Noise Waveform Generation Using GANs and Charged Particle Identification Using Pulse Shape Discrimination in the Belle II Electromagnetic Calorimeter

by

Alexandre Beaubien
B.Sc.H, McGill University, 2019

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Physics and Astronomy

Noise Waveform Generation Using GANs and Charged Particle Identification Using Pulse Shape Discrimination in the Belle II Electromagnetic Calorimeter

by

Alexandre Beaubien
B.Sc.H, McGill University, 2019

Supervisory Committee

Dr. J. Michael Roney, Supervisor
(Department of Physics and Astronomy)

Dr. Robert Kowalewski, Departmental Member
(Department of Physics and Astronomy)

ABSTRACT

This thesis investigates the use of generative adversarial networks (GANs) as an alternative method to simulate noise waveforms for Belle II CsI(Tl) calorimeter crystals. Presented is a deep convolutional GAN (DCGAN) trained using background waveforms recorded in the ECL during a physics run. Results are presented showing good agreement in the distribution of metrics comparing data and simulated noise waveforms using a two-sample Kolmogorov-Smirnov test. The models are shown to be difficult to train, and many possible improvements are identified.

Secondly, this thesis showcases the development of a particle identification tool relying on pulse shape discrimination (PSD) as an input to a gradient boosted decision tree (GBDT) classifier. Two models are trained to discriminate $\mu^{\pm}$, $\pi^{\pm}$ and $e^{\pm}$, $\pi^{\pm}$. Results show that PSD charged particle identification in the ECL improves the $e^{\pm}$, $\pi^{\pm}$ discrimination, but result in smaller improvements to the $\mu^{\pm}$, $\pi^{\pm}$ discrimination. Results also show an improvement to the result obtained with the currently implemented PSD discriminator trained on neutral particles (γ, K_L^0).

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank:

Prof. J. Michael Roney for the constant guidance, support and insight that he offered me. His help was instrumental in the fruition of these projects on an organizational, technical and personal level.

Prof. Robert Kowalewski for his feedback and comments throughout my M.Sc. that provided me with many learning opportunities.

Dr. Alexei Sibidanov for his constant feedback and ideas, and for the dedicated time spent on teaching me much of what was needed to accomplish these projects.

My fellow graduate students and office mates for their continuous willingness to bounce ideas around, their help with any sorts of problems, and for all the enjoyable moments.

Dedication

À ma famille, en particulier à Samuel qui a bien mérité cette mention, à mes amis et à Anqian

Chapter 1

Introduction

The Belle II experiment is part of the next generation of B-Factories, and is a direct successor to the Belle experiment. It is located in Tsukuba, Japan at the SuperKEKB e^+e^- accelerator, the highest luminosity collider in the world [1]. This improvement in luminosity will allow Belle II to collect a dataset of around 50ab^{-1} by the end of its experiment lifetime, about 30 times the combined dataset of the Belle and BaBar experiments [2]. This will come at the cost of significantly higher background in the detector, on the order of 10 to 20 times the conditions observed in Belle [3]. Higher background in the detector has ramifications in its operation, and in the physics results coming from it.

The electromagnetic calorimeter (ECL) is the sub-detector responsible for measuring the electromagnetic energy deposition of particles; it specializes in the detection of photons, the identification of electrons from heavier particles and the measurement of K_L^0 alongside with the K_L^0 and μ detector (KLM). The ECL will also experience higher background and pileup of detected particles from the luminosity increase. The current procedure used to simulate beam-related background conditions in the ECL consists of saving the background measured during run time using a mix of random and delayed physics sampling, and overlaying them onto simulations. This method requires extensive storage and file transfer capabilities, and will soon cause problems across the grid systems used for the simulation due to the limited capacities of Worldwide LHC Computing Grid (WLCG). To circumvent these limitations, work done in this thesis explores the use of generative adversarial networks (GANs) to simulate the background conditions of the ECL as a replacement for the overlaying procedure.

Although many improvements are being implemented to address new and increased background conditions, the large physics output of Belle II is still limited by recurrent systematics such as particle misidentification. An example case is the discrimination of $\mu^{\pm}$ and $\pi^{\pm}$ whose similar masses, 105.66 MeV/c^2 and 139.57 MeV/c^2 [4] respectively, make it particularly difficult to identify using track based or Cherenkov based methods. The work in this thesis studies the usage of pulse shape discrimination (PSD), a new tool implemented in the ECL, as a way to discriminate between energy depositions from charged hadrons and charged leptons. A gradient boosted decision tree is developed only using ECL information to capture and convert PSD information obtained from a particle shower into a probabilistic output characterizing the likelihood of this shower having been produced by a hadron.

The thesis is organized as follow:

- Chapter 2 presents an overview of the Standard Model of particle physics (SM), highlights the need for collider experiments such as Belle II, and offers a broad overview of the Belle II operation and its sub-detectors. It also goes into details about the relevant particle interactions present in the ECL.

- Chapter 3 presents a detailed view of the ECL's operation and of its modelling in the GEANT4 software. It describes tasks performed by the author to calibrate the PSD tool, and to improve the GEANT4 implementation.

- Chapter 4 gives an overview of the machine learning theory behind GANs and offers a demonstration of a GAN model capable of simulating the background conditions within the ECL. It explores the challenges of training GANs, defines metrics to be used to evaluate its performance, and describes the shortcomings of the model. Next steps are given to direct how the broad implementation of a GAN would proceed in the Belle II collaboration, as well as ways to address shortcomings and challenges present in the current implementation.

- Chapter 5 presents the building of a machine learning tool used to discriminate $\mu^{\pm}$, $\pi^{\pm}$ and $e^{\pm}$, $\pi^{\pm}$using only ECL information. It further defines the limitations of PSD as a discrimination tool in charged particles and explores how such a tool would be used.

- Chapter 6 concludes the thesis with an overview of the contributions made and reflects on the improvements and future work to be done.

Chapter 2

Standard Model and the Belle II Experiment

This chapter begins with an overview of the Standard Model (SM) of particle physics, currently the best model for particle physics and the one being probed by the Belle II experiment. This is followed by a description of SM physics questions and the need for collider experiments to address them. Then, a brief description of the Belle II detector and of its sub-detectors is given in addition to a short summary of its physics program. Finally, a more detailed description of the physics interactions studied within this thesis is given.

2.1 SM Foundations

The SM of particles is a theory that explains the nature of elementary particles and their behaviour. The interactions by which particles can interact with one another are the electromagnetic force, the strong nuclear force and the weak nuclear force. The SM describes the effects of these interactions through the coupling of force mediators (gauge bosons) with the different particles where each gauge boson is associated to a specific symmetry. The representation of the different particles under the symmetries determines their quantum numbers and their degree of interaction with each force. In the phenomenology of the SM, the different particles are excitations of quantum fields unique to each particle type.

The SM is thus built from four main ingredients:

1. The framework of quantum field theory (QFT) which describes the interaction of the different quantum fields (particles) with one another.

2. The internal local (gauge) symmetries given by the direct product SU(3) $\times$ SU(2)$_L$ $\times$ U(1)$_Y$ where SU(3) is the quantum chromodynamics symmetry (or

strong nuclear force) and SU$(2)_L$ × U$(1)_Y$ is the electroweak symmetry (electroweak force). Each of these symmetries possesses a number of gauge bosons by which the associated interaction can be mediated.

3. The spacetime symmetries described by the Poincaré group which accounts for relativity.

4. The Higgs mechanism spawned by the scalar Higgs field which spontaneously breaks down the symmetries SU(3) × SU$(2)_L$ × U$(1)_Y$ into SU(3) × U(1), where U(1) represents the residual electromagnetic interaction. In the process, the massive bosons $W^\pm$ and Z^0 are generated mediating the weak nuclear interaction. The Higgs mechanism has the further effect of giving mass to the different matter particles.

The SM as previously detailed explicitly violates the discrete charge (C) and parity (P) symmetries due to the lack of chirality in the SU$(2)_L$ × U$(1)_Y$ symmetry as the spontaneously broken weak interaction only interacts with left-handed particles. Furthermore, the combination of both discrete symmetries into the discrete CP symmetry is also violated by the flavour currents predicted within CKM matrix processes. This was first experimentally measured in the K^0 system in 1964 and more recently in the B system at the BaBar and Belle experiments. With the addition of the time-reversal symmetry (T) to CP, the CPT symmetry is predicted to remain unbroken within the SM. This is currently supported by experiments.

2.1.1 Interactions and Particles

The current phenomenology of the SM particles can be separated into three categories: gauge bosons which act as force mediators to the other particles, fermions which are matter particles that interact with one another through interactions mediated by the gauge bosons and the Higgs particle coming from the Higgs field.

Figure 2.1 shows a table containing the elementary particles of the SM with their appropriate intrinsic properties (mass, electric charge, spin). In the table, particles are further separated according to which symmetries apply to them.

First, the gauge bosons (in red) are the interaction propagators. The gluons (g) are the carrier of the strong nuclear force (quantum chromodynamics); there are 8 types of gluons representing the different color charge interactions. The photon (γ) is the carrier of the electromagnetic force and is a stable particle. The Z^0 and $W^\pm$

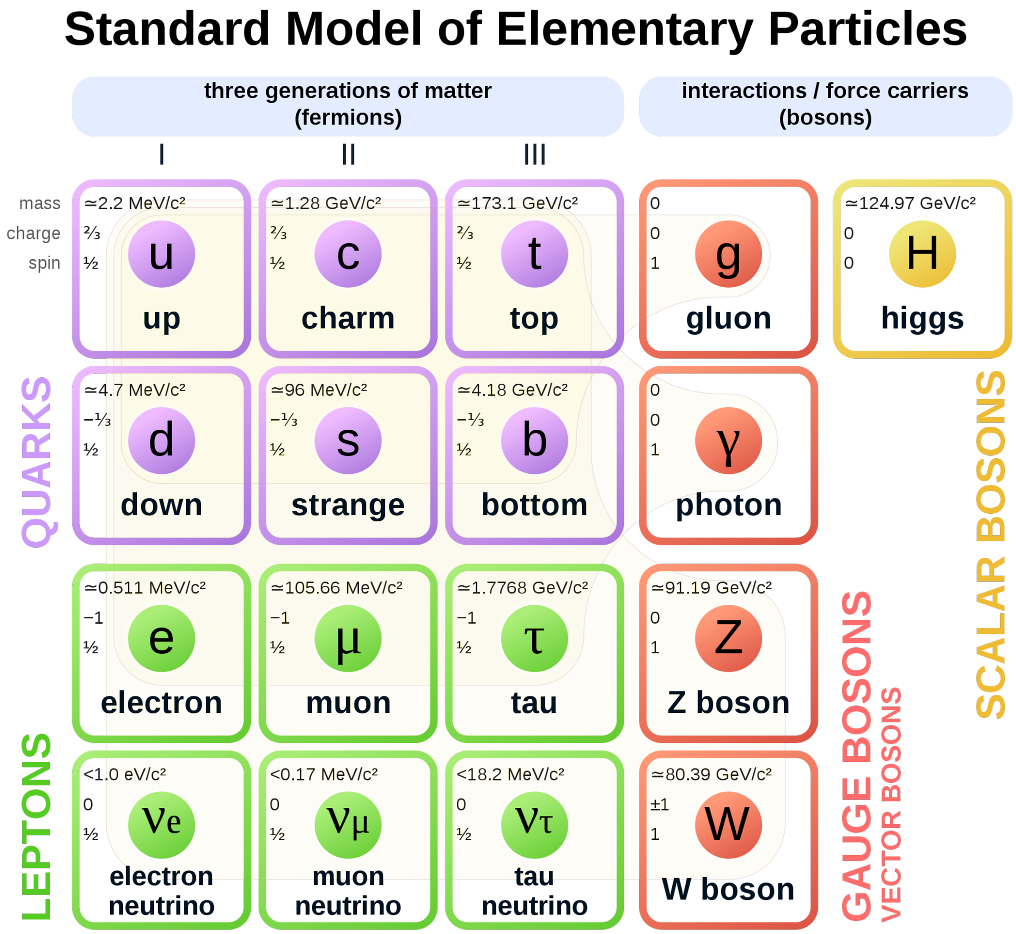


Figure 2.1: Table of elementary particles.

bosons are the carrier of the weak nuclear force. Particles coupling with the strong force (gluons) have a color charge, particles coupling with electromagnetism (photon) have an electric charge and particles coupling with weak nuclear force have weak isospin/hypercharge.

The fermions are broadly broken down into two categories: quarks and leptons; quarks have color charge while leptons do not. This means that quarks interact through the strong nuclear interaction. Both quarks and leptons come in three generations, each successive one being more massive than the previous. Furthermore, each generation comes in pairs of particles with an electric charge difference of one. Both quarks and leptons possess a weak coupling and an electromagnetic coupling except for the neutrinos which are electrically neutral.

Finally, the Higgs boson is a scalar boson without electromagnetic coupling or strong coupling. The Higgs couples to weak interaction bosons (Z^0 and $W^{\pm}$) and can couple directly to fermions through their mass coupling terms. Finally, it can interact with other Higgs particles through direct coupling.

2.1.2 Need for Colliders

However successful the SM has been, it is incomplete. The absence of gravity within a standard model of particles shows a lack of unity between the pillar theories of modern physics as evidence that gravity is mediated by gravitons remains lacking. Other anomalies like the presence of dark matter in the universe have yet to be explained and many physicists believe that dark matter could be explained through one or many undiscovered particles. Furthermore, the discovery of neutrino oscillations by the Super-Kamiokande experiment in 1998 [5] and by the SNO experiment in 2001 [6] implies that neutrinos are massive, which is not predicted by the SM. Together with many other open questions in physics, these discrepancies point to the existence of physics beyond the SM and provide areas of research into new physics.

The search for new physics can be done in many ways, one of them being the use of collider experiments. The high luminosity of colliders leads to a large number of low cross section processes that are difficult to observe in the natural world. One such example is the production of $B\bar{B}$ meson pairs whose high mass and short lifetimes renders passive observation of these particles difficult. This is mainly due to the lack of information about the initial state conditions (e.g. time, location and energy) and the inability to capture all particle decays coming from the event. By creating colliders capable of producing $B\bar{B}$ mesons, and more importantly by building a well studied detector around the production area, precision measurements of their behaviour can be achieved. This allows physicists to peer into the SM in ways that passive observation experiments cannot. For instance, the CP violation in mixing between neutral $B\bar{B}$ requires knowledge of initial conditions and decays products. Furthermore, since the effect of CP violation are small, a large number of events must be observed to significantly compare results with the SM.

Two common approaches to collider physics are the high energy frontier and the high intensity frontier. High energy experiments provide a window into processes that cannot be seen at regular energies, and help push the boundary of new physics by exploring processes that may be too difficult to see at lower energies. The LHC, currently the highest energy collider in the world [7], experimentally measured the

Higgs boson for the first time in 2012 [8]. This discovery required both high energies and the accuracy of collider environments.

The other common approach is the usage of high intensity colliders which provide a window into rare processes by providing a very large dataset with very high accuracy. The Belle and BaBar experiments are example of high intensity experiments. Although their energies are lower than the LHC, the clean initial state provided by their e^+e^- collision combined with the higher integrated luminosity of the experiment allowed for the measurement of very rare processes and measurements in flavour physics. One such measurement is the discovery of CP violation in the $B\bar{B}$ system [9][10] that validated the CKM theory and led to the 2008 Nobel Prize.

2.2 Belle II

The Belle II experiment is a new generation B-Factory located at the SuperKEKB e^+e^- collider. It started collecting collision data with its full detector in early 2019, and is expected to achieve an integrated luminosity of 50ab^{-1} by the end of its experiment lifetime [2]. The Belle II data set will be around 30 times [3] that of the previous B-Factories, namely BaBar and Belle, combined.

Its physics program mainly involves precision electroweak measurements and searches for new physics beyond the SM. To do so, it operates at the intensity frontier. The clean environment of the e^+e^- collision provides high accuracy in the initial state of the collision, and where an array of sub-detectors can measure the final states of the collision. The sub-detectors of Belle II are presented in section 2.3.

The common fermion-antifermions states produced through e^+e^- collisions include $b\bar{b}$, $c\bar{c}$, $s\bar{s}$, $d\bar{d}$, $u\bar{u}$ in the quark sector and $\tau^+\tau^-$, $\mu^+\mu^-$, e^+e^- in the lepton sector. Other initial states such as $\gamma\gamma$ can also be produced through e^+e^- collisions.

2.2.1 SuperKEKB

Based in Tsukuba, Japan, the SuperKEKB e^+e^- collider operates at a center-of-mass energy of $\sqrt{s} = 10.58$ GeV [11]. This energy was chosen to be above the threshold for $B\bar{B}$ production so that the B particles are created in an entangled state enabling the initial state of the B meson to be known based on the decay of the $\bar{B}$ meson (and vice-versa) using the time difference of their decays. This energy maximizes the production cross-section of $e^+e^- \to \Upsilon(4s)$ resonance states, which are strong bound states of $b\bar{b}$ quarks. The $\Upsilon(4s)$ has slightly higher energy than that of $B\bar{B}$ meson

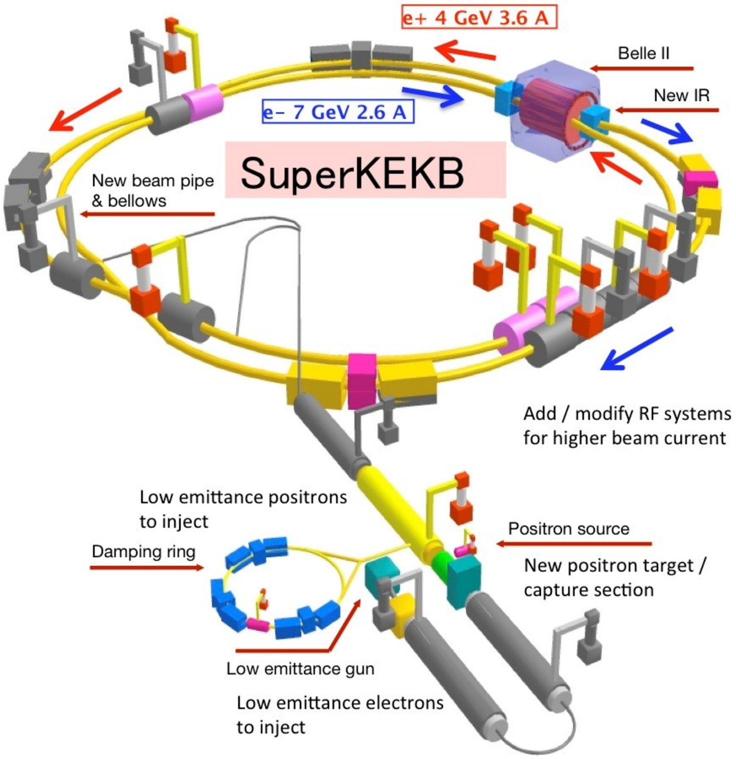


Figure 2.2: A schematic of the SuperKEKB e^+e^- collider providing collisions for the Belle II experiment.

pairs, and has a large branching fraction to $B\bar{B}$ meson pairs (B^+B^- and $B^0\bar{B}^0$) of about $> 96\%$ [4]. Due to the high number of $B\bar{B}$ produced, these type of experiments are often called B-Factories.

The SuperKEKB collider has an asymmetric beam energy design with its electron and positron beams operating at 7 GeV and 4 GeV respectively [11]. This asymmetric beam design allows for collisions to have a net final momentum in the lab frame. This mainly provides sufficient momentum for produced particles with short lifetimes to traverse a measurable distance from the interaction point. Their decay vertices are then measured inside of the vertex detector.

The SuperKEKB collider has a target instantaneous luminosity of $8 \times 10^{35}\text{cm}^{-2}\text{s}^{-1}$ which is around 40 times higher than that of KEKB, its predecessor. On the 21th June 2020, it achieved the world's highest instantaneous luminosity for a colliding-beam accelerator ($2.40 \times 10^{34}\text{cm}^{-2}\text{s}^{-1}$) making it the highest luminosity collider in the world [1].

2.3 Belle II Sub-Detectors

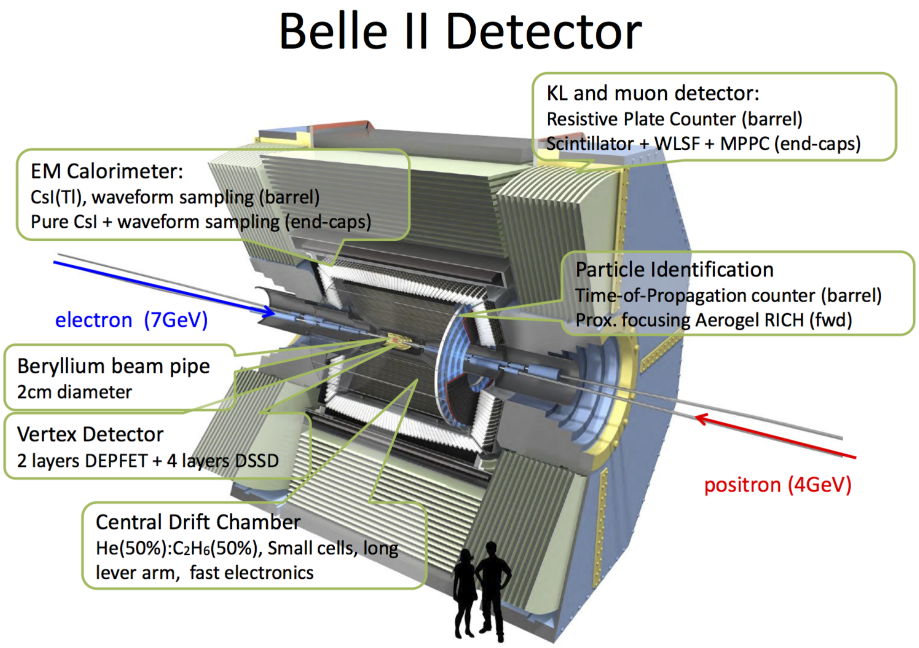

Belle II Detector
KL and muon detector:
Resistive Plate Counter (barrel)
Scintillator + WLSF + MPPC (end-caps)
EM Calorimeter:
CsI(Tl), waveform sampling (barrel)
Pure CsI + waveform sampling (end-caps)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (fwd)
electron (7GeV)
Beryllium beam pipe
2cm diameter
Vertex Detector
2 layers DEPFET + 4 layers DSSD
positron (4GeV)
Central Drift Chamber
He(50%):C2H6(50%), Small cells, long
lever arm, fast electronics

Figure 2.3: A schematic of the Belle II detector to scale. Included are brief descriptions of all the sub-detectors present in Belle II. Image credit is the Belle II collaboration.

To detect the particles emanating from the collisions taking place in the SuperKEKB collider, the Belle II detector is built with multiple sub-detectors arranged in layers cylindrically around the interaction point (IP) where collisions happen. Figure 2.3 shows a schematic view of the Belle II detector with all its components.

This section will go into details about the different sub-detectors in Belle II starting at the IP and finishing at the outer edges of the detector. It will further go into detail about the magnets present in Belle II and give an overview of the trigger system used for data acquisition.

2.3.1 Solenoid Magnet

The inner part of the Belle II detector is inside of a superconducting solenoid that provides a magnetic field of 1.5 T in a cylindrical volume of 4.4 m in length and 3.4 m in diameter [12]. The field created allows charged particle's tracks to bend which in turns allows the measurement of their momenta.

An iron yoke, which is part of the KLM detector (section 2.3.6), serves as the return path of the magnetic flux, and also as interaction material.

2.3.2 Vertex

The vertex detector is the first and closest detector to the IP. Its purpose is to accurately measure the vertices of short lived particles, such as B mesons, D mesons and τ particles. It is made of two components: the pixel detector (PXD) with two layers a few millimetres from the beam pipe and the silicon vertex detector (SVD) positioned a few centimetres from the beam pipe. Together, they span six layers from 14 mm to 140 mm in distance from the IP.

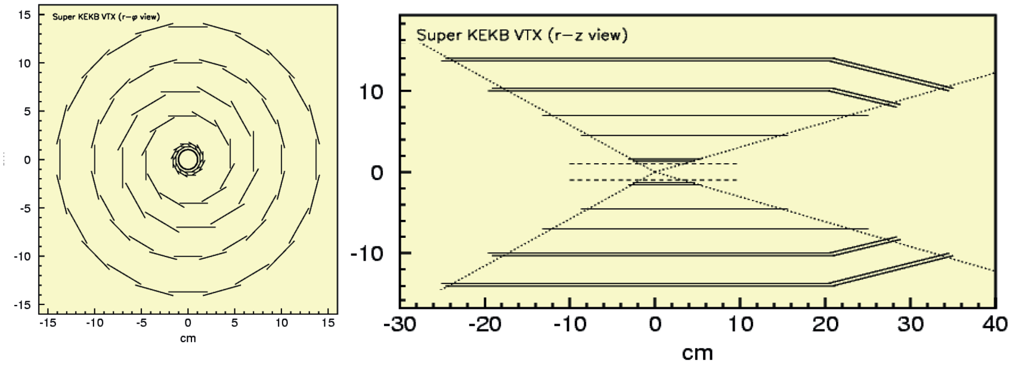


Figure 2.4: A schematic view of the six layers that constitute the VXD. Image from [3].

The first two layers are part of the pixel detectors (PXD), and are composed of pixelated sensors of depleted field effect transistor DEPFET [11]. These pixel detectors are resilient to the high hit rates present close to the IP making them appropriate for the first layers.

The DEPFET technology allows the sensors to be very thin, down to 50 microns which minimizes scattering. The PXD is arranged in panels as shown in figure 2.4, which are pixelized to allow vertex resolution on the order of 10 microns. The readout electronics can digitize the pixels of a panels in about 20 microseconds.

The next four layers are part of the silicon vertex detector (SVD), and are composed of double-sided silicon strip detectors (DSSDs) [11]. These layers are not pixelized, due to prohibitive costs and channel counts, but provide high precision and immunity to background hits and radiation tolerance at farther distances from the IP.

Together, the PXD and the SVD provide very accurate position information of decaying particles. This allows the reconstruction of decay vertices for short lived particles such as B mesons, D mesons and τ leptons.

2.3.3 CDC

The central drift chamber (CDC) is the detector directly following the vertex detectors around the IP. As charged particles pass through the inner detector, they experience the magnetic field present and produce curved tracks.

The CDC has three main tasks:

- The measurement of particles' trajectories as they cross the CDC, whose curvature allows for the measurement of the particle's momentum. Furthermore, knowing the trajectory allows the reconstruction of events by associating vertices, tracks and energy depositions of particles together.
- The measurement of the energy loss within its gas volume (dE/dx) provides particle identification information. Low momentum particles that do not reach outer detectors can sometimes be identified entirely from the CDC using this information.
- The triggering of charged particle measurements in the detector. This is detailed further in 2.3.7.

The drift chamber is a cylinder with outer radius of 1.130 m. It posseses 56 layers together containing 14336 wires of diameter 30 μm which are either oriented in the "axial" direction (along the magnetic field) or in the "stereo" direction (skewed to the magnetic field) [11]. As charged particles cross the 50% He - 50% C_2H_6 gas, they ionize the gas particles along their tracks. Secondary electrons produced by the ionization cause an electron avalanche that drifts towards signal wires due to the electric field present in the CDC. By measuring the drift time of electron avalanches using both orientation of wires, the full 3D reconstruction of tracks' positions and shapes can be performed.

2.3.4 PID

In between the calorimeters and tracking/vertexing detectors are two particle identification detectors. Both detectors are Cherenkov detectors, and are used to measure the velocity of charged particles. By combining the velocity of particles with their momentum measured in the CDC, the mass of the charged particle can be reconstructed providing a particle identification hypothesis.

TOP

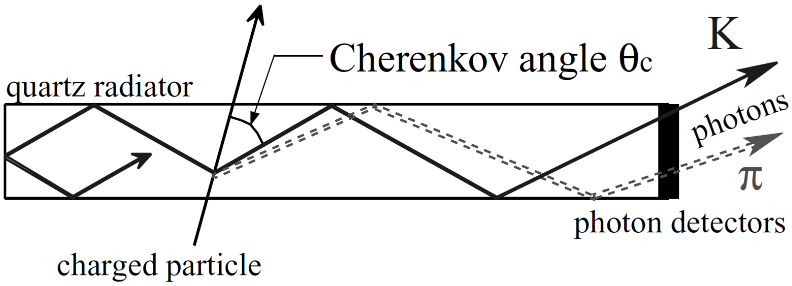


Figure 2.5: A schematic view of a slab in the TOP detector. Image from [11].

The time-of-propagation (TOP) detector is composed of sixteen 260 × 45 × 2 cm quartz bars arranged around the barrel region [11]. As charged particles cross the slabs, they emit Cherenkov light cones which are reflected along the walls of the bar until they reach the readout system at the end. The readout is composed of two rows of sixteen multi-anode photon detectors. The Cherenkov angle is obtained from the time of arrival and impact position of Cherenkov photons emitted by the particle.

ARICH

The aerogel-ring-imaging-Cherenkov detector (ARICH), located at the forward endcap, is composed of two 2 cm thick layers of aerogel with different refractive indices ($n = 1.045$ upstream, $n = 1.055$ downstream).

As charged particles cross the aerogel, a Cherenkov light cone is emitted and allowed $\sim$ 20 cm to expand, at which point the ring of light is measured on the readout plane. The refractive indices of the aerogel are chosen so that both cones converge at the same place in the readout plane. This allows for particles to interact with more material while still producing a focused ring [11].

The readout system consists of the hybrid avalanche photon detector (HAPD), an array of pixelated avalanche photon detectors. The Cherenkov angle is obtained by calculating the radius of the ring [11].

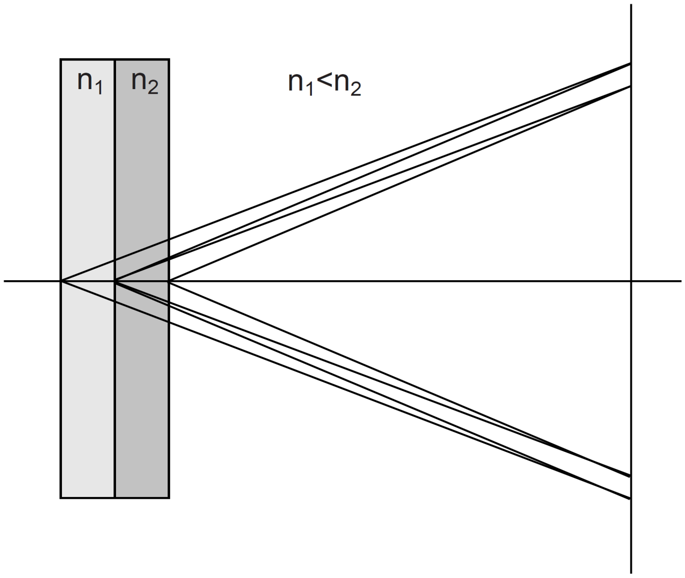


Figure 2.6: A schematic of the aerogel layers of the ARICH detector. Image from [11].

2.3.5 Calorimeter (ECL)

The electromagnetic calorimeter (ECL) is the sub-detector in charge of detecting particle's energies through energy depositions. It specializes in the detection and energy measurement of gamma rays, as well as the identification and energy measurement of electrons.

The ECL sub-detector is made of 8736 CsI(Tl) crystals arranged in a cylinder (barrel) and two endcaps (forward and backward) around the interaction point. Once assembled, the ECL is 3 m in length with an inner radius of 1.25 m [11].

As particles cross the ECL crystals, they deposit their energy in the form of particle showers. By reconstructing showers, the energy of the particles can be measured. The ECL also possesses pulse shape discrimination abilities to help identify particles (section 3.4). Further detail is given about the ECL detector in chapter 3.

2.3.6 KLM

The K_L^0 and μ detector (KLM) consists of alternating plates of iron and active detector elements, and is located outside the superconducting solenoid with the iron material guiding the return flux of the magnet. The active element plates are layers of scintillator strips with wavelength shifting fibers. In total, the KLM is composed of 15 detector layers and 14 iron plates in the barrel region, and 14 detector layers and 14 iron plates in each endcap. The readout of the detector is provided by silicon photomultiplier (SiPMs) light sensors [11].

The extra material present in the KLM offers extra length for the K_L^0 to interact and shower hadronically. Muons are also detected as their passing through the material creates straight tracks of ionized particles.

2.3.7 Trigger

As Belle II lives under high background conditions, triggers are used to determine which observed events should be saved and which should not. The Belle II trigger scheme is based on the Belle trigger scheme. It relies on sub-decisions from the subdetectors, and a final logical decision that regroups the information from all detectors. Each sub-system has FPGAs to permit reprogrammable trigger decisions.

Each sub-detector provides specific information useful for trigger. The PID detectors provide timing trigger due to their good innate time resolution. The CDC provides trigger decisions based on charged track information (momentum, position, charge, number of tracks). The ECL provides trigger decisions based on energy depositions (size, position) which are essential in the identification of Bhabha ($e^+e^- \to e^+e^-$ scattering) and cosmic background. Finally, the KLM provides triggers relating to muon and kaon tracks. After every sub-detector provides a trigger decision, a global decision logic (GDL) is used to make a final decision for the first level of triggers.

Once triggered, the sub-detector's information is sent to the high level trigger (HLT) where a full reconstruction using all detector is performed and a software trigger is made using physics-level event selection software.

2.3.8 Physics Program at Belle II

The main physics program of Belle II is the search for new physics in the flavour sector at the intensity frontier, and the precision measurements of SM processes. Here are some of these questions that Belle II aims to address in the flavour sector:

1. Are there any new CP violating phases in the quark sector?

2. Are there any processes that create lepton flavour violation in SM?

3. Are there more than one Higgs particle, and are there Higgs-operated processes that create CP violation?

The high number of $B\bar{B}$, as well as the high efficiency reconstruction of B, D and τ provide a unique opportunity to explore these questions.

Moreover, there are some questions that Belle II aims to address in other areas of physics such as if dark matter can be produced through e^+e^- collisions at the 10 GeV scale, possibly through a dark sector. The high integrated luminosity will allow for high statistics measurements of possible low cross section new processes.

2.4 Interactions in the ECL

As particles cross the ECL, their energy is deposited in the CsI(Tl) crystals through electromagnetic and hadronic processes. In this section, the different processes of interest by which particles deposit their energies are described.

Electromagnetic Showers

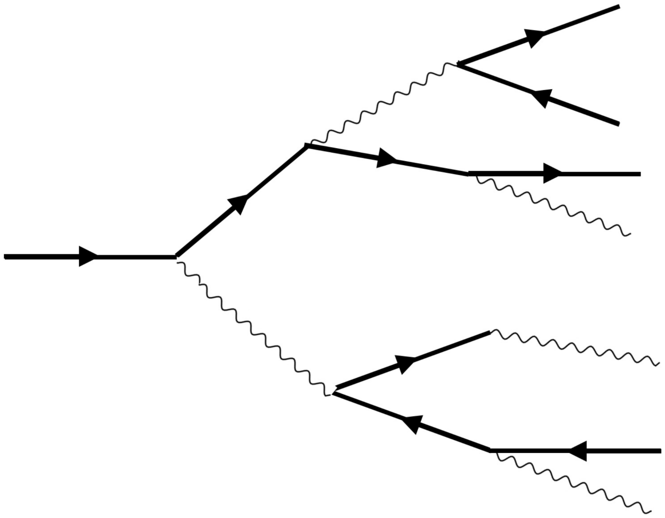


Figure 2.7: A schematic of an electromagnetic shower. Arrows pointing right are electrons, arrows pointing left are positrons and wavy lines are photons. Image from [13].

In the energy region of Belle II, lighter particles such as electrons and photons will have their energy depositions dominated by showering. The lighter nature of electrons mean their energy loss is dominated by bremsstrahlung radiation while the energy loss of photons is dominated by pair production and Compton scattering. This causes a chain reaction in the material where an incoming photon creates an e^+e^- pair, whose bremsstrahlung radiation produces more photons that will undergo further pair production. Figure 2.7 illustrates this principle. Once the particle energies are small enough, ionization and the photoelectric effect start to dominate. The energy is thus absorbed in the material which will re-emit it through scintillation photons. The energy of showers can propagate through multiple nearby crystals and be recorded to identify the energy of electrons and photons.

Heavier charged particles such as muons and pions have a more diverse foot print of energy deposition. Their energy deposition are not dominated by bremsstrahlung and contain significantly more contributions from ionization than electrons. This means that heavier particles deposit less energy per momentum than electrons. Particles like muons will often leave ion trails and continue through the ECL to be measured by the KLM.

2.4.1 Hadronic Showers

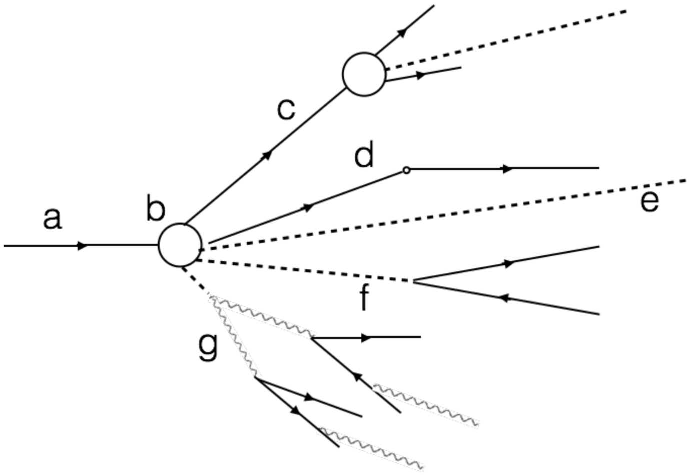


Figure 2.8: A schematic of a hadronic shower. Arrows pointing right are particles, arrows pointing left are their anti-particles, wavy lines are photons and dotted lines are non interacting particles. Image from [13].

When particles are composed of quarks, they can also interact hadronically with the nucleus of the material. This allows both neutral and charged hadrons to deposit energy in the ECL. Here are some possibilities of interactions happening during a hadronic shower:

- A high energy charged hadron is produced, and either will travel while ionizing and interact hadronically somewhere else (c), or scatter and continue through the material (d).
- A long lived neutral hadron is produced that is not interacting and goes straight through the material without interacting (e). Note that long lived neutral hadrons could interact in the material as seen in (a).
- A short lived neutral hadron is produced (e.g. K_S^0) that will decay into a particle pair leaving either an ionization trail or an electromagnetic shower (f). Decays into three or four particles also exist but tend to have smaller cross sections.
- A short lived neutral hadron is produced (e.g. π^0) that will decay into photons and produce an electromagnetic shower (g).

Because of the many possibilities for secondary particles out of hadron showers to produce electromagnetic showers, hadronic showers are always accompanied by a electromagnetic contributions. The different nature of the hadronic particle interactions will have a different scintillation signature in the CsI(Tl) crystals which can be captured to provide particle identification information. Section 3.4 describes the novel pulse shape discrimination (PSD) tool that captures these differences in the ECL crystals.

Chapter 3

ECL and its Modelling for Physics

As is common in high energy physics (HEP) experiments, the modelling and simulating of detectors is an integral part of physics experimentation. Simulations of particle interactions with Belle II allow collaborators to better understand the behaviour of the detector by providing them with a way to obtain observed results that can trace back to "truth" values — the original particle's observables.

To perform accurate simulations, Belle II has to be accurately described inside simulation softwares. This section covers the working of the ECL detector, and its modelling in the GEANT4 software framework used to simulate the Belle II detector's interaction with matter [14] [15]. It further covers calibrations done to increase the quality of ECL related tools.

Contributions

Section 3.3.2 describes the solving of overlap issues between the simulated ECL detector and the simulated ARICH detector caused by the recent re-alignment of the ECL in the early part of 2021.
Section 3.3.3 describes the implementation of functions to create a bounding box for the various custom GEANT4 objects created for the ECL modelling through custom GEANT4 classes.
Section 3.4.2 describes the calibration of the novel Pulse Shape Discrimination (PSD) tool developed in the Belle II calorimeter.

3.1 General

The ECL subdetector is made of 8736 CsI(Tl) crystals arranged in a cylinder (barrel) and two endcaps (forward and backward) around the interaction point. Figure 3.1

shows a profile of the ECL crystal arrangement. Once assembled, the ECL is 3 m in length with an inner radius of 1.25 m [11]. The calorimeter covers the polar angle region of $12.4° < \theta < 155.1°$, except for two $\sim 1°$ holes between the endcaps and barrel [11] where the polar angle θ is measured from the z-axis. The z-axis is defined as the Belle II solenoid axis which is the bisector of the two beams with the positive direction pointing along the electron trajectory. The x-axis is defined as the other horizontal axis in the laboratory [16].

There are a total of 98 crystal shapes, all of which are truncated pyramids. Their size varies; the typical dimension of barrel crystals having a cross section of 5.5×5.5 cm^2 (front) and 6.5×6.5 cm^2 (back) and the endcap crystals having dimension varying from 4.5 cm to 7.1 cm (front) and 5.4 cm to 8.2 cm (back) [18]. The average length of the crystals is 30 cm [18]. Each crystal is equipped with a PIN photodiode for scintillation light detection from showering particles [11].

3.2 Data Collection

This section details the procedure by which the ECL detector collects, saves and reconstructs data to higher level objects. It covers a description of the electronics used to collect the raw measurements (waveforms), and an overview of the main higher level objects reconstructed through the Belle II analysis software (basf2).

3.2.1 Electronics and Waveforms

The electronic readout of each ECL crystal includes two principal modules. Figure 3.2 shows the scheme for the ECL crystal electronics system. The diagram includes two modules:

1. The ShaperDSP, which contains shaper circuits and DSP (Digital Signal Processing) circuits such as an ADC (Analogue to Digital Conveter) circuit and an FPGA (Field-Programmable Gate Array). The ShaperDSP module takes an input signal from the crystal readout, integrates it, filters it and shapes it to obtain the CsI(Tl) integrated emission. The shaped signal is then digitized by a 1.76 MHz flash-ADC into 31 ADC points, making the time distance of ADC points on the order of 0.5 μs. This 31-length array, called a waveform, is sent to a FPGA for fitting and pre-processing [11] [19].

2. The Collector has for the purpose of grouping waveforms by trigger signal,

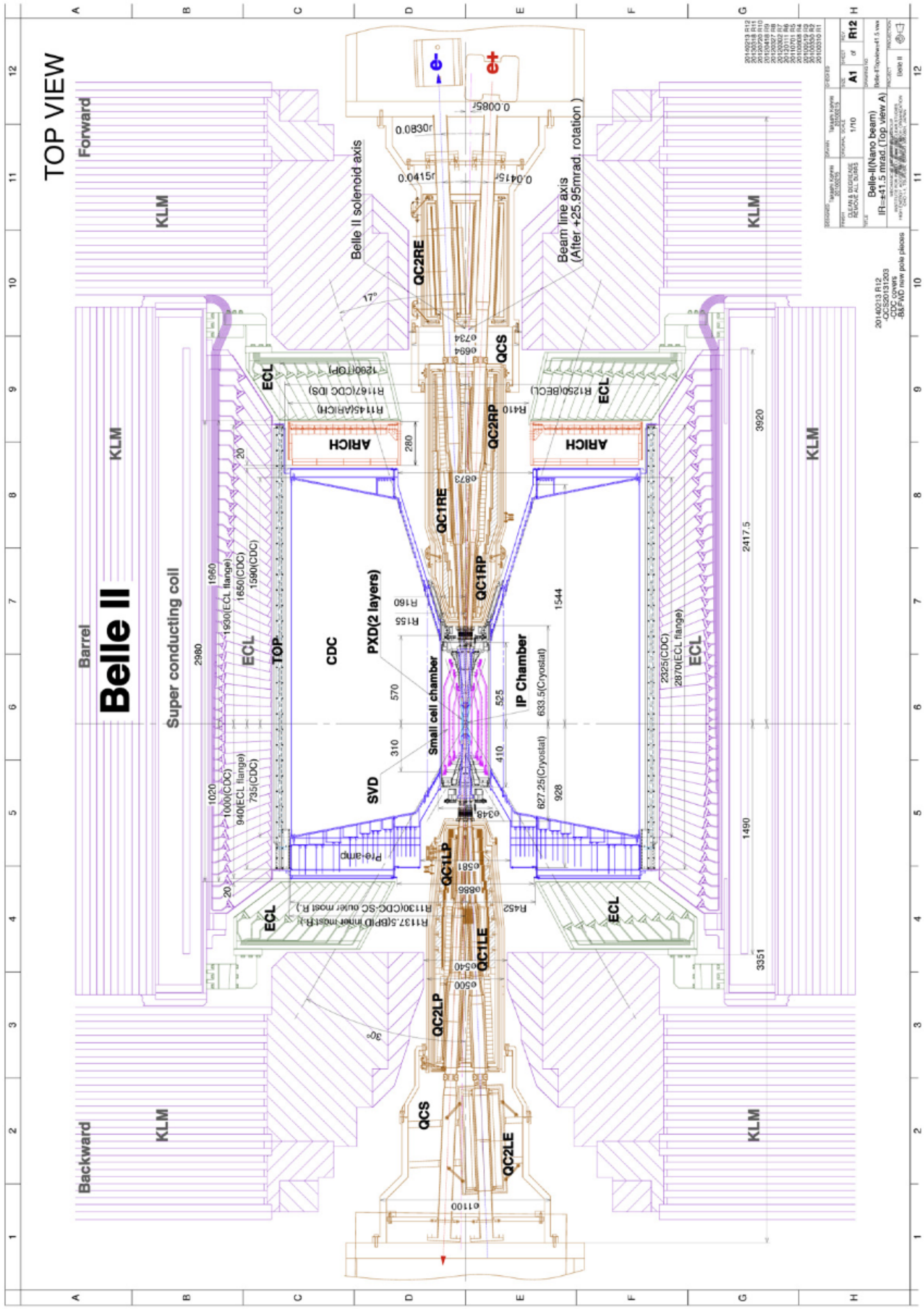


Figure 3.1: A profile of the ECL detector which is arranged cylindrically around the interaction point (IP) where collisions occur. The ECL has a barrel, a backward endcap (on the left) and a forward endcap (on the right). Image is from [17].

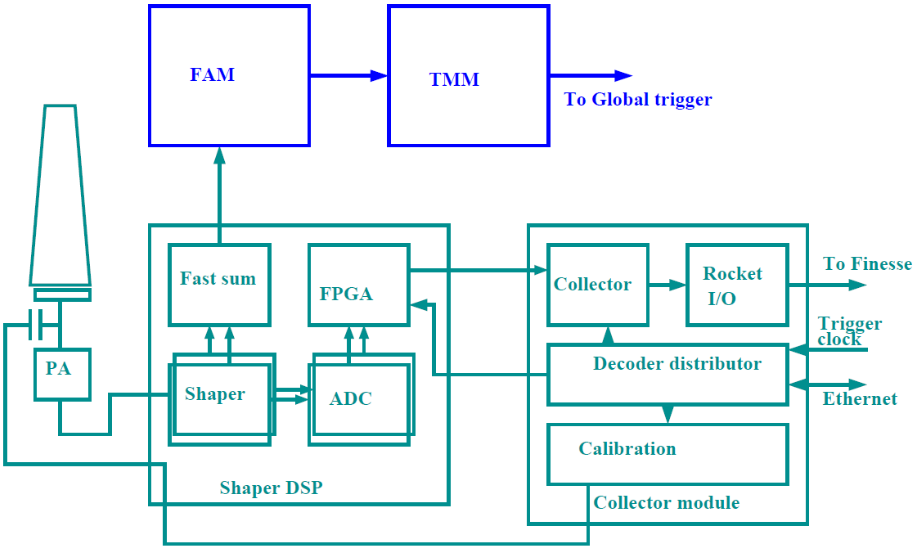


Figure 3.2: Block diagram explaining the main components of the Belle II ECL CsI(Tl) readout system. Image is from [11].

pack the data and send it down the pipeline towards the HLT (High Level Trigger) system through the FINESSE (Front-End Instrumentation Entity for Sub-Detector Specific Electronics) where it will undergo further processing and trigger test [11] [19].

The ShaperDSP module has two types of shapers, a fast one and a slow one. The signal from the fast shaper is sent in parallel towards the internal ECL trigger system composed of a FAM (Flash-ADC Trigger Module) and a TMM (Trigger and Monitoring Module) to allow the rejection of unwanted/noise signals. Waveform sampling can be triggered by different types of events and mainly serves as a way to select physics events, eliminate unwanted background, and select wanted background for further analysis [20].

3.2.2 Digits and Calibrated Digits

The raw waveforms obtained by the ECL are then reconstructed to higher level objects.

The first objects of interest are the “ECLDigits” and “ECLCalDigits”, which are the grouping of raw information in crystal-level information. Waveforms from a crystal are fit to obtain integrated amplitudes, times and fit qualities which are then calibrated to obtain variables such as crystal energy, crystal time of measurement, resolution of energy, time, etc. Information about the quality of the fit used to obtain the physics variables is saved as a χ^2 value [21]. As will be described in section 3.4, the ECLCalDigits also contain pulse shape discrimination (PSD) information.

3.2.3 Showers and Clusters

The ECLCalDigits obtained through the reconstruction of raw information usually originate from processes and events that may be measured by more than one crystal. The process by which the highest level ECL objects, the clusters, are obtained is as follows [21]:

1. ECLCalDigits are grouped into connected regions (ECLConnectedRegions objects) based on the energy spread across adjacent crystals.

2. The connected regions are formed from multiple showers, described through ECLShower objects, which are reconstructed using hypotheses of the particle's origin based on the connection of certain high energy crystals with tracks.

3. Further correction and calibrations are applied on ECLShowers to properly calculate their physical attributes.

4. Finally, a clustering algorithm creates ECLClusters objects (refered to as clusters) to group compatible showers into a single object which is loosely related to the previously defined connected regions.

ECLClusters contain cluster-level information such as: the total energy deposited in the ECL within the cluster, the energy distribution between the crystals of the cluster, the shape of the cluster (shower-shape variables), the calibrated timing of the ECL shower, etc.

ECLClusters are meant to be used directly by collaborators as they contain the completely reconstructed particle interaction with the ECL. Clusters are generally related to a single particle interaction, though overlaps of multiple showers coming from different particles, as well as single particles creating more than one shower outside of a connected region (and hence more than one cluster) are possible.

3.3 GEANT4 Geometry

The simulation of physics interactions with the Belle II detector is an integral part of the physics program. To obtain accurate simulations, the geometry of the Belle II detector was implemented in the GEANT4 software framework, which simulates the interaction of particles with the detector.

3.3.1 GEANT4 Design

Like the real detector geometry, the GEANT4 design of the ECL contains three section: the barrel, the forward endcap and the backward endcap. All 8736 CsI(tl) crystals are implemented with the GEANT4 "Belle Crystals" custom class written for the GEANT4 implementation in the Belle II software framework. There are also a variety of support structures holding the ECL in place which are implemented mainly through general GEANT4 shapes, and the "Belle Lathe" custom class. A visualization of the entire ECL detector can be seen in figure 3.3.

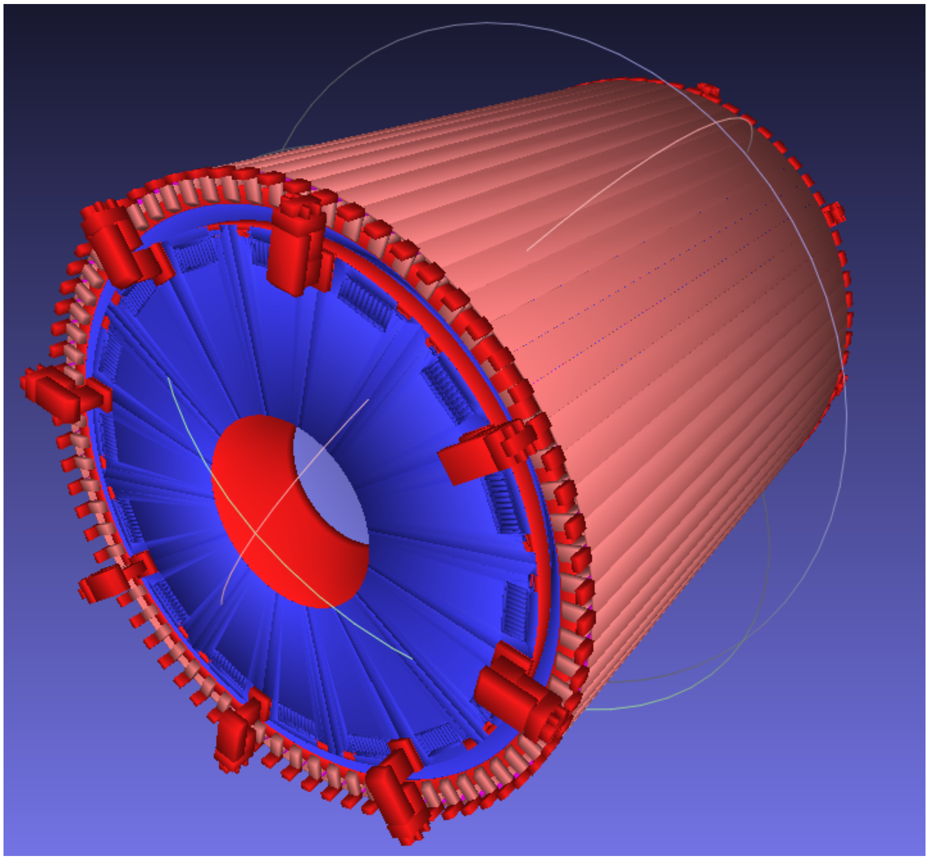


Figure 3.3: Visualization of the full ECL detector and its support structure as seen through the GEANT4 software.

While the implementation of the ECL in GEANT4 is based on the original ECL design, shifts in the ECL position due to the installation process or other causes will create a discrepancy between the real detector's position and the implementation. This means that the GEANT4 implementation needs occasional changes to be realigned with the current ECL position.

The current realignment method consists of finding the Data-MC discrepancy between the energy of the highest energy crystal in a cluster vs the spherical angle of impact of the cluster in muon events. Because the energy of the highest energy crystal is sensitive to crystal transitions, crystal positions can be compared between the two data sets and the discrepancy measured.

Conceptually, a fit is performed to find the optimal alignment of the ECL implementation to minimize this discrepancy. The whole ECL implementation, that is the active detector elements and the support structures, is then shifted by the new alignment distance.

3.3.2 Changing the Geometry of the ECL Inner Support Ring Structure

The alignment of the ECL created an overlap between the ARICH detector (shown in figure 3.4a) and the ECL inner support ring (shown in figure 3.4b). A visualization of the combined shapes is shown in figure 3.4c.

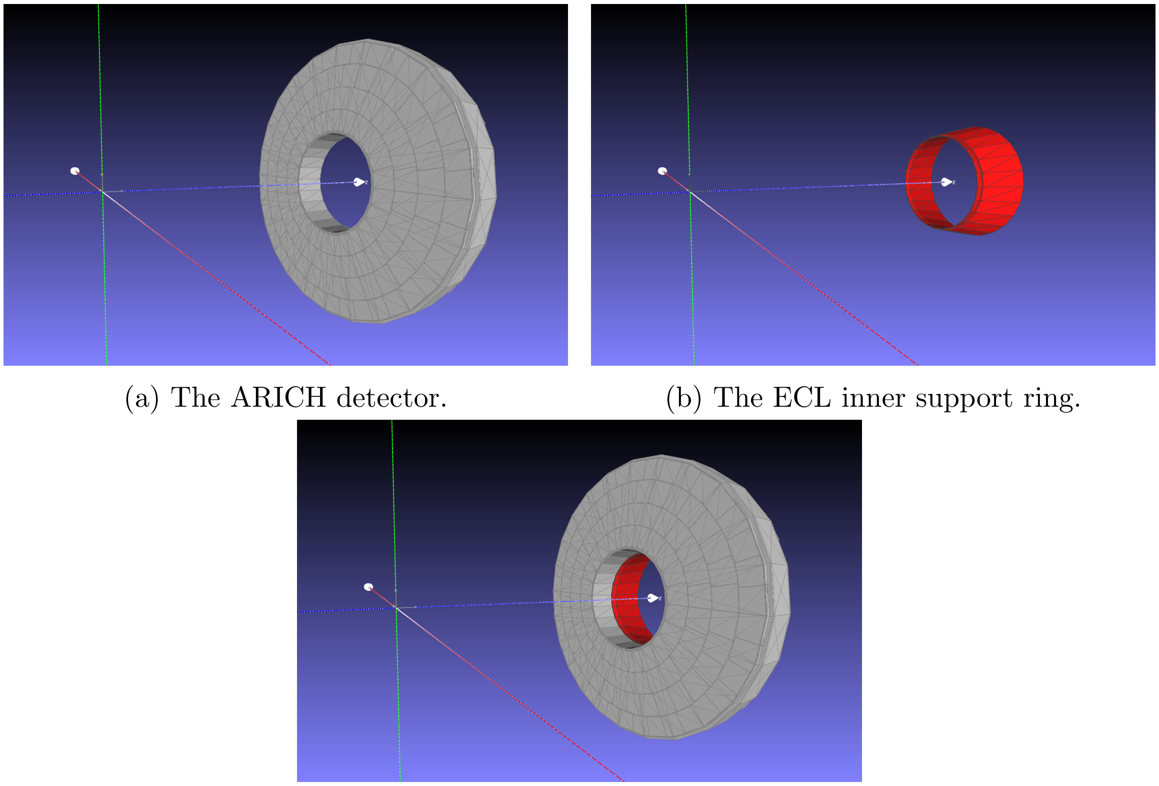
The overlap occurs in the y-axis direction, which points up in the laboratory, in the region where the ARICH and ECL inner support ring are adjacent. Due to the tight space in the detector's implementation, it is not possible to move any of the overlapping pieces by themselves without creating new overlaps, and as such a change in the geometry of the shapes was required.

Figure 3.5 shows a profile of the ECL inner support ring overlayed with the modified profile of the ECL inner support ring used to remove the overlay with the ARICH detector. The modification shown is a 1.95mm shift in the thinner part of the shape towards the inner radial direction.

3.3.3 Implementing Custom Boundary Box Algorithms

This section describes the contribution made to the Belle II software by implementing an algorithm to find a bounding box capable of encapsulating the custom shapes defined in the Belle II GEANT4 implementation.

This function is part of the required functions implemented by all geometrical object classes in the GEANT4 software framework version 10.6. Since the Belle II implementation uses custom classes, these functions had to be manually implemented



(a) The ARICH detector.

(b) The ECL inner support ring.

(c) The ARICH and ECL inner support ring.

Figure 3.4: Visualizations of the ECL inner support ring and of the ARICH detector as they are positioned in the GEANT4 design.

after updating the software version.

By adding a function capable of creating a bounding box, certain tasks within GEANT4 can be optimized. One such task is the detection of overlap between two shapes, which is generally done by randomly generating a number of points on the surface of one shape and calculating if any of these points lies within the volume of the other shape. Since shapes whose bounding boxes do not overlap cannot themselves overlap, the longer verification requiring the generation of points can be avoided provided the bounding boxes of the shapes do not overlap which optimizes the overall process.

The function takes as input two 3D-vectors, and assigns to them the positions corresponding to two opposite corners of a box containing the shape. The box edges are aligned with the x,y and z axes, and the vectors returned point to the box corners with the smallest and largest distance to the origin.

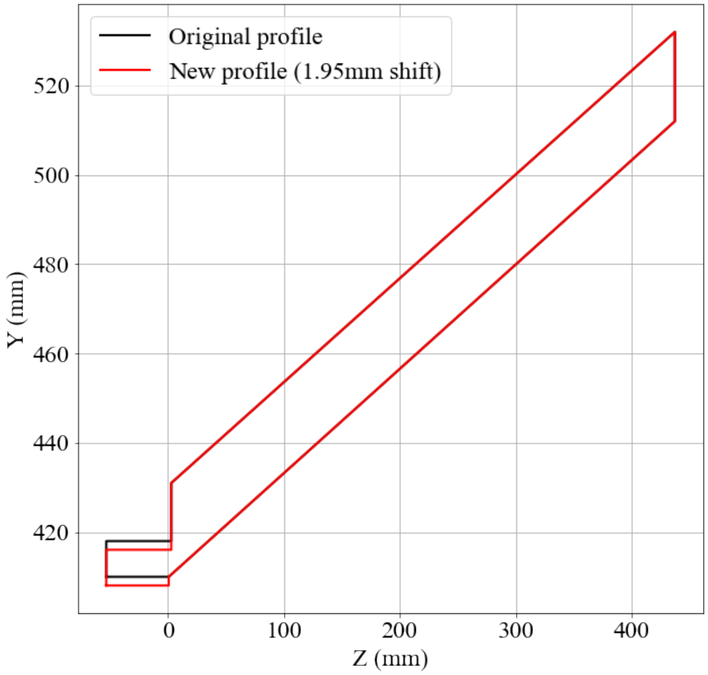


Figure 3.5: The profile of the ECL inner support ring overlaid with the new profile whose inner part is shifted radially inwards.

Belle Crystals

The ECL crystals are implemented within the “Belle Crystal” class. The shape is a truncated pyramid with four sided caps. Figure 3.6 shows an example shape defined by the Belle Crystal class.

To find the bounding box of the crystals, it is only necessary to know the vertex positions of the shape. The vertices are looped through to find the largest coordinate in each axis (these will form one of the returned vectors), and the smallest coordinate in each axis (these will form the other returned vector). This simple algorithm ensures that the totality of the shape is within the returned box.

Belle Lathe

The Belle Lathe shape is more complex. It is defined by providing a profile in the cylindrical r-z plane, that is then rotated around the z-axis by a given angular distance. To allow a more general definition, a starting angle is given in addition to the

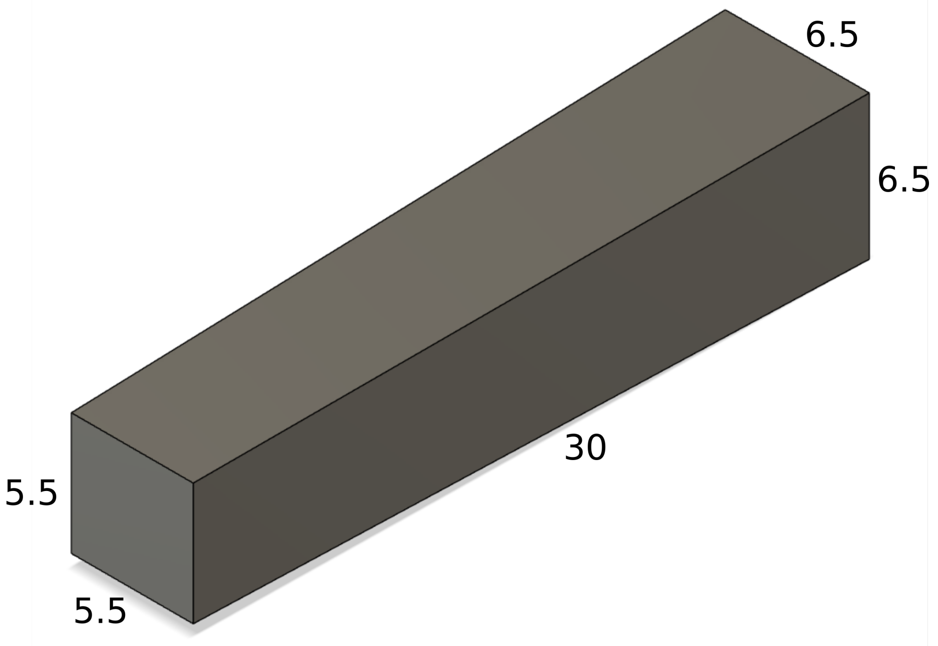
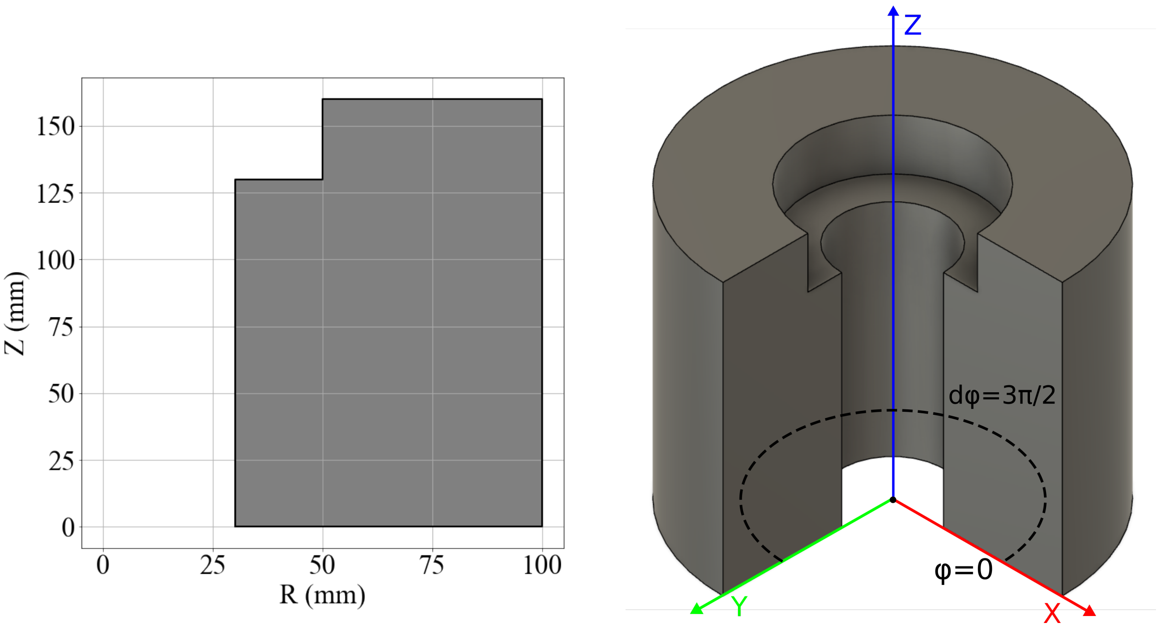


Figure 3.6: An example shape as defined by the Belle crystal class. The shape is a truncated pyramid with four sided caps. All dimensions are given in cm.

rotation angle to allow shapes starting with an angular offset from the x-axis. Figure 3.7a shows an example of a Belle Lathe shape profile. Figure 3.7b shows the shape obtained by rotating the profile by $d\phi = 3\pi/2$ from an initial angle $\phi = 0$ measured from the x-axis.

This shape is most commonly used to describe support structure in the ECL detector. As an example, the inner support structure shown in figure 3.4b is defined through the Belle Lathe class.



(a) Profile of the Lathe.

(b) Rotated profile of the Lathe.

Figure 3.7: An example shape as defined by the Belle Lathe class. This shape is defined as a profile that is rotated along the z-axis by $d\phi = 3\pi/2$ measured from the x-axis. The initial angle $\phi = 0$.

Because of the cylindrical nature of the shape, the extremum of the bounding box might not be at a vertex, such that a more complex method is needed to find the bounding box. Furthermore, the rotation of the profile may be small enough to allow the center of the cylinder to be an extremum of the bounding box.
An algorithm is found by noting the following:

- The z-axis range of the bounding box will always be the lowest to largest z value in the profile.
- Each time the shape crosses the x-axis or y-axis during its rotation, this crossing point will be an extremum of the bounding box along the given axis.
- The remaining extremum will be at inner or outer vertices depending on where the shape begins and ends its rotation.

The algorithm is implemented as follows:

1. Find the outer vertices by using

$$x_1 = r_{max} \cos(\phi) \qquad y_1 = r_{max} \sin(\phi)$$
$$x_2 = r_{max} \cos(\phi + d\phi) \qquad y_2 = r_{max} \sin(\phi + d\phi)$$

   where r_{max} is the maximum R-axis value in the given profile, ϕ is the anti-clockwise angle from the x-axis at which the rotation starts and $d\phi$ is the counter-clockwise rotation distance. Since the z-axis values of the bounding box are directly obtainable from the profile, only the x-y coordinates of the vertices are needed.

2. Find the inner vertices by using

$$x'_1 = r_{min} \cos(\phi) \qquad y'_1 = r_{min} \sin(\phi)$$

   where r_{min} is the smallest R-axis value in the given profile. If $r_{min} = 0$, there can only be a single inner vertex at $x = 0$, $y = 0$. If $r_{min} > 0$, then there will be a second inner vertex at

$$x'_2 = x_{min} \cos(\phi + d\phi) \qquad y'_2 = x_{min} \sin(\phi + d\phi).$$

3. Find which axes are crossed by looking at the angular span of the shape's rotation. An axis is crossed if the angle of the axis (e.g. $\pi/2$ for the positive y-axis) is greater or equal to the starting angle, but smaller or equal to the stopping angle when normalizing to $\phi = [0, 2\pi)$ and $d\phi = [0, 2\pi)$. Define an axis crossing at a distance of r_{max} on the crossed axis.

4. Loop through the list of vertices and axis crossings to find the largest coordinate in each axis (these will form one of the returned vectors), and the smallest coordinate in each axis (these will form the other returned vector).

The algorithm to return the bounding box for the Belle Crystal (3.3.3) and Belle Lathe (3.3.3) shapes were programmed into the Belle II GEANT4 implementation to be used during further simulations.

3.4 Pulse Shape Discrimination

In the years immediately preceding this thesis, a novel pulse shape discrimination (PSD) tool was developed in the Belle II calorimeter [22]. This section will describe the PSD tool as much of the work done in this thesis builds upon it.

3.4.1 General

The pulse shape discrimination of the ECL crystals' signal at Belle II is a technique used to extract information about the nature of the particles and showers that create the observed signals.

The waveforms captured in the ECL detector, as detailed in 3.2.1, have certain characteristics such as rise and fall times and shape of the pulse that depend on the nature of the interaction between the showering particle and CsI(Tl) crystals in which it showers. In particular, hadronic and electromagnetic showers have distinct pulse shapes.

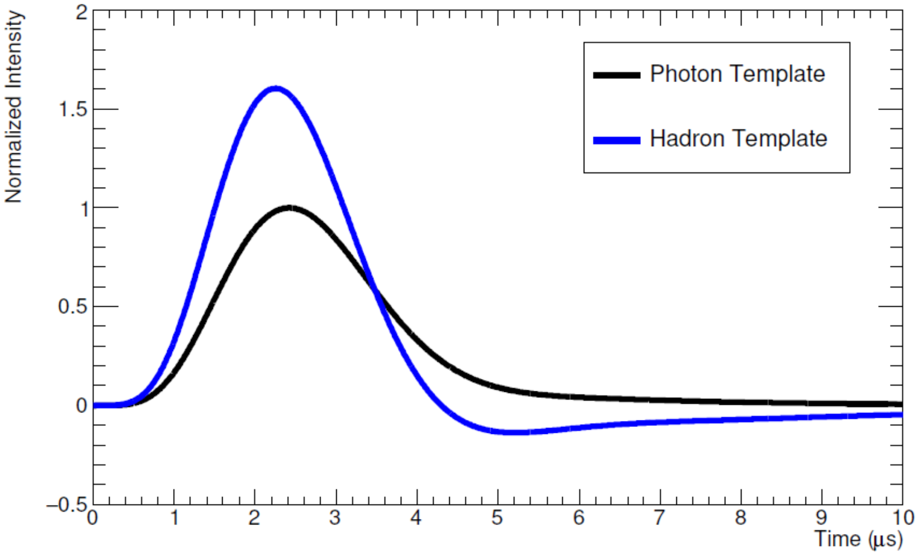


Figure 3.8: The different templates used with the PSD tool to fit waveforms. The photon template fits electromagnetic showers, while the hadron template fits hadronic showers.

This information can be extracted by fitting the observed pulse shapes with templates representing hadronic and electromagnetic showers respectively. Figure 3.8 shows an example of the templates used in the fitting of waveforms for PSD information. A noticeable feature includes the negative “dip” after the peak of the hadronic pulse where the signal becomes negative.

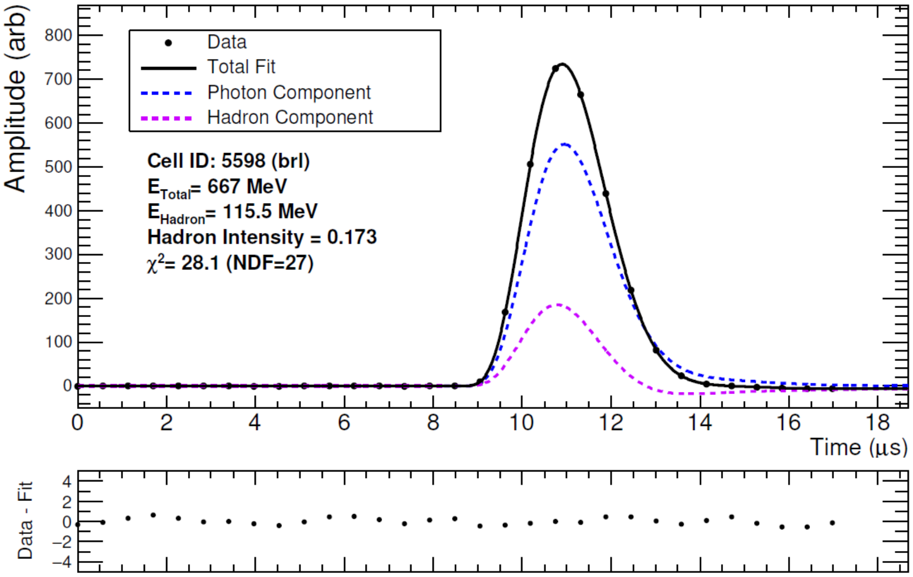


Figure 3.9: The fit results of a waveform containing both hadronic and electromagnetic (photon) components. Image is from [22].

Figure 3.9 shows an example waveform containing both hadronic and electromagnetic components, and the resulting fit of both templates. The hadron intensity is a ratio of the hadronic energy contribution to the shower by the total energy of the shower, and contains the information about the nature of the shower that produced the signal.

A purely electromagnetic shower, which could be obtained from a photon shower, will have a hadron intensity of 0, as it contains no hadron component. Hadronic showers will have both an electromagnetic component and a hadronic component as the interaction produces both types of showers simultaneously.

The PSD templates are fit to the waveform by minimizing the χ^2 defined in equation 3.1

$$\begin{aligned}\chi^2 &= \mathbf{D}\mathbf{C}^{-1}\mathbf{D} \\ \mathbf{D} &= \mathbf{S} - \mathbf{G}\end{aligned} \tag{3.1}$$

where $\mathbf{S} = s_1 ... s_{31}$ is the waveform, $\mathbf{G} = g_1 ... g_{31}$ is the template and $\mathbf{C}^{-1}$ is the noise matrix defined as the inverse of the covariance matrix $\mathbf{C}$.

The covariance matrix $\mathbf{C}$ is calculated using equation 3.2

$$C_{(i,j)} = \frac{1}{n-1}\sum_{k=1}^{n}(S_{k,i} - \bar{S}_i)(S_{k,j} - \bar{S}_j) \tag{3.2}$$

where

$S_{k,i} = \mathbf{S}_k = s_{k,1}...s_{k,31}$, the k^{th} waveform out of n waveforms.

$\bar{S}_i = \frac{1}{n}\sum_{k=1}^{n}(s_{k,i} - b_i)$ with $b_i = \frac{1}{n}\sum_{k=1}^{n} s_{k,i}$.

The variables are calculated on raw waveforms, and as such will be included into ECLCalDigit (see section 3.2.2) objects during reconstruction. The current implementation only records the PSD information of an ECLCalDigit if its waveform has and energy above 30 MeV, and if the fit has a $\chi^2 < 60$ (27 NDF) [22].

3.4.2 Calibration of the PSD tool with Collision Noise

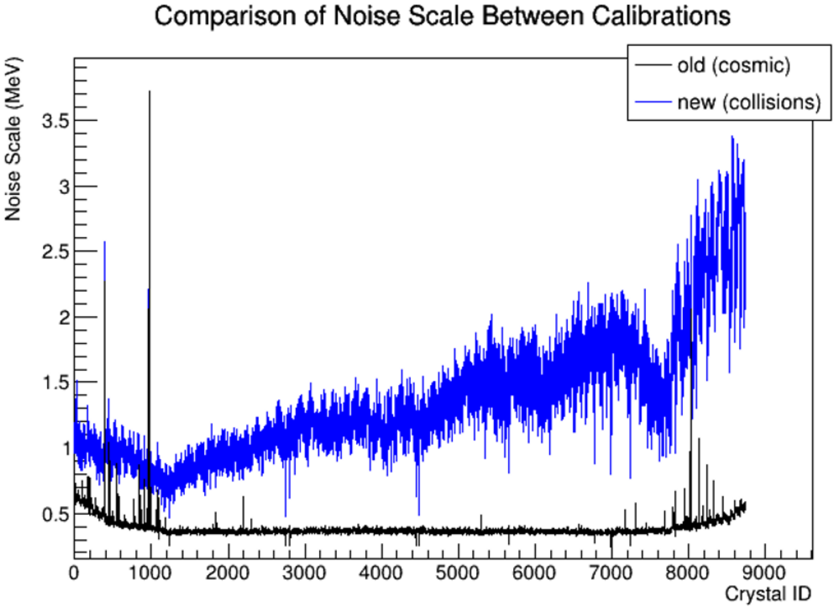


Figure 3.10: Comparison of the noise scale from the old calibration (electronic and cosmic), and the new calibration (collisions). The x-axis labels all 8736 crystals.

Since the fitting procedure of the waveforms relies on a covariance matrix, it is important that the covariance matrix be calibrated to best match noise waveforms from the experiment. The first calibration was performed before the Belle II experiment started collecting collision data, and as such needed to be updated by a new calibration once the dataset of collision noise waveforms was available. Figure 3.10

shows the difference between the noise scale of Belle II before and after collision noise was introduced.

To calculate the new covariance matrix, noise waveforms from a randomly selected data set are used with equation 3.2. A new database object is created containing the inverted covariance matrix information, and uploaded to the Belle II database to be used as the new default PSD noise matrix. For this calibration, run 2062, experiment 8 was selected.

PDG ID	Momentum (GeV)	θ (cos isotropic)	ϕ (isotropic)	N
22	[0-5]	[0-180]	[0-360]	200 000

Table 3.1: Kinematic values of the simulated photons where θ is the polar angle as measured from the z-axis and ϕ is the azimutal angle measured from the positive x-axis.

To measure the performance of the new calibration, photon events were simulated to interact with the Belle II detectors, and background noise was overlaid from the different available experiments. Table 3.1 shows the important kinematic values of the simulated photons.

The effects of the calibration are measured with the following quantities:

- $N^{failed}_{\%recovered}$: The percentage of clusters without PSD information due to failed fit or small energy that acquired PSD information after calibration. Defined as

$$N^{failed}_{\%recovered} = 1 - N^{failed}_{calibrated}/N^{failed}_{non-calibrated}. \tag{3.3}$$

- There are two comparisons done using the number of crystals with hadron information within a cluster.
  - $N^{digitN}_{\%changed}$: The percentage of clusters that saw a change in their number of crystals with hadron information. Defined as

$$N^{digitN}_{\%changed} = N^{digitN}_{changed}/N_{total}. \tag{3.4}$$

  - $N^{zero-digitN}_{\%recovered}$: The percentage of clusters that saw a change from zero hadron crystal to a non-zero hadron crystal number. Defined as

$$N^{zero-digitN}_{\%recovered} = 1 - N^{zero-digitN}_{changed}/N^{zero-digitN}_{non-calibrated} \tag{3.5}$$

where $N^{zero-digitN}_{changed}$ is the number of clusters that saw their number of hadronic crystals go from zero to a non-zero number after calibration.

- $N^{MVA}_{\%changed}$: The number of clusters that saw a change in the value predicted by the PSD MVA. The PSD MVA is trained to discriminate between neutrally charged clusters created by hadronic and electromagnetic showers. Since the MVA is sensitive to the hadron information of the crystals, this metric is sensitive to changes in hadronic information that are too small to affect the other metrics. Defined as

$$N^{MVA}_{\%changed} = N^{MVA}_{changed}/N_{total}. \tag{3.6}$$

Calibration performance

To show the performance of the calibration, the difference between the old and new calibrations for the PSD metrics of the simulated photons are shown. The photons were simulated with all available background conditions at the time, which are experiment 7,8,10 and 12. Important numerical values are shown in table 3.2.

Metrics	exp. 7	exp. 8	exp. 10	exp. 12
$N^{failed}_{\%recovered}$	66.9	66.7	65.4	60.6
$N^{digitN}_{\%changed}$	44.4	35.8	34.2	32.3
$N^{zero-digitN}_{\%recovered}$	31.9	26.6	25.5	24.6
$N^{MVA}_{\%changed}$	83.9	87.8	87.6	87.1

Table 3.2: Numerical effects of the calibration on the photon samples generated with different experiment background conditions.

The graphical results for the simulations using the background conditions of experiment 10 are shown in figures 3.11, 3.12 and 3.13.

Figure 3.11 shows the number of ECL clusters with PSD information as a comparison between the old and new calibrations. As shown in table 3.2, 65.4% of the clusters lacking PSD due to bad fits or low energy are recovered.

Figure 3.12 shows the number of crystals containing hadronic information per ECL clusters as a comparison between the old and new calibrations. The threshold for a crystal to be considered hadronic is to have $E_{Hadron} > 3$ MeV or $E_{Hadron}/E_{Total} >$

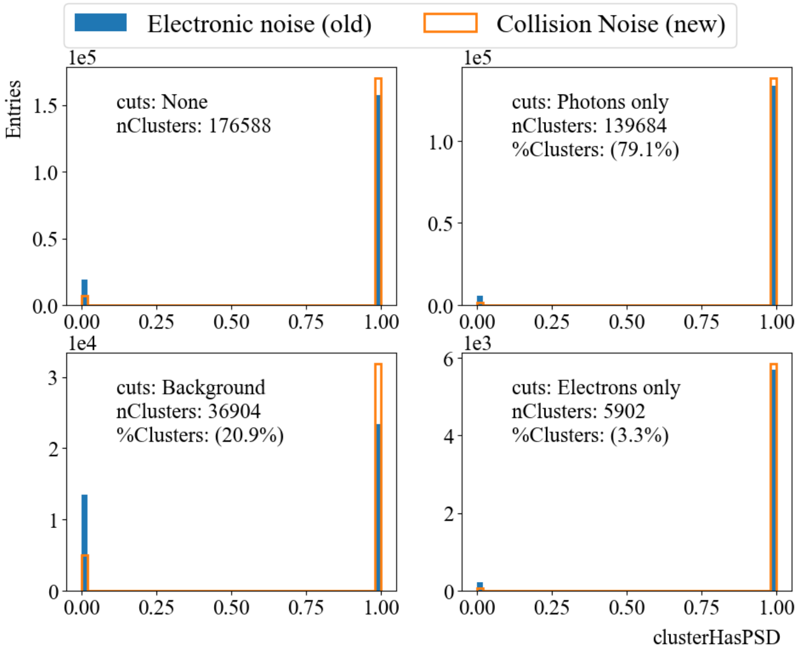


Figure 3.11: A binary distribution indicating if clusters contain PSD information in the photon sample simulated with the background conditions of experiment 10. Clusters without PSD information are labelled with 0. Clusters whith PSD information are labelled with 1.

0.005 where E_{Hadron} is the hadronic component energy. The number of crystals changes to be larger in all scenarios. The sub-plot with background cuts shows that a large number of clusters who previously could not be fitted can now be fit, but show no significant hadron contributions in their fits. Table 3.2 shows that 34.2% of clusters have their numbers of crystals change.

Figure 3.13 shows the neutral PSD MVA prediction of clusters. This MVA outputs a probabilistic prediction of the particle's nature. Particles near 0 are predicted to be hadronic, and particles near 1 are predicted to be electromagnetic. Clusters with no PSD information have a MVA value of -1. The calibration caused clusters with no PSD information to acquire PSD information, and clusters generally moved towards the extremum of the distribution showing an increase in the certainty of the MVA decisions.

Figure 3.14 shows the energy distribution of the clusters that recovered failed PSD information. Most clusters are from background events, or low energy events which highlights the higher rate of PSD failure in low energy events.

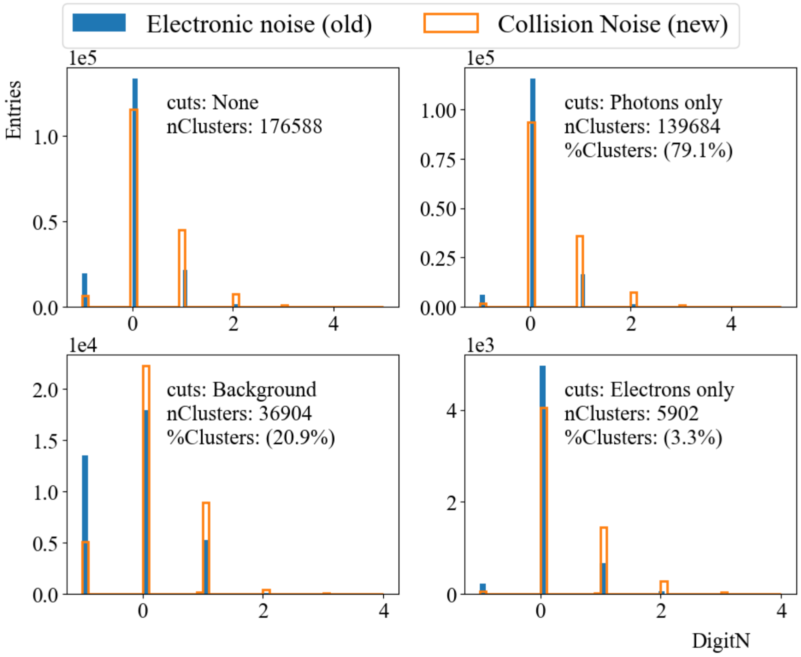


Figure 3.12: The number of crystals per cluster containing PSD information in the photon sample using experiment 10 background. Clusters without PSD information are labelled with -1.

3.5 Background Overlays

When simulating event interactions in the Belle II detector, having an accurate representation of the background conditions is essential to having an accurate simulation.

To include background waveforms in the ECL, the current procedure is to overlay a sample of real background data onto the simulation. Background data is collected for each run by sampling the ECL during data taking randomly at the time when the e^+e^- beams cross at the IP. These events usually do not have an actual e^+e^- collision.

This method provides accurate representation of noise in the ECL during simulations, but it requires the saving and usage of large files. This creates the problem of having to move these large files every time a background overlay needs to be used. This is an ongoing issue with the grid system used to run the Belle II simulations, whose file transfer capacity is limited.

As substantially larger samples of simulated events will be produced in the years to come, potential solutions are being investigated to solve this problem. One solution involves the production of low amplitude noise via machine learning techniques,

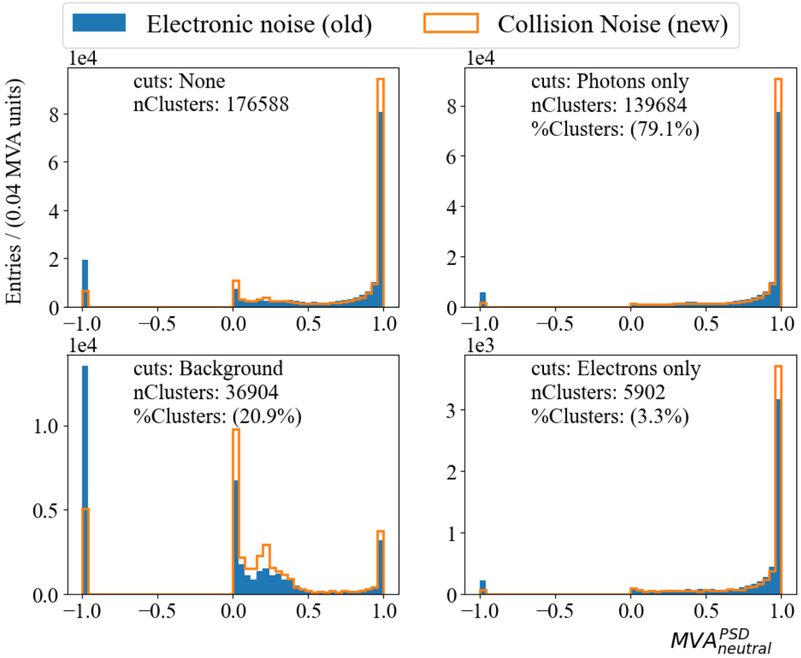


Figure 3.13: Neutral MVA distribution of the photon sample using experiment 10 background. The MVA is trained to identify neutral electromagnetic clusters with 1, and neutral hadronic clusters with 0. Clusters without PSD information are labelled with -1.

specifically the usage of generative adversarial neural networks. This project is discussed further in chapter 4.

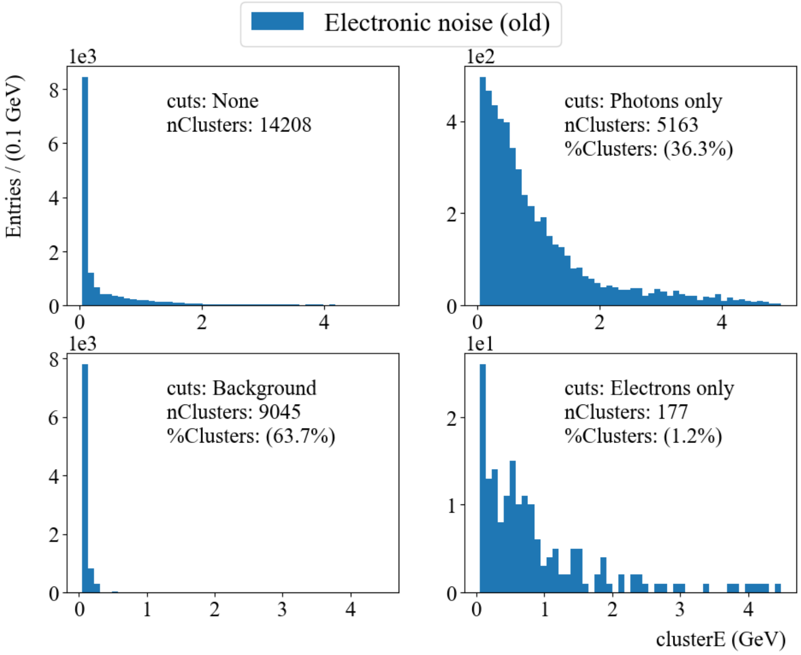


Figure 3.14: The cluster energy distribution of events whose failed PSD fits were recovered by the calibration for the photon sample using experiment 10 background.

Chapter 4

Waveform Generation Using GANs

This chapter covers the studies done to evaluate the viability of using generative adversarial networks (GANs) as tools to generate low energy noise waveforms to be used as background in simulations of the ECL detector.

Section 4.4 describes the procedure used to train GANs, and explains the framework created to build, train, and evaluate GANs. The framework uses the Keras library and Tensorflow library for management and training of the neural networks. Section 4.5 describes in detail the metrics designed to quantify the quality of GAN generated waveforms. It further displays a demonstration GAN built using a convolutional neural network architecture to showcase the possible quality of GAN generated waveforms.

4.1 Introduction and Motivation

The simulation of particle interactions with detectors is commonly used in the HEP field to obtain simulated data sets, which contain both the measured and truth values of the four-momentum particles produced in physics processes of interest. While performing a measurement or a search using the simulated samples, physicists can predict exactly which criteria will select the data in the way that allows their measurement or searches to be optimized for signal and that have a minimal amount of uncertainty.

To accurately simulate detector interactions, the background conditions present during data taking have to either be simulated or overlaid onto the simulated particle interactions in the detector. As described in section 3.5, overlaying noise waveforms onto simulations creates both large storage and large file transfer demands through the grid systems used to produce the simulations. This becomes a problem when grid system capabilities are not able to accommodate the demands. This project is concerned with the simulation of background conditions in the ECL detector as a way to replace the current method of overlaying background noise waveforms measured during data taking.

Multiple solutions are being explored to replace the current background overlay procedure. The approach explored within this thesis consists of replacing the background overlays with a mixture of saved high energy background events and randomly generated low energy noise using GANs. The work done in this thesis focuses on the generation of low energy noise using GANs.

4.2 GANs

GANs were first introduced in 2014 [23] as a way to improve the performance of generative neural networks, and use a competition-based training procedure. They have been studied in other areas of academia, where they have achieved high level of success in the generation of realistic images [24]. The term GAN stands for Generative Adversarial Network:

Generative: a machine learning model trained to generate a specific type of data. In this project, the generated data are noise waveforms from CsI(Tl) crystals (described in section 3.2.1).

Adversarial: a model trained through adversity (or competition) with another model; the exact procedure for this is described in details in section 4.4. This is in contrast with the more common training method of minimizing a loss (or cost) function that is used in most machine learning training. It is important to note that the individual models in the GAN are still trained using a loss function minimization, but that the model's real objective is to outperform the adversary model not to explicitly minimize the loss of generated waveforms.

Network: the models used are neural networks.

In summary, the key distinction between GANs and regular neural networks are their training framework. A GAN could possess the same network architecture as a non-adversarially trained model. Figure 4.1 shows the general structure of a GAN within its application in this thesis.

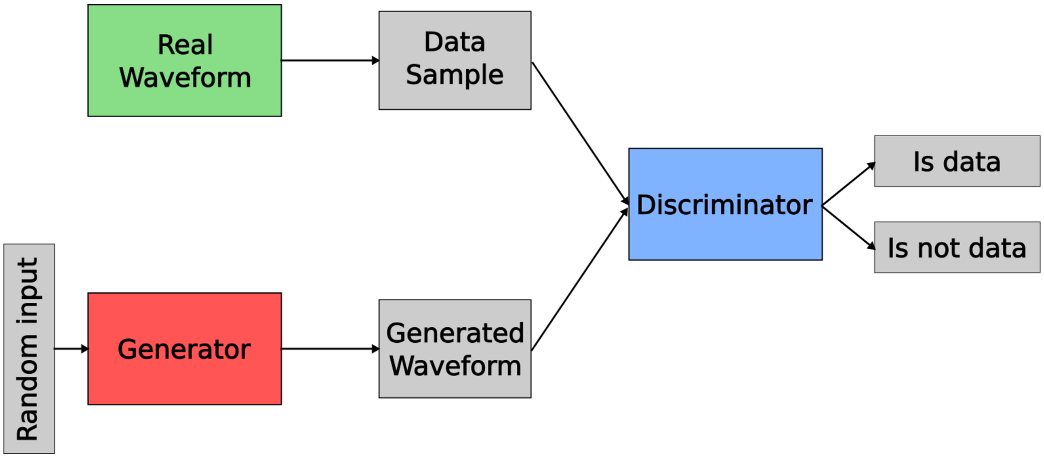


Figure 4.1: The structure of a GAN consisting of two neural networks. The generator (in red) takes as input a random array of numbers, and outputs a generated waveform. The discriminator (in blue) takes as input a waveform, either generated or from a data sample, and estimates the probability that the sample came from the training data rather than the generator.

4.3 Training Data

This section describes the data used to train the GANs.

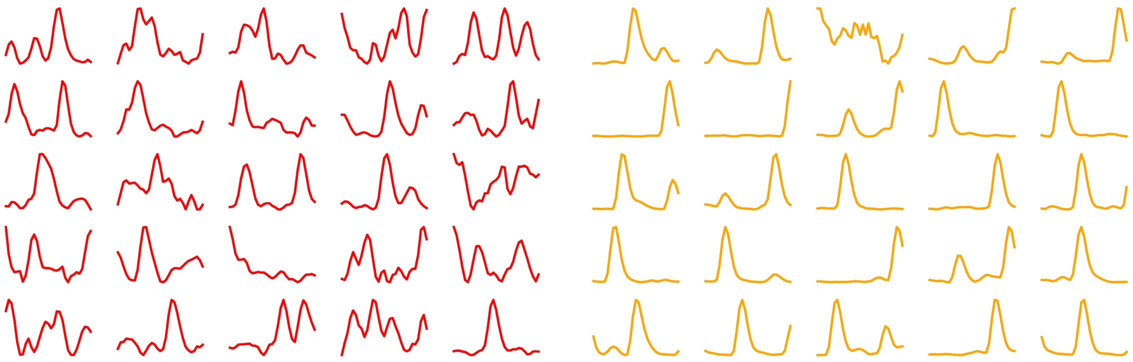
Selection

Waveforms from Belle II experiment 12, run 4941, crystal number 8571 (this crystal is in the backward part of the ECL) are used. Only the waveforms associated with random triggers and delayed physics triggers are kept to provide a background sample.

Waveforms are kept if the average value of their ADC values are < 6000, and if the distance between their maximal and minimal ADC values are < 1000. These cuts get rid of higher energy events which are often associated with physics events; these events are to be saved independently. Figure 4.2 shows a visual comparison between the low amplitude waves and the high amplitude waves. A qualitative distinction is observed where the high amplitude events have a smaller contribution from high frequencies and large peaks likely caused by physics events.

Normalization

A normalization is applied to the remaining waveforms to improve the performance and quality of the neural networks' training. First, a pedestal is subtracted from each



(a) Data waveforms with low amplitude differences.

(b) Data waveforms with high amplitude differences.

Figure 4.2: Comparison of the different types of data waveforms.

waveform ADC value using equation 4.1:

$$\text{Pedestal} = \frac{1}{N}\sum_{i=0}^{N} \bar{S}_i \tag{4.1}$$

where $\bar{S}_i$ is the average amplitude of a waveform and N is the total number of waveforms.

The waveform ADC values are then scaled to be between -1 and 1. Figure 4.3 shows the distribution of means. The distribution of waveform means was kept to preserve the overall amplitude feature of the noise. Since the waveforms may be overlaid on physics measurements, all observable features can have an impact on the measured quantities.

4.4 Training a GAN

In this section, the training procedure of GANs is described. A description of the general procedure used to train neural networks is detailed, followed by a description of the framework that was built to train the GANs adversarially.

4.4.1 Backpropagation

While training a neural network, the performance of a model is defined through a loss function that quantifies how different its output is from the target result. A common

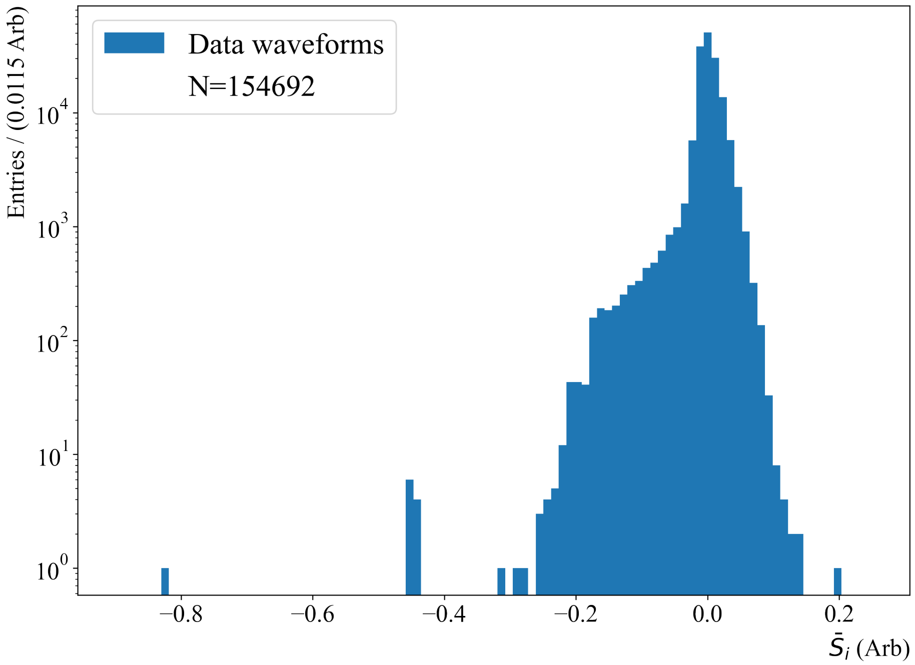


Figure 4.3: Distribution of waveform pedestals after normalization.

loss function used for neural networks with a unique output neuron is binary cross-entropy which is described by equation 4.2:

$$L(y, \hat{y}) = -\frac{1}{m}\sum_{i=0}^{m} \hat{y}_i \log(y_i) + (1 - \hat{y}_i)\log(1 - y_i) \tag{4.2}$$

where m is the number of events in the training batch, y is the values predicted by the model and $\hat{y}$ is the target label (true value). Loss is minimized when the prediction perfectly aligns with the target, and maximized when the prediction is farthest from the label.

The output of neural networks is composed of neurons, whose value is determined by neurons in previous layers passed through a linear function using weights and biases. An example is shown in equation 4.3 for a neuron x_j^l on the layer l whose value is determined by two neurons x_1^{l-1}, x_2^{l-1} on the previous layer $l-1$ weighted by their respective weights w_{j1}^l, w_{j2}^l and an overall bias b_j^l for the layer:

$$x_j^l(x_1^{l-1}, x_2^{l-1}) = w_{j1}^l x_1^{l-1} + w_{j2}^l x_2^{l-1} + b_j^l. \tag{4.3}$$

The weights are unique to each neuron connections, and the bias is unique to the output neuron.

The model is trained by updating the weights and biases of each neuron to minimize the loss function. This is achieved through gradient descent where the weights of each neurons are changed based on how much they affect the loss value. As an example, weights are updated proportionally to how they affect the loss function as seen in equation 4.4:

$$w_{ji}^{l} = w_{ji}^{l} - \mathrm{lr}\frac{\partial L}{\partial w_{ji}^{l}} \tag{4.4}$$

where lr, the learning rate, decides how aggressively the model is trained.

The gradient of the loss function for a given weight is found using the algorithm of backpropagation which uses the chain rule

$$\frac{\partial L}{\partial w_{ji}^{l}} = \frac{\partial L}{\partial x_{j}^{l}}\frac{\partial x_{j}^{l}}{\partial w_{ji}^{l}} \tag{4.5}$$

where the right hand terms are well defined within equation 4.2 and 4.3 respectively.

The gradient for weights on deeper layers is obtained using the chain rule as shown in equation 4.6:

$$\frac{\partial L}{\partial w_{ji}^{l-1}} = \frac{\partial L}{\partial x_{j}^{l-1}}\frac{\partial x_{j}^{l-1}}{\partial w_{ji}^{l-1}} \tag{4.6}$$

with

$$\frac{\partial L}{\partial x_{j}^{l-1}} = \frac{\partial L}{\partial x_{k}^{l}}\frac{\partial x_{k}^{l}}{\partial x_{j}^{l-1}}. \tag{4.7}$$

where k is summed over the layer l. The substitution from equation 4.7 provides two terms: one which is previously calculated in equation 4.5, while the other is well defined within equation 4.3. This recursive algorithm is effectively propagated backwards allowing the previously calculated gradients to be re-used until all weights are updated.

4.4.2 CNN

Convolutional neural networks are a variation of the fully connected neural network shown in section 4.4.1. The main characteristic of CNNs is that neurons share weights with one another. The weight vector is called the kernel.

Similarly to regular neural networks, a neuron array x^{l-1} of length n_{l-1} connects to another neuron array x^l of length n_l. Instead of having each neuron in x^{l-1} have a unique weight to neurons in x^l as shown in equation 4.3, we have an array of weights w^l of length m that is shared by the neurons.

The output neuron will be

$$x_j^l = \sum_{i=0}^{m} w_i^l x_{i+j}^{l-1} + b_j^l \tag{4.8}$$

where $m < n_{l-1}$. The different output neurons in layer l will all use the same weights, which shows that updating the weights in the kernel will cause a change in the output of multiple neurons.

The backpropagation algorithm for CNNs' weights is similar to the original one, but includes a few changes due to shared weights. For a loss function described by equation 4.2, we have that

$$\frac{\partial L}{\partial w_m^l} = \sum_{i=0}^{n_l} \frac{\partial L}{\partial x_i^l} \frac{\partial x_i^l}{\partial w_m^l} \tag{4.9}$$

where the x_i are summed for all neurons in the layer since they all have a contribution from the same weight kernel. The first right-hand term is well defined in equation 4.2, and the second term is well defined in equation 4.8.

Similarly, the gradient for deeper layers is found using equation 4.10:

$$\frac{\partial L}{\partial w_m^{l-1}} = \sum_{i=0}^{n_{l-1}} \frac{\partial L}{\partial x_i^{l-1}} \frac{\partial x_i^{l-1}}{\partial w_m^{l-1}} \tag{4.10}$$

with

$$\frac{\partial L}{\partial x_i^{l-1}} = \sum_{j=0}^{n_{l-1}} \frac{\partial L}{\partial x_j^l} \frac{\partial x_j^l}{\partial x_i^{l-1}} \tag{4.11}$$

where the first right-hand term is already calculated in equation 4.10 and the second term is well defined within equation 4.8.

Finally, an additional type of CNN operation is used called a transpose convolution. Transpose convolutions resembles an inverted convolution operation where each input neuron produces m outputs based on the length of the weight kernel w^l_m. This is used when the convolution operation needs to increase the dimensions of the input.

Convolutional Hyperparameters

Convolutional neural networks can be further costumized by choosing certain hyperparameters. The first one of interest is the kernel size of w^l_m, which simply determines how many input neurons will be included in the output on the next layer.

The second one of interest is striding, which determines how many inputs are skipped when going to the next output neuron. A striding of k would imply the kernel shifts by k neurons in the input layer for each output neuron. This is described in equation 4.12:

$$x^l_j = \sum_{i=0}^{m} w^l_i x^{l-1}_{i+kj} + b^l_j. \tag{4.12}$$

The last one of interest is the use of filters. Each filter adds a new kernel vector, making the weights vector a 2D matrix. Each filter will create a separate output when applied to the one dimensional neuron array used as input. The filter outputs will be summed up at the end of the convolutional operations to obtain the final output of the neural network as a one dimensional object.

The idea behind filters is that they individually may represent features of the input; e.g. a filter might look at the edges of the input only, while another might look at the high amplitude components only, while another might look at the low amplitude components only. In reality, the filters do not provide such a simplistic characterisation but do help the network specialize its weights.

Other Parameters

In a real implementation, an activation function is often added to equation 4.3 and 4.8. Within this thesis, four different activation functions are used: ReLu, leaky ReLu, tanh, and sigmoid. The activation functions are non-linear functions applied to the usual output result of a neuron. Figure 4.4 shows the different activation functions used where the neuron output is shown on the x-axis and the activated neuron output is shown on the y-axis.

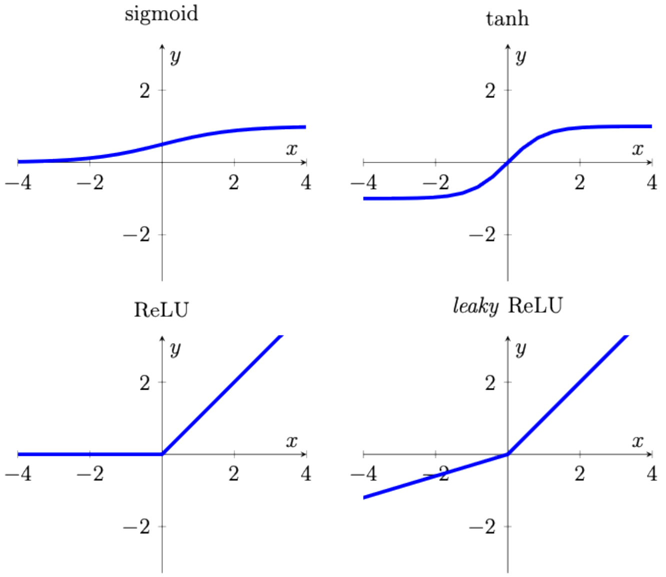


Figure 4.4: The different activation function used on neuron outputs. Image is from [25].

4.4.3 Adversarial training

Adversarial training in GANs is defined as a zero-sum game between two models; the better a model performs at its task, the worst the other model does. In the context of this project, the adversity is characterised by the ability of the discriminator to identify a waveform generated by the generator from a waveform sampled from the data set.

As the generator trains, it learns ways to “fool” the discriminator into thinking its generated waveforms are data waveforms. With a well designed GAN, this usually ends in the generator learning to replicate the features of real data, though not always. Section 4.4.4 describes some of the frequent problems encountered with GAN training.

Similarly, as the discriminator trains, it learns about key features that differentiate generated waveforms from data waveforms allowing it to better discriminate them.

To perform this training, the two models are separately trained one at a time, starting with the discriminator. Figure 4.5 shows the flow chart of the GAN training procedure that was developed. Since the networks are competing, the output of one model is required to train the other model.

Both networks use the discriminator output with opposite loss function. The loss of the discriminator is minimized when the prediction matches the target. The target label are then reversed during the training of the generator such that loss

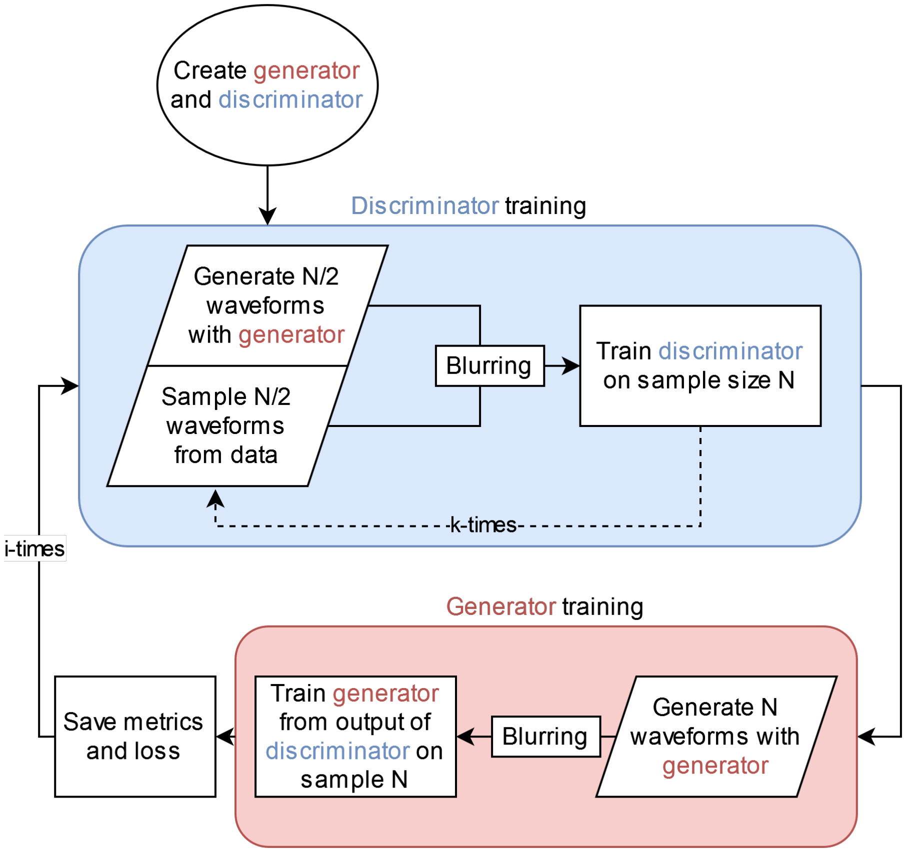

Create generator and discriminator
Discriminator training
Generate N/2 waveforms with generator
Sample N/2 waveforms from data
Blurring
Train discriminator on sample size N
k-times
i-times
Generator training
Save metrics and loss
Train generator from output of discriminator on sample N
Blurring
Generate N waveforms with generator

Figure 4.5: Flowchart of the GAN training procedure. The blurring procedure is described in section 4.4.3. The metrics calculated are described in section 4.5.

is minimized when the prediction of the discriminator are opposite to the target (generated waveforms are predicated as data waveforms).

In each training procedure, a sample of total size N is used as input, and goes through a blurring procedure where targets are inverted and spread out. This procedure is further detailed in section 4.4.3.

Figure 4.6 illustrates the training of the discriminator. A sample of generated waveforms are produced with the current generator, and are mixed with a sample of real waveforms. A blurring procedure is applied to the input. The discriminator is then trained using backpropagation of its loss function using the mixed sample as input. The target label is 0 if the waveform is generated and the target label of 1 if the waveform is from the data sample.

Figure 4.7 illustrates the training of the generator. A sample of generated wave-

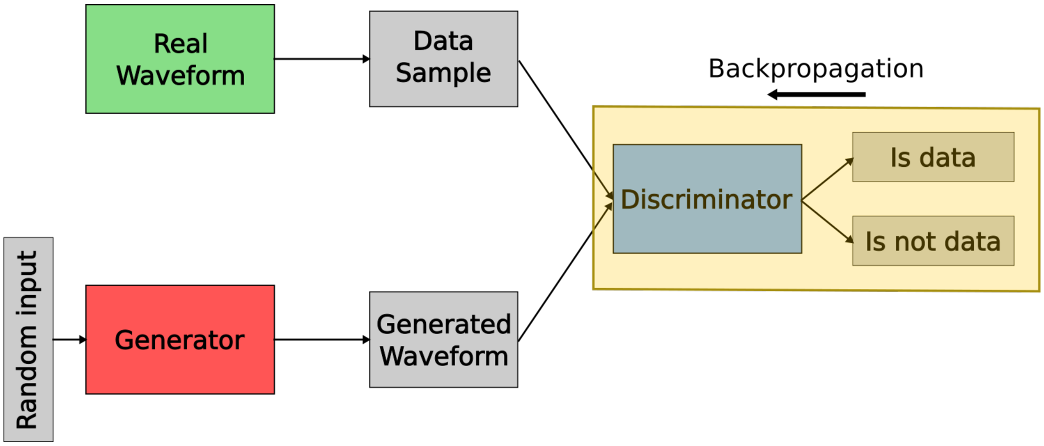


Figure 4.6: The first step in the training of a GAN. The discriminator network is trained on an input of both generated and data waveforms. Its output is a single neuron activated by a sigmoid function to normalize outputs between 0 and 1.

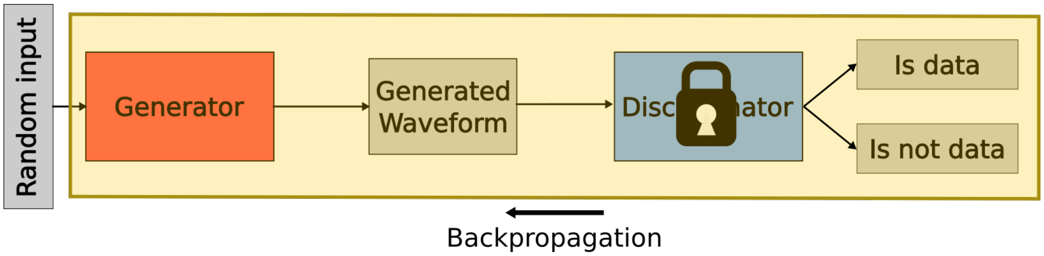


Figure 4.7: The second step in the training of the GAN. The generator trains by generating a waveform, inputting it into the discriminator, and backpropagating all the way back to itself. The discriminator is locked in this step such that the backpropagation only propagates through, but does not train the discriminator.

forms is produced with the current generator. The generated waveforms are then given as input to the current discriminator which will produce an output between 0 and 1 characterizing its certainty that the waveform originates from the data set. The target of the generated waveforms are reversed to be 1 so that the discriminator loss is maximal when it correctly identifies them as being generated. The backpropagation of the discriminator loss is then propagated all the way to the generator without updating any weights on the discriminator (it is locked). Recall that this is possible as the input of the discriminator is the output of the generator such that there is a direct connection between each network's neuron.

Once both networks have been trained, metrics characterizing the waveforms quality are saved and the training cycle is repeated.

Blurring

As seen in figure 4.5, this procedure is applied to waveform samples before being used for training. There are two operations in this procedure:

1. Introducing noisy labels: A randomly selected fraction of the target labels are flipped. If the target is 0, it becomes 1. If the target is 1, it becomes 0.

2. Label smoothing: Target labels are randomly spread out over a range. The target label is 0, it will change to be between x^0_{low}, x^0_{high}. If the target label is 1, it will change to be between x^1_{low}, x^1_{high}. These values are chosen prior to training.

These operations are introduced to make the network more robust to overtraining. They also help the GAN to avoid mode collapse or convergence that doesn't produce quality waveforms. Further details on the challenges that GANs face are described in section 4.4.4.

Hyper parameters

The GAN framework is parametrized by the following hyperparameters:

- N_{epoch}: Each training cycle is called an epoch, and N$_{epoch}$ determines how many epochs are performed during the training procedure.

- N_{batch}: The number of waveforms (either generated or from the data sample) that are used to train each networks during each epoch. Currently, the discriminator is trained with $\frac{N_{batch}}{2}$ generated and $\frac{N_{batch}}{2}$ real waveforms while the generator is trained with N_{batch} generated waveforms.

- k: The number of times the discriminator is trained for each generator training. If k is 3, the discriminator is trained three times and the generator is trained once during each epoch.

- $\%_{noise}$: The percentage of waveforms that have their target label flipped.

- x^0_{low}, x^0_{high}: The range over which to smooth the target label 0.

- x^1_{low}, x^1_{high}: The range over which to smooth the target label 1.

- l_{latent}: The size of the random number vector that serves as input to the generator.
- lr: learning rate of the neural networks.

4.4.4 GAN challenges

Although GANs have a potential to generate quality data, their training procedure contains many challenges that need to addressed. In this section, the main challenges and their solutions are described.

Lack of Convergence

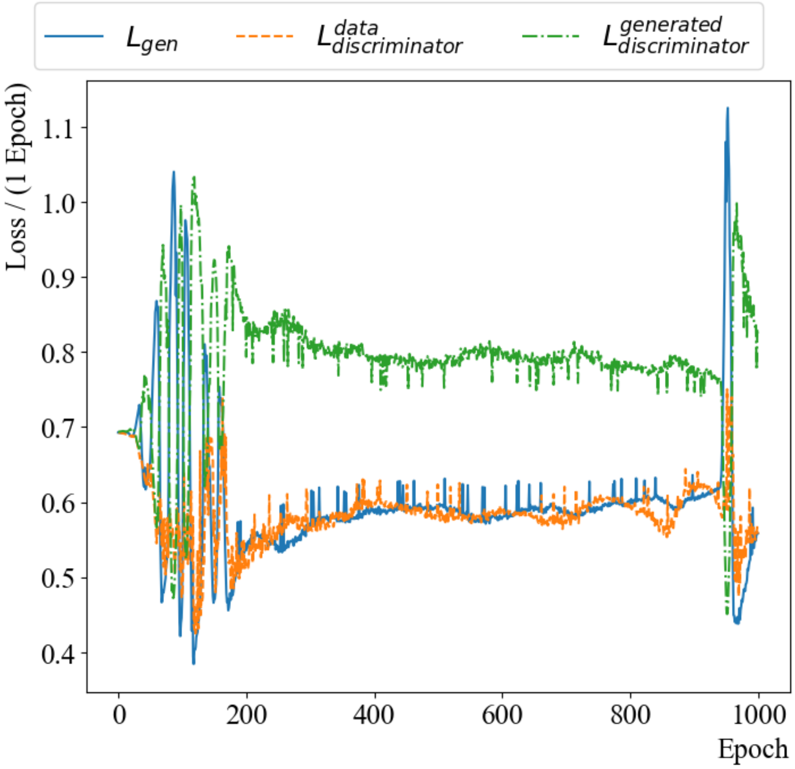


Figure 4.8: Loss functions of the GAN networks after each epoch training. The discriminator loss is plotted for data inputs and generated inputs.

Because the loss function of a GAN network is related to the performance of the other network, the loss function of any given network has no role in determining when the models are sufficiently trained. In fact, a stable model experiencing continuous improvement will have a loss that is on average constant over time due to the networks being at similar “skill” levels. Figure 4.8 shows the evolution of the loss function during a 1000 epoch long training. A stagnation of the models can be seen during the region of almost constant loss from epoch 200 to epoch 900. When the discriminator learns of a new way to discriminate the waveforms, its loss dips while the loss of the generator peaks as seen around epoch 950. The generator then proceeds to learn a new method to fool the discriminator, bringing its loss back to a level similar to the period of constant loss. It is important to note that the generator may have significantly improved its generation quality while remaining at an overall similar loss due to the discriminator having improved by a similar amount.

Another problem with the lack of loss convergence is that the networks are not guaranteed to converge at all. The stochastic process of the training implies they tend to learn new techniques to improve results and then forget them after some time until they re-learn them. This means that the quality of generated waveforms must be monitored closely to know when the model produces something of high quality.

Generator Collapse

Because the loss of the generator is defined through the output of the discriminator, there are possible results that both networks might find satisfying even though their quality may not be good to human standards.

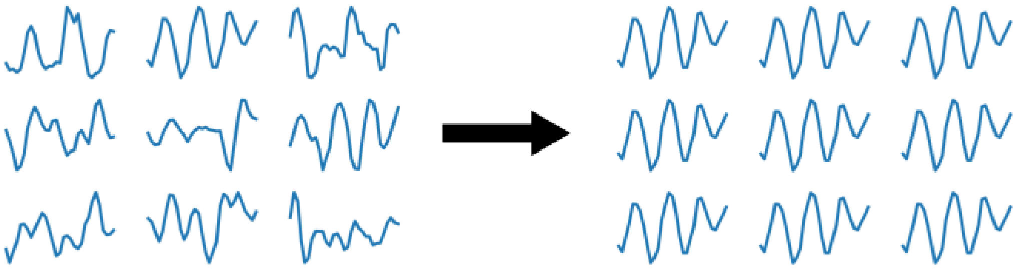


Figure 4.9: An illustration of the mode collapse of generated waveforms. The generator goes from generating varied and believable looking waveforms to always generating the same waveform.

The first such example is called mode collapse. It happens when both networks fail to grasp the generality of their objectives. The generator finds a particular waveform shape that is indistinguishable from data to the discriminator, and optimises itself to always produce that same waveform. The discriminator consistently mislabels the waveform because training against it would overall reduce the number of correct predictions of data waveforms — in other words the training is not strong enough to overcome the mislabelling of the specific waveform. Figure 4.9 illustrates the mode collapse of a generated sample.

More generally, the generator might find a feature that looks odd to the human eye, but that the discriminator isn't able to efficiently learn how to discriminate against. This might involve having a peak at the same location in every waveform.

This problem can be circumvented by making the models more robust. For example, random noise can be introduced in the training samples with the blurring procedure as shown in section 4.4.3 to discourage stagnation and collapse of the output space.

4.5 Evaluating GANs

Because of the adversarial training method, the loss function of any given epoch cannot be used to determine the quality of the generator as it does not converge. Furthermore, the stochastic nature of GANs can cause the models to deteriorate in quality at later stages in their training. As such, GANs require independent evaluations at every epoch.

This section describes the development of metrics used to identify the quality of GANs, and how they are used to select a model with good quality. These metrics are currently not included in the training process as the GAN architecture looks only at raw waveforms. This is discussed further in section 4.7.2.

The results of a selected model is demonstrated for a GAN using a convolutional neural network (CNN) architecture; these GANs are usually referred to as deep convolutional GANs (DCGAN).

4.5.1 Metrics Used to Evaluate Performance

To evaluate the quality of a GAN, the following metrics are calculated for both the data waveforms and the generated waveforms:

χ^2: The χ^2 metric is calculated using equation 3.1 and is the same metric that is affected by the calibration of the PSD tool described in section 3.4. This χ^2 characterizes the difference of the waveform compared to a flat pedestal. The variance and covariance of the ADC points are described through the covariance matrix in equation 3.2.

ΔA_{max}: The maximum amplitude difference is defined as the difference between the maximum value and minimum value of a waveform.

$$\Delta A_{max} = \max(\mathbf{S}) - \min(\mathbf{S}). \tag{4.13}$$

where $\mathbf{S} = S_i = s_1..s_{31}$ is a waveform.

ΔA_{avg}: The average amplitude difference is defined as

$$\Delta A_{avg} = \frac{1}{31}\sum_{i=1}^{31} |S_i - \bar{S}| \tag{4.14}$$

where

$$\bar{S} = \frac{1}{31}\sum_{i=1}^{31} S_i. \tag{4.15}$$

f: The frequency space of waveforms calculated using a discrete Fourier transform (DFT) which are binned into 16 bins. The first bin is dropped as it represents the $f = 0$ components of the waveform (overall shift) and is another metric itself. This metric is visualized by displaying the sum of frequencies' powers for each bin across the entire dataset.

After each epoch training, a new sample of 154692 generated waveforms is created. The metrics are calculated for each new waveform and overlaid with those of the data waveforms (also of sample size 154692). Figure 4.10 shows the four metrics comparing the data and generated waveform samples.

To find the best generator models, a quantitative comparison of the metrics distributions is obtained through a two-sample Kolmogorov-Smirnov test (KS2) [26]. The test is performed and saved after each epoch between the 154692 generated and data metric entries to allow retroactive comparison. Figure 4.11 shows a plot of all KS2 results for the metrics across a training of 1000 epochs.

As a note, the KS2 score is usually converted to a p-value which is used to accept/reject the null hypothesis; the null hypothesis here is that the two distributions are drawn from the same underlying statistical distribution. Due to the large sample size used for the KS2 tests and the considerable differences between the distributions, the p-value is calculated to be ~ 0 for all KS2 scores. However, in the context of

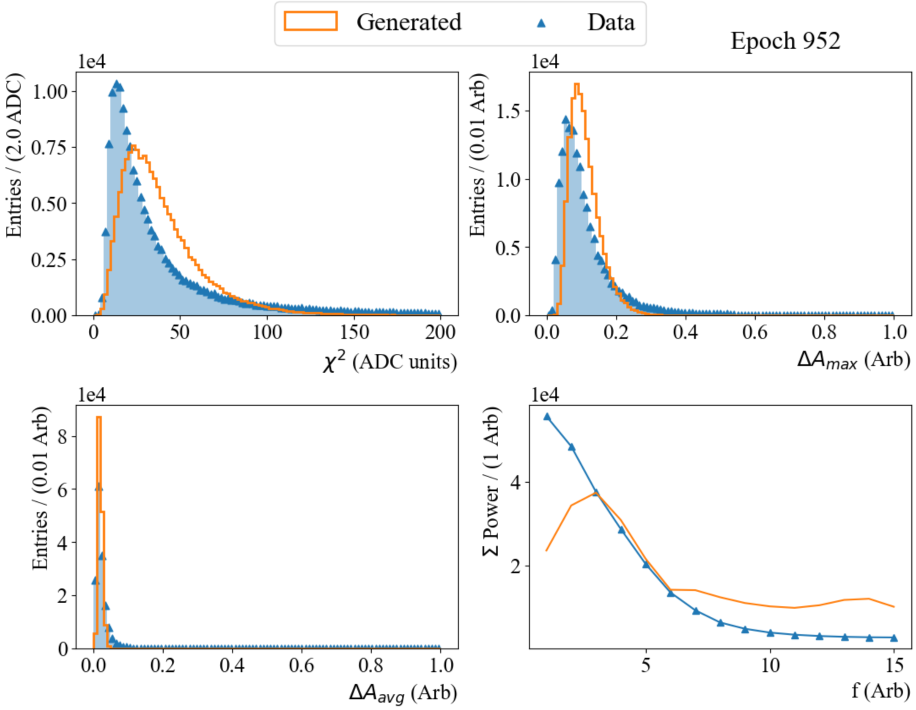


Figure 4.10: The metrics comparison between the data and generated waveforms. Both samples contain 154692 waveforms. The epoch at which the evaluated generator was last trained is displayed.

GAN evaluation, the objective of this test is to determine at which epoch the distributions are most similar. Recall that the stochastic training of GANs implies that conventional convergence of loss function cannot be used to find the optimal model; a manual evaluation is required. Because the sample sizes never change — and since all metric's samples share the same size — it is enough to compare the KS2 scores relative to one another to obtain a measurement of quality between epochs or models.

KS2 test

The KS2 test compares two one-dimensional samples by returning the maximum difference in their empirical distribution functions. The KS2 score is given by

$$D = \sup_x |F_1(x) - F_2(x)| \tag{4.16}$$

where $F_1(x)$, $F_2(x)$ are the empirical distribution functions of the samples at a

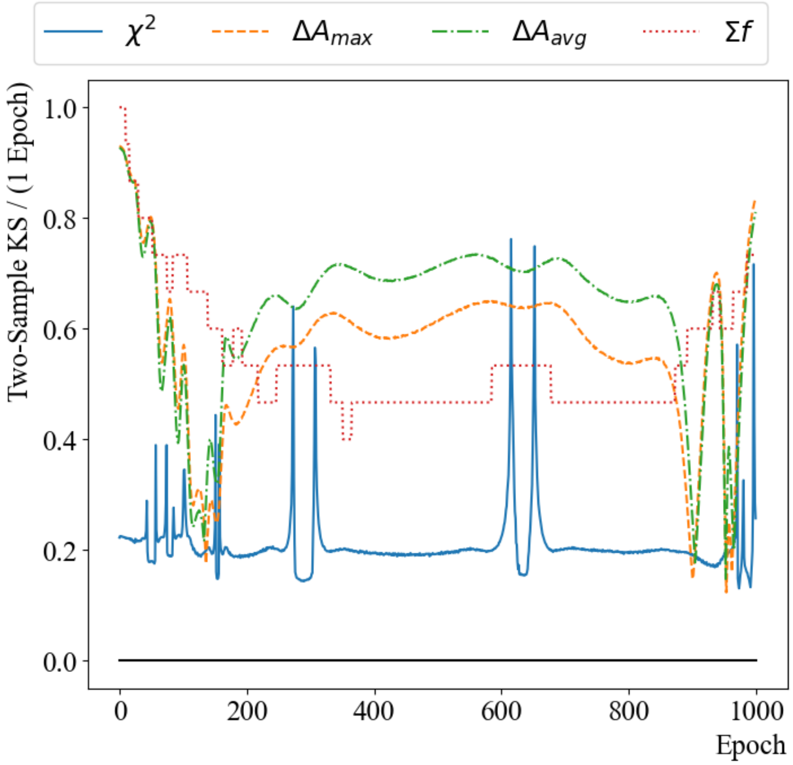


Figure 4.11: The result of the KS2 test on the metrics at each epoch.

value x. The KS2 test is 0 when the two distributions are identical.

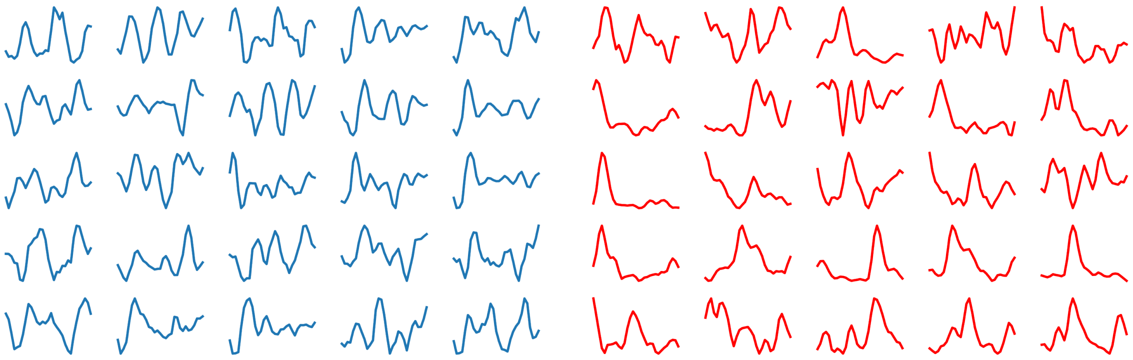
The KS2 test is performed normally on the χ^2, ΔA_{max} and ΔA_{avg} distributions, each of which contains 154692 entries in the data and simulated samples respectively (one for each waveform). Since f contains 15 one-dimensional arrays, a KS2 test is performed for each frequency for a total of 15 KS2 tests. An average is performed to obtain a single KS2 values. This variable is identified using f_{avg}.

Because the importance of the metrics on actual physics observables are yet unknown, an average of the four KS2 values obtained is taken to find the best epoch. Section 4.6 describes further comparisons needed to validate the quality of generated waveforms. A visual inspection can still be done to find regions where KS2 are overall low which determines at which epoch the models are best.

Visual Comparison of Waveforms

An additional method of comparison is the visual inspection of generated waveforms. This is not intended as a permanent metric, but is useful to determine the viability of networks — e.g. convergence failures or mode collapse can easily be visually identified.

Figure 4.12 shows a sample of generated waveforms compared with data waveforms



(a) A sample of generated waveforms.

(b) A sample of data waveforms.

Figure 4.12: Comparison of a generated waveform sample (left) versus a randomly selected data sample (right).

4.5.2 Network obtained

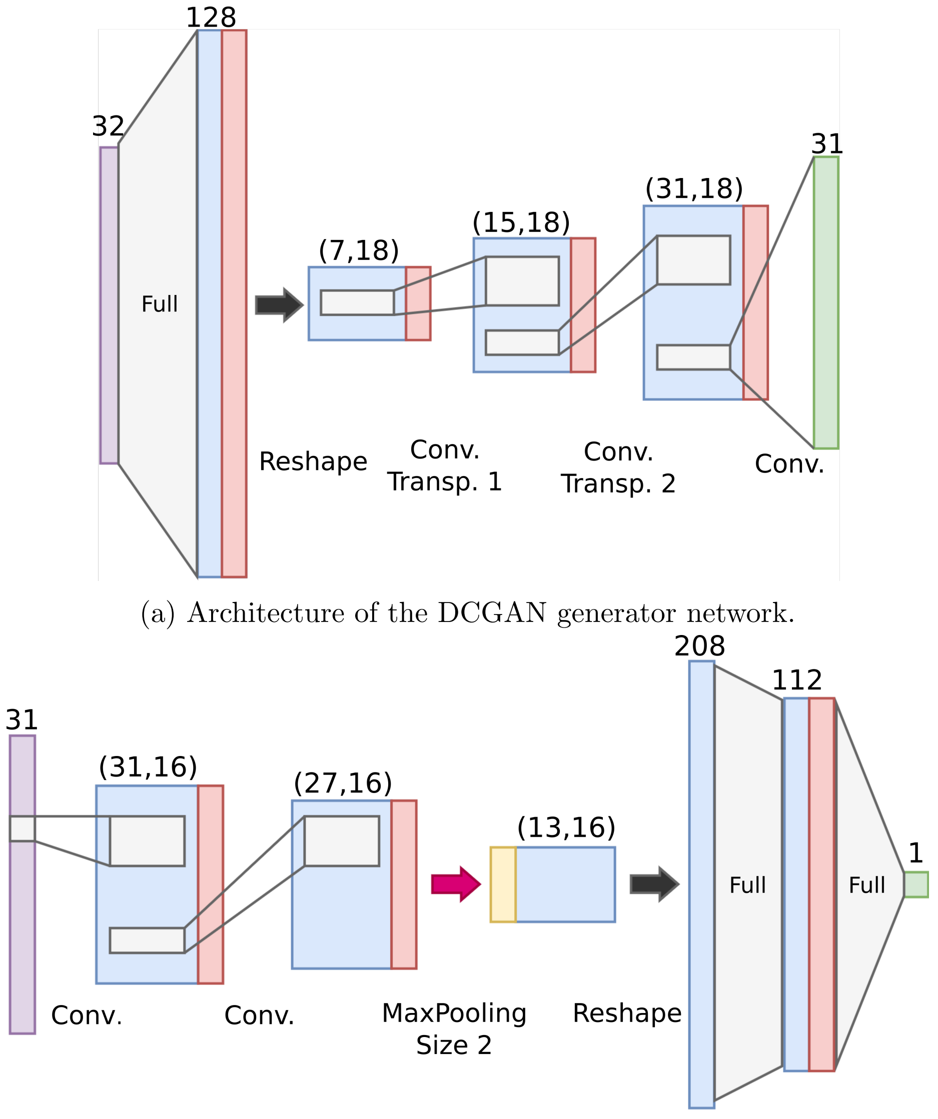
From the work done exploring the space of GANs, it was found that regular neural network architectures were unstable and would often fail to converge, or experience mode collapse. A much more robust architecture was found to be the deep convolutional GAN (DCGAN).

Figure 4.13 shows the architectures of the generator and the discriminator. A latent vector of size 32 was chosen as input to the generator. The schematic is color coded: the purple layers are input vectors, the blue layers are neuron layers, the red layers are Leaky ReLU activations, the yellow layers are dropout operations where outputs are randomly reset to 0, and the green layers are network output layers. The models uses a mix of different layer operations including convolutions, transpose convolutions and fully connected dense layers which are represented as connections between layers. The black arrows indicate a reshaping of the neurons. The red arrows indicate a MaxPooling operation where neurons are pooled in pairs whose maximum value is forwarded to an output neuron.

A sigmoid activation function is applied on the discriminator output to bound the prediction between 0 and 1, while a tanh activation function is applied on the generator output to bound the waveform between 0 and 1.

The network uses the following hyper-parameters:

- $N_{epoch} = 1000$



(a) Architecture of the DCGAN generator network.

(b) Architecture of the DCGAN discriminator network.

Figure 4.13: Schematics of the architectures of the DCGAN models. (a) shows the generator used to simulate waveforms from a random number input, and (b) shows the discriminator used to discriminate between generated and real waveforms.

- $N_{batch} = 16$
- $k = 3$
- $\%_{noise} = 5\%$
- $x^0_{low} = -0.3$, $x^0_{high} = 0.2$
- $x^1_{low} = 1$, $x^1_{high} = 1.3$
- $l_{latent} = 32$
- $\text{lr} = 0.0002$

Using the KS2 distribution for a 1000 epoch training, the epoch with lowest average KS2 is epoch 874, followed by epoch 842. Figure 4.14 shows the KS2 distribution with the best two epochs marked by black lines vertical lines. The DCGANs converge to have their best epochs in the region $700 < \text{epoch} < 1000$.

Figure 4.15 shows the metrics comparison for epoch 874. Good agreement is observed between the variables overall. There is a tradeoff between χ^2, f and the ΔAs. In the best epoch, ΔA_{max} and ΔA_{avg} are optimal while χ^2 and f observe some discrepancies.

Shortcomings

Some other comparisons can be performed to show some of the shortcomings of the networks. In figure 4.16, a comparison of the waveform averages between the data and generated waveforms shows disagreement between the distributions. This effect is due to the important physically related features being present in the tails of the distribution such as physically different contributions to the background. The model struggles to capture these distributions, likely because the discriminator does not prioritize overall shift of waveforms as a variable of consequence. Possible solutions are addressed in section 4.7.2.

Another possible visualization of f consists of picking out the frequency that contributes the most power to the waveform, called f_{max}. This frequency can be binned for all waveforms in the sample, allowing the reduction of the f metric dimension from 15 to 1. Figure 4.17 shows the comparison of f_{max} between data and generated waveforms. Because the KS2 of f is calculated for each bins, the discrepancy shown is also captured within the choosing process of best models. One note is that this

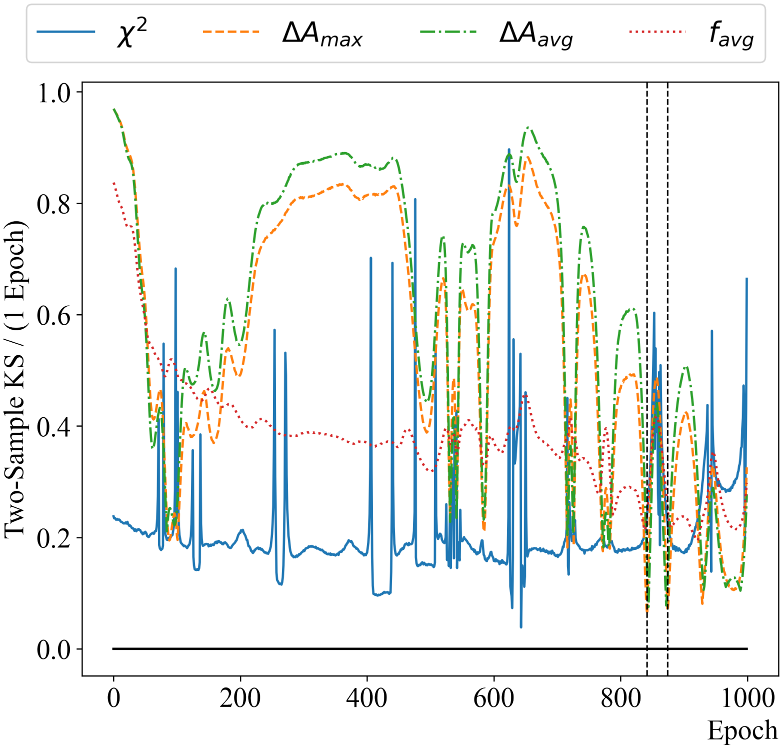


Figure 4.14: KS2 over a 1000 epoch training of a DCGAN. The best epochs are marked with a black line at 842 and 874.

metric doesn't show discrepancy between low power frequencies as they never appear a maximal contribution.

4.5.3 Generation time and Training time

To estimate the performance of GANs, another dimension of metrics used are the time needed for training and generation of waveforms. The following tests are run to see the time performance of the model:

$T_{training}$: The amount of time required to train the model over 100 epochs. This metric is given in seconds, and is performed on a local machine with 3 cores at 1.8 GHz.

$T_{generation}$: The amount of time required to generate 8736 waveforms — one for each crystal. This metric is given as a percentage of the time needed to simulate and reconstructe a $B\bar{B}$ event.

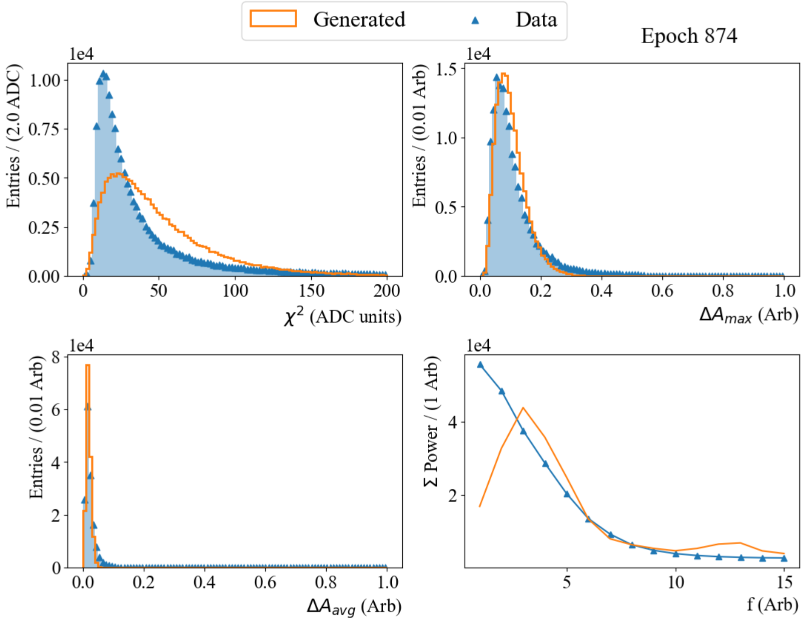


Figure 4.15: The metrics comparison of the epoch 874.

Values for $T_{training}$ obtained for trainings with different amount of metrics saving are given in table 4.1.

Test	Time (s)	Ratio with Test 3	Description
1	560	0.356	Generating waves, calculating ΔA_{max}
2	1118	0.710	Generating waves, all calculations
3	1574	1	Generating waves, all calculations, plotting

Table 4.1: Values of $T_{training}$ on a local machine with 3 cores at 1.8G GHz. Test 1 involves the minimum amount of metric calculations and saving; only one metric is calculated and saved. Test 2 involves the calculation and saving of all metrics. Test 3 involves the calculation and saving of all metrics and their plots. Time ratios between the different tests and test number 3 are given to compare time improvements between the methods.

In a real implementation, plots would not have to be saved. It is further likely that a trigger consisting of the calculation of a single metric could be used to determine whether all metrics should be calculated and saved.

When generating with a trained network, a value of $T_{generation} = 7.7\%$ was found.

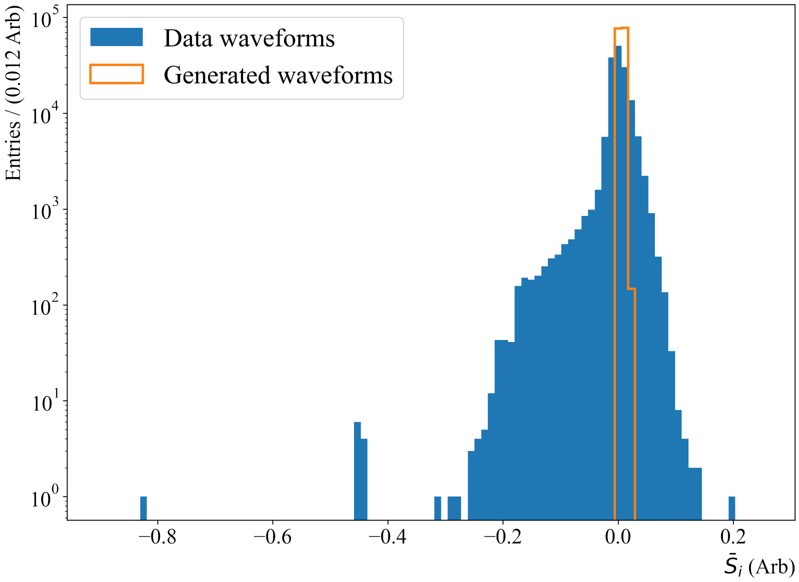


Figure 4.16: Comparison of waveform averages between data and generated waveforms.

Possible Improvements

Although the time metrics are currently long, there are many areas where they could be shortened. The current implementation is done in Python, and as such much of the overhead is expected to be slower than if implemented in a compiled language such as C++. The generation of random numbers, the cycle of the framework and the various calculation would benefit from an optimized implementation. This is equally true for the post-training generation of waveforms.

A further point is that it may not be necessary to train a complete model for each crystal, for each run. If the background conditions are similar enough between runs, the same networks can be reused as GANs learn to replicate the statistics of data, not the sample itself. In addition, the current framework tends to provide mature GANs at epochs higher than 700 which implies that a model partially trained up to epoch 500 could be used as the initial weights in the GAN training of another crystal or run. This would allow the networks to have a reduced training time, only enough to capture finer details of the distribution.

The last point is that some aspects of this training could benefit from parallelization. This is especially relevant when performing batch training of multiple samples whose loss calculation is independent, or generating multiple waveforms to be evaluated independently. Together with a dedicated batch server, much faster clock-time

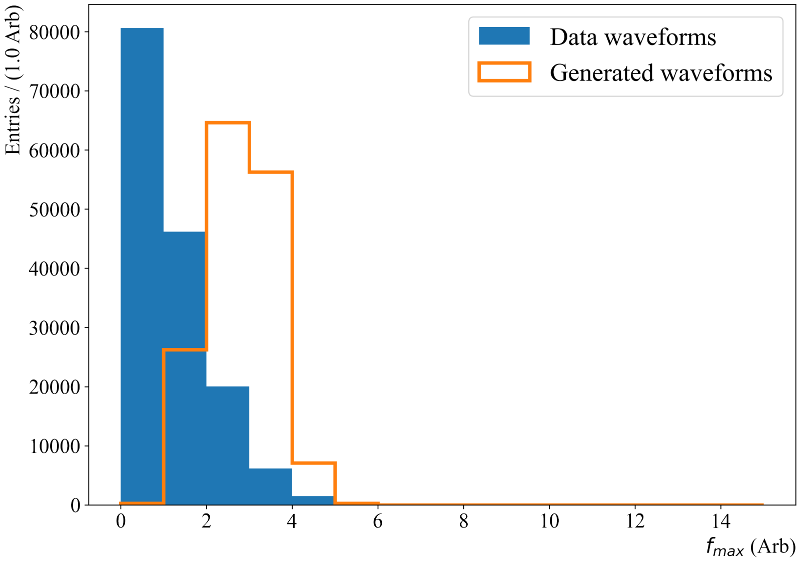


Figure 4.17: Comparison of f_{max} between data and generated waveforms.

could be achieved for the training of GANs.

4.6 Next Steps

The scope of the GAN project defined within this thesis was the building of a framework capable of training GANs and the demonstration of GANs as potential generators of noise waveforms. The resulting networks showed generated distributions with acceptable metric agreement and high generation time.

This section enumerates the next steps that would be required in the implementation of a GAN background generation network. It also identifies some areas of potential improvement, and areas that would require further consideration in the scope of such a generator.

4.6.1 Selection of the Best Model

As of the writing of this thesis, the training process involves the calculation, plotting and saving of all metrics to file. The metrics are then reviewed using helpful tools such as the KS2 test to find epochs with good metric agreement with the data sample. The best epochs are then selected manually via visual comparison.

For this method to be implemented at scale, the selection of the best model would need to be automatized entirely. To achieve this, a pipeline needs to be built that would take the the KS2 scores of the metrics and perform a weighted sum to obtain a single performance metric. This metric would then need to be minimized and a threshold would have to be found at which models should be saved for usage.

This process would require further analysis of which metric matters most when measuring performance, and weight them appropriately while selecting the best model trained. One way this importance can be found is by using physics observables as validation metrics. Because these waveforms are to be used in physics simulations, they can only be as good as their ability to replicate the physics effects of the data waveforms.

One metric of interest is E_{extra}, the extra energy measured in the ECL that is not expected from the reconstructed event. This variables is sensitive to the background conditions in the ECL, and is a good candidate to judge the quality of simulated background. Simulations of simple particle interactions in the ECL are performed using data waveform overlays and are compared to one with generated waveform overlays. The comparison of E_{extra} across the run will reveal the statistical behaviour of generated waveforms.

One step further is to go backwards and to use E_{extra} as a way of calibrating the selection of the best model in a training. Since there are trade-offs between the metrics involved in evaluating GANs, models with different statistical agreements (e.g. better χ^2 vs better ΔA_{max}) can be used to compare the relative importance of any metric in the E_{extra} final distribution.

A second metric of interest is the distribution of cluster energies for particles bellow a few hundred MeVs. If the generated noise is sufficiently accurate, the cluster energies reconstructed using generated noise should follow the same distribution as the cluster energies reconstructed using background overlays.

Lastly, another metric of interest is the reconstructed π^0 mass. Since the main decay mode of π^0s is $\pi^0 \rightarrow \gamma\gamma$, the mass calculation requires the measurement of photon energies. If the generated noise waveforms are inaccurate, a measurable change in the ECL resolution would be expected. This change in resolution would affect the π^0 mass measurements and quantify the accuracy of noise generation.

4.6.2 Pipeline

Once a training and evaluation method is selected, a pipeline will need to be built to automatically include the saving of background waveforms, the training of models until desired quality is achieved and the saving of the model weights to be used when later called within a simulation script.

4.7 Comments

4.7.1 Quality Cost Ratio

To make sure that GANs are the best approach to generating noise waveforms, a baseline of other methods is necessary for comparison. Other methods such as network generation from an auto-encoder network were initially dismissed as not very good, but the findings of better architectures in this GAN project have renewed interest in comparing auto-encoder performances with GAN performances for the same architecture.

Although quality of waveform generation is important, the cost of training and operating a model is equally important. The comparison between the quality cost ratio of different methods can be characterized with equation 4.17:

$$Q_{training} = \frac{\text{Training Time}}{\text{Quality of Model}}, \qquad Q_{generation} = \frac{\text{Generation Time}}{\text{Quality of Model}}. \tag{4.17}$$

Auto-encoders may ultimately provide lower quality, but also lower costs. A decision will have to be taken on the desired values for $Q_{training}$ and $Q_{generation}$ before moving forward.

4.7.2 Scope of the Generation

Finally, shortcomings of the DCGAN model used were shown in section 4.5.2. One of concern is the discrepancy between the waveform averages. Because much of the physics that must be captured is in the tails of this distribution, capturing it with the current model which optimizes overall agreement over tail specific agreement will be a challenge. A potential solution to this is simply the reconsideration of the scope of generation. It is possible that the entirety of the details cannot be captured by any one model, and as such the models should only attempt to generate noise around the peak of averages to focus on their physics. Alternatively, different models could target different sections of this distribution in an attempt to capture it entirely.

More studies will need to be performed to answer these questions, but mostly a discussion about the scope of these generating models is in order.

Lastly, another major concern is the discrepancy in the frequency power spectrum. It may be possible to force the generator into incorporating more frequency components by including a frequency component directly in the training process. It is also possible that the entirety of the frequency details cannot be captured by any one model, and that further separation of the data into frequency regions will be required.

Chapter 5

PSD Based Machine Learning Particle Classification

Although Belle II has seen considerable improvements to its detector components compared to the Belle design, its physics output, which is focused on greater precision measurements compared to the previous generation B-Factories, will still be affected by systematic uncertainties such as particle misidentification. A notorious example of particle misidentification is that of $\mu^{\pm}$ and $\pi^{\pm}$. Their masses, 105.66 MeV and 139.57 MeV respectively, are close enough to one another to render Cherenkov based particle identification difficult. To discriminate between the two particles, a mix of variables such as E/p analysis, shower shapes in the ECL and the presence of tracks in the KLM can be used.

These techniques can however fail when the particles have similar energy depositions and shower shapes. This section describes the work done to create a machine learning based particle discriminator using only ECL information with the addition of the new PSD tool (section 3.4). Classifiers are trained for $\mu^{\pm}$, $\pi^{\pm}$ and $e^{\pm}$, $\pi^{\pm}$ discrimination.

5.1 Contributions

Section 5.3 describes the simulation and processing of the data used in the training of the discrimination models.
Section 5.5 shows the obtained results for the different discriminator model trained.

5.2 Motivation

When interacting in the ECL, $\pi^{\pm}$ have a $\sim 50\%$ probability of producing a hadronic shower [27]. The hadronic component of these showers is captured by the PSD tool, and can be used to identify the nature of the showers. This information is measured on a crystal-by-crystal basis, and has yet to be summarized and transformed into cluster level variables for charged particles.

A machine learning model is introduced to capture the crystal level information within a shower, and output a probability of this shower having been produced by a hadron. Because many other detectors are involved in the discrimination of $\mu^{\pm}$ and $\pi^{\pm}$, the scope of this discriminator is mainly to extract available ECL information and provide predictions of the shower nature to be used by other tools whose scope is to group multiple sub-detectors. Belle II currently has a PSD discriminator focused on neutral hadron identification present in its software framework, and this tool aims to complement it by offering a similar abstraction for charged showers.

The physics motivation of this tool is to reduce the mis-identification of leptons and hadrons. An example of relevance is for τ decays where the measurement of $\pi^{\pm}$ can be contaminated by $\mu^{\pm}$ contributions. To use the full potential of Belle II integrated luminosity, it is desirable to achieve discrimination of these particles in the regions where their E/p and shower shape (due to few interactions) are identical.

5.3 Classifier Inputs

For each ECLCluster reconstructed in the ECL, a mix of cluster variables and crystal variables are used. For each cluster, the following variables are used for the 20 highest energy crystals in the shower:

1. $F_{crystal}$: The fit type used on the crystal waveform to obtain the PSD information. This variable take the value 0 if one gamma and one hadron templates were used, 1 if two gamma and one hadron templates were used and 2 if diode crossing templates were used [22].

2. $E^{total}_{crystal}$: The total energy of the crystal.

3. $E^{hadron}_{crystal}$: The hadron energy contribution to the crystal.

4. $R_{crystal}$: The distance of a crystal to the center of the cluster.

5. $W_{crystal}$: The weight of the crystal. This variable is obtained from the clustering algorithm and describes how strongly a crystal relates to its associated cluster.

If the cluster has less than 20 crystals in total, empty crystals are added to the training data with the following values:

1. $F_{crystal} = -1$
2. $E^{total}_{crystal} = 0$
3. $E^{hadron}_{crystal} = 0$
4. $R_{crystal} = -1$
5. $W_{crystal} = 0$

The following cluster variables are added for every ECLCluster:

1. $E_{cluster}$: The total energy of the ECLCluster.
2. $E1E9$: The ratio of energies of the central crystal, E1, and 3x3 crystals, E9, around the central crystal.

This technique is based on the similarly trained PSD discriminator currently present at Belle II. This new model however trains on clusters created by charged particles as opposed to the current which is trained on neutral clusters [22].

5.3.1 Training Sample: Simulation and Processing

Momentum (GeV)	θ (cos isotropic)	ϕ (isotropic)	N
[0-7]	[17-150]	[0-360]	1 000 000

Table 5.1: Kinematic values of the simulated particles where θ is the polar angle as measured from the z-axis and ϕ is the azimutal angle measured from the positive x-axis.

Samples of $\mu^{\pm}$, $\pi^{\pm}$ and $e^{\pm}$ are simulated using the parameters shown in table 5.1. Specific clusters are then selected for the training data set using the following cuts:

1. The ECLCluster is located in the barrel.

2. The ECLCluster is clearly matched to one of the simulated particles; i.e. its energy contribution originates from a unique simulated particle.

3. The ECLCluster has at least one crystal containing a waveform that can be used for PSD. In the implementation of this tool, particles without PSD are labeled with -1.

For each ECLCluster, its viable crystals are kept and added to the data sample if they respect the following requirements:

4. The PSD fit of the crystal has $0 < \chi^2 < 60$.

5. The PSD fit must be $F_{crystal} = 0$ or $F_{crystal} = 1$. Crystals with $F_{crystal} = 2$ (diode template) are not currently used for this MVA.

Table 5.2 details the number of clusters removed with each cut. Each sample starts with 1000000 simulated particles as seen in table 5.1.

Cut	$\pi^{\pm}$	$\mu^{\pm}$	$e^{\pm}$
Number of clusters reconstructed	1063743	1010974	1009004
Dropped due to requirements 1 or 2	309934	239257	211601
Dropped due to requirement 3	108642	36878	229826
Dropped due to all crystals failing 4 or 5	9458	19610	390
Number of clusters added to the training set	635709	715229	567187

Table 5.2: Description of the cuts applied on the training data set. The number of clusters is greater than the number of simulated particles due to some particles leaving more than one cluster in the ECL, or clusters being associated with secondary particles.

5.4 Description of the training procedure

The maximize the effectiveness of these tools, the output is chosen to be continuous. This will allow users more control over the way they choose to use this tool. The model chosen is a gradient boosted decision tree (GBDT). These models are resilient to over training, and the decision tree algorithm is virtually immune to drawbacks caused by the high dimensionality of features.

This section briefly describes the working and training of decision trees and GBDTs. It also describes the general pipeline used to train the models used in this work.

5.4.1 Decision Tree

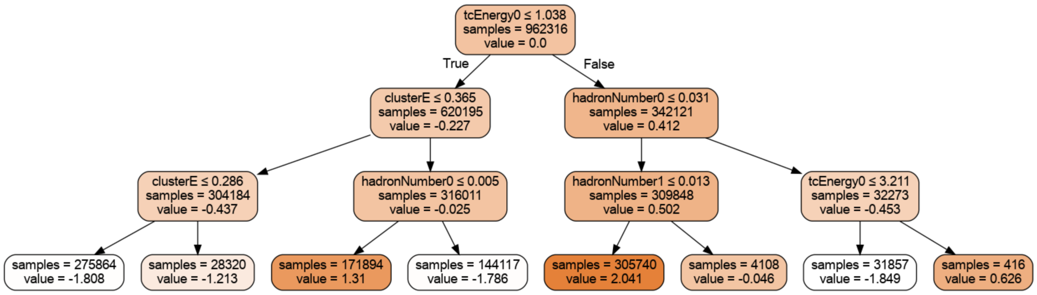


Figure 5.1: An example decision tree.

At the core of GBDTs are decision tree. The algorithm behind decision tree aims to separate the data according to their targets (0 or 1) using successive logical cuts. Figure 5.1 shows a trained decision tree attempting to classify $e^{\pm}$ and $\pi^{\pm}$. Starting at the root, the data set is separated based on the energy of the highest energy crystal in the ECLCluster. The data set is further separated using hadronic contributions of crystals, and the cluster energy.

The training procedure of decision trees is also a gradient descent algorithm, but it is calculated differently than for neural networks. The qualitative approach is simple: each time a branch is created on the decision tree, a single feature k and a threshold t_k (e.g. $k < t_k$) is selected to produce the purest subsets (weighted by their size) on each side of the decision branch. This procedure is repeated on the next tree depth for each branch, until the maximal depth permitted for the tree has been reached. A further discussion of the gradient descent algorithm used to find which feature produces the purest subsets is detailed in [28].

A probability can be associated with each leaf of the tree based on how many of each target class end up in the given leaf. The probability, defined in equation 5.2, is obtained in terms of the "logarithmic odds" defined in equation 5.1:

$$\text{log odds} = \log(\frac{p}{1-p}) \tag{5.1}$$

where p is the fraction of all items with the binary target "1".

$$\text{probability} = \frac{e^{\text{log odds}}}{1 + e^{\text{log odds}}}. \tag{5.2}$$

For instance, a leaf containing a hundred events with binary target of “0”, and two hundred events with binary target of “1” would have a logarithmic odds of

$$\log(\frac{200}{100}) = 0.693 \tag{5.3}$$

and a probability of

$$p = \frac{e^{\log(\frac{200}{100})}}{1 + e^{\log(\frac{200}{100})}} = 0.667. \tag{5.4}$$

5.4.2 GBDT

The implementation of the gradient boosted decision tree is complex and this section will only do an overview of the training process. The details of the procedure can be found in [28].

The main concept of GBDT is the usage of multiple decision trees (~ 100) to obtain progressively better predictions by “boosting” the results of the trees coming before them. The GBDT model creates trees that are evaluated in series, i.e. the training of each new tree aims to eliminate the mistakes made by the previous trees. This is opposed to other ensembles or random forests which create and evaluate trees in parallel, and use their collective votes to make a prediction.

Similarly to a regular decision tree, a GBDT classifier outputs a probability which is calculated from the logarithmic odds of a given item. The first prediction is thus obtained by finding the logarithmic odds of the entire sample and by calculating its associated probability using equations 5.1 and 5.2 respectively. Then, the residual r_i of the GBDT’s prediction for an item i is calculated using equation 5.5:

$$r_i = p_i - \hat{y}_i \tag{5.5}$$

where p_i is the last probability obtained from the trees currently in the GBDT and $\hat{y}_i$ is the target value of the item (0 or 1).

A new tree is then created and trained to have its leaves predict the residual of each given item. The output value of a leaf is calculated using equation 5.6:

$$\frac{\sum r_i}{\sum p_i(1-p_i)} \tag{5.6}$$

which is summed over all items in a leaf.

Then, a new probability can be calculated for each item using equation 5.2 where the addition of the logarithmic odds with the output value of the new tree is used as the new logarithmic odds. The output value of trees are usually weighted by a learning rate, typically of value 0.1, before being summed to make the training more stable. This procedure is repeated until the number of trees specified has been achieved.

This method further contrasts other boosted decision trees such as the “AdaBoost” algorithm. AdaBoost uses the same target (0 or 1) for each event at each iteration, but weights the data to more accurately capture the events that are difficult to discriminate. Gradient boosting, on the other hand, does not weight the data, but changes the target value of items to account for the errors of previous trees.

This procedure strongly weighs “outlier” events in the data as it trains to replicate the discrepancy they cause in the regular data.

5.4.3 Method

Once the training data is obtained, it is separated in two sets: a training set and a validation set. The training set is used for the training, while the validation set is kept aside to test the models.

Two different training methods are used. The first method involves training the model on the entire training set and the second method separates the training set into different bins based on the momentum of the particle creating the ECLCluster, and trains the bins separately.

The main metric used to evaluate the models is the receiver operating characteristic (ROC) curve, as well as the area under the ROC curve called the AUC. The ROC curve has the $\pi^{\pm}$ fake rate on its x-axis and the lepton efficiency ($\mu^{\pm}$ or $e^{\pm}$) on its y-axis. They are defined using equation 5.7 and 5.8 respectively.

$$\pi^{\pm} \text{ Fake rate} = \frac{\text{\# of real pions predicted as leptons}}{\text{total \# of pions}} \tag{5.7}$$

$$l^{\pm} \text{ Efficiency} = \frac{\#\text{ of real leptons predicted as leptons}}{\text{total }\#\text{ of leptons}} \tag{5.8}$$

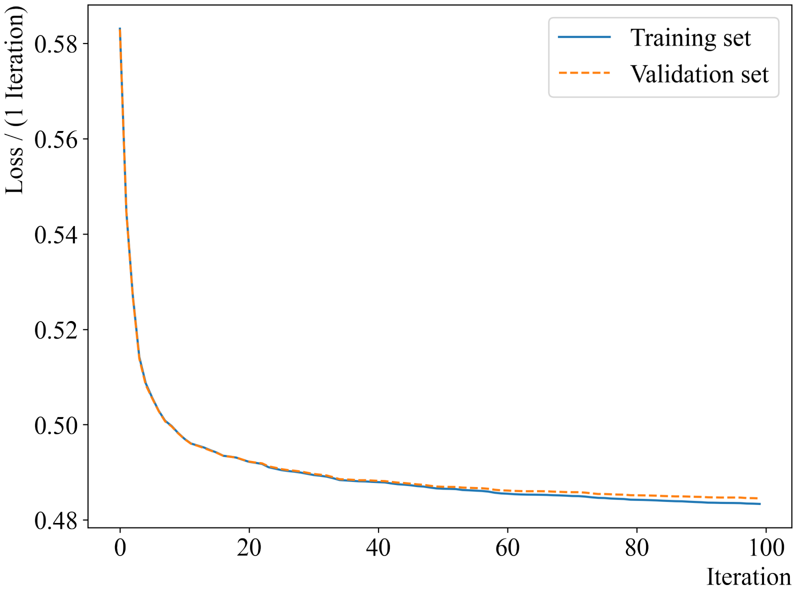


Figure 5.2: Loss function per iteration of a $\mu^{\pm}$, $\pi^{\pm}$ classifier plotted for both the training and validation set.

For every training, the loss of the model is calculated after each iteration (i.e. the addition of an extra tree) to verify for overtraining. Figure 5.2 shows an example of the loss obtained at every iteration during the training of a $\mu^{\pm}$, $\pi^{\pm}$ discriminator using PSD information. Both the loss of the training set and the validation set are shown

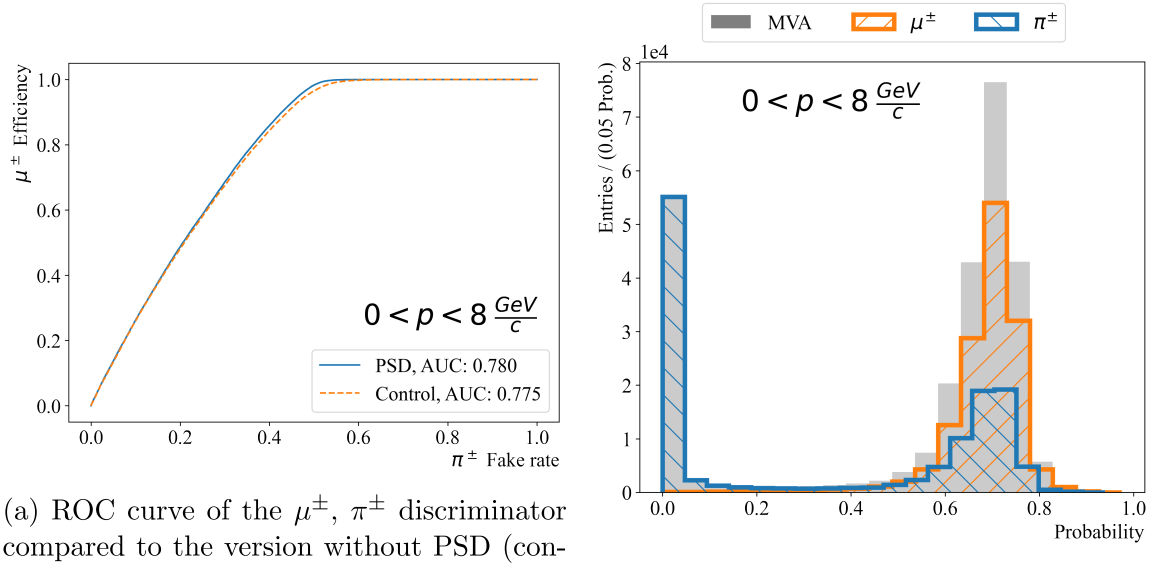
5.5 Validation

To compare the quality of models, their ROC curves are plotted in comparison to a model trained without PSD information. The ROC curve is further compared to the one obtained by the model currently implemented, which is trained on neutral clusters.

5.5.1 Classification of $\mu^{\pm}$, $\pi^{\pm}$

In this section, the model trained to classify $\mu^{\pm}$ and $\pi^{\pm}$ are shown. When the shape variables are added to the training set, the addition of PSD to the training brings little advantage to the regular shape based method.

Non-Binned



(a) ROC curve of the $\mu^\pm$, $\pi^\pm$ discriminator compared to the version without PSD (control). The AUC is shown in the legend for each metric.

(b) Probability output of the $\mu^\pm$, $\pi^\pm$ discriminator.

Figure 5.3: Metrics of the $\mu^\pm$, $\pi^\pm$ discriminator trained over the entire momentum range.

Figure 5.3a shows the ROC curve of a GBDT trained on the entire training set. Only a very minor improvement can be seen compared to the model without PSD, showing that much of the discrimination is performed using shower-shape information.

Figure 5.3b shows the probability distribution of the GBDT trained on the entire training set. This representation visually shows that a pure sample of pions can be obtained, and that a pure sample of muons cannot be obtained. This can also be observed by plotting the ROC curve for inverted targets. Figure 5.4 shows the ROC curve of a GBDT trained on the entire training set where the x-axis shows the $\mu^\pm$ fake rate and the y-axis shows the $\pi^\pm$ efficiency.

Figure 5.5 shows the comparison between the currently implemented neutral model and the new charged model. As expected, the charged model performs significantly better than the neutral model. It is worth noting that the current neutral model has a worse performance while discriminating charged particles than a model trained on cluster-shape variables only.

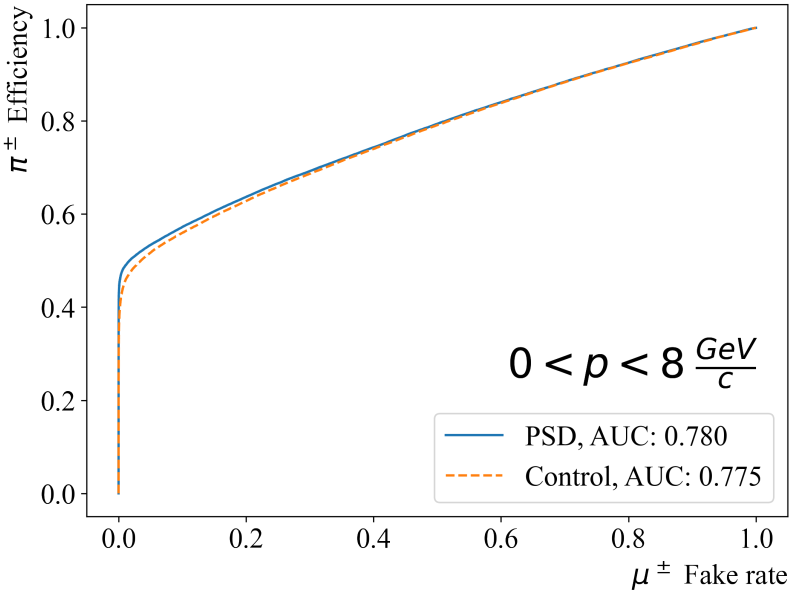


Figure 5.4: Inverted targets ROC curve of the $\mu^\pm$, $\pi^\pm$ discriminator trained over the entire momentum range compared to the model trained on neutral clusters.

Momentum-Binned

Figure 5.6 shows the momentum binned training of the GBDT model. As expected, the model offers only slight improvement over the shower-shape based version. The main area of effect is for muons in the $p < 1$ GeV region.

Figure 5.7 shows the probability distribution for the momentum trained model. The distributions are similar to that of the non-binned model allowing for the selection of pure $\pi^\pm$ samples. This is further shown in the ROC curve for inverted targets as seen in figure 5.8.

5.5.2 Classification of $e^\pm$, $\pi^\pm$

In this section, the models trained to classify $e^\pm$, $\pi^\pm$ are shown. With the shape variables added, the addition of PSD to the training brings considerable improvement to the discrimination potential compared to the regular cluster-shape based method.

Non-Binned

Figure 5.9a shows the ROC curve of a GBDT trained on the entire training set. A considerable improvement can be seen compared to the model without PSD showing that extra information can be gained from the PSD signature of the showers.

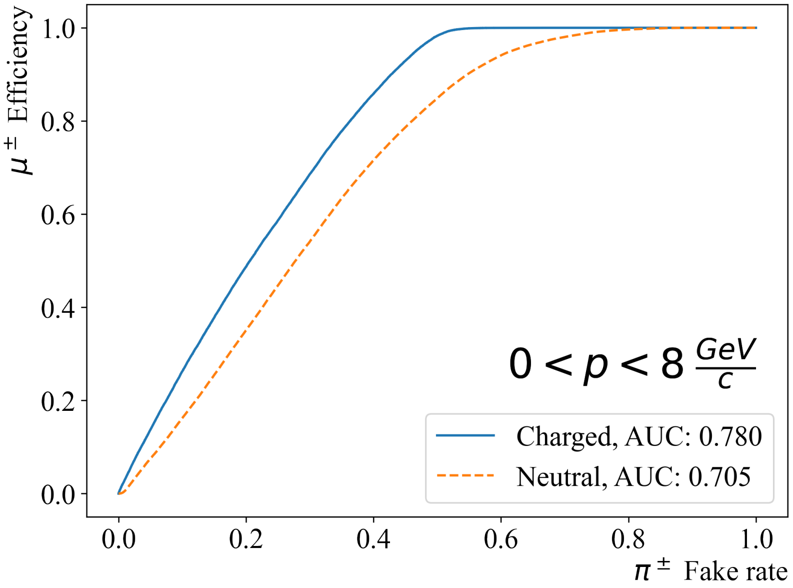


Figure 5.5: ROC curve of the $\mu^\pm$, $\pi^\pm$ discriminator trained over the entire momentum range compared to the model trained on neutral clusters.

Figure 5.9b shows the probability distribution of the GBDT trained on the entire training set. This representation visually shows that pure samples of pions and electrons can be obtained.

Figure 5.10 shows the comparison between the currently implemented neutral model and the new charged model. As expected, the charged model performs significantly better than the neutral model when discriminating for charged particles.

Momentum-Binned

Figure 5.11 shows the momentum binned training. Higher energy electrons are easily identifiable with shape variables, and improvements are seen in the lowest momentum bin of $p < 2$ GeV whith the biggest improvement taking place in the discrimination of particles with $p < 1$ GeV.

Figure 5.12 shows the probability distributions in momentum bins. The distributions are similar to that of the non-binned model, and all models offer the possibility of obtaining pure samples for both particles.

5.6 Next Steps

The principal next step of this project consists of testing the models on data samples. A sample of muons can be obtained from the decay $e^+e^- \to \mu^+\mu^-$ where final state radiation can be used to obtain a continuum of muon momenta. A sample of pions can be obtained using three-prong τ decays $e^+e^- \to \tau^+\tau^- \to (\mu^+\bar{\nu}_\tau\nu_\mu)(\pi^-\pi^+\pi^-\nu_\tau)$. Finally, a sample of electrons can be obtained from $e^+e^- \to e^+e^-$ scattering where final state radiation can be used to obtain a continuum of electron momentums.

Furthermore, the lack of strong PSD dependence in the $\mu^\pm$, $\pi^\pm$ classifier may imply that the current model is not sensitive enough to the PSD variables. A training of the models at lower momenta should be performed where the 1 GeV bin can be separated by using a momentum threshold where $\mu^\pm$ no longer reach the solenoid magnet (and do not cross over to the KLM). The momentum at which muons stop reaching the solenoid magnet is around 600 MeV, though a more precise analysis is necessary as this momentum is different in different areas of the detector. In addition, the separation of training data into high momentum bins in both $e^\pm$ and $\mu^\pm$ was found to not provide much improvement to their classification, and as such these bins can be collapsed into fewer bins while the model focuses on lower momentum bins.

Lastly, before implementing this tool inside the Belle II analysis software, the models should be trained by charge. This is already done in the current BDT models used for charged particle identification at Belle II, and provides more accurate discrimination results overall.

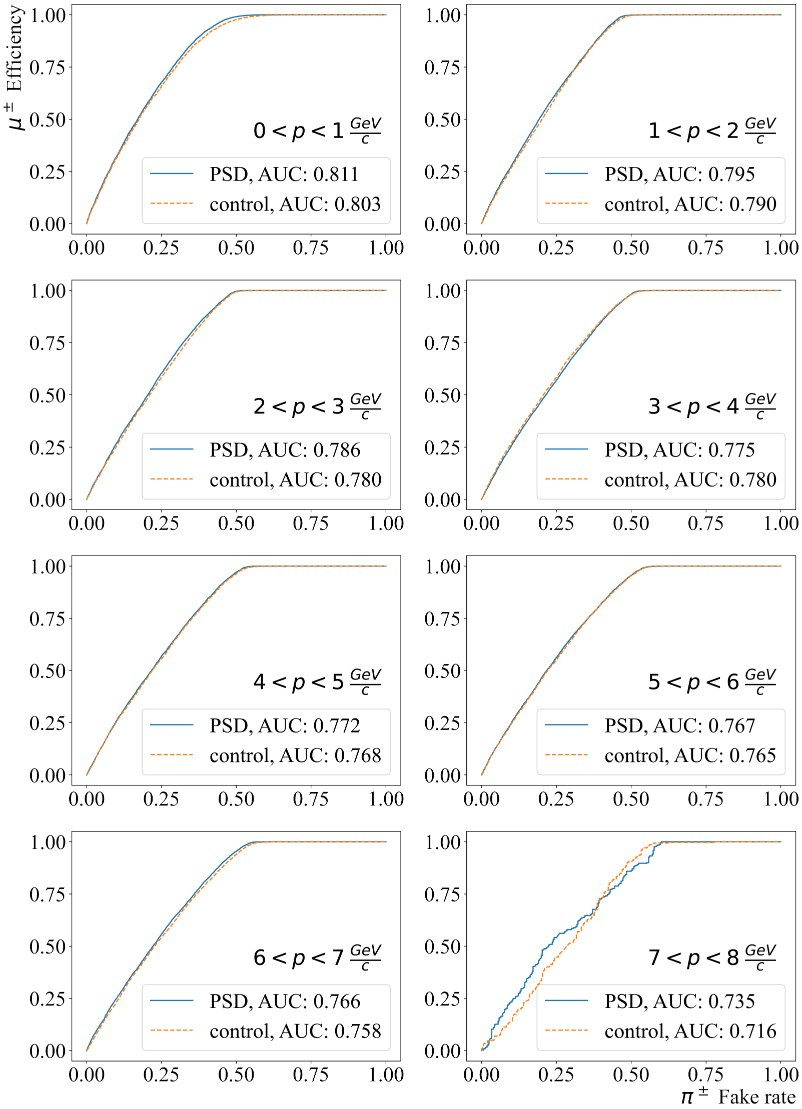


Figure 5.6: ROC curve of the $\mu^{\pm}$, $\pi^{\pm}$ discriminator trained in momentum bins, compared to the version without PSD (control). The AUC is shown in the legend for each metric.

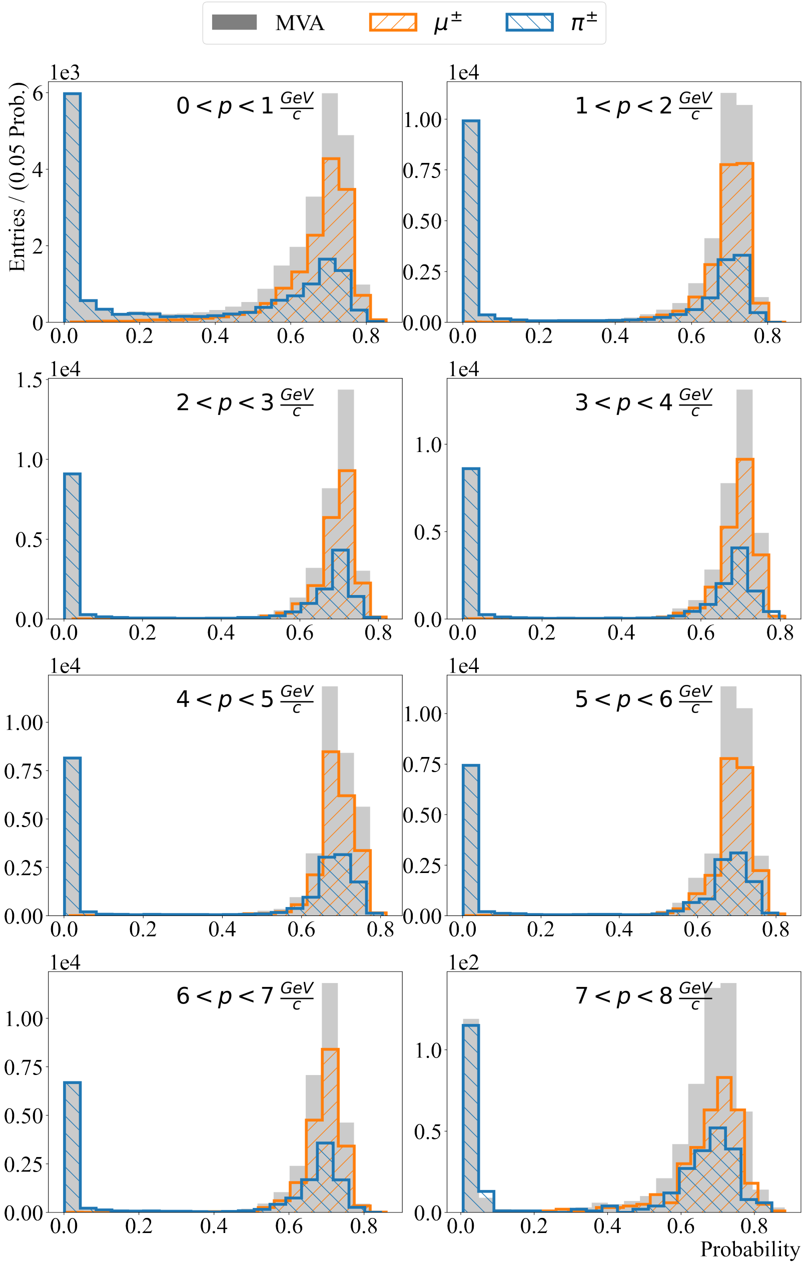


Figure 5.7: Probability output of the $\mu^{\pm}$, $\pi^{\pm}$ discriminator trained in momentum bins.

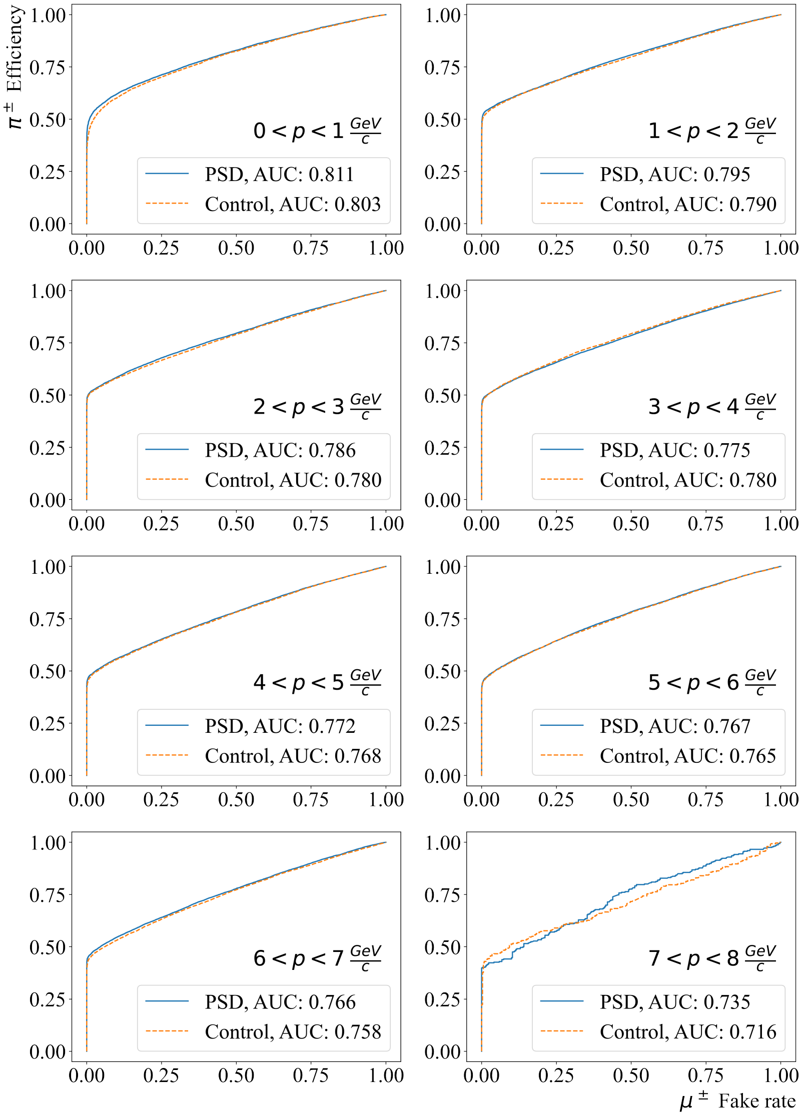
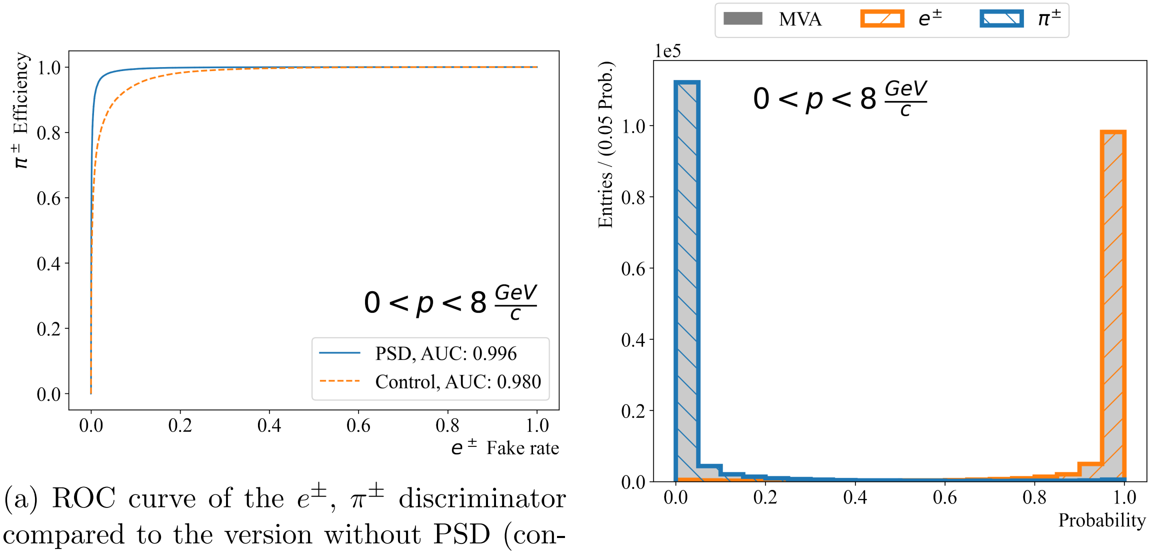


Figure 5.8: Inverted target ROC curve of the $\mu^\pm$, $\pi^\pm$ discriminator trained in momentum bins, compared to the version without PSD (control). The AUC is shown in the legend for each metric.



(a) ROC curve of the $e^{\pm}$, $\pi^{\pm}$ discriminator compared to the version without PSD (control). The AUC is shown in the legend for each metric.

(b) Probability output of the $e^{\pm}$, $\pi^{\pm}$ discriminator.

Figure 5.9: Metrics of the $e^{\pm}$, $\pi^{\pm}$ discriminator trained over the entire momentum range.

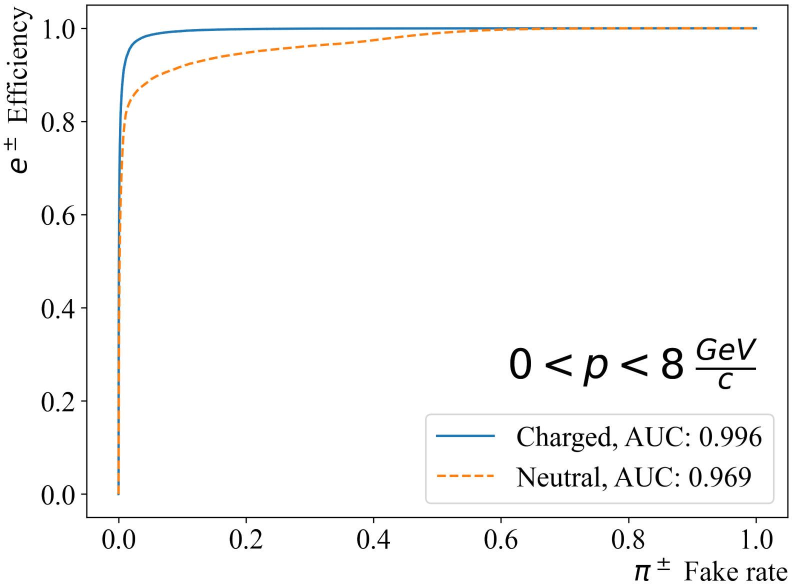


Figure 5.10: ROC curve of the $e^{\pm}$, $\pi^{\pm}$ discriminator trained over the entire momentum range compared to the model trained on neutral clusters.

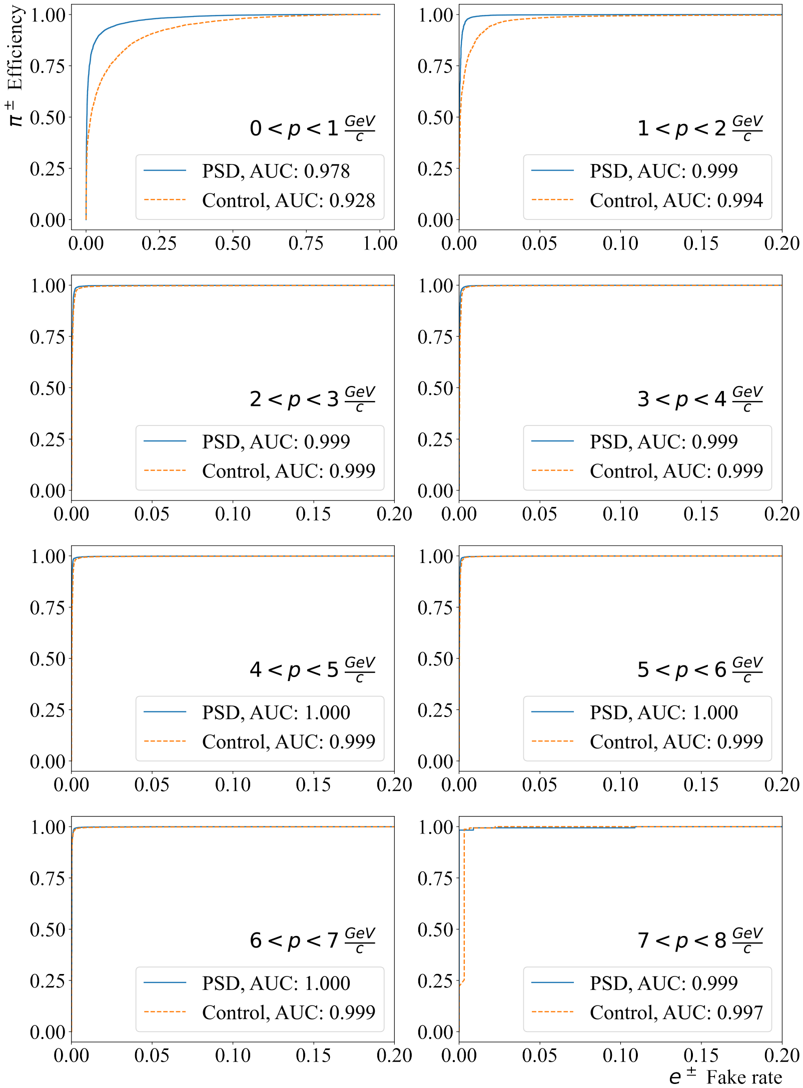


Figure 5.11: ROC curve of the $e^{\pm}$, $\pi^{\pm}$ discriminator trained in momentum bins, compared to the version without PSD (control). The AUC is shown in the legend for each metric.

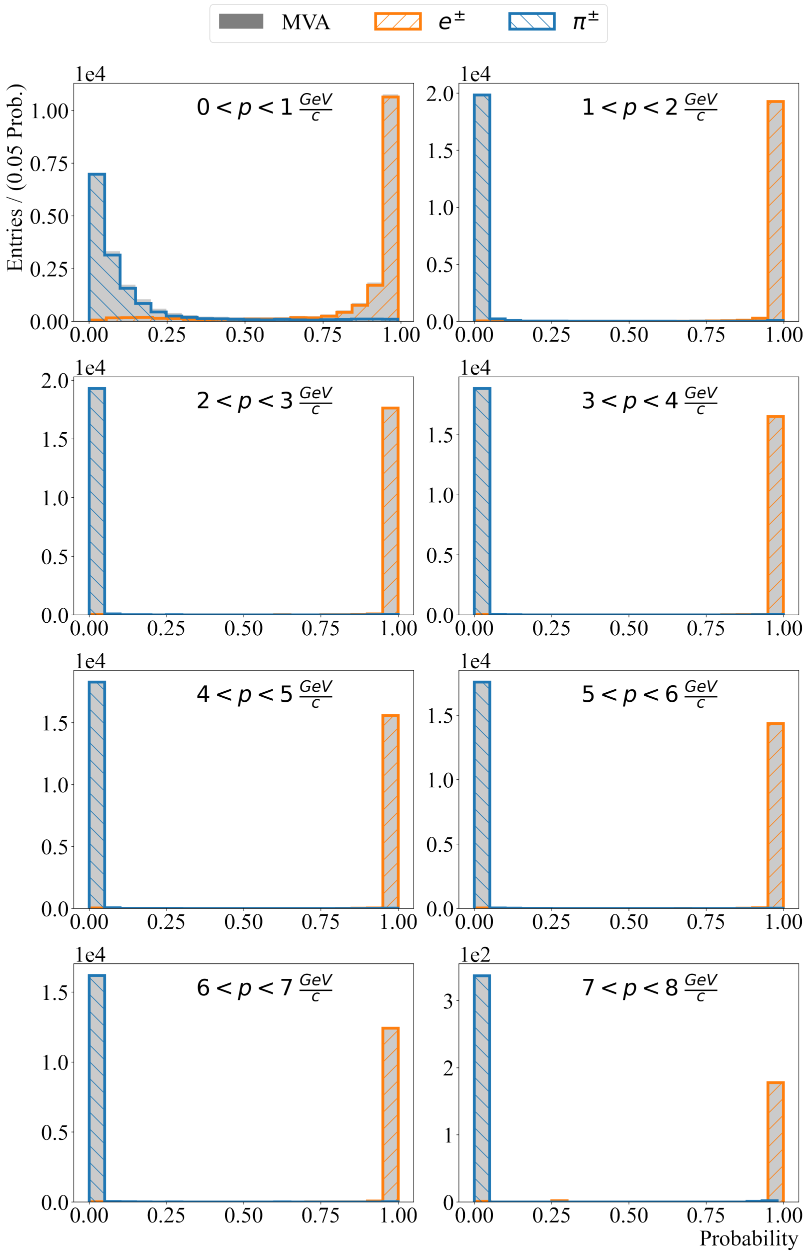


Figure 5.12: Probability output of the $e^{\pm}$, $\pi^{\pm}$ discriminator trained in momentum bins.

Chapter 6

Conclusion

This thesis covered two projects related to the use of machine learning techniques in the Belle II software framework.

The first project explored the use of generative adversarial networks (GANs), and more specifically GAN architectures using deep convolutional networks (DCGANs), as a mean to simulate background waveforms in the ECL. The studies showed that a DCGAN trained on background waveforms recorded during physics runs can generate waveforms similar to data. The comparison is done with multiple metrics: the χ^2 from a flat pedestal, the maximum amplitude difference, the average amplitude difference and the discrete frequency spectrum of the waveforms. The similarity between the simulated and data waveforms is quantified using a two-sample Kolmogorov-Smirnov (KS2) test on the distribution of each metric for the data waveform sample and a generated waveform sample. The model with the most accurate simulation is selected by finding the training step with lowest averaged KS2 score. GANs have shown to be accurate in their generation, but long and difficult to train. Next steps were described regarding the implementation of such a tool in Belle II, and modifications to improve speed and quality were suggested.

The second project detailed the development of a machine learning based particle discriminator to identify between charged hadrons and leptons in the ECL. The novel PSD tool implemented in the ECL CsI(Tl) crystals is used to obtain information about the contributions of ECL showers coming from hadron interactions. Two models are trained: one to provide $\mu^{\pm}$-$\pi^{\pm}$ discrimination and another to provide $e^{\pm}$-$\pi^{\pm}$ discrimination. The results showed an improvement in the discrimination capabilities of the ECL when using PSD instead of only using shower-shape based methods. The $\mu^{\pm}$-$\pi^{\pm}$ discriminator showed little improvement at higher momentums ($p > 1$ GeV), and a small improvement at low momentum ($p < 1$ GeV). The $e^{\pm}$-$\pi^{\pm}$ discriminator showed overall big improvements, with all the improvements taking place in the low momentum region ($p < 1$ GeV). Both classifiers showed an improvement in the discrimination of charged particles compared to the currently used PSD model which was trained on neutral clusters (γ, K_L^0). The model chosen for the discrimination were gradient boosted decision trees trained on simulated particle interactions with the ECL. Next steps were detailed on the implementation of this tool, and on the potential uses for it.

The work done in this thesis is part of a widespread effort to optimize the software framework used in Belle II by including the latest computer science techniques availabe. The usage of PSD contributes to the improvement of Belle II particle identification capabilities, and hence to the quality of its physics output. The development of GAN based simulations, on the other hand, pushes the Belle II collaboration towards advanced simulation techniques being spearheaded in other areas of academia. The usage of machine learning and other artificial intelligence tools in high energy physics has been on the rise for some time, and some tools widely used in other fields show great promise to the improvement of physics research.

Bibliography

[1] KEK. Superkekb collider achieves the world's highest luminosity. https://www.kek.jp/en/newsroom/2020/06/26/1400/. [Online, accessed 2021-11-10].

[2] Belle II. SuperKEKB and Belle II. https://www.belle2.org/project/super_kekb_and_belle_ii/. [Online, accessed 2021-11-10].

[3] E Kou, P Urquijo, W Altmannshofer, F Beaujean, G Bell, M Beneke, I I Bigi, F Bishara, M Blanke, C Bobeth, and et al. The belle ii physics book. *Progress of Theoretical and Experimental Physics*, 2019(12), Dec 2019.

[4] P.A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020.

[5] Y. Fukuda, T. Hayakawa, E. Ichihara, K. Inoue, K. Ishihara, H. Ishino, Y. Itow, T. Kajita, J. Kameda, S. Kasuga, and et al. Evidence for oscillation of atmospheric neutrinos. *Physical Review Letters*, 81(8):1562–1567, Aug 1998.

[6] A. W. P. Poon. Neutrino observations from the sudbury neutrino observatory. *AIP Conference Proceedings*, 2002.

[7] CERN. The Large Hadron Collider. https://home.cern/science/accelerators/large-hadron-collider. [Online, accessed 2021-11-10].

[8] G. Aad, T. Abajyan, B. Abbott, J. Abdallah, S. Abdel Khalek, A.A. Abdelalim, O. Abdinov, R. Aben, B. Abi, M. Abolins, and et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, Sep 2012.

[9] K. et al. Abe. Observation of large CP violation in the neutral B meson system. *Phys. Rev. Lett.*, 87:091802, Aug 2001.

[10] B. et al. Aubert. Observation of CP violation in the b^0 meson system. *Phys. Rev. Lett.*, 87:091801, Aug 2001.

[11] T. Abe and al. Belle II Technical Design Report, 2010.

[12] Y. Makida, H. Yamaoka, Y. Doi, J. Haba, F. Takasaki, and A. Yamamoto. *Development of a Superconducting Solenoid Magnet System for the B-Factory Detector (BELLE)*, pages 221–228. Springer US, Boston, MA, 1998.

[13] S. Longo. First application of csi(tl) pulse shape discrimination at an electron-positron collider to improve particle identification at the belle ii experiment, 2019.

[14] S. Agostinelli et al. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[15] Doris Y Kim et al. The simulation library of the belle ii software system. unpublished internal document. May 2017. CHEP 2016.

[16] T. Kuhr T. Hara and Y. Ushiroda. Belle ii coordinate system and guideline of belle ii numbering scheme. unpublished internal document. August 2011.

[17] I. Adachi, T.E. Browder, P. Križan, S. Tanaka, and Y. Ushiroda. Detectors for extreme luminosity: Belle ii. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 907:46–59, 2018. Advances in Instrumentation and Experimental Methods (Special Issue in Honour of Kai Siegbahn).

[18] A. Abashian and al. The belle detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 479(1):117–232, 2002. Detectors for Asymmetric B-factories.

[19] A. Kuzmin. Electromagnetic calorimeter of belle ii. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 958:162235, 2020. Proceedings of the Vienna Conference on Instrumentation 2019.

[20] S Lee, R Itoh, N Katayama, and S Mineo. Development of high level trigger software for belle II at SuperKEKB. 331(2):022015, dec 2011.

[21] C. Hearty T. Ferber and J. M. Roney. Design of the ecl software for belle ii. unpublished internal document. January 2016.

[22] S. Longo, J.M. Roney, C. Cecchi, S. Cunliffe, T. Ferber, H. Hayashii, C. Hearty, A. Hershenhorn, A. Kuzmin, E. Manoni, F. Meier, K. Miyabayashi, I. Nakamura, M. Remnev, A. Sibidanov, Y. Unno, Y. Usov, and V. Zhulanov. Csi(tl) pulse shape discrimination with the belle ii electromagnetic calorimeter as a novel method to improve particle identification at electron–positron colliders. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 982:164562, 2020.

[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[25] Remi Delassus. *Apprentissage automatique pour la détection d'anomalies dans les données ouvertes : application à la cartographie*. Theses, Université de Bordeaux, November 2018.

[26] D. A. Darling. The kolmogorov-smirnov, cramer-von mises tests. *The Annals of Mathematical Statistics*, 28(4):823–838, 1957.

[27] K.A. Olive. Review of particle physics. *Chinese Physics C*, 40(10):100001, oct 2016.

[28] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras and tensorFlow*. O'Reilly Media, 2 edition, 2019.