNN-ETM into the L1 trigger system requires as a minimum sending of the cluster information to the GDL, so that this information can be part of

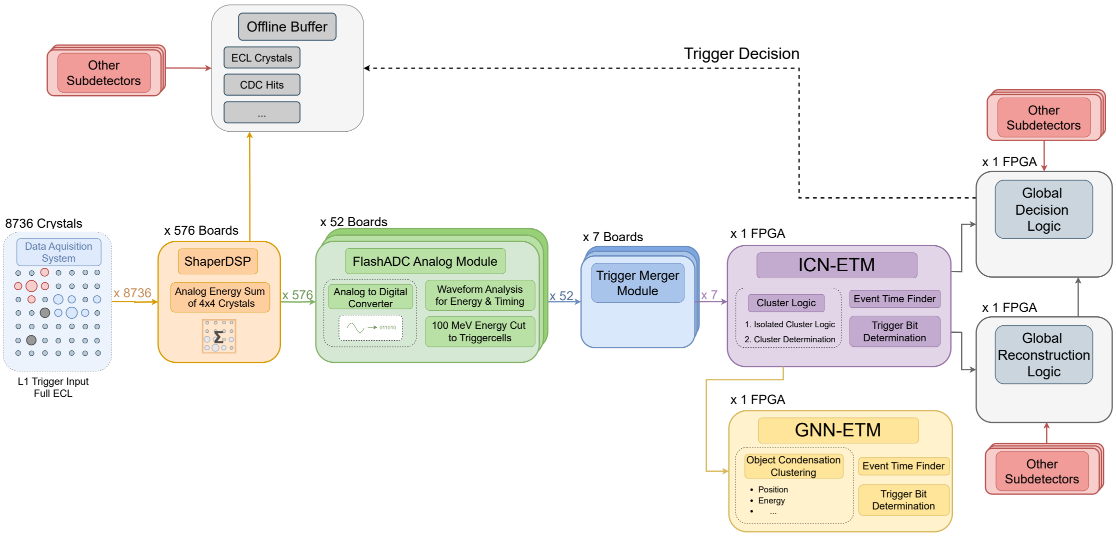


Figure 6.23: Overview of the ECL L1 trigger trigger scheme with the inclusion of the GNN-ETM. The GNN-ETM receives the TCs from the ICN-ETM, runs the preprocessing, the full network, and the post-processing, and stores the TCs after preprocessing and the clusters and predicted values after the post-processing in a buffer. When the GDL sends a trigger signal, the values from the GNN-ETM are read out and sent via the Belle2Link to the readout system.

the trigger decision. This integration has a very strict latency requirement, as the whole readout to GDL decision can only take 5 µs, as described in Chapter 3. With the current design of the full ECL L1 trigger chain, the GNN-ETM implementation has a latency requirement of approximately 1.5 µs. This is not currently achievable by the GNN-ETM implementation. Therefore, and to simplify the integration of the GNN-ETM in the whole L1 trigger system, a parasitic integration setup is used. The link from GNN-ETM to GDL is omitted, so that the GNN clusters are not part of the GDL calculation and are only written to data when a trigger signal is issued. This setup allows for a significantly higher latency than 5 µs. The necessary throughput of 8 MHz is a hard requirement also for the parasitic implementation.

Additionally, when writing out the GNN-ETM results via the `Belle2Link`, the total bandwidth available for the full L1 trigger system cannot be exceeded. The maximum bandwidth needed by GNN-ETM depends on the amount of data sent per trigger signal and the number of trigger signals issued in a given time frame. The number of trigger signals as a function of time is non-uniformly distributed and highly depends on the machine conditions, which makes it difficult to predict for future data-taking periods. To set an upper limit for the implementation to the bandwidth for the GNN-ETM readout, the readout size for the largest single module readout, a track trigger module, is used. This module has a readout size of 196 kbits, which is regarded as an upper limit for the GNN-ETM as the data size per event.

6.5.2 Implementation Design and Results

The final hardware implementation is separated into four modules: The preprocessing, the actual GNN-ETM network inference, the condensation point selection algorithm, and the postprocessing. In the following, a short description of the modules is given.

Preprocessing

The preprocessing module converts the sparse input matrix of 576 TCs to a matrix that contains a maximum of 32 TCs for one trigger decision window. If more than 32 TCs are in one trigger decision window, the remaining TCs are dropped and not used as input to the network. This dropping is random and has not been optimized yet. The number of TCs in the data taken with the GNN-ETM can be seen in Fig. 4.8. In addition to the energy and timing information of each TC, the TC ID is also kept. This ID is used for adding the Cartesian coordinates of the TC center to the input and to store the information in the output data for later identification. The module also identifies the highest-energetic TC per trigger data window and calculates the relative time of all TCs to the highest-energetic TC in one data window. The preprocessing module runs at a frequency of 254 MHz.

GNN-ETM network

The GNN-ETM network is implemented and deployed with the same structure as described in Section 6.4 and Section 6.3. Each dense layer is implemented with a subsequent clipping of the output values to the ranges of the fixed point quantization for this layer. This ensures that no overflow can happen in case the values after dense layers exceed the quantization range, and mimics the behavior of `QKeras` more closely.

The dense layers are calculated for each TC in order without buffering, as the calculations are independent of each other. In comparison, the GravNet layer relies on the information of all points within one event for the dynamic graph building and message passing. For the implementation of the GravNet layer, an All-Nearest-Neighbour (ANN) algorithm [97] is implemented followed by a hierarchical Top-K sorting algorithm. For the message passing step, sorted features are taken from a Block RAM (BRAM) buffer and passed through the exponential function. The values of the exponential function are stored in a LUT optimized for the quantized values of the feature output.

Condensation Point Selection Algorithm

The inference of the OC algorithm, as described in Section 6.2.2, requires the sorting of all output points by their β value and calculating the distances between points to apply the isolation criterion.

For an optimized implementation, an ANN algorithm calculates the distances between all points. Then, the isolation criterion is applied to these distances by applying the distance threshold, resulting in an array containing bitmasks of 0 and 1. In parallel, a candidate selection using the β threshold is done, which creates a second bitmask of 0 and 1 depending on if their β value is above the threshold. The array IDs of all points are sorted by their β value to ensure the condensation point selection is done according to this order. To now calculate the condensation points, an array filled with 0 is initialized. Then, starting from the point with the highest β value, a 1 is set at its position in the condensation point array, and its ID is removed from the candidate list. To remove the points not fulfilling the isolation criterion to this condensation point, an AND operation is done between the isolation bitmask and the candidates bitmask. This sets all points to 0 that are within the distance threshold of the current condensation point. This continues with the next point in the sorted ID list. If this point has already been set to 0, its position in the condensation point array is also left at 0. In Algorithm 6.1, this algorithm is written down in pseudocode as well.

Algorithm 6.1
Condensation Point Candidate Selection (CPCS). Provided by Marc Neu.

```
procedure CPCS($isolations, candidates, ids$)
    $cps \leftarrow \{0\}^N$
    $flags \leftarrow$ candidates
    for $i \leftarrow 0, I_{init} - 1$ do
        parallel for $p \leftarrow 0, P - 1$ do
            $id \leftarrow ids.pop()$
            $cps[id] \leftarrow flags[id]$
            $flags \leftarrow flags \,\&\, isolations[id]$
        end for
    end for
    return $cps$
end procedure
```

To encode the decision of the condensation point selection algorithm while avoiding to add an extra feature to reduce bandwidth, outputs that get selected as condensation points have the sign bit of the β value flipped. As the β value is strictly positive due to the sigmoid activation function, flipping the sign bit is optimal for storing the decision information.

Postprocessing

The postprocessing step ensures the merging of all TCs used in the network and the output clusters with their corresponding features.

Data Readout

The ICN-ETM module writes out the clusters and TCs for eight trigger data windows: The two data windows in which the trigger decision was made and the three windows

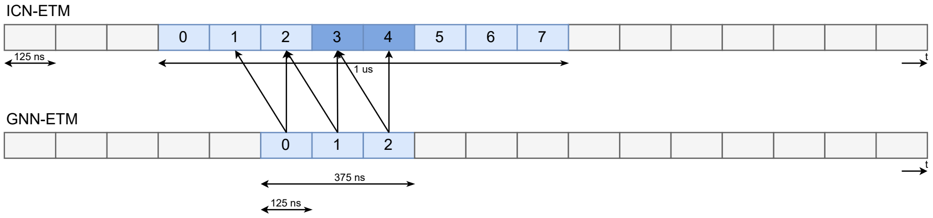


Figure 6.24: Figure to show the relation between GNN-ETM timing and ICN-ETM timing. The ICN-ETM module writes out eight data windows, with data windows 3 and 4 constituting the trigger decision window. The GNN-ETM module writes out three 250 ns windows, where GNN-ETM window 0 corresponds to data windows 1 and 2, GNN-ETM window 1 to data windows 2 and 3 and GNN-ETM window 2 to data windows 3 and 4. The alignment of windows between GNN-ETM and ICN-ETM is adjustable via a register. This alignment is used in the runs in Table 6.4.

before and after. To reduce the amount of data sent via the `Belle2Link` and stored in raw data, for the GNN-ETM data only five trigger data windows are written out. The GNN-ETM module builds its own windows out of two data windows and writes out three of these GNN-ETM 250 ns windows. In Fig. 6.24, the relation between the ICN-ETM data windows and the GNN-ETM windows can be seen. GNN-ETM window 2 corresponds to the trigger decision window of ICN-ETM, which consists of trigger data windows 3 and 4.

The GNN-ETM module writes out three GNN-ETM windows with the corresponding input TCs and the output clusters. Each window has a dynamic size depending on the number of TCs, with the information stored in the header of each window data packet. The design of the data packet is shown in Fig. 6.25. Each TC has six features, each encoded in 16 bits: ID, energy, relative timing, and (x,y,z) position, whereas each cluster has 10 features, also encoded in 16 bits: ID, β value, signal classifier, energy, (x,y,z) position and three latent coordinates. TCs or clusters are always sent in packets of two, so if an odd number of clusters is in one window, an extra cluster with all features set to 0 will be added. The packet size ranges from 1760 bits, when no TC is in any data window, to 24800 bits, if all three windows contain 32 TCs. This is well below the maximum of 196 kbits. The unpacker for this data, which reads out the dynamic information and stores the GNN-ETM TCs, the clusters, and the header data in `basf2` data objects, has been included in `basf2`.

The overall design is implemented on the UT4. The final implementation has a latency of 3.2 µs, with the split between the modules seen in Fig. 6.26. The GNN-ETM itself needs approximately 2 µs. The usage of the resources on the FPGA is shown in Fig. 6.27. The dense layers utilize the majority of the DSPs, especially those with a 16-bit quantization. 8-bit dense layers are mapped to distributed logic. The implementation of the GravNet logic significantly relies on Flip-Flop registers (FFs) and LUT. In total, 76 % of the configurable logic blocks are utilized.

The GNN-ETM network, the condensation point selection algorithm, and the postpro-

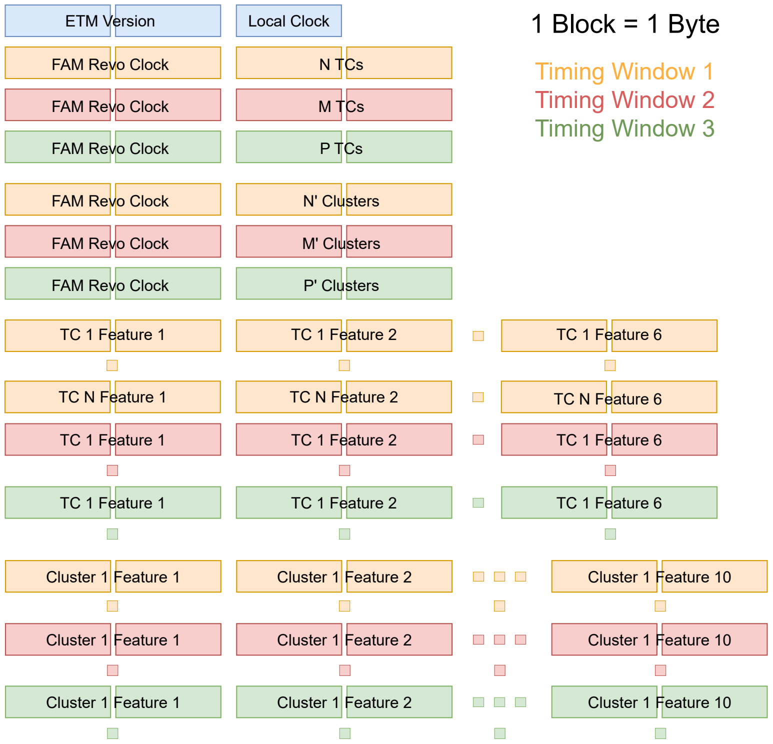


Figure 6.25: Overview of the data packet design of the GNN-ETM module. The module sends out three timing windows with the TCs features and the cluster features in that window. The packet size is dynamic and the window headers store the information of the number of TCs and clusters within that window. The maximum data size of one packet is 24 800 bits, if each window contains 32 TCs.

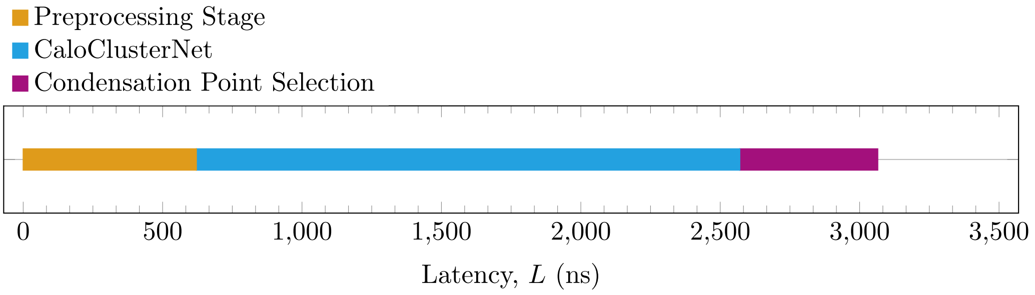


Figure 6.26: End-to-end latency for the complete inference chain of the GNN-ETM including preprocessing and the condensation point selection algorithm implemented on the UT4, featuring the AMD Ultrascale XCVU190. The GNN-ETM itself needs the highest amount of latency with almost 2 µs. Figure by Marc Neu.

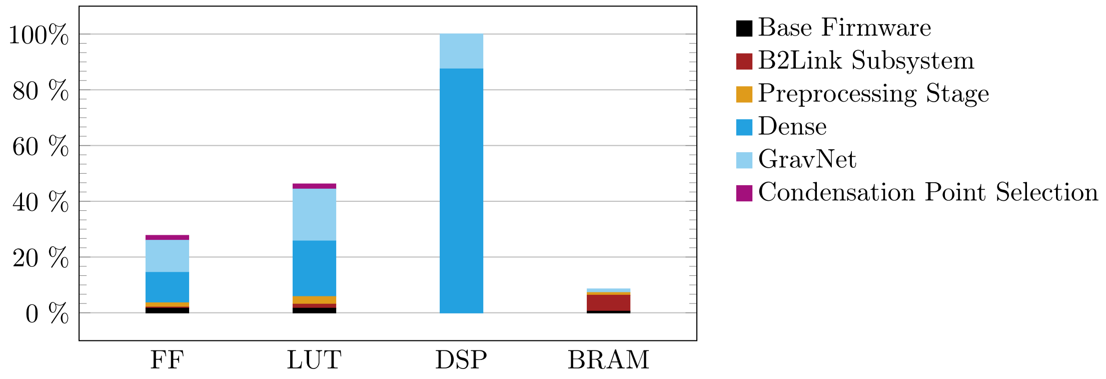


Figure 6.27: Utilization of system resources on the UT4, featuring an AMD Ultrascale XCVU190. The usage of FFs, LUTs, DSPs and BRAMs separated for the base firmware, the `B2Link` system and the different steps of the clustering is shown. The GNN-ETM implementation is additionally separated into the dense layers and the GravNet layers. The DSP usage is at 100 % and primarily used by dense layers with 16 bit quantization. Figure by Marc Neu.

cessing all run at a frequency of 127 MHz. Running at a frequency of 254 MHz would very likely approximately halve the latency but the high resource usage of the implementation currently does not allow for this. Improvements to the network design, such as a reduction of quantization ranges and calculations in general, are necessary for this step. This is difficult due to the multi-step process of training, evaluating and implementing to get the final results, as well as due to the large parameter space.

6.5.3 Evaluation Setup for Hardware Performance

The hardware implementation can be verified and crosschecked with both the results given by `QKeras` as well as different steps in the implementation design. For this work, there are three outputs to be compared:

- The output of the GNN-ETM and the condensation point selection algorithm in `QKeras`;
- The output of the transaction-level C-simulation, which implements the GNN-ETM and the condensation point selection algorithm in Vitis 2024.2 [98] without the preprocessing and the postprocessing step;
- The hardware output of the GNN-ETM board when taking data with the Belle II detector.

The comparison between the C-simulation and the hardware output is useful to debug implementation processes, as these should be bitwise accurate. In Fig. 6.28, this comparison for a cosmic run taken with the GNN-ETM board, exemplary for the energy and x output values, can be seen. The values are identical.

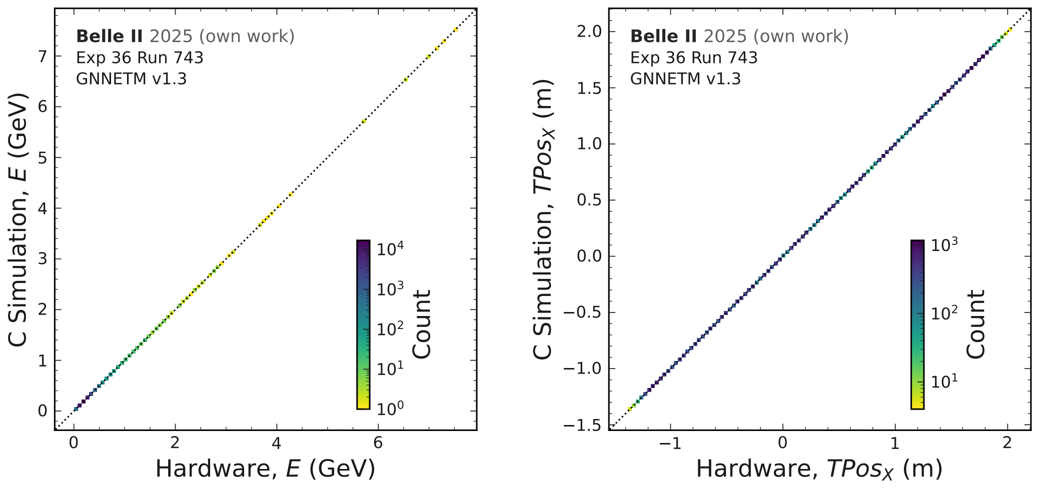


Figure 6.28: Comparison between the C-simulation and the hardware energy output (left) and the x output (right) for a cosmics run taken with the GNN-ETM board. The TCs, as written out by the GNN-ETM module in raw data, are used as input to the C-simulation. The output values are identical. Figure by Marc Neu.

Consequently, for the comparison between the QKeras simulation and the final hardware output, the C-simulation is used, which can be run locally. The training data is processed via the C-simulation and the outputs are compared. In Fig. 6.29, a comparison of the x and energy output values can be seen. The left plots show the comparison between all output values, while the right plot only shows the comparison between output values selected as condensation points by both algorithms. Except for the high energy region for the energy output, the condensation points show a good agreement. This might be due to the clipping of the input energy values at 8 GeV, which can worsen the performance of the GNN-ETM for high energies. The values of the non-condensation points behave very erratically. While this does not affect the performance of the overall clustering, this should still be improved in further work. One reason for this behavior might be that the network learns to push non-condensation points to the edges of the quantization range and therefore lead to quantization errors in the different implementations of QKeras and the C-simulation.

A special case is the output agreement for the β value as well as the signal classifier output due to the Smooth Sigmoid activation function (see Section 6.3.3). In Fig. 6.30, the left plot shows the β value output comparison after the condensation point selection algorithm. Points in the lower left quadrant have been selected by both algorithms as condensation points, points in the upper right quadrant are not selected as condensation points by either algorithm. A high discrepancy between the C-Simulation and the QKeras output is visible, with differences up to the maximum value of 1. The reason for this is shown in the plot on the right. This shows the output of the β dense layer before the

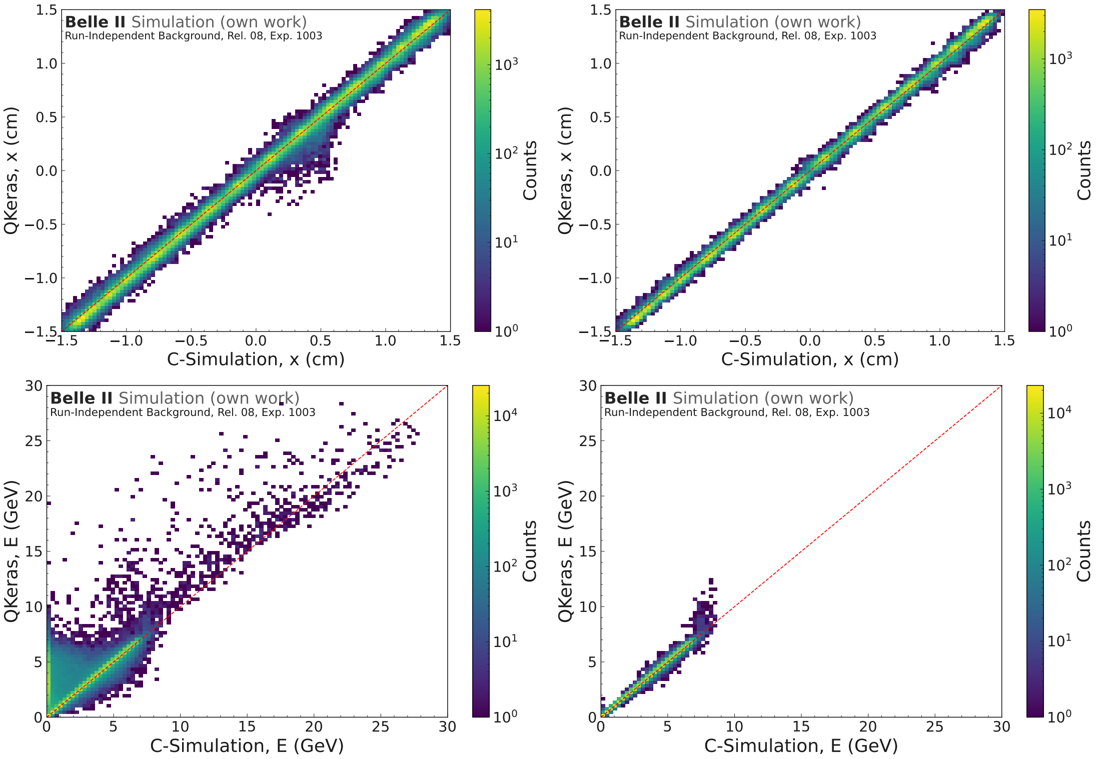


Figure 6.29: Comparison between the `QKeras` simulation and the C-simulation for the x (top) and energy (bottom) prediction values. The left column shows the output of all points, while the right column shows the output of only points that have been chosen as condensation points in both the `QKeras` and the C-simulation.

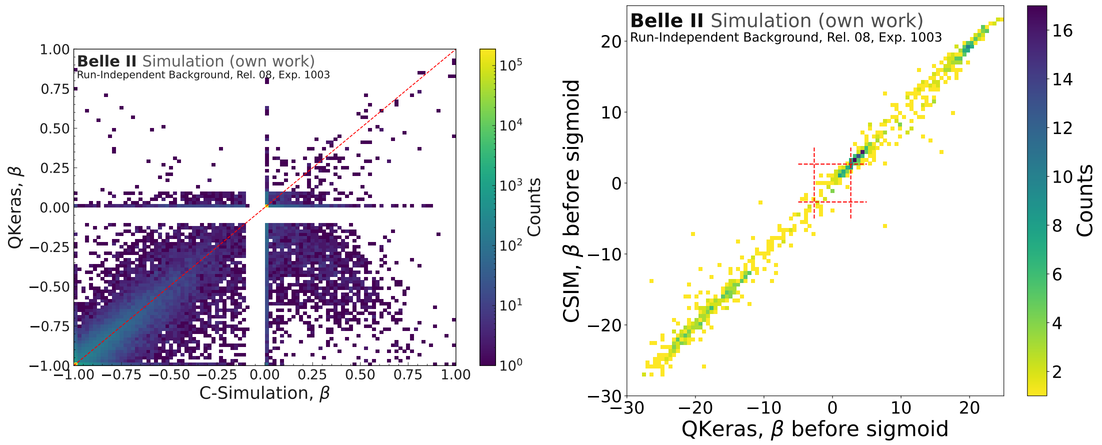


Figure 6.30: Comparison between the QKeras simulation and the C-simulation for the β value. The left plot shows the final β value output after the condensation point selection algorithm. Values in the lower left quadrant have been selected by both algorithms as condensation points. A large discrepancy between QKeras and the C-simulation is visible. On the right, the output of the β dense layer before the Smooth Sigmoid activation function is shown. The red dotted lines mark the region, where the activation function is linear. Everything outside this region is mapped to 0 or 1.

Smooth Sigmoid activation function for both the C-Simulation and QKeras. The red dotted lines mark the region in which the activation function has a linear mapping, while points outside this region are mapped to 0 or 1. The overall agreement of this curve is in the same range as the agreement for the values in Fig. 6.29. The Smooth Sigmoid then acts as a magnifying glass of the region inside the linear part and enhances the quantization error significantly.

The differences in the β value have a large impact on the selection of the condensation points and subsequently on the predicted clusters. For future work, a different activation function has to be chosen or the overall quantization error reduced.

6.5.4 Data Taken with GNN-ETM

The GNN-ETM network has by now been parasitically running in Belle II multiple times during collision and cosmics data taking, and is the first real-time GNN deployed in a realistic environment. Due to the scheduled end of data taking of Belle II at the end of 2024, deploying the GNN-ETM in runs with collision data was only possible for a limited amount of time.

In Table 6.4, all runs with collision data and an included GNN-ETM module are shown. The runs are color-coded with red indicating a run stop caused by the GNN-ETM module, orange indicating corrupted GNN-ETM data without impacting the general data taking of Belle II, and green showing runs, which had all subdetectors included and valid data from GNN-ETM. The data corruption is due to an irregular data transfer between the ICN-ETM and GNN-ETM boards and not due to an error in the GNN-ETM implementation.

Because of yet unknown reasons, at irregular times some TCs are lost in the transfer from ICN-ETM to GNN-ETM. As the GNN-ETM preprocessing is dependent on all TCs being transferred in a fixed order, this leads to wrong data alignment and corrupted data packets.

In the runs in Table 6.4, we showed that the implementation of the GNN-ETM algorithm can successfully run GNN-based clustering in a real-time environment with very high beam backgrounds. The GNN-ETM was also included in Run 2886 without any internal difficulties, where SuperKEKB delivered the world-record instantaneous luminosity [6], showing that the algorithm can cope with high data rates. Both the preprocessed inputs and the outputs of the network were written out to the official data of Belle II and can be read out via `basf2` with the implemented GNN-ETM unpacker. We showed a full end-to-end implementation including integration into the current L1 trigger setup of Belle II.

The data taken in these runs is additionally useful to test the preprocessing, postprocessing, and the unpacker of the raw data packets. The full performance of the actual clustering could not be tested. The weights deployed are weights taken from a trained model but had several key components missing to perform correct clustering. This does not change any of the implementation functionality but impacts the output of the network in such a way that no performance analysis can be done for the clustering.

To test the preprocessing and in combination the unpacker in `basf2`, the preprocessed input TCs to the network, which are written out into raw data, are compared to the ICN-ETM TCs. To test this agreement, first the TCs in the correct timing windows for both ICN-ETM and GNN-ETM are selected. As the most TCs are in the data windows 3 and 4 of ICN-ETM, these windows are selected. The corresponding window for GNN-ETM is window 2. In Fig. 6.31, the hit TCs with their ID are shown for both GNN-ETM and ICN-ETM, for the subset of events with a maximum of 32 TCs. The distributions agree perfectly. Furthermore, the reconstructed energy per TC is shown in Fig. 6.32. The cutoff due to the quantization range for the GNN-ETM TCs can be seen at 8 GeV in the left plot. The distribution below 8 GeV agrees very well, but small discrepancies can be seen. These discrepancies are explained with the right plot, which shows the energy range between 0.1 and 0.16 GeV with the bin size being $1/2^{12}$, which is the quantization precision for the GNN-ETM inputs. The ICN-ETM energy values are stored as ADC counts in the raw data and converted to GeV with floating point precision afterward. In comparison, the GNN-ETM conversion happens online in Q4.12 fixed point precision resulting in small differences in energy values. In summary, this shows that the preprocessing of the input values, the value merging in the postprocessing, and the unpacking of the raw data works as intended.

Table 6.4: Runs taken with the GNN-ETM included into official Belle II data taking during the 26. and 27. December 2024. Physics runs have all subdetectors besides PXD included, while Debug runs additionally had the CDC and SVD detectors on standby due to the very high beam backgrounds. Red columns show runs where the GNN-ETM board threw an error that stopped data taking whereas orange runs are runs that have corrupted GNN-ETM data due to unstable links between the ICN-ETM and GNN-ETM boards. Runs with all subdetectors included and no problem with the GNN-ETM board are marked in green.

Run	Type	Duration	$\mathcal{L}_{\text{max}}$ ($10^{30}\,\text{cm}^{-2}\,\text{s}^{-1}$)	N(Events)	$\int \mathcal{L}\,dt$ (nb^{-1})	Version
2476	Physics	8m 34s	2.25×10^{6}	2249081	3.48×10^{3}	V25
2477	Physics	4m 23s	1.75×10^{5}	175150	6.37×10^{3}	V25
2478	Physics	2m 23s	4.79×10^{5}	479291	1.18×10^{4}	V25
2734	Physics	10m 52s	1.23×10^{4}	1391497	6.91×10^{3}	V27
2880	Debug	16m 18s	4.89×10^{4}	1642822	3.51×10^{4}	V31
2881	Debug	30m 35s	2.68×10^{4}	2154924	4.30×10^{4}	V31
2882	Physics	31m 52s	4.33×10^{4}	2229687	5.93×10^{4}	V31
2883	Debug	17m 39s	4.58×10^{4}	980574	3.21×10^{4}	V31
2884	Physics	8m 10s	4.06×10^{4}	349826	1.03×10^{4}	V31
2885	Physics	4m 01s	3.98×10^{4}	230241	6.36×10^{3}	V31
2886	Debug	22m 25s	5.10×10^{4}	2267967	4.54×10^{4}	V31
2887	Physics	13m 37s	4.25×10^{4}	708037	2.04×10^{4}	V31
2888	Debug	39m 1s	4.72×10^{4}	2017994	7.24×10^{4}	V31
2889	Physics	09m 38s	3.25×10^{4}	427369	1.08×10^{4}	V31
2890	Physics	10m 02s	3.35×10^{4}	371138	1.09×10^{4}	V31
2891	Physics	34m 22s	3.85×10^{4}	2334215	6.35×10^{4}	V31
2892	Physics	01m 24s	3.71×10^{4}	93286	2.40×10^{3}	V31
2894	Physics	42m 19s	3.84×10^{4}	3047148	8.21×10^{4}	V31
2895	Physics	06m 47s	3.24×10^{4}	201125	7.02×10^{3}	V31
2896	Physics	10m 15s	3.85×10^{4}	564738	2.04×10^{4}	V31
2897	Physics	17m 26s	3.81×10^{4}	731807	2.75×10^{4}	V31
2899	Physics	10m 43s	3.75×10^{4}	626276	2.05×10^{4}	V31
2901	Physics	15m 32s	3.84×10^{4}	894545	3.08×10^{4}	V31
2902	Physics	12m 15s	3.82×10^{4}	976893	2.28×10^{4}	V31
2903	Physics	04m 58s	3.71×10^{4}	284071	8.95×10^{3}	V31
2904	Physics	11m 15s	3.77×10^{4}	865036	2.11×10^{4}	V31

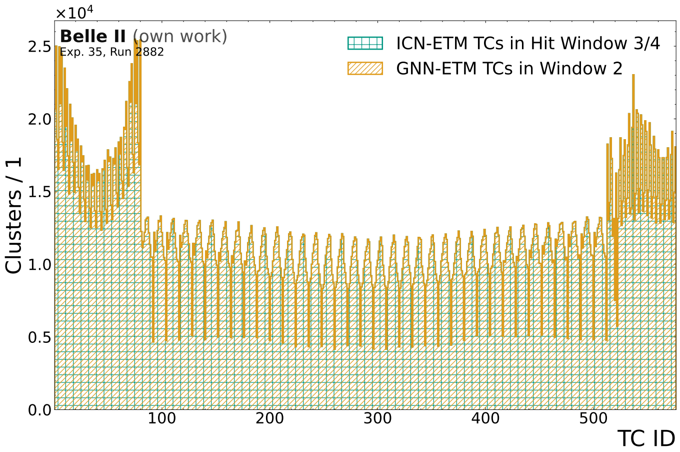


Figure 6.31: TC IDs for the ICN-ETM and GNN-ETM TCs in timing window 3 and 4 and window 2, respectively. The TCs are taken from Exp. 35, Run 2882, and the subset of events with a maximum of 32 TCs is shown. The distributions are hatched to show the agreement. Both distributions are identical.

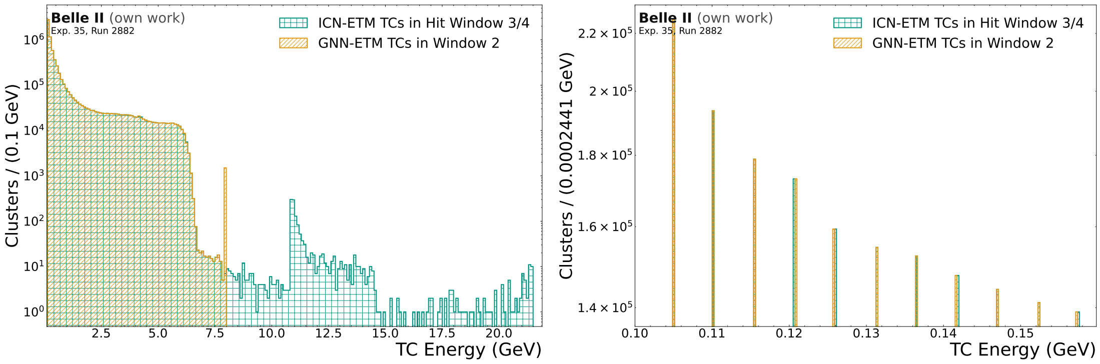


Figure 6.32: TC IDs for the ICN-ETM and GNN-ETM TCs in timing window 3 and 4 and window 2, respectively. The ICN-ETM energy has been converted from ADC counts to GeV. The TCs are taken from Exp. 35, Run 2882, and the subset of events with a maximum of 32 TCs is shown. On the left side, the entire energy range is shown, with a cutoff at 8 GeV for the GNN-ETM TC due to the quantization range. The right plot shows the range between 0.1 and 0.16 GeV. The mismatches are due to the offline floating point precision conversion of the ADC values of the ICN-ETM TCs, while the GNN-ETM TCs are converted with fixed point Q4.12 precision online.

Chapter 7

Evaluation on Technical Datasets

To correctly gauge the performance of the GNN-ETM, first an evaluation on technical, simulated datasets is done. This allows for: A validation of performance metrics on special datasets for overlapping particles; to compare the effect of different β cuts on the metric values; and to test the signal/classifier discrimination performance.

In this chapter, first an overall evaluation of the metrics presented in Chapter 5 on the technical test datasets explained in Section 4.3 is shown. The variations inbetween trainings is shown and the procedure to choose a final network is described. Then, one model is selected and evaluated in depth, with a focus on the discrimination performance of the signal/background classifier. Additionally, the effect of different β cuts is shown here.

As one major drawback of the ICN-ETM is the performance on close-by or overlapping clusters, a dedicated dataset with two close-by particles with equal energies is being evaluated as well. For this, the performance of the network is tested in dependency of the distance between clusters.

At last, the performance of events with pure beam background simulation is evaluated. For these samples, the overall trigger bit efficiencies and trigger rates are tested including a test of the signal/background classification. Decreasing the trigger rate on beam background hits can directly improve the data taking of Belle II due to a reduction in triggered, but later discarded events.

For all evaluations, a β threshold of 0.04 and a latent space coordinate cut of 0.3 is set.

7.1 Comparison of Different Trainings

The variance in performance metrics between trainings with identical setups has been significantly higher than the overall training and validation loss suggests. Due to this, for every training configuration, at least five trainings are always performed to choose the best-performing. In this section, first the differing performances of the five network trainings with the overall configuration shown in Fig. 6.7 and the final parameter values in Table 6.1 are evaluated. The final model design consists of 4803 trainable parameters, with

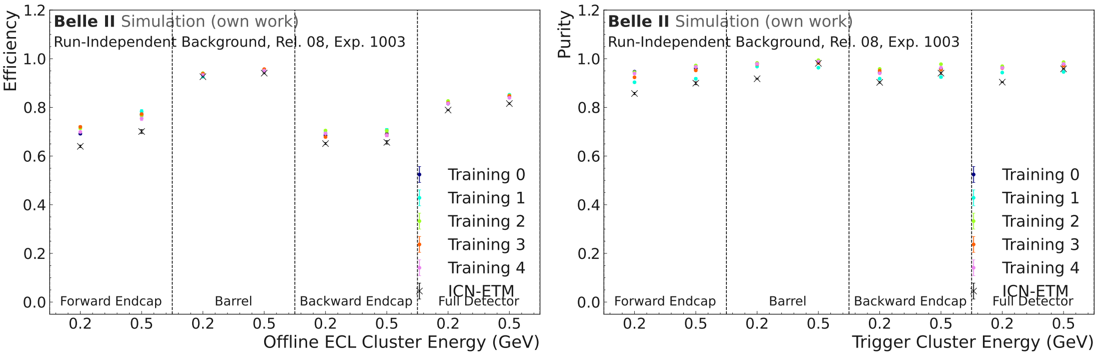


Figure 7.1: Efficiency (left) and purity (right) for two energy ranges of offline ECL clusters for the efficiency and trigger clusters for the purity, 0.15 - 0.25 GeV and 0.45 - 0.55 GeV for five trainings evaluated on the Category-One test dataset. For a direct comparison, the ICN-ETM efficiency and purity is also shown.

40 % pruning applied, resulting in 2882 parameters. One model is chosen in the end for implementation and evaluation. This model is then also used for evaluation in Chapter 8.

To choose the best-performing final model, all trained models are evaluated on the Category-One test dataset and their efficiency, purity, energy and position resolution, and the signal/background classifier performance is compared. This has to be done for different offline ECL clusters energies and for the different detector regions to ensure consistent performance of the final model overall. For a first evaluation, two energy regions are chosen, 0.15 - 0.25 GeV and 0.45 - 0.55 GeV. As the energy threshold for TCs is 100 MeV, an evaluation bin starting at 150 MeV ensures no cut-off effects but still allows for a significant amount of true background offline ECL clusters. As the number of true background offline ECL clusters drops significantly with higher energies, to correctly evaluate the signal/background classifier, the second evaluation point is chosen to be of higher energy to show improvement in terms of efficiency and resolution but to still test the classifier performance.

In this section, selected metrics for the different trainings are shown. The remaining distributions can be found in Chapter C.

In Fig. 7.1, the efficiency (left) and the purity (right) for five trainings in comparison to the efficiency and purity of ICN-ETM is shown. The offline ECL clusters in the endcaps often show higher overlap due to possible interactions with inactive material before the calorimeter, as shown in Section 4.2.2 and are therefore more difficult to reconstruct separately, which is shown by the drop in efficiency in comparison to the barrel region. The purity is shown dependent on the predicted energy of GNN-ETM or ICN-ETM clusters and is very close to 1 for all regions. Nevertheless, differences up to 5 % can be seen in both purity and efficiency for different trainings.

In Fig. 7.2, the energy and position resolution widths for x, y, and z, with the definition shown in Section 5.2.2, can be seen. The here shown widths are calculated after correcting the resolution distributions for any potential bias described later in more detail. As can be

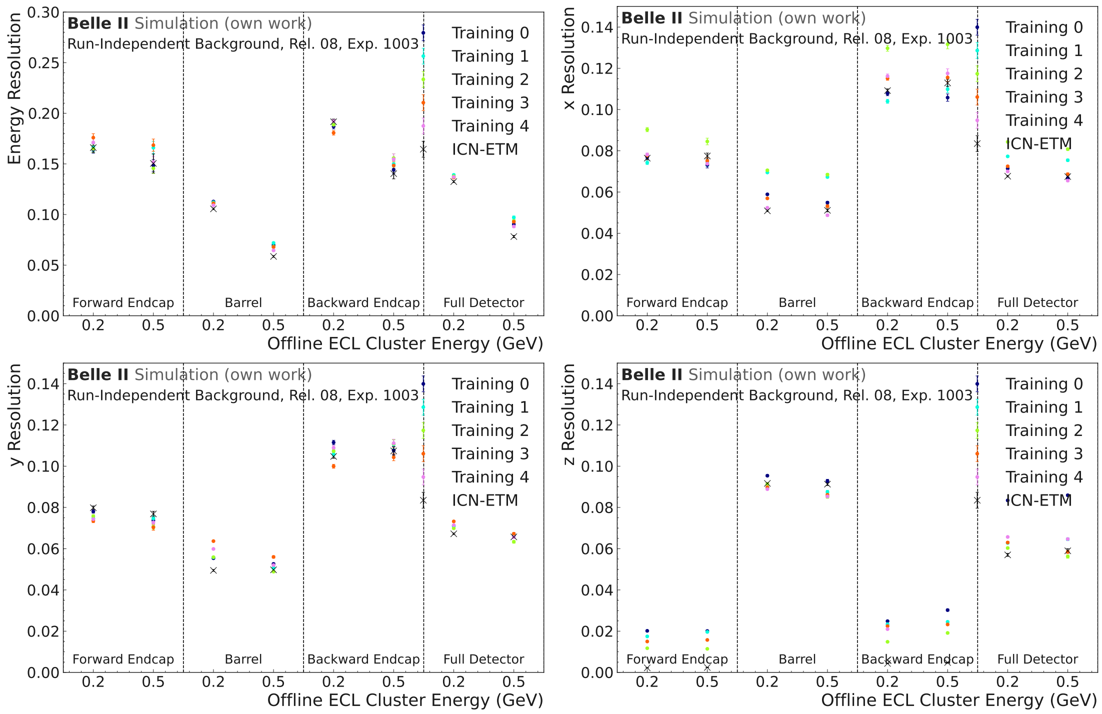


Figure 7.2: Energy (upper left), x (right), y (lower left) and z (lower right) resolution widths for an offline ECL cluster energy between 0.15 - 0.25 GeV and 0.45 - 0.55 GeV for five trainings evaluated on the Category-One test dataset. For a direct comparison, the ICN-ETM resolution width is also shown. All resolutions are after bias correction.

seen, the variations are very small. For the x resolution width, Training 2, which has one of the higher efficiencies in Fig. 7.1, performs worse in the overall x prediction.

To evaluate for possible structures, biases and other effects, which cannot be detected by the width of the resolution, the resolution distributions for offline ECL cluster energies between [0.15, 0.25] GeV for the z prediction and the energy prediction can be seen in Fig. 7.3 and in Fig. 7.4, respectively. The bias towards smaller energy values for the ICN-ETM is clearly visible, which can be attributed to energy deposited in crystals outside of the main TCs. The GNN-ETM can directly learn this correction factor, which is visible in the distribution centered around 0. Nevertheless, a more visible tail towards smaller energy predictions is seen for all networks. For the z prediction, while the width of all five trainings is nearly identical, as shown in Fig. 7.2, Training 0 shows a clear shift towards smaller z values. While this is a small effect, this can have a visible impact on the efficiency of trigger bits such as ecl_mumu oder bha3d (see Table 3.2), which depend on position cuts. The four distinct peaks for the ICN-ETM are the distances of the center of a TC to the crystal centers within that TC. The crystal centers are favoured as positions for reconstructed ECL clusters, while the ICN-ETM, as explained in Section 3.2, always takes the center of the highest-energetic TC as the position of the cluster.

For the signal/background classifier, the background rejection rate at 95 % signal efficiency can be seen in Fig. 7.5. The corresponding ROC curves for each separate detector

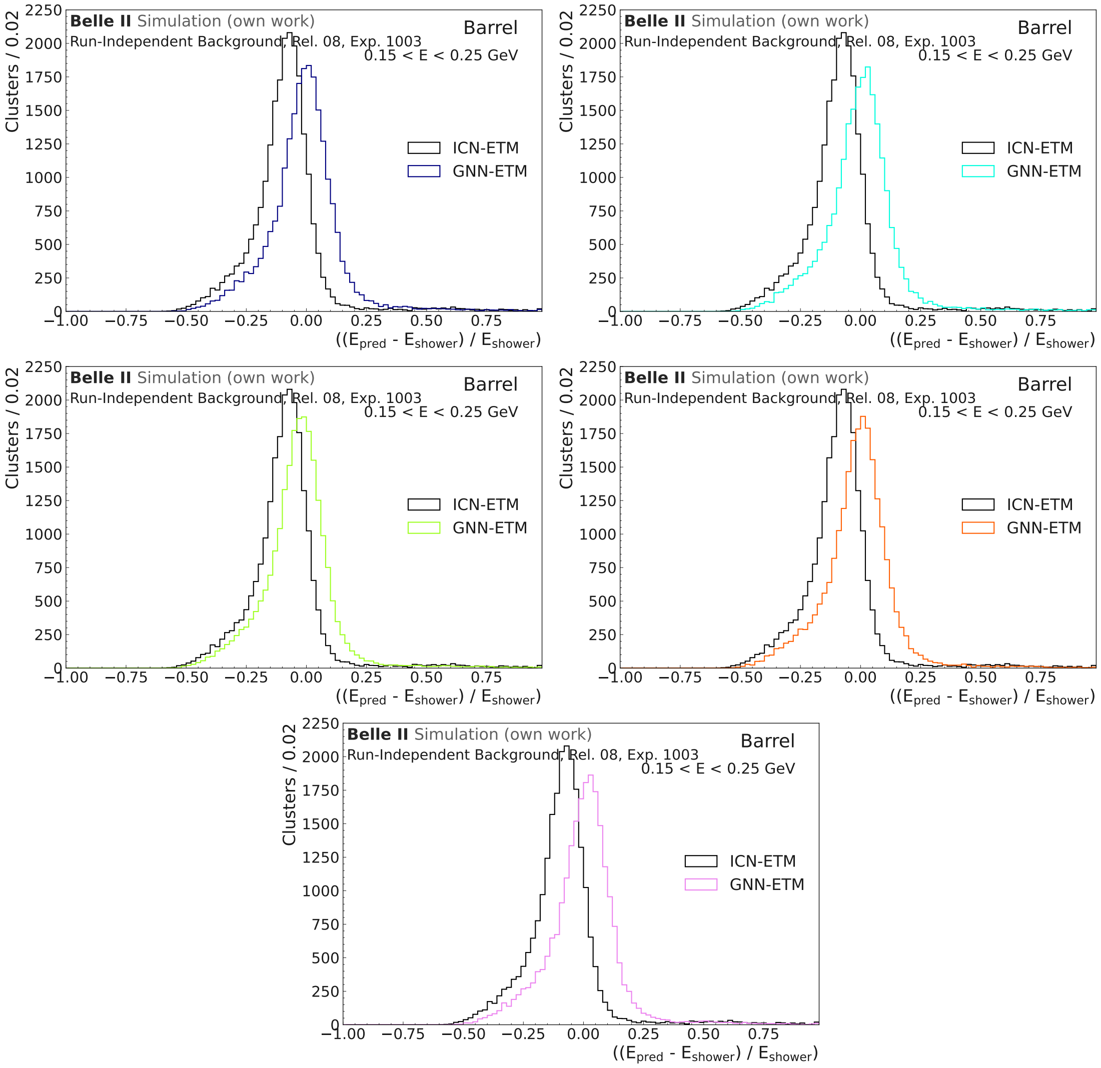


Figure 7.3: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with an offline ECL cluster energy between 0.15 - 0.25 GeV. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

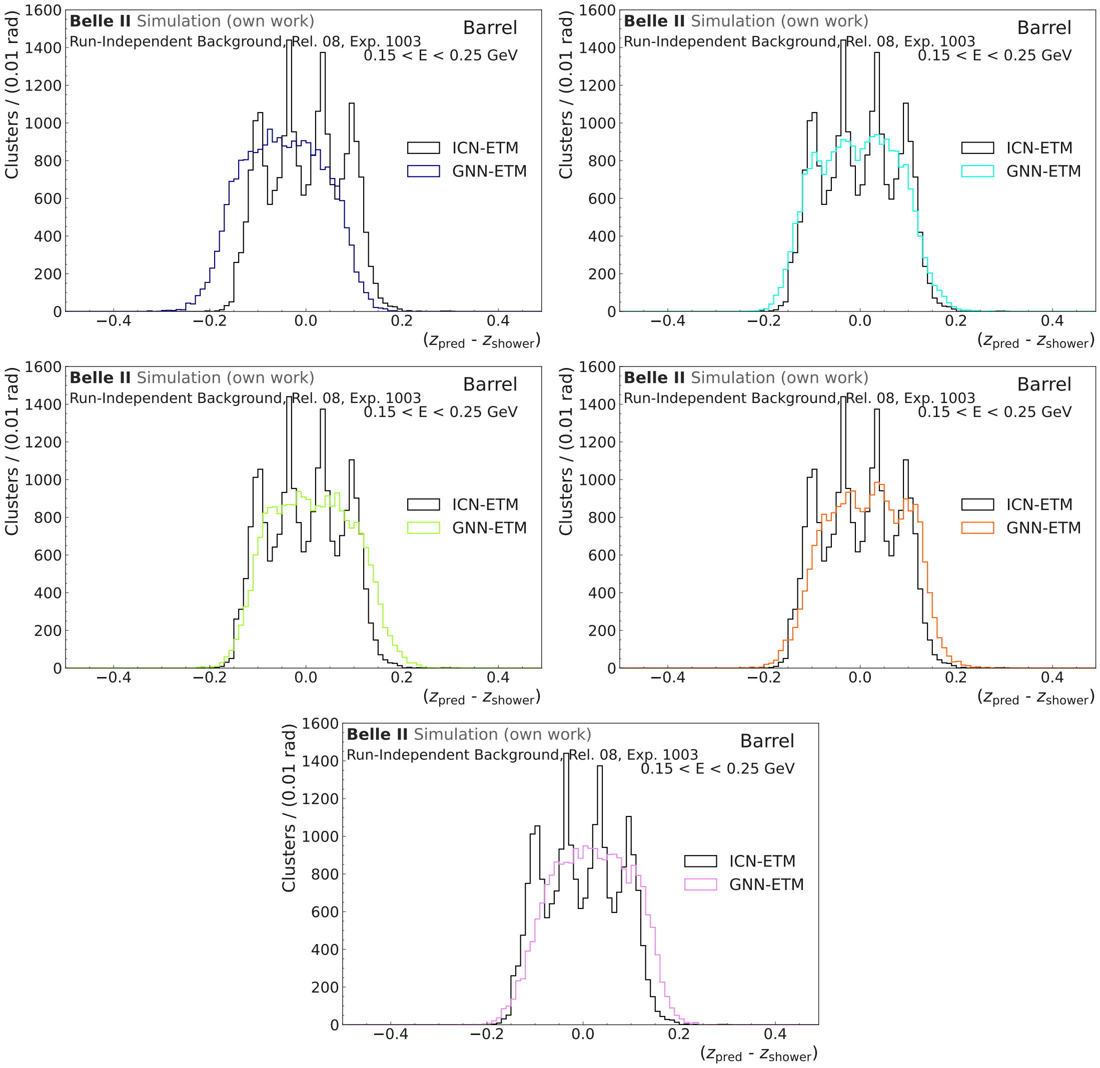


Figure 7.4: z resolution histograms for five trainings evaluated on the Category-One test dataset for clusters with an offline ECL cluster energy between 0.15 - 0.25 GeV. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

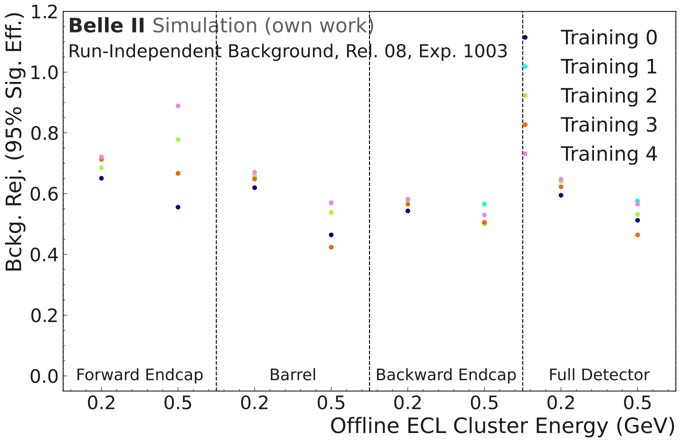


Figure 7.5: Background rejection rate for 95 % signal efficiency for an offline ECL cluster energy between 0.15 - 0.25 GeV and 0.45 - 0.55 GeV for five trainings evaluated on the Category-One test dataset.

region and the two energy regions are shown in Fig. 7.6. The number of true background offline ECL clusters for an offline ECL cluster energy between 0.45 - 0.55 GeV is about 3 % of the number of true signal offline ECL clusters in the backward endcap and significantly less in the other detector regions. This causes the high variation in the background rejection rate in Fig. 7.5 for the higher energy region. Nevertheless, it is important to still monitor the classifier at higher energies to guarantee good performances. The nuances in performance between the different trainings can be seen for lower offline ECL cluster energies. Training 0 has a slightly steeper slope in the ROC curve and therefore also a worse background rejection rate at 95 % signal efficiency. Training 4 shows the overall best signal/classifier performance.

In Table 7.1, the final values for clusters between 0.15 - 0.25 GeV for all five trainings and ICN-ETM in comparison are shown. The best training for each detector region and metric value is marked in green. As can be seen, there is no single training that consistently outperforms the others. Training 2 shows the best performance in terms of efficiency and purity and good performances for the energy resolution and background rejection. While position resolution is a factor for the calculation of trigger bits, differences in efficiency up to 3 % for other trainings significantly hurt the performance overall. Due to this, the network resulting from Training 2 is chosen for further evaluations.

7.2 Signal/Background Classification Performance

The signal classifier is evaluated on the Category-One test dataset to check the separation performance. Additionally, the performance on pure beam background samples is tested.

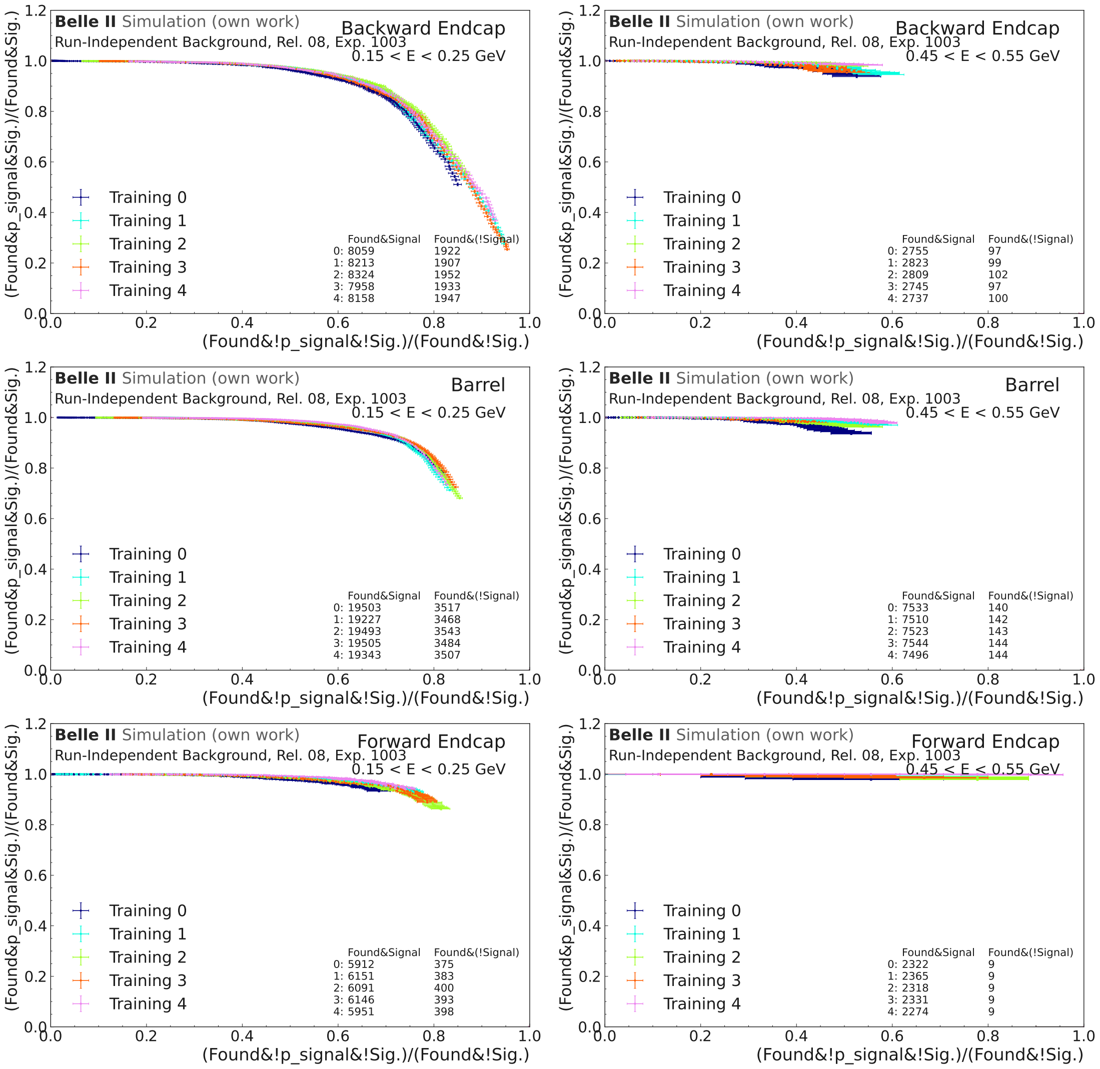


Figure 7.6: ROC curve for the signal/background classifier for the three detector parts. In the upper row, clusters in the backward endcap are shown, the center row depicts the barrel region and the lower row the forward endcap. In the left column, the offline ECL cluster energy is between 0.15 - 0.25 GeV and in the right column between 0.45 - 0.55 GeV for five trainings evaluated on the Category-One test dataset.

	T_i	efficiency	purity	η_{energy}	η_x	η_y	η_z	Bckg Rej.
fwd	0	$0.692^{+0.005}_{-0.005}$	$0.947^{+0.003}_{-0.003}$	$0.164^{+0.004}_{-0.004}$	$0.076^{+0.001}_{-0.001}$	$0.077^{+0.001}_{-0.001}$	$0.020^{+0.000}_{-0.000}$	0.651
	1	$0.719^{+0.005}_{-0.005}$	$0.903^{+0.003}_{-0.003}$	$0.166^{+0.004}_{-0.004}$	$0.076^{+0.001}_{-0.001}$	$0.076^{+0.001}_{-0.001}$	$0.017^{+0.000}_{-0.000}$	0.715
	2	$0.715^{+0.005}_{-0.005}$	$0.945^{+0.003}_{-0.003}$	$0.165^{+0.004}_{-0.004}$	$0.079^{+0.001}_{-0.001}$	$0.074^{+0.001}_{-0.001}$	$0.012^{+0.000}_{-0.000}$	0.685
	3	$0.720^{+0.005}_{-0.005}$	$0.924^{+0.003}_{-0.003}$	$0.176^{+0.004}_{-0.004}$	$0.077^{+0.001}_{-0.001}$	$0.074^{+0.001}_{-0.001}$	$0.015^{+0.000}_{-0.000}$	0.712
	4	$0.699^{+0.005}_{-0.005}$	$0.939^{+0.003}_{-0.003}$	$0.171^{+0.004}_{-0.004}$	$0.079^{+0.001}_{-0.001}$	$0.074^{+0.001}_{-0.001}$	$0.017^{+0.000}_{-0.000}$	0.721
	ICN	$0.640^{+0.005}_{-0.005}$	$0.857^{+0.004}_{-0.005}$	$0.166^{+0.005}_{-0.005}$	$0.078^{+0.001}_{-0.001}$	$0.075^{+0.001}_{-0.001}$	$0.002^{+0.000}_{-0.000}$	-
barrel	0	$0.940^{+0.002}_{-0.002}$	$0.981^{+0.001}_{-0.001}$	$0.113^{+0.001}_{-0.001}$	$0.057^{+0.000}_{-0.000}$	$0.055^{+0.000}_{-0.000}$	$0.091^{+0.000}_{-0.000}$	0.620
	1	$0.927^{+0.002}_{-0.002}$	$0.968^{+0.001}_{-0.001}$	$0.112^{+0.001}_{-0.001}$	$0.054^{+0.000}_{-0.000}$	$0.057^{+0.000}_{-0.000}$	$0.090^{+0.000}_{-0.000}$	0.656
	2	$0.940^{+0.001}_{-0.002}$	$0.983^{+0.001}_{-0.001}$	$0.111^{+0.001}_{-0.001}$	$0.055^{+0.000}_{-0.000}$	$0.056^{+0.000}_{-0.000}$	$0.092^{+0.000}_{-0.000}$	0.662
	3	$0.939^{+0.002}_{-0.002}$	$0.977^{+0.001}_{-0.001}$	$0.111^{+0.001}_{-0.001}$	$0.056^{+0.000}_{-0.000}$	$0.058^{+0.000}_{-0.000}$	$0.091^{+0.000}_{-0.000}$	0.648
	4	$0.933^{+0.002}_{-0.002}$	$0.979^{+0.001}_{-0.001}$	$0.109^{+0.001}_{-0.001}$	$0.052^{+0.000}_{-0.000}$	$0.057^{+0.000}_{-0.000}$	$0.089^{+0.000}_{-0.000}$	0.670
	ICN	$0.926^{+0.002}_{-0.002}$	$0.918^{+0.002}_{-0.002}$	$0.106^{+0.001}_{-0.001}$	$0.051^{+0.000}_{-0.000}$	$0.049^{+0.000}_{-0.000}$	$0.092^{+0.000}_{-0.000}$	-
bwd	0	$0.684^{+0.004}_{-0.004}$	$0.942^{+0.002}_{-0.002}$	$0.187^{+0.002}_{-0.002}$	$0.108^{+0.001}_{-0.001}$	$0.113^{+0.001}_{-0.001}$	$0.024^{+0.000}_{-0.000}$	0.543
	1	$0.694^{+0.004}_{-0.004}$	$0.918^{+0.003}_{-0.003}$	$0.189^{+0.002}_{-0.002}$	$0.110^{+0.001}_{-0.001}$	$0.107^{+0.001}_{-0.001}$	$0.023^{+0.000}_{-0.000}$	0.577
	2	$0.705^{+0.004}_{-0.004}$	$0.958^{+0.002}_{-0.002}$	$0.189^{+0.002}_{-0.002}$	$0.113^{+0.001}_{-0.001}$	$0.109^{+0.001}_{-0.001}$	$0.015^{+0.000}_{-0.000}$	0.579
	3	$0.678^{+0.004}_{-0.004}$	$0.951^{+0.002}_{-0.002}$	$0.181^{+0.002}_{-0.002}$	$0.113^{+0.001}_{-0.001}$	$0.102^{+0.001}_{-0.001}$	$0.023^{+0.000}_{-0.000}$	0.565
	4	$0.693^{+0.004}_{-0.004}$	$0.940^{+0.002}_{-0.002}$	$0.191^{+0.002}_{-0.002}$	$0.110^{+0.001}_{-0.001}$	$0.109^{+0.001}_{-0.001}$	$0.021^{+0.000}_{-0.000}$	0.582
	ICN	$0.651^{+0.004}_{-0.004}$	$0.903^{+0.003}_{-0.003}$	$0.192^{+0.003}_{-0.003}$	$0.109^{+0.001}_{-0.001}$	$0.108^{+0.001}_{-0.001}$	$0.004^{+0.000}_{-0.000}$	-

Table 7.1: Values for the efficiency, purity, resolution widths for the energy and the position predictions, and the background rejection at 95 % signal efficiency for the five trainings T_i in comparison to ICN-ETM for an offline ECL cluster energy between 0.15 - 0.25 GeV. The results are separated for offline ECL clusters in the forward endcap (FWD), the barrel region, and the backward endcap (BWD), for purity the position of the predicted clusters is used. For each metric value and each detector region, the best-performing training is marked in green. The energy resolution values are bias corrected.

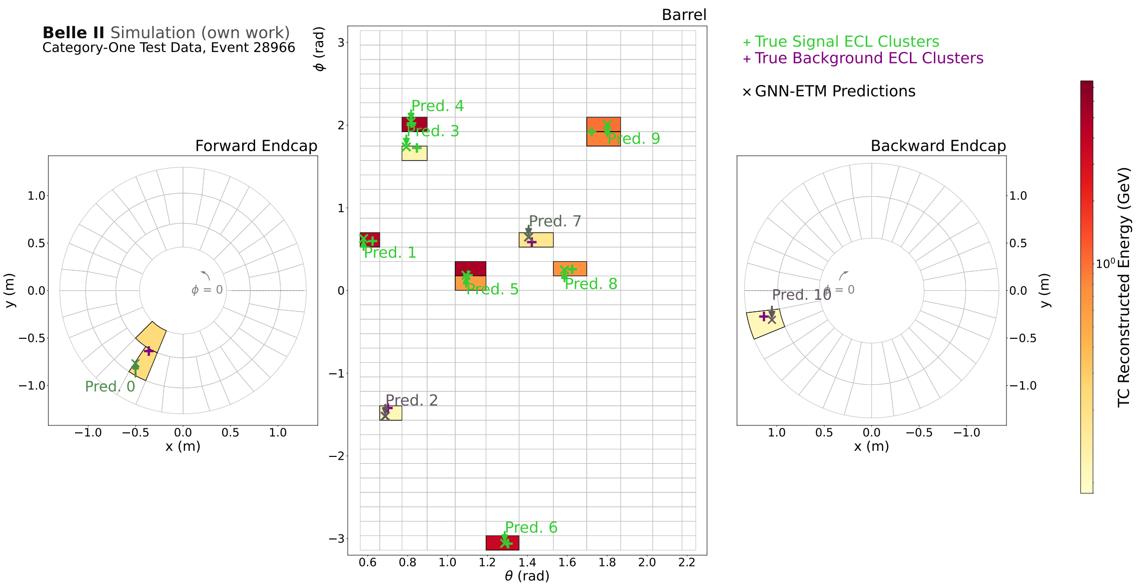


Figure 7.7: Example event with TCs and their reconstructed energies, the target offline ECL clusters and the GNN-ETM predictions. Only ECL clusters, which can be found on trigger level, with the definition in Section 4.2, are shown here. The ECL clusters are colorcoded with green depicting true signal clusters and purple being true background clusters. The GNN-ETM predictions are given a color value between purple, if the signal classifier output is 0, and green, if the signal classifier output is 1. A more vivid color shows a stronger confidence of the network in the prediction.

7.2.1 Category-One Dataset

In Fig. 7.7, an example event with several true signal and true background clusters and the corresponding predictions of the network is shown. As can be seen, the two true background clusters in the barrel region and the one in the backward endcap are also classified by the network as background. On the other hand, the background cluster in the forward endcap is misidentified as signal.

The overall output of the signal classifier value is shown in Fig. 7.8, separated into four distinct energy regions. The number of true background clusters decreases with higher energies, which the network learns and classifies clusters above a certain energy threshold nearly always as signal. For lower-energetic clusters, the classifier output distribution is flat in the case of true background clusters, with two peaks at 0 and 1. In comparison, the distribution for true signal clusters has a rising slope towards one. The usual U-shape of classifier distributions is lost due to the quantization of the sigmoid activation function, as explained in Section 6.3.3.

Even though the reconstructed energy for the distinction between signal and background clusters is one characterizing element, the network does not only learn an energy cut. In Fig. 7.9, the classifier output versus the reconstructed energy of the offline ECL clusters is shown. While an upper energy cut of approximately 0.5 GeV is seen, above which every cluster is classified as signal, no further structure can be identified.

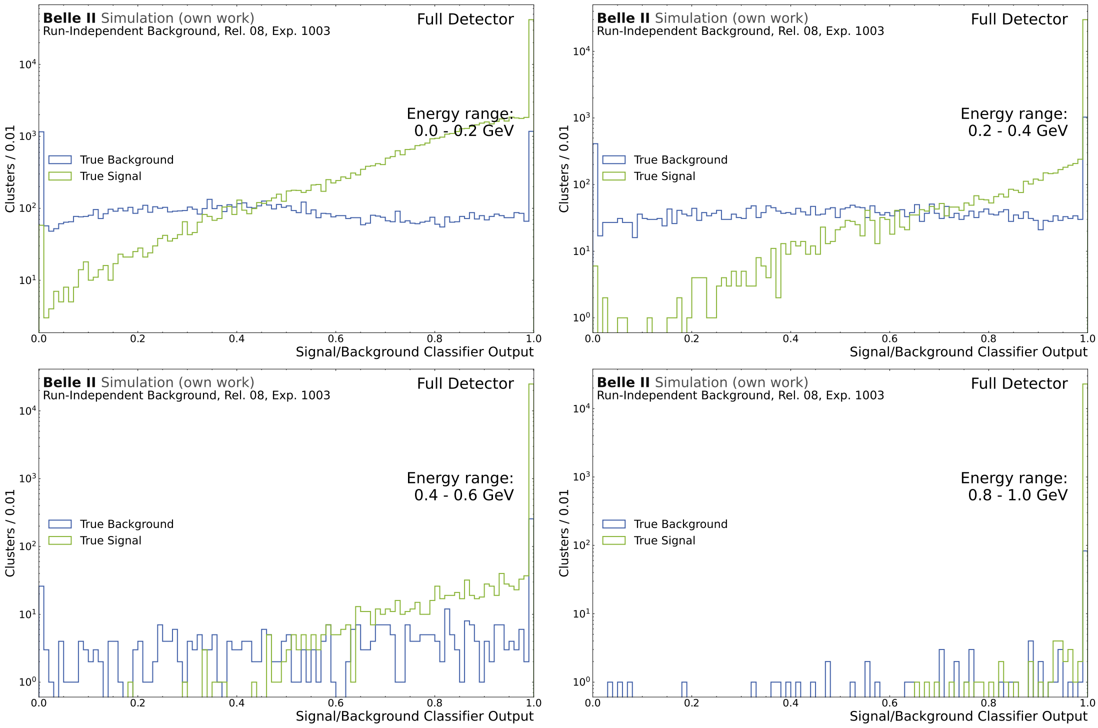


Figure 7.8: The distributions of the signal classifier output for true signal and true background clusters for four different energy regions. Top left shows the distribution for clusters with a reconstructed energy between 0.0 and 0.2 GeV, top right between 0.2 and 0.4 GeV, lower left between 0.4 and 0.6 GeV and lower right 0.8 and 1 GeV. The number of true background clusters drops significantly with higher energies. While the network output shows a peak at 1 also for true background clusters, a clear separation between the labels is visible.

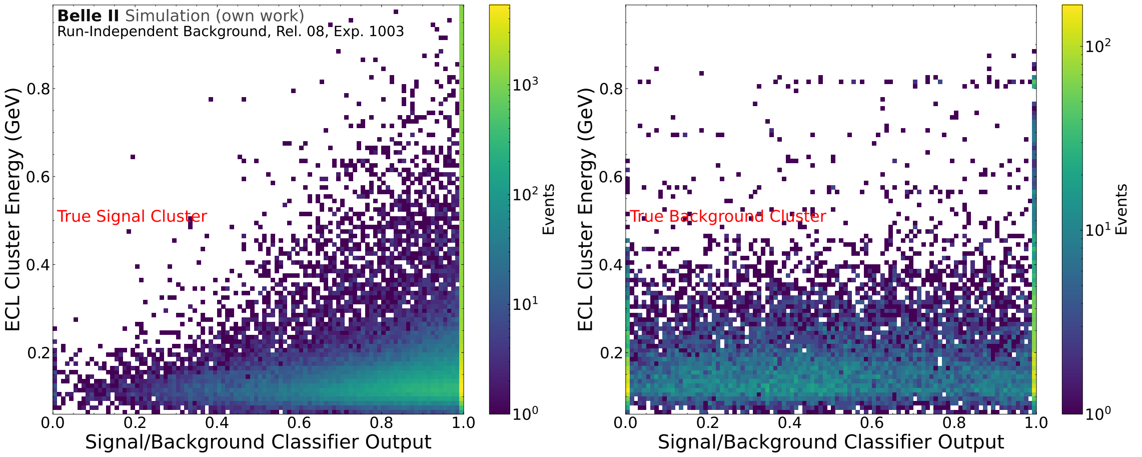


Figure 7.9: Output of the signal classifier versus the energy of the matched offline ECL cluster. The left plot shows only true signal offline ECL clusters, while the right plot shows true background offline ECL clusters. Clusters above 0.5 GeV are almost exclusively classified as signal. Below this energy, no significant structure is visible.

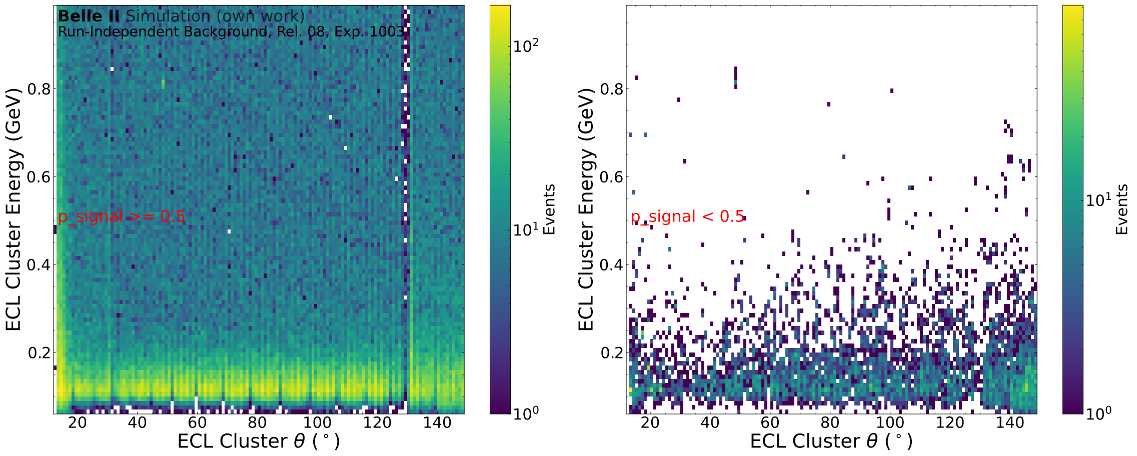


Figure 7.10: The energy of the matched offline ECL cluster versus its reconstructed θ position. The left plot shows clusters with a signal classifier value ≥ 0.5, so more likely being signal, while the right plot shows clusters with a signal classifier value below 0.5. While structures such as the gap in the ECL at 130° are visible, no evident structure of the classifier output can be seen.

In general, a higher percentage of background offline ECL clusters can be found in the backward endcap. In Fig. 7.10, the reconstructed θ angle versus the reconstructed energy of the offline ECL clusters found by the network is shown. The distributions are separated in clusters with a signal classifier value ≥ 0.5 and below 0.5. The distribution for clusters predicted as background follows the overall distribution of true background offline ECL clusters, but no further correlation is visible.

7.2.2 Evaluation of Trigger Rate on Pure Beam Background

Events simulated without any MC particles can mimic the behaviour of the trigger outside of correctly triggered physics processes. Due to the high bunch crossing rate but low collision probability at Belle II, the trigger rate on events not containing detector hits of particles coming from the interaction is the driving factor of the overall trigger rate. These events are then filtered out by the HLT. To avoid a total trigger rate above the capabilities of the current L1 trigger system, trigger bits have to be prescaled which leads to inefficiencies in physics analyses. Reducing the overall trigger rate on beam background-induced events with the help of the signal/background classifier can aid in loosening the requirements on trigger decisions needed for analyses.

To test this, the simulated pure beam background sample explained in Section 4.4 is used, which contains 500000 events with ri beam background. Out of these, only 8.9 % of events have a simulated ECL trigger response with at least one TC above the threshold of 100 MeV. For the ICN-ETM, the trigger bits are taken from TSIM, while for the GNN-ETM they are calculated by hand. The cut threshold for the signal classifier output is set as 0.7, which is the threshold for the barrel region for the Category-One dataset for

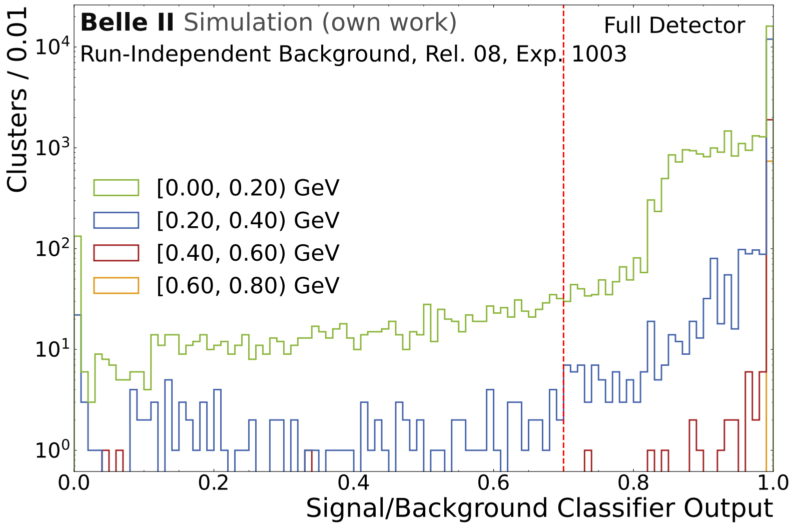


Figure 7.11: Signal/background classifier output value for all predicted GNN-ETM clusters in pure beam background samples. The clusters are separated into four different energy regions, based on their predicted energies. The red dotted line marks the signal cut threshold of 0.7.

200 MeV clusters.

The general output of the classifier for each predicted cluster can be seen in Fig. 7.11. As already seen in the Category-One dataset, clusters above approximately 0.5 GeV are always classified as signal. Below that, the probability of a classification as background increases, with clusters in the two lower energy regions showing a peak at 0.

In Fig. 7.12, the FTDL rates for the beam background events are seen. The rates are calculated according to Section 5.2, for the TSIM ICN-ETM bits, the GNN-ETM bits, and the trigger bits using only GNN-ETM clusters with a classifier output above 0.7. The total bit is the inclusive sum of all other bits, counting each event only once, even if it's triggered by several bits. The highest rate is generated by the *c2* bit, which is currently not used as an active trigger bit due to this reason. Using the signal classifier with the GNN-ETM, a rate reduction of 10 % for the *c2* bit is seen. For the *lml0* bit, which requires at least 3 clusters, the rate is decreased using the classifier threshold by nearly 1 kHz. This would have a significant impact on the overall data taking.

In summary, the signal/background classifier helps in reducing the beam background trigger rate while showing a good signal/background separation for MC events. The total rate of all ECL trigger bits is reduced by 10 % between the full GNN-ETM rate and the rate with the applied signal threshold. In Section 8.4, a similar evaluation is done on data as well.

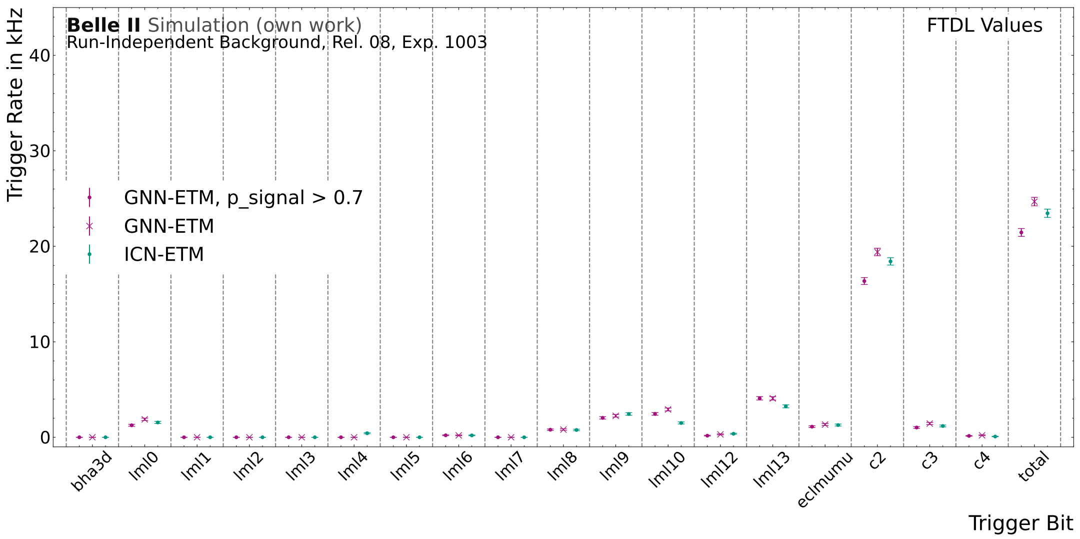


Figure 7.12: FTDL rate for the ECL trigger bits and the inclusive sum of all shown bits for the ICN-ETM, the GNN-ETM and the GNN-ETM using only clusters with a signal output value above 0.7.

7.3 Performance on Overlapping Clusters

The ICN-ETM can due to the algorithmic design not separate clusters that hit adjacent TCs or have any overlap (see Section 3.2). This is one area of possible improvement for the GNN-ETM. Particles such as π^0's, which decay into two photons, often leave signatures with close-by clusters. Separating those clusters opens up new possibilities in terms of trigger bits. Additionally, processes such as $e^+e^- \to a(\to \gamma\gamma)\gamma$ can mimic the signature of $e^+e^- \to e^+e^-(\gamma)$ and therefore be often vetoed by the L1 trigger Bhabha veto. Being able to separate the decay photons of the ALP can increase the trigger efficiency for this or similar processes.

To estimate the performance of the network in terms of close-by clusters, the fixed-energy overlap sample is used as explained in Section 4.4. Two photons with a small opening angle and the same energy are simulated. For the evaluation, only those events are used that contain exactly two signal clusters that can be found on trigger level. Clusters, where both only hit crystals in the TC, are indistinguishable and are therefore not used for this study. For both the network and ICN-ETM, the finding efficiency of both clusters and the resulting resolutions are evaluated.

In Fig. 7.13, the cluster finding efficiency in the barrel for an offline ECL cluster energy between [0, 0.5] GeV, [1, 1.5] GeV, [2.0, 2.5] GeV and [3, 3.5] GeV is shown depending on the cluster's distance to the nearest other cluster. The efficiency of the network is consistently higher than that of ICN-ETM, with a larger improvement for offline ECL clusters which are more than 20 cm apart.

The cluster finding efficiency for the different detector regions is shown in Fig. 7.14, exemplary for an offline ECL cluster energy between [1, 1.5] GeV. Clustering in the endcap

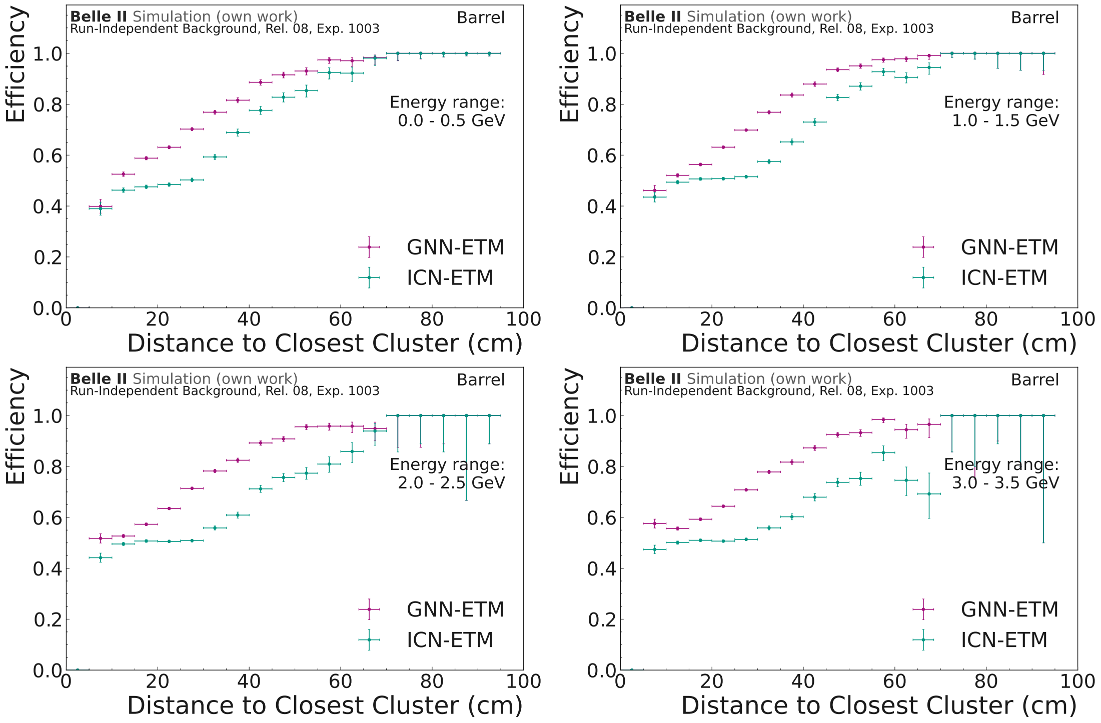


Figure 7.13: Efficiency of finding the two overlapping clusters in the barrel region dependent on the distance between them. The clusters are separated into offline ECL cluster energies between [0, 0.5] GeV (top left), [1, 1.5] GeV (top right), [2.0, 2.5] GeV (bottom left) and [3, 3.5] GeV (bottom right) are shown here. Only if both offline ECL clusters have at least one TC, in which they contributed the highest percentage of energy, the event is used in these plots. The network consistently outperforms the ICN-ETM by up to 20 %.

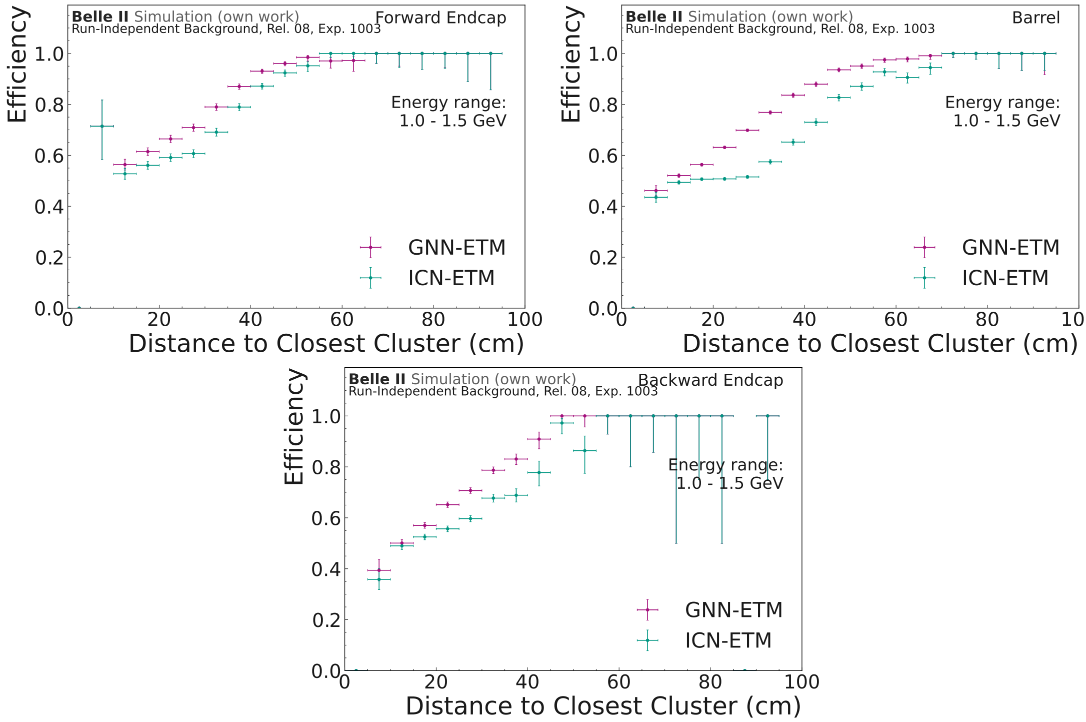


Figure 7.14: Efficiency of finding the two overlapping clusters dependent on the distance between them. The clusters have a reconstructed offline energy between 1 and 1.5 GeV. The upper left plot shows the finding efficiency for the forward endcap, the upper right plot for the barrel and the lower plot for the backwards endcap. Only if both clusters have at least one TC, in which they contributed the highest percentage of energy, the event is used in these plots.

is in general more difficult due to higher beam background energy depositions, higher material budget, and irregular shape of the TCs. This is visible as a worse separation efficiency for both ICN-ETM and the network. While the network still improves the finding efficiency in the endcaps, the improvement is less pronounced than in the barrel region.

A big factor for the efficiency on close-by offline ECL clusters is their respective energy, due to the matching done between ICN-ETM/ GNN-ETM clusters and the offline ECL clusters. The ICN-ETM returns in most cases one cluster instead of two with the energy of all TCs summed up. The requirement for the matching between trigger and offline clusters described in Section 5.1 then favors the matching to the higher-energetic cluster. This can be seen in Fig. 7.15. There the finding efficiency for the higher-energetic, the lower-energetic, and both clusters is shown in terms of the energy of both offline ECL clusters. The ICN-ETM algorithm has a significantly higher efficiency for the higher-energetic cluster. This is also visible, but slightly less prominent for the GNN-ETM cluster efficiency. The GNN-ETM network has much higher efficiency for finding both offline ECL clusters, especially for higher-energetic offline ECL clusters.

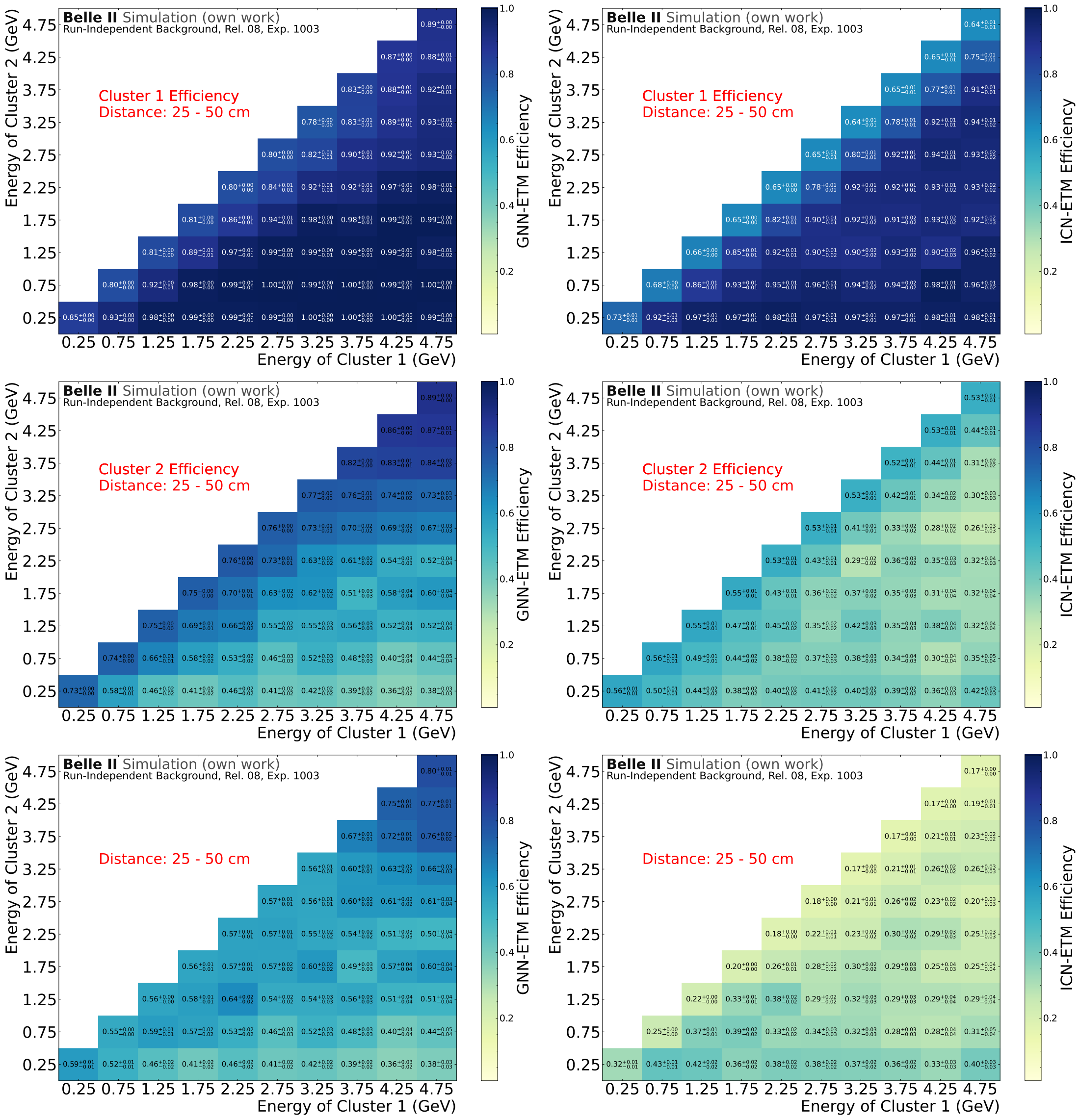


Figure 7.15: Efficiency of finding one of two offline ECL clusters or both in bins of the offline ECL cluster energy. The offline ECL clusters are sorted by energy with cluster 1 being the higher-energetic cluster. The left plot shows the GNN-ETM finding efficiency while the right plot shows the ICN-ETM efficiency. In the upper row, the efficiency of finding the higher-energetic cluster is shown, the center row shows the efficiency for finding the lower-energetic cluster and the lower row shows the efficiency for finding both offline ECL clusters. Especially the finding efficiency for both offline ECL clusters is significantly lower for the ICN-ETM.

Chapter 8

Evaluation on Physics Processes

In this chapter, the performance of the GNN-ETM is evaluated on two high rate processes at Belle II, $e^+e^- \to e^+e^-(\gamma)$ and $e^+e^- \to \mu^+\mu^-(\gamma)$. For both channels, signal MC data containing these processes, detector data taken in June 2022, called Exp. 26 data, and detector data taken in December 2024, called Exp. 35 data, is used. The advantage of using Exp. 35 data is the availability of the preprocessed TCs provided by the deployed GNN-ETM module described in Table 6.4. A disadvantage of this sample is the very high beam background, which lead to a degradation of the tracking efficiency. A selection to obtain a pure sample of the respective processes is applied, and the performance of the GNN-ETM in comparison to the ICN-ETM is shown.

Additionally, a background classifier test is performed on data, that has been triggered by a Poisson trigger and on the events that are triggered to be used as beam background overlays.

In this chapter, I will first explain the setup to analyze the performance of the GNN-ETM and ICN-ETM on collision data in Section 8.1. Then I will show the results on $e^+e^- \to e^+e^-(\gamma)$ in Section 8.2, including the selection, trigger metric performances and trigger rates. Afterwards, I will show the same studies for $e^+e^- \to \mu^+\mu^-(\gamma)$ in Section 8.3. At last, the performance of the signal/background classifier will be shown in Section 8.4.

8.1 Data Evaluation Setup

8.1.1 Selection of ECL Clusters

To correctly evaluate the performance of the GNN-ETM on data, several selections have to be applied that are not necessary for MC data. As described in Section 6.5, when a trigger signal is issued by the GDL, the GNN-ETM board not only writes out data in the main GNN-ETM trigger window which corresponds to ICN-ETM data windows 3 and 4 but also the data in the two windows before that. For a correct comparison to ICN-ETM, only the GNN-ETM clusters in the main GNN-ETM trigger window are used.

While ideally the ICN-ETM trigger decision window should consist of ICN-ETM data windows 3 and 4, as described in Section 3.2, sometimes the trigger decision is made using other ICN-ETM data windows. To simplify comparisons, only events in which the trigger decision window is correctly made up out of ICN-ETM data windows 3 and 4 are used for the comparison between GNN-ETM and ICN-ETM.

Furthermore, for ICN-ETM the here used unpacker in release-08 of `basf2` writes out a maximum of six clusters. Due to a bug discovered during my studies, the clusters written out are not necessarily the clusters in the trigger decision window, which would be the intended behavior. Instead, the unpacker returns the six highest-energetic clusters from all eight trigger data windows of this event. Since this can worsen the result of ICN-ETM when matched to offline clusters if clusters from the trigger decision window were lost in the unpacking, but also improve the result, if additional clusters from other data windows are present, all ICN-ETM clusters are used for comparison to the offline clusters with the assumption of them being from the trigger decision window. A fix for this is implemented and will be available in the next reprocessing of Belle II data.

To correctly compare the trigger clusters to the offline ECL clusters, only offline ECL clusters with a reconstructed time within the main GNN-ETM trigger window are used for the metric calculations. For this selection, several steps have to be applied. Fig. 8.1 shows an overview of the timing relations between offline timing, ICN-ETM time and GNN-ETM time. The timing of the offline ECL cluster is relative to the central timing position of the event, the global $\mathrm{T_0}$. This global $\mathrm{T_0}$ is determined by either the SVD, CDC, or ECL subdetector in this order. This calculation provides $EventT_0$, which is the offset between the global $\mathrm{T_0}$ and the $TriggerT_0$. $TriggerT_0$ is the $\mathrm{T_0}$ of the event determined online on the trigger system as a first estimate. This is given by the ECL, CDC, or TOP trigger system in this order. The ECL trigger determines the $TriggerT_0$ in over 99 % of events, where the time is given by the highest-energetic TC in that event. The ICN-ETM data window of this TC is also returned by the unpacker in `basf2`.

To correctly select offline ECL clusters that are within the trigger decision window, the start time of the trigger decision window relative to global $\mathrm{T_0}$ has to be determined. This can be done via

$$T_{\text{windowstart}} = 0 - EventT0 - TriggerT_0. \tag{8.1}$$

An offline ECL cluster which could be found by the trigger algorithms has to have a time between $[T_{\text{windowstart}}, T_{\text{windowstart}} + 250]$ ns.

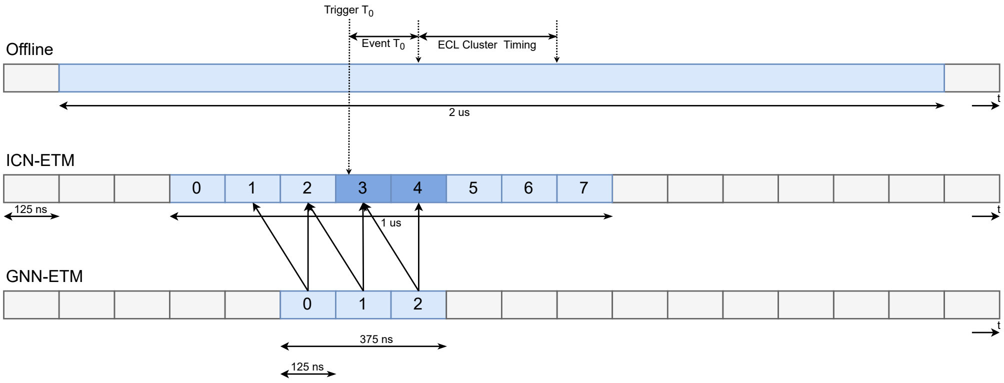


Figure 8.1: Relation between event timing, GNN-ETM timing and ICN-ETM timing. The upper line shows the timeline of one event, with the global T_0 marked by the central straight line. The offline ECL cluster timing is determined relative to that. *EventT0* is the difference between the global T_0 and the TriggerT_0. TriggerT_0 is determined by the trigger and usually lies within trigger data window 3. GNN-ETM data window 2 corresponds exactly to ICN-ETM data windows 3 and 4.

8.1.2 Data Samples

For this study, three datasets are used, simulated signal MC data for each process as well as two datasets consisting of different collision data. For the signal MC data, a dataset of 10 million events for $e^+e^- \to e^+e^-(\gamma)$, corresponding to an integrated luminosity of $0.034\,\text{fb}^{-1}$, and 1 million events for $e^+e^- \to \mu^+\mu^-(\gamma)$, corresponding to an integrated luminosity of $0.872\,\text{fb}^{-1}$, is used for both the verification of the high-level selection as well as for the evaluation of the GNN-ETM performance. Both are described in more detail in Section 4.4. For the collision data, the high-level selection is done on centrally reconstructed files, which are provided by the collaboration and contain already reconstructed objects such as offline ECL cluster, tracks, etc. They do not have the necessary objects available that are needed for the GNN-ETM performance analysis. For this, non-processed detector data is needed, of which only a subset is available for direct access. The two collision datasets used for this work are:

1. Exp. 26:

   a) Full dataset: Exp. 26, Runs 803, 849, 898; centrally reconstructed files, corresponding to an integrated luminosity of $0.8660\,\text{fb}^{-1}$

   b) Raw dataset: Exp. 26, Run 898; *bhabha_calib* and *mumu_tight_or_highm_calib* skims for $e^+e^- \to e^+e^-(\gamma)$ and $e^+e^- \to \mu^+\mu^-(\gamma)$, respectively,

2. Exp. 35:

   a) Full dataset: Exp. 35, Runs 2817 - 2905; centrally reconstructed files, corresponding to an integrated luminosity of $1.063\,\text{fb}^{-1}$

b) Raw dataset: Exp. 35, Runs 2882, 2890, 2895, 2896; runs taken with GNN-ETM module included.

The reasons for using two different collision datasets are twofold: firstly, the Exp. 35 dataset contains the preprocessed TCs returned from the GNN-ETM module as described in Section 6.5.4. This can serve as the input to the GNN-ETM, emulating the performance on the hardware as closely as possible. For the Exp. 26 dataset, the ICN-ETM TCs are used as input to the GNN-ETM. Secondly, both datasets have very different beam background conditions. As a comparison, Exp. 26, Run 898 has on average approximately 190 Out-of-Time ECL crystals (see Section 4.1.3), while Exp. 35, Run 2882 contains on average 390 Out-of-Time ECL crystals. The performance of GNN-ETM in both datasets can be used as a robustness study against different beam background levels.

As a general comment, collision data taken in Exp. 26 has undergone an offline luminosity calculation [99], making the integrated luminosity very precise. Collision data from Exp. 35 has not been used in any Belle II analysis yet and the integrated luminosity is taken from the online luminosity measurement [100], which can differ by 1 - 2 % from the correct integrated luminosity.

The selection and correction procedure for both processes is adapted from [99].

8.2 Analysis of the GNN-ETM performance for $e^+e^- \to e^+e^-(\gamma)$ events

The process $e^+e^- \to e^+e^-(\gamma)$ is the most common process at Belle II with a cross-section of 74.4 nb, if at least one of the charged particles is in the acceptance of the ECL detector. The general signature and structure of this process in Belle II is shown in Section 4.4. As explained in Section 3.1, one major task of the ECL L1 trigger is to detect and potentially veto these events. A pure selection of these events on trigger level can lead to a precise rejection without losing efficiency for other processes. Additionally, an efficient, downsampled sample must be triggered for luminosity measures and precision studies. Therefore, the GNN-ETM performance is evaluated on these events to compare the performance to the ICN-ETM.

As the GNN-ETM is trained on MC, the behavior on data is not necessarily the same. To correctly compare the performance of the GNN-ETM between data and MC, first, the selection procedure to obtain a clean sample for this process, independent of the GNN-ETM, is explained. After this selection, a data/MC comparison between high-level reconstruction objects is shown to show that a pure sample of $e^+e^- \to e^+e^-(\gamma)$ events is selected in both data and MC. This serves to remove additional effects introduced by the selection when evaluating the GNN-ETM performance on both data and MC and instead allows a direct comparison of the performance of the GNN-ETM. To quantize the agreement between data and MC, the pull per bin between both distributions is calculated via

$$\mathrm{Pull} = \frac{N(\mathrm{data}) - \mathrm{N(MC)}}{\sqrt{(\Delta\mathrm{Data})^2 + (\Delta\mathrm{MC})^2}} \quad , \tag{8.2}$$

with N(data) and N(MC) being the number of data or MC events in the respective bin, and ΔData and ΔMC being the statistical error in one bin calculated with the quadratic sum of weights of all events in one bin.

Then, the performance of the GNN-ETM on offline ECL cluster level, in comparison to the ICN-ETM, is shown for the selected events. The finding efficiency and purity for the e^- and e^+ offline ECL clusters is evaluated, as well as the energy and position resolution. At last, the trigger efficiency for $\mathrm{e}^+\mathrm{e}^- \to \mathrm{e}^+\mathrm{e}^-(\gamma)$ is shown and improvements to the trigger bits for GNN-ETM are discussed.

8.2.1 Selection Procedure and High-Level Data/MC Agreement

To obtain a clean sample of $\mathrm{e}^+\mathrm{e}^- \to \mathrm{e}^+\mathrm{e}^-(\gamma)$ events, the selection strategy is based on selecting two tracks with a large opening angle. Additionally, each track has to have a matched offline ECL cluster with an energy above 1 GeV, which rejects muon tracks efficiently. Selection criteria for both tracks and offline ECL cluster are used. Selections on values in the CM frame, such as the particle's reconstructed energy E boosted from the lab frame into the CM frame, are denoted by $E_{\mathrm{c.m.}}$.

General Selections

The following selections are applied to reconstructed tracks in data and MC:

- the transversal momentum p_t of each track must be $p_t > 0.2\,\mathrm{GeV/c}$,
- the CM energy of all particle candidates $E_{\mathrm{c.m.}}$ has to be $2.5 < E_{\mathrm{c.m.}} < 0.55 \cdot 10.58\,\mathrm{GeV}$,
- the cluster energy E_{cluster} of all offline ECL clusters matched to the tracks has to be $E_{\mathrm{cluster}} > 1.0$.

All selected tracks are then ranked by the CM track momentum $p_{\mathrm{c.m.}}$ and the two highest tracks are combined to an $\Upsilon(4\mathrm{S})$ candidate, if they fulfil the following selections:

- the sum of the two polar angles in the CM system $\theta_{\mathrm{c.m.}}$ has to be $|\theta_{\mathrm{c.m.}}(\mathrm{e1}) + \theta_{\mathrm{c.m.}}(\mathrm{e2})$ - 180°$| < 5°$,
- the absolute difference of the two azimuthal angles in the CM system $\phi_{\mathrm{c.m.}}$ has to be $||\phi_{\mathrm{c.m.}}(\mathrm{e1})$ - $\phi_{\mathrm{c.m.}}(\mathrm{e2})|$ - 180°$| < 5°$,
- the two tracks have to have opposite charges $q(\mathrm{e1})\times q(\mathrm{e2}) =$ -1.

This results in both particles being back-to-back in the CM system and of opposite charge.

For a very tight selection, an angle selection for the offline ECL clusters matched to the tracks to be only in the center barrel region is applied. This is done via the selection

- the offline ECL cluster angle θ_{cluster} has to be $37.8° < \theta_{\text{cluster}} < 120.5°$.

As for the trigger offline ECL clusters in the endcaps are of particular interest, as CDC-based track triggers have rather low efficiency in this region, this requirement will be loosened for the evaluation of the GNN-ETM performance.

HLT and L1 trigger selections

To correctly compare differential distributions for data and MC, the same trigger lines have to be applied. For $e^+e^- \to e^+e^-(\gamma)$ events, the HLT trigger lines

- ee_flat_30_35 = *eeFlat4* & $\textit{lml1}_{\text{PSNM}}$ & $\textit{bhapur}_{\text{FTDL}}$
- ee_flat_35_45 = *eeFlat5* & $\textit{lml1}_{\text{PSNM}}$ & $\textit{bhapur}_{\text{FTDL}}$
- ee_flat_45_60 = *eeFlat6* & $\textit{lml1}_{\text{PSNM}}$ & $\textit{bhapur}_{\text{FTDL}}$
- ee_flat_60_90 = *eeFlat7* & $\textit{lml1}_{\text{PSNM}}$ & $\textit{bhapur}_{\text{FTDL}}$
- ee_flat_90_180 = *eeFlat8* & $\textit{lml1}_{\text{PSNM}}$ & $\textit{bhapur}_{\text{FTDL}}$

are used. The lines *eeFlatX* are selections based on HLT reconstruction, whereas *lml1* and *bhapur* are based on the L1 trigger. The requirements of *eeFlatX* are one out of two options:

1. Two tracks with:
   - the two tracks have to have opposite charges $q(\text{e1}) \times q(\text{e2}) = \text{-1.}$,
   - the reconstructed mass of the tracks must be $M_{\text{2tracks}} > 5.29\,\text{GeV/c}^2$,
   - at least one offline ECL cluster matched to a track must be $E_{\text{cluster}}(\text{e2}) > 1.5\,\text{GeV}$ or $E_{\text{cluster}}(\text{e2}) > 1.5\,\text{GeV}$,
2. One track with a matched offline ECL cluster and one offline ECL cluster not matched to a track with:
   - the reconstructed energy of the offline ECL cluster matched to the track must be $E_{\text{cluster}} > 1.5\,\text{GeV}$,
   - the reconstructed mass of the track-cluster system must be $M_{\text{track,clst}} > 5.29\,\text{GeV}$.

The θ angle of the negative track determines the corresponding HLT trigger line. If the track in the single track case has a positive charge, the θ angle of the non-matched offline ECL cluster is used. So, for *eeFlat4* for example, the negative track has to have a θ angle in the laboratory frame between 30° and 35°. For *lml1*$_{\text{PSNM}}$, the input bit *ecl_ lml_ 1* is required, which is defined in Table 3.3, and the PSNM value, as explained in Section 3.1, is used. For *bhapur*$_{\text{FTDL}}$, the input bit *ecl_ bhapur* is required (see Table 3.2) and the FTDL value is used.

For data, exactly one of the mentioned HLT trigger lines including prescale has to be 1, with the corresponding prescale factors listed in Table 8.1. For MC, the filters before prescale are applied to not increase the statistical uncertainty by removing events. To correctly compare data and MC distributions, each event in data is weighted with the corresponding prescale factor. As the HLT trigger lines already require L1 trigger lines, no additional L1 trigger requirement is added.

Table 8.1: HLT Lines used for the selection of $e^+e^- \to e^+e^-(\gamma)$ and their corresponding prescale values in the different data-taking periods of Exp. 26 and Exp. 35.

HLT Line	Exp. 26 Prescale	Exp. 35 Prescale
ee_flat_30_35	18	36
ee_flat_35_45	15	30
ee_flat_45_60	5	10
ee_flat_60_90	2	4
ee_flat_90_180	1	2

HLT and L1 trigger corrections

As trigger efficiencies can differ in data and MC, dedicated trigger efficiency studies have to be made. For this, an orthogonal trigger line is used as a reference line and the trigger efficiency of the test trigger line is calculated in relation to the reference line. The trigger efficiency is then given as

$$\epsilon_{\text{TRG}} = \frac{N(\text{Ref\&Test})}{N(\text{Ref})}. \tag{8.3}$$

In this case, the efficiency of the HLT lines *eeFlatXX* and the efficiencies of the L1 trigger lines *lml1* and *bhapur* are tested separately. As both L1 trigger trigger lines are ECL trigger lines, the CDC trigger line *bfyo* is used as an orthogonal reference line. The *bfyo* line requires two full tracks in the CDC with an opening angle in the laboratory system of more than 90°. A completely orthogonal trigger line for the HLT lines is not available due to their dependence on both the ECL and CDC. Therefore, the HLT line *Estargt2_ GeV_ cluster* is used as a reference bit, as its requirement is very loose in comparison to the overall selection. This line requires at least one offline ECL cluster in the ECL with a CM energy above 2 GeV.

Table 8.2: Trigger efficiencies for $e^+e^- \to e^+e^-(\gamma)$ for the HLT and L1 trigger lines used in selection for MC, data taken in Exp. 26 and data taken in Exp. 35. The correction factors are calculated as $\epsilon_{\mathrm{data}}/\epsilon_{\mathrm{MC}}$.

Test Line	ϵ_{MC}	$\epsilon_{\mathrm{Exp.26}}$	$\epsilon_{\mathrm{Exp.35}}$	$f_{\mathrm{Exp.26}}$	$f_{\mathrm{Exp.35}}$
eeFlatXX	$1.000^{+0.0}_{-0.0011}$	$1.000^{+0.0}_{-0.0026}$	$0.997^{+0.0012}_{-0.0022}$	1	0.997
lml1	$1.000^{+0.0}_{-0.0241}$	$0.999^{+0.001}_{-0.0363}$	$0.997^{+0.0012}_{-0.0022}$	0.999	0.997
bhapur	$0.998^{+0.0011}_{-0.0023}$	$0.997^{+0.0016}_{-0.0042}$	$0.992^{+0.0022}_{-0.0032}$	1	0.995

In Table 8.2, the trigger efficiencies ϵ for MC, data from Exp. 26 and data from Exp. 35 are shown. Additionally, the correction factors $\epsilon_{\mathrm{data}}/\epsilon_{\mathrm{MC}}$ are shown. Necessary for this work is the understanding of the trigger efficiencies and the validation that the simulation of the trigger is reasonable in comparison to the real behaviour. For a full data/MC comparison, the correction factors are applied to each MC event. The trigger efficiency for both data and MC is very good, with the highest inefficiency being 0.8 % for Exp. 35 for the *bhapur* trigger bit. The GNN-ETM has to be able to also reach this performance on $e^+e^- \to e^+e^-(\gamma)$ events.

Offline ECL Cluster Reconstruction Efficiency

As the evaluation of the ICN-ETM and GNN-ETM performance relies on the existence of offline ECL cluster in the ECL, which for this analysis are matched to tracks, a study of the offline ECL cluster finding and reconstruction efficiency in both data and MC is done. For this, a track-based selection is applied. This selection follows the selection shown in Section 8.2.1, but removes the requirement of both tracks having matched offline ECL clusters and thereby also the selection on the offline ECL cluster θ angles. An angular selection is applied to the tracks:

- the track angles θ_{track} have to be $37.8^\circ < \theta_{\mathrm{track}} < 120.5^\circ$.

Again only the two tracks with the two highest momenta per event are used. The same HLT and L1 trigger requirements as before are used.

As the HLT filter requires at least one track matched to a offline ECL cluster with an energy above 1.5 GeV, the offline ECL cluster reconstruction efficiency is evaluated by testing the case of both tracks being matched to offline ECL clusters against only one track matched to a offline ECL cluster via

$$\epsilon_{\mathrm{cluster}} = \frac{N_{12}}{N_1 + N_2 + N_{12}} \quad , \tag{8.4}$$

with N_1 being the number of events, where $E_{\mathrm{cluster}}(\mathrm{e1}) > 1\,\mathrm{GeV}$ and $E_{\mathrm{cluster}}(\mathrm{e2}) \leq 1\,\mathrm{GeV}$, N_2 the orthogonal case and N_{12} the number of events, where both offline ECL clusters have

$E_{\text{cluster}} > 1\,\text{GeV}$. The calculated reconstruction efficiencies and the respective correction factors can be found in Table 8.3. The cluster reconstruction efficiency is very good, with a maximum inefficiency of 0.05 % for Exp. 26. Therefore, the GNN-ETM will be evaluated on these offline ECL clusters.

Table 8.3: Cluster reconstruction efficiencies for $\mathrm{e^+e^-} \to \mathrm{e^+e^-}(\gamma)$ for MC, data taken in Exp. 26 and data taken in Exp. 35. The correction factors are calculated as $\epsilon_{\text{data}}/\epsilon_{\text{MC}}$.

ϵ_{MC}	$\epsilon_{\text{Exp.26}}$	$\epsilon_{\text{Exp.35}}$	$\text{f}_{\text{Exp.26}}$	$\text{f}_{\text{Exp.35}}$
$0.999^{+0.0001}_{-0.0001}$	$0.995^{+0.0002}_{-0.0002}$	$0.999^{+0.0002}_{-0.0002}$	0.996	1

After application of all selections and correction factors, the corresponding data/MC comparisons can be seen in Fig. 8.2 for the CM energy of both tracks and Fig. 8.3 for the offline ECL cluster opening angle. For the CM energy, the two particles are sorted according to their $\theta_{\text{c.m.}}$, with particle 1 having the larger angle. The signal MC differential distributions agree very well with both the Exp. 26 and the Exp. 35 data distributions. While trigger effects and cluster inefficiencies result in scale factors for the MC distributions, these factors do not change the shape of the distributions at all. Especially the offline ECL cluster opening angle in both θ and ϕ show remarkable agreement. While the angular resolution for trigger clusters is worse than for offline ECL clusters due to the large TCs, the opening angles in both θ and ϕ are used for the detection of $\mathrm{e^+e^-} \to \mathrm{e^+e^-}(\gamma)$ events on trigger level. The agreement allows evaluating the GNN-ETM on both collision data and MC data and drawing conclusions on the performance of the GNN-ETM on data, without having to take additional effects into account. Performance differences can therefore be attributed to different prediction capabilities or different beam background levels instead of different process signatures.

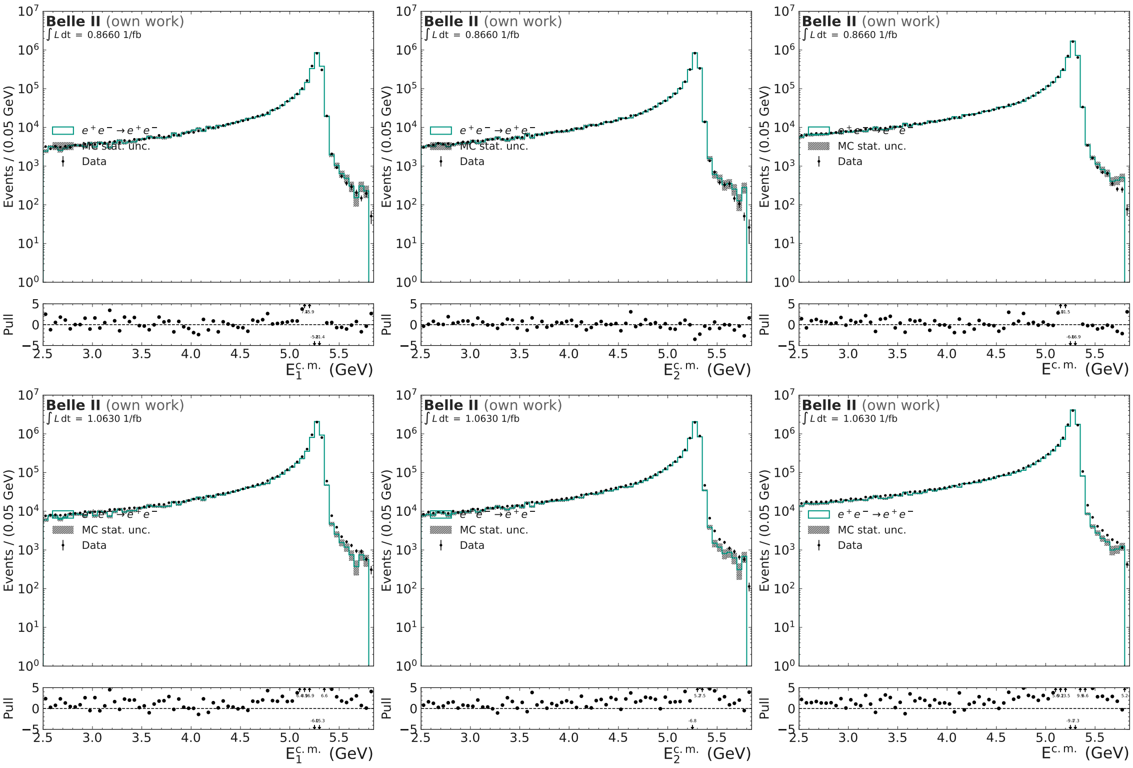


Figure 8.2: Center-of-Mass energy for the two reconstructed particles and their combined distribution for the $e^+e^- \to e^+e^-(\gamma)$ process. The particles are sorted by their center-of-mass θ angle, with particle 1 having the larger angle. MC events are scaled with the corresponding data luminosity. The data in the upper row is taken from Exp. 26, Runs 803, 848 and 898, while the data in the lower row is taken from Exp. 35, Runs 2817 - 2905.

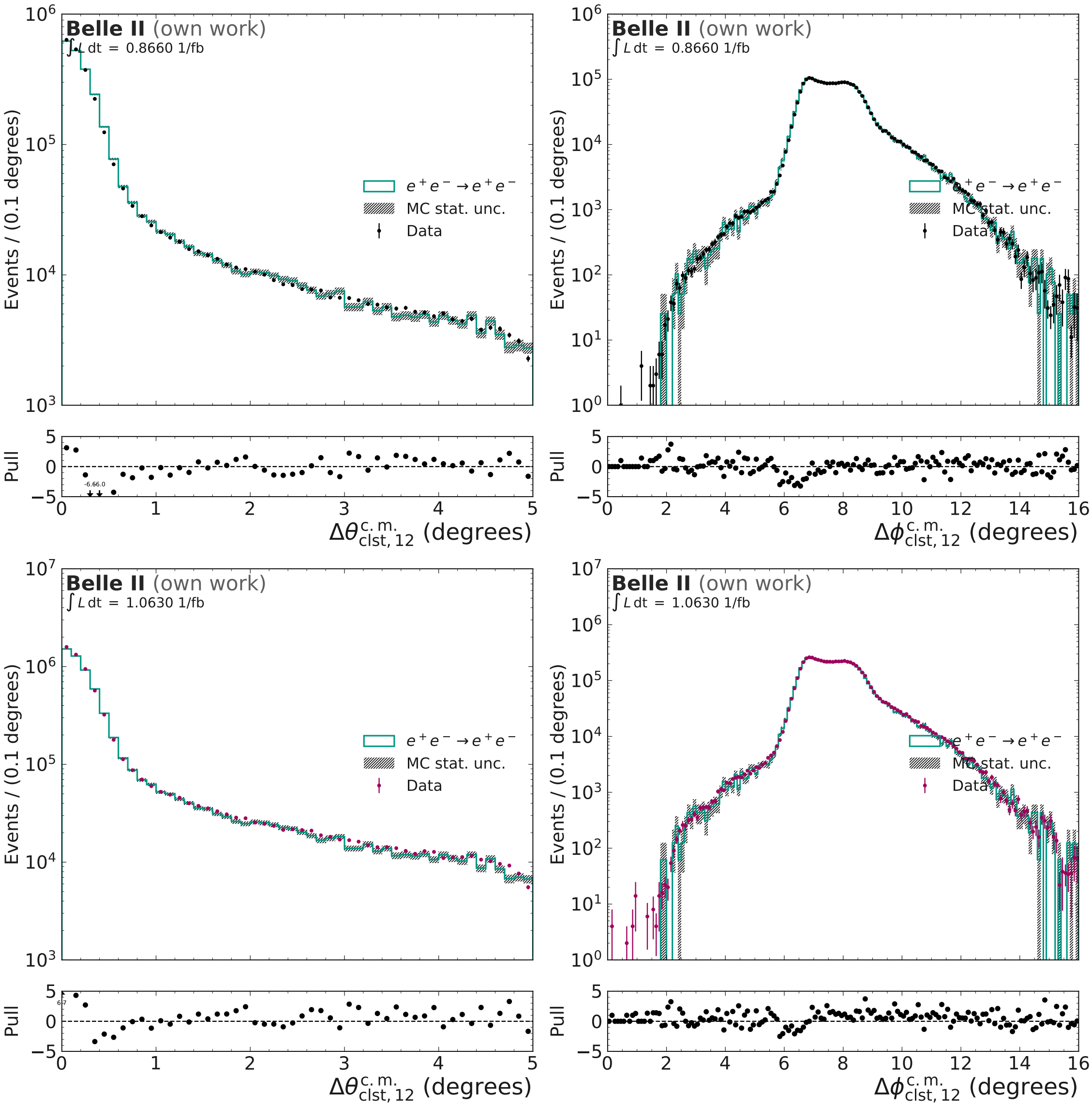


Figure 8.3: ϕ (left) and θ (right) offline ECL cluster opening angle for the two reconstructed particles for the $e^+e^- \to e^+e^-(\gamma)$ process. The offline ECL cluster opening angle is the relevant measure to reject $e^+e^- \to e^+e^-(\gamma)$ events in the ECL trigger due to missing track information. MC events are scaled with the corresponding data luminosity. The data in the upper row is taken from Exp. 26, Runs 803, 848 and 898, while the data in the lower row is taken from Exp. 35, Runs 2817 - 2905.

8.2.2 Trigger Data Sample Selection

Two raw data samples are used for the trigger studies for GNN-ETM, as explained in Section 8.1.2.

For Exp. 26, the *bhabha_ calib* skim sample of Run 898 is reprocessed and the ICN-ETM TCs are used as input to the GNN-ETM. For this skim, three different event signatures pass the selections:

1. Skim with two offline ECL clusters:
   - the ϕ difference between the two offline ECL cluster in the CM system has to be $165^\circ < |\phi_{\text{c.m.,cluster}}(\text{e1})$ - $\phi_{\text{c.m.,cluster}}(\text{e2})| < 178.5^\circ$,
   - the sum of the θ angles of both offline ECL cluster in the CM system has to be $178^\circ < \theta_{\text{c.m.,cluster}}(\text{e1}) + \theta_{\text{c.m.,cluster}}(\text{e2}) < 182^\circ$,
   - both offline ECL cluster have to have a CM reconstructed energy $E_{\text{c.m.,cluster}}(\text{e1})/10.58\,\text{GeV} > 0.4$ and $E_{\text{c.m.,cluster}}(\text{e2})/10.58\,\text{GeV} > 0.4$,
   - at least one offline ECL cluster has to have a CM reconstructed energy of $E^1_{\text{c.m.,cluster}}/10.58\,\text{GeV} > 0.45$ or $E^2_{\text{c.m.,cluster}}/10.58\,\text{GeV} > 0.45$,
2. Skim with 2 straight tracks:
   - the number of tracks has to be ≥ 2,
   - the maximum 3D opening angle between two tracks in CM frame has to be $\alpha_{\text{3D,max}} > 165^\circ$,
   - at least one track with a matched offline ECL cluster has to have an offline ECL cluster energy to track momentum ratio $E_{\text{cluster}}/p > 0.8$ and an absolute momentum $|p| > 1.85\,\text{GeV/c}$,
   - the tracks have to have opposite charge $q(\text{e1}) \times q(\text{e2}) =$ -1,
   - the sum of reconstructed energy of all tracks and offline ECL cluster in the event has to be $\sum_{\text{tracks,clsts}} E > 4\,\text{GeV}$,
3. Skim for possible radiative Bhabha processes:
   - Number of tracks $= 2$, with at least 1 CDC hit each and a valid dE/dx calculation,
   - the tracks have to have opposite charge $q(\text{e1}) \times q(\text{e2}) =$ -1,
   - both tracks have an energy to momentum ratio $0.7 < E/p < 1.3$
   - At least one track has to have a dE/dx within 30 % of that expected for an electron.

As not all data that passes this skim is needed for the calibration purposes of Belle II, a prescale of 3 is applied as well before storing the raw data.

Table 8.4: Number of events for the full datasets used for the trigger study and after selection. No angular selection on the offline ECL clusters is made here.

	MC sample	Exp. 26	Exp. 35
Before Selection	10^7	633350	3366688
After Selection	146723	76648	70670

For Exp. 35, the full collision data taken in the runs 2882, 2890, 2895, and 2896 is available and used for this study.

All selection criteria from Section 8.2.1 are applied to the full Exp. 35 dataset and to the Exp. 26 *bhabha_ calib* skim dataset besides the angle requirement

- $37.8^\circ < \theta_{\text{cluster}} < 120.5^\circ$

to also allow for particles to reach the endcaps.

The full and the remaining events for all three samples for the trigger study can be seen in Table 8.4.

8.2.3 Trigger Cluster Efficiencies and Resolutions

$e^+e^- \rightarrow e^+e^-(\gamma)$ often have high-energetic offline ECL clusters located in the endcaps (see Section 4.4), where reaching a high finding efficiency of these offline ECL clusters is by itself not a difficult task. The challenge for these events lies in the best possible energy and position reconstruction to allow for a clear identification to achieve a pure veto. The GNN-ETM is evaluated on the offline ECL clusters matched to the electron and positron particle candidates and compared to the performance of the ICN-ETM. The efficiency is expected to be high for both algorithms, while the GNN-ETM should improve both energy and position resolutions to improve the L1 trigger Bhabha veto.

For each dataset, the electron and positron offline ECL clusters are used to evaluate the performance for both trigger algorithms, if they are within the trigger timing window explained in Section 8.1.1 and fulfill the requirements of an offline ECL cluster visible on trigger level as described in Section 4.2.2. The GNN-ETM and ICN-ETM clusters in the events after selection are then matched to those offline ECL clusters using the matching requirements explained in Section 5.1. If a GNN-ETM or ICN-ETM cluster is matched to an offline ECL cluster, this offline ECL cluster is considered as found. The relative difference between the reconstructed energy of the offline ECL cluster and the matched trigger cluster, and the absolute difference in position values is then evaluated in the resolutions. This is done for the MC data, Exp. 26 data and Exp. 35 data separately.

Cluster Finding Efficiencies

Fig. 8.4 shows the finding efficiency for offline ECL clusters exemplary in the forward endcap for the electron offline ECL clusters and in the backward endcap for the positron offline ECL clusters. The finding efficiencies for the other detector regions can be found in The ECL trigger is highly important for the endcap regions, due to the inefficiency of the track trigger. The GNN-ETM outperforms the ICN-ETM for lower cluster energies, while the efficiency for higher energies is for both algorithms very close to 1.

The GNN-ETM efficiency for clusters around 4 GeV for electrons and 2.5 GeV is worse than the ICN-ETM efficiency, which is inconsistent to the studies on the technical datasets in Section 7.1. This and the fact that both algorithms show a large inefficiency for offline ECL clusters below 4 GeV for electrons and 2.5 GeV for positrons, shows the effect of Bremsstrahlung. Fig. 8.5 shows the distribution for the number of offline ECL clusters matched to the electron MC particle. The distributions are shown dependent of the energy of the offline ECL cluster to which the electron particle candidate in the selection is matched to. The MC matching algorithm of `basf2` attributes offline ECL cluster originating from secondary particles that do not stem from in-flight decays but from material interactions to the primary particle. If the electron radiates a high-energetic bremsstrahlung photon, it creates a second offline ECL cluster in the ECL, often very close to the main electron offline ECL cluster. Two effects then lead to the inefficiency for the trigger algorithms, which are shown as a schematic in Fig. 8.6: a) the trigger algorithm, either ICN-ETM or GNN-ETM, reconstructs one cluster instead of two, with the reconstructed energy being the sum of energies. This cluster then fails the requirements from Section 5.1 to be matched to the offline ECL clusters. b) The trigger algorithm reconstructs a cluster which can be matched to one of the offline ECL clusters. The cluster-track matching of `basf2` uses the offline ECL cluster as the particle candidate cluster, which is closer to the extrapolated track entry point in the ECL. The trigger cluster is then matched to one of the other close-by offline ECL clusters originating from the electron, either due to being a better energy or better position match.

The effect can be reduced to a minimum by requiring all offline ECL clusters in an event to be isolated, i.e. having a minimum distance to the next offline ECL clusters of 40 cm, which is the distance between two TC centers. In Fig. 8.7, the efficiency for isolated offline ECL clusters is shown. Both algorithms now have an efficiency of nearly 1 for all energy bins.

The higher efficiency for the GNN-ETM in several of the lower-energy bins is due to the GNN-ETM learning the existence of overlapping offline ECL clusters and separating them in some cases. One example can be seen in the event display in Fig. 8.8, where the GNN-ETM predicts two clusters for two offline ECL clusters, unlike the ICN-ETM.

However, there is also in the loss of efficiency for offline ECL cluster energies around 4 GeV for the electron and 2.2 GeV for the positron. By design, the GNN-ETM should

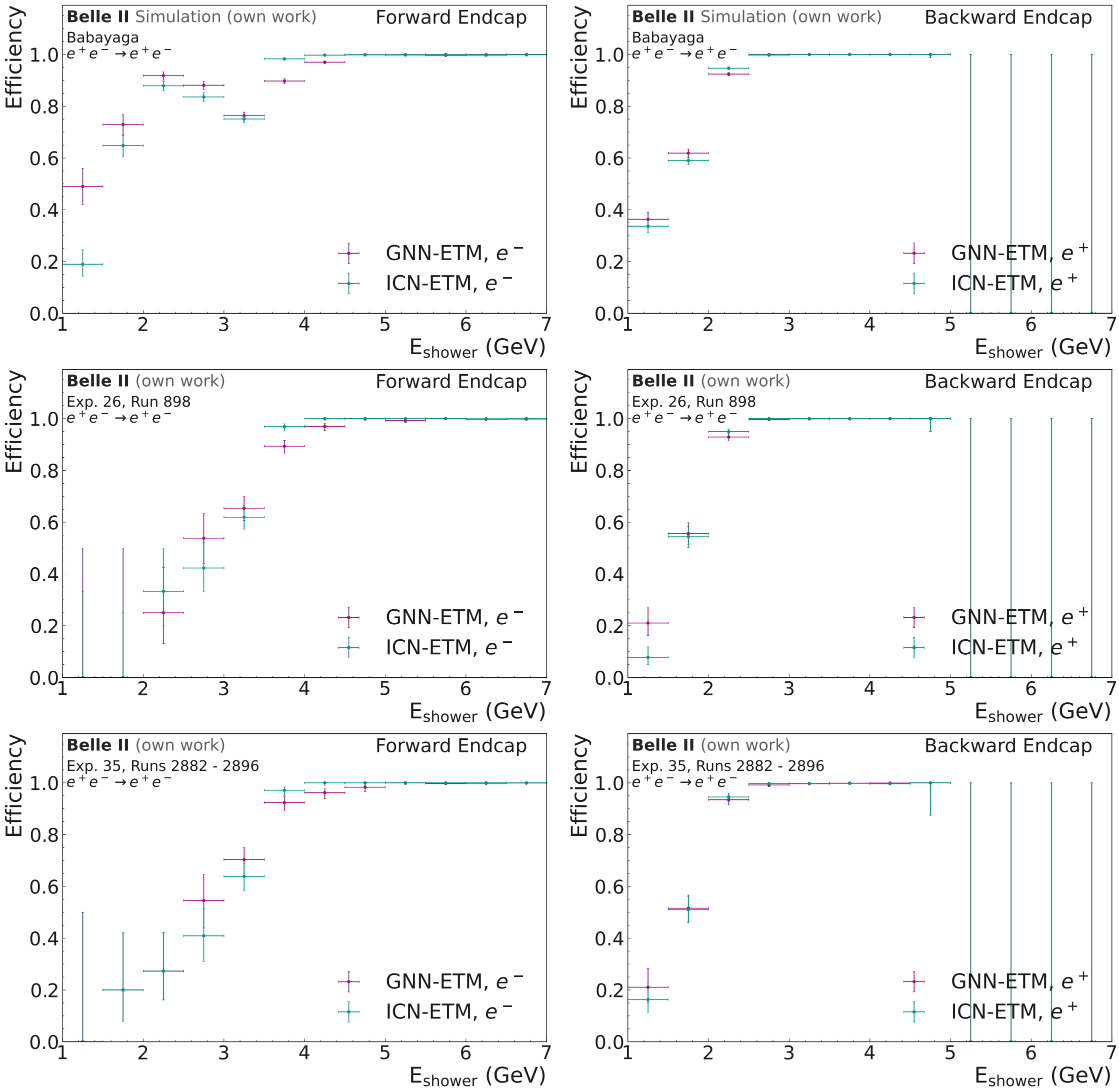


Figure 8.4: Efficiency for the electron offline ECL clusters in the backward endcap (left) and positron offline ECL clusters in the forward endcap (right) cluster for the GNN-ETM and the ICN-ETM algorithm. The top row shows the efficiency on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset.

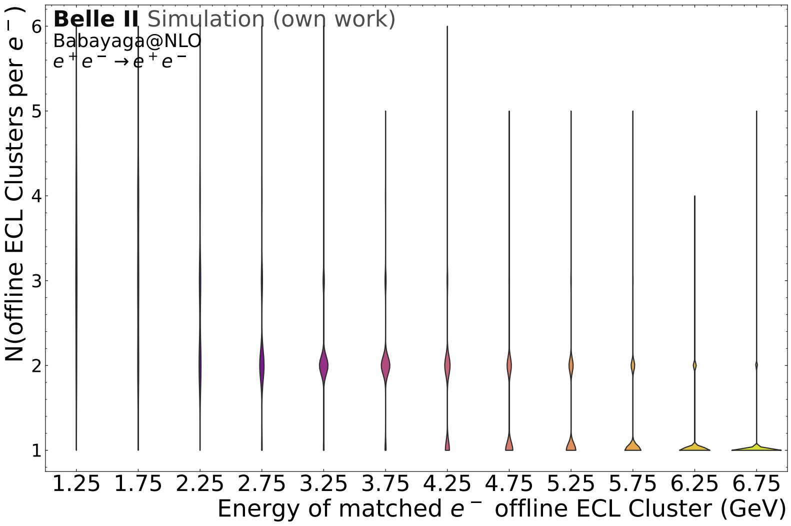


Figure 8.5: Distribution for number of offline ECL clusters MC above 100 MeV matched to the electron MC particle dependent on the energy of the offline ECL cluster attributed to the electron particle candidate. A large number of offline ECL clusters per electron MC particle show the effect of Bremsstrahlung or other material interactions, as offline ECL clusters from particles originating non-inflight decays are MC matched to the primary MC particle.

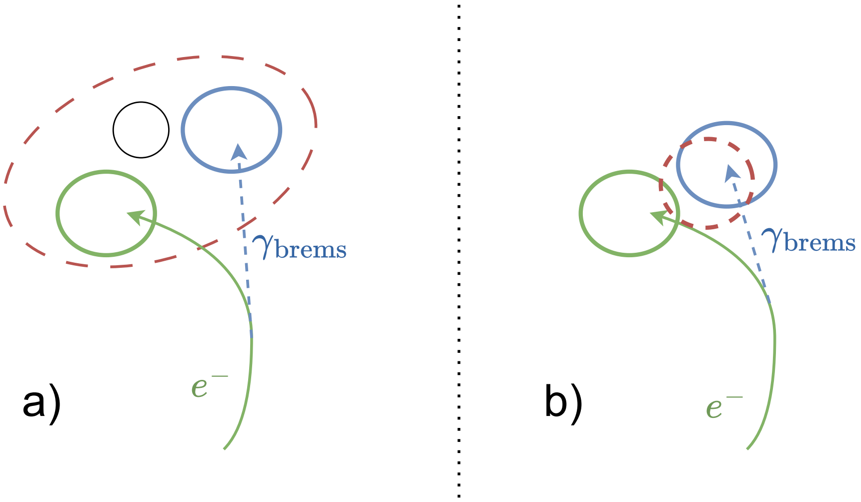


Figure 8.6: Schematic of an electron emitting a Bremsstrahlung photon, resulting in two offline ECL clusters for each particle. Both offline ECL clusters are MC matched to the electron. The red dotted line symbolizes a trigger cluster.

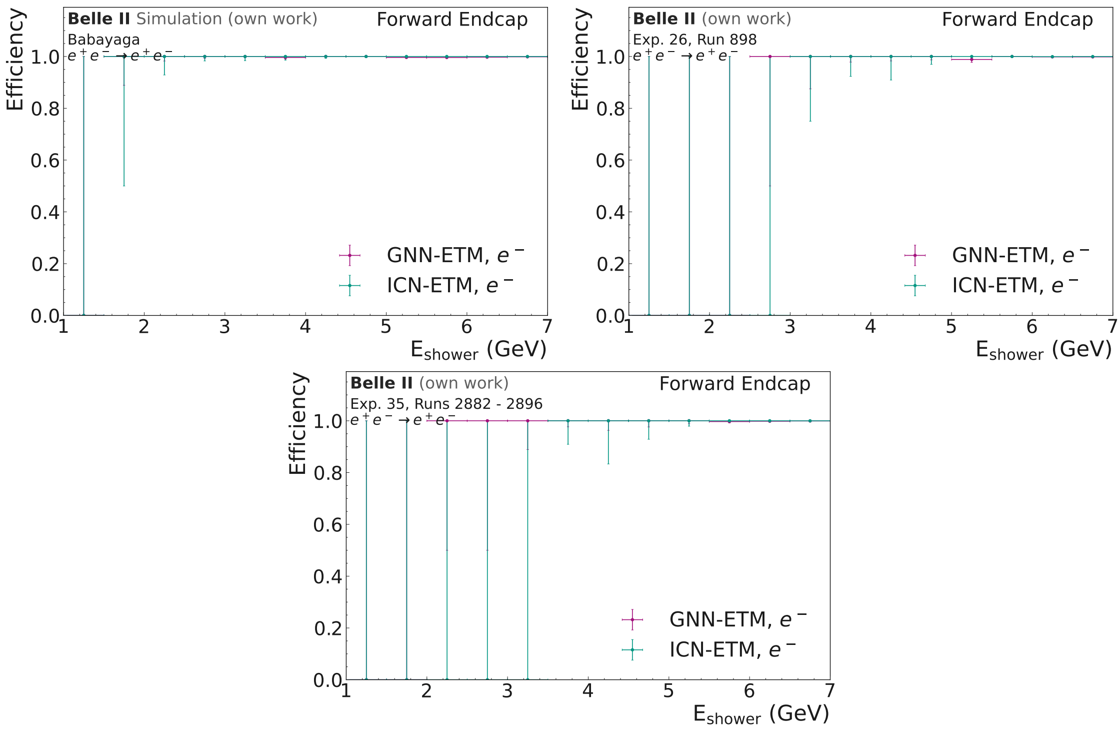


Figure 8.7: Efficiency for the electron offline ECL cluster for the GNN-ETM and the ICN-ETM algorithm. Only events, where all offline ECL clusters are at least 40 cm distance apart, are used for this efficiency calculation. The top left plot shows the efficiency for the MC dataset, the top right plot presents the Exp. 26 dataset, and the bottom plot the Exp. 35 dataset. The efficiency is shown for offline ECL clusters in the forward endcap.

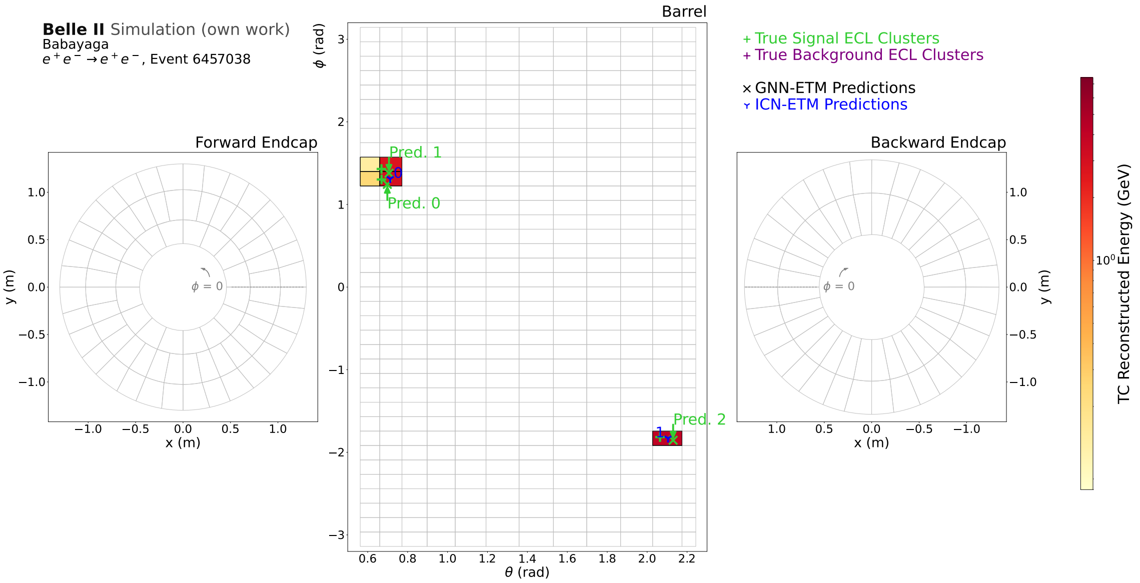


Figure 8.8: Event display of a $e^+e^- \to e^+e^-(\gamma)$ event with the offline ECL cluster, the ICN-ETM cluster and the GNN-ETM cluster. The offline ECL cluster in the bottom right part of the barrel is matched to the positron, while one of the offline ECL clusters in the top left part is matched to the electron. This is due to Bremsstrahlung, which produces a close-by offline ECL cluster due to the additional photon.

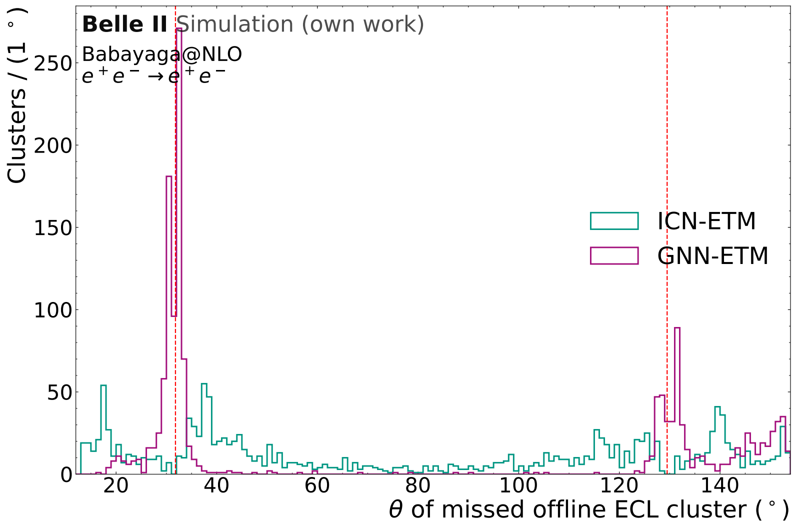


Figure 8.9: Angular distribution of split-up offline ECL clusters not found by either GNN-ETM or ICN-ETM. The offline ECL clusters are always MC matched to particles which have at least one other offline ECL cluster. The red dotted lines denote the gaps between the two endcaps and the barrel region. The sample evaluated is the signal MC sample for $e^+e^- \to e^+e^-(\gamma)$.

always have a higher efficiency, when the electron or positron is split up into several offline ECL clusters, as the ICN-ETM has no possibility of finding more than one of these offline ECL clusters. To check this, the offline ECL clusters for these split-up particles are evaluated. In Fig. 8.9, the θ distributions of offline ECL clusters not found by either ICN-ETM or GNN-ETM, where the matched MC particle has made at least one other offline ECL cluster, is shown for both electrons and positrons. While the angular distribution for the offline ECL clusters not found by ICN-ETM is following approximately the general angular distribution of the offline ECL clusters of Bhabha events, the missed GNN-ETM offline ECL clusters are notably very likely close to the gaps between one of the endcaps and the barrel region. This is a shortcoming of the GNN-ETM, most likely due to the training sample. Photons, which are the only particles used in the training sample, are less likely to split up into multiple offline ECL clusters, but generate one offline ECL cluster across the gap. This has to be mitigated in future trainings by adapting the training sample.

The finding efficiency for electron offline ECL clusters for offline ECL clusters with a reconstructed energy between 3 and 5 GeV and a reconstructed energy between 5 and 7 GeV for the three detector regions can be seen in Table 8.5. For higher energies, the efficiency is approximately 1 for both algorithms. The GNN-ETM can improve the finding efficiency in the forward endcap, which is the favoured direction of the electrons, by up to 10 % for all three datasets. For both algorithms, the finding efficiency in the forward endcap is significantly lower for the Exp. 26 and Exp. 35 datasets, which can partly be attributed to high beam background. Trainings with higher beam background datasets

Table 8.5: Cluster finding efficiency ϵ_{clst} for the GNN-ETM and ICN-ETM for offline ECL clusters matched to the electron for the three datasets for the different detector regions. The efficiency is shown for offline ECL clusters with a reconstructed energy between 1 and 3 GeV and for offline ECL cluster with a reconstructed energy between 5 and 7 GeV.

Dataset	**Region**	**Algorithm**	ϵ_{clst}, E $\in$ 1.0 - 3.0 GeV	ϵ_{clst}, E $\in$ 5.0 - 7.0 GeV
MC	Barrel	GNN-ETM	$0.6799^{+0.0179}_{-0.0185}$	$0.9996^{+0.0001}_{-0.0001}$
		ICN-ETM	$0.6228^{+0.0184}_{-0.0188}$	$1.0000^{+0.0000}_{-0.0000}$
	FWD	GNN-ETM	$0.8522^{+0.0113}_{-0.0120}$	$0.9983^{+0.0001}_{-0.0001}$
		ICN-ETM	$0.7865^{+0.0129}_{-0.0135}$	$1.0000^{+0.0000}_{-0.0000}$
Exp. 26	Barrel	GNN-ETM	$0.7586^{+0.0140}_{-0.0146}$	$0.9996^{+0.0001}_{-0.0001}$
		ICN-ETM	$0.7540^{+0.0138}_{-0.0143}$	$0.9999^{+0.0000}_{-0.0001}$
	FWD	GNN-ETM	$0.4444^{+0.0832}_{-0.0802}$	$0.9983^{+0.0003}_{-0.0004}$
		ICN-ETM	$0.3500^{+0.0782}_{-0.0709}$	$0.9997^{+0.0001}_{-0.0002}$
Exp. 35	Barrel	GNN-ETM	$0.6618^{+0.0142}_{-0.0145}$	$0.9995^{+0.0001}_{-0.0001}$
		ICN-ETM	$0.6417^{+0.0139}_{-0.0142}$	$0.9999^{+0.0000}_{-0.0001}$
	FWD	GNN-ETM	$0.4103^{+0.0800}_{-0.0756}$	$0.9989^{+0.0003}_{-0.0005}$
		ICN-ETM	$0.3333^{+0.0788}_{-0.0705}$	$0.9999^{+0.0001}_{-0.0002}$

might mitigate this loss in efficiency. In the barrel region, the GNN-ETM can improve the finding efficiency for the MC dataset, while being equal within errors for both Exp. 26 and Exp. 35.

Resolutions

The energy resolution η_E for GNN-ETM and ICN-ETM clusters matched to either electron or positron offline ECL clusters is defined as

$$\eta_E = \frac{E(\text{trigger cluster}) - E(\text{offline ECL cluster})}{E(\text{offline ECL cluster})}. \tag{8.5}$$

This is always calculated separately for GNN-ETM and ICN-ETM and can contain offline ECL clusters that have not been found by the other algorithm.

In this section, the resolutions for selected detector regions and offline ECL cluster energies are shown. In Section D.1.1, the remaining resolutions for ICN-ETM and GNN-ETM clusters matched to the electron are shown, while in Section D.1.1 the same is shown for the positron.

Fig. 8.10 shows the energy resolution for the electron for $4 < E(\text{offline ECL cluster}) < 4.5\,\text{GeV}$ on the left and $6 < E(\text{offline ECL cluster}) < 6.5\,\text{GeV}$ on the right.

For lower energies, both algorithms have a poor resolution distribution with a bias towards higher energy predictions. This is due to the aforementioned split-ups, where the trigger algorithms predict the energy deposition as one cluster, but the offline reconstruction returns two clusters with approximately half the energy each. The ICN-ETM shows for all three datasets a broad peak at around 0.5, which is due to an overshoot of the offline ECL cluster energy to a maximum of 50 %. The ICN-ETM returns as trigger cluster energy the sum of the TCs energy and the truthmatching explained in Section 5.1 matches the trigger cluster, if several offline ECL clusters would be matched to one trigger cluster, to the offline ECL cluster with the best energy agreement. Therefore, if a split-up into two close-by offline ECL clusters happens, the ICN-ETM will be matched to the higher-energetic one.

The GNN-ETM for the MC data has a clear peak around 0, but also displays the tail towards higher energies. For Exp. 26, this peak at 0 is significantly less pronounced, while for Exp. 35 both GNN-ETM and ICN-ETM consistently return higher energies than the offline ECL cluster. This worse performance is also consistent with the worse efficiency for Exp. 26 and Exp. 35 data in Fig. 8.4. One cause is very likely the higher beam background levels, as the MC data is simulated with the very low Exp. 1003 ri background levels. Training with higher beam background levels might improve this performance.

For higher energies, both ICN-ETM and GNN-ETM show a very narrow energy resolution distribution. The GNN-ETM distribution is less biased and more likely to return the correct offline ECL cluster energy, which can be used as an improvement for future trigger bit designs. The distributions are also very consistent over the three datasets.

The position resolution, calculated as

$$\eta_x = x(\text{trigger cluster}) - x(\text{offline ECL cluster}), \tag{8.6}$$

for x, y, and z, respectively, is especially important for the calculation of trigger bits, as they often depend on the position or region of the trigger cluster. In Fig. 8.11, the x and y resolution for all three samples for offline ECL clusters matched to the e^+ for $4 < \mathrm{E} < 4.5\,\mathrm{GeV}$ in the backward endcap can be seen. For this offline ECL cluster energy, most e^+ did not emit Bremsstrahlung photons, resulting in only one offline ECL cluster per particle and removing additional effects in position resolution due to the split-ups. The backward endcap has the highest beam background occupancy, resulting in worse position reconstruction performance in general.

The x resolution for the GNN-ETM has a bias towards smaller values. This can also be seen in the technical dataset evaluation in Chapter 7 and is a specific property of this model. In future trainings, models can be chosen that do not show this bias. The y resolution for the GNN-ETM is unbiased and centered around 0, with an additional smaller width than the resolution distribution for ICN-ETM. The final values for the y resolution width r_y after bias correction for ICN-ETM and GNN-ETM clusters matched to electron offline ECL clusters can be seen in Table 8.7. The GNN-ETM can improve r_y over the ICN-ETM

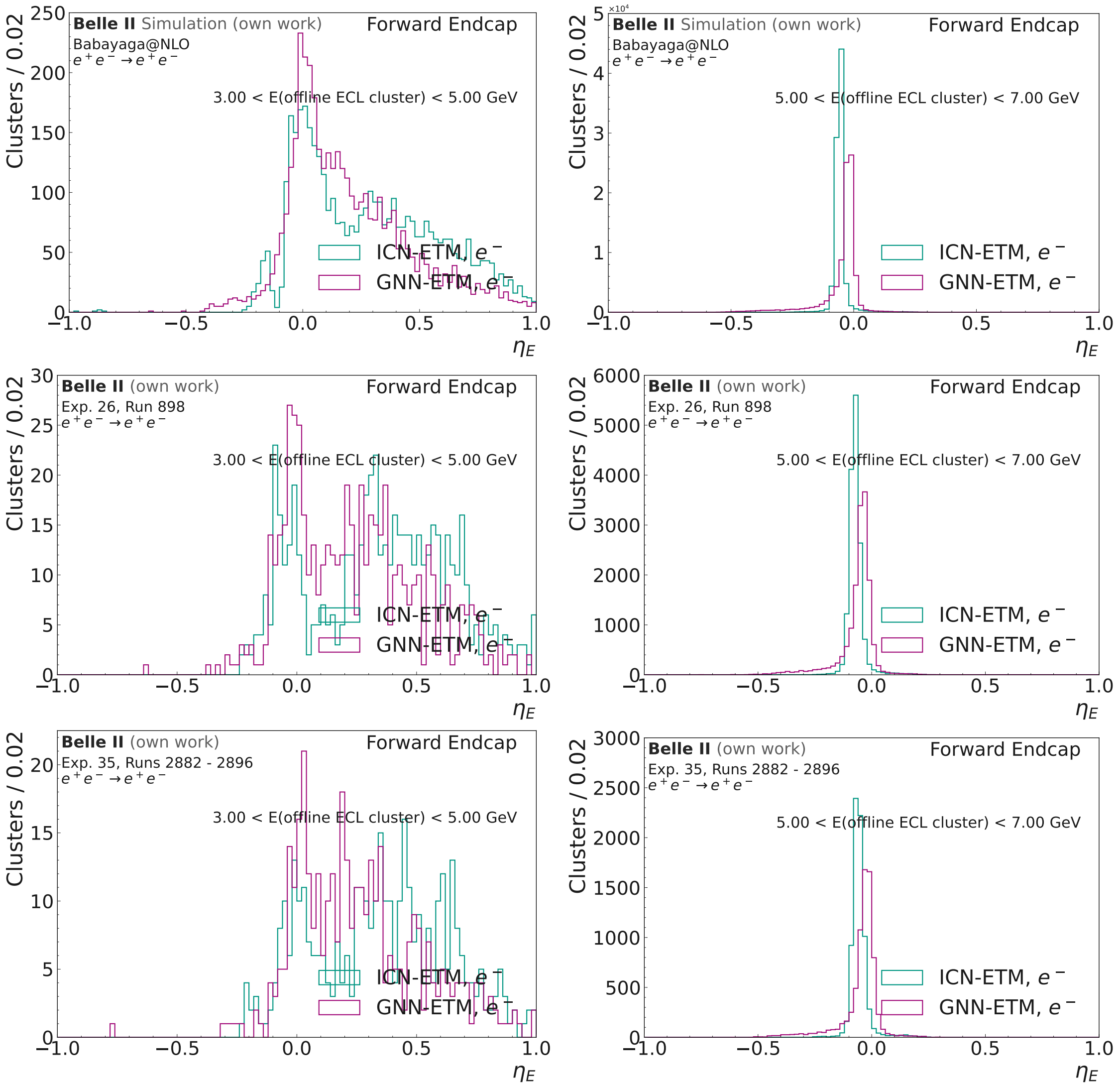


Figure 8.10: Energy resolution η_E for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster. The left column shows η_E for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$, while the right column shows η_E for $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$. The top row shows the energy resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the forward endcap.

Table 8.6: Energy resolution width r_{E} for the GNN-ETM and ICN-ETM for offline ECL clusters matched to the electron for the three datasets for the different detector regions. The efficiency is shown for offline ECL clusters with a reconstructed energy between 3 and 5 GeV and for offline ECL clusters with a reconstructed energy between 5 and 7 GeV. The backward endcap is omitted due to low entries. The resolution width is calculated with bias correction.

Dataset	**Region**	**Algorithm**	$\eta_{\mathrm{E},1}$	$\eta_{\mathrm{E},2}$
MC	Barrel	GNN-ETM	0.1013±0.0011	0.0231±0.0001
		ICN-ETM	0.1854±0.0027	0.0181±0.0001
	FWD	GNN-ETM	0.2101±0.0039	0.0268±0.0002
		ICN-ETM	0.2020±0.0035	0.0135±0.0001
Exp. 26	Barrel	GNN-ETM	0.0468±0.0006	0.0300±0.0002
		ICN-ETM	0.0493±0.0006	0.0252±0.0001
	FWD	GNN-ETM	0.1332±0.0053	0.0378±0.0004
		ICN-ETM	0.2227±0.0071	0.0225±0.0002
Exp. 35	Barrel	GNN-ETM	0.0454±0.0006	0.0265±0.0001
		ICN-ETM	0.0550±0.0008	0.0229±0.0001
	FWD	GNN-ETM	0.1955±0.0107	0.0375±0.0007
		ICN-ETM	0.1971±0.0072	0.0241±0.0003

consistently for both the barrel region and the forward endcap for all three datasets by up to 30 %.

The final energy resolution widths r_E, explained in Section 5.2, for ICN-ETM and GNN-ETM clusters matched to electron offline ECL clusters is shown in Table 8.6, for offline ECL clusters with an energy $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$. The backward endcap is omitted, due to the low number of entries, as the electron flight direction is predominantly in the forward direction. The widths are after bias correction. The GNN-ETM can improve the energy resolution for lower energies for the forward endcap for all datasets. For the higher energies, the GNN-ETM shows a larger width due to larger tails towards lower energies.

In Fig. 8.12, the output of the classifier can be seen as a cross-check. As all clusters have at least 1 GeV due to the selection, the classifier, as shown in Section 7.2.1, has learned to classify them as signal.

To summarize, the GNN-ETM can improve the cluster finding efficiency for low-energetic offline ECL clusters over the ICN-ETM and have equal performance for isolated or high-energetic offline ECL clusters. The GNN-ETM shows an inefficiency in predicting multiple offline ECL clusters located around the gaps of the detector regions, which should

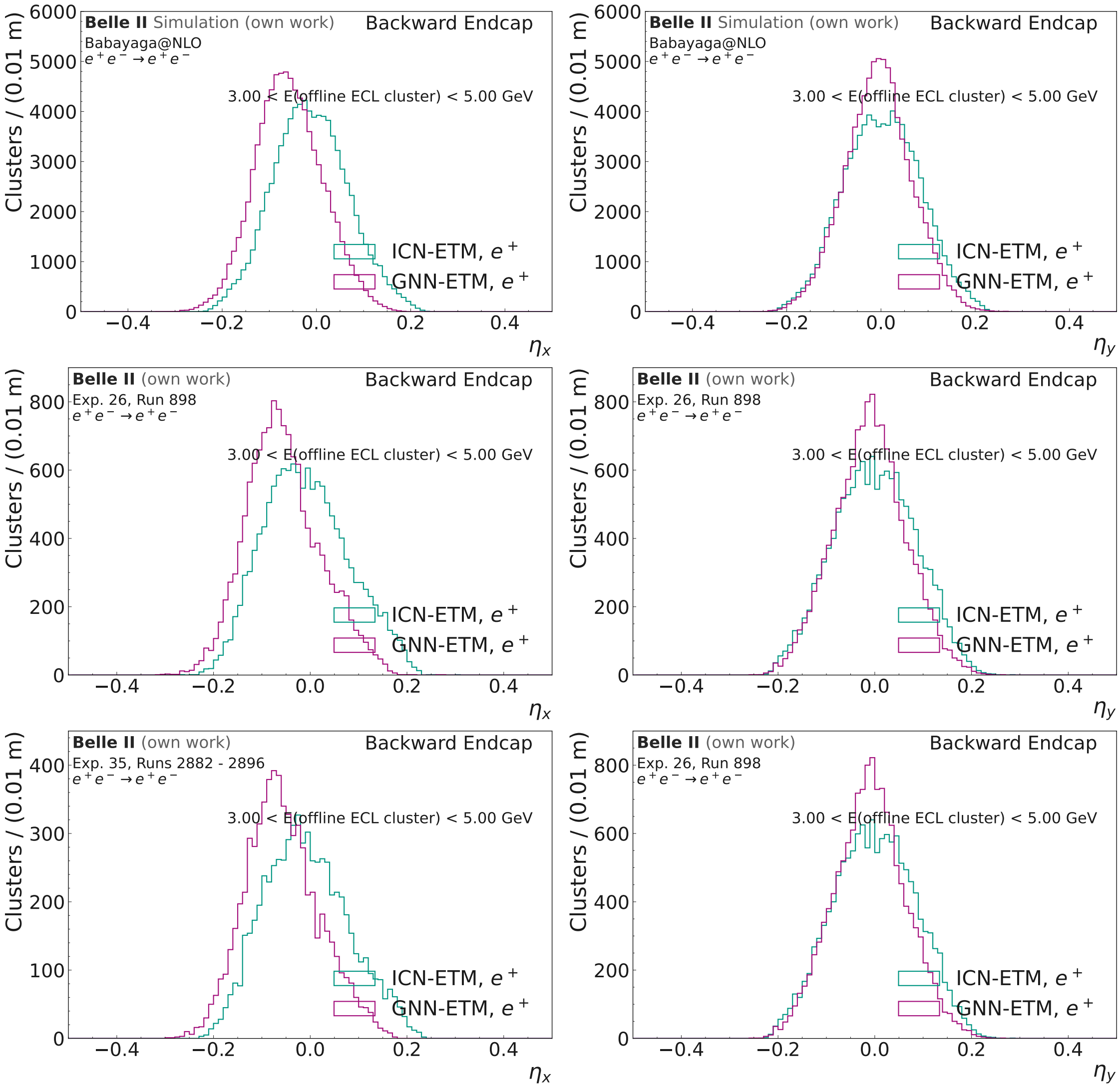


Figure 8.11: η_x and η_y for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$. The left column shows the η_x, while the right column shows the η_y. The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

Table 8.7: y resolution width r$_y$ for the GNN-ETM and ICN-ETM for offline ECL clusters matched to the electron for the three datasets for the different detector regions. The efficiency is shown for offline ECL clusters with a reconstructed energy between 3 and 5 GeV and for offline ECL clusters with a reconstructed energy between 5 and 7 GeV. The backward endcap is omitted due to low entries. The resolution width is calculated without bias correction.

Dataset	**Region**	**Algorithm**	$r_{\mathrm{y},3-5}$	$r_{\mathrm{y},5-7}$
MC	Barrel	GNN-ETM	0.0459±0.0003	0.0383±0.0001
		ICN-ETM	0.0501±0.0004	0.0501±0.0002
	FWD	GNN-ETM	0.0626±0.0010	0.0616±0.0002
		ICN-ETM	0.0699±0.0010	0.0852±0.0002
Exp. 26	Barrel	GNN-ETM	0.0399±0.0003	0.0362±0.0002
		ICN-ETM	0.0506±0.0004	0.0505±0.0003
	FWD	GNN-ETM	0.0663±0.0029	0.0573±0.0005
		ICN-ETM	0.0771±0.0023	0.0844±0.0005
Exp. 35	Barrel	GNN-ETM	0.0391±0.0003	0.0357±0.0002
		ICN-ETM	0.0510±0.0004	0.0495±0.0003
	FWD	GNN-ETM	0.0653±0.0036	0.0556±0.0006
		ICN-ETM	0.0747±0.0036	0.0861±0.0008

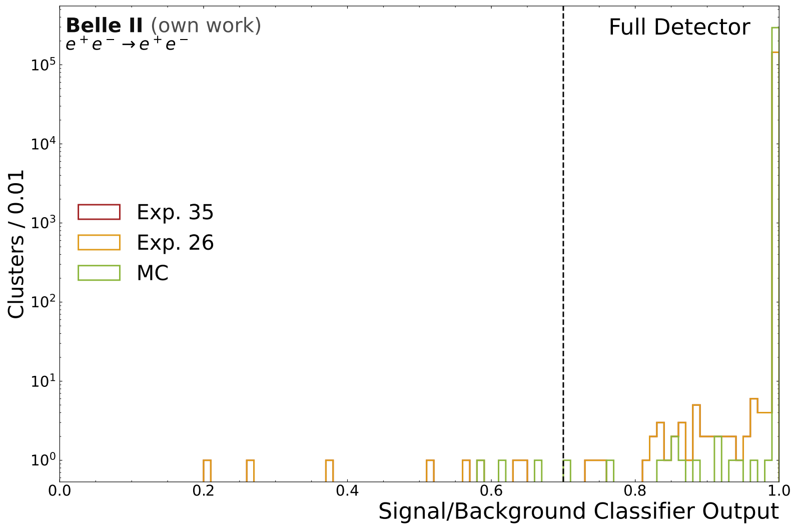


Figure 8.12: Signal classifier output for GNN-ETM clusters matched to either an e^+ or e^- offline ECL cluster. The distributions are shown for all three data samples. The black dotted line shows the chosen signal cut.

be improved in further trainings. The GNN-ETM can improve the cluster finding efficiency by up to 10 % in the backward endcap consistently for all three datasets and improve the position resolution by up to 30 % while keeping the performance for different beam background levels. The energy resolution is slightly wider than that of ICN-ETM but has a reduced bias.

8.2.4 Trigger Bit Efficiency

For the final trigger performance, the most relevant metric is the trigger efficiency or by relation, the trigger rate, often in dependence of the deposited energy or the angular distribution. Since the selection returns a very pure $e^+e^- \to e^+e^-(\gamma)$ sample, in this case the overall trigger efficiency is evaluated. For $e^+e^- \to e^+e^-(\gamma)$, the ECL trigger needs to have a high trigger efficiency to be able to ensure a pure veto for these events.

In Fig. 8.13, the trigger efficiency for all current Belle II ECL trigger bits is shown. The definition for these bits can be found in Table 3.2 and in Table 3.3. For the GNN-ETM, only clusters that have a signal classifier output above 0.7, which is determined on the technical sample (see Section 7.2.1) are used for the trigger bit calculation. For both algorithms, the efficiency for the *bha3d* bit is very high, which is currently used for detecting $e^+e^- \to e^+e^-(\gamma)$. For lower beam background levels, which are present in the MC data and to a lesser extent in the Exp. 26 dataset, the ICN-ETM has a slightly higher efficiency of 0.5 %. With high beam background levels however, for the Exp. 35 dataset, the GNN-ETM has a minor increase in efficiency for the *bha3d* bit. This is likely due to the improved energy resolution of the GNN-ETM over the ICN-ETM.

The current trigger bits however are optimized for the spatial TC boundaries used by the ICN-ETM. The *bhad3d* bit requires two clusters with a sum of θ angles between 165° and 190°, and a difference in ϕ angles between 160° and 200°, in the CM frame. As the transformation between the laboratory and the CM frame for the ICN-ETM follows a fixed LUT, the angle cuts are exactly in between two TC centers.

The GNN-ETM does not have the same boundaries, as the position prediction is allowed to be a floating value and especially for higher-energetic offline ECL clusters improved over the ICN-ETM position determination. In Fig. 8.14, the distributions for $\sum \theta^{\text{cm}}_{\text{clst}}$ and $\Delta\phi^{\text{cm}}_{\text{clst}}$ are shown including the selections of the *bha3d* trigger on those values. Both plots only contain the GNN-ETM or ICN-ETM cluster, which are matched to an electron or a positron offline ECL cluster. The plots on the right show the ICN-ETM distribution and the optimization of the boundaries for the veto are visible. They align with the edges of the TCs. Both GNN-ETM and ICN-ETM have nearly 100 % of events within these boundaries. For the GNN-ETM distributions on the left however, these requirements are not optimized. Especially for the distributions for collision data, Exp. 26 and Exp. 35, the upper ϕ requirements and lower θ could be tightened to improve the purity of the *bha3d* veto.

Additionally, an improvement of the GNN-ETM performance is the better energy

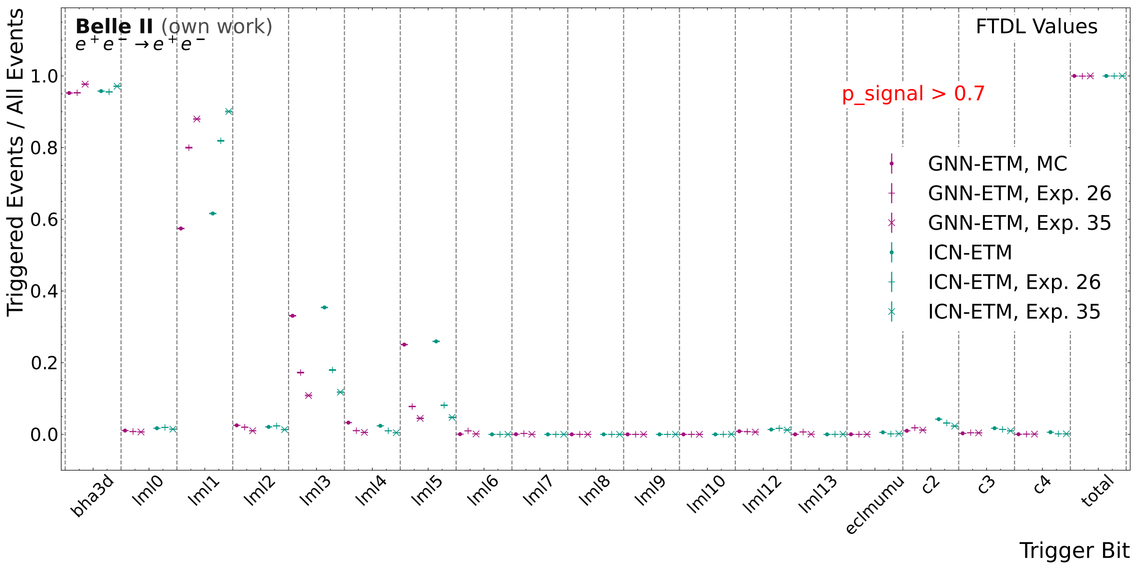


Figure 8.13: Trigger efficiency for the different ECL trigger bits. The trigger efficiency is shown for all three data samples and for GNN-ETM and ICN-ETM in comparison. The trigger efficiency is shown without any prescale factor applied.

resolution. This is also used in the calculation of the *bha3d* veto, as both clusters have to be above 3 GeV and at least one of them above 4.5 GeV. A better energy resolution allows for tighter cuts which improve the purity of the sample. In Fig. 8.15, the CM energy distributions for ICN-ETM and GNN-ETM are shown including the two energy thresholds and fractions of events that pass the respective cut. The energy distribution of the GNN-ETM is significantly closer to the energy distribution of the offline ECL cluster and has a higher percentage of events within the given cut. A tightening of the cut value is feasible with the GNN-ETM performance.

As a summary, within the given boundaries of the current trigger bits, the GNN-ETM performance is equivalent to the ICN-ETM performance with a high efficiency for this sample. Possible improvements in the trigger bit determination are a stricter energy selection for the *bha3d* veto, to increase the purity of the sample. This has to be tested in comparison to other processes, such as $e^+e^- \rightarrow \gamma\gamma(\gamma)$.

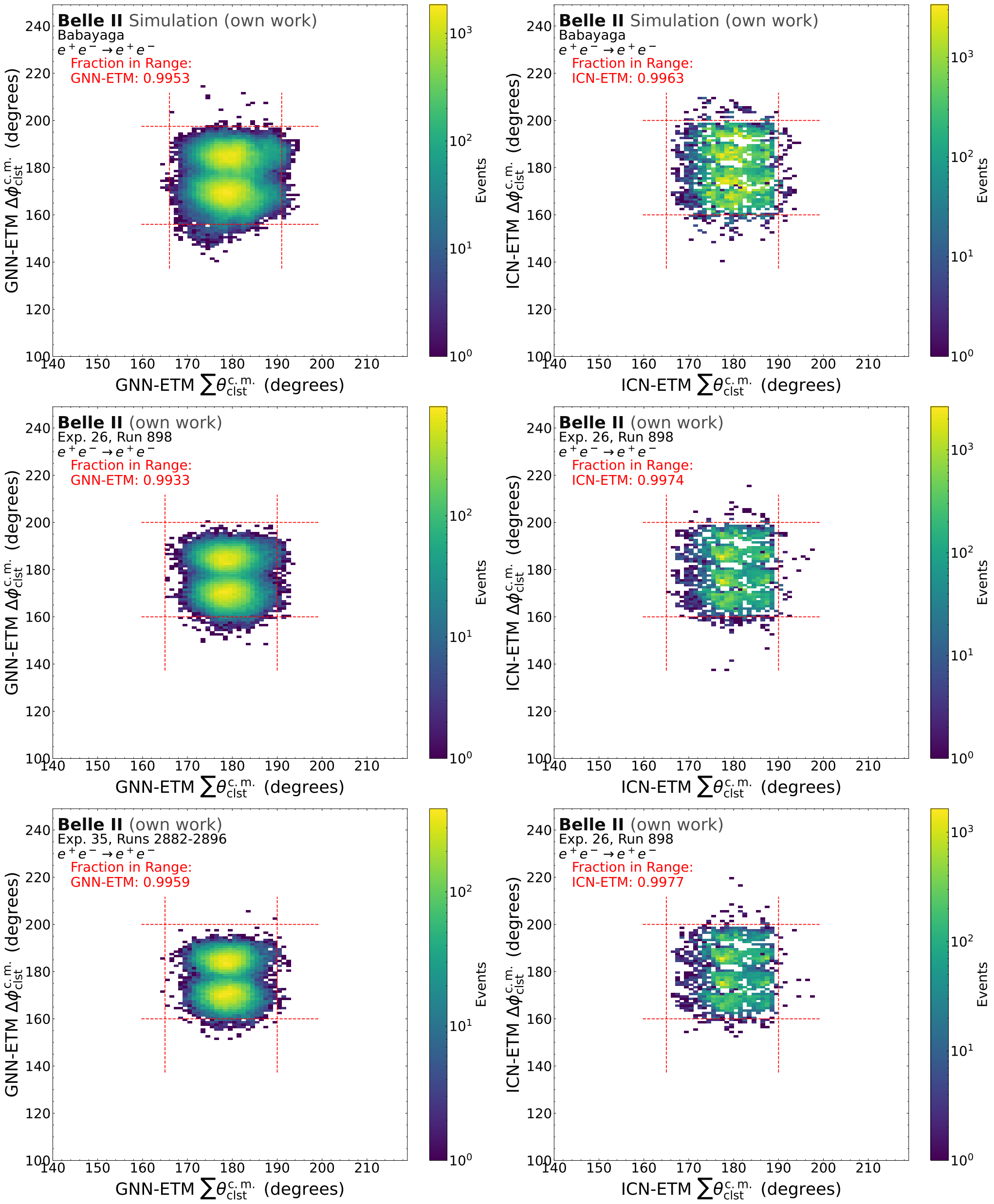


Figure 8.14: Comparison of the values of the sum of the CM θ angles and the difference of the CM ϕ angles between the two signal clusters. The upper row shows the distribution for MC, the center for Exp. 26 data and the lower for Exp. 35 data. The left column shows the GNN-ETM distribution, while the right column shows the ICN-ETM distribution. The red dotted lines mark the requirements for the bha3d trigger bit. The fraction of events in these lines is given in the plot.

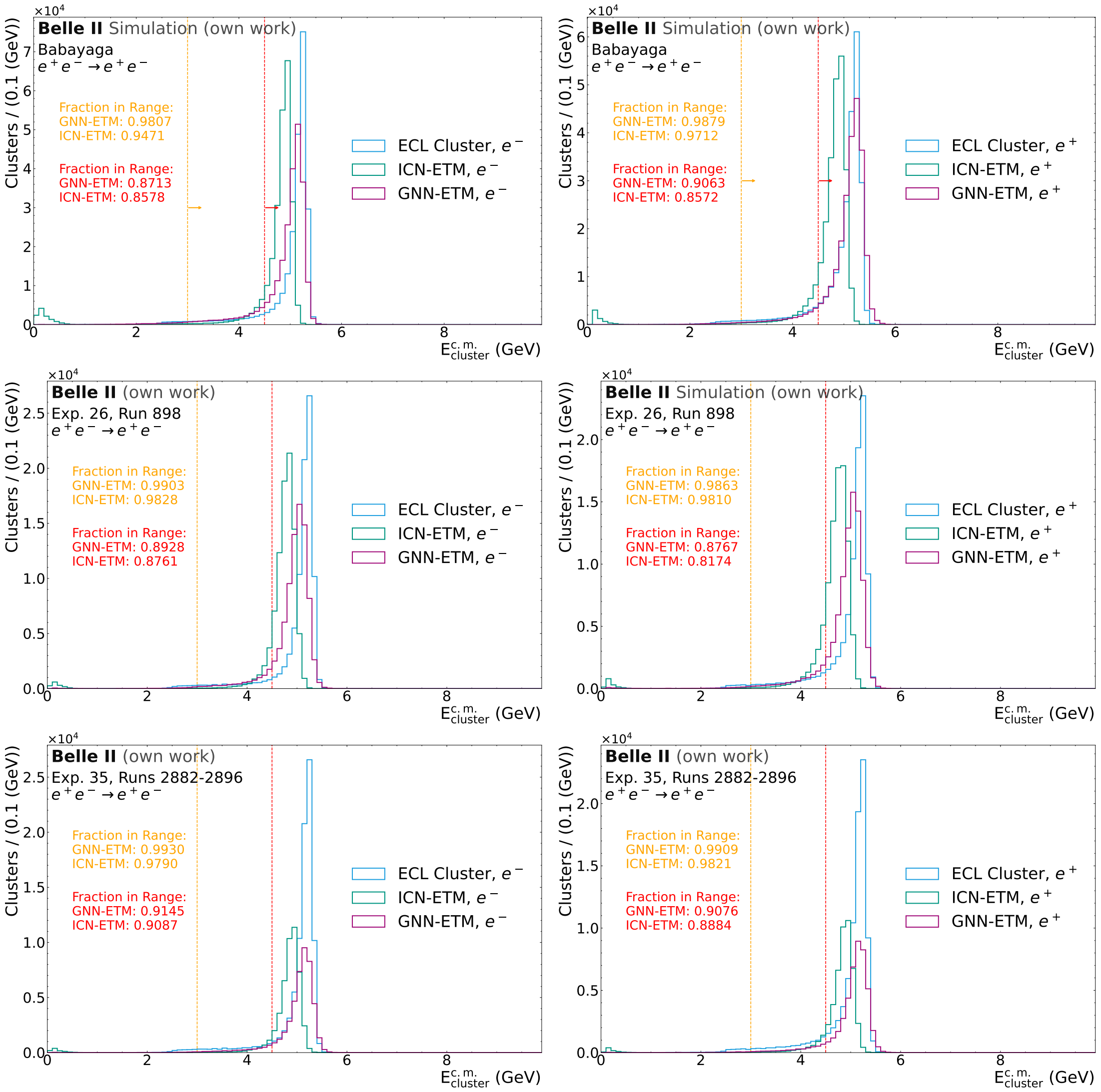


Figure 8.15: CM energy distributions for the energy of the ICN-ETM and GNN-ETM cluster matched to the electron offline ECL cluster (left) and matched to the positron offline ECL cluster (right), with the CM offline ECL cluster energy distribution shown additionally. The top row shows the distributions for MC, the center row for Exp. 26 data and the lower row for Exp. 35 data. The orange cut and fraction show the values for the 3 GeV energy cut, while the red cut and fraction show the values for the 4.5 GeV cut. No bias correction is applied.

8.3 Analysis of the GNN-ETM performance for $e^+e^- \to \mu^+\mu^-(\gamma)$ events

With a rather high cross-section of 1.14 nb, $e^+e^- \to \mu^+\mu^-(\gamma)$ is a frequent process at Belle II and has, due to two high-energy tracks with low-energy offline ECL clusters matched to it, a very clean signature. Besides the generic CDC-based track triggers, which select single tracks or two tracks with a high opening angle, the ECL subdetector trigger provides a significant part of the trigger efficiency for $e^+e^- \to \mu^+\mu^-(\gamma)$ with the trigger bit *ecl_mumu* (see Table 3.2) for tracks in the endcap regions. Improving the performance of this trigger bit, which relies on the finding efficiency of low-energetic clusters and their position reconstruction, is a challenge to be tackled with the GNN-ETM.

The procedure for $e^+e^- \to \mu^+\mu^-(\gamma)$ is the same as for $e^+e^- \to e^+e^-(\gamma)$. First, the selection procedure for this process is presented and the general data/MC agreement for high-level reconstructed objects is shown to highlight effects that may differ between data and MC. For the MC samples, 1 million events are simulated with the `KKMC` generator, as explained in Section 4.4. Then, the GNN-ETM performance on cluster level is evaluated on these selected events for both data and MC events. The finding efficiency and purity as well as the resolutions of the predicted clusters is presented, in comparison to the performance of the ICN-ETM. At last, the trigger bit efficiency for $e^+e^- \to \mu^+\mu^-(\gamma)$ is shown, and improvements to the trigger bits for GNN-ETM are discussed.

8.3.1 Selection Procedure and General Data/MC Agreement

The selection strategy is based on selecting two tracks with high momenta, which have low-energy offline ECL clusters matched to those tracks. The cut on low-energy offline ECL clusters rejects $e^+e^- \to e^+e^-(\gamma)$ events. Additionally, an invariant mass cut is applied to reject radiative dimuon events, $e^+e^- \to \mu^+\mu^-\mu^+\mu^-$ events and similar processes.

The selections applied are

- the transversal momentum p_t of each track has to be $p_t > 0.2\,\mathrm{GeV/c}$,
- the closest distance $|d_0|$ of the track in the $r-\phi$ plane to (0,0,0) and the transversal distance $|z_0|$ between the track and (0,0,0) have to be $|d_0| < 2\,\mathrm{cm}$ and $|z_0| < 4\,\mathrm{cm}$,
- the momentum of the track in the CM system has to be $p_{\mathrm{c.m.}} > 1.5\,\mathrm{GeV/c}$,
- the CM energy of all particle candidates $E_{\mathrm{c.m.}}$ has to be $2.5 < E_{\mathrm{c.m.}} < 0.55 \cdot 10.58\,\mathrm{GeV}$,
- the track has to have a matched offline ECL cluster with $0 < E_{\mathrm{cluster}} < 0.5\,\mathrm{GeV}$.

The two muon candidates with the highest momenta are combined into a $\Upsilon(4S)$ with the following selections applied:

- the sum of the two polar angles in the CM system $\theta_{\text{c.m.}}$ has to be $|\theta_{\text{c.m.}}(\mu 1) + \theta_{\text{c.m.}}(\mu 2)$ - $180^\circ| < 5^\circ$,

- the absolute difference of the two azimuthal angles in the CM system $\phi_{\text{c.m.}}$ has to be $||\phi_{\text{c.m.}}(\mu 1)$ - $\phi_{\text{c.m.}}(\mu 2)|$ - $180^\circ| < 5^\circ$,

- the two tracks have to have opposite charges $q(\mu 2) \times q(\mu 1) =$ -1,

- the reconstructed energy of the dimuon system has to be $M_{\mu\mu} > 0.85 \cdot 10.58\,\text{GeV/c}^2$.

For a very tight selection, an angle selection for the offline ECL clusters matched to the tracks to be only in the center barrel region is applied. This is done via the selection

- the offline ECL cluster angle θ_{cluster} has to be between $37.8^\circ < \theta_{\text{cluster}} < 120.5^\circ$.

As for the trigger offline ECL clusters in the endcaps are of particular interest, as CDC-based track triggers have rather low efficiency in this region, this requirement will be loosened for the evaluation of the GNN-ETM performance.

HLT and L1 trigger selections

For the $e^+e^- \to \mu^+\mu^-(\gamma)$ process, the HLT and L1 trigger trigger lines are separated, as opposed to the trigger requirements for the $e^+e^- \to e^+e^-(\gamma)$ process.

The HLT line used here for selection is the *selectmumu* line. The selections for this HLT line are:

- two tracks with two matched offline ECL clusters,

- at least one matched offline ECL cluster has to have $E_{\text{cluster}} < 1\,\text{GeV}$,

- The momentum of one track has to be $|p| > 2.5\,\text{GeV/c}$, while the momentum of the other track has to be $|p| > 3\,\text{GeV/c}$,

- the absolute difference of the two azimuthal angles in the CM system $\phi_{\text{c.m.}}$ has to be $||\phi_{\text{c.m.}}(\mu 1)$ - $\phi_{\text{c.m.}}(\mu 2)|$ - $180^\circ| < 15^\circ$.

This HLT line has a prescale of 1.

For the L1 trigger requirement, there is a dedicated ECL trigger bit *ecl_ mumu*, as mentioned before. While this trigger bit should have the best efficiency for $e^+e^- \to \mu^+\mu^-(\gamma)$ and was used in [99], due to a faulty configuration with the database for the trigger configuration, the *ecl_ mumu* bit has a significant efficiency loss for data taken in 2022. This efficiency loss was not uniform but disfavoured events, where the muons deposited more energy in the ECL, leading to a non-consistent shape between data and MC. In Fig. 8.16, a shape comparison of the full selection including applied correction factors for a trigger selection with *ecl_ mumu* in comparison to a trigger selection with *stt* is shown.

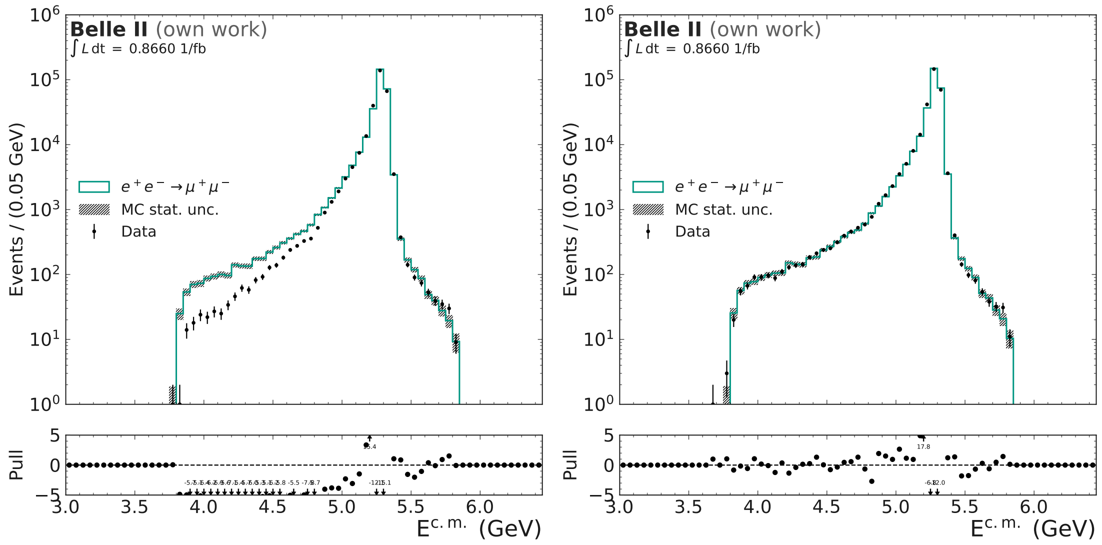


Figure 8.16: Center-of-Mass energy of both tracks after selection and applying correction factors obtained by the trigger efficiency calculation. For the left plot, the trigger bit *ecl_ mumu* was applied while for the right plot, the trigger bit *stt* was used. Lower-energy tracks travel longer within the ECL and deposit more energy, which is disfavoured by the wrongly configured *ecl_ mumu* bit.

Lower-energy tracks do not pass straight through the ECL, but curl due to the magnetic field. Due to the longer path, the energy deposition is higher, which results in lower efficiency due to the misconfiguration of the *ecl_ mumu* bit. Therefore, the CDC based *stt* bit is used, which requires at least one track found by the neural network track trigger.

HLT and L1 trigger corrections

The trigger corrections are calculated analogously to the trigger corrections for $e^+e^- \to e^+e^-(\gamma)$. The full selections are applied besides the trigger requirement. The trigger efficiency is then calculated by

$$\epsilon_{\mathrm{TRG}} = \frac{N(\mathrm{Ref\&Test})}{N(\mathrm{Ref})}. \tag{8.7}$$

For the HLT test trigger line *selectmumu*, the reference bit *ge1_ tight_ track* is used. For this reference bit, at least one track with a maximum distance in z-direction from (0,0,0) of 2 cm is required.

For the L1 trigger trigger efficiency, two efficiencies are calculated. To show the shape difference for the *ecl_ mumu* bit in Fig. 8.16 and to later use this in a comparison for GNN-ETM, a correction factor for the data/MC comparison is calculated with the reference bit being the *bfyo* bit. The *bfyo* bit requires two tracks with an opening angle over 90°. For the *stt* trigger bit, the reference bit is the *lml10* trigger bit. This requires the *ecl_ lml_ 10* input bit, which is defined in Table 3.3 and is solely based on ECL trigger information.

The trigger efficiencies and the corresponding correction factors can be found in Table 8.8. For the *ecl_mumu* bit, the improvement in efficiency between Exp. 26 and Exp. 35 can clearly be seen. The drop in efficiency for the HLT trigger bit *selectmumu* between Exp. 26 and Exp. 35 is due to a loss in offline tracking efficiency, both due to the high beam backgrounds and due to reconstruction software misconfiguration. In general, due to the high dependency on the tracking performance for this selection, the overall efficiency loss is up to 5.5 % for the *stt* for Exp. 26 and for the HLT line *selectmumu* for Exp. 35. Effects due to this trigger efficiency loss have to be checked for in the analysis of the GNN-ETM performance.

After application of all selections and correction factors, the corresponding data/MC comparisons can be seen in Fig. 8.17 for the CM energy of both tracks and Fig. 8.18 for the cluster opening angle. For the CM energy, the two particles are sorted according to their $\theta_{\mathrm{c.m.}}$, with particle 1 having the larger angle. In general, the differential distributions between the signal MC data and the collision data both from Exp. 26 and Exp. 35 agree very well. Especially the offline ECL cluster opening angle, which also for $e^+e^- \to \mu^+\mu^-(\gamma)$ is the deciding factor for the *ecl_mumu* L1 trigger trigger decision, show the same features. Therefore, the evaluation of GNN-ETM can be done on all three datasets. Differences in the performance of the GNN-ETM can then be attributed to either different predictions for collision data and MC data or to different background levels in the different datasets, for example, but not to differences in the signal process signatures.

Especially the CM energy in Fig. 8.17 shows discrepancies for the shape of the distribution for data taken in Exp. 35. This could be due to tracking difficulties, as the CDC is especially sensitive to high beam background conditions and had significant problems towards the end of the data-taking period of Exp. 35.

Table 8.8: Trigger efficiencies for $e^+e^- \to \mu^+\mu^-(\gamma)$ for the HLT and L1 trigger lines used in selection for MC, data taken in Exp. 26 and data taken in Exp. 35. The correction factors are calculated as $\epsilon_{\mathrm{data}}/\epsilon_{\mathrm{MC}}$.

Test Line	ϵ_{MC}	$\epsilon_{\mathrm{Exp.26}}$	$\epsilon_{\mathrm{Exp.35}}$	$\mathrm{f}_{\mathrm{Exp.26}}$	$\mathrm{f}_{\mathrm{Exp.35}}$
selectmumu	$0.995^{+0.0001}_{-0.0001}$	$0.997^{+0.0021}_{-0.0053}$	$0.945^{+0.0103}_{-0.0125}$	1.001	0.950
stt	$0.995^{+0.0001}_{-0.0001}$	$0.945^{+0.0119}_{-0.0149}$	$0.989^{+0.0042}_{-0.0068}$	0.950	0.994
ecl_mumu	$0.983^{+0.0002}_{-0.0002}$	$0.916^{+0.0175}_{-0.0215}$	$0.980^{+0.0063}_{-0.0090}$	0.932	0.997

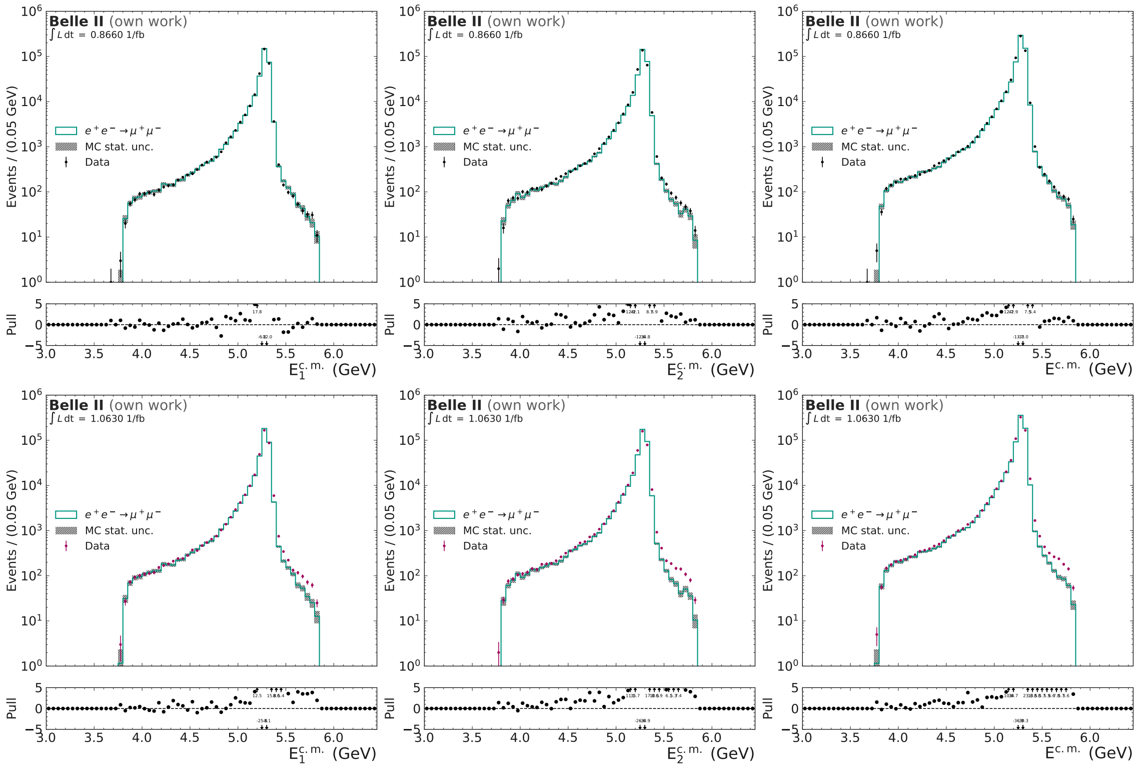


Figure 8.17: Center-of-Mass energy for the two reconstructed particles and their combined distribution for the $e^+e^- \to \mu^+\mu^-(\gamma)$ process. The particles are sorted by their center-of-mass θ angle, with particle 1 having the larger angle. MC events are scaled with the corresponding data luminosity. The data in the upper row is taken from Exp. 26, Runs 803, 848 and 898, while the data in the lower row is taken from Exp. 35, Runs 2817 - 2905.

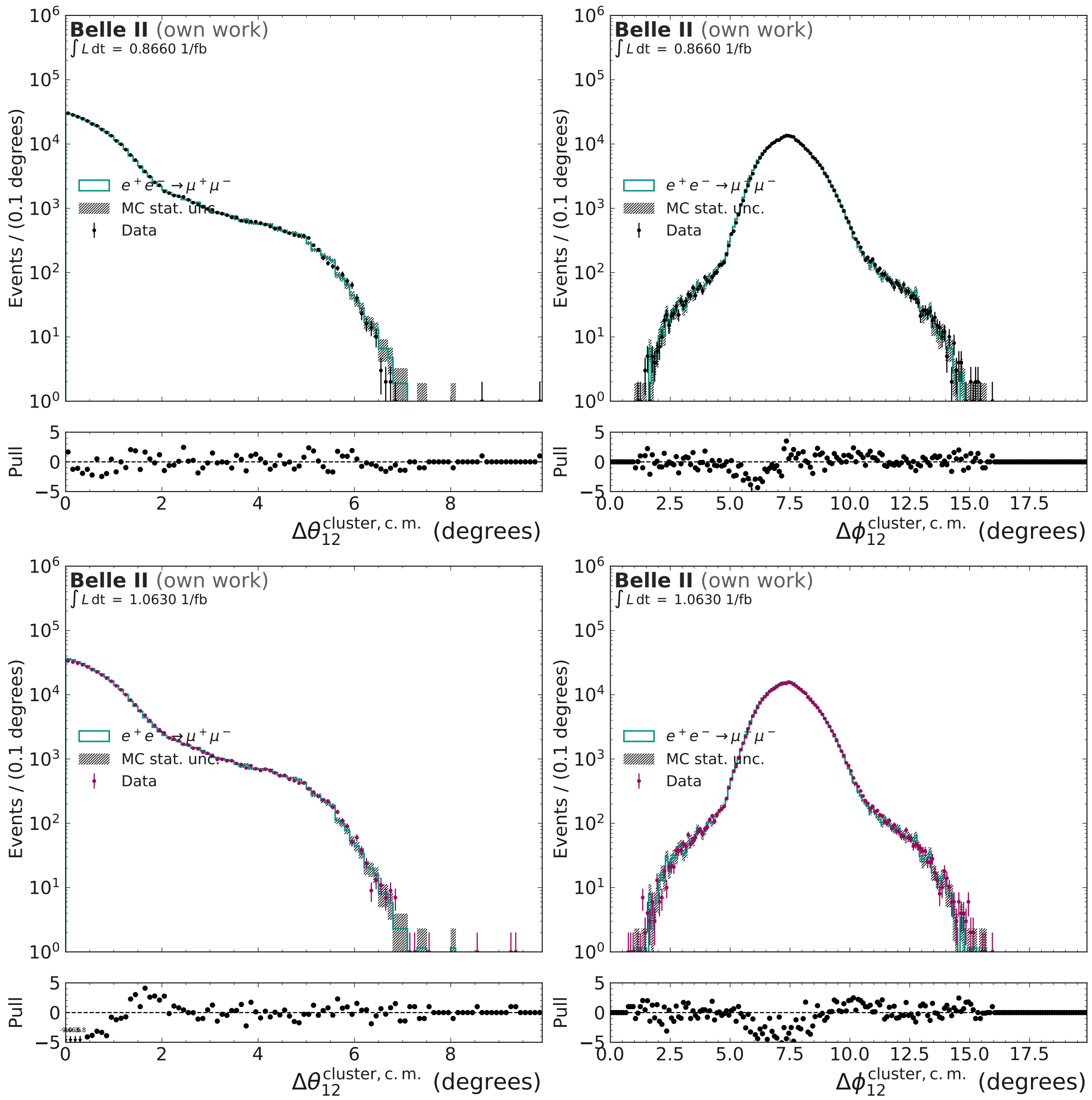


Figure 8.18: ϕ (left) and θ (right) cluster opening angle for the two reconstructed particles for the $e^+e^- \to \mu^+\mu^-(\gamma)$ process. The cluster opening angle is the relevant measure to trigger on $e^+e^- \to \mu^+\mu^-(\gamma)$ events in the ECL trigger due to missing track information. MC events are scaled with the corresponding data luminosity. The data in the upper row is taken from Exp. 26, Runs 803, 848 and 898, while the data in the lower row is taken from Exp. 35, Runs 2817 - 2905.

8.3.2 Trigger Data Sample Selection

Similarly as for the $e^+e^- \to e^+e^-(\gamma)$ sample, two raw data samples are used for the study of the GNN-ETM performance on data. The dataset for Exp. 35 remains the same as for $e^+e^- \to e^+e^-(\gamma)$, described in Section 8.2.2.

For the sample for Exp. 26, the reprocessed *mumu_tight_or_highm* skim is used. This skim selects dimuon events with ideally no radiative photon component.

For this skim, two different selections are used and the events, that are kept by either selection, pass the skim. The selections are:

1. Selection for two tracks with matched clusters:
    - the number of tracks in the event has to be 2,
    - both tracks are matched to an offline ECL cluster with $E_{\text{cluster}} < 0.5\,\text{GeV}$,
    - the total energy of all offline ECL clusters is $\sum E_{\text{cluster}} < 2\,\text{GeV}$,
    - the opening angle between the two tracks is $||\phi(\mu1) - \phi(\mu2)| - 180^\circ| < 10^\circ$ and $||\theta(\mu1) + \theta(\mu2)| - 180^\circ| < 10^\circ$,
    - the radius of both tracks has to be $R > 0.5\,\text{m}$.
2. Selection allowing for one track to not be matched to an offline ECL cluster:
    - At least one track is matched to an offline ECL cluster,
    - if an offline ECL cluster is matched, its energy has to be $E_{\text{cluster}} < 1\,\text{GeV}$,
    - the tracks have opposite charges $q(\mu1) \times q(\mu2) = \text{-}1$,
    - the reconstructed mass of both tracks has to be $8\,\text{GeV/c}^2 < M_{\text{track1,track2}} < 12\,\text{GeV/c}^2$,
    - the opening angle between the two tracks is $||\phi(\mu1) - \phi(\mu2)| - 180^\circ| < 10^\circ$ and $||\theta(\mu1) + \theta(\mu2)| - 180^\circ| < 10^\circ$.

For both the Exp. 35 and the Exp. 26 dataset, the selections from Section 8.3.1 are applied, besides the angle selection

- $37.8^\circ < \theta_{\text{cluster}} < 120.5^\circ$,

to allow for tracks reaching the forward and backwards endcap.

The number of events before and after the selection can be seen in Table 8.9.

Table 8.9: Number of events for the full datasets used for the trigger study and after selection. No angular selection on the offline ECL clusters is made here.

	MC sample	Exp. 26	Exp. 35
Before Selection	10^6	92588	3366688
After Selection	492433	64089	36365

8.3.3 Trigger Cluster Efficiencies and Resolutions

Improving the cluster finding efficiency and the energy and position reconstruction on trigger level can improve the overall event reconstruction efficiency and aid in designing more efficient trigger decision bits. The performance of the GNN-ETM is therefore first evaluated on cluster level, checking the cluster finding efficiency, purity, and resolutions described in Section 5.2. This performance is always compared to the ICN-ETM performance.

To test this, the offline ECL clusters, which are reconstructed as the offline ECL clusters belonging to the muon candidates by the selection in Section 8.3.1, are used as targets for both trigger algorithms. Only offline ECL clusters are used that pass the timing selection explained in Section 8.1 and additionally have a high enough energy deposition to be visible on trigger level due to the TC threshold cut of 100 MeV. An offline ECL cluster is visible on trigger level, if it passes the requirements explained in Section 4.2.2. The ICN-ETM and GNN-ETM clusters of those events are then matched to the offline ECL clusters using the cluster matching requirements of Section 5.1. If a GNN-ETM or ICN-ETM cluster is matched to an offline ECL cluster, this offline ECL cluster is considered found. The relative difference between the reconstructed energy of the offline ECL cluster and the matched trigger cluster, and the absolute difference in position values is then evaluated in the resolutions. This is done for the MC data, Exp. 26 data and Exp. 35 data separately.

Cluster Finding Efficiencies

In Fig. 8.19, the finding efficiency for both GNN-ETM and ICN-ETM is shown. The top row shows the efficiencies for the simulated MC dataset, the center row for the Exp. 26 dataset, and the lower row for the Exp. 35 dataset. The GNN-ETM outperforms the ICN-ETM in the lower energy bins for all three datasets. For energies above 200 MeV, both algorithms have a cluster finding efficiency of nearly 100 %. The performance is for both algorithms very consistent over all three datasets and has no visible difference for the μ^- and the μ^+.

In Fig. 8.20, the efficiency for the forward and backward endcap is shown additionally. Reconstruction in those detector regions is in general more difficult due to the irregular geometry and the higher background levels. The finding efficiency drops for GNN-ETM in the energy bins below 200 MeV for the Exp. 26 and Exp. 35 datasets. Lower-energetic clusters are more affected by the presence of beam background, which is more abundant in data. This can explain the drop in efficiency. Nevertheless, the efficiency of GNN-ETM still exceeds the efficiency of ICN-ETM in those energy bins.

Resolutions

For the GNN-ETM and ICN-ETM clusters, which are matched to an offline ECL cluster, the resolutions η for the reconstructed energy and the x, y, and z positions can be determined, similar as in Section 8.2.3.

In this section, the resolutions for selected detector regions and offline ECL cluster energies are shown. In Section D.2.1, the remaining resolutions for ICN-ETM and GNN-ETM clusters matched to the μ^- are shown, while in Section D.2.1 the same is shown for the μ^+.

For the energy resolution, Fig. 8.21 shows the resolution in the forward and backward endcap for offline ECL clusters with a reconstructed energy between 0.16 and 0.2 GeV. The resolution is in general wider for the Exp. 35 dataset, which is due to the very high beam background level in these runs. The GNN-ETM energy resolution is better centered around 0 in comparison to the ICN-ETM energy distribution, while the width is comparable.

All distributions have pronounced tails towards lower values, which shows an underestimation of the offline ECL cluster energy for the trigger clusters. This tail is especially visible in the MC dataset, due to the very narrow peak around 0. This can be explained by the position of the offline ECL cluster within the TC. Muons, as described earlier, interact as minimum-ionizing particles in the ECL and do not get stopped by the ECL. The muon energy deposition is therefore usually contained in the crystal(s), which the muon has crossed. In Fig. 8.22, the distance in θ over the distance in ϕ between the ICN-ETM cluster and the offline ECL cluster, to which it is matched, is shown. The left plot shows this distance for all offline ECL clusters in the barrel region. As the ICN-ETM always reconstructs the position as the center of the highest-energetic TCs, all entries in this plot are within the boundaries of one TC, which spans a 10° region in both angle directions. The crystal center positions can also be seen within that TC by the high amount of entries in 2.5° steps, as the offline reconstruction algorithm tends to reconstruct the position of the muon offline ECL clusters in the center of the corresponding crystal. The right plot only shows this distance for ICN-ETM clusters, which have an energy resolution below -0.3. The majority of these entries are positioned at the edges of the TC. The muon in these events does not cross a single crystal, but two where only one of those crystals is within a TC that has energy above the 100 MeV threshold. The other energy deposition of the muon is therefore not visible on trigger level and the trigger algorithms underestimate the total energy of the offline ECL cluster.

The general position resolution for offline ECL clusters in the forward endcap can be seen in Fig. 8.23 in the left column for the x position and in the right column for the y position for a reconstructed energy of the offline ECL clusters between 0.1 and 0.2 GeV.

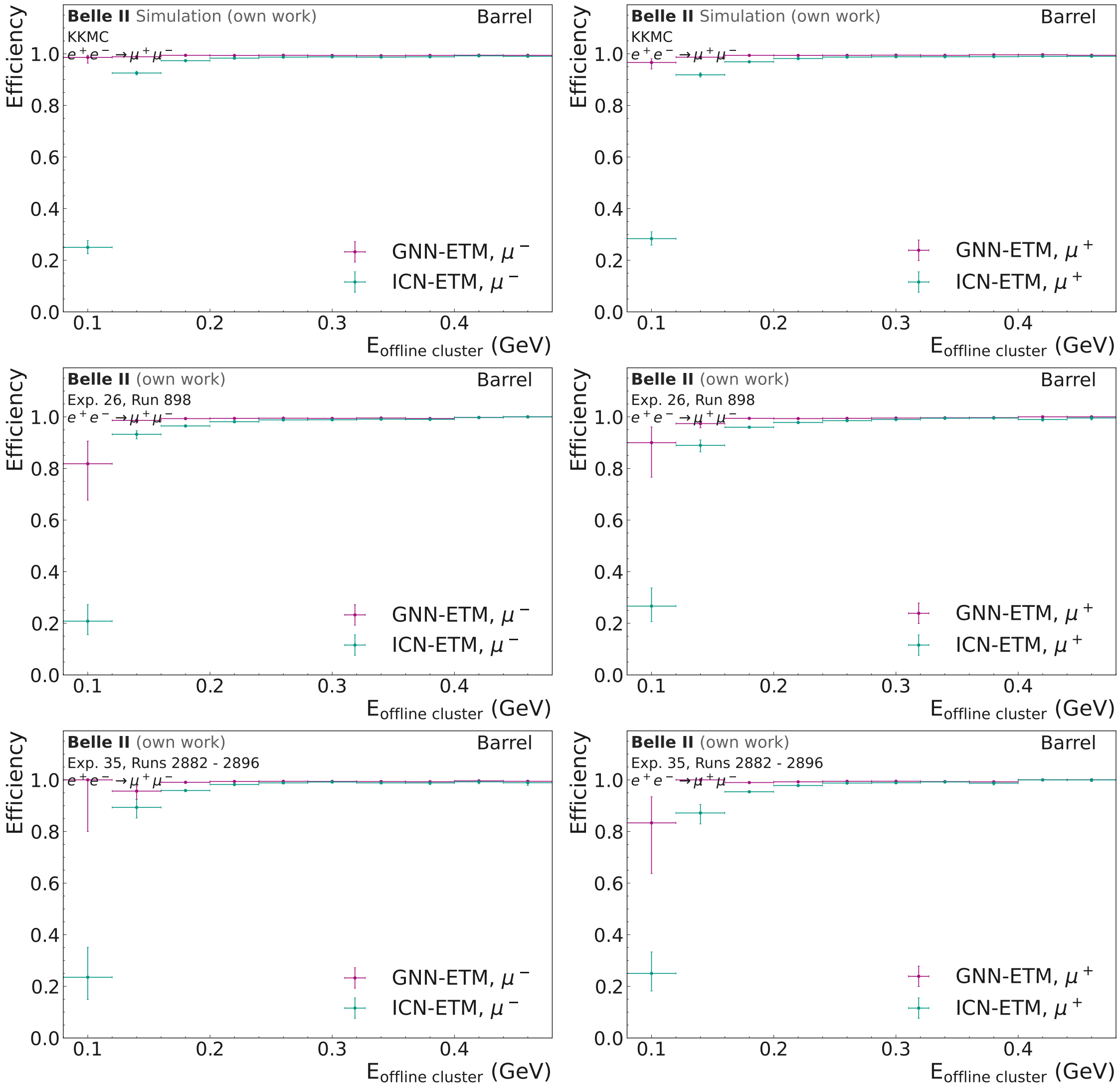


Figure 8.19: Finding efficiency for offline ECL clusters matched to the μ^- (left) and to the μ^+ (right) for the GNN-ETM and the ICN-ETM algorithm dependent on the offline ECL cluster reconstructed energy. The top row shows the efficiency on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The efficiency is shown for offline ECL cluster in the barrel region.

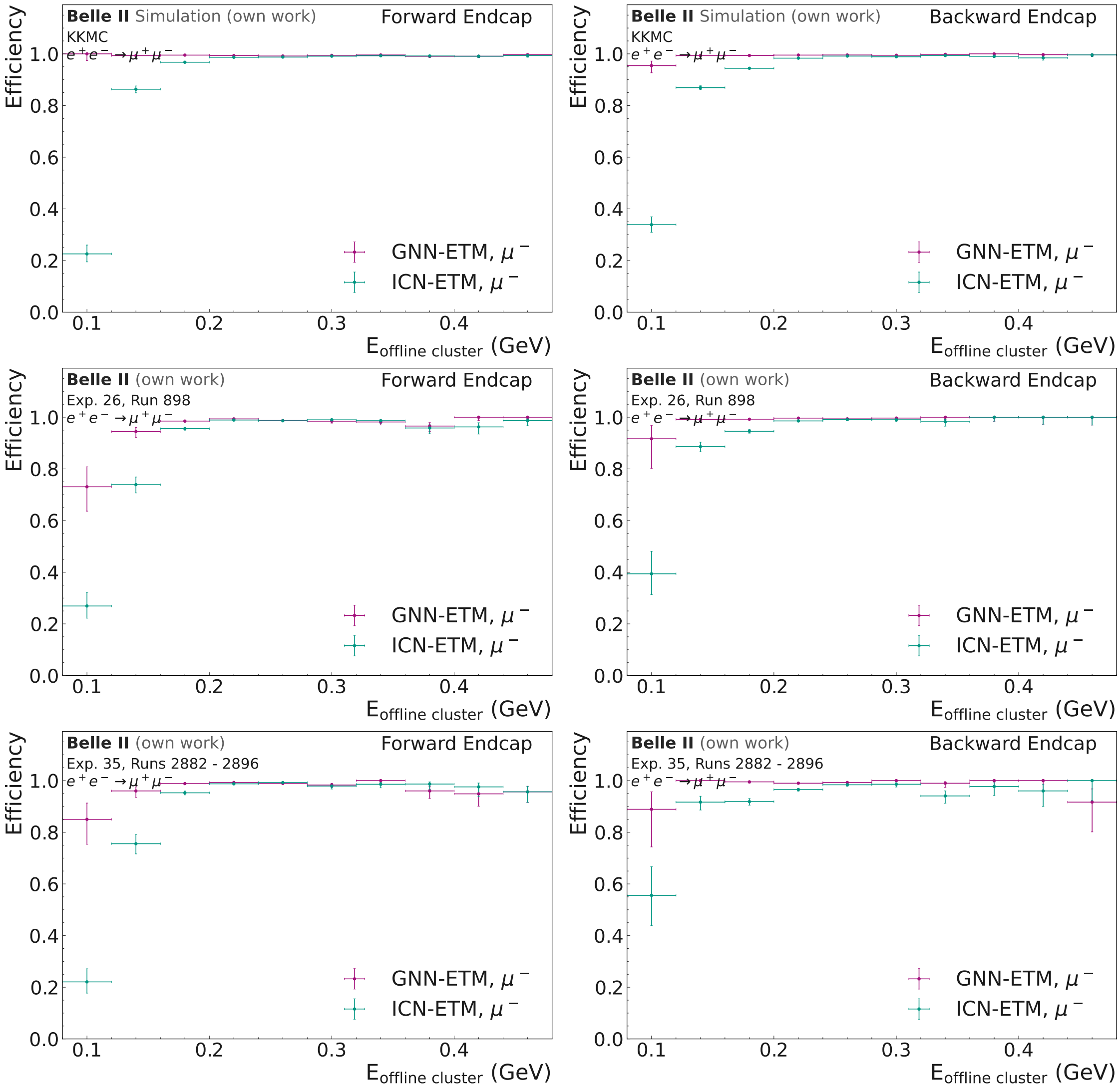


Figure 8.20: Finding efficiency for offline ECL clusters matched to the μ^- for the GNN-ETM and the ICN-ETM algorithm dependent on the offline ECL cluster reconstructed energy for offline ECL clusters in the forward endcap (left) and in the backward endcap (right). The top row shows the efficiency on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset.

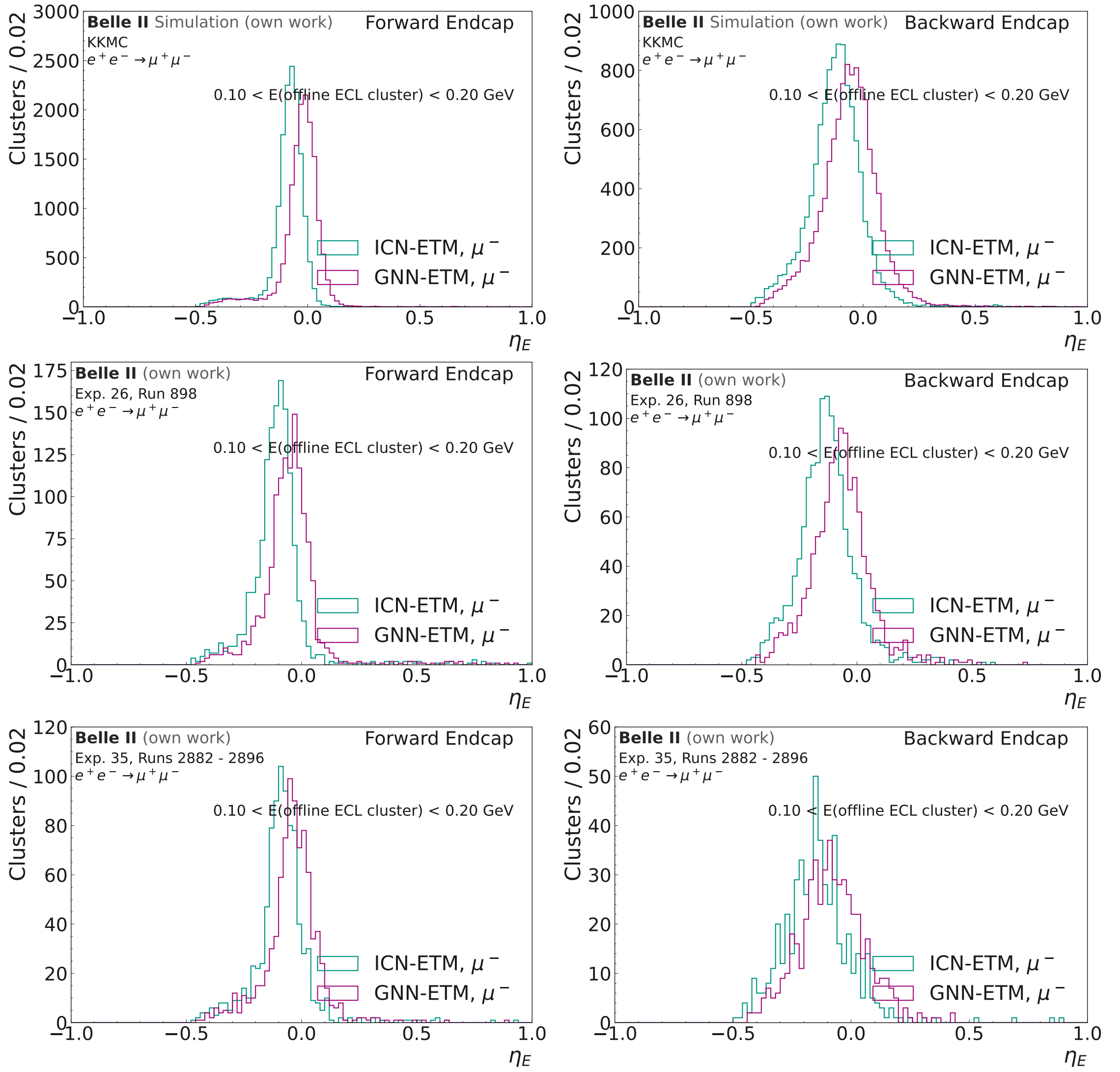


Figure 8.21: Energy resolution for offline ECL clusters matched to the μ^- for the GNN-ETM and the ICN-ETM algorithm for offline ECL clusters in the forward endcap (left) and the backward endcap (right) for $0.1 < E$(offline ECL cluster) $< 0.2\,\text{GeV}$. The top row shows the energy resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset.

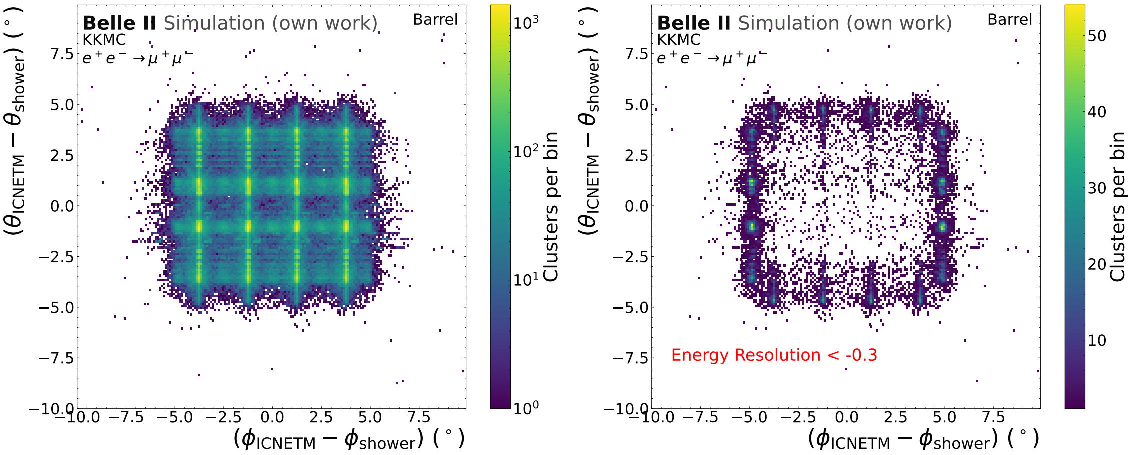


Figure 8.22: The difference in θ positions over the difference in ϕ positions for matched ICN-ETM to offline ECL clusters for the MC dataset for offline ECL clusters in the barrel region. The left plot shows this distance difference for all matched clusters, while the right plot only for those with an energy resolution < -0.3.

As muons only hit a single crystal, there is no additional information available to the GNN-ETM in terms of position reconstruction besides the TC position. The position resolution is as wide as the ICN-ETM position resolution, which is to be expected. The x position has a shift towards lower x values, which is a known problem of this trained network and can be fixed with further trainings.

To summarize, the finding efficiency for offline ECL clusters for two energy bins, 0.1 - 0.2 GeV and 0.3 - 0.4 GeV, is shown in Table 8.11. The GNN-ETM algorithm can improve the cluster finding efficiency in the lower-energetic bin by up to 7 % for the backward endcap in the MC dataset and by 7 and 8 % for the forward endcap for the Exp. 26 and Exp. 35 datasets, respectively. The efficiency for the higher-energetic bins is for both algorithms at around 1. The energy resolution width and the resolution bias for the same energy bins are shown in Table 8.10. The resolution width is comparable between ICN-ETM and GNN-ETM, with GNN-ETM having a slightly broader resolution in general. However, the bias of the distribution is drastically reduced by a factor of 2-3 for all datasets by the GNN-ETM in comparison to the ICN-ETM. While this bias could be corrected in operation for the ICN-ETM, this adds an extra layer of complexity.

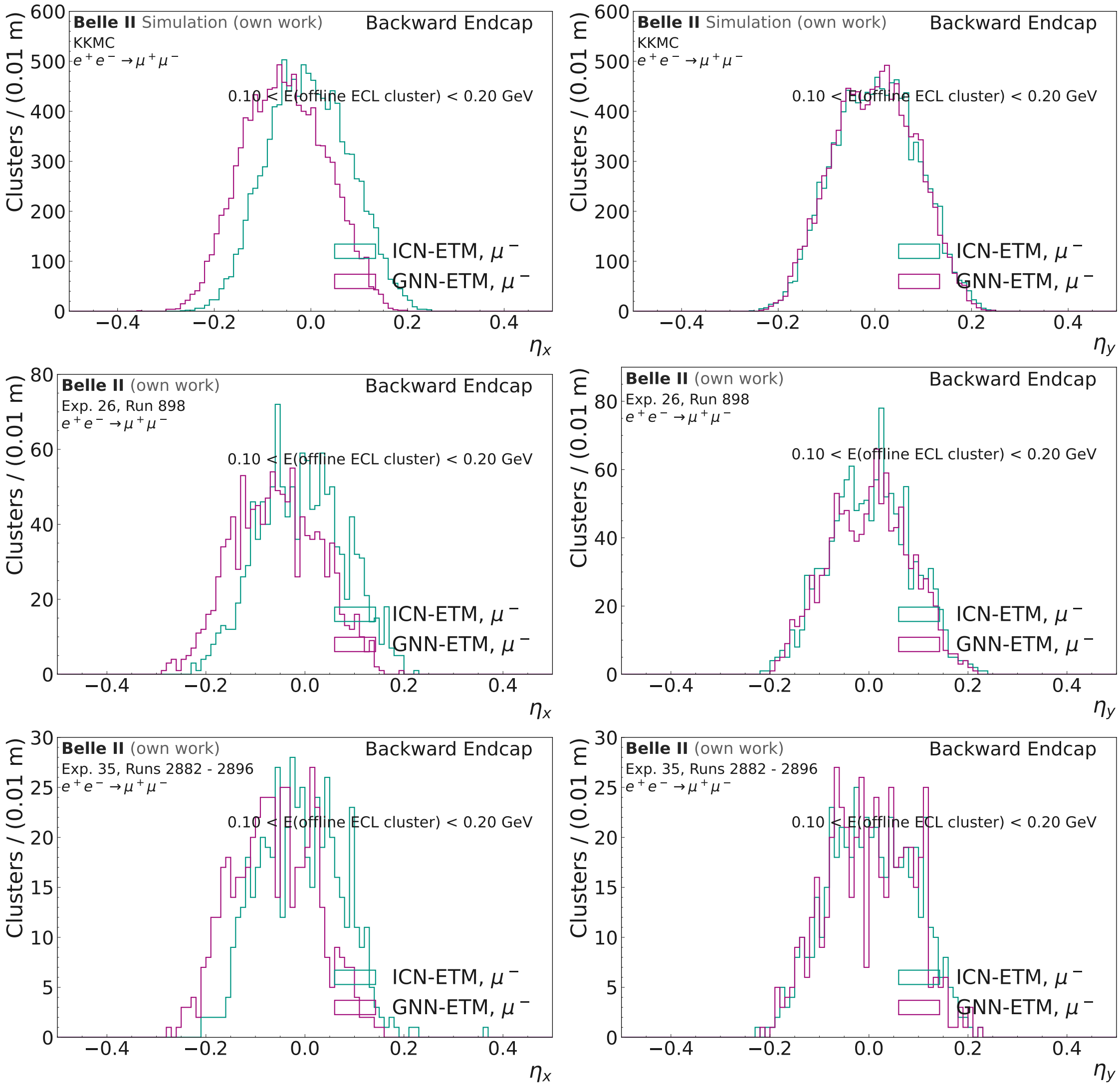


Figure 8.23: x position resolution (left) and y position resolution (right) for offline ECL clusters matched to the μ^- for the GNN-ETM and the ICN-ETM algorithm for offline ECL clusters in the backward endcap with $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$. The top row shows the energy resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset.

Table 8.10: Energy resolution width r_{E} after bias correction and mean m_{E} of the Gaussian fit for the bias correction for the GNN-ETM and ICN-ETM clusters for offline ECL clusters matched to the μ^- for the three datasets. The efficiency is shown for offline ECL clusters with a reconstructed energy between 0.1 and 0.2 GeV and for offline ECL clusters with a reconstructed energy between 0.3 and 0.4 GeV.

Dataset	Region	Algorithm	$\eta_{\mathrm{E},0.1-0.2}$	$m_{\mathrm{E},0.1-0.2}$	$\eta_{\mathrm{E},0.3-0.4}$	$m_{\mathrm{E},0.3-0.4}$
MC	BWD	GNN-ETM	0.0993±0.0006	-0.079	0.0911±0.0019	-0.060
		ICN-ETM	0.0966±0.0006	-0.138	0.0848±0.0018	-0.122
	Barrel	GNN-ETM	0.0655±0.0001	-0.042	0.0595±0.0005	-0.026
		ICN-ETM	0.0611±0.0001	-0.096	0.0549±0.0004	-0.088
	FWD	GNN-ETM	0.0511±0.0003	-0.018	0.0477±0.0008	-0.017
		ICN-ETM	0.0472±0.0003	-0.079	0.0450±0.0009	-0.083
Exp. 26	BWD	GNN-ETM	0.0948±0.0017	-0.088	0.0904±0.0048	-0.063
		ICN-ETM	0.0929±0.0017	-0.148	0.0814±0.0041	-0.128
	Barrel	GNN-ETM	0.0690±0.0004	-0.060	0.0633±0.0015	-0.042
		ICN-ETM	0.0645±0.0003	-0.112	0.0584±0.0013	-0.105
	FWD	GNN-ETM	0.0656±0.0011	-0.050	0.0675±0.0043	-0.048
		ICN-ETM	0.0616±0.0009	-0.108	0.0657±0.0040	-0.114
Exp. 35	BWD	GNN-ETM	0.1253±0.0033	-0.090	0.1041±0.0091	-0.072
		ICN-ETM	0.1246±0.0030	-0.152	0.1021±0.0081	-0.127
	Barrel	GNN-ETM	0.0768±0.0006	-0.051	0.0656±0.0017	-0.035
		ICN-ETM	0.0721±0.0005	-0.103	0.0613±0.0021	-0.095
	FWD	GNN-ETM	0.0699±0.0014	-0.028	0.0842±0.0057	-0.033
		ICN-ETM	0.0647±0.0013	-0.085	0.0779±0.0050	-0.096

Table 8.11: Cluster finding efficiency ϵ_{clst} for the GNN-ETM and ICN-ETM for offline ECL clusters matched to the μ^- for the three datasets. The efficiency is shown for offline ECL clusters with a reconstructed energy between 0.1 and 0.2 GeV and for offline ECL cluster with a reconstructed energy between 0.3 and 0.4 GeV.

Dataset	**Region**	**Algorithm**	ϵ_{clst}, E $\in$ [0.1, 0.2] GeV	ϵ_{clst}, E $\in$ [0.3, 0.4] GeV
MC	BWD	GNN-ETM	$0.9935^{+0.0007}_{-0.0008}$	$0.9966^{+0.0010}_{-0.0013}$
		ICN-ETM	$0.9253^{+0.0024}_{-0.0025}$	$0.9902^{+0.0017}_{-0.0021}$
	Barrel	GNN-ETM	$0.9941^{+0.0003}_{-0.0003}$	$0.9932^{+0.0005}_{-0.0006}$
		ICN-ETM	$0.9717^{+0.0006}_{-0.0006}$	$0.9878^{+0.0007}_{-0.0007}$
	FWD	GNN-ETM	$0.9951^{+0.0005}_{-0.0006}$	$0.9948^{+0.0011}_{-0.0013}$
		ICN-ETM	$0.9601^{+0.0015}_{-0.0016}$	$0.9923^{+0.0013}_{-0.0016}$
Exp. 26	BWD	GNN-ETM	$0.9923^{+0.0022}_{-0.0030}$	$0.9968^{+0.0020}_{-0.0052}$
		ICN-ETM	$0.9250^{+0.0068}_{-0.0075}$	$0.9873^{+0.0049}_{-0.0080}$
	Barrel	GNN-ETM	$0.9927^{+0.0009}_{-0.0010}$	$0.9945^{+0.0012}_{-0.0015}$
		ICN-ETM	$0.9618^{+0.0020}_{-0.0021}$	$0.9904^{+0.0016}_{-0.0019}$
	FWD	GNN-ETM	$0.9792^{+0.0036}_{-0.0043}$	$0.9794^{+0.0055}_{-0.0075}$
		ICN-ETM	$0.9125^{+0.0070}_{-0.0075}$	$0.9819^{+0.0051}_{-0.0070}$
Exp. 35	BWD	GNN-ETM	$0.9942^{+0.0025}_{-0.0044}$	$0.9957^{+0.0027}_{-0.0070}$
		ICN-ETM	$0.9129^{+0.0111}_{-0.0125}$	$0.9660^{+0.0100}_{-0.0139}$
	Barrel	GNN-ETM	$0.9900^{+0.0015}_{-0.0017}$	$0.9942^{+0.0015}_{-0.0020}$
		ICN-ETM	$0.9553^{+0.0032}_{-0.0034}$	$0.9901^{+0.0020}_{-0.0025}$
	FWD	GNN-ETM	$0.9841^{+0.0037}_{-0.0048}$	$0.9871^{+0.0050}_{-0.0081}$
		ICN-ETM	$0.9039^{+0.0091}_{-0.0099}$	$0.9872^{+0.0050}_{-0.0081}$

Trigger Bit Efficiency

While high cluster efficiencies and good energy and position resolution are important for the performance of a trigger algorithm, the final performance metric is the efficiency and the rate of the trigger bits. These determine whether an event is kept or not and should, for these events which contain a clean signature for the $e^+e^- \to \mu^+\mu^-(\gamma)$ process, have a trigger efficiency of close to 1. The trigger bits currently in use for the ICN-ETM-based trigger are shown in Table 3.2 and Table 3.3. The trigger bit *eclmumu* is a dedicated trigger bit for the $e^+e^- \to \mu^+\mu^-(\gamma)$ signature and should have an efficiency close to 1.

To test the trigger efficiency of the GNN-ETM, the events remaining after the selection in Section 8.3.1, besides the angle and stt trigger requirement, are used. For the ICN-ETM, the trigger bit decision is taken from either TSIM in the case of MC events or from the online trigger decision recorded in raw data. The FTDL value of the bits are taken, to remove prescale effects.

For the GNN-ETM, the trigger bits are calculated as given by the definition in Table 3.2 and Table 3.3. When a trigger bit needs position or energy values in the CM frame, the value is boosted to the CM frame using a Lorentz transformation in `ROOT`. For this Lorentz transformation, the four-vector is built from the cluster energy, θ and ϕ values. This is a slight difference to the ICN-ETM calculation, where, for implementation simplification, the conversion values are fixed values for each TC, assuming a photon cluster with 1 GeV.

In Fig. 8.24, the trigger bit efficiency for both ICN-ETM and GNN-ETM for the three datasets is shown. The trigger bits for the GNN-ETM are calculated using only clusters with a signal classifier output value above 0.7. The most relevant trigger bit for the $e^+e^- \to \mu^+\mu^-(\gamma)$ process is the *eclmumu* bit, which looks for two back-to-back clusters below 2 GeV. The trigger bit has a high efficiency for MC events, with ICN-ETM outperforming the ICN-ETM by 2 %. The threshold of 0.7 for the signal classifier was set to achieve 95 % signal efficiency. This means that for all three datasets, the classifier is more likely to classify GNN-ETM clusters matched to muon offline ECL clusters as signal, because the expected efficiency loss for *eclmumu* would be 10 %. For the ICN-ETM, the drop in efficiency for Exp. 26 data can be explained by the wrong configuration of the *eclmumu* bit explained in Section 8.3.1. In Exp. 35, this configuration was fixed and the trigger bit efficiency increased again.

For the GNN-ETM, a significant drop in efficiency for both Exp. 26 and Exp. 35 can be seen. In Fig. 8.25, the same trigger bit efficiency is shown, only now the GNN-ETM trigger bits are calculated on all clusters, without taking the signal classifier into account. This increases the trigger efficiency for GNN-ETM in Exp. 26 and Exp. 35. However, the signal classifier output for the GNN-ETM clusters matched to the muon offline ECL clusters, seen in Fig. 8.26, shows that the vast majority of muon clusters are classified as signal. For the MC dataset, 2.7 % of events have at least one of the GNN-ETM clusters matched to the muon offline ECL cluster classified as background, for the Exp. 26 dataset,

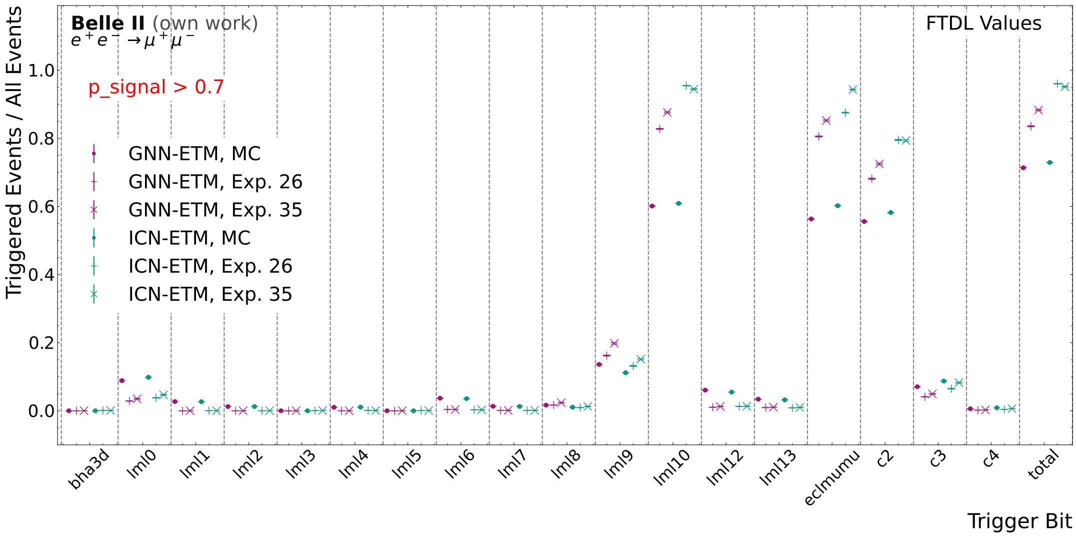


Figure 8.24: Trigger bit efficiency for $e^+e^- \to \mu^+\mu^-(\gamma)$ events for the GNN-ETM and ICN-ETM. The exclusive ECL trigger bits are shown, with the total trigger bit being the inclusive sum of all other shown trigger bits. The trigger bit efficiency is shown for the three datasets. For the GNN-ETM trigger bit determination, only clusters with a signal/background classifier output above 0.7 are used.

6.4 % are classified as background and for Exp. 35 8.9 % of the GNN-ETM clusters. This can already explain part of the effect. Nevertheless, the trigger bit efficiency for Exp. 35 for the *eclmumu* bit is larger than for Exp. 26, while the number of clusters classified as signal is larger in Exp. 26, which has to be due to an additional effect.

In Fig. 8.27, the distributions for the two angle requirements for the *eclmumu* bit are shown for the GNN-ETM and ICN-ETM clusters. The plots show the difference in ϕ positions between the μ^- and μ^+ over the sum of the CM θ value for the two particles. The red lines mark the cuts of the *eclmumu* trigger bit. For the Exp. 26 and Exp. 35 dataset, the distribution for the GNN-ETM is shifted to lower values for the ϕ difference. This leads to a loss in efficiency for the *eclmumu* bit, which a change in trigger bit requirements could mitigate. The overall distribution for the GNN-ETM is wider for all three datasets than the ICN-ETM, which is due to the floating position prediction of the GNN-ETM. The ICN-ETM is restricted to the TC positions.

As a summary, the calculated trigger bit efficiency using the current bit definition decreases the GNN-ETM efficiency in comparison to the ICN-ETM efficiency. This is not due to incorrect predictions of the GNN-ETM but to the less restricted position reconstruction. However, especially for muons, the position reconstruction could be improved in future trainings to enforce clearer boundaries. Additionally, a check on the signal classifier for clusters matched to the muon offline ECL clusters shows a good performance in the classification of these clusters as signal in MC, but a decrease in classification performance for data. Further trainings with an addition of muons to the training samples or different background conditions might mitigate this effect.

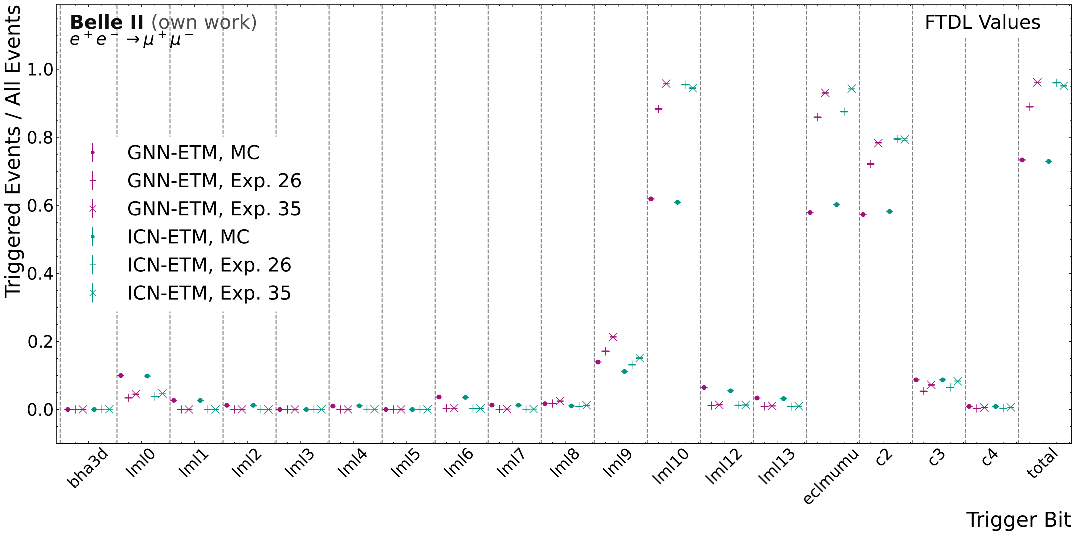


Figure 8.25: Trigger bit efficiency for $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$ events for the GNN-ETM and ICN-ETM. The exclusive ECL trigger bits are shown, with the total trigger bit being the inclusive sum of all other shown trigger bits. The trigger bit efficiency is shown for the three datasets.

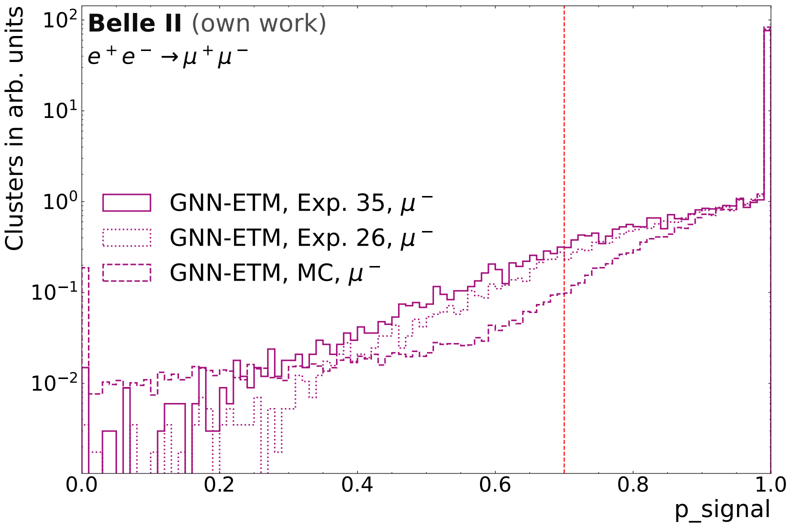


Figure 8.26: Signal classifier output for the GNN-ETM cluster matched to the μ^- offline ECL cluster for the three different datasets. The distributions are normed for shape comparison. The red dotted line marks the threshold above which clusters are marked as signal.

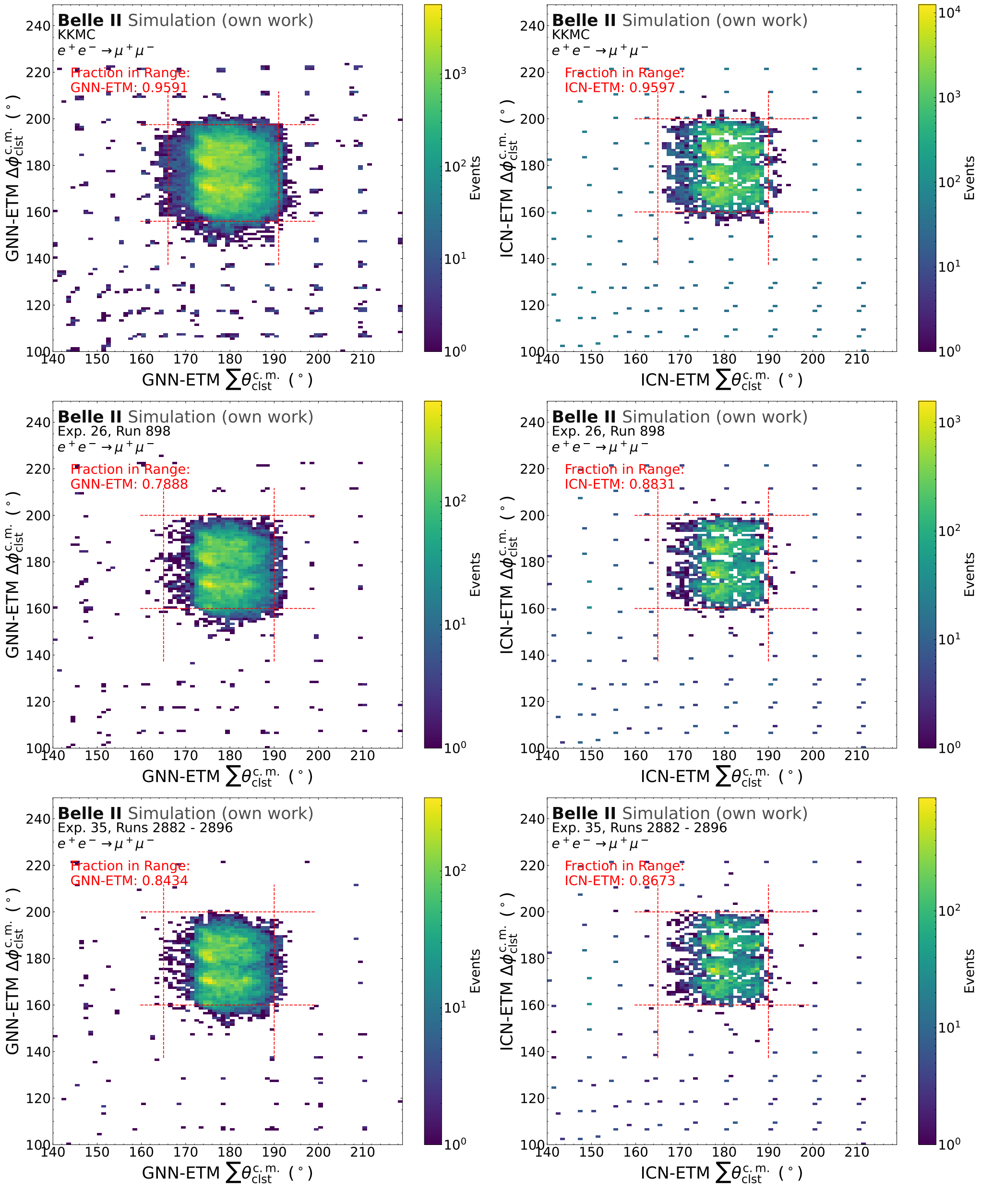


Figure 8.27: Comparison of the values of the sum of the CM θ angles and the difference of the CM ϕ angles between the two signal clusters. The upper row shows the distribution for MC, the center for Exp. 26 data and the lower for Exp. 35 data. The left column shows the GNN-ETM distribution, while the right column shows the ICN-ETM distribution. The red dotted lines mark the requirements for the *eclmumu* trigger bit. The fraction of events in these lines is given in the plot.

8.4 Signal Classifier Performance on Data

Adding a signal/background classification on trigger level can reduce the overall trigger rate and improve event selection by removing background trigger clusters within one event. This can especially reduce the rate of trigger bits such as c3 or c4, which test if at least three or at least four clusters, respectively, are reconstructed by the trigger algorithm. Removing potential background clusters from this count can help improve the purity of these trigger bits.

Testing the signal/classifier performance on data is difficult due to the missing signal label information. A clean sample of events or clusters, in which the probability of coming from beam background is high, has to be selected to evaluate the performance and verify the results of the classifier on MC.

Two options for testing the classifier are possible: The first option is to use events triggered not by physics trigger bits, but by so-called random trigger bits. For these events, I assume that the majority of the energy depositions in the ECL are due to beam background. In the current L1 trigger setup, three random bits are available:

1. random: trigger signal is given by a random number generator using an independent local clock
2. poisson: random trigger using a Poisson distribution
3. bg: trigger signal used for background overlays (see Section 4.1.3), trigger 5 µs after bha3d has triggered.

For the random and the bg trigger bit, the injection veto is additionally applied. All three trigger bits are prescaled, so that the resulting rate is below 1 kHz. Events, that are triggered by those bits, are very likely events where no collision has taken place during the bunch crossing, as the interaction rate per bunch crossing is very small. Therefore all energy depositions in these events should originate from beam background and the GNN-ETM should classify all predicted clusters as background. Only a small amount of this data is available due to the prescales, but the overall trigger efficiency and the distribution of the signal/background classifier can be tested.

The second method is the complete reconstruction of an event containing a clean physics process such as $e^+e^- \to \mu^+\mu^-(\gamma)$ or $e^+e^- \to e^+e^-(\gamma)$, removing all offline ECL cluster coming from the signal process, and evaluating the classifier on the offline ECL cluster in the remaining event. Even if two tracks apass the very clean selection presented in Section 8.3.1, additional offline ECL cluster with a reconstructed energy up to a few 100 MeV do not alter the tracks' kinematics significantly, even with the excellent momentum resolution of Belle II [58]. This necessitates a larger amount of events and an even tighter selection than presented in Section 8.2.1 and Section 8.3.1 to factor out this effect. This will be tested in the future when more data taken with the GNN-ETM module is available.

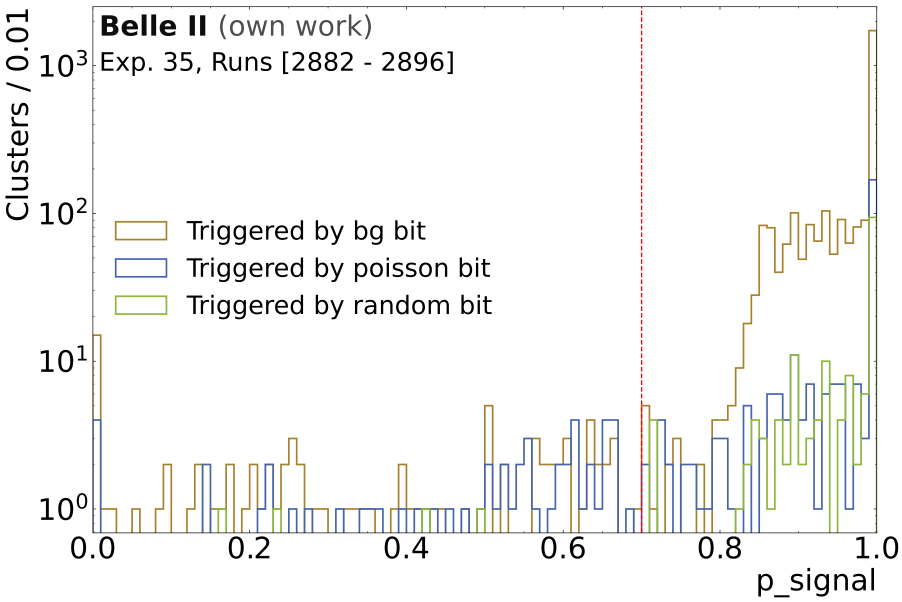


Figure 8.28: The output of the signal/background classifier value p_signal for the clusters in the events triggered by the random, poisson, and bg trigger bits. The red dotted line marks the cut value for the signal threshold.

To test events triggered by random triggers, the data from Exp. 35, Runs 2882, 2890, 2895, 2896 is used. Data from Exp. 26 cannot be used due to the previously applied skims on the raw data (see Section 8.2.2), which already select events that very likely include a signal process coming from a collision. The available data corresponds to 3366688 events in total. First, the events are selected in which at least one of the random, poisson, or bg trigger bit has a value of 1. As a second step, only those events are selected where the trigger decision window for the ICN-ETM consists of data windows 3 and 4, as explained in Section 8.1. This guarantees a fair comparison for the GNN-ETM and ICN-ETM performance, as for the GNN-ETM only the GNN-ETM window 2 is selected for the trigger evaluation. This also ensures that these events have significant energy depositions in the ECL.

After selection, 1702 events remain, of which 76 are triggered by the random trigger, 220 by the poisson trigger, and 1406 by the bg trigger. Additionally, only events, where the injection veto has not been active, are selected. Most ECL trigger bits have an additional requirement of the injection veto being inactive, which makes a comparison between GNN-ETM and ICN-ETM performance for events with an active injection veto. This further reduces the events triggered by the poisson trigger to 159.

In Fig. 8.28, the GNN-ETM signal classifier output for all clusters in the remaining events is shown, color-coded for the events coming from the random, poisson, and bg trigger bits. The cut value for a signal cluster is chosen as 0.7, which is determined by the cut value for the technical sample in Section 7.2.1. Further optimization of this cut value might improve the performance.

Table 8.12: Trigger efficiency for events triggered by the random, poisson and bg trigger bit. The efficiency is calculated for the ICN-ETM, the GNN-ETM and GNN-ETM with only clusters with a signal classifier value above 0.7. The denominator is the number of events triggered by the respective random, poisson and bg trigger bit and with the ICN-ETM trigger decision window consisting of data windows 3 and 4.

Algorithm	ϵ_{total}(random==1)	ϵ_{total}(poisson==1)	ϵ_{total}(bg==1)
ICN-ETM	$0.0021^{+0.0011}_{-0.0007}$	$0.0016^{+0.0006}_{-0.0004}$	$0.0022^{+0.0002}_{-0.0002}$
GNN-ETM	$0.0014^{+0.0009}_{-0.0006}$	$0.0017^{+0.0006}_{-0.0004}$	$0.0018^{+0.0002}_{-0.0002}$
GNN-ETM (p_signal > 0.7)	$0.0014^{+0.0009}_{-0.0006}$	$0.0017^{+0.0006}_{-0.0004}$	$0.0013^{+0.0002}_{-0.0002}$

For all events, the ECL trigger bits are calculated for the GNN-ETM and taken from data for the ICN-ETM. In general, for both GNN-ETM and ICN-ETM, the trigger efficiency for the ECL trigger bits for these events is very low, as expected. The total trigger efficiency, which is the inclusive rate of all ECL trigger bits with their FTDL value is shown in Table 8.12. The efficiency is shown for the ICN-ETM trigger bits, the trigger bits calculated with all GNN-ETM clusters, and the trigger bits calculated only using clusters with a signal classifier output above 0.7. Especially for the events triggered by the bg trigger, the classifier cut can reduce the overall efficiency. This is a desired result as these events with a very high chance do not contain processes originating from actual collisions. The trigger efficiency for the different trigger bits can be found in Section D.3.

Nevertheless, as the bunch-crossing rate is at 250 MHz, a very small trigger efficiency can still lead to a very high trigger rate. For the calculation of the trigger rate, the denominator used are all events which are triggered by the respective random, poisson, or bg trigger bit, regardless of the ICN-ETM decision window. The numerator uses the aforementioned events where the ICN-ETM trigger window is the correct one for comparison to the GNN-ETM, as the assumption is a correct working ECL trigger setup for these events. The rate is calculated as described in Section 5.2.

In Fig. 8.29, Fig. 8.30, and Fig. 8.31, the trigger bit rates for the separate ECL trigger bits and the total trigger rate for the combined bits is shown for events triggered by the random, poisson, and bg trigger bit, respectively. For all three cases, the c2 trigger bit has the highest contribution. For the current Belle II data-taking, this has a prescale of 0, which takes it out of the active trigger decision due to its very high rate. However, especially in case of events triggered by the bg trigger, the signal classifier can reduce this trigger rate by more than a factor of 2. For the poisson and random triggered events, the signal classifier has no impact on the total GNN-ETM trigger rate. However, the total amount of events which have any active ECL trigger bit for the GNN-ETM are 4 and 20, for the random and the poisson trigger, respectively. For further conclusions, the amount of data has to be increased.

In Fig. 8.30, another effect can be seen which is given by the low number of tested

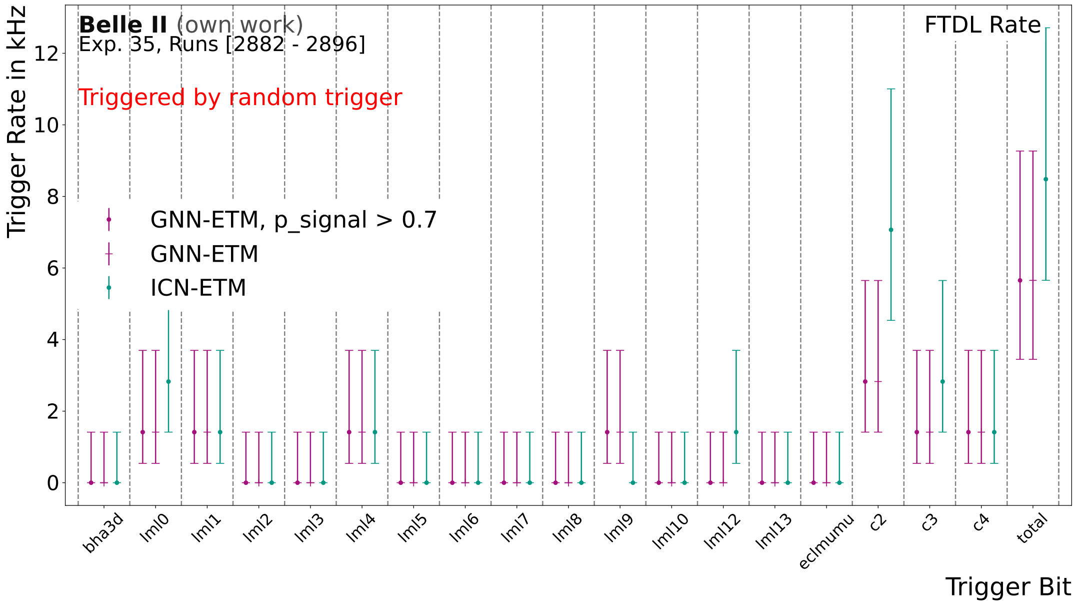


Figure 8.29: Trigger bit rate for the ECL trigger bits for events triggered by the random trigger bit. The trigger bit rate for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit efficiency for the GNN-ETM, both when using all predicted clusters and using only predicted clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the random trigger has fired.

events. The trigger rate for the GNN-ETM with the signal classifier cut is lower for the lml10 and the c2 bit than the ICN-ETM rate and only higher for the lml13 bit. In each case this is a single event difference. Nevertheless, the overall trigger rate of the GNN-ETM is higher than the ICN-ETM. This is due to the fact that the same event is triggered by multiple bits and enters the total value only once. The event for lml13 however, is only triggered by the GNN-ETM for this single bit and therefore has a high impact. With a higher number of test events, this effect can be reduced.

In summary, the signal/background classifier can improve the overall trigger rate on pure background events by removing clusters from the trigger bit calculation when they are identified as background. The reduction in rate by the GNN-ETM for events triggered by the bg trigger bit is more than 50 % respective to the rate of the ICN-ETM. These events are, as explained in Section 4.1.3, used as official beam background overlays for the Belle II collaboration. Clusters in these events should therefore be classified as background and the high reduction in rate due to the classifier performance is promising. Further tests with more data can improve this result and help develop the signal classifier further.

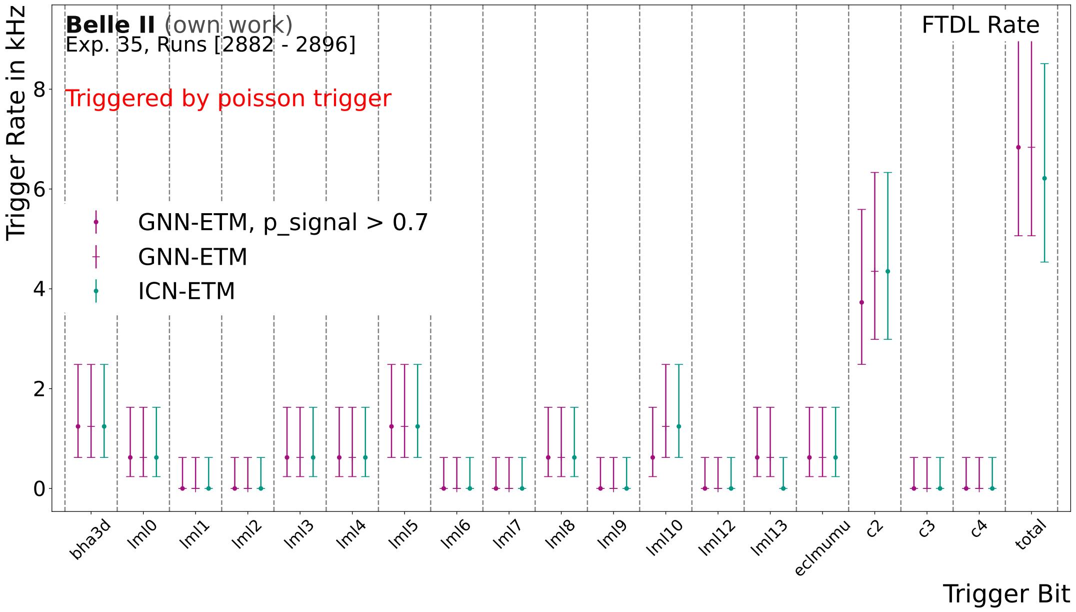


Figure 8.30: Trigger bit rate for the ECL trigger bits for events triggered by the poisson trigger bit. The trigger bit rate for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit rate for the GNN-ETM, both when using all predicted clusters and using only predicted clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the poisson trigger has fired.

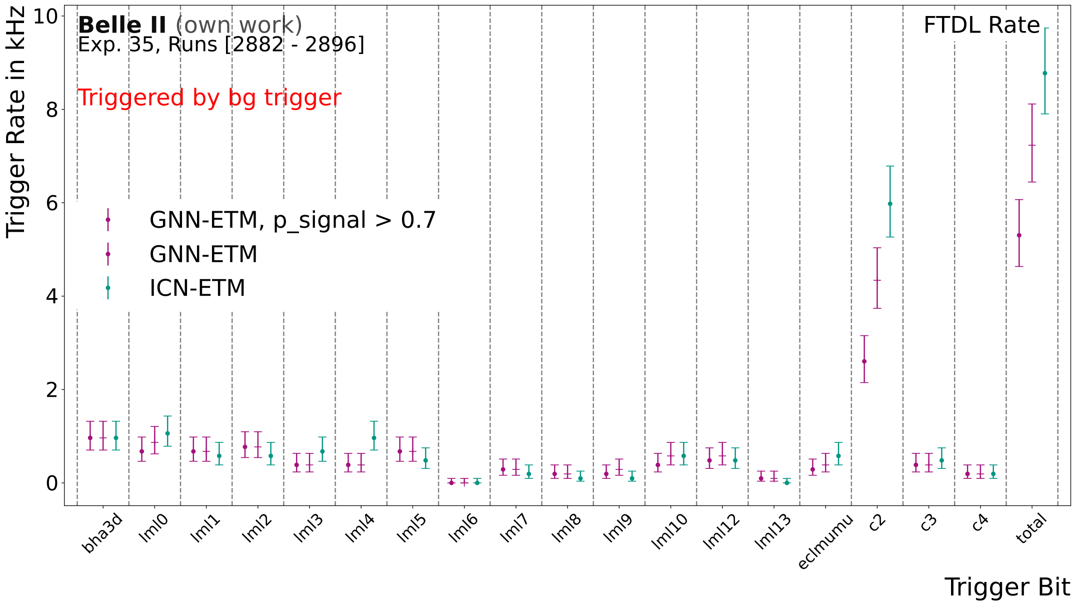


Figure 8.31: Trigger bit rate for the ECL trigger bits for events triggered by the bg trigger bit. The trigger bit rate for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit rate for the GNN-ETM, both when using all predicted clusters and using only predicted clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the bg trigger has fired.

Chapter 9

Conclusion

In this thesis, I present the design, implementation, and evaluation of a GNN-based trigger algorithm for the Belle II ECL L1 trigger. This demonstrates the feasibility and performance of an ML algorithm designed for and deployed in a real-time environment with hard latency and throughput constraints.

The GNN-ETM algorithm is designed with the restrictions of an implementation on the UT4 FPGA, while still achieving a good performance. The challenges and techniques for a hardware-software codesign are shown and the effects of reduced precision, pruning of parameters, and simplifications of activation functions are evaluated. The algorithm is implemented on the UT4 board and successfully deployed in the Belle II L1 trigger in a parasitic implementation, which it was operated in during collision data taking in December 2024. I show the successful pre- and postprocessing of the GNN-ETM module and the storing and unpacking of the GNN-ETM data in the raw data of Belle II. The performance of the GNN-ETM is tested on technical datasets and two selected physics processes, $e^+e^- \to e^+e^-(\gamma)$ and $e^+e^- \to \mu^+\mu^-(\gamma)$.

The GNN-ETM improves the offline ECL cluster finding efficiency for $e^+e^- \to e^+e^-(\gamma)$ by 10 % and the position resolution by up to 30 %. The trigger efficiency for ECL-based trigger decisions for this process is comparable to that of ICN-ETM and can be optimized further with a redesign of the trigger bits.

For the $e^+e^- \to \mu^+\mu^-(\gamma)$ process, the GNN-ETM matches the performance of the ICN-ETM. Due to the inherent position and energy resolution of offline ECL cluster contained in one TC, an improvement of energy and position resolutions cannot be done without changing the ECL trigger input geometry. A 8 % improvement in efficiency for clusters below 0.2 GeV is achieved by the GNN-ETM over the ICN-ETM. Additionally, the bias of the energy resolution is reduced up to a factor of 3.

A key contribution of this work is the development of a signal/background classifier, introducing a capability not available in the existing ECL L1 trigger system. This classifier is evaluated on both simulated beam background as well as on background samples in collision data. Due to the classifier, the trigger rate for beam background-induced events taken in December 2024 can be reduced by a factor of 2.

In general, the GNN-ETM module is the first GNN-based trigger with dynamic graph building implemented and deployed in a realistic collider environment with high background levels. Full deployment is planned in ongoing cosmics data taking and in the upcoming data taking period of Belle II in the fall of 2025.

The overall latency of approximately 3 µs is about twice the maximum latency allowable for a full integration in the L1 trigger system. The latency has to be reduced to 1.6 µs. To achieve this latency, work is underway to reduce network depth, the size of the network layers, and the bit widths.

Additionally, specific GNN-ETM trigger bits are currently being developed. The different performance of the GNN-ETM in comparison to the ICN-ETM allows the change or tightening of trigger bit requirements, especially with the inclusion of the signal/background classifier.

For future upgrades of Belle II, increasing the granularity of the TCs is expected to improve the possibilities of the trigger designs distinctly. Going from 4×4 TCs, which is the current input to the ECL trigger, to 2×2 or 1×1 can help reject background clusters more efficiently and use shower shape information for the detection and identification of different cluster signatures. I have implemented a possible 2×2 TC design in the `basf2` trigger simulation to test the performance of the GNN-ETM with a different input geometry. This is specifically challenging for the implementation on hardware, as the average number of inputs will increase from currently 10 to 40 for the 2x2 geometry and to 250 for the 1x1 geometry, but opens up new possibilities for algorithmic design and trigger decisions.

In conclusion, I have shown the integration of a first GNN-based trigger algorithm for clustering in the ECL of Belle II, performing within the throughput constraints of the L1 trigger system and improving the ECL trigger efficiency, purity, and position and energy resolution for high energy clusters.

Danksagung

Ohne die Menschen an meiner Seite wäre es mir nicht möglich gewesen, dieses Projekt durchzuziehen und diese Arbeit am Ende fertigzustellen.

Als allererstes möchte ich meinem Referenten und Supervisor Prof. Dr. Torben Ferber danken. Torben ist mit absolutem Herzblut und Hingabe bei jedem Projekt dabei, das er anfängt oder betreut und so involviert in jedem einzelnen Schritt seiner PhDs, wie ich es von keinem Supervisor kenne. Es gibt fast keine Frage, auf die er keine Antwort weiß. Seine Tür stand und steht immer offen und wenn ich das nicht genutzt habe, wurde ich von ihm mit Nachdruck dazu gebracht, mir Rat und Unterstützung zu holen. Mit jemand anderem als Referent*in hätte ich den PhD nach spätestens einem Jahr abgebrochen. Ausserdem wäre sonst niemand auf die Idee gekommen, eine Konferenz mit 1000 Personen voll vegan zu organisieren und es auch noch so unglaublich gut hinzubekommen. Vielen Dank für alle Möglichkeiten, ob Konferenzen, B2GMs, DPGs, FSPs, Talks, wöchentliche Meetings, und jedes Feedback, jede Hilfe während und auch nach meinem PhD.

Ich möchte auch meinem Korreferenten Prof. Dr. Markus Klute danken. Viele Outreach-Projekte, ETP-Veranstaltungen und Diskussionen mit ihm haben mir immer viel Spaß gemacht und ich bin sehr dankbar für alle Möglichkeiten und Diskussionen, die ich mit ihm hatte.

Ohne meine Kolleg*innen am ETP hätte ich diese Arbeit auf gar keinen Fall fertiggestellt. Die Zeit am Institut, die Zusammenarbeit und die Freundschaften, die dadurch entstanden sind, waren und sind für mich unglaublich wertvoll. Zu Beginn möchte ich Marc Neu danken, mit dem ich gemeinsam an diesem Projekt die letzten zwei Jahre meines PhDs gearbeitet habe und ohne den wir niemals auch nur annähernd so weit gekommen wären. An Weihnachten Bugs finden und irgendwie Daten nehmen zu können, zusammen mit Marc und Torben, war ein Erlebnis (aber einmal reicht dann auch). Weiterhin möchte ich auch Dr. Lars Sowa und Lea Reuter danken. Beide haben mit mir Masterarbeit gemacht bei unserem Supervisor James und ich bin sehr froh, dass sie die gesamte Zeit an meiner Seite waren. Lea und ich haben zudem noch thematisch viel Überlapp gehabt und viele Projekte gemeinsam gestaltet, sodass ich jederzeit jemanden hatte, der genau wusste, was die Probleme sind. Besonderen Dank gilt auch Dr. Patrick Ecker, der mir am Anfang seinen kompletten Code bereitgestellt hat, und mich vorallem am Ende viel unterstützt hat, durch Lesen der Arbeit, Feedback zum Talk, und beruhigende Worte a la "Ich hab es auch überlebt".

Weiterhin möchte ich Alexander Heidelbach danken, der mit vielen Organisationsprojekten, Outreachprojekten und auch Trips in verschiedenste Städte ein wichtiger Mensch für mich in dieser Zeit war. Ein weiterer Kollege, dem ich sehr dankbar bin, ist Dr. Giacomo de Pietro, bei welchem mir unklar ist, wie Menschen, die nicht einfach in sein Büro gehen können und ihm jede nur denkbare Frage stellen können, ihren PhD überhaupt schaffen. Zusätzlich hat er mir stets Bilder von Ottern zum Aufmuntern geschickt. Jonas Eppelt und Lennard Damer, die mit mir momentan das Team ECL bilden, möchte ich auch danken für alle hilfreichen Konversationen, technischen Diskussionen, Kaffeepausen und Japantrips. Weiterhin möchte ich Greta Heine, Dr. Slavomira Stefkova, Dr. Moritz Bauer, Dr. Felix Metzner, Dr. Nils Faltermann, Dr. Pablo Goldenzweig und Dr. Matthias Schnepf danken, für jeden Rat, Hilfe und die immer offenen Türen.

Meinen Freund*innen außerhalb der Arbeit bin ich unglaublich dankbar für jede Möglichkeit, Zeit mit ihnen zu verbringen, wunderbare Urlaube (auch wenn ich etwas zu oft vielleicht arbeiten musste in dem einen oder anderen Urlaub), tolle Gespräche und einfach ihre Präsenz in meinem Leben. Ich kann nicht alle aufzählen, ich möchte aber ein paar der wichtigsten nennen: allen voran möchte ich Chrissi und Lina danken, die seit dem ersten Semester immer an meiner Seite sind, Laura, die mit mir bei Belle II Masterarbeit gemacht hat, Ioannis für die Kaffeepausen am PHI, die Heidelberger Marcus, Ananda, Philipp, Annette, Nicho, Joschka, die Maschinenbauer Steffen, Joscha und Paul, meine Physikerfreund*innen Tobi, Maria, Jakob, Jana und Laurin, meine beiden ältesten Schulfreundinnen Dani und Anna, und so vielen mehr.

Als letztes möchte ich meiner Familie danken. Ohne sie wäre es von Anfang an nicht möglich gewesen, das Studium zu machen, geschweige denn den PhD. Meine Eltern Jeanette und Manfred haben mich immer, in jeder Situation, unterstützt und geholfen, wo es ging, im Falle des PhDs von Wohnung suchen in Karlsruhe 2014 bis zur Organisation meiner PhD-Feier 2025. Mein Bruder Ludwig, der glücklicherweise auch in Karlsruhe gelandet ist, war auch immer bereit, mich zu unterstützen (oder mir das Auto zu bringen). Auch in schwierigen Zeiten haben mir alle drei stets Motivation zugesprochen und mir geholfen.

Bibliography

[1] W. Altmannshofer et al. "The Belle II Physics Book". In: *PTEP* 2019 (2019). Ed. by E. Kou and P. Urquijo, p. 123C01. DOI: 10.1093/ptep/ptz106. arXiv: 1808.10567.

[2] Belle II Collaboration et al. "Evidence for $B^+ \to K^+\nu\bar{\nu}$ decays". In: *Physical Review D* 109 (2024). Publisher: American Physical Society, p. 112006. DOI: 10.1103/PhysRevD.109.112006.

[3] D. Bečirević et al. "Scalar leptoquarks from grand unified theories to accommodate the B-physics anomalies". In: *Physical Review D* 98 (2018). Publisher: American Physical Society, p. 055003. DOI: 10.1103/PhysRevD.98.055003.

[4] J. M. Camalich et al. "Quark flavor phenomenology of the QCD axion". In: *Physical Review D* 102 (2020). Publisher: American Physical Society, p. 015023. DOI: 10.1103/PhysRevD.102.015023.

[5] A. Filimonova, R. Schäfer, and S. Westhoff. "Probing dark sectors with long-lived particles at Belle II". In: *Physical Review D* 101 (2020). Publisher: American Physical Society, p. 095006. DOI: 10.1103/PhysRevD.101.095006.

[6] *SuperKEKB/Belle II Complete 2024 Operations.* URL: https://www2.kek.jp/ipns/en/news/7015/?simply_static_page=1642969 Access date: 2025-05-20.

[7] Y. Iwasaki, B. Cheon, E. Won, and G. Varner. "Level 1 trigger system for the Belle II experiment". In: *2010 17th IEEE-NPSS Real Time Conference.* 2010, pp. 1–9. DOI: 10.1109/RTC.2010.5750454.

[8] M. T. Prim et al. "Design and Performance of the Belle II High Level Trigger". In: *PoS* ICHEP2020 (2021), p. 769. DOI: 10.22323/1.390.0769.

[9] B. Cheon. "Design of a Electromagnetic Calorimeter Trigger System for the Belle II Experiment". In: *Journal of the Korean Physical Society* 57 (2010), pp. 1369–1375. DOI: 10.3938/jkps.57.1369.

[10] S. Bähr et al. "The neural network first-level hardware track trigger of the Belle II experiment". In: *Nucl. Instrum. Meth. A* 1073 (2025), p. 170279. DOI: 10.1016/j.nima.2025.170279. arXiv: 2402.14962.

[11] J. Duarte et al. "Fast inference of deep neural networks in FPGAs for particle physics". In: *JINST* 13 (2018), P07027. DOI: 10.1088/1748-0221/13/07/P07027. arXiv: 1804.06913.

[12] C. N. Coelho et al. "Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors". en. In: *Nature Machine Intelligence* 3 (2021). Publisher: Nature Publishing Group, pp. 675–686. DOI: 10.1038/s42256-021-00356-5.

[13] S. Summers et al. "Fast inference of Boosted Decision Trees in FPGAs for particle physics". In: *Journal of Instrumentation* 15 (2020), P05026. DOI: 10.1088/1748-0221/15/05/P05026.

[14] Y. Iiyama et al. "Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics". In: *Frontiers in Big Data* Volume 3 - 2020 (2021). DOI: 10.3389/fdata.2020.598927.

[15] S. R. Qasim, J. Kieseler, Y. Iiyama, and M. Pierini. "Learning representations of irregular particle-detector geometry with distance-weighted graph networks". In: *The European Physical Journal C* 79 (2019), p. 608. DOI: 10.1140/epjc/s10052-019-7113-9. arXiv: 1902.07987 [physics].

[16] J. Kieseler. "Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data". In: *Eur. Phys. J. C* 80 (2020), p. 886. DOI: 10.1140/epjc/s10052-020-08461-2. arXiv: 2002.03605.

[17] K. Akai, K. Furukawa, and H. Koiso. "SuperKEKB Collider". In: *Nucl. Instrum. Meth. A* 907 (2018), pp. 188–199. DOI: 10.1016/j.nima.2018.08.017. arXiv: 1809.01958.

[18] M. Bona et al. *SuperB: A High-Luminosity Asymmetric e+ e- Super Flavor Factory. Conceptual Design Report.* 2007. arXiv: 0709.0451.

[19] E. Kou et al. "The Belle II Physics Book". In: *Progress of Theoretical and Experimental Physics* 2019 (2019). DOI: 10.1093/ptep/ptz106.

[20] T. Abe et al. *Belle II Technical Design Report.* 2010. arXiv: 1011.0352.

[21] A. Abashian et al. "The Belle Detector". In: *Nucl. Instrum. Meth. A* 479 (2002), pp. 117–232. DOI: 10.1016/S0168-9002(01)02013-7.

[22] M. D. Bauer. "Measuring the Branching Fraction of B $\rightarrow \rho\ell\nu_\ell$ Decays with the Belle II Experiment". PhD Thesis. KIT, Karlsruhe, KIT, Karlsruhe, ETP, 2023. DOI: 10.5445/IR/1000165627.

[23] P. Ahlburg et al. "The new and complete Belle II DEPFET pixel detector: Commissioning and previous operational experience". In: *Nucl. Instrum. Meth. A* 1068 (2024), p. 169763. DOI: 10.1016/j.nima.2024.169763.

[24] J. Kemmer and G. Lutz. “New detector concepts”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 253 (1987), pp. 365–377. DOI: `https://doi.org/10.1016/0168-9002(87)90518-3`.

[25] K. Ravindran et al. *Operational experience and performance of the Silicon Vertex Detector after the first long shutdown of Belle II.* 2025. arXiv: `2504.17715`.

[26] N. Taniguchi. “Central Drift Chamber for Belle-II”. In: *JINST* 12 (2017). Ed. by L. Shekhtman, p. C06014. DOI: `10.1088/1748-0221/12/06/C06014`.

[27] H. Atmacan et al. “The Imaging Time-of-Propagation Detector at Belle II”. In: (2025). arXiv: `2504.19090`.

[28] K. Uno. “Operation and performance of the Belle II Aerogel RICH detector”. In: *Nucl. Instrum. Meth. A* 1056 (2023), p. 168635. DOI: `10.1016/j.nima.2023.168635`.

[29] C. Ketter et al. *Design and Commissioning of Readout Electronics for a K_L^0 and μDetector at the Belle II Experiment.* 2025. arXiv: `2502.02724`.

[30] T. Aushev et al. “A scintillator based endcap K_L^0 and muon detector for the Belle II experiment”. In: *Nucl. Instrum. Meth. A* 789 (2015), pp. 134–142. DOI: `10.1016/j.nima.2015.03.060`. arXiv: `1406.3267`.

[31] The Belle II Collaboration. *Belle II Analysis Software Framework (basf2).* Version release-06-02-00. 2025. DOI: `10.5281/zenodo.14710811`.

[32] **Belle II Framework Software Group**, T. Kuhr et al. “The Belle II Core Software: Belle II Framework Software Group”. In: *Computing and Software for Big Science* 3 (2018). DOI: `10.1007/s41781-018-0017-9`.

[33] S. Agostinelli et al. “GEANT4 - A Simulation Toolkit”. In: *Nucl. Instrum. Meth. A* 506 (2003), pp. 250–303. DOI: `10.1016/S0168-9002(03)01368-8`.

[34] B. Shwartz and on behalf of BELLE II calorimeter group. “Electromagnetic calorimeter of the Belle II detector”. In: *Journal of Physics: Conference Series* 928 (2017), p. 012021. DOI: `10.1088/1742-6596/928/1/012021`.

[35] S. Longo 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”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 982 (2020), p. 164562. DOI: `10.1016/j.nima.2020.164562`.

[36] **Belle II Collaboration**, I. Adachi et al. “Measurement of the $e^+e^- \to \pi^+\pi^-\pi^0$ cross section in the energy range 0.62–3.50 GeV at Belle II”. In: *Phys. Rev. D* 110 (11 2024), p. 112005. DOI: `10.1103/PhysRevD.110.112005`.

[37] I. Adachi et al. “Search for lepton-flavor-violating $\tau^- \to \mu^-\mu^+\mu^-$ decays at Belle II”. In: *Journal of High Energy Physics* 2024 (2024). DOI: `10.1007/jhep09(2024)062`.

[38] **Belle II Collaboration**, I. Adachi et al. *Search for a dark Higgs boson produced in association with inelastic dark matter at the Belle II experiment.* 2025. arXiv: 2505.09705 [hep-ex]. URL: https://arxiv.org/abs/2505.09705.

[39] F. Abudinén et al. "Search for Axion-Like Particles produced in e^+e^- collisions at Belle II". In: *Phys. Rev. Lett.* 125 (2020), p. 161806. DOI: 10.1103/PhysRevLett.125.161806. arXiv: 2007.13071.

[40] Y.-T. Lai et al. "Design of the Global Reconstruction Logic in the Belle II Level-1 Trigger system". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1078 (2025), p. 170577. DOI: 10.1016/j.nima.2025.170577.

[41] Y. Kim et al. "Overview of the Electromagnetic Calorimeter Trigger system at the Belle II Experiment". In: 2019, p. 804. DOI: 10.22323/1.340.0804.

[42] S. Kim, I. Lee, Y. Unno, and B. Cheon. "Status of the electromagnetic calorimeter trigger system at Belle II." In: *J. Phys. Conf. Ser.* 928 (2017), p. 012022. DOI: 10.1088/1742-6596/928/1/012022.

[43] S.-H. Kim. "Status of the electromagnetic calorimeter trigger system at the Belle II". CALOR 2016. 2016. URL: https://indico.cern.ch/event/472938/contributions/1150758/.

[44] B. G. Cheon et al. "Electromagnetic calorimeter trigger at Belle". In: *Nucl. Instrum. Meth. A* 494 (2002). Ed. by G. V. Fedotovich and B. Khazin, pp. 548–554. DOI: 10.1016/S0168-9002(02)01547-4.

[45] M. G. Pia and G. Weidenspointner. *Monte Carlo Simulation for Particle Detectors.* 2012. arXiv: 1208.0047 [physics.comp-ph]. URL: https://arxiv.org/abs/1208.0047.

[46] V. Aulchenko et al. "Time and energy reconstruction at the electromagnetic calorimeter of the Belle-II detector". In: *Journal of Instrumentation* 12 (2017), p. C08001. DOI: 10.1088/1748-0221/12/08/C08001.

[47] F. Wemmer et al. "Photon Reconstruction in the Belle II Calorimeter Using Graph Neural Networks". In: *Comput. Softw. Big Sci.* 7 (2023), p. 13. DOI: 10.1007/s41781-023-00105-w. arXiv: 2306.04179.

[48] *Atomic and nuclear properties of materials: cesium iodide (csi).* URL: https://pdg.lbl.gov/2009/AtomicNuclearProperties/HTML_PAGES/141.html Access date: 2025-06-02.

[49] A. Natochii et al. "Measured and projected beam backgrounds in the Belle II experiment at the SuperKEKB collider". In: *Nucl. Instrum. Meth. A* 1055 (2023), p. 168550. DOI: 10.1016/j.nima.2023.168550. arXiv: 2302.01566.

[50] KEK Accelerator Laboratory. *Strategic Accelerator Design (SAD)*. https://acc-physics.kek.jp/SAD/. Accessed: 2025-06-04. 2020.

[51] R. Kleiss and H. Burkhardt. "BBBREM: Monte Carlo simulation of radiative Bhabha scattering in the very forward direction". In: *Comput. Phys. Commun.* 81 (1994), pp. 372–380. DOI: 10.1016/0010-4655(94)90085-X. arXiv: hep-ph/9401333 [hep-ph].

[52] S. Jadach, W. Placzek, and B. F. L. Ward. "BHWIDE 1.00: O(alpha) YFS exponentiated Monte Carlo for Bhabha scattering at wide angles for LEP-1 / SLC and LEP-2". In: *Phys. Lett. B* 390 (1997), pp. 298–308. DOI: 10.1016/S0370-2693(96)01382-2. arXiv: hep-ph/9608412 [hep-ph].

[53] F. A. Berends, P. H. Daverveldt, and R. Kleiss. "Complete Lowest Order Calculations for Four Lepton Final States in Electron-Positron Collisions". In: *Nucl. Phys. B* 253 (1985), pp. 441–463. DOI: 10.1016/0550-3213(85)90541-3.

[54] G. Balossini et al. "Matching perturbative and parton shower corrections to Bhabha process at flavour factories". In: *Nucl. Phys. B* 758 (2006), pp. 227–253. DOI: 10.1016/j.nuclphysb.2006.09.022. arXiv: hep-ph/0607181 [hep-ph].

[55] D. E. Groom, N. V. Mokhov, and S. I. Striganov. "Muon stopping power and range tables 10-MeV to 100-TeV". In: *Atom. Data Nucl. Data Tabl.* 78 (2001), pp. 183–356. DOI: 10.1006/adnd.2001.0861.

[56] S. Jadach, B. F. L. Ward, and Z. Was. "The Precision Monte Carlo event generator KK for two fermion final states in e^- e^+ collisions". In: *Comput. Phys. Commun.* 130 (2000), pp. 260–325. DOI: 10.1016/S0010-4655(00)00048-5. arXiv: hep-ph/9912214 [hep-ph].

[57] M. Oreglia. "A Study of the Reactions $\psi' \to \gamma\gamma\psi$". Thesis. 1980.

[58] V. Bertacchi et al. "Track finding at Belle II". In: *Comput. Phys. Commun.* 259 (2021), p. 107610. DOI: 10.1016/j.cpc.2020.107610. arXiv: 2003.12466.

[59] R. Brun and F. Rademakers. "ROOT: An object oriented data analysis framework". In: *Nucl. Instrum. Meth. A* 389 (1997). Ed. by M. Werlen and D. Perret-Gallix, pp. 81–86. DOI: 10.1016/S0168-9002(97)00048-X.

[60] J. Matusche. *Graph Building for Graph Neural Networks for Photon Reconstruction in the Belle II Calorimeter*. 2023. URL: https://publish.etp.kit.edu/record/22184.

[61] J. M. Stokes et al. "A Deep Learning Approach to Antibiotic Discovery". English. In: *Cell* 180 (2020). Publisher: Elsevier, 688–702.e13. DOI: 10.1016/j.cell.2020.01.021.

[62] C. Eksombatchai et al. “Pixie: A System for Recommending 3+ Billion Items to 200+ Million Users in Real-Time”. In: *Proceedings of the 2018 World Wide Web Conference*. WWW ’18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, pp. 1775–1784. DOI: 10.1145/3178876.3186183.

[63] D. Matsunaga, T. Suzumura, and T. Takahashi. *Exploring Graph Neural Networks for Stock Market Predictions with Rolling Window Analysis*. 2019. DOI: 10.48550/arXiv.1909.10660. arXiv: 1909.10660 [cs].

[64] I. J. King and H. H. Huang. “Euler: Detecting Network Lateral Movement via Scalable Temporal Link Prediction”. In: *ACM Trans. Priv. Secur.* 26 (2023), 35:1–35:36. DOI: 10.1145/3588771.

[65] L. Reuter et al. “End-to-End Multi-track Reconstruction Using Graph Neural Networks at Belle II”. In: *Computing and Software for Big Science* 9, 6 (2025), p. 6. DOI: 10.1007/s41781-025-00135-6. arXiv: 2411.13596 [physics.ins-det].

[66] J. Pata et al. “MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks”. In: *Eur. Phys. J. C* 81 (2021), p. 381. DOI: 10.1140/epjc/s10052-021-09158-w. arXiv: 2101.08578.

[67] M. Abumusabh, J. Cerasoli, G. Dujany, and C. Santos. *Graph-based Full Event Interpretation: a graph neural network for event reconstruction in Belle II*. 2025. arXiv: 2503.09401.

[68] O. Atkinson et al. “Anomaly detection with convolutional Graph Neural Networks”. In: *JHEP* 08 (2021), p. 080. DOI: 10.1007/JHEP08(2021)080. arXiv: 2105.07988.

[69] I. Adachi et al. “New graph-neural-network flavor tagger for Belle II and measurement of $\sin 2\Phi_1$ in $\mathrm{B}^0 \to \mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ decays”. en. In: *Physical Review D* 110 (2024), p. 012001. DOI: 10.1103/PhysRevD.110.012001.

[70] Y. Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Trans. Graph.* 38 (2019), 146:1–146:12. DOI: 10.1145/3326362.

[71] X. Ju et al. “Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors”. In: *33rd Annual Conference on Neural Information Processing Systems*. 2020. arXiv: 2003.11603.

[72] P. Goswami and D. Bhatia. “Predicting Post-Route Quality of Results Estimates for HLS Designs using Machine Learning”. In: *2022 23rd International Symposium on Quality Electronic Design (ISQED)*. 2022, pp. 45–50. DOI: 10.1109/ISQED54688.2022.9806201.

[73] J. Zhou et al. “Graph neural networks: A review of methods and applications”. In: *AI Open* 1 (2020), pp. 57–81. DOI: 10.1016/j.aiopen.2021.01.001.

[74] G. DeZoort, P. W. Battaglia, C. Biscarat, and J.-R. Vlimant. “Graph neural networks at the Large Hadron Collider”. en. In: *Nature Reviews Physics* 5 (2023). Publisher: Nature Publishing Group, pp. 281–303. DOI: 10.1038/s42254-023-00569-0.

[75] G. Aad et al. “Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1”. In: *Eur. Phys. J. C* 77 (2017), p. 490. DOI: 10.1140/epjc/s10052-017-5004-5. arXiv: 1603.02934.

[76] A. M. Sirunyan et al. “Particle-flow reconstruction and global event description with the CMS detector”. In: *JINST* 12 (2017), P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965.

[77] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV]. URL: https://arxiv.org/abs/1506.02640.

[78] J. Redmon and A. Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016. arXiv: 1612.08242 [cs.CV]. URL: https://arxiv.org/abs/1612.08242.

[79] M. Fey and J. E. Lenssen. *Fast Graph Representation Learning with PyTorch Geometric*. 2019. DOI: 10.48550/arXiv.1903.02428. arXiv: 1903.02428 [cs].

[80] A. Paszke et al. “PyTorch: an imperative style, high-performance deep learning library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 721. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 8026–8037. Access date: 2025-04-13.

[81] F. Chollet et al. *Keras*. 2015. URL: https://keras.io.

[82] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. DOI: 10.48550/arXiv.1603.04467. arXiv: 1603.04467.

[83] *Advanced Mini-Batching — pytorch_geometric documentation*. URL: https://pytorch-geometric.readthedocs.io/en/latest/advanced/batching.html Access date: 2025-04-13.

[84] S. Ioffe and C. Szegedy. “Batch normalization: accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456. Access date: 2025-04-13.

[85] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. en. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[86] L. Biewald. *Experiment Tracking with Weights and Biases*. 2020. URL: https://www.wandb.com/.

[87] "IEEE Standard for Binary Floating-Point Arithmetic". In: *ANSI/IEEE Std 754-1985* (1985), pp. 1–20. DOI: 10.1109/IEEESTD.1985.82928.

[88] *Variable-Precision DSP Block*. en. URL: https://www.intel.com/content/www/us/en/docs/programmable/683332/current/variable-precision-dsp-block.html Access date: 2025-05-04.

[89] D. L. N. Hettiarachchi, V. S. P. Davuluru, and E. J. Balster. "Integer vs. Floating-Point Processing on Modern FPGA Technology". In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. 2020, pp. 0606–0612. DOI: 10.1109/CCWC47524.2020.9031118.

[90] P. Yin et al. *Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets*. 2019. DOI: 10.48550/arXiv.1903.05662. arXiv: 1903.05662 [cs].

[91] A. F. Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. 2019. DOI: 10.48550/arXiv.1803.08375. arXiv: 1803.08375 [cs].

[92] M. Zhu and S. Gupta. *To prune, or not to prune: exploring the efficacy of pruning for model compression*. 2017. DOI: 10.48550/arXiv.1710.01878. arXiv: 1710.01878 [stat].

[93] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. eng. Adaptive computation and machine learning. Cambridge, Massachusetts ; London, England: The MIT Press, 2016.

[94] L. Li et al. "Efficient Hyperparameter Optimization and Infinitely Many Armed Bandits". In: *CoRR* abs/1603.06560 (2016). arXiv: 1603.06560. URL: http://arxiv.org/abs/1603.06560.

[95] D. Sun, Z. Liua, J. Zhao, and H. Xu. "Belle2Link: A Global Data Readout and Transmission for Belle II Experiment at KEK". In: *Physics Procedia*. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011) 37 (2012), pp. 1933–1939. DOI: 10.1016/j.phpro.2012.01.036.

[96] S. Yamada et al. "Data Acquisition System for the Belle II Experiment". In: *IEEE Trans. Nucl. Sci.* 62 (2015), pp. 1175–1180. DOI: 10.1109/TNS.2015.2424717.

[97] P. Vaidya. "An O(n log n) Algorithm for the All-Nearest-Neighbors Problem." In: *Discrete & computational geometry* 4 (1989), pp. 101–116. URL: http://eudml.org/doc/131067.

[98] AMD. *Vitis unified software platform*. https://www.amd.com/de/products/software/adaptive-socs-and-fpgas/vitis.html. Version 2024.1, accessed 2025-06-04. 2025.

[99] I. Adachi et al. “Measurement of the integrated luminosity of data samples collected during 2019-2022 by the Belle II experiment*”. In: *Chin. Phys. C* 49 (2025), p. 013001. DOI: 10.1088/1674-1137/ad806c. arXiv: 2407.00965.

[100] E. Kovalenko et al. “Luminosity online monitor for the Belle II detector”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (2025), p. 170614. DOI: 10.1016/j.nima.2025.170614.

Appendices

Appendix A

Angle ID Mapping

Table A.1: Mapping of θ and ϕ IDs to the approximate angles. The values vary for the different detector regions.

θ ID	θ in °
1	15.62
2	22.49
3	29.1
4	34.84
5	40.78
6	47.54
7	55.15
8	63.57
9	72.74
10	82.47
11	92.52
12	102.45
13	111.93
14	120.74
15	126.8
17	145.89
16	134.31

ϕ ID	ϕ in °
1	17.59
2	20.25
3	40.09
4	42.75
5	62.59
6	65.25
7	85.09
8	87.75
9	107.59
10	110.25
11	130.09
12	132.75
13	152.59
14	155.25
15	175.09
16	177.75
17	197.59
18	200.25
19	220.09
20	222.75
21	242.59
22	245.25
23	265.09
24	267.75
25	287.59
26	290.25
27	310.09
28	312.75
29	332.59
30	335.25
31	355.25
32	357.78
33	327.52
34	337.52
35	347.52
36	357.52

Appendix B

Input quantizations

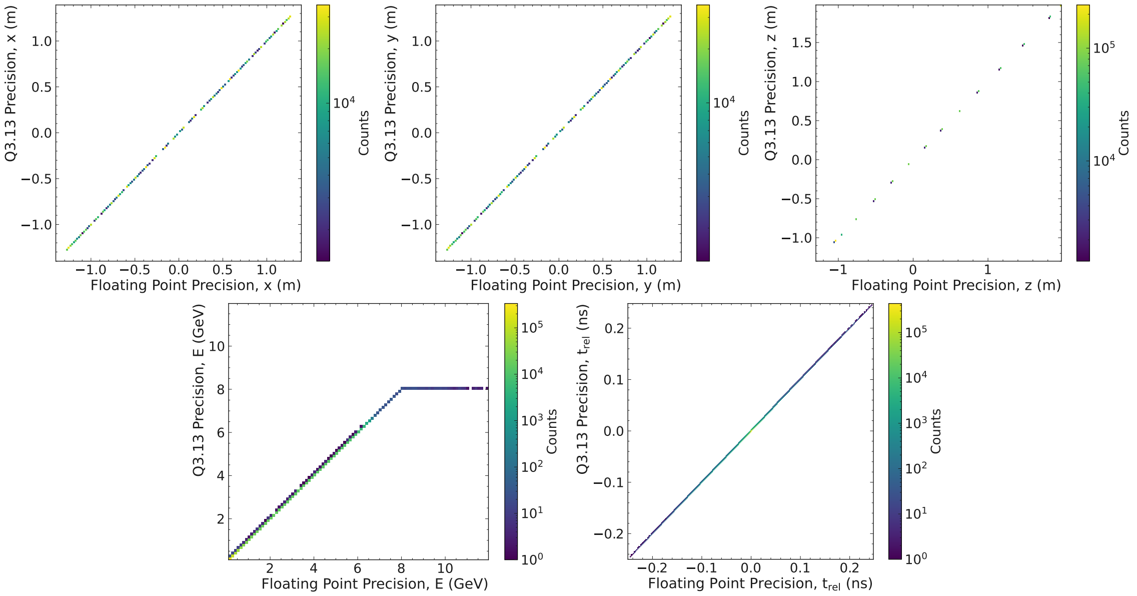


Figure B.1: 2D Distributions of input values with floating point single precision versus Q3.13 fixed point precision. All values are quantized with the same fixed point quantization. The dataset is the Category-One test sample using 15000 events. The top row shows the x, y, and z distributions, while the bottom row shows the energy and the relative timing. For the energy, the clipping due to the quantization range at 8 GeV is clearly visible. The bins are chosen as multiples of the quantization step to make differences more visible.

Appendix C

Resolutions of five Final Configuration Trainings

In this section, the remaining resolution distribution plots for the energy, x, y, and z resolutions for all five trainings in Section 7.1 are shown.

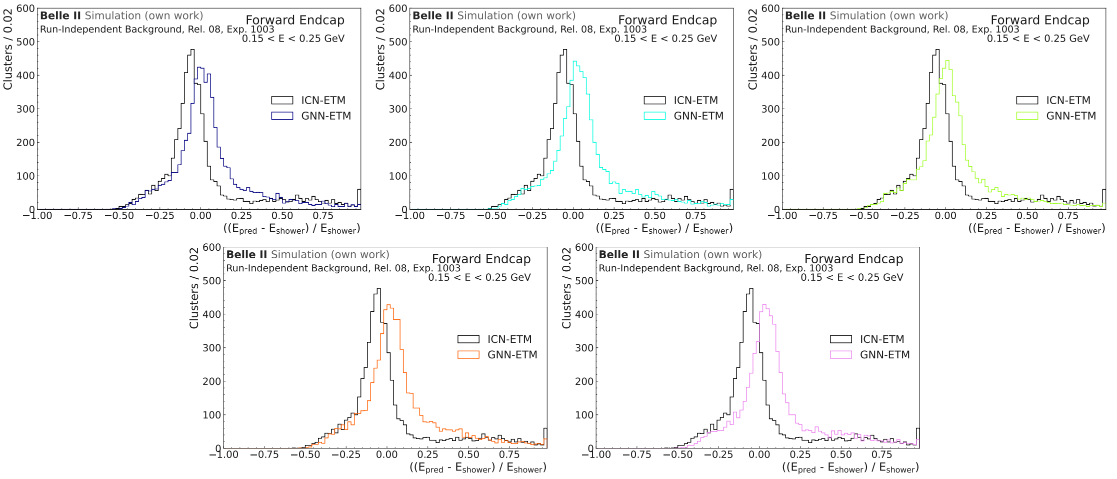


Figure C.1: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

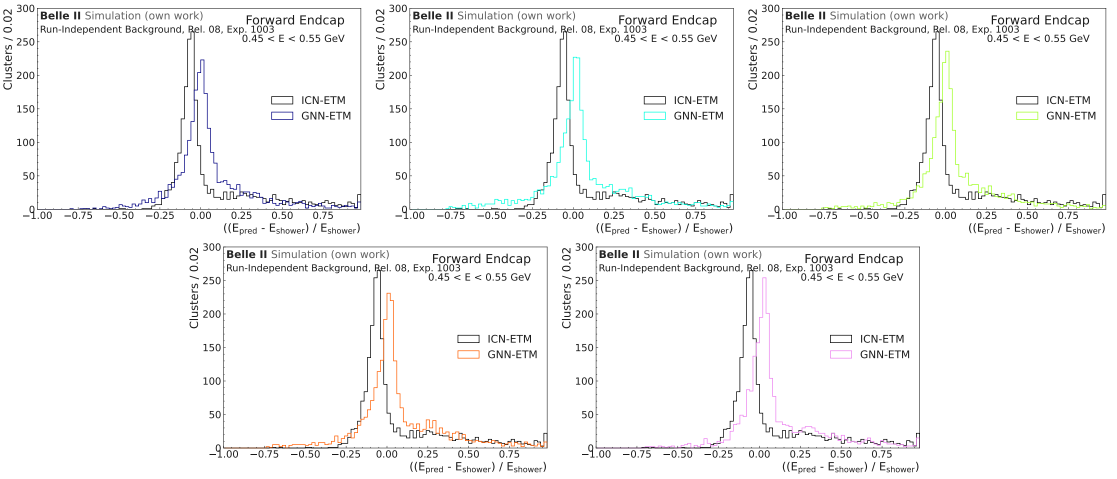


Figure C.2: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

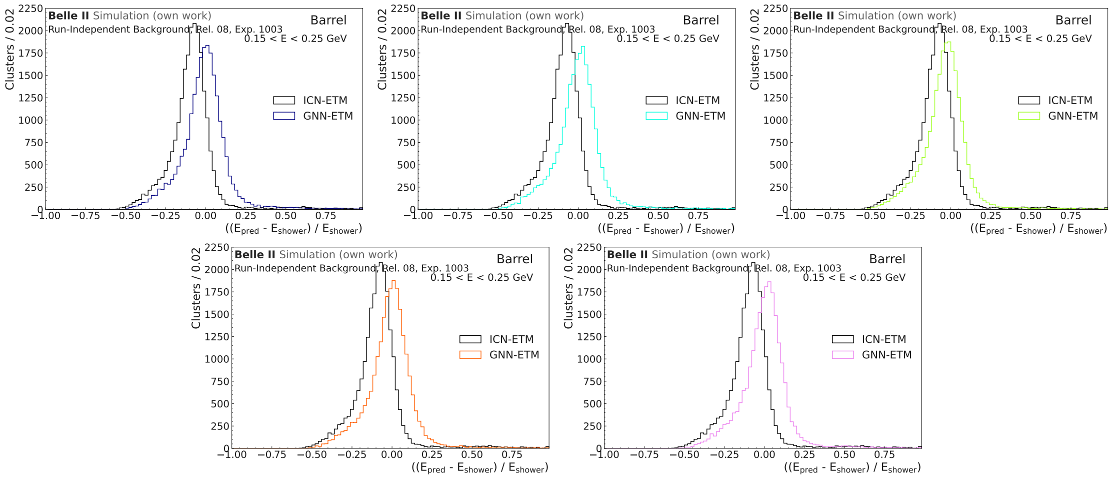


Figure C.3: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

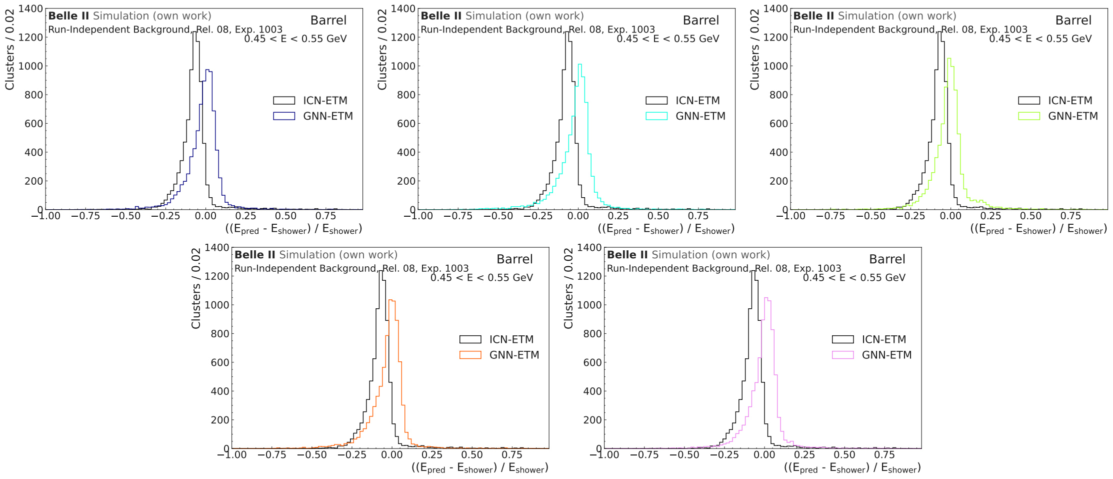


Figure C.4: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

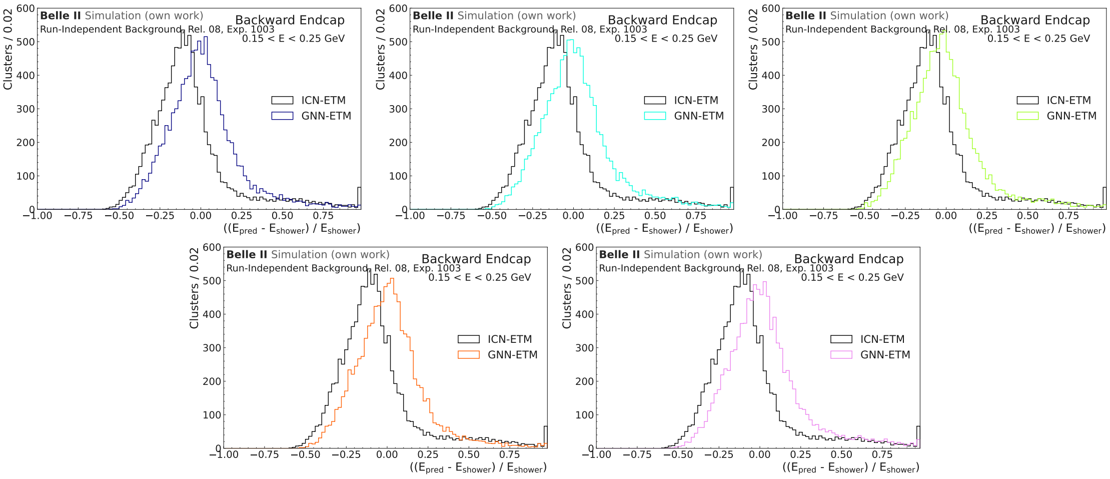


Figure C.5: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

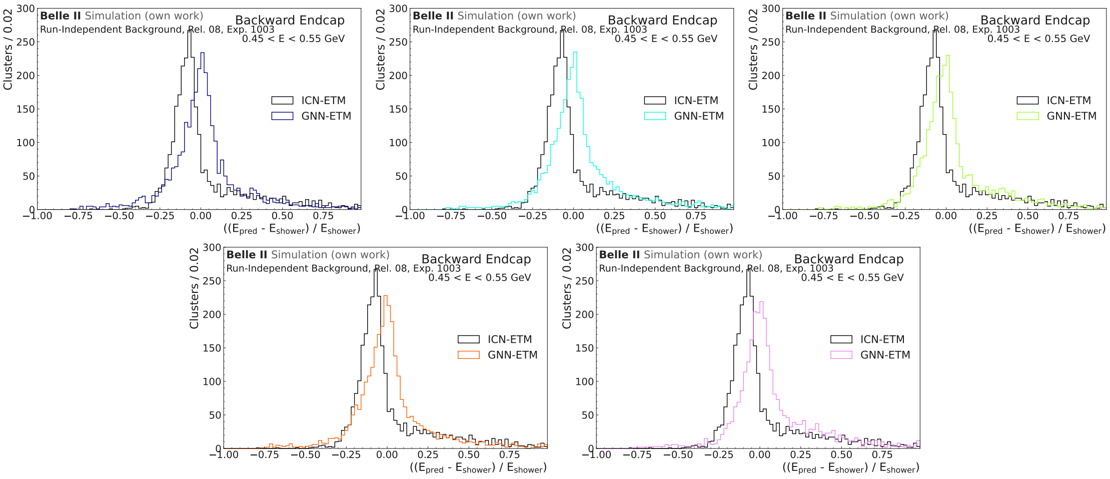


Figure C.6: Energy resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

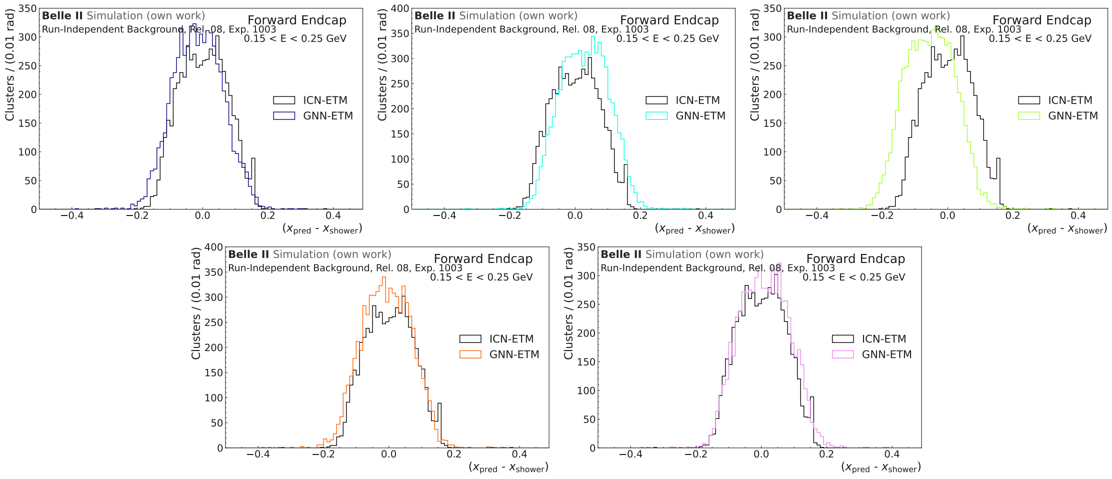


Figure C.7: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

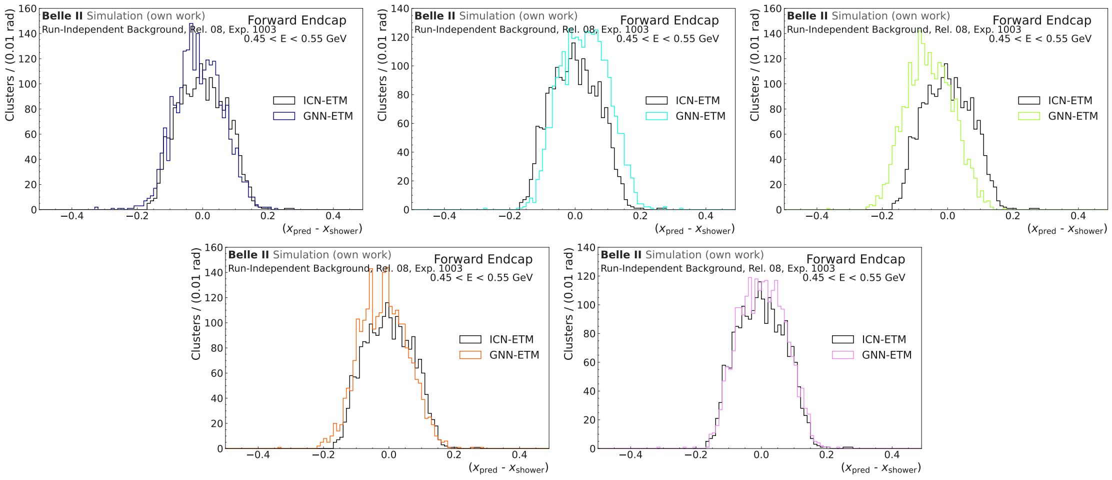


Figure C.8: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

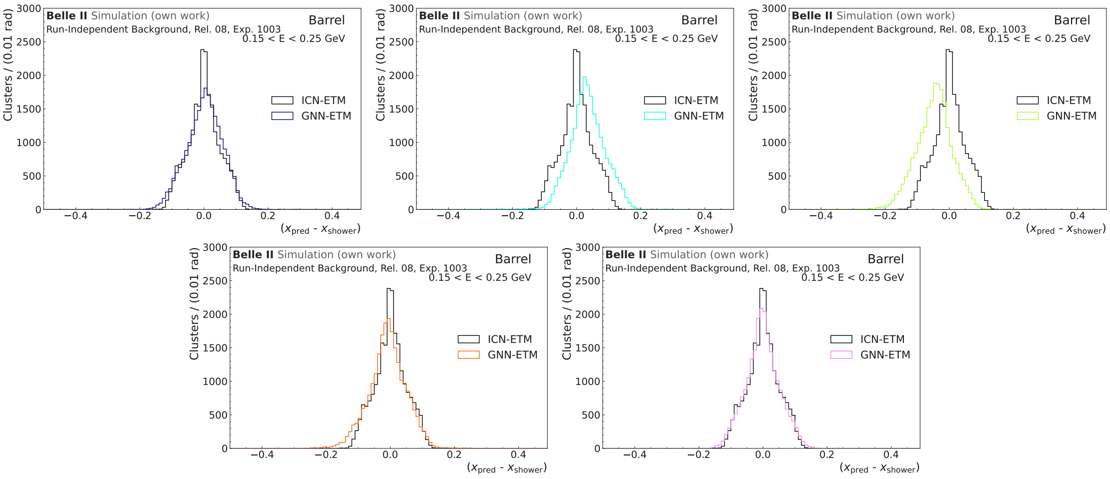


Figure C.9: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

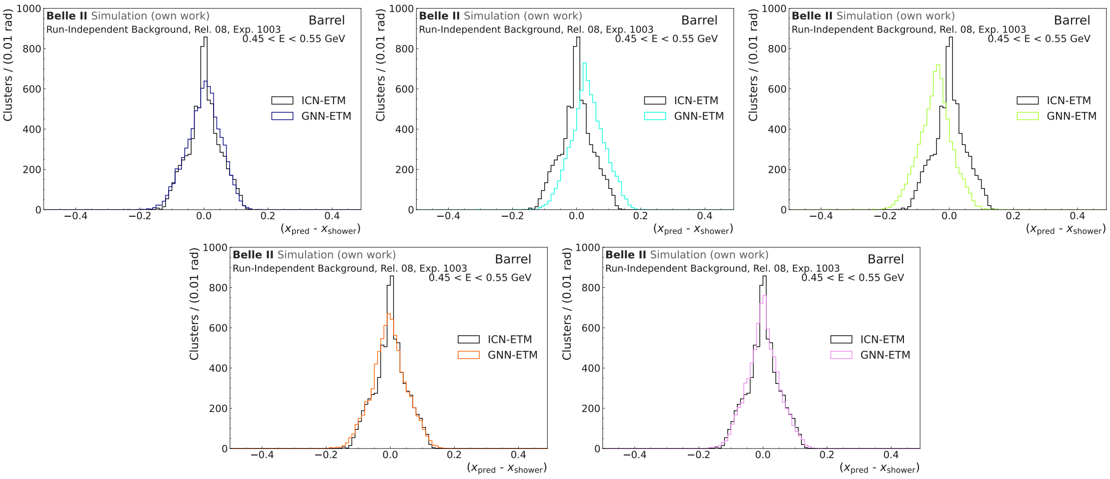


Figure C.10: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

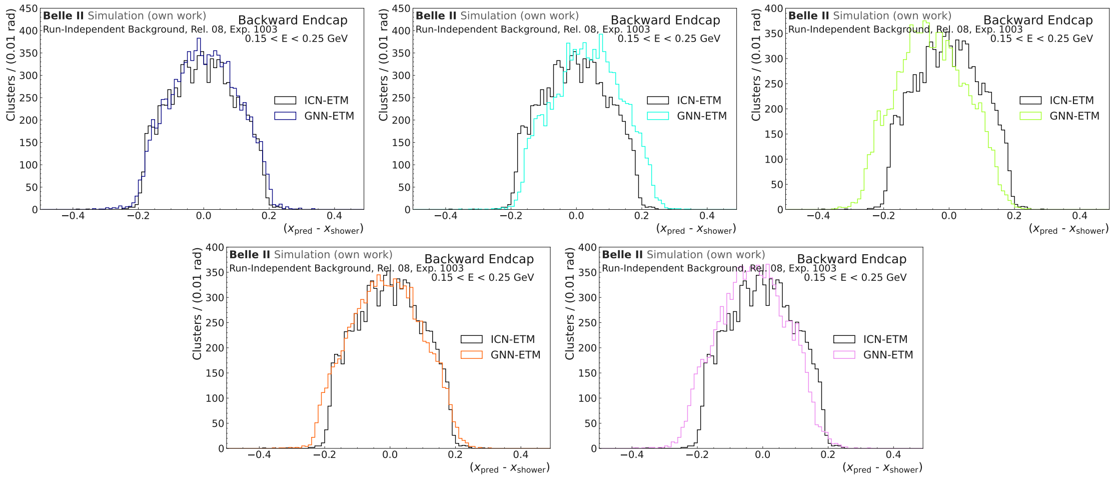


Figure C.11: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

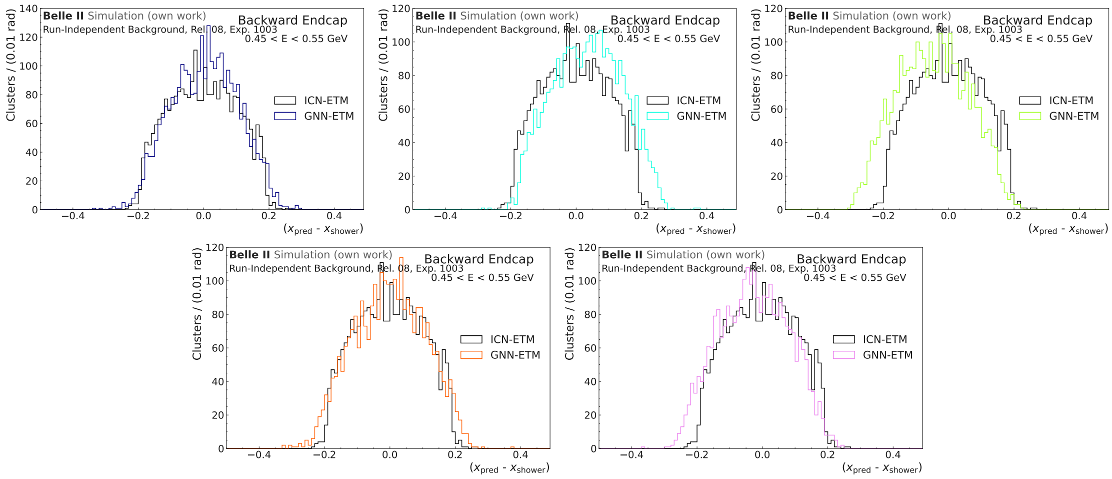


Figure C.12: x resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

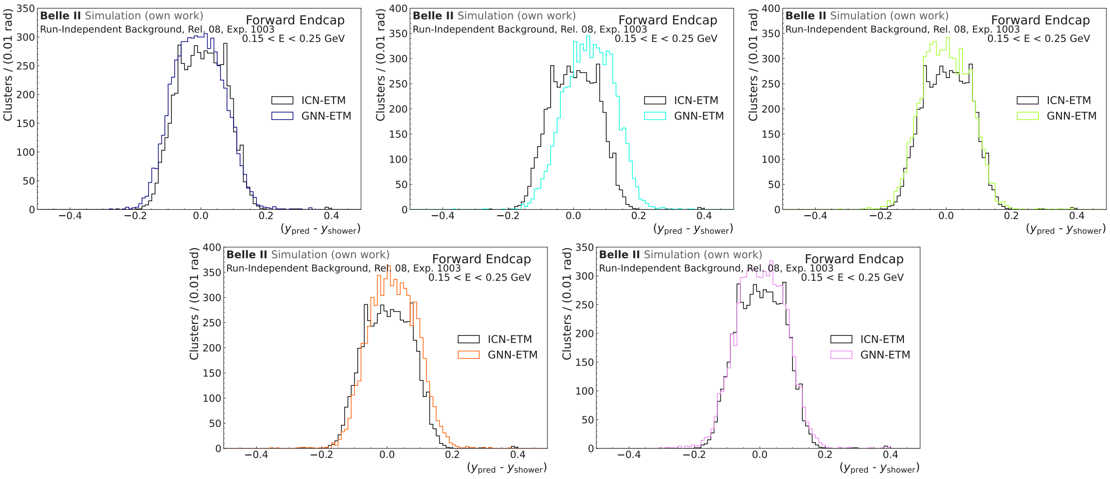


Figure C.13: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

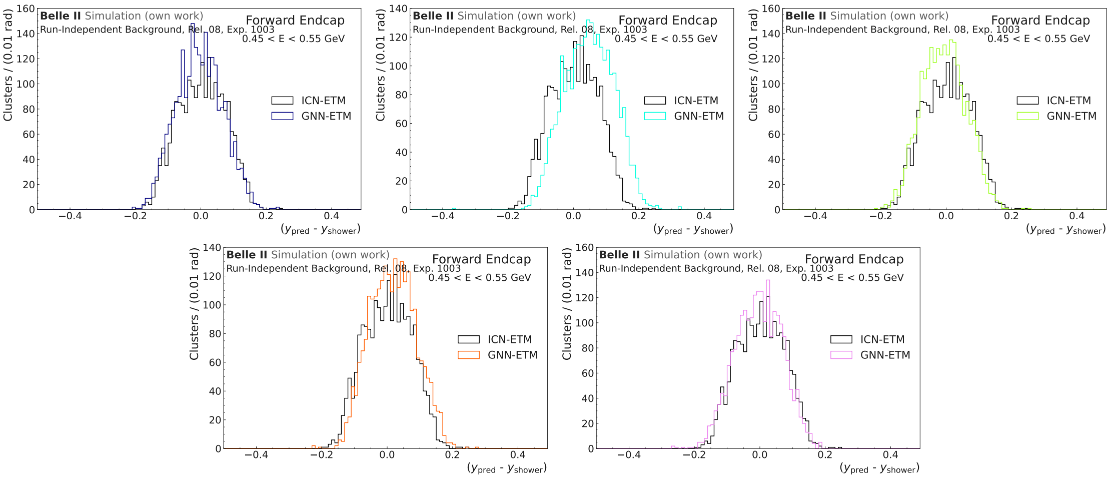


Figure C.14: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

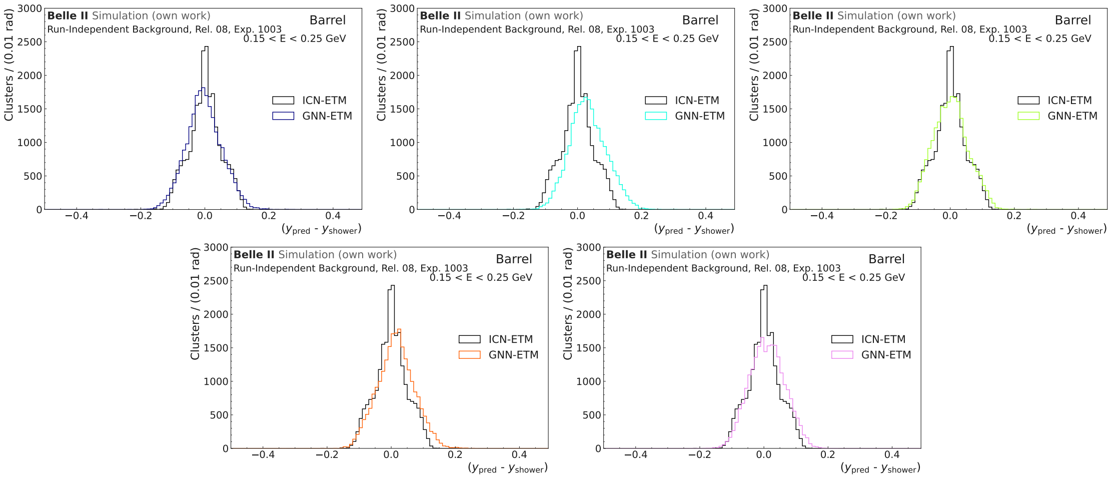


Figure C.15: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

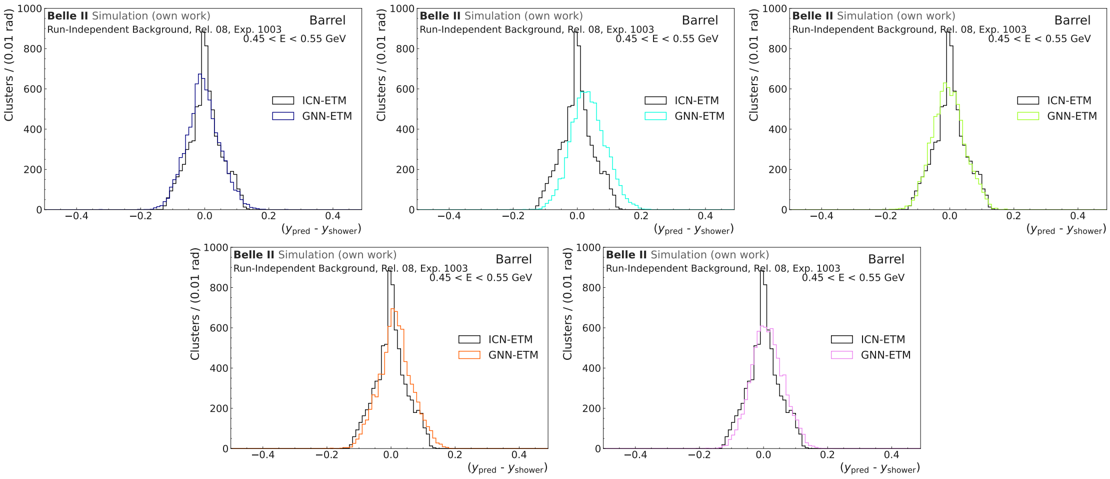


Figure C.16: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

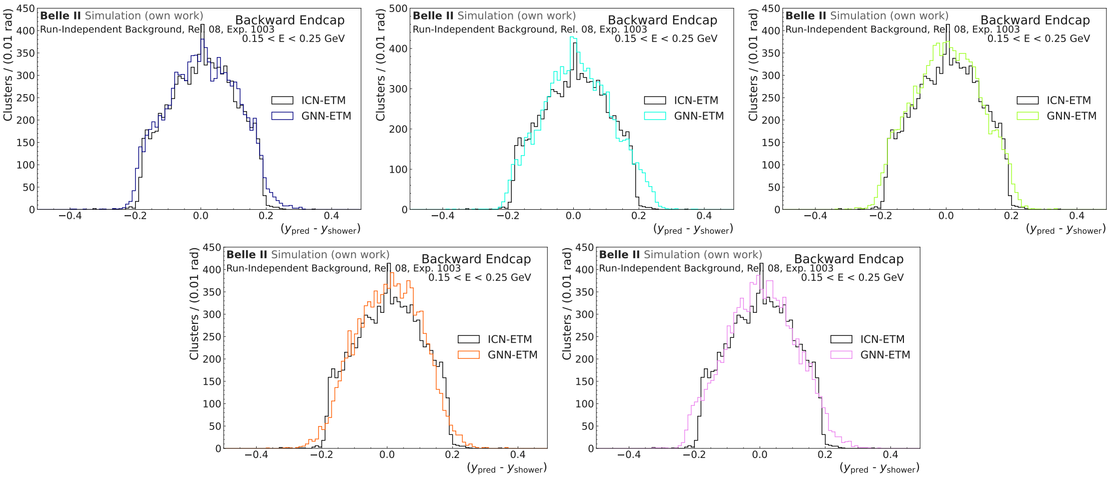


Figure C.17: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

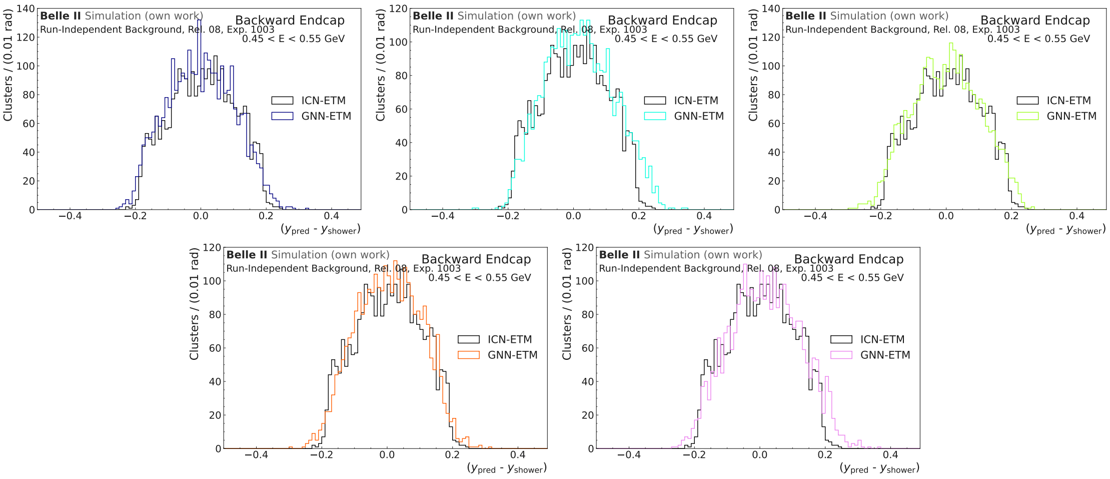


Figure C.18: y resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

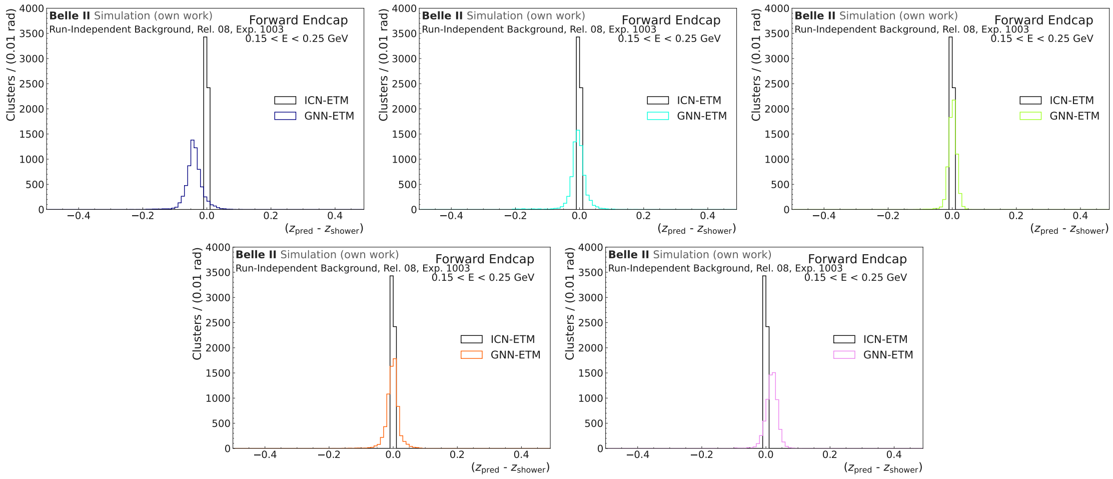


Figure C.19: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

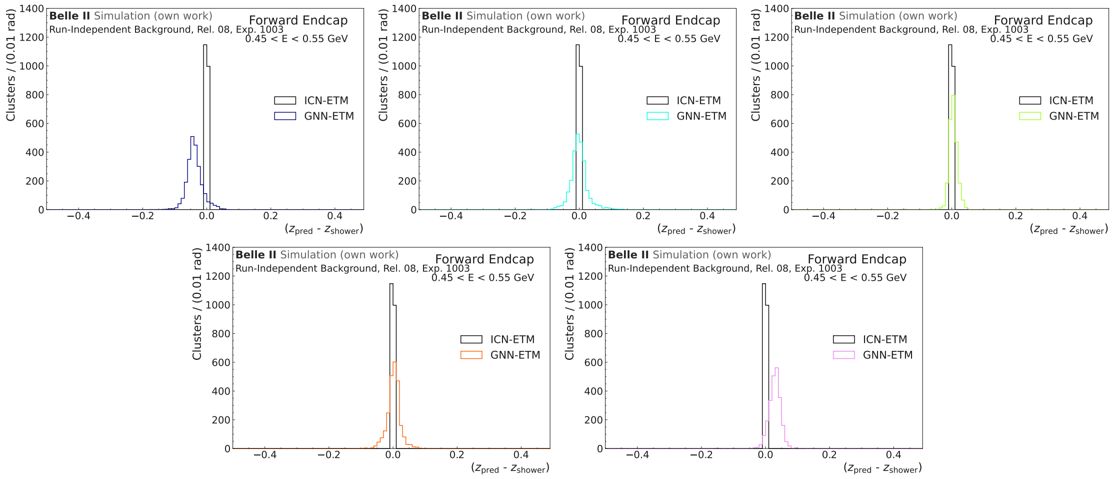


Figure C.20: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the forward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

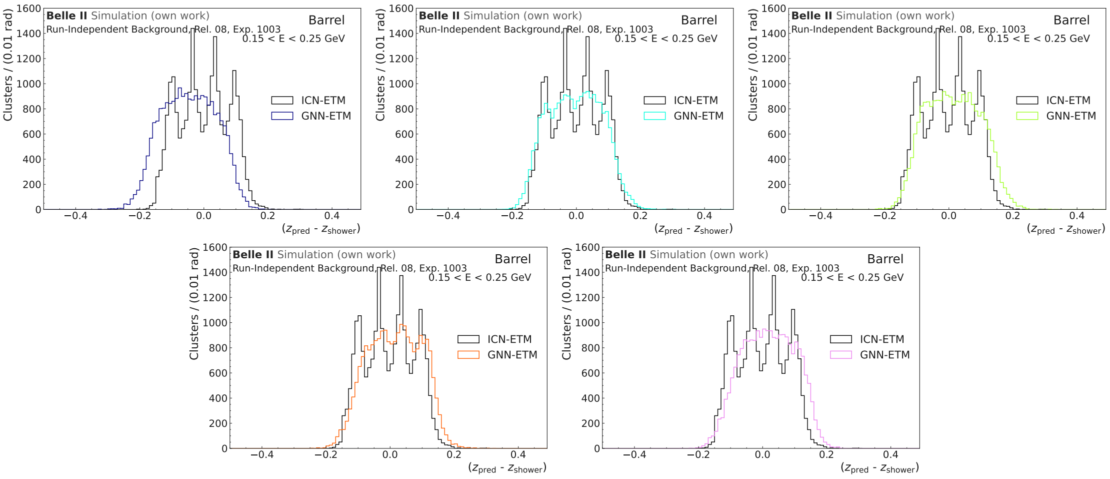


Figure C.21: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

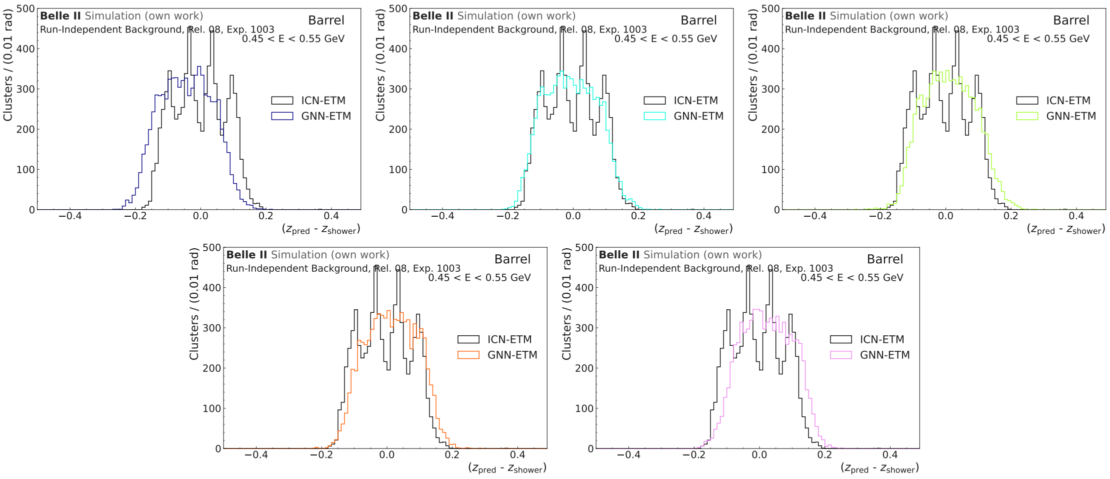


Figure C.22: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the barrel. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

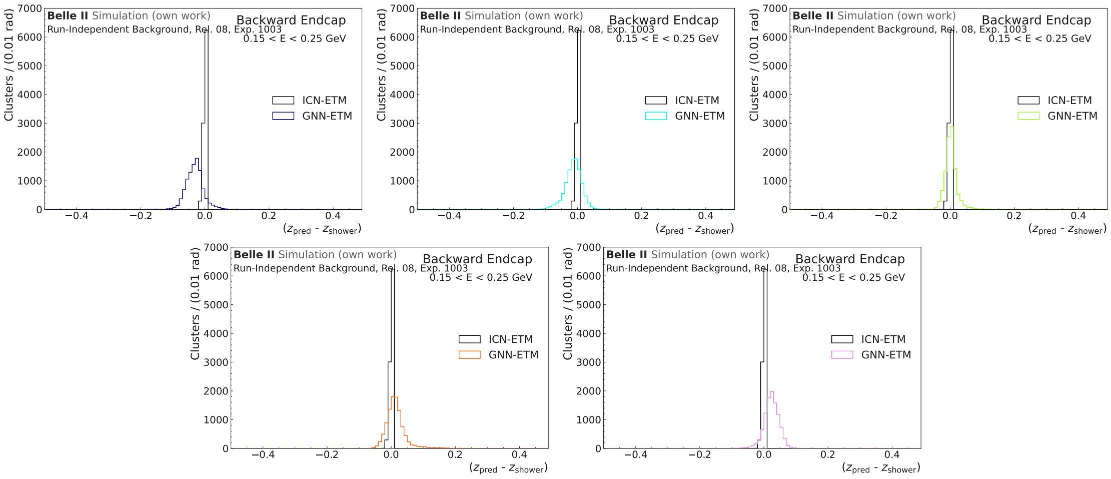


Figure C.23: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.15 - 0.25 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

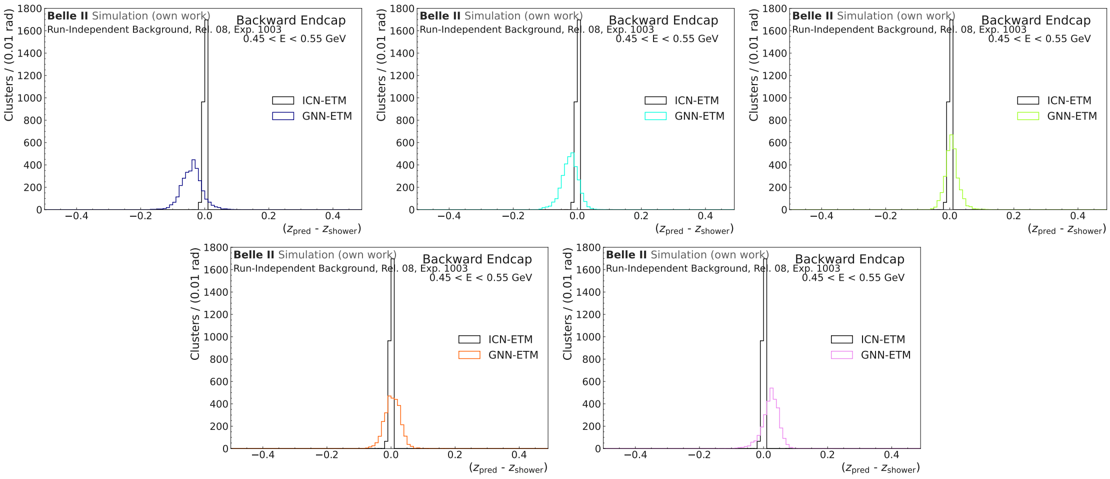


Figure C.24: z resolution histograms for five trainings evaluated on the Category-One test dataset for offline ECL clusters with a true energy between 0.45 - 0.55 GeV in the backward endcap. For a direct comparison, the ICN-ETM resolution is also shown in each figure. The resolutions are before bias correction to see possible offsets within the predictions.

Appendix D

Resolution Distributions and Trigger Efficiencies

D.1 GNN-ETM results for $e^+e^- \rightarrow e^+e^-(\gamma)$

D.1.1 Energy and Position Resolutions

e^- Resolutions

This section shows the energy and position resolutions for ICN-ETM and GNN-ETM clusters matched to the e^- offline ECL cluster from the selection shown in Section 8.2.3. The energy resolution η_E is shown for the barrel region and the forward endcap in Fig. D.1 and Fig. D.2, respectively. The position resolutions η_x, η_y, and η_z are also shown for the barrel region and the forward endcap in Fig. D.3, Fig. D.4, Fig. D.5, Fig. D.6,Fig. D.7, and Fig. D.8.

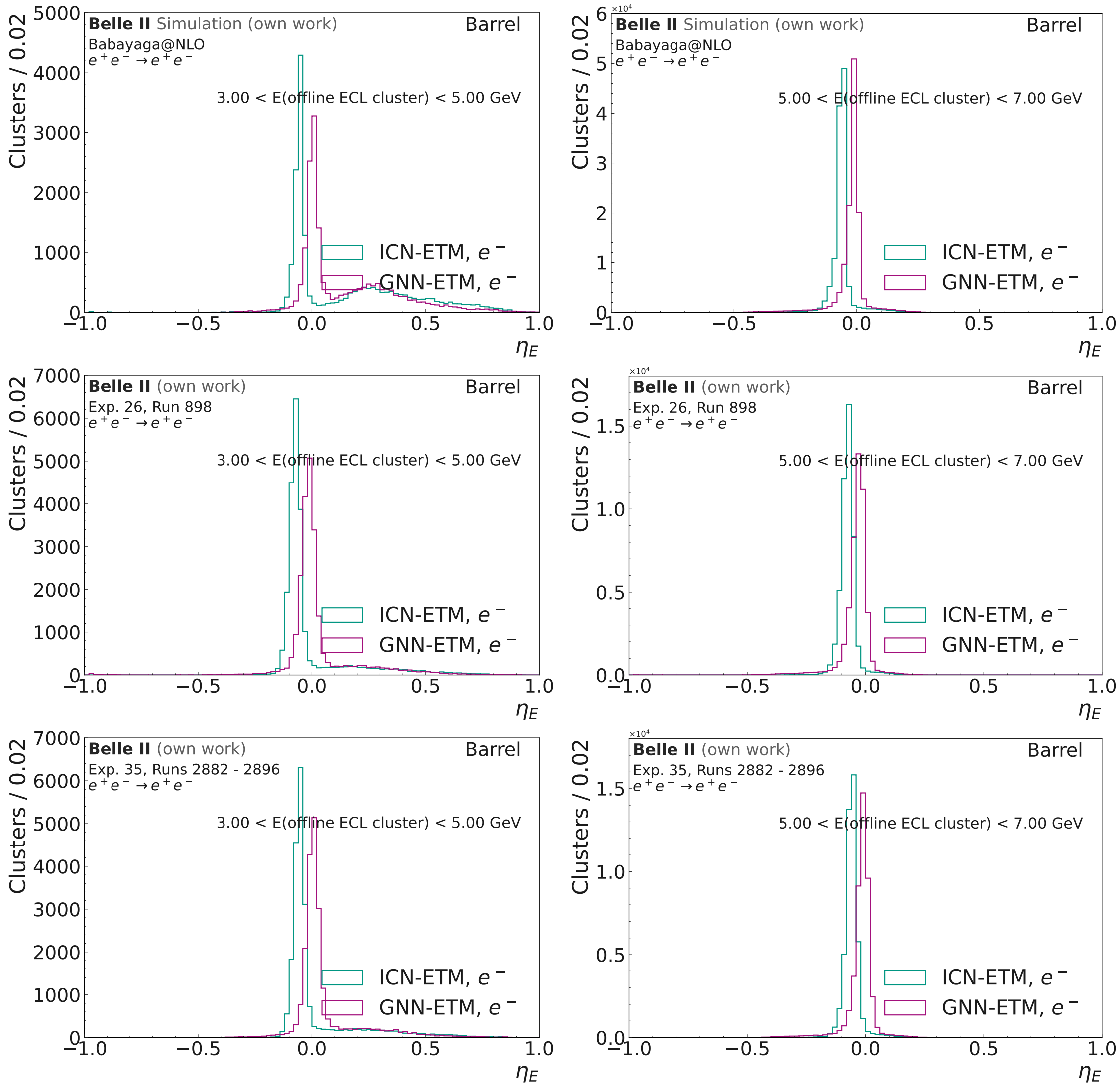


Figure D.1: η_E for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (left) and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

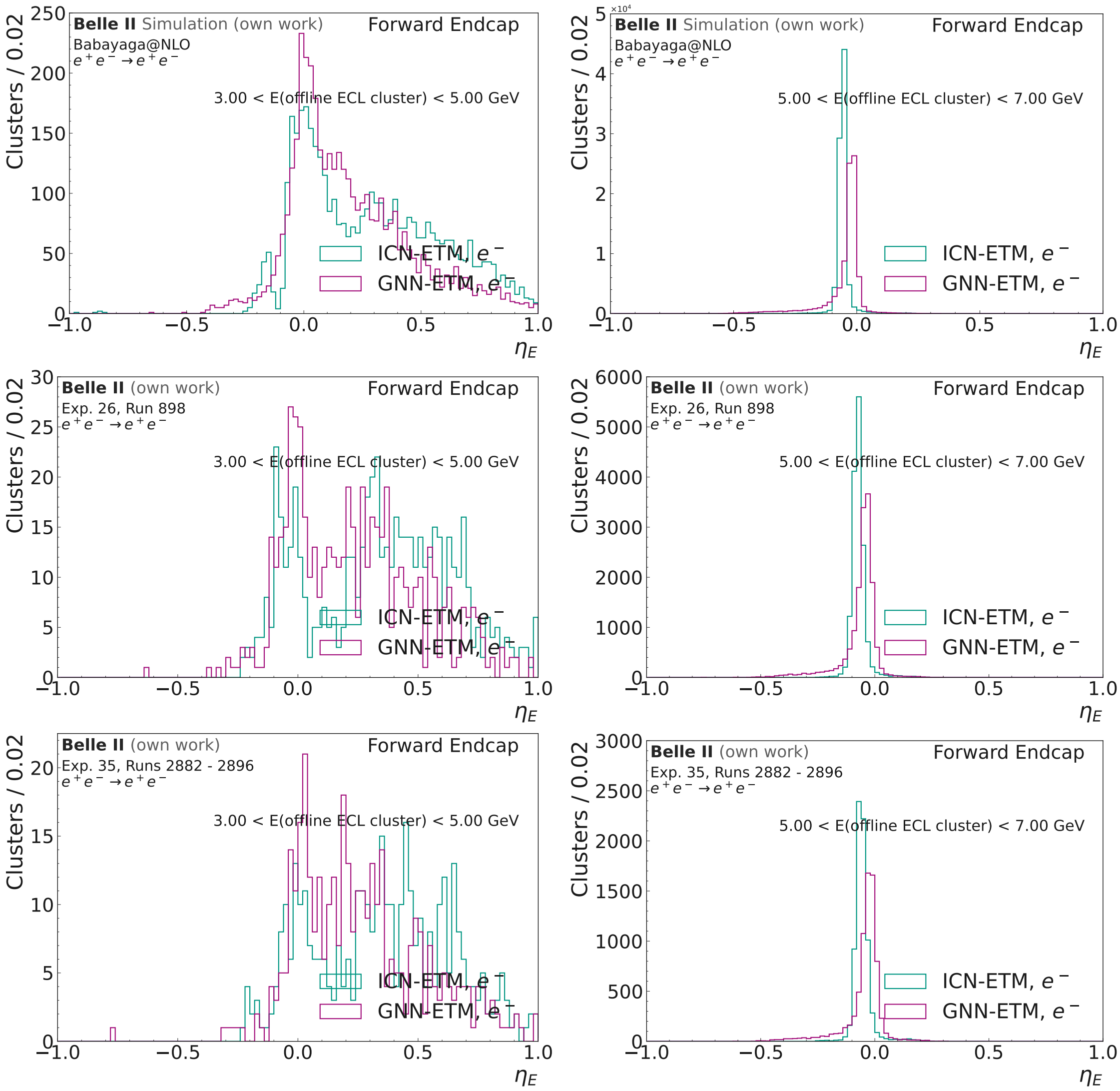


Figure D.2: η_E for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E$(offline ECL cluster) $< 5\,\text{GeV}$ (left) and $5 < E$(offline ECL cluster) $< 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the forward endcap.

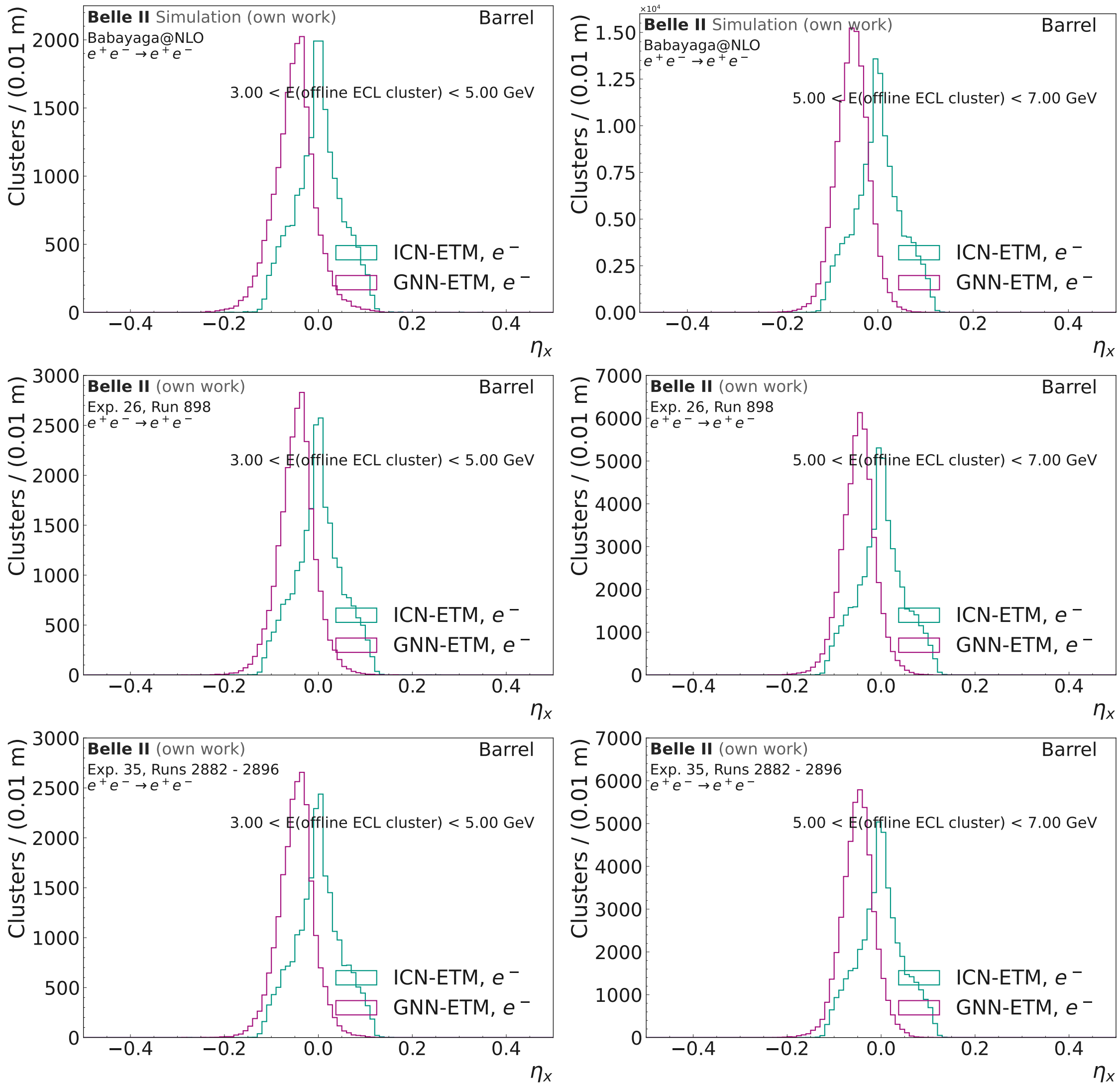


Figure D.3: η_x for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (left) and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

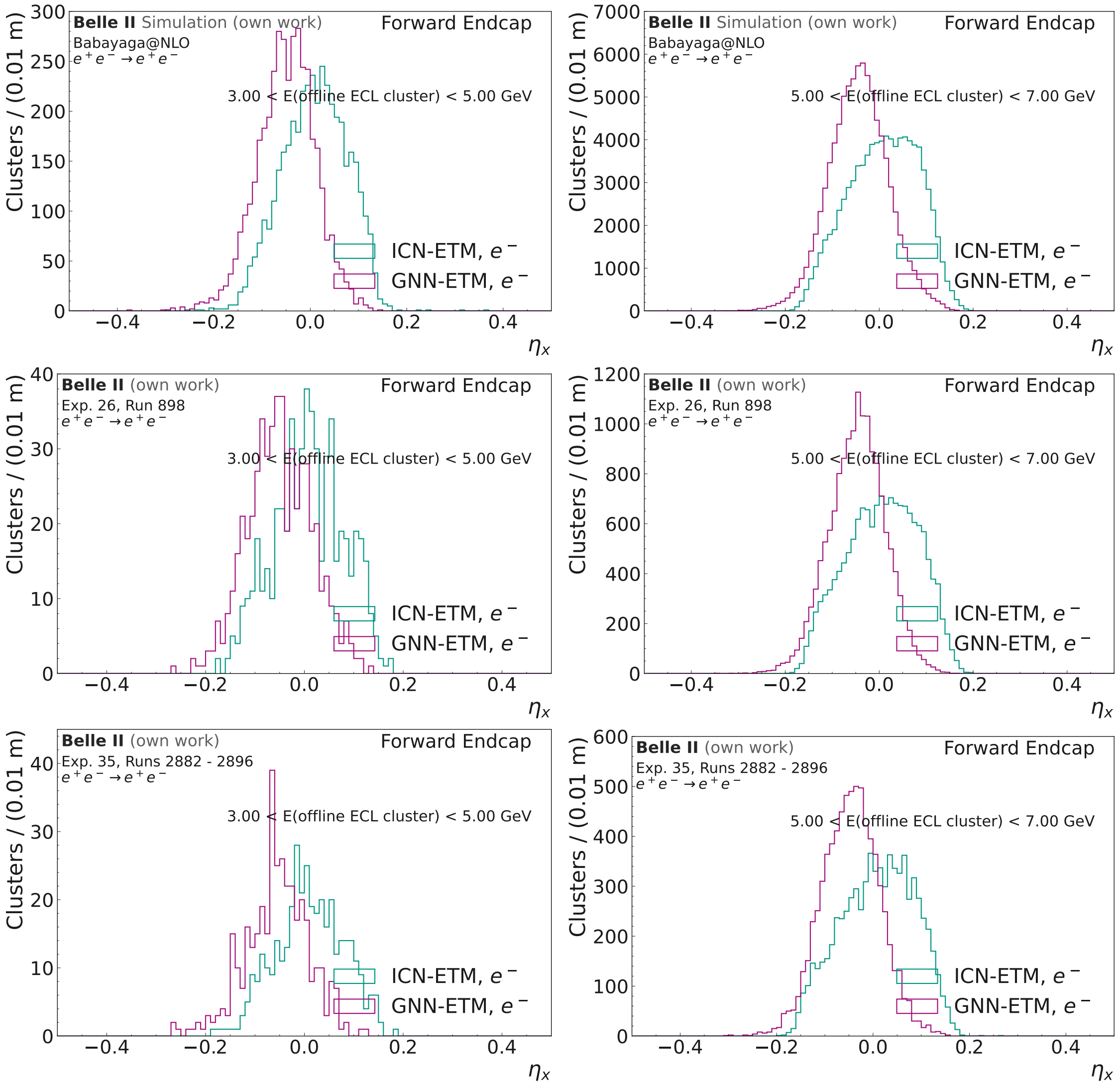


Figure D.4: η_x for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (left) and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

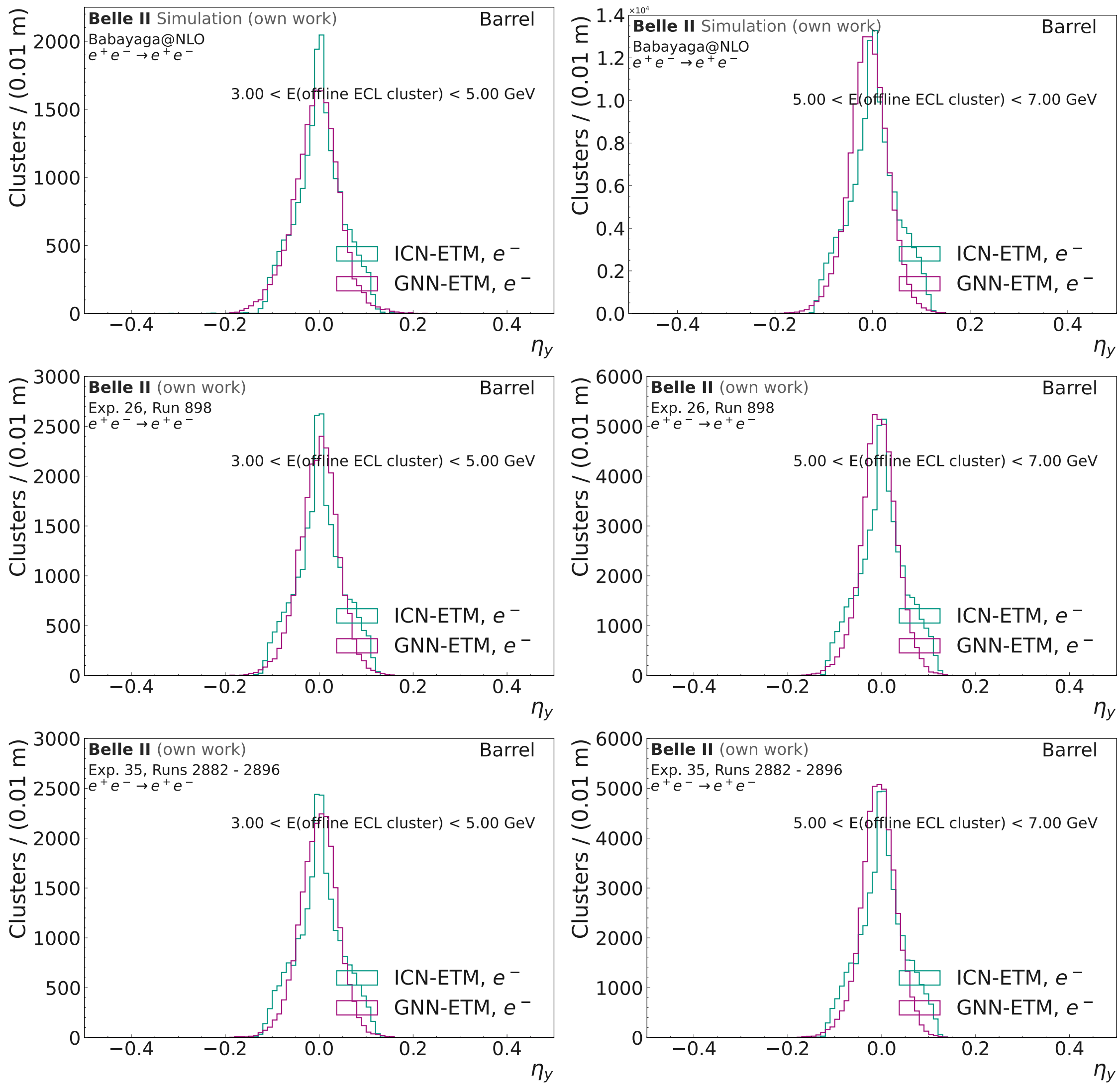


Figure D.5: η_y for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (left) and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

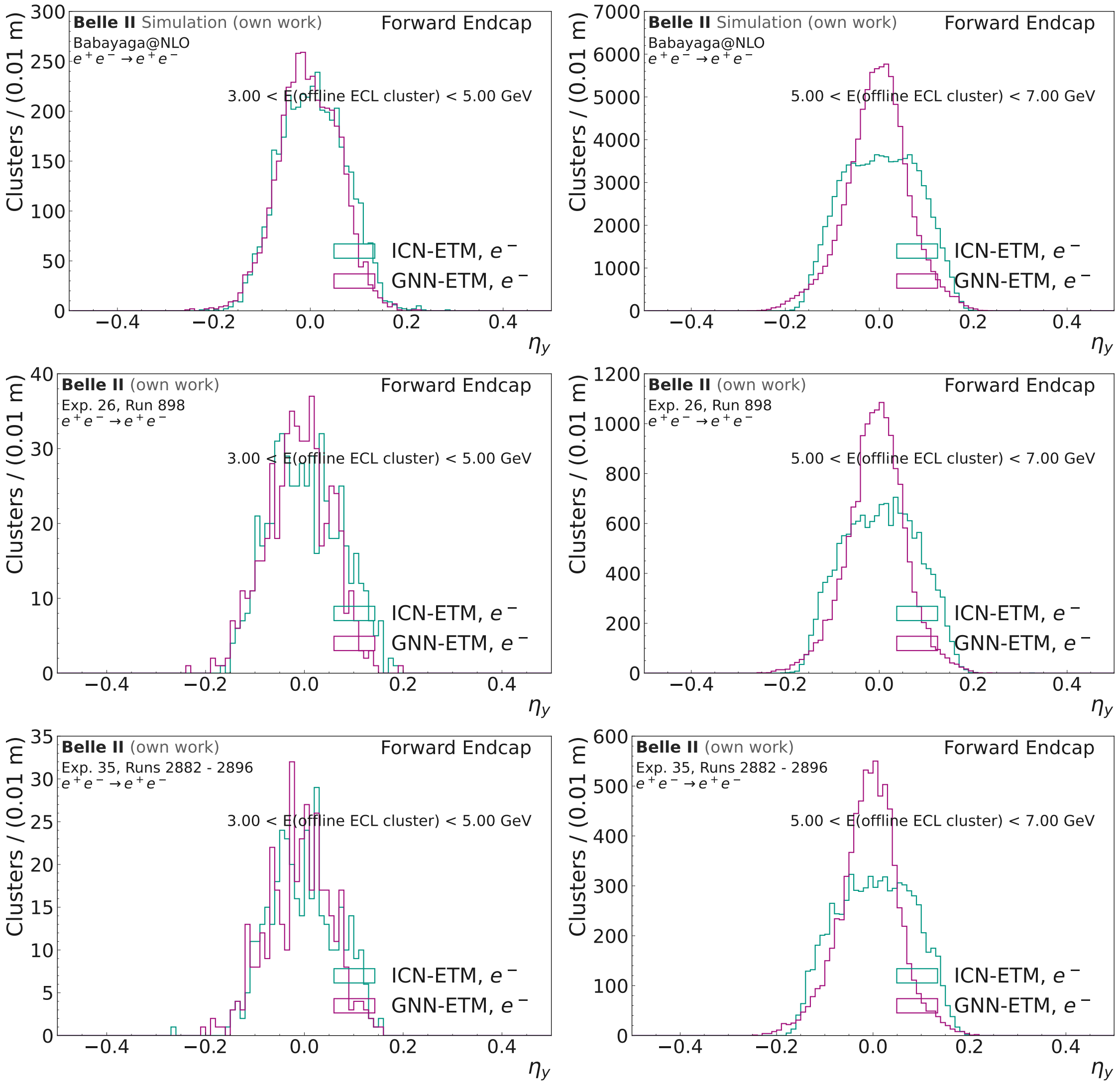


Figure D.6: η_y for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E$(offline ECL cluster) < 5 GeV (left) and $5 < E$(offline ECL cluster) $<$ 7 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

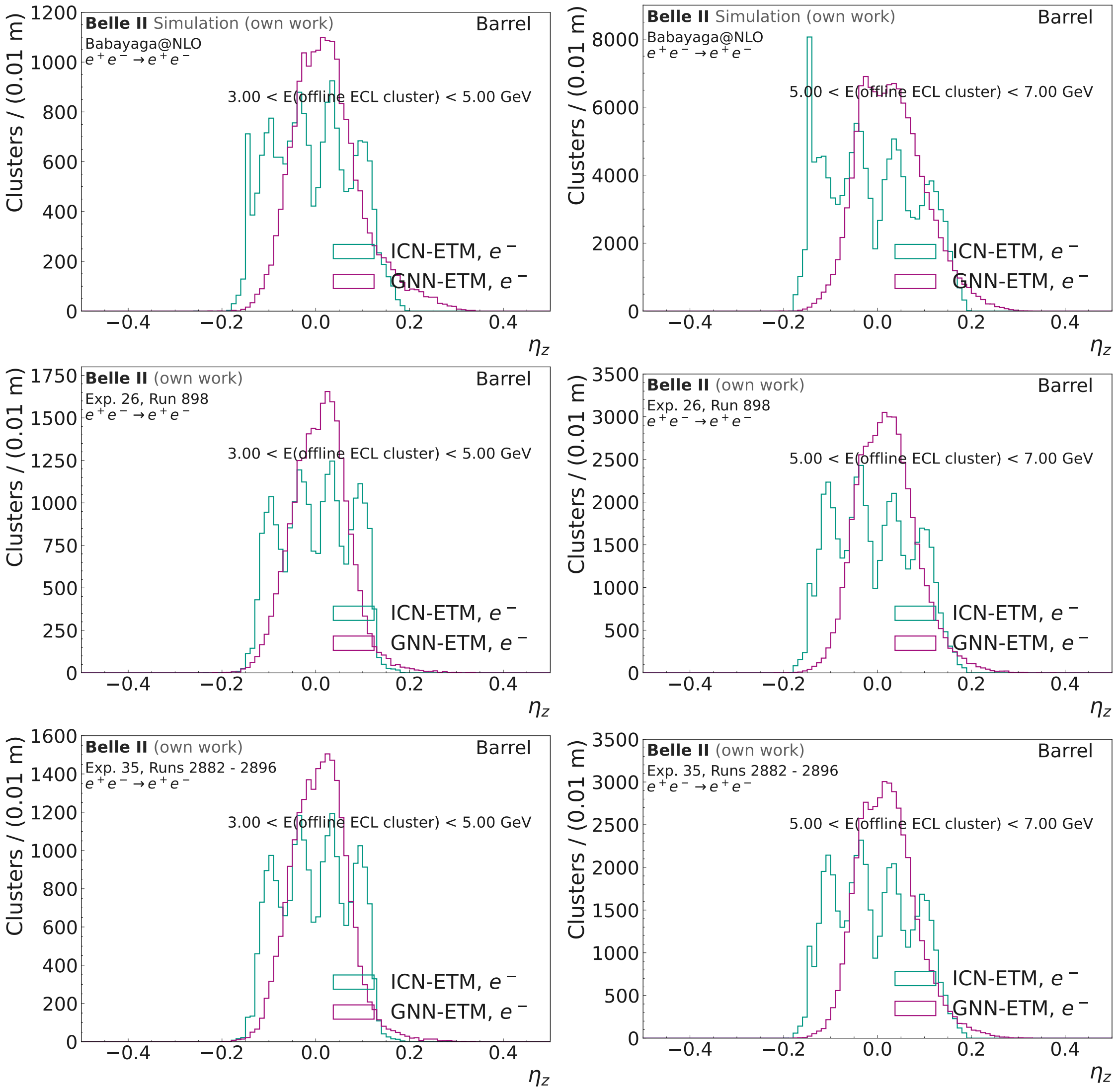


Figure D.7: η_z for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E$(offline ECL cluster) < 5 GeV (left) and $5 < E$(offline ECL cluster) $<$ 7 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

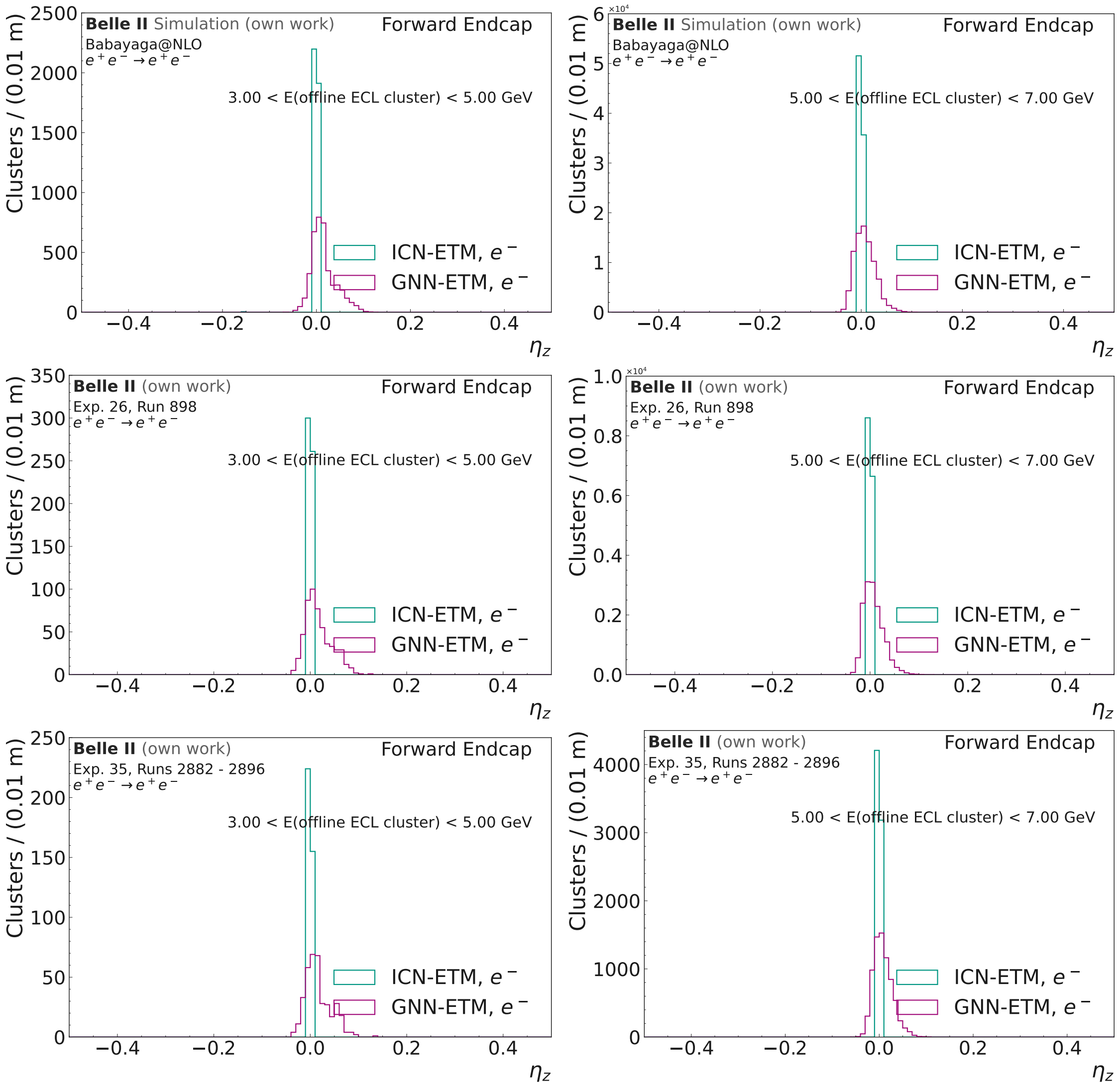


Figure D.8: η_z for GNN-ETM and ICN-ETM clusters matched to the e^- offline ECL cluster for $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (left) and $5 < E(\text{offline ECL cluster}) < 7\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

e^+ Resolutions

This section shows the energy and position resolutions for ICN-ETM and GNN-ETM clusters matched to the e^+ offline ECL cluster from the selection shown in Section 8.2.3. The energy resolution η_E is shown for the barrel region and the backward endcap in Fig. D.10 and Fig. D.9, respectively. The position resolutions η_x, η_y, and η_z are also shown for the barrel region and the backward endcap in Fig. D.11, Fig. D.12, Fig. D.13, Fig. D.14,Fig. D.15, and Fig. D.16.

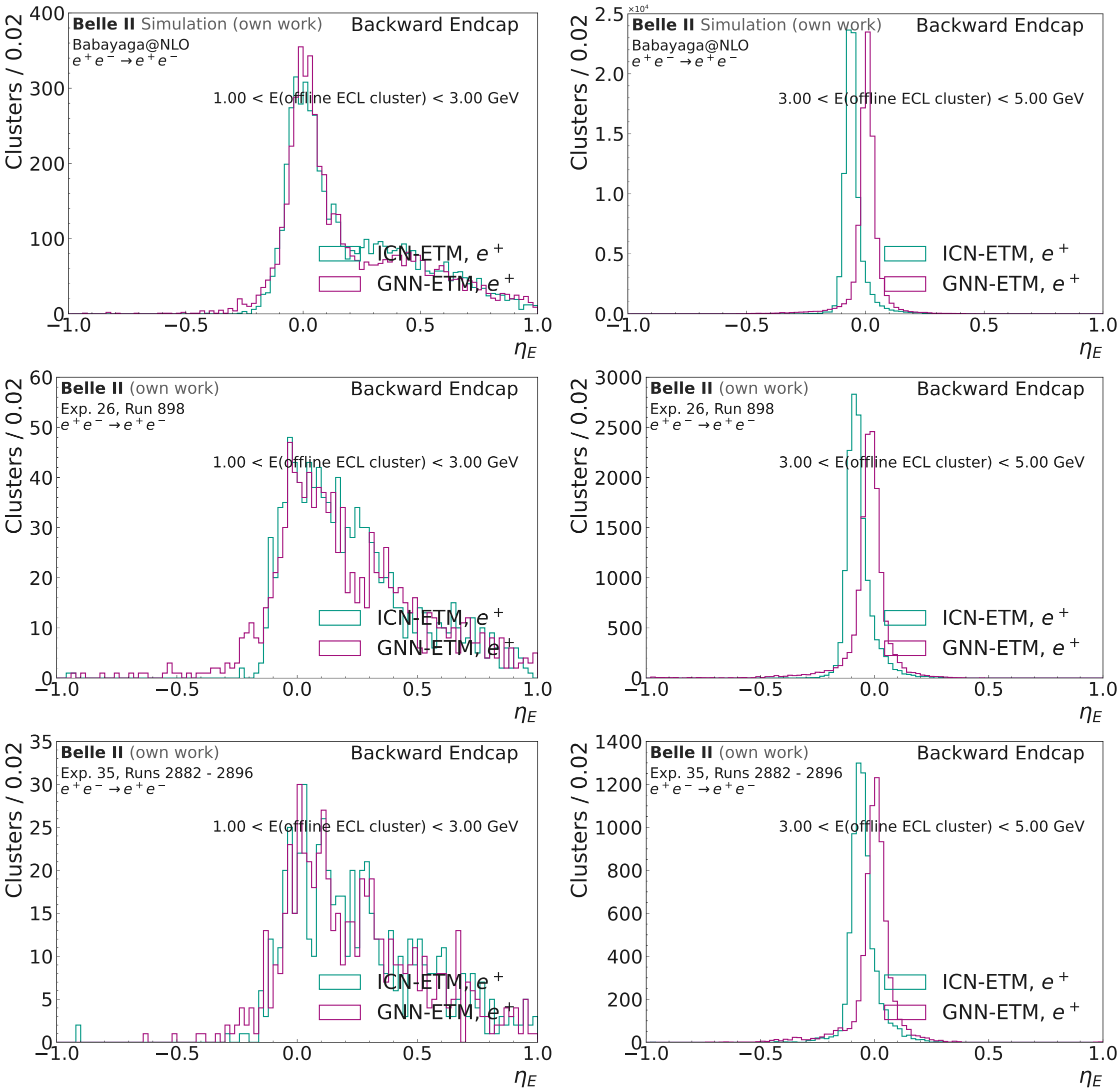


Figure D.9: η_E for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E$(offline ECL cluster) < 3 GeV (left) and $3 < E$(offline ECL cluster) $<$ 5 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

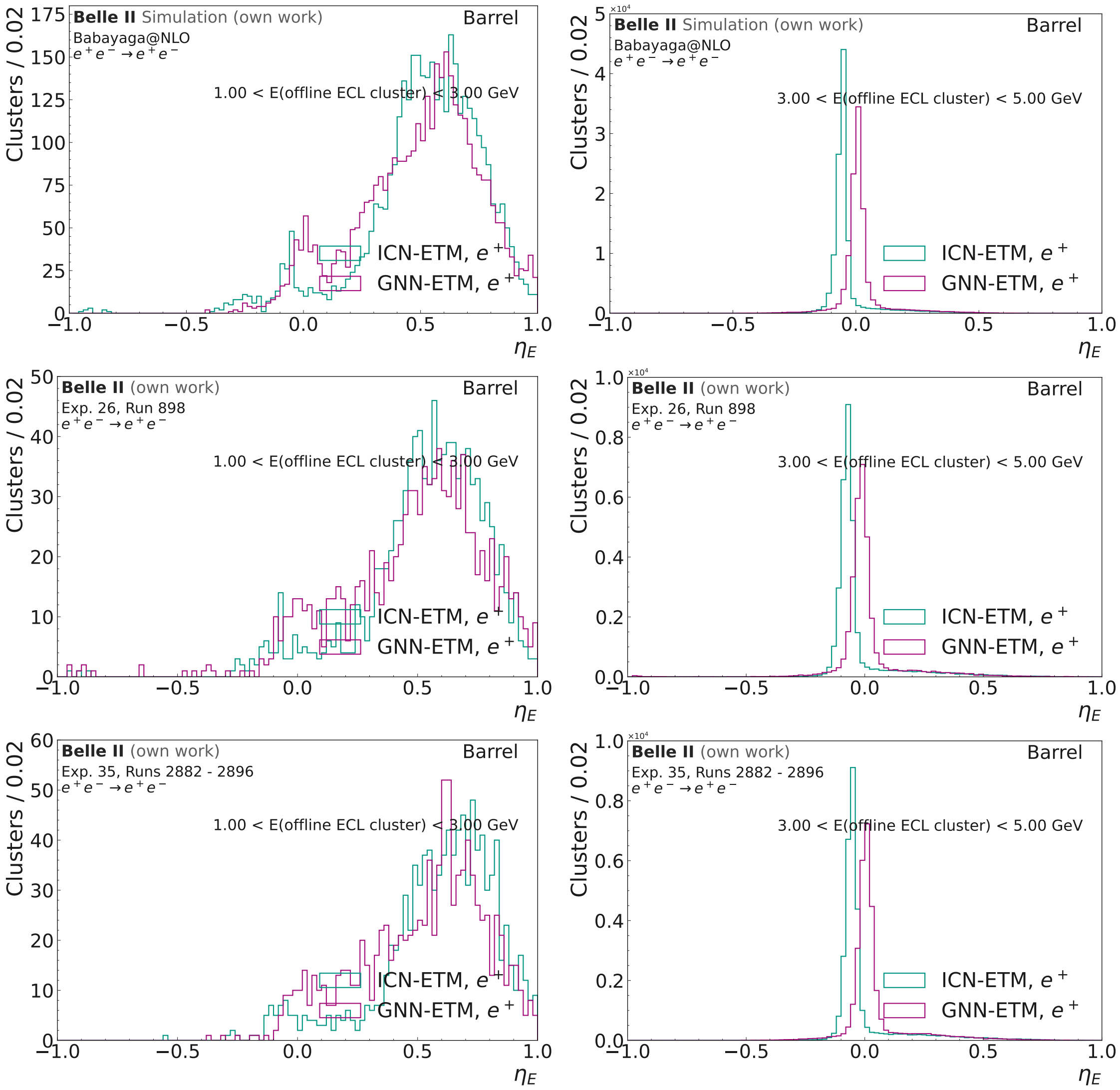


Figure D.10: η_E for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

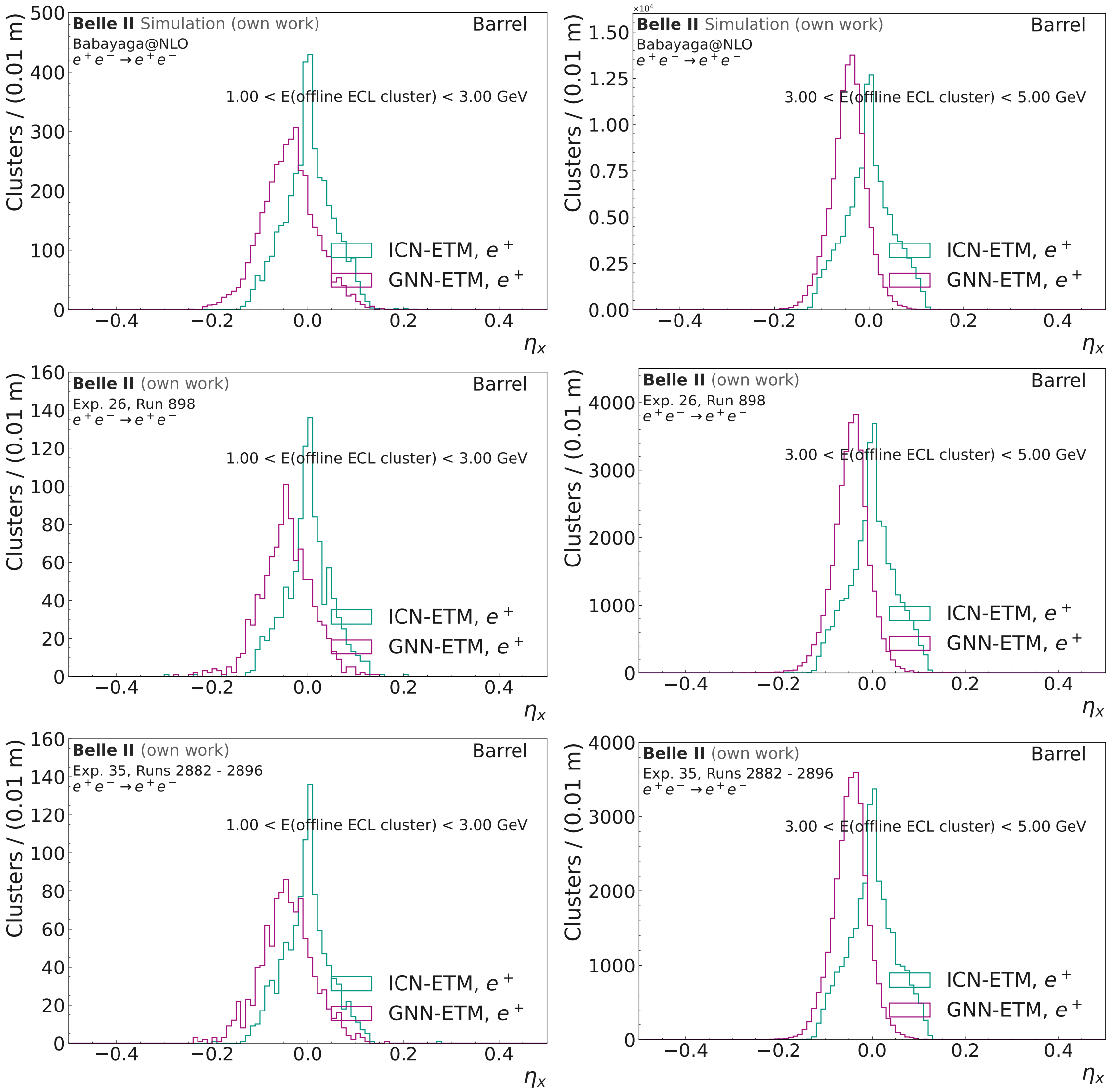


Figure D.11: η_x for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

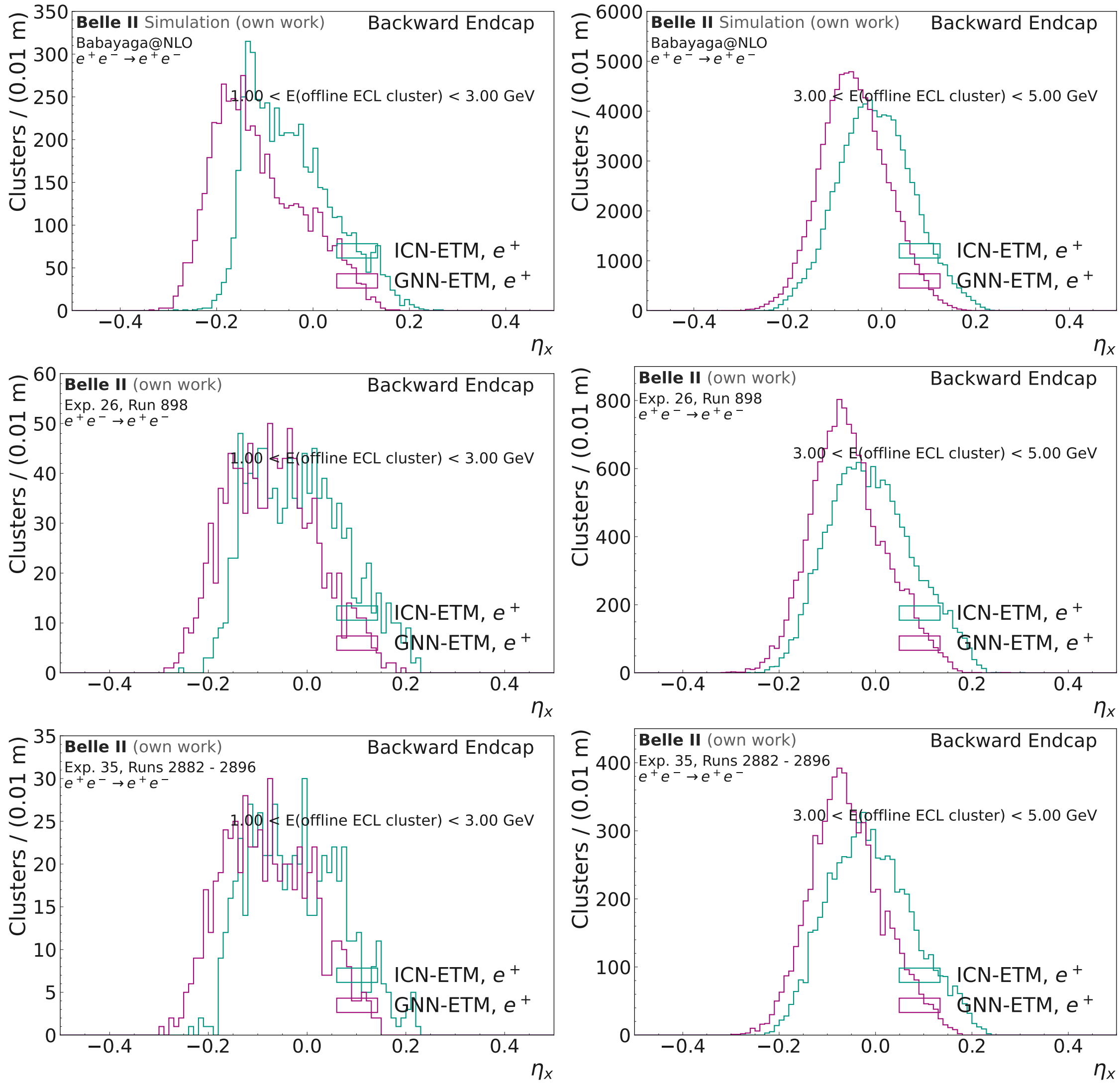


Figure D.12: η_x for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

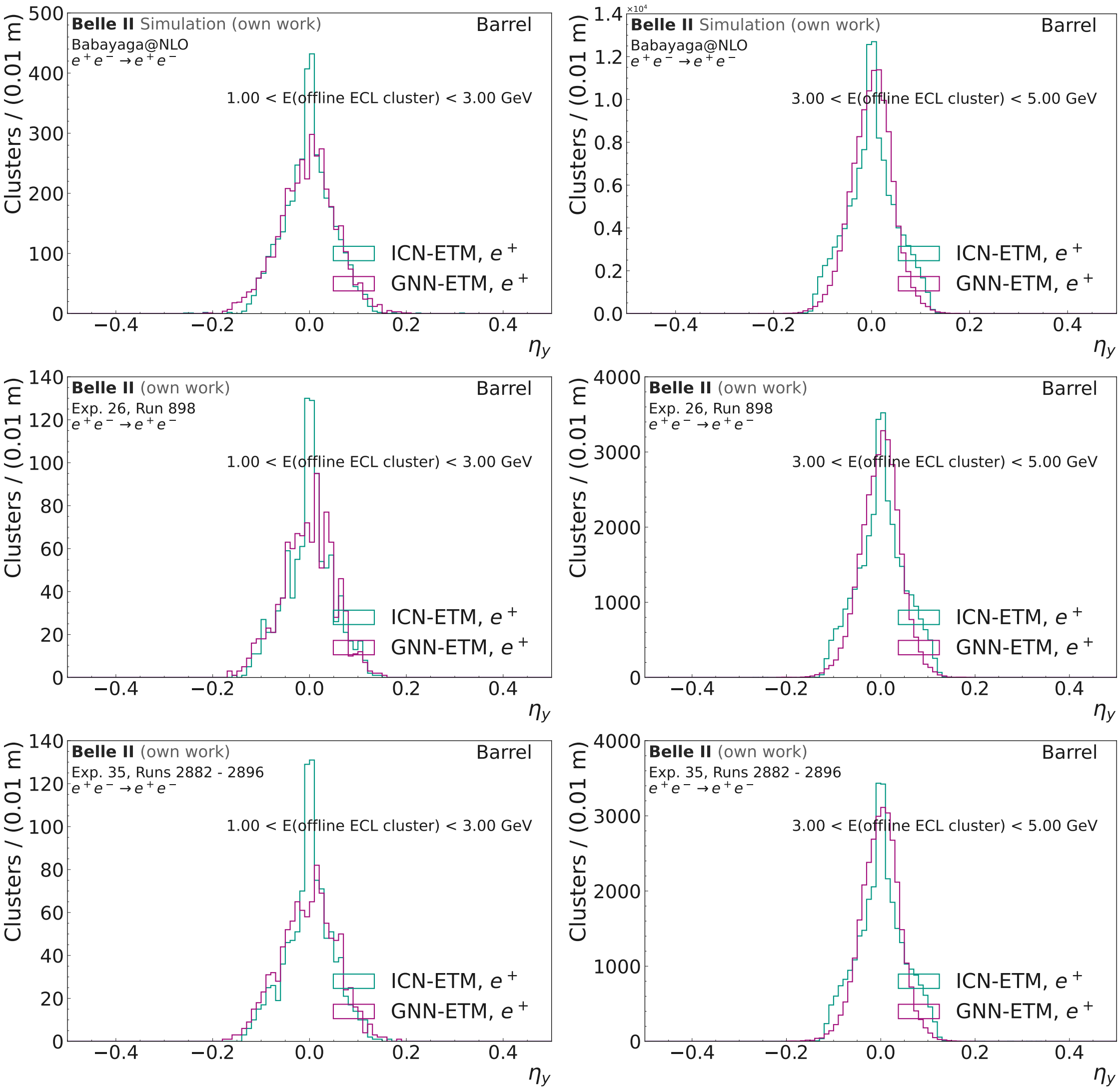


Figure D.13: η_y for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

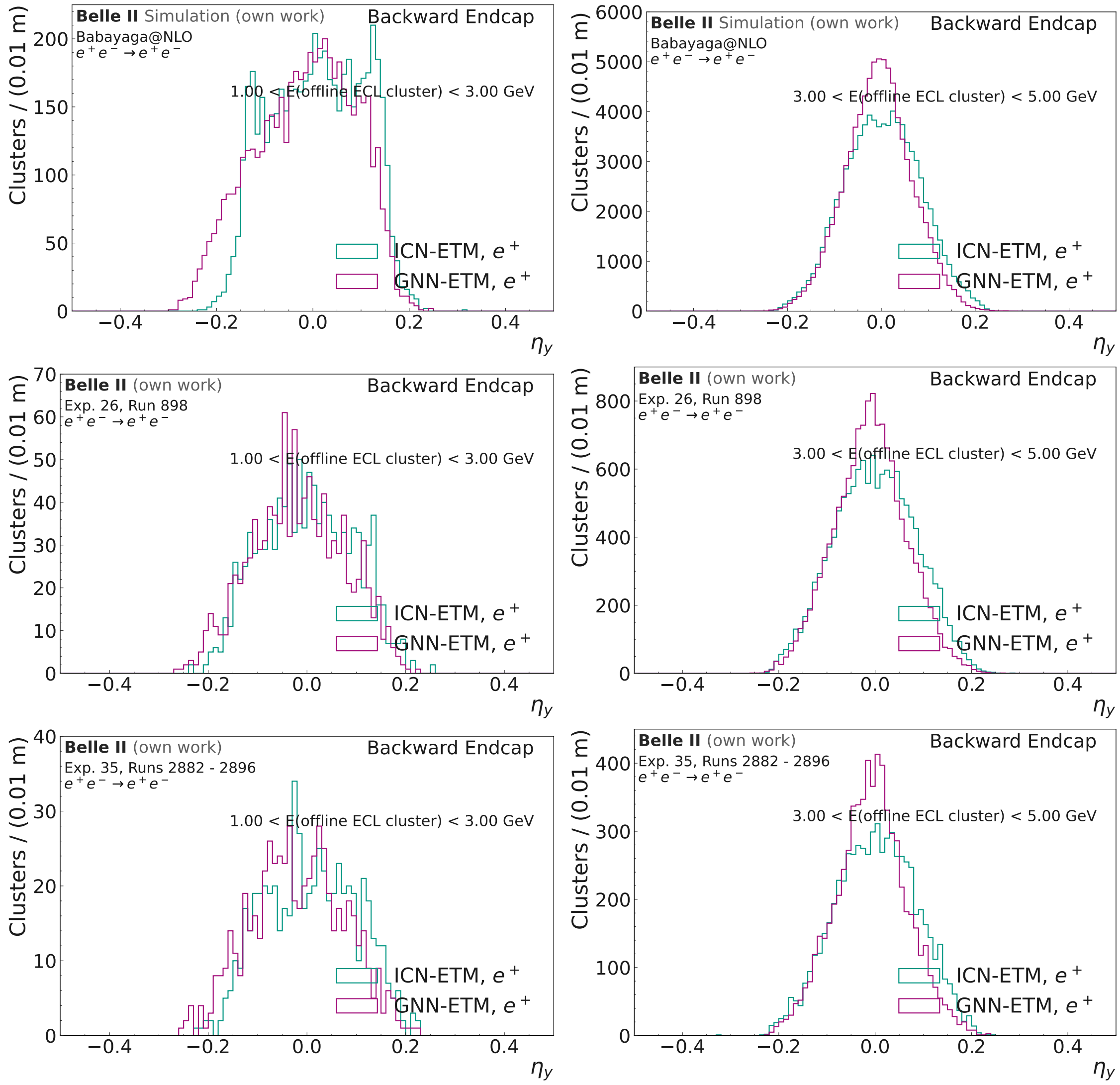


Figure D.14: η_y for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

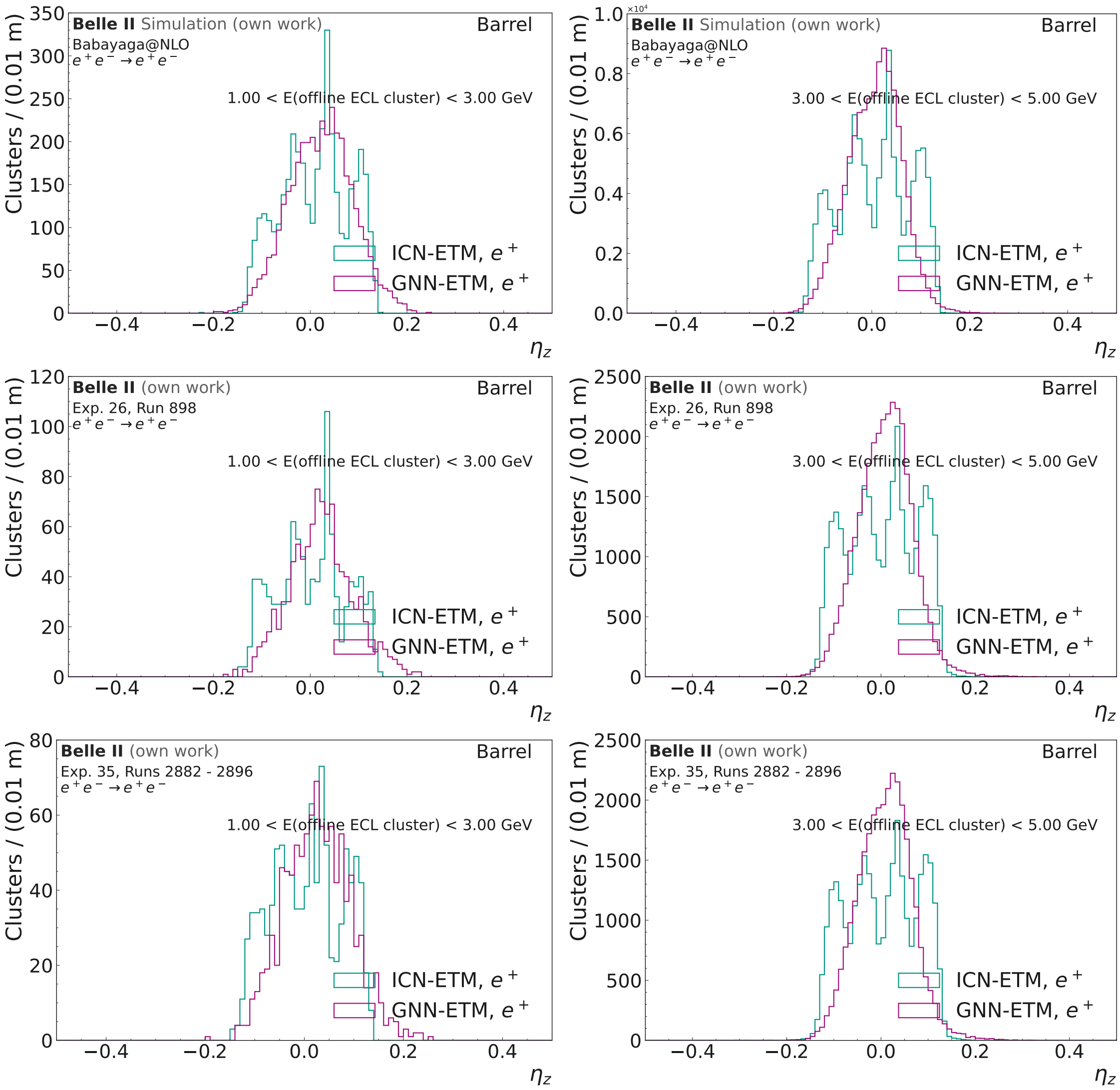


Figure D.15: η_z for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

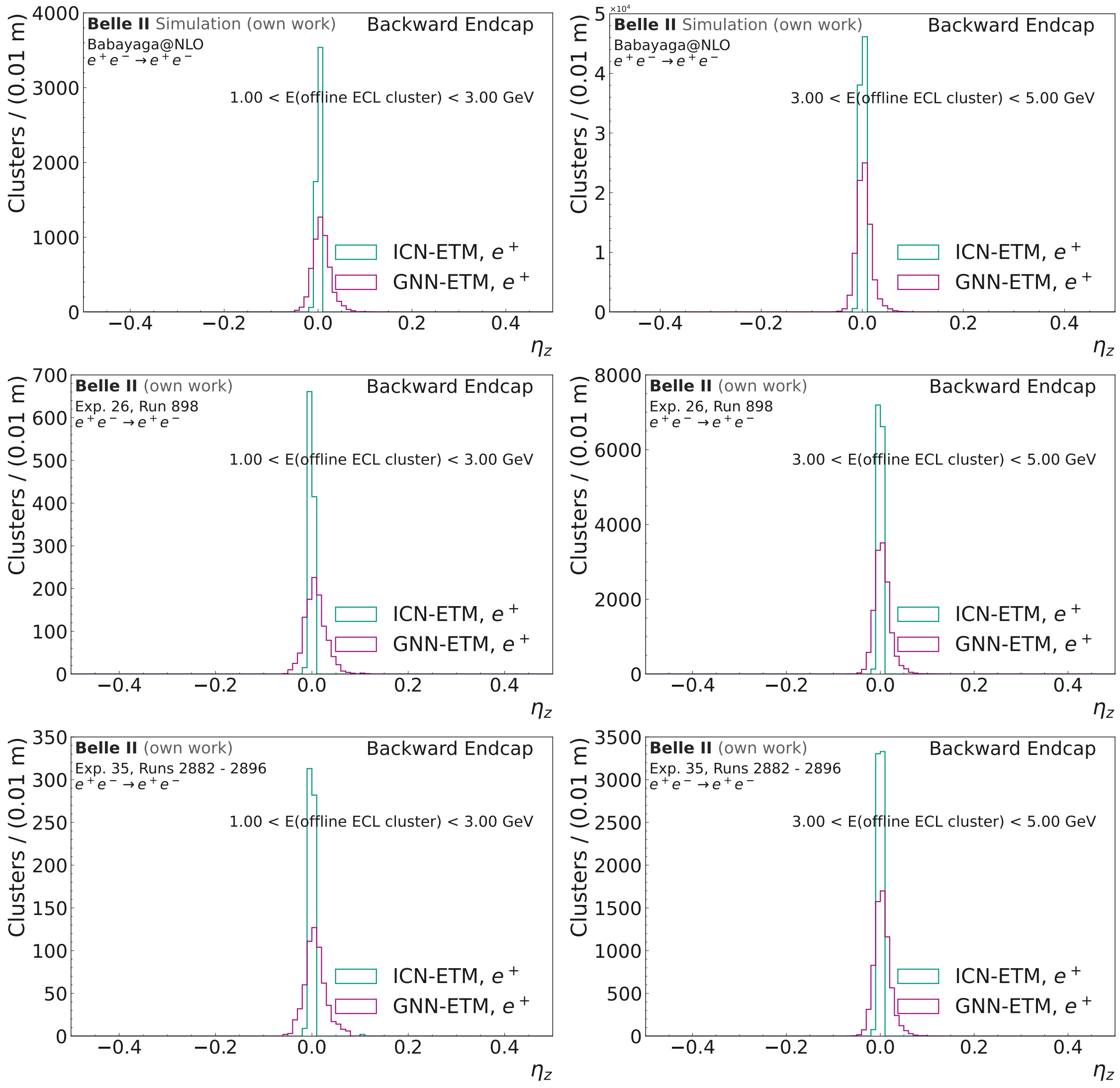


Figure D.16: η_z for GNN-ETM and ICN-ETM clusters matched to the e^+ offline ECL cluster for $1 < E(\text{offline ECL cluster}) < 3\,\text{GeV}$ (left) and $3 < E(\text{offline ECL cluster}) < 5\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

D.2 GNN-ETM results for $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$

D.2.1 Energy and Position Resolutions

μ^- Resolutions

This section shows the energy and position resolutions for ICN-ETM and GNN-ETM clusters matched to the μ^- offline ECL cluster from the selection shown in **??**. The energy resolution η_E is shown for the barrel region and the forward endcap in Fig. D.17 and Fig. D.18, respectively. The position resolutions η_x, η_y, and η_z are also shown for the barrel region and the forward endcap in Fig. D.19, Fig. D.20, Fig. D.21, Fig. D.22,Fig. D.23, and Fig. D.24.

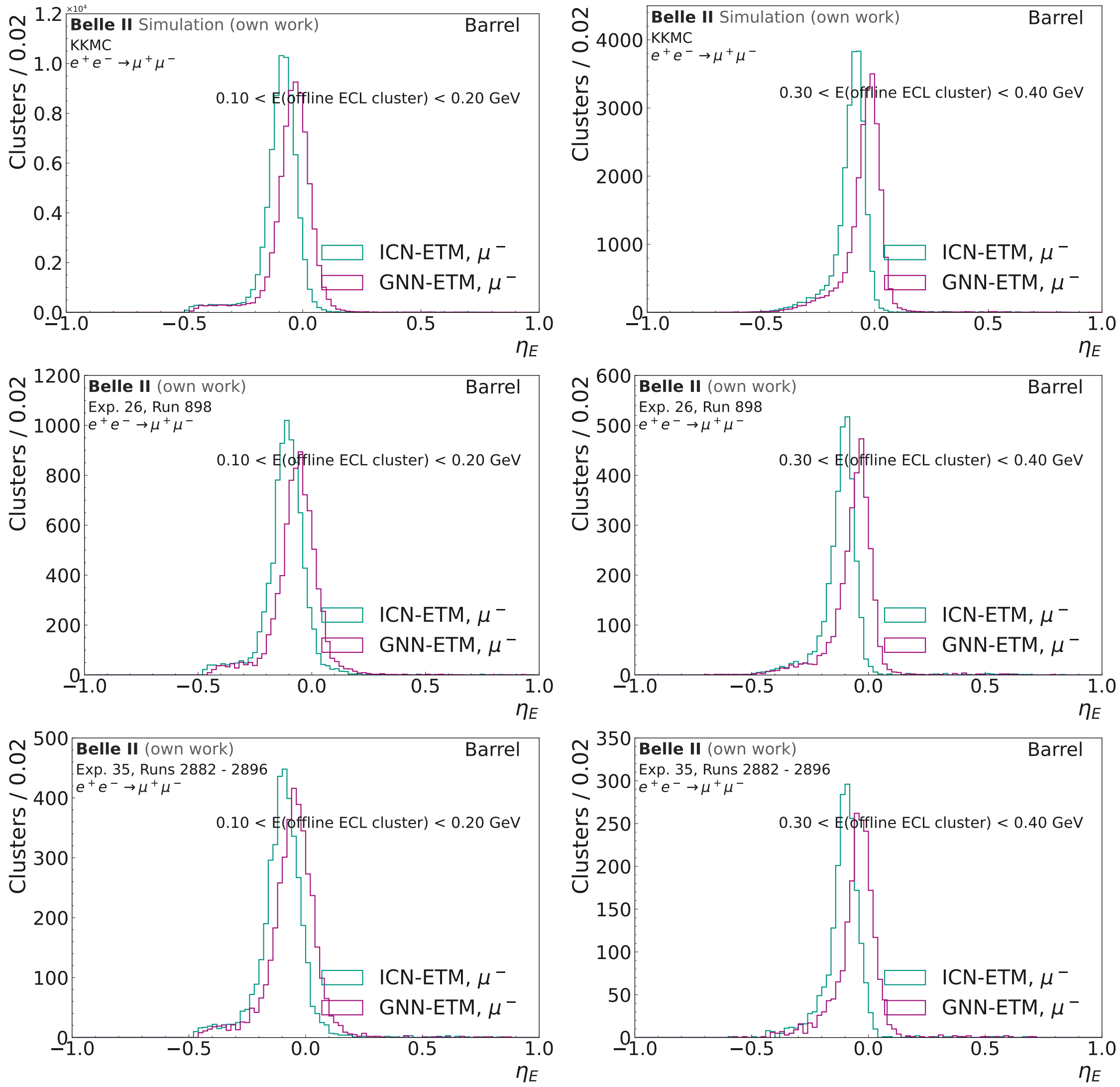


Figure D.17: η_E for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

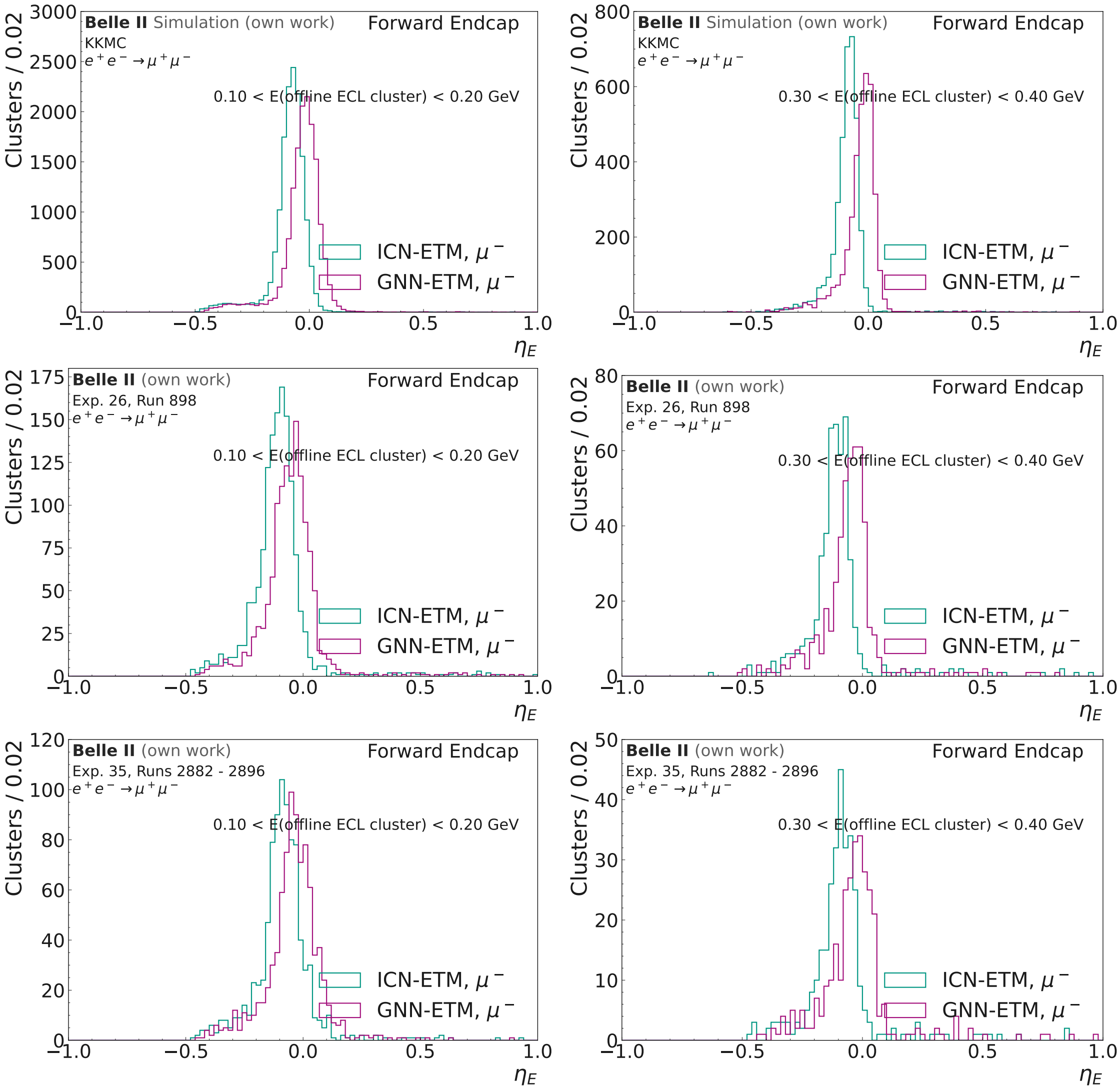


Figure D.18: η_E for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the forward endcap.

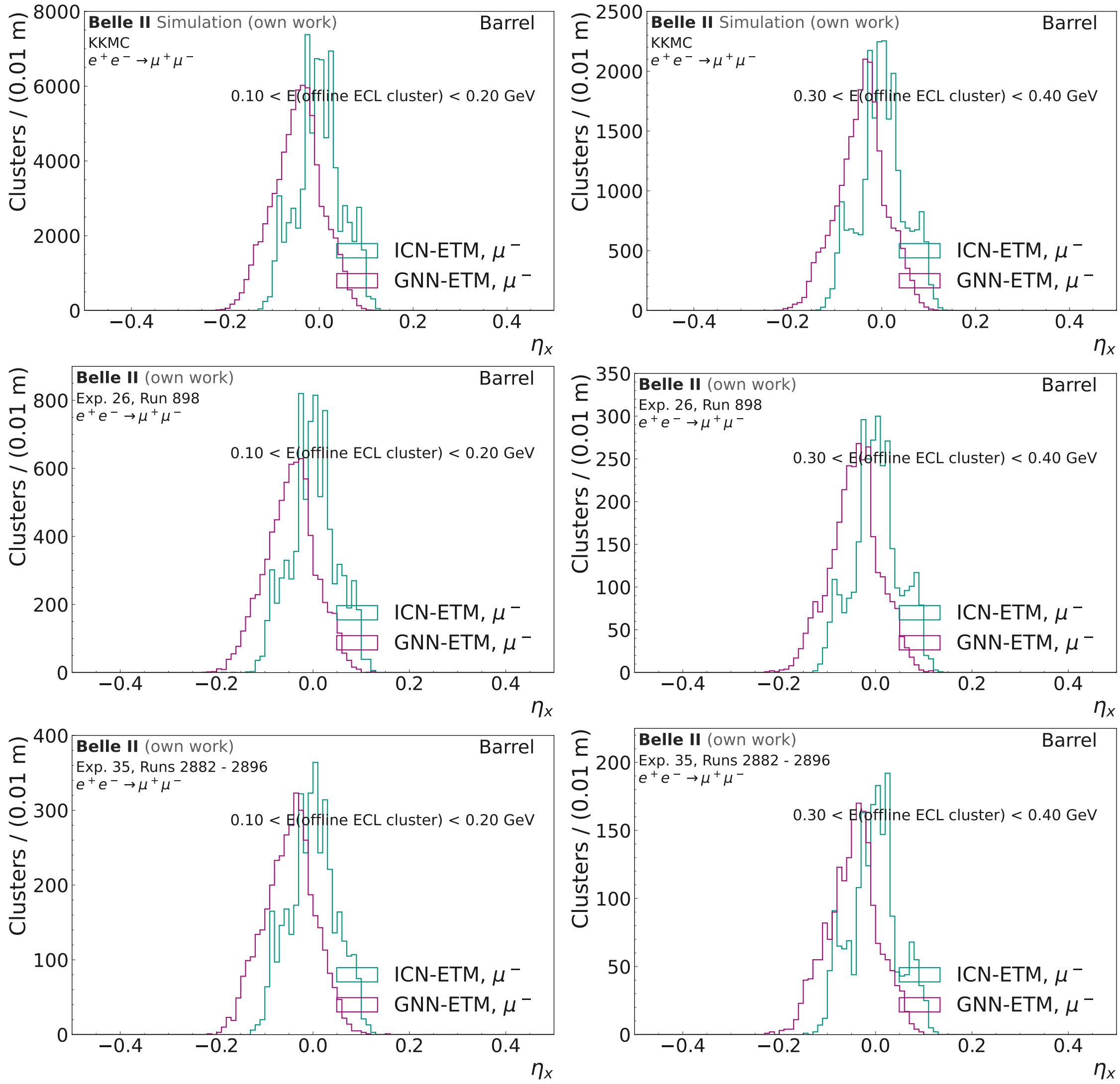


Figure D.19: η_x for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

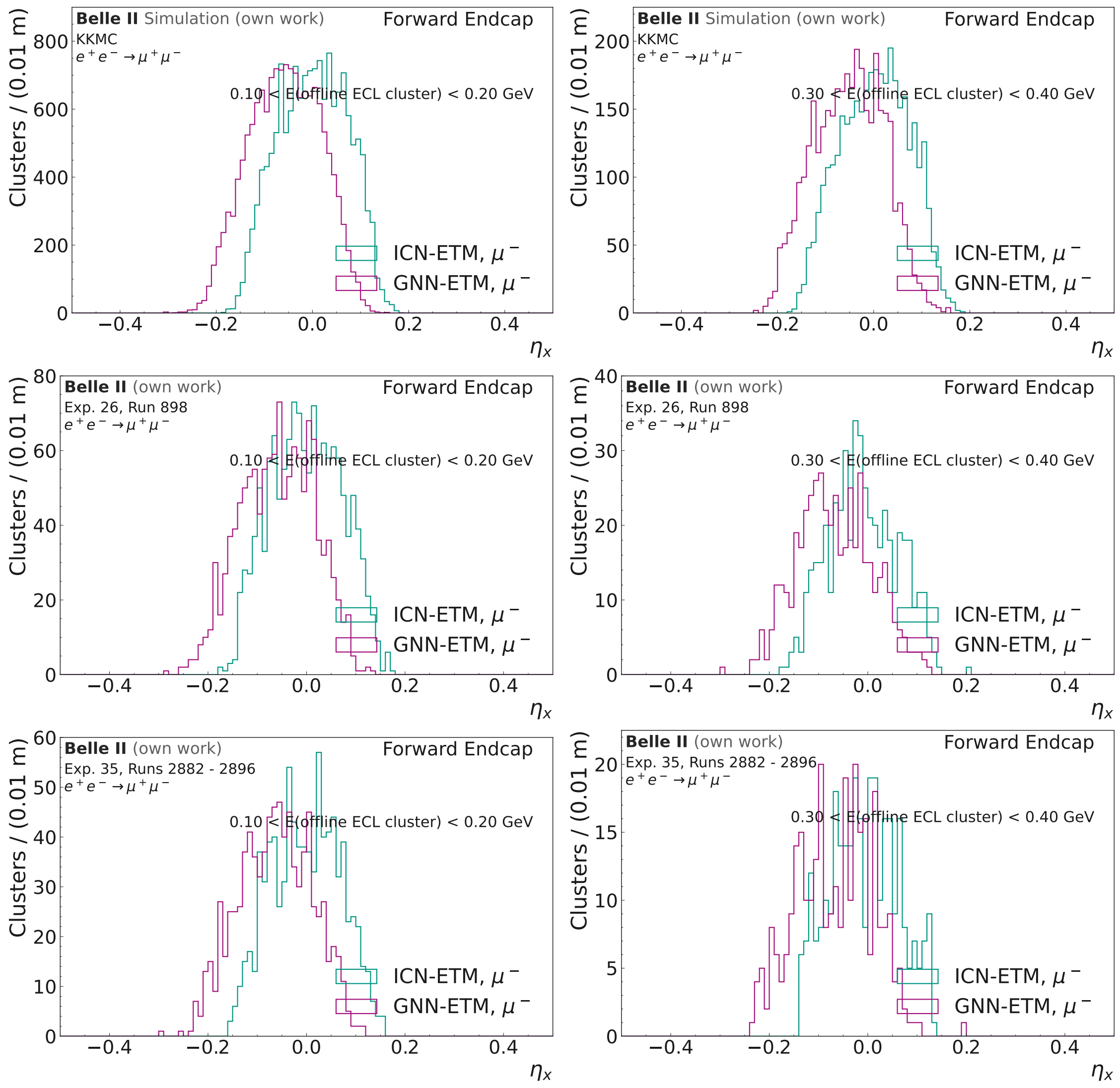


Figure D.20: η_x for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

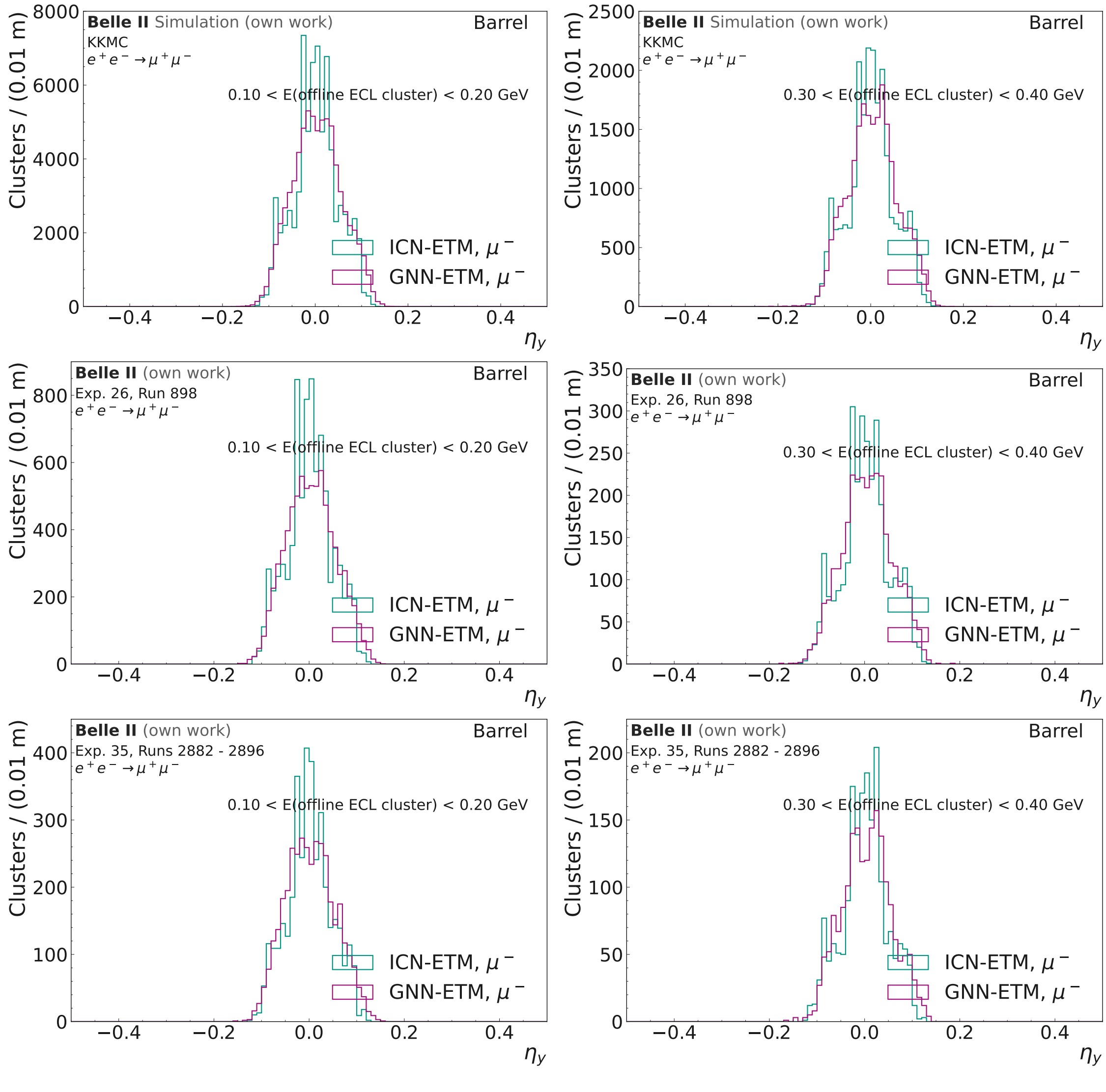


Figure D.21: η_y for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

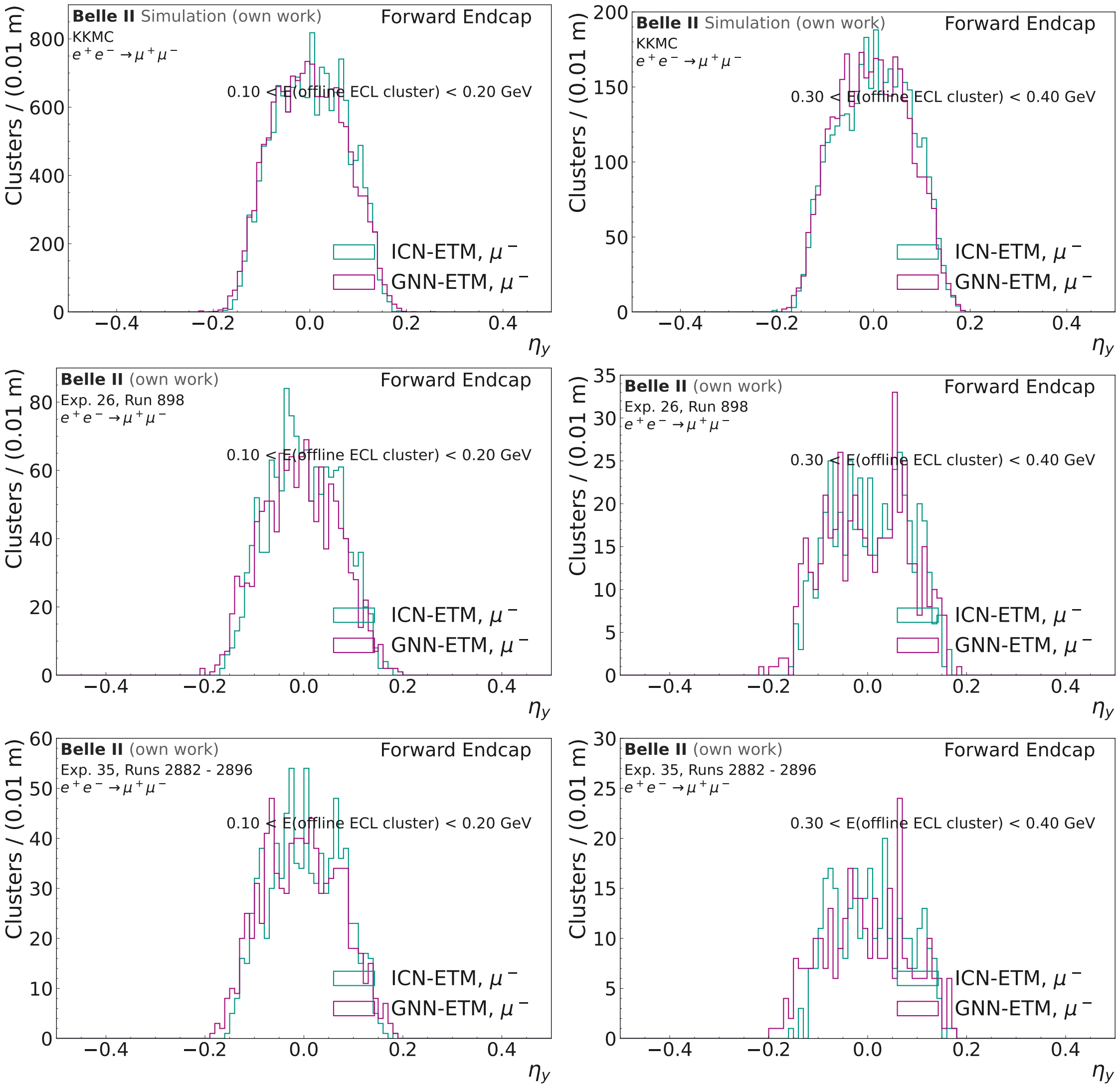


Figure D.22: η_y for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

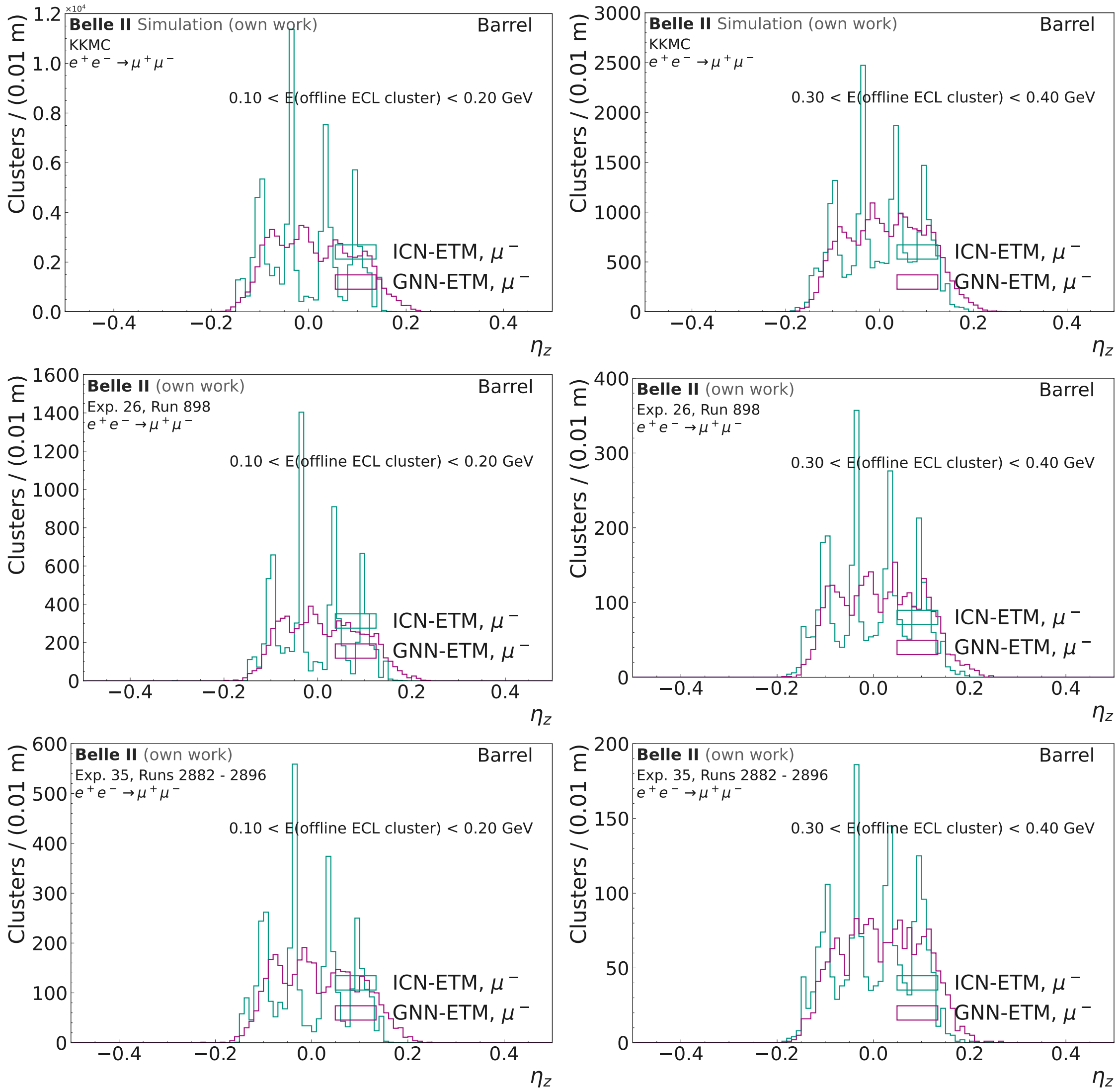


Figure D.23: η_z for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

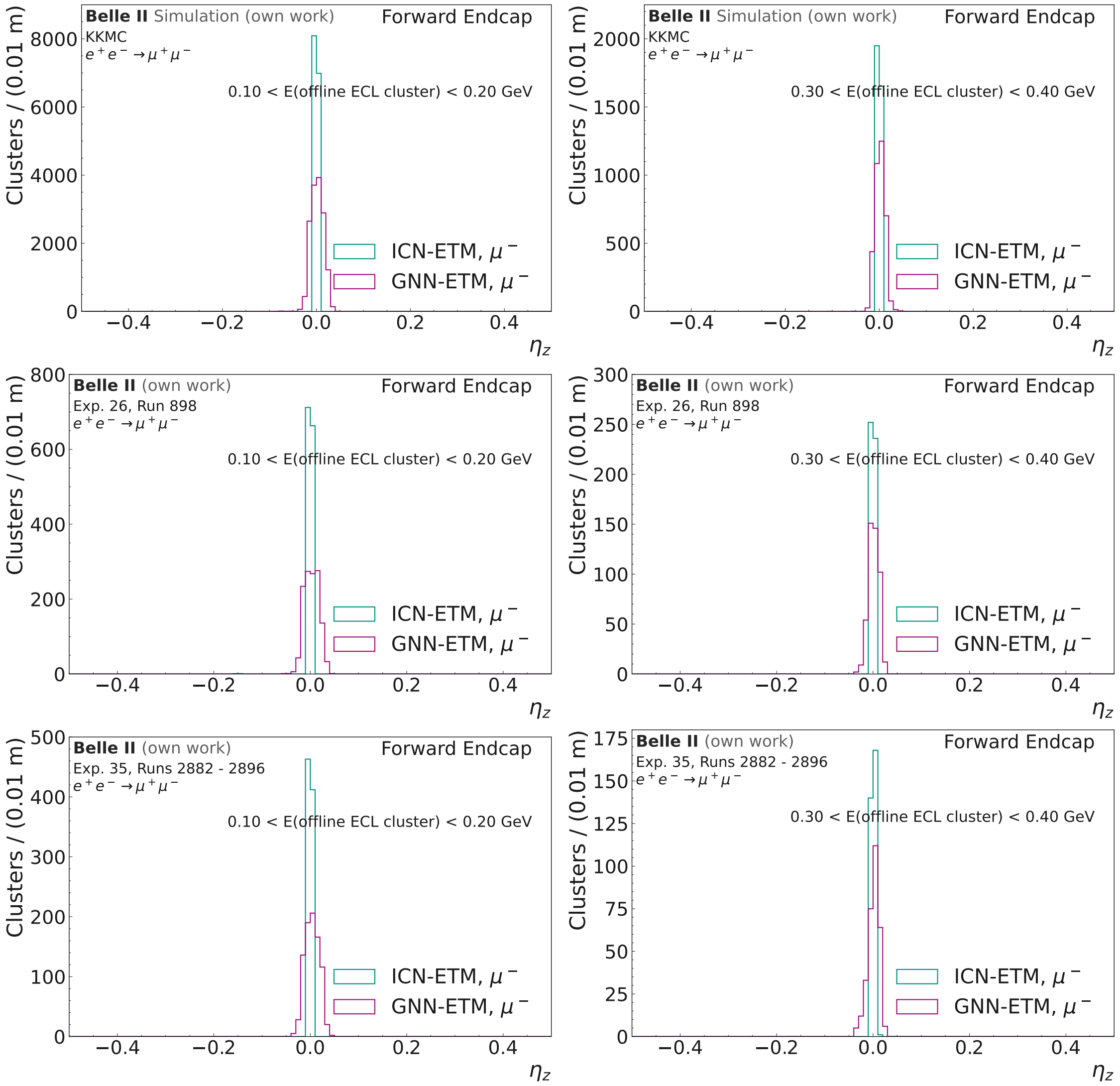


Figure D.24: η_z for GNN-ETM and ICN-ETM clusters matched to the mu^- offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

μ^+ Resolutions

This section shows the energy and position resolutions for ICN-ETM and GNN-ETM clusters matched to the μ^+ offline ECL cluster from the selection shown in **??**. The energy resolution η_E is shown for the barrel region and the backward endcap in Fig. D.26 and Fig. D.25, respectively. The position resolutions η_x, η_y, and η_z are also shown for the barrel region and the backward endcap in Fig. D.27, Fig. D.28, Fig. D.29, Fig. D.30,Fig. D.31, and Fig. D.32.

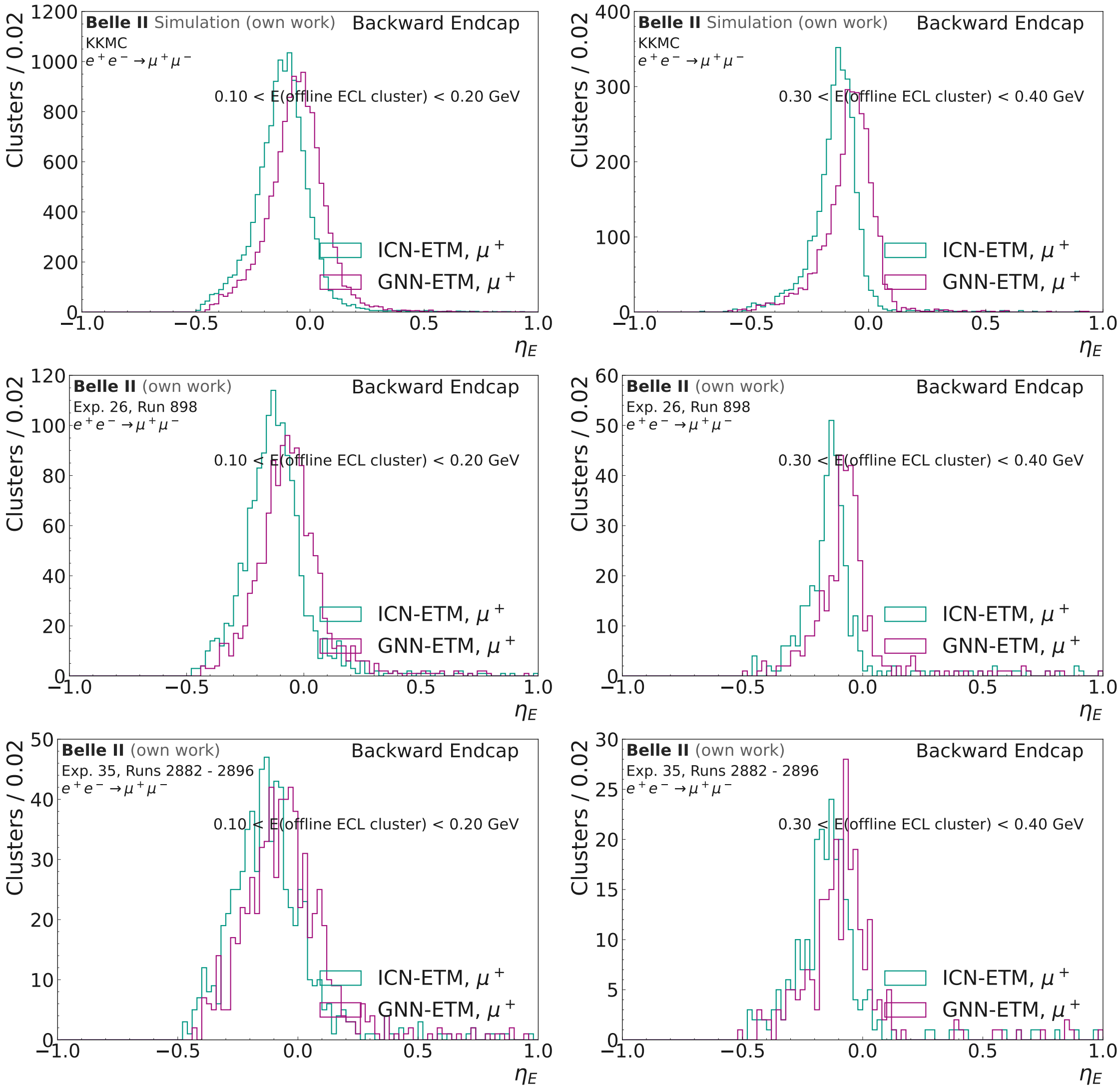


Figure D.25: η_E for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

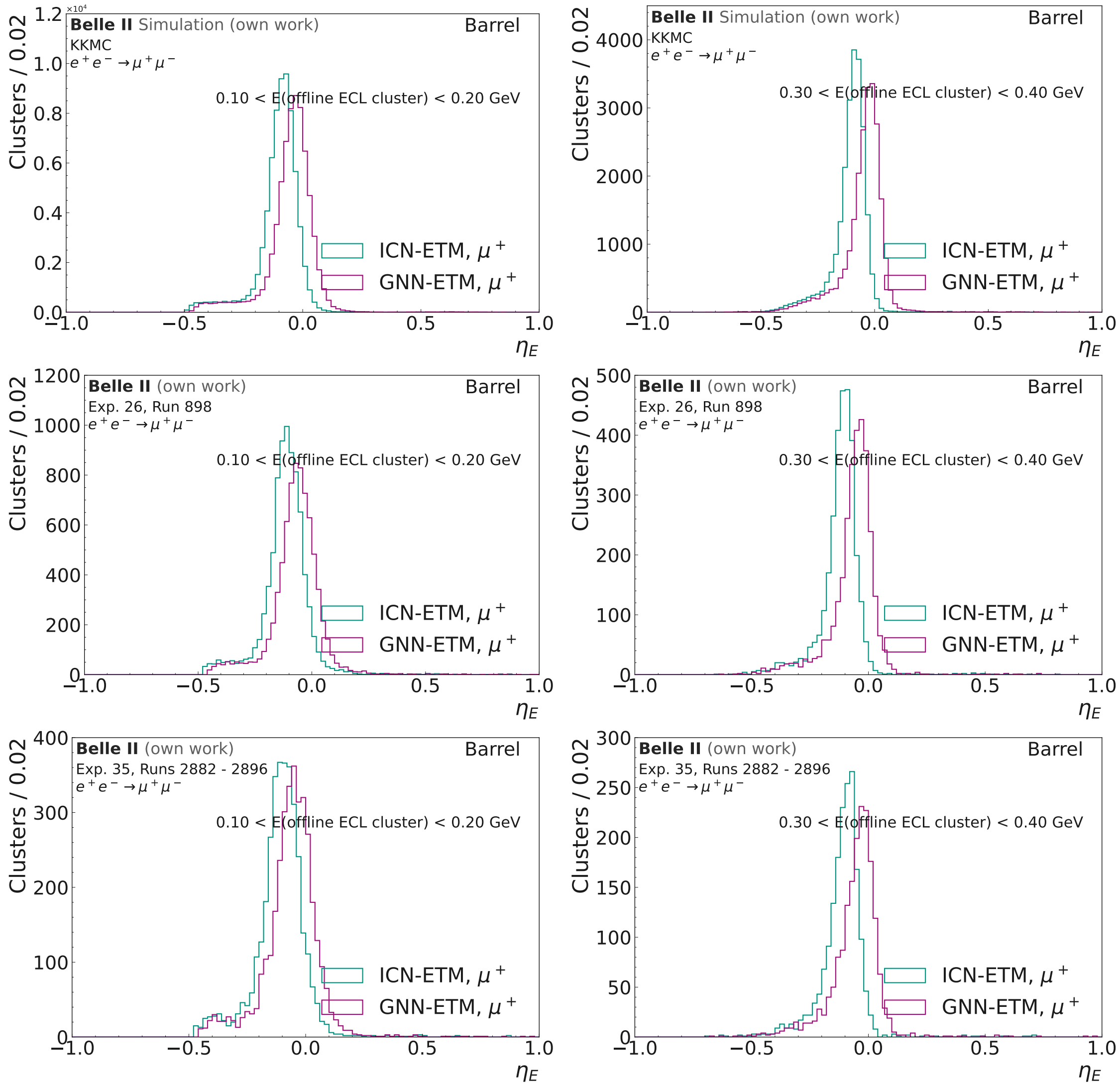


Figure D.26: η_E for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster})$ $< 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

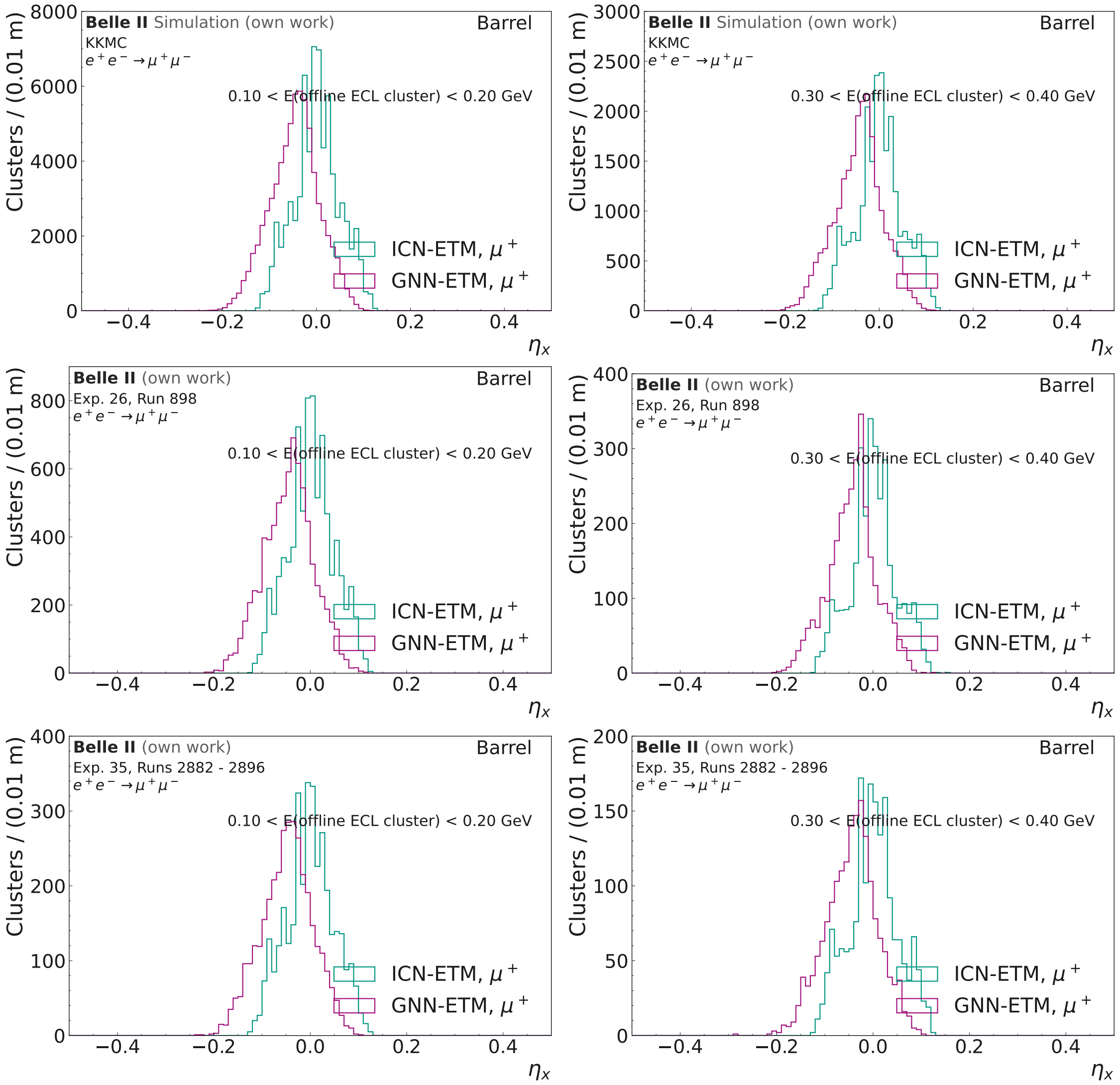


Figure D.27: η_x for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E$(offline ECL cluster) $< 0.2\,\text{GeV}$ (left) and $0.3 < E$(offline ECL cluster) $< 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

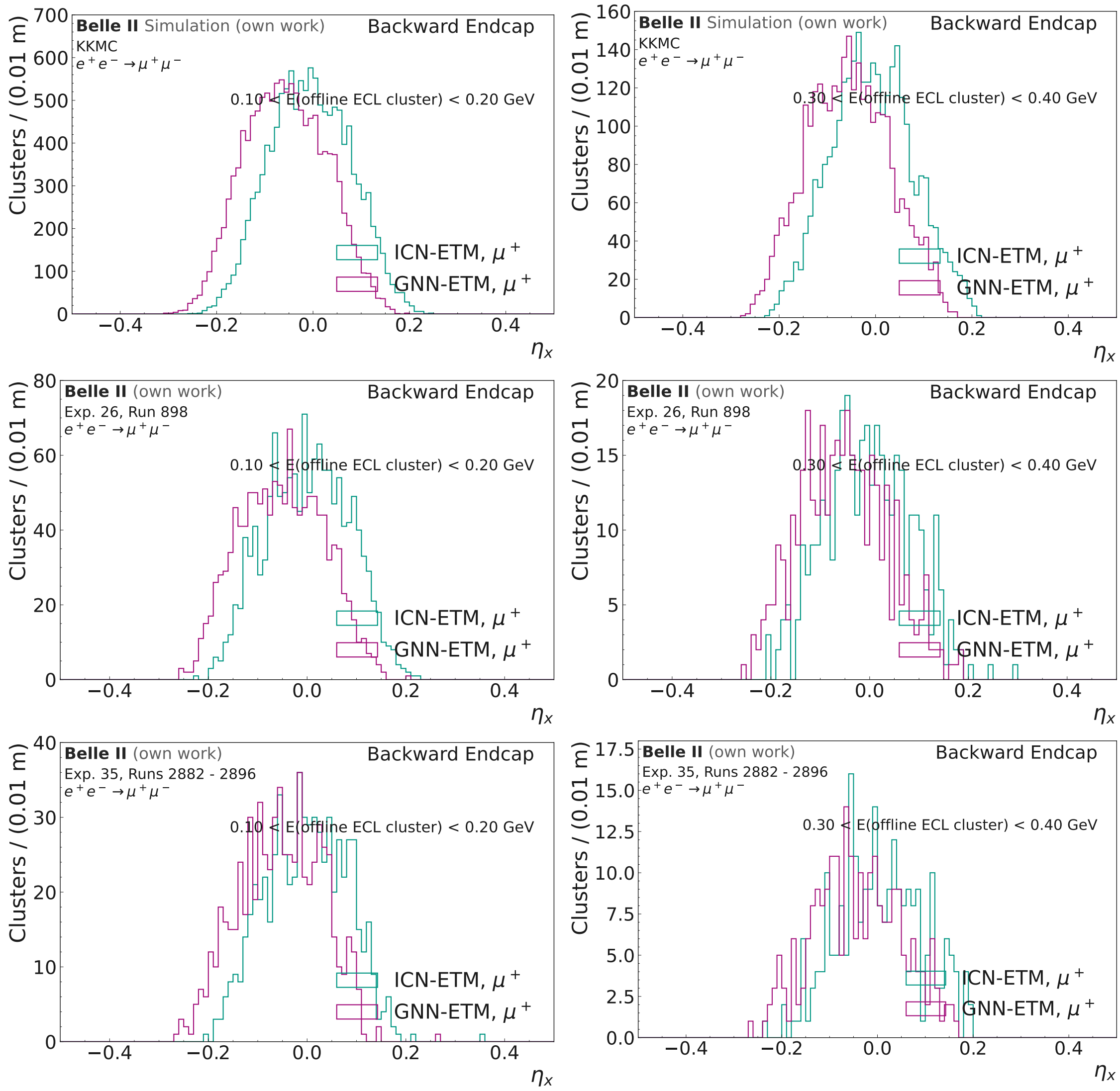


Figure D.28: η_x for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E$(offline ECL cluster) $< 0.2\,$GeV (left) and $0.3 < E$(offline ECL cluster) $< 0.4\,$GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

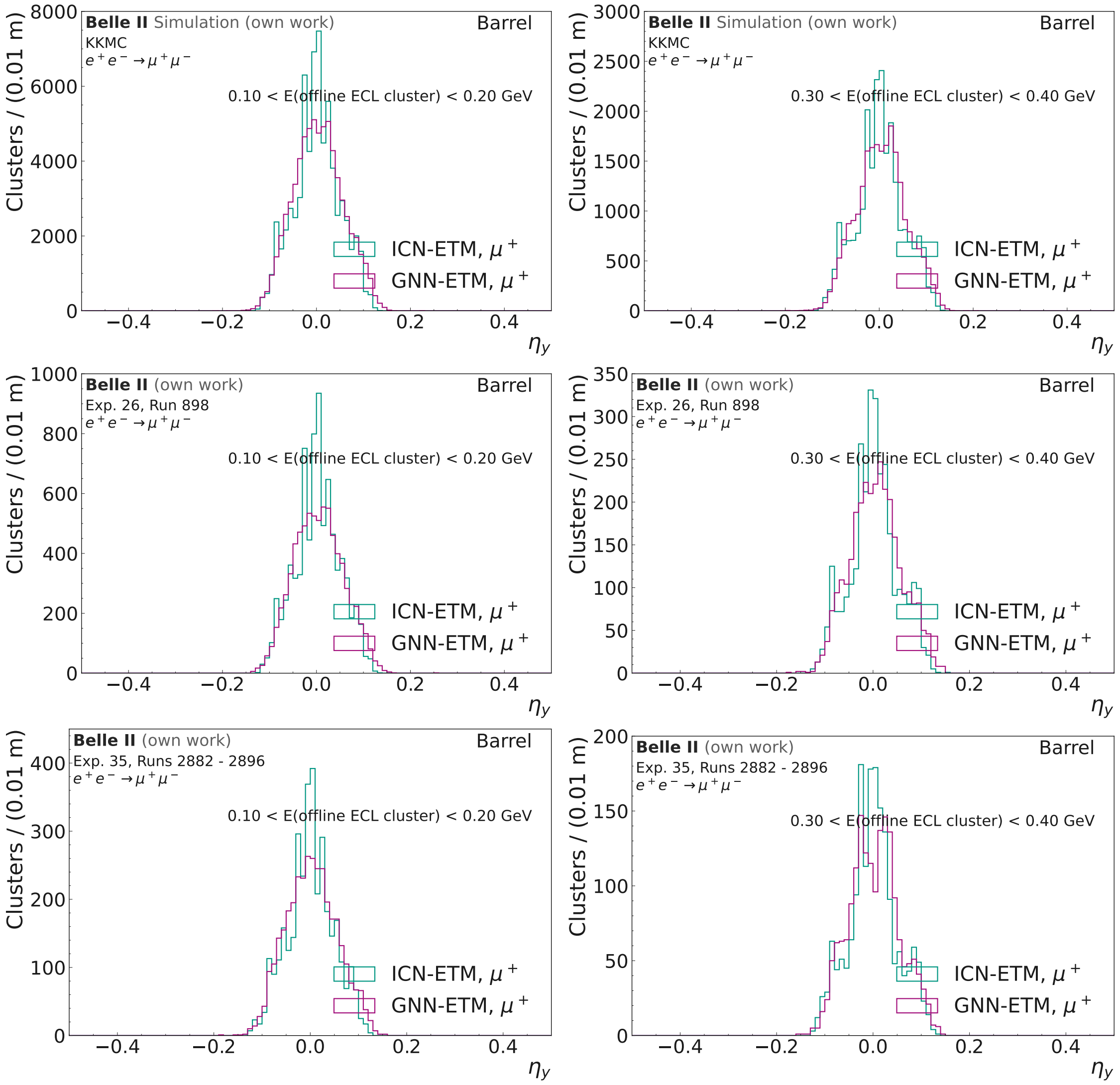


Figure D.29: η_y for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

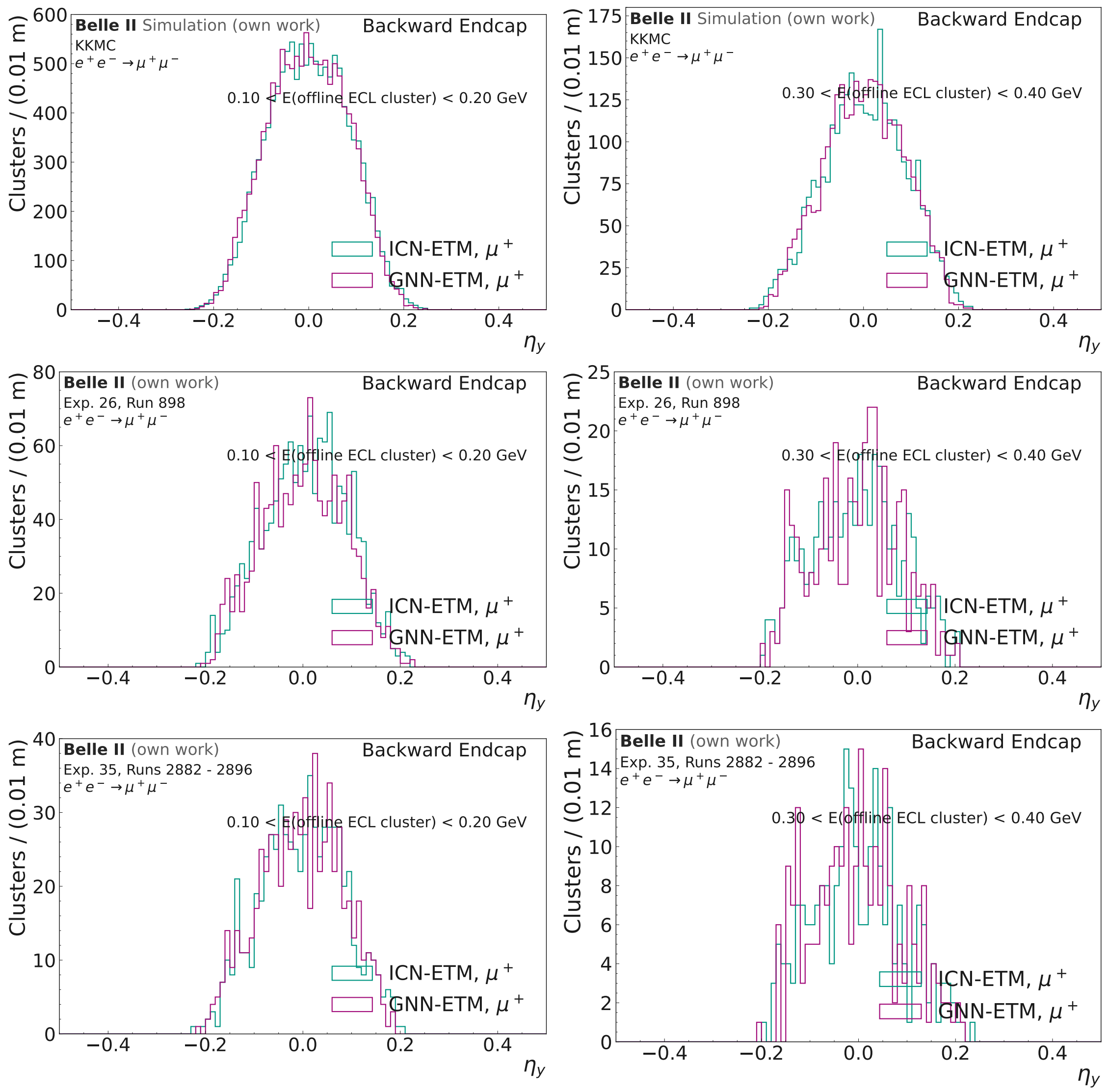


Figure D.30: η_y for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E$(offline ECL cluster) < 0.2 GeV (left) and $0.3 < E$(offline ECL cluster) < 0.4 GeV (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

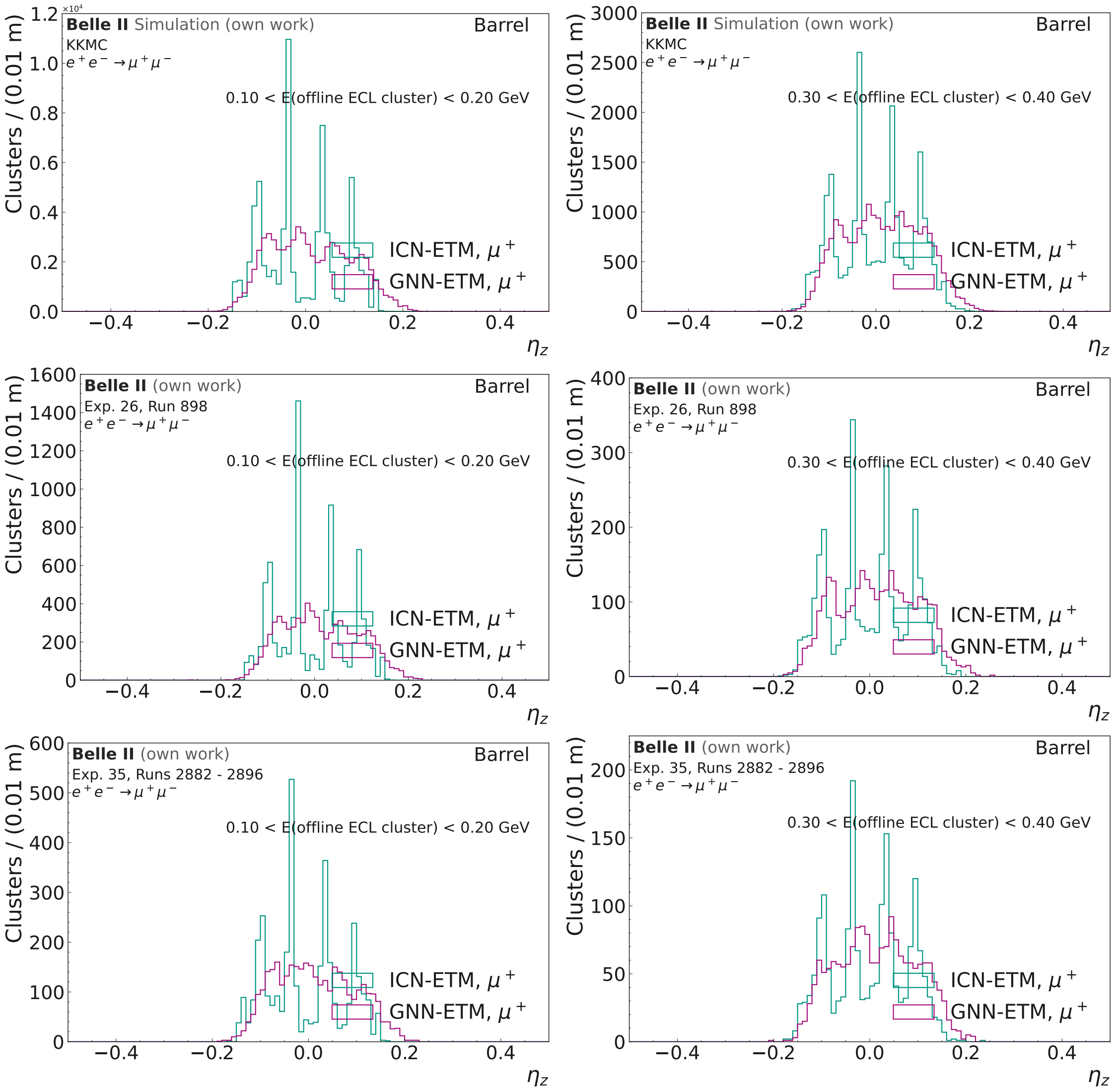


Figure D.31: η_z for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the barrel region.

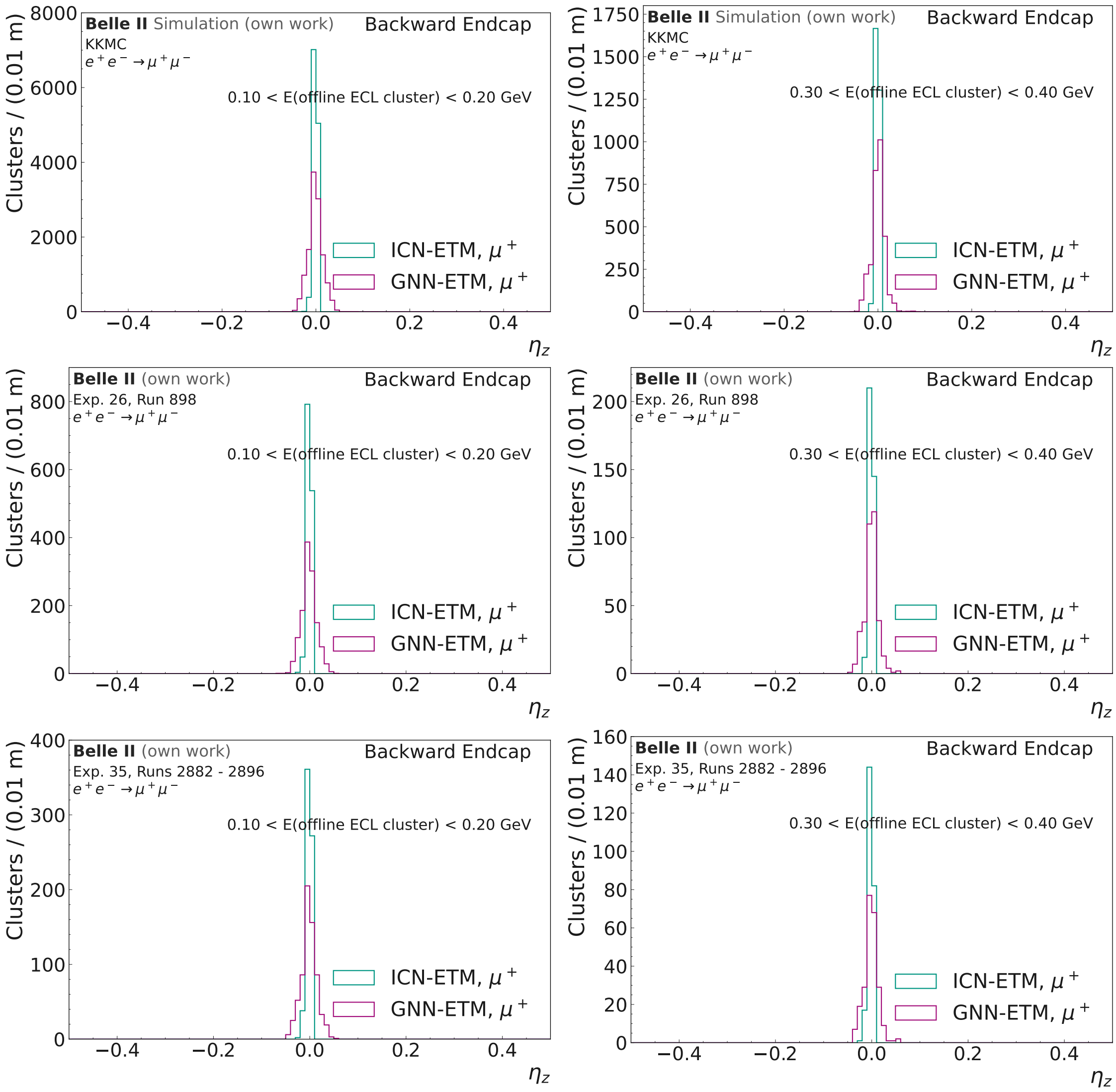


Figure D.32: η_z for GNN-ETM and ICN-ETM clusters matched to the mu^+ offline ECL cluster for $0.1 < E(\text{offline ECL cluster}) < 0.2\,\text{GeV}$ (left) and $0.3 < E(\text{offline ECL cluster}) < 0.4\,\text{GeV}$ (right). The top row shows the position resolution on the MC dataset, the center row on the Exp. 26 dataset and the bottom row on the Exp. 35 dataset. The resolution is shown for offline ECL clusters in the backward endcap.

D.3 Trigger Bit Efficiency for Random Triggered Events

This section shows the trigger bit efficiency for the GNN-ETM and ICN-ETM. The trigger bit efficiency for events triggered by the random trigger is shown in Fig. D.33, for events triggered by the Poisson trigger in Fig. D.34, and for events triggered by the bg trigger in Fig. D.35.

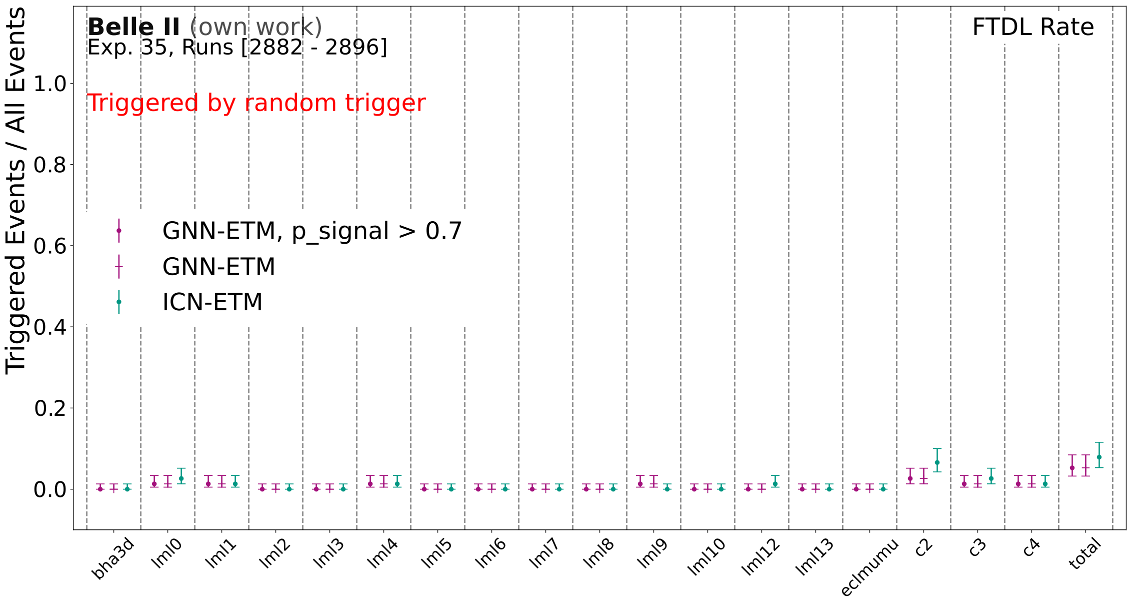


Figure D.33: Trigger bit efficiency for the ECL trigger bits for events triggered by the random trigger bit. The trigger bit efficiency for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit efficiency for the GNN-ETM, both for all clusters as well as for clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the random trigger has fired and the trigger decision window for ICN-ETM consists of data windows 3 and 4.

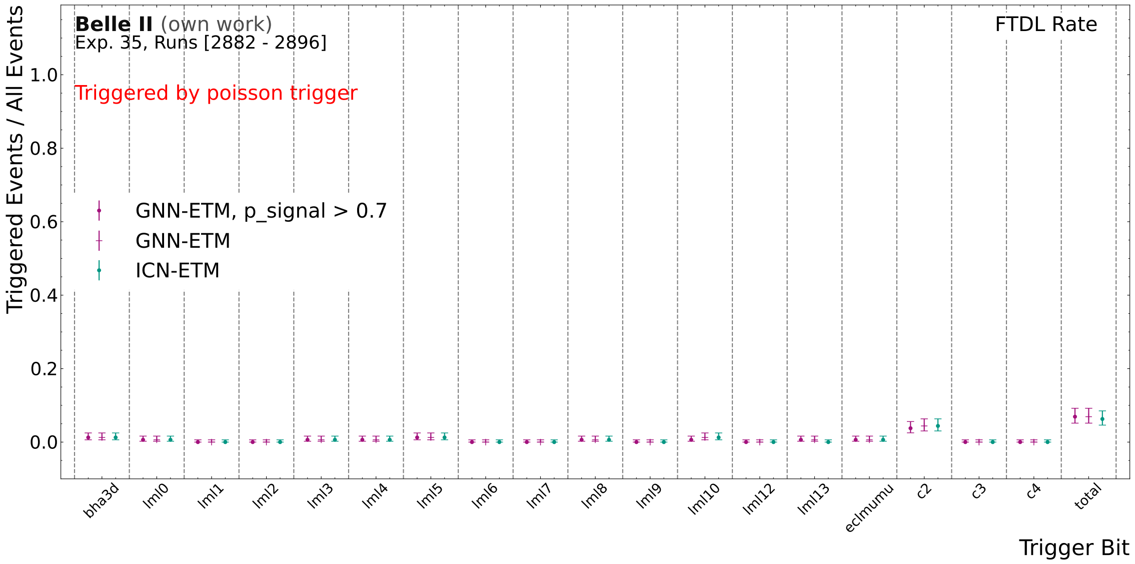


Figure D.34: Trigger bit efficiency for the ECL trigger bits for events triggered by the poisson trigger bit. The trigger bit efficiency for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit efficiency for the GNN-ETM, both for all clusters as well as for clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the poisson trigger has fired and the trigger decision window for ICN-ETM consists of data windows 3 and 4.

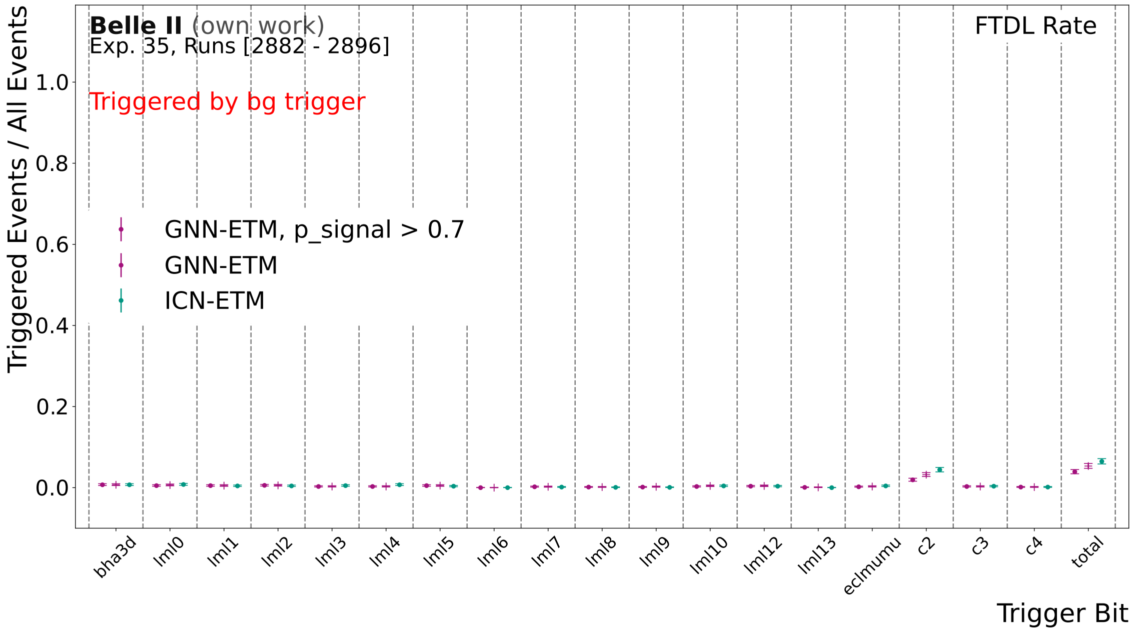


Figure D.35: Trigger bit efficiency for the ECL trigger bits for events triggered by the bg trigger bit. The trigger bit efficiency for the ICN-ETM is shown, which is taken from raw data, as well as the trigger bit efficiency for the GNN-ETM, both for all clusters as well as for clusters with a signal classifier value > 0.7. The total trigger bit is 1, if at least one of the other trigger bits is 1. The denominator is the number of events, in which the bg trigger has fired and the trigger decision window for ICN-ETM consists of data windows 3 and 4.

Glossary

Belle II A second-generation B factory and the successor of the Belle experiment. v, 1–9, 12, 16, 19, 25, 26, 29, 31– 33, 35, 36, 45, 49, 51, 94, 100, 103– 105, 107, 117, 123, 124, 126, 134, 147, 151, 171, 173, 174, 176, 177

GNN-ETM The network design for the graph neural network-based trigger algorithm. iii, vi, vii, 2, 25, 27, 37, 38, 74, 85, 89, 94– 96, 98– 101, 103– 109, 115, 117–119, 121– 128, 130, 131, 134– 137, 139– 154, 157– 177, 207– 245

ICN-ETM The current implementation for clustering for the Belle II electromagnetic calorimeter trigger, based on the Isolated Cluster Number logic. 2, 15– 19, 22– 24, 26, 34, 44, 52, 53, 55– 58, 74, 89– 92, 94, 95, 97, 98, 103– 112, 114, 117–127, 130, 134– 137, 139– 151, 158– 170, 172– 177, 195– 245

SuperKEKB An upgrade of the KEKB electron-positron collider and the accelerator at which the Belle II experiment is located. 1, 3, 4, 13, 60, 104

Acronyms

ANN All-Nearest-Neighbour. 96

ARICH Aerogel Ring-Imaging Cherenkov detector. 5, 7, 12

basf2 Belle II Analysis Software Framework. 11, 29, 32, 37, 39, 40, 43, 53, 54, 58

BRAM Block RAM. 96, 100

CDC Central Drift Chamber. 2, 5– 8, 11– 15, 23, 32, 105, 124, 128, 129, 151– 154

CM Center-of-Mass. 1, 3, 22, 27, 58, 127– 129, 131, 134, 147– 152, 154, 167, 168, 170

CMS Compact Muon Solenoid. 62

CNN Convolutional Neural Network. 61

CR Connected Region. 27– 30

CsI(Ti) Thallium-doped Cesium-Iodide. 8, 16, 28

DAQ Data Acquisition. 31

DEPFET Depleted Field Effect Transistor. 5

DSP Digital Signal Processor. 62, 72, 73, 98, 100

ECL Electromagnetic Calorimeter. 2, 5, 7– 20, 22, 25– 30, 32, 33, 35, 36, 38– 49, 51– 61, 68, 71, 73, 76, 84, 88, 90, 94, 95, 108– 122, 124– 131, 133– 148, 150– 154, 156– 169, 171– 177, 195– 242, 244, 245

ETM ECL Trigger Master Module. 15

FADC Fast Analog-to-Digital Converter. 17

FAM FlashADC Analog Module. 15– 18, 21, 26

FF Flip-Flop register. 98, 100

FPGA Field-Programmable Gate Array. 1, 2, 11– 13, 15, 17, 60, 72– 74, 78, 81, 85, 89, 93, 98, 176

FTDL Final Trigger Decision Line. 14, 35, 118, 119, 129, 167, 173

GDL Global Decision Logic. 11, 13– 15, 21– 23, 94, 95, 123

GNN Graph Neural Network. 2, 61, 62, 73, 103, 104, 176, 177

GRL Global Reconstruction Logic. 13– 15, 22

HEP High Energy Physics. 1, 2, 25, 61

HER high-energy ring. 3, 51

HLT High Level Trigger. 1, 11, 27, 117, 128– 130, 152– 154

IP interaction point. 1, 3, 5, 6, 8, 13, 53, 54

KLM K_L^0 and Muon Detector. 2, 5, 7, 8, 11– 14, 31, 43

kNN k-Nearest-Neighbour. 60– 63, 79, 89, 93

L1 trigger Level 1 Trigger. 1, 2, 6– 8, 11– 15, 20, 25, 29, 43, 44, 60, 73– 75, 94, 95, 104, 117, 119, 126, 128– 130, 135, 152– 154, 171, 176, 177

LER low-energy ring. 3, 51

LHC Large Hadron Collider. 62

LM Local Maximum. 28– 30

LUT Lookup Table. 22, 27, 37, 53, 62, 68, 72– 74, 81, 96, 98, 100, 147

MAE Mean Absolute Error. 89– 91

MC Monte Carlo. 2, 8, 25, 27, 32, 35– 43, 45– 48, 51, 117, 118, 123, 125– 133, 135– 147, 149– 152, 154– 156, 158– 168, 170, 171, 208– 215, 217– 224, 226– 233, 235– 242

ML Machine Learning. 2, 60, 72, 87, 176

NN Neural Network. 60

OC Object Condensation. 2, 52, 62, 64– 66, 71, 83, 86, 89, 96

PID particle identification. 5, 7

PSNM Pre-Scale And Mask. 15, 129

PXD Pixel Detector. 5, 6, 12, 31, 105

rd run-dependent. 32, 33, 35, 36

ri run-independent. 32, 33, 35, 36, 45– 47, 51, 117, 142

ROC Receiver Operating Characteristic. 52, 53, 58, 59, 82– 84, 90– 92, 109, 112, 113

SM Standard Model of particle physics. 1

SVD Silicon Vertex Detector. 5, 6, 11, 12, 31, 105, 124

TC Trigger Cell. 15– 23, 26, 27, 29, 34, 36– 38, 42, 44, 45, 53, 54, 57, 60, 61, 64, 65, 68, 71, 74, 88, 89, 91, 92, 94– 99, 101, 104, 106, 108, 109, 115, 117, 119– 121, 123, 124, 126, 131, 134, 136, 142, 147, 158, 159, 163, 167, 168, 176, 177

TMM Trigger Merger Module. 15, 18

TOP Time-Of-Propagation counter. 5, 7, 11– 15, 124

TSIM Trigger Simulation. 26, 37, 44, 53, 58, 117, 118

UT3 Universal Trigger Board 3. 12

UT4 Universal Trigger Board 4. 13, 60, 94, 98– 100, 176