Master's Thesis

Improving the Belle II Neural Track Trigger with Deep Neural Networks

Submitted by
Timo Forsthofer

Supervised by
Prof. Dr. Christian Kiesling

Ludwig-Maximilians-University Munich
Faculty of Physics

Munich, 30th of June 2024

Masterarbeit

Verbesserung des Belle II Neural Track Trigger mit tiefen neuronalen Netzwerken

Vorgelegt von
Timo Forsthofer

unter Betreuung von
Prof. Dr. Christian Kiesling

Ludwig-Maximilians-Universität München
Fakultät für Physik

München, den 30.06.2024

Abstract

In order to provide precision tests of the Standard Model and to enable searches for new physics at the intensity frontier, the SuperKEKB asymmetric energy electron positron collider, housing the Belle II detector, aims to reach an instantaneous luminosity of $6 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. An important tool for coping with the resulting high background levels is the Neural Track Trigger (NTT). This first level track trigger, based on the wire information from the Central Drift Chamber (CDC), makes use of a neural network to predict the origin ("vertex") of events along the beam ("z") direction, rejecting tracks with a large displacement. In previous running periods, it has been very successful in triggering particularly low multiplicity events with a high efficiency while maintaining a low trigger rate. However, during high luminosity runs before the Long Shutdown (LS1) in June 2022, the trigger rate drastically increased due to excessive background, making an upgrade necessary to stay within the rate limits of the data acquisition system.

This thesis studies how new trigger hardware and new software developments can be utilized for optimizing the performance of the NTT, which at present is realized in a single hidden layer neural architecture. As a first step, different deep neural network architectures within the hardware restrictions are tested. Next, the number of input nodes is increased to include more information from the CDC, thereby enhancing the z resolution. Using noise suppression in the CDC wires and an enhanced track finding model, which has become available recently, background and fake tracks are suppressed and the quality of the input parameters is improved. Finally, an additional output node for classification is added. All of this significantly improves both the single-track efficiency and the background rejection rate, making the trigger stable even at the high background rates expected in the future.

Contents

Chapter 1

Introduction

In order to challenge the Standard Model (SM) at the precision frontier, the SuperKEKB asymmetric energy electron positron collider, housing the Belle II detector, aims to reach an instantaneous luminosity of $6 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [1]. With background processes making up a dominant share of events, a good trigger system is necessary for keeping the trigger rate within the bounds given by the data acquisition system. The first level (L1) of this trigger works on a FIFO ("first in first out") pipeline that can only store data for $5\,\mu\mathrm{s}$, so it needs to be implemented on fast field-programmable gate arrays (FPGAs). This means that there are not only strict time constraints, but also restrictions on computing power. The CDC trigger uses track finding algorithms for making a trigger decision, with the aim of a high efficiency especially for low multiplicity events, which are difficult to discriminate from the background. From 2018 to 2020, this trigger relied on a Track Segment Finder (TSF) and a track finder using a 2-dimensional Hough transformation (2DFinder). An event was accepted, if two or more of those 2-dimensional tracks were found, not using any information from the beam ("z") axis. As the machine background causes tracks mainly originating from the beam pipe, many uninteresting events were recorded, resulting in a high trigger rate that saturated the data acquisition system. Therefore, a Neural Track Trigger (NTT) was installed, where tracks found by the 2DFinder are combined with stereo track segments and fed to a neural network [2]. The network provides an estimate for the z-origin ("vertex") of a track, so that tracks with large values of $|z|$ can be rejected. Using this concept, a minimum bias Single Track Trigger (STT) was installed, where a single track with $|z| < 15\,\mathrm{cm}$ and a momentum $p > 0.7\,\mathrm{GeV}$ is enough to make an unprescaled trigger decision. Events with two or more 2D track candidates are also accepted, but at least one neural track is required in addition.

This trigger was implemented until the Long Shutdown (LS1) in June 2022 and worked very well, but there were still some problems. In the last running period before LS1, the machine background increased drastically, leading to the STT taking up 50 % of the total trigger budget instead of the usual 20 % [2]. When approaching the target luminosity, this problem is expected to get even worse. Luckily, new FPGAs, the UT4 boards, allow for the implementation of deeper neural networks and a number of other innovations, which can improve the z prediction of the network.

The goal of this thesis is to study the impact of deep neural network architectures and improved input on the network performance and optimize the network while still keeping it implementable into hardware. Chapter 2 gives a short theoretical motivation of the Belle II experiment and events that require a high trigger efficiency are presented. In chapter 3, an overview of the Belle II experiment is given, with a special focus on the Central Drift Chamber (CDC). An introduction to the current CDC trigger is provided in chapter 4. Here, the main focus is the neural network, as this is what most of this thesis is about. Chapter 5 discusses the data that was used for training and evaluating the neural networks. In chapter 6, the performance of the network installed before LS1 is evaluated using different plots and quantities that are presented in this chapter and used throughout the thesis. Chapter 7 investigates the effect of retraining a network on more noisy data and tests different network architectures. The benefits of using expert networks and quantization aware training are also evaluated. New approaches to improving the network input are studied in chapter 8, including extended input, an ADC-cut and input from the 3DFinder described in [3] and [4]. In chapter 9, some more ways to possibly optimize the network performance are explored, most notably the introduction of a classification output node, that can be used for triggering instead of the z-cut. In chapter 10, all modifications to the original network are summarized, and the resulting performance is compared to the previously installed network. Finally, chapter 11 gives a conclusion of the thesis and an outlook on future developments.

Chapter 2

The Standard Model of Particle Physics

While the Standard Model (SM) of particle physics is, along with general relativity, one of the most successful and best-tested theories in physics, there are still some open questions that it fails to answer. One example is the asymmetry of matter and antimatter in the universe, which cannot be explained with SM processes. The SM also has no explanation for dark matter and dark energy, which are necessary to explain the movement of galaxies and the expansion of the universe. There is also strong evidence for neutrino mixing, which requires a neutrino mass [5], and the anomalous magnetic moment of the muon shows a discrepancy of more than 4σ from the standard model [6]. It is therefore important to thoroughly study known deviations from the SM and find new ones.

Contrary to experiments like the Large Hadron Collider at CERN, which searches for new physics at the energy frontier, the Belle II experiment is working on the intensity frontier [7]. This means that SM parameters are measured with high precision using a high luminosity collider in order to find small deviations from the theoretical predictions.

2.1 CKM-Matrix

In the SM, the charge conjugation symmetry (C) and the parity symmetry (P) are maximally violated, as W bosons do not couple to right-handed quarks and C maps left-handed fermion fields to right-handed anti-fermion fields [8]. However, in order to explain the asymmetry between matter and antimatter in the universe, a combination of both, i.e. a CP violation is needed [9]. This asymmetry is described by the Cabibbo-Kobayashi-Masakawa (CKM) matrix [10].

The Lagrangian that characterizing the coupling of a W boson to quarks is given as [8]

$$\mathcal{L}_W^q = \frac{g}{\sqrt{2}} \left[V_{jk} \bar{u}_{Lj} \gamma^\mu d_{Lk} W_\mu^+ + V_{jk}^* \bar{d}_{Lk} \gamma^\mu u_{Lj} W_\mu^- \right], \tag{2.1}$$

where V_{jk} are the elements of the unitary CKM matrix

$$\mathbf{V} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}. \tag{2.2}$$

An example for a process described by the Lagrangian in eq. (2.1) is given in fig. 2.1.

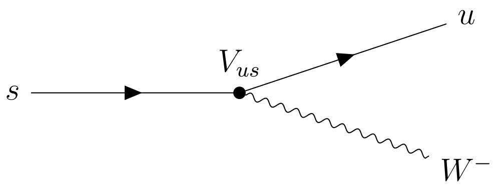


Figure 2.1: Feynman diagram of a strange quark decaying to an up quark. The coupling strength is proportional to the mixing matrix element V_{us}.

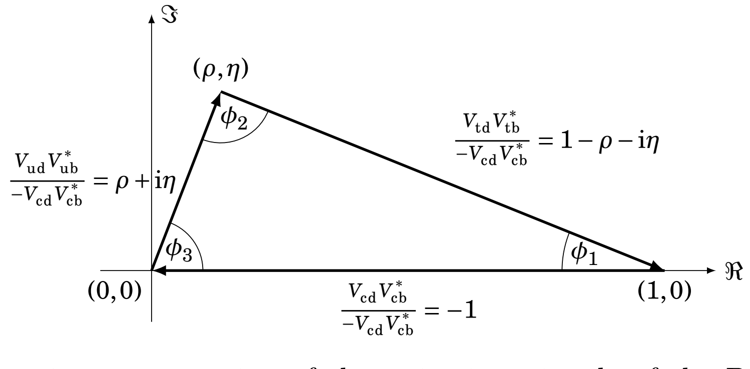
Using the unitary condition of $\mathbf{V}\mathbf{V}^\dagger = 1$, the matrix can be parametrized by the four Wolfenstein parameters and written as [11]:

$$\mathbf{V} = \begin{pmatrix} 1-\frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1-\frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1-\rho-i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4), \tag{2.3}$$

where from experimental observations it is known that $\lambda \approx 0.22$. There are now six equations resulting from the unitarity condition, the most interesting one for the Belle II experiment being [7]

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0. \tag{2.4}$$

In the complex plane it can be represented as a triangle, as shown in fig. 2.2. The area of the triangle is proportional to the strength of the CP violation.



$\Im$
(ρ,η)
ϕ_2
$\frac{V_{td}V_{tb}^*}{-V_{cd}V_{cb}^*} = 1-\rho-i\eta$
$\frac{V_{ud}V_{ub}^*}{-V_{cd}V_{cb}^*} = \rho+i\eta$
ϕ_3
ϕ_1
$\Re$
$(0,0)$
$\frac{V_{cd}V_{cb}^*}{-V_{cd}V_{cb}^*} = -1$
$(1,0)$

Figure 2.2: Geometric representation of the unitarity triangle of the B meson system in the complex plane [7]. The equation was divided by $-V_{cd}V_{cb}^*$, fixing two of the corners at (0, 0) and (1, 0).

At the original Belle experiment, the CP asymmetry was measured, confirming the Kobayashi Masakawa theory and leading to a Nobel Prize in 2008 [12]. However, the effect is not strong enough to explain the asymmetry of matter and antimatter in the universe [7]. The standard model also does not include any processes violating baryon number conservation, which would be necessary to explain the complete absence of macroscopic regions containing antimatter in the universe [7].

The Belle II experiment is therefore aiming to measure the unitarity triangle with a higher precision in order to find deviations from the standard model. This is done by colliding electrons and positrons at a center-of-mass energy of 10.58 GeV, which is equivalent to the mass of the $\Upsilon(4S)$ meson. This way, two B mesons can be created, either $B^0\bar{B}^0$ or $B{+}B{-}$, which can then be studied extensively.

As these processes usually have a high multiplicity, they are not the main focus of the track trigger discussed in this thesis, but it is still important to ensure a high trigger efficiency on these events.

2.2 Rare Decays

Apart from measuring parameters of the SM more precisely, the Belle II experiment is also searching for rare processes not described by the SM. These include for example invisible B decays, where one of the B mesons decays into an invisible final state, e.g. $B^0 \to \nu\nu$ [3]. As two B mesons are produced at a time, decays like this can be reconstructed using the decay products of the other B meson. Belle II also has a large cross section for τ pair production (compare table 3.1), so lepton number violations in decays like $\tau \to \ell\ell\ell$ or $\tau \to \mu\gamma$ may be observed [3]. Collisions at Belle II could also produce dark photons A', that couple to the electromagnetic force. They can be observed either by missing energy or by decays into SM particles with a displaced vertex [13].

Many of those rare decays have a low multiplicity, so it is very important to trigger with a high efficiency when only few tracks are available.

Chapter 3

The Belle II Experiment

The asymmetric electron-positron collider KEKB was operated in Tsukuba, Japan, from 1998 to 2010, achieving a world record instantaneous luminosity of $2.11 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [14]. With high statistic measurements on B mesons, the Belle experiment has proven the CP-violation in the quark sector as predicted by Kobayashi and Maskawa [1] [10]. In order to reach even higher luminosities to challenge the Standard Model, the collider was replaced with a new SuperKEKB, while Belle was upgraded to Belle II.

3.1 SuperKEKB

SuperKEKB was built inside the original tunnel of KEKB, with a circumference of 3 km. A sketch of the accelerator structure can be seen in fig. 3.1. A linear accelerator (LINAC) is used to accelerate the positrons to 4.0 GeV and the electrons to 7.0 GeV. The electrons are then injected into the high-energy ring (HER), the positrons into the low-energy ring (LER). At the interaction point (IP), both beams cross each other with an angle of 83 mrad after having been focused by superconducting quadrupole magnets.

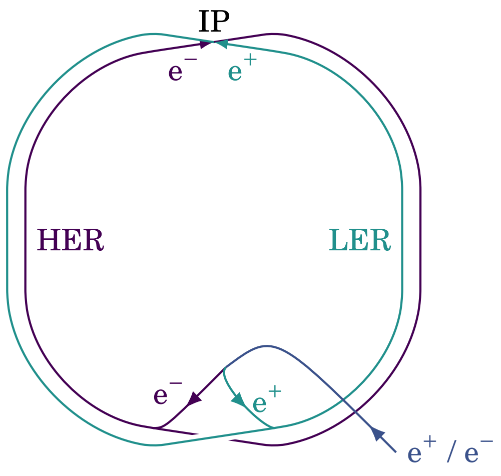


Figure 3.1: Schematic overview of the SuperKEKB accelerator [7]. The electrons and positrons are first accelerated in LINAC, then stored in HER (electrons) and LER (positrons).

As previously mentioned, the purpose of SuperKEKB is to surpass the luminosity of KEKB, given by [7]

$$\mathcal{L} = \frac{N_+ N_- f_c}{4\pi\sigma_x\sigma_y}, \tag{3.1}$$

where $N_{+/-}$ are the particle numbers per beam, f_c is the crossing frequency and $\sigma_{x/y}$ are the horizontal and vertical bunch sizes at the crossing point. About 2500 bunches of both electrons and positrons are circulating in each storage ring, so that there is a collision approximately every four nanoseconds [15]. The beam currents have design values of 3.60 A (positrons) and 2.60 A (electrons). In order to decrease the beam sizes, a novel nano beam scheme has been implemented [16], making a vertical beam size of only about 60 nm possible [1]. Using those improvements, the original goal was an instantaneous luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. This estimate has by now been corrected to $6 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, which still beats the original collider by a factor of 30 [2].

With the beam energies provided by LINAC, a center-of-mass energy equivalent to the boosted $\Upsilon(4S)$ resonance of 10.58 GeV is maintained. The asymmetry of the beam energies causes a boost in the positive z direction, which is accounted for by an asymmetric detector design. A list of expected physics processes, together with their cross sections and rates at the original design luminosity can be seen in table 3.1. For making sure that all those processes and possibly new physics events can be detected and properly reconstructed, a general purpose particle detector is needed, which is described in the following section.

Physics process	Cross section (nb)	Rate (Hz)
$\Upsilon(4S) \rightarrow B\bar{B}$	1.2	960
Hadron production from continuum	2.8	2200
$\mu^+\mu^-$	0.8	640
$\tau^+\tau^-$	0.8	640
Bhabha ($\theta_{lab} \geq 17°$)	44	350[1]
$\gamma\gamma(\theta_{lab} \geq 17°)$	2.4	19[1]
2γ processes $\theta_{lab} \geq 17°, p_t \geq 0.1\,\mathrm{GeV/c}$)	~ 80	~ 15000
Total	~ 130	~ 20000

Table 3.1: Expected event rates at the original design luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [1].

3.2 Belle II Detector

Figure 3.2 shows a schematic overview of the detector. In the coordinate system used throughout this thesis, the z-axis is the symmetry axis of the detector, also defining the direction of the solenoidal magnetic field, with the origin at the interaction point. The polar angle θ of a track is measured from the positive z-axis and has an acceptance range from about 17° to 150°, accounting for asymmetry created by the boost. The azimuth angle ϕ is measured from the x-axis and has a full 360° acceptance range.

[1]Rates of already well studies processes are pre-scaled by a factor of 1/100.

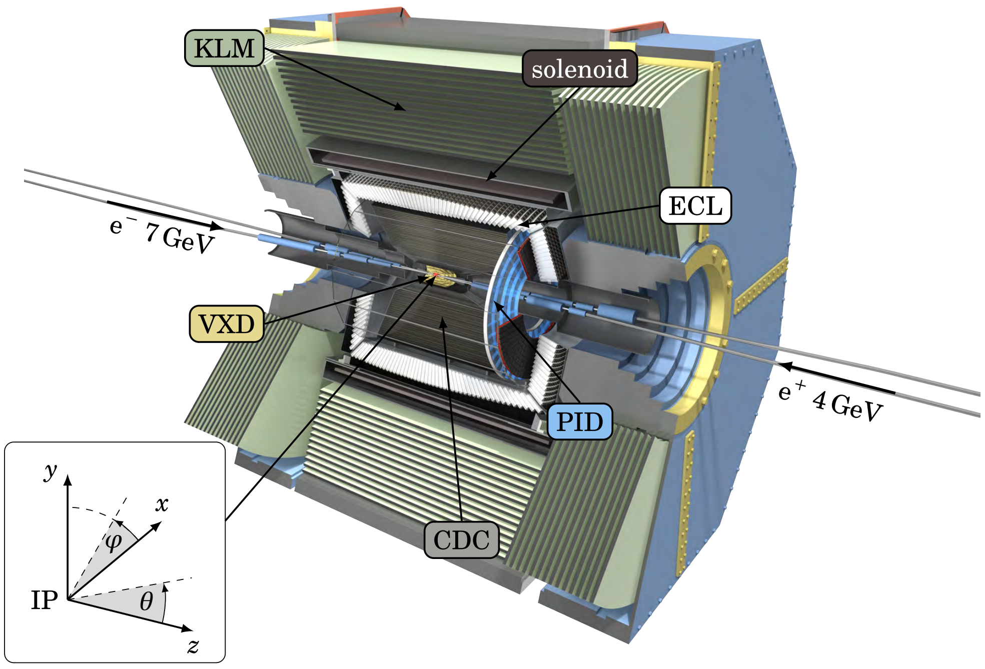


Figure 3.2: Overview of the Belle II detector including the coordinate system [17].

The innermost subdetector is a vertex detector (VXD), which consists of a pixel detector (PXD) and a silicon strip detector (SVD). The PXD is equipped with in total 8×10^6 pixels and can provide very precise vertex position measurements using DEPFET (DEPleted Field Effect Transistor) technology [1]. In the future it may be possible to use it for identifying slow pions that do not leave the VXD [18]. The SVD consists of four layers of double-sided silicon strip detectors [1] and, together with the PXD, makes a spatial resolution of 15 µm possible.

The VXD is followed by the Central Drift Chamber(CDC), the main tracking detector of Belle II [7]. Since this is the subdetector used for the trigger that is the subject of this thesis, it will be described in more detail in section 3.3.

Outside the CDC there is a particle identification system (PID), which again includes two different subdetectors. The barrel region is covered with a Time-Of-Propagation (TOP) counter, and the forward end cap with an Aerogel Ring-Imaging Cherenkov (ARICH) detector [1]. Both use Cherenkov radiation for measuring the velocity of a particle. Using the momentum measurements from the tracking detectors, the mass and thus the identity of a particle can be determined.

An electromagnetic calorimeter (ECL) is then used for energy measurements and for identifying electrons and photons [1]. Electromagnetic particles that reach the ECL produce showers of e^+e^- pairs in the scintillating CsI(TI) crystals and photons and electrons usually fully deposit their energy in the ECL.

The next element is a solenoid that creates a homogeneous magnetic field of 1.5 T in the positive z direction.

Particles that pass the ECL are mainly K_L and muons, which are then detected by the

KLM in its alternating layers of iron plates and active detector elements [7].

3.3 Central Drift Chamber

The Central Drift Chamber (CDC) stretches from an inner radius of 16 cm to an outer radius of 113 cm. There are in total 14,336 sense wires and 42,240 field wires inside the CDC, so that every sense wire is surrounded by eight field wires, as shown in fig. 3.3. It is filled with a gas mixture of 50 % helium and 50 % ethane [1]. Charged particles ionize the gas, leaving free electrons along their track. An electric field is created by the field wires, so that the free electrons drift to the sense wires with an almost constant velocity of 40 µm ns^{-1} [7]. Therefore, the drift time can be used for calculating the distance of the particle track from the sense wire, with a spatial resolution of about 100 µm.

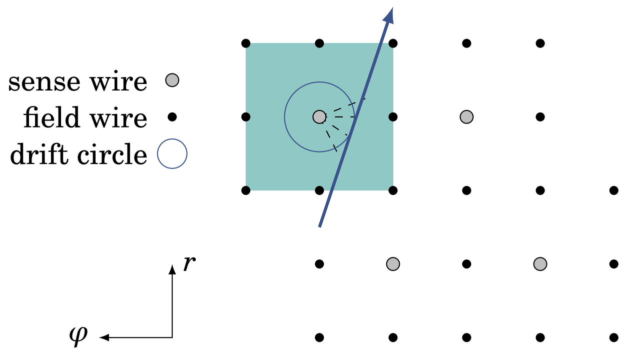


Figure 3.3: Illustration of a particle passing through the CDC [7]. Gas particles are ionized and free electrons drift towards the sense wires due to the electric field created by the field wires.

The wires are arranged in 56 layers, which are again organized into nine so-called superlayers, in which all layers have the same number of wires. Every layer in the first superlayer has 160 wires, while the number of wires grows to 384 in the last layers [1]. The superlayers alternate in orientation between axial and stereo superlayers, with the first and last one being axial superlayers. Axial layers are parallel to the z-axis and therefore do not provide any information on the z coordinate. This is where the stereo layers come into play. They are slightly skewed, meaning that the azimuth angle ϕ is not constant over the length of the wire (see fig. 3.4).

The homogeneous magnetic field generated by the solenoid outside the ECL plays an important role in the CDC. Using the Lorentz force, it causes a curvature of all charged particles passing through the detector. The corresponding transverse momentum p_T can be calculated from the two-dimensional radius by equating the Lorentz force and the centripetal force:

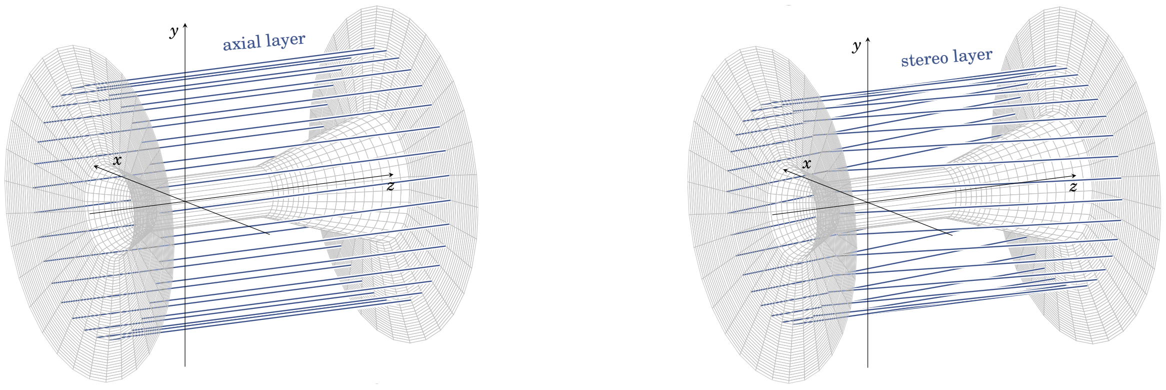


Figure 3.4: Schematic illustration of axial wires, which are parallel to the z-axis, and stereo wires, which are slightly skewed [7].

$$\begin{aligned} |\vec{F_L}| &= |\vec{F_c}| \\ q\,|\vec{v} \times \vec{B}| &= m \cdot \frac{v^2}{r} \\ \frac{q\,p_T B}{m} &= \frac{p_T^2}{m\,r} \\ \Rightarrow p_T &= q\,r B \end{aligned} \tag{3.2}$$

Particles with a momentum of about 254 MeV/c have a radius of 56.5 cm, so they will only travel 113 cm from the beam pipe and not leave the CDC. Therefore, these particles will not be found by the ECL trigger, but with missing hits in the outer superlayers the CDC trigger will also become less precise.

Chapter 4

The CDC Trigger at Belle II

4.1 Expected Background

Apart from the physical events presented in chapter 3, there is a dominating share of background events that do not originate from the IP. The task of the Neural Track Trigger (NTT) is to reject those events. They arise in the following ways: [19]

- **Touschek effect**: This happens when particles interact inside a bunch, exchanging some transverse momentum. Subsequently, both particles will leave the design orbit and eventually hit the beam pipe, causing a shower that can be detected if it is close enough to the interaction region.
- **Beam-gas scattering**: While in principle the inside of the beam pipe is highly evacuated, there will always be some residual gas particles with which the electrons and positrons can scatter, leading to tracks coming from outside the interaction point.
- **Synchrotron radiation**: The acceleration of the charged beam particles causes synchrotron photons with relatively low energies that can lead to ionization of gas particles in the CDC.
- **Luminosity background**: In QED events like Bhabha or two-photon processes, low energy electrons can be created and hit the beam pipe, causing particle showers near the interaction point.
- **Injection background**: In order to keep the beam current constant, there need to be regular injections of new particles into the beam bunches. During the first tens of milliseconds, they are subject to betatron oscillations around the original beam orbit. The strong magnetic fields in the interaction region may then deflect those particles into the detector.
- **Random noise**: Next to actual particles crashing into the detector, there are also some electronic artifacts, most notably the electronic cross-talk. This happens when a wire is activated and some signal is leaking within the preamplifier electronics to a different wire. Those events can contaminate the signal and lead to fake tracks.

4.2 Trigger System

The trigger system at Belle II consists of four first-level subtriggers that pass events in a deadtime-free pipeline to the Global Decision Logic (GDL). This has to happen within 5 µs, because the amount of data that can be stored by the data acquisition system is limited. Such fast calculations have to be executed directly on hardware, on field-programmable gate array (FPGA) boards. In the next step, the selected events are forwarded to the high level trigger (HLT) with a maximum rate of 30 kHz, where they are fully reconstructed, selected based on physics criteria and saved to permanent storage.

The first-level triggers include the CDC trigger, the ECL trigger, the TOP trigger, and the KLM trigger [20]. In the TOP trigger, precise timing information from the TOP subdetector is used to identify particles. The KLM trigger is used for identifying muon events and helps with detector calibrations. However, the main trigger decisions are made by the ECL trigger and the CDC trigger. The electromagnetic calorimeter provides total energy and cluster count information, allowing the ECL trigger to detect events with high energy deposition and multi-hadronic events. It can also identify Bhabha events, which due to their large cross section need to be heavily prescaled by the GDL in order to maintain a low overall trigger rate.

This thesis will only deal with the CDC trigger, which uses an artificial neural network to predict the origin of particle tracks and accept those that come from the IP. This approach provides high efficiency predominantly for low-multiplicity events that are not easily detected by the ECL trigger.

Before a reasonably sized neural network can be applied, the input from the 14,336 sense wires in the CDC has to be reduced to the relevant information. In order to do this, the signal from the front end electronics (FEE) is first organized into so-called track segments by the track segment finder. Aligning track segments are then combined into track candidates by the 2DFinder and the parameters are fed to the neural network. The latency for the neural network is currently only 300 ns, making deep networks with more than one hidden layer difficult to implement.

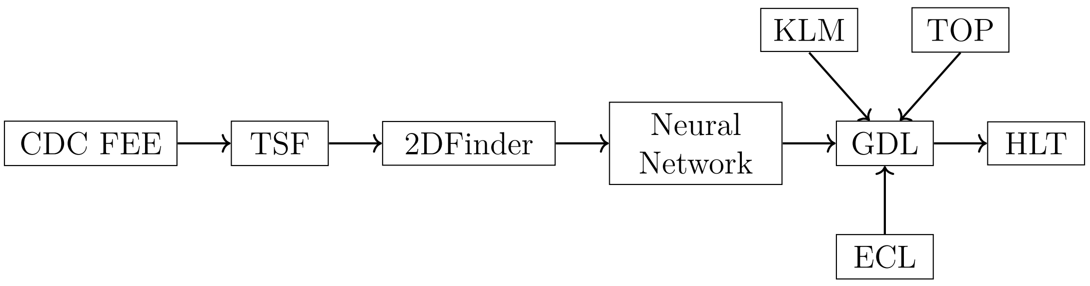


Figure 4.1: Data pipeline for the CDC trigger.

4.3 Track Segment Finder

The first step in reducing the input to a reasonable size is a track segment finder (TSF). Its task is to find track segments like the ones in fig. 4.2. In the first superlayer, the three innermost layers are not used due to high background levels, resulting in a pyramid shape (see fig. 4.2, left side). Track segments in all following superlayers have an hourglass shape, where the last layer is excluded (see fig. 4.2, right side). The green colored cells are the three possible priority wires, with the innermost one being the first choice. If it is missed, one of the two wires behind it will be selected. In case both of the secondary priority wires are hit, the lowest drift time acts as a tie-breaker.

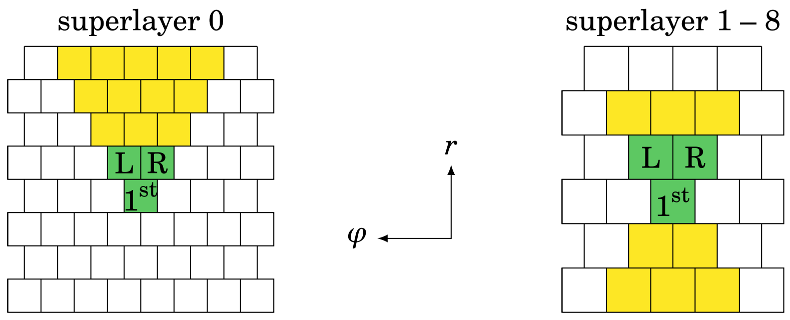


Figure 4.2: Schematic representation of track segments in different superlayers [7].

In order to find the track segments, the TSF uses Lookup Tables (LUT) with possible wire patterns and compares those with the hits in the CDC. This is already a first step to suppressing background, as random noise, indicated by the orange circles in fig. 4.3, is not found in the first place. The LUTs also provide an estimate on whether the particle passed left or right of the priority wire. Combined with the priority drift time, this allows for a precise determination of the space point of the track passing the track segment. In the neural network, the drift time is then taken as positive or negative depending on the left/right decision and set to zero if the direction cannot be determined (for example for a symmetric hit pattern).

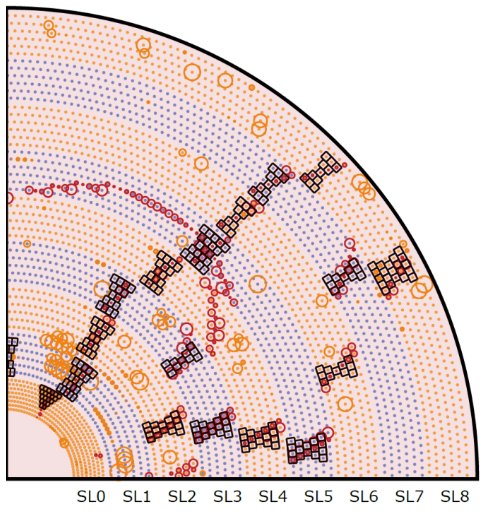


Figure 4.3: Example for found track segments in an event. The tracks appear as zigzag lines because of the skew angle of the stereo superlayers.

4.4 2DFinder

Now that there are candidates for where a particle might have passed through the detector, they have to be combined into possible tracks. For this purpose, a Hough transformation is used, first proposed in [21] in order to detect straight line segments in photographs of bubble chambers. For now, only axial track segments are included that do not contain any z information, as they are parallel to the z-axis.

The principle of a Hough transformation is that a point in geometrical space is transformed to a line in parameter space, the Hough curve [7]. All lines in parameter space corresponding to points on a line in geometrical space will then meet at the parameters of the original line. This way, a line can be found by looking for a point in the Hough space that is crossed by many Hough curves. As shown in [22], the Hough transform can be generalized to arbitrary shapes. In the case of the 2DFinder, it is used for detecting curved trajectories coming from the beam pipe.

Because only realistic physical tracks need to be found, a few assumptions can be made in order to reduce fake tracks. As all tracks should originate from the interaction point, it can be demanded that all tracks pass through the beam pipe, i.e. through $(x, y) = (0, 0)$. There is also a sufficiently homogeneous magnetic field in the CDC to assume every track to be part of a circle. Therefore, for a given point, all possible curves through the IP and said point can be translated into a parameter space of $\frac{1}{r}$ and ϕ_0 (see fig. 4.4). For the radius the inverse is chosen in order to avoid high values for nearly straight lines.

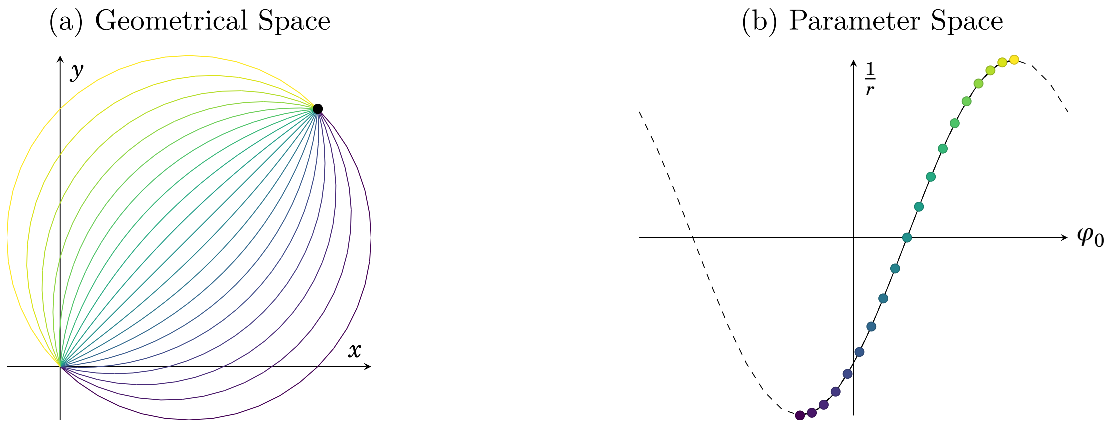


Figure 4.4: Example for the transformation of one point in geometrical space to a slope in parameter space. Every imaginary curve on the left corresponds to a point of the same color on the right [7].

After repeating this process for every track segment, the curves for all points on a particle trajectory will approximately intersect at the point where the parameters are equal to the real track parameters. Due to the inherent inaccuracies, those intersections will not be perfect, so both parameters are divided into grids. This is also necessary for making an implementation into hardware possible. Grid cells that contain many curves, i.e. maxima in the Hough space, are then passed on to the neural network as track candidates (see fig. 4.5). The included track segments provide the input parameters for the axial layers, the corresponding stereo track segments are estimated in an additional step. A track is only accepted if at least three of the four possible stereo track segments are found.

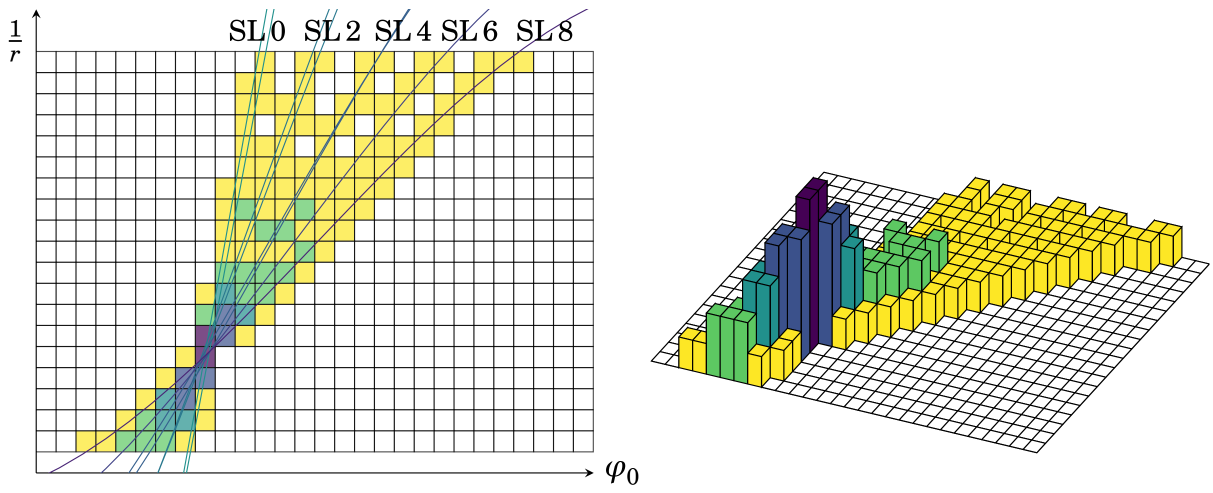


Figure 4.5: Binning and maximum search in the Hough space [7].

4.5 Neural Network

In the first two years of operation, the CDC trigger consisted only of the TSF and the 2DFinder. An event was triggered when the 2DFinder found two or more tracks in the $r\phi$ plane, so no z information was used [2]. However, this was not enough, as many background events produce tracks coming from the beam pipe. Accordingly, too much background was triggered, as shown in fig. 4.6.

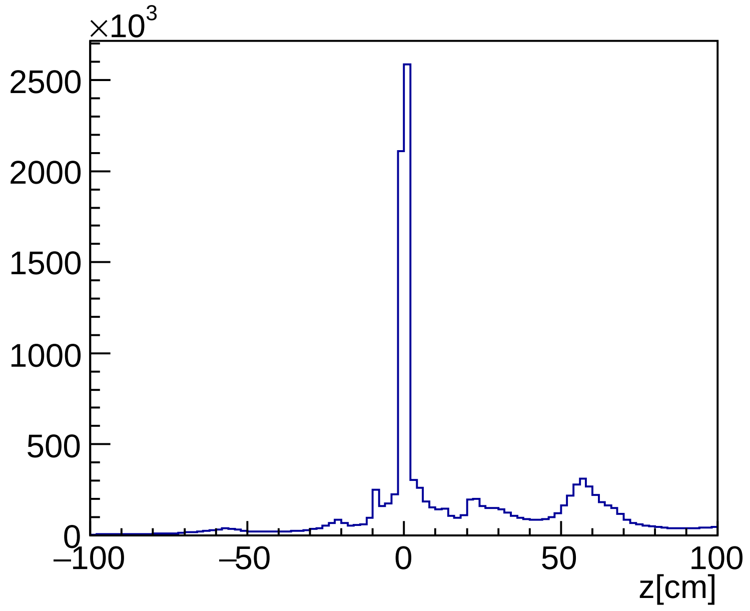


Figure 4.6: z-distribution of tracks found by the old 2D trigger in 2020[2].

A natural approach for solving this problem is to apply track fitting algorithms using information from the TSF and the 2DFinder in order to estimate the z position. However, those iterative algorithms do not have a fixed execution time, making them difficult to implement into hardware. When only the first order is used, the resolution is not sufficient for significantly reducing the background [2].

However, there is another method that works much better. A neural network is a suitable way for predicting the features of interest, like the z-vertex, for a track candidate. It can flexibly learn an analytical function and parallelize the calculations, making it ideal for an online trigger with strict time constraints.

For each of the nine superlayers, the network uses three input parameters, adding up to a total of 27 input nodes. The drift time, including the left/right information, can be taken directly from the TSF and scaled to values between -1 and 1. The remaining two parameters are the azimuthal position of the wire relative to the crossing point in the parameter space, ϕ_{rel}, and the crossing angle α. They have to be calculated using the results of the 2DFinder.

The crossing angle α is the angle between the track and the radial direction at the location of the priority wire, so it can be calculated using [3]

$$\alpha = \arcsin\left(\frac{r_{wire}}{2r}\right). \tag{4.1}$$

Here, r is the curve radius calculated by the 2DFinder, while r_{wire} is the radial distance of the priority wire from the vertex. As α is restricted to the range $\alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, it can also be scaled to values between -1 and 1.

The angle ϕ_{rel} is calculated by subtracting the intersection angle of the 2DFinder track with the priority wire layer from the azimuth angle of the priority wire. For stereo layers, ϕ_{rel} provides information on the z position of the hit [7]. Therefore, a combination of the three parameters allows for an accurate calculation of the z value. An overview of all input parameters can be seen in fig. 4.7.

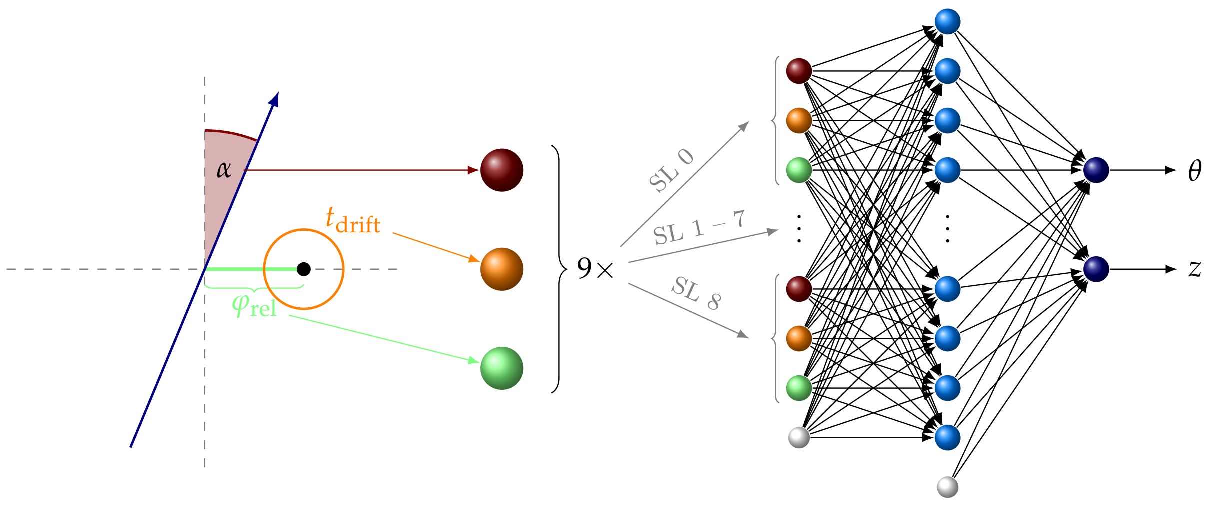


Figure 4.7: Illustration of the different input parameters and the neural network [7].

The network that was implemented at the start of this thesis has 27 input nodes, one hidden layer consisting of 81 nodes and two output nodes, which are used for predicting z and the negative cosine of the polar angle θ. For numerical reasons, z is scaled to a range of $[-1, 1]$, meaning that only values with $|z| \leq 100\,\text{cm}$ can be reached. Although bigger values are possible in principle, this is not an issue because they will be rejected anyway.

The reason for the network having only one relatively small hidden layer is the strong hardware constraint. Until now, the neural network calculation has to be done on a UT3 board in only 300 ns, so it is not possible to implement more than one hidden layer. The aim of this thesis will be to explore how new hardware can be used to improve the neural network performance. The network training was performed using PyTorch, embedded into the repository found in [23]. For all layers, the activation function $\tanh \frac{x}{2}$ was used, limiting each node to values between -1 and 1. A full overview of the used hyperparameters is shown in table 4.1. Other than the number of nodes, the hyperparameters were not changed throughout this thesis.

Hyperparameter	**Value**
Learning Rate	0.001
Batch Size	2048
Optimizer	Adam
Number of Input Nodes	27
Number of Hidden Layers	1
Nodes per Hidden Layer	81
Number of Output Nodes	2
Activation Function	$\tanh \frac{x}{2}$
Loss Function	MSELoss
Validation Split	0.2

Table 4.1: Hyperparameters used for the original training.

As mentioned in the previous section, a track that is passed on to the neural network has to include at least three of the four possible stereo track segments. Tracks without any missing stereo superlayers are typically more likely to come from the IP and give better network predictions because no stereo information is missing (see left side of fig. 4.8). Otherwise, the location of the missing superlayer is very important. When the outermost stereo track segment is missing, the z distribution does not really suffer, but with the innermost stereo superlayer missing, only about 53 % of the tracks come from the IP. This is mainly caused by many tracks having a displacement of more than 50 cm, so they only enter the detector after the first stereo superlayer (see right side of fig. 4.8).

One strategy for increasing the network accuracy on tracks with missing stereo superlayers is the use of so-called expert networks. Instead of only using a single neural network, five networks with the same architecture are used. During training, every expert network is only given tracks with the same configuration of missing stereo track segments. Expert 0 is only trained on tracks with the full stereo information. Experts 1, 2, 3 and 4 are used for tracks missing the first, second, third or fourth stereo superlayer, respectively. In hardware, all five networks are implemented, and each of them is used on the types of tracks it was

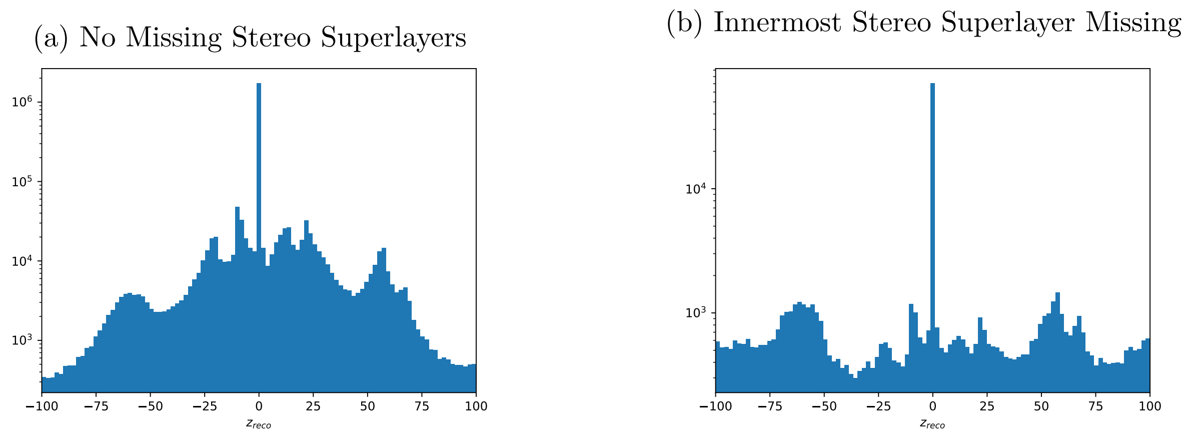


Figure 4.8: Logarithmic distribution of z values for tracks without missing stereo superlayers (expert 0) and for tracks missing the innermost stereo superlayer (expert 4). Expert 0 has over 70 % tracks within 1 cm of the IP, while expert 4 only has about 53 %.

trained on. For an analysis on the effectiveness of expert networks, see section 7.3.

The trigger was implemented in the beginning of the year 2021 and has been running successfully until the Long Shutdown in June 2022 (LS1). Using the condition that $|z_{neuro}| < 15\,\text{cm}$ and a momentum cut of $p > 0.7\,\text{GeV}$, a minimum bias Single Track Trigger (STT) was installed. This means that a single track in an event that fulfills the condition is enough to activate the trigger, resulting in a high sensitivity to low multiplicity events (see fig. 4.9). There is no prescaling necessary, as in case of too high trigger rates, the z-cut can be reduced to reach a higher signal purity.

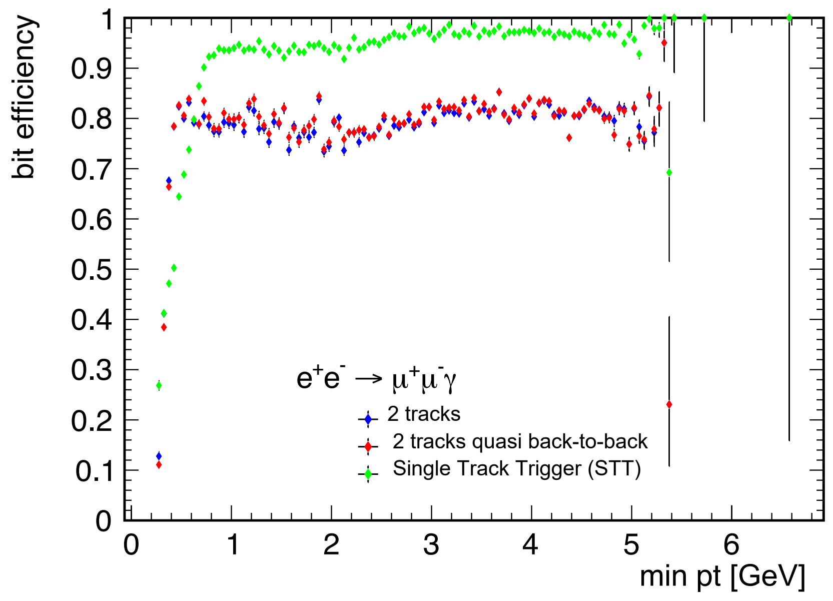


Figure 4.9: Efficiency of the STT as a function of the transverse momentum compared to two-track triggers [2].

In the past, where the maximum luminosity was $2.22 \times 10^{34}\,\text{cm}^{-2}\text{s}^{-1}$, the trigger worked very well. Going towards the design luminosity of $6 \times 10^{35}\,\text{cm}^{-2}\text{s}^{-1}$ however, there are still some problems with the trigger, that will be addressed in this thesis. While the efficiency is much better than the efficiency of two-track triggers (compare fig. 4.9), it can be significantly improved by bigger network architectures and better preprocessing. A more important concern is the background rejection. During Exp. 26, the data taking period right before LS1, which is where the input data for this thesis came from (compare chapter 5), there was a very high background in the CDC, leading to increased trigger rates. The STT used around 50 % of the total trigger budget instead of the usual 20 % [2]. One main reason for this is the so-called "Feed-Down" effect, depicted in fig. 4.10.

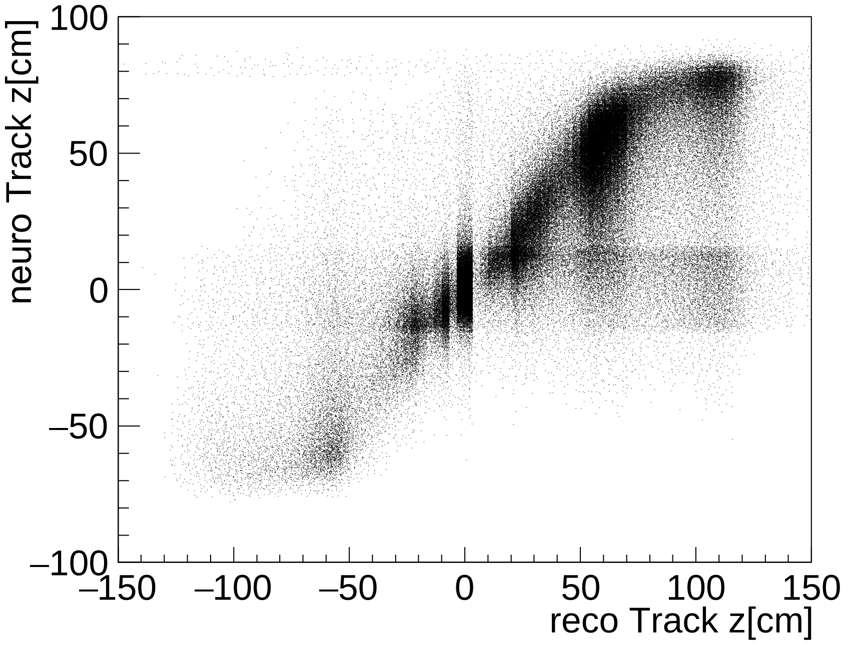


Figure 4.10: Correlation of the reconstructed z value and the one predicted by the network. Many of the tracks with high values of z_{reco} are predicted within the acceptance band of $|z| < 15\,\text{cm}$, adding to the total trigger rate [2].

The performance of the network implemented until LS1 will be analyzed in more detail in chapter 6.

Chapter 5

Training Data

The first hurdle before training a neural network is gathering a large amount of input data with a good balance between signal and background tracks. To ensure that the training data matches the conditions in the detector as accurately as possible, we decided to not use any Monte-Carlo simulations and stick to real data. The data used for all trainings throughout this thesis comes from the so-called neuroskim. This is a rule within the High Level Trigger (HLT) that every event triggered by the L1-Trigger is saved, as long as it contains the full trigger information. While in general data taking only one bunch cross is stored, 48 bunch crosses are stored for the neuroskim. Therefore, one event of the neuroskim contains multiple tracks of which at least one was classified by the current trigger to be signal. Naturally the dataset contains more signal tracks than background tracks. In Exp. 26, which was used for training and evaluation in the following chapters, around 70 % of the tracks come from the vertex, so there is a notable bias (see fig. 5.1).

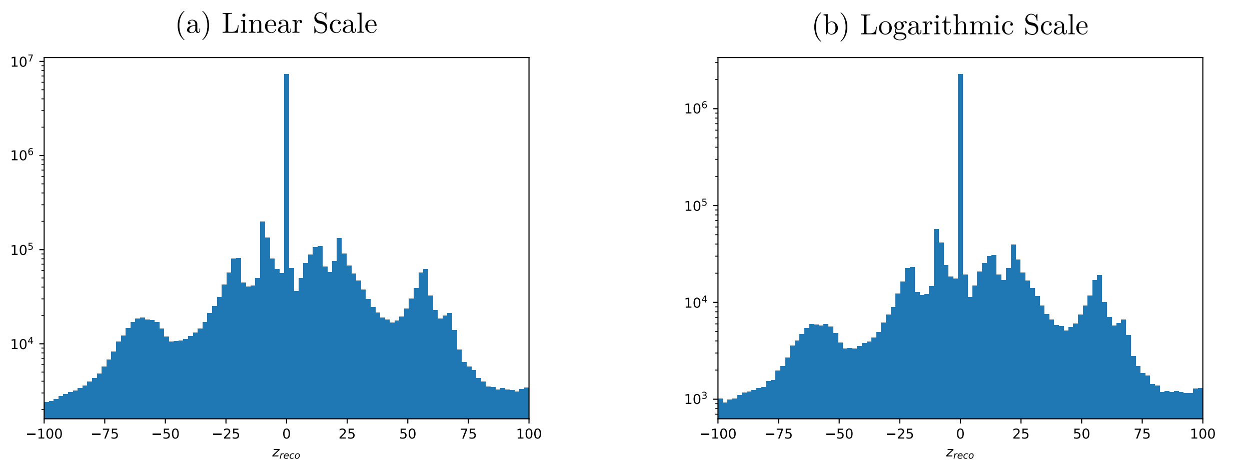


Figure 5.1: z Distribution of neuroskim tracks in Exp. 26. The parameters, including the z value, are taken from the reconstruction of the HLT. Those tracks are therefore called reco tracks, while the neural network predictions are called neuro tracks.

In the training data, there is no parameter that labels a track as signal or background.

Therefore, a sensible definition based on the z position is needed. As shown in fig. 5.2, the peak of the signal tracks ends at 0.25 cm, so it would be reasonable to use this as a separator between signal and background. Throughout this thesis however, signal tracks are defined at tracks with $|z_{reco}| < 1$ cm. While this may not be the optimal definition, it will not have too much impact, because less than 2 % of signal tracks have a distance of more than 0.25 cm from the vertex.

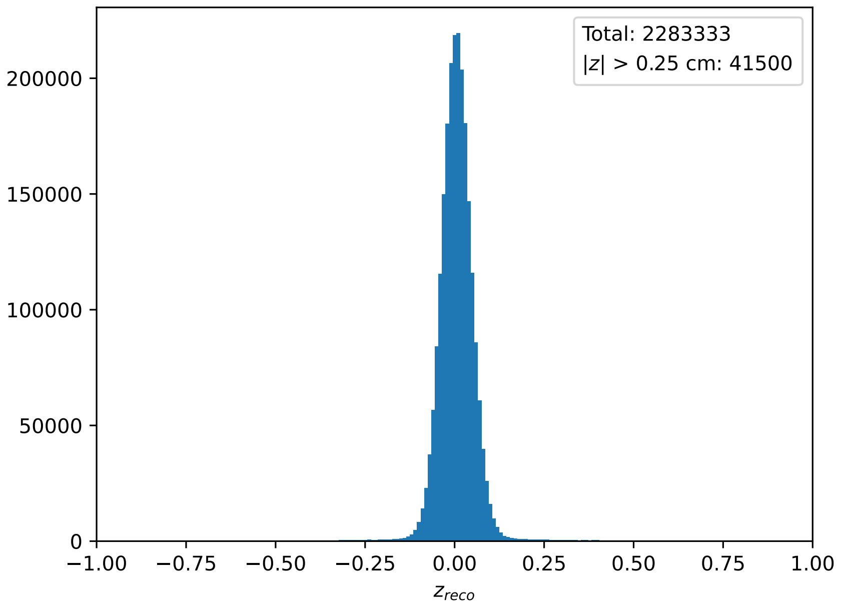


Figure 5.2: z Distribution of signal tracks.

When only displaying tracks with $|z_{reco}| > 1$ cm, cutting away the signal peak (fig. 5.3), one can take a closer look at the background distribution. Several peaks appear near the positions of bigger structures inside the detector, like the focussing magnets. As the red lines at the current position of the z-cut at ± 15 cm indicate, much of the background cannot be rejected with loose cuts, even with a perfect z prediction. Therefore, the aim is drastically improving the resolution for vertex tracks in order to be able to define very strict cuts and exclude many background tracks.

A distribution of $-\cos\theta$ can be seen in fig. 5.4. This is the same prescaling that is applied in the neural network input to keep the values between -1 and 1. Because of the acceptance range of the CDC, that only covers angles from 17° to 150°, there are relatively sharp cuts at shallow angles. It should also be noted that due to the asymmetry of the beam energies, there are more tracks with small angles than there are with large angles. A final observation to be made is that tracks with shallow polar angles make up a lot of the total data, so it is important to ensure a high efficiency on them.

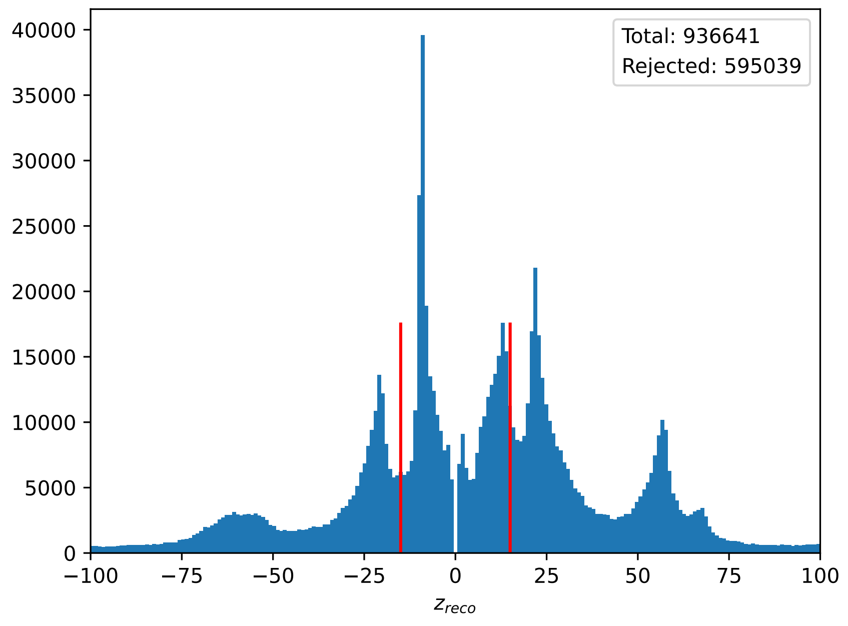


Figure 5.3: z Distribution of background tracks.

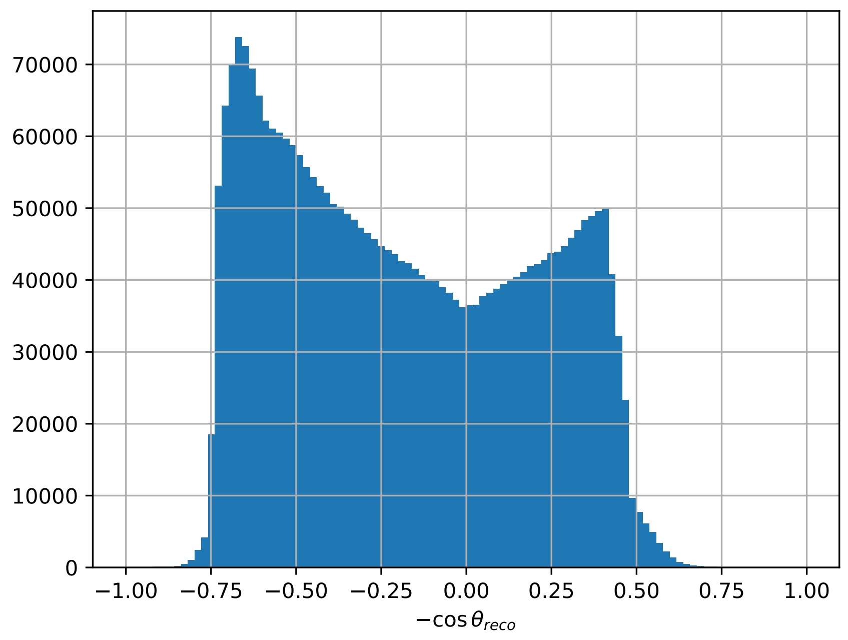


Figure 5.4: Distribution of the negative cosine of the polar angle.

Chapter 6

Evaluation of the Network Implemented Before LS1

In the following some of the different plots that were used for evaluating neural network performance throughout this thesis will be presented. They will be applied to the currently used network and some possible areas of improvement will be highlighted.

6.1 Double Gaussian

Until recently, the main metric of evaluation was the double Gaussian. The difference of the z-value predicted by the network, z_{neuro}, and the z-value calculated during the reconstruction, z_{reco}, is plotted in a histogram and fitted with the sum of two Gaussian functions. This function does not have a strict physical meaning, but it generally fits the data with reasonable accuracy.

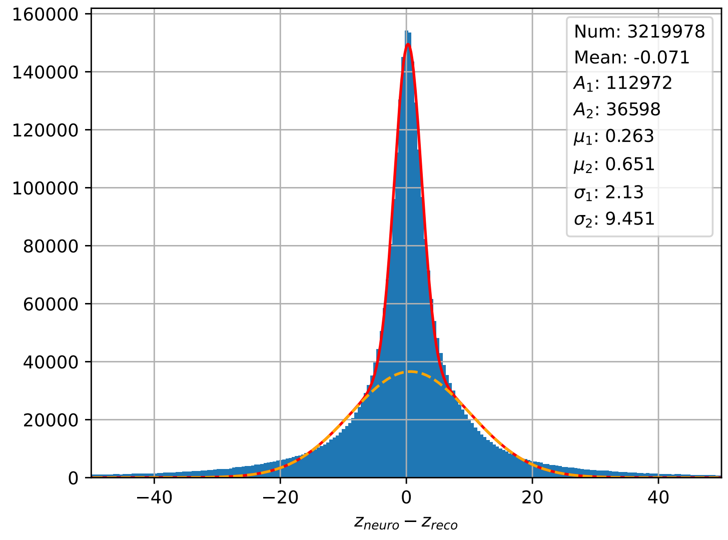


Figure 6.1: Histogram of $z_{neuro} - z_{reco}$ with double Gaussian fit.

The parameters most relevant for interpretation are the σ-values of both Gaussians. The core Gaussian indicates the peak accuracy of the network, but the wider one has more significance in determining the range of z-cuts that can be applied without losing too many signal tracks. Here the broader Gaussian is about 9.45 cm wide, so a z-cut of 15 cm is reasonable, although 6.25 % of signal tracks will be missed (see fig. 6.3).

6.2 z-Scatterplot

Another way of visually representing the output of a network is the z-scatterplot. Here, z_{neuro} is plotted over z_{reco}, so the goal is to be as close as possible to a diagonal line.

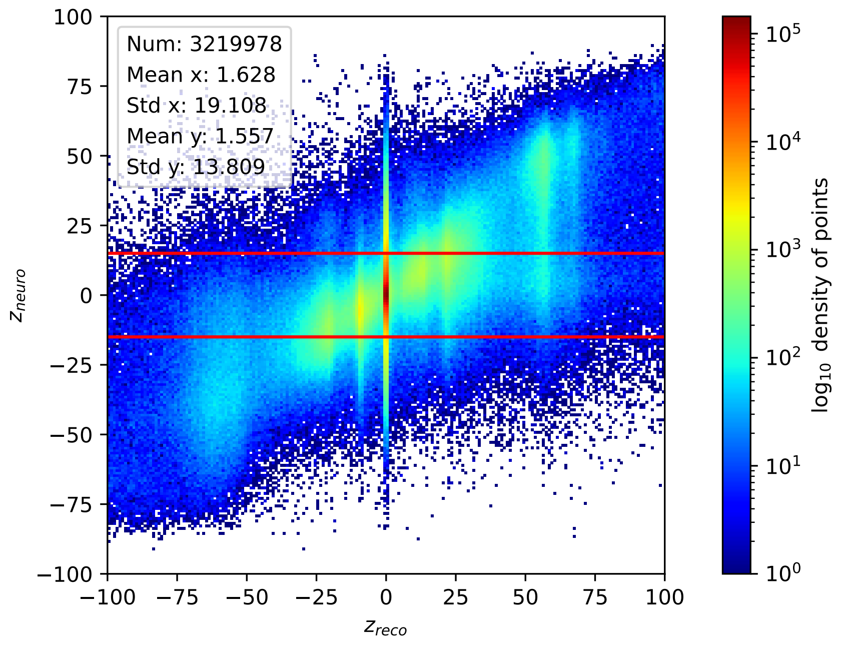


Figure 6.2: Correlation plot of z_{neuro} and z_{reco}.

The red lines at $z_{neuro} = \pm 15$ cm indicate the z-cut that is currently used for triggering, i.e. the acceptance interval. All tracks between the lines are accepted, all others rejected. Around $z_{reco} = 0$ cm, one can see that most but not all the signal tracks, defined as tracks with $|z_{reco}| < 1$ cm, are accepted by the trigger. The background tracks are usually reasonably close to the diagonal line, but many tracks, especially between 50 cm and 75 cm on both sides, are predicted to lie inside the acceptance interval. This is called the Feed-Down-Effect, of Feed-Up on the side of negative values of z_{reco}. It is mainly a result of the strong bias towards the vertex in the input data, as discussed later in this chapter.

6.3 Efficiency and Rejection Rate

In order to be able to precisely quantify the performance of a network, two key values were used. The efficiency is defined by the number of signal tracks that are accepted by a given z-cut divided by the total number of signal tracks. The rejection can be calculated as the number of background tracks rejected by the z-cut divided by the total amount of background tracks. A visualization of this can be seen in fig. 6.3.

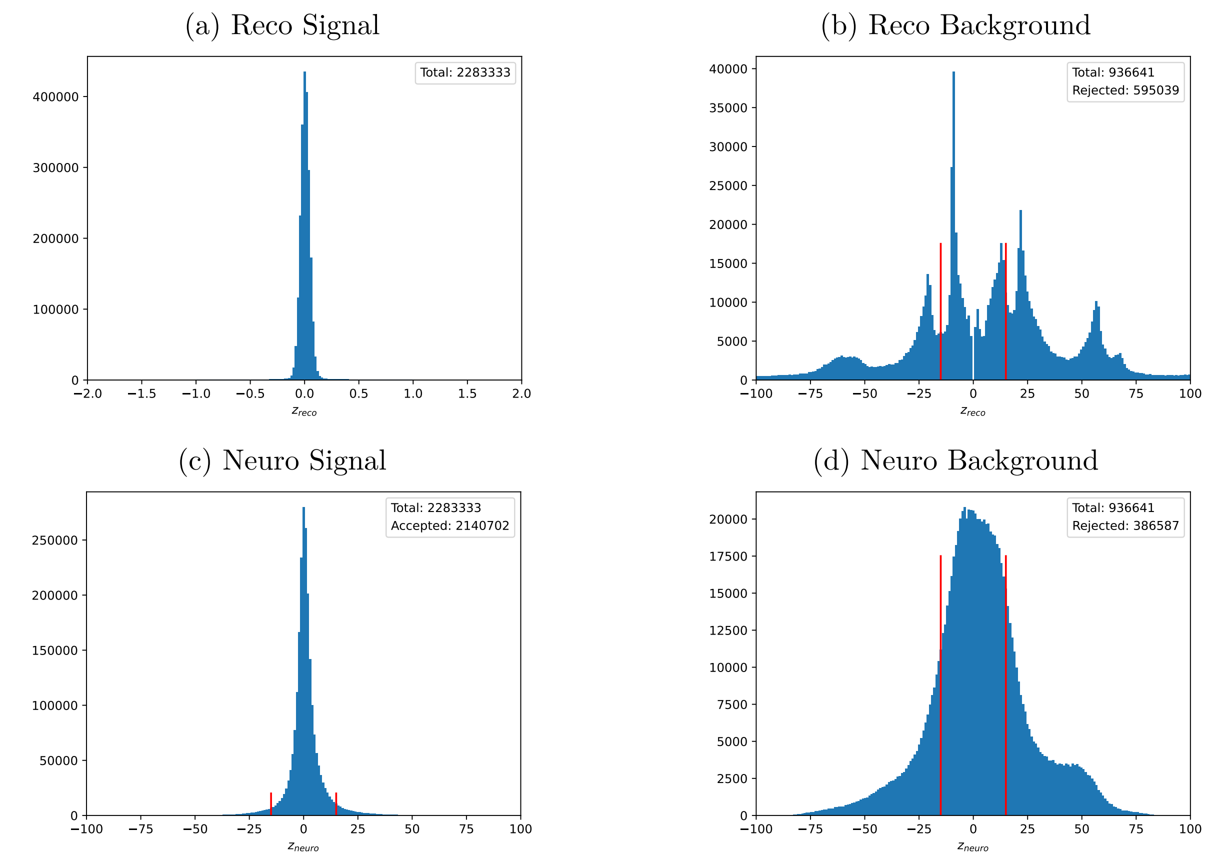


Figure 6.3: Visualization of the calculations for efficiency and rejection for a z-cut at ± 15 cm.

As indicated before, the network does predict some background structures accurately (see fig. 5.3), but more than half of the background is predicted to be inside the acceptance interval. With a z-cut at ± 15 cm, a rejection of around 41.3 % can be achieved. Most of the signal tracks are predicted to be relatively close to the vertex, but the tails are unfortunately fairly stretched out, so that the efficiency for the same z-cut is only at 93.75 %. Combining those two calculations leads to the red dot in fig. 6.4, and the whole curve emerges when repeating the same procedure for different z-cuts. This will be the key method of evaluation in the scope of this thesis.

One can also plot efficiency and rejection as a function of the z-cut, as seen in fig. 6.5. Evidently there is a clear, almost linear correlation between the z-cut and the rejection rate, so it is necessary to tighten the z-cut. However, as the signal prediction in fig. 6.3 already suggests, the efficiency drops rapidly when the z-cut is decreased, prohibiting stricter z-cuts.

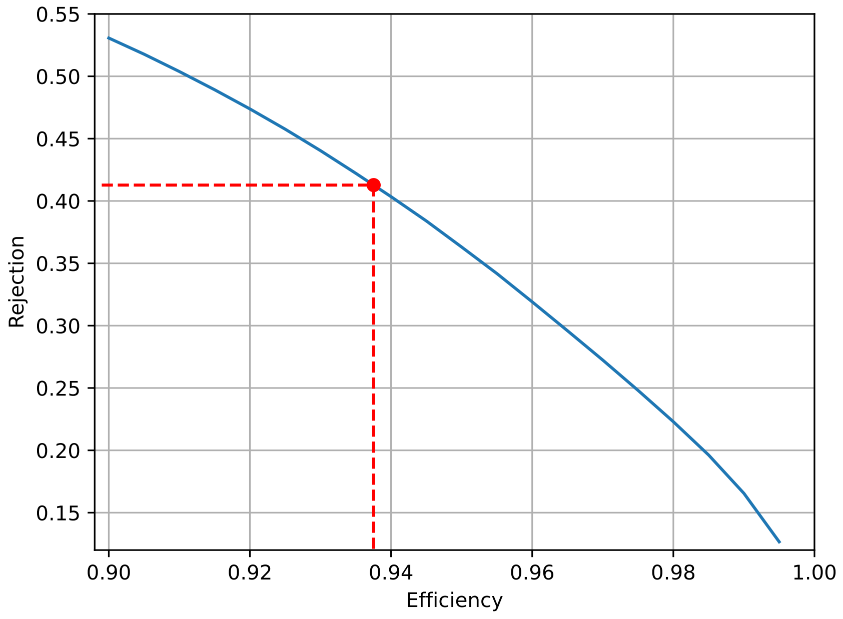


Figure 6.4: Rejection over efficiency (ROE-curve). The red dot indicates efficiency and rejection at a z-cut of ± 15 cm (see text).

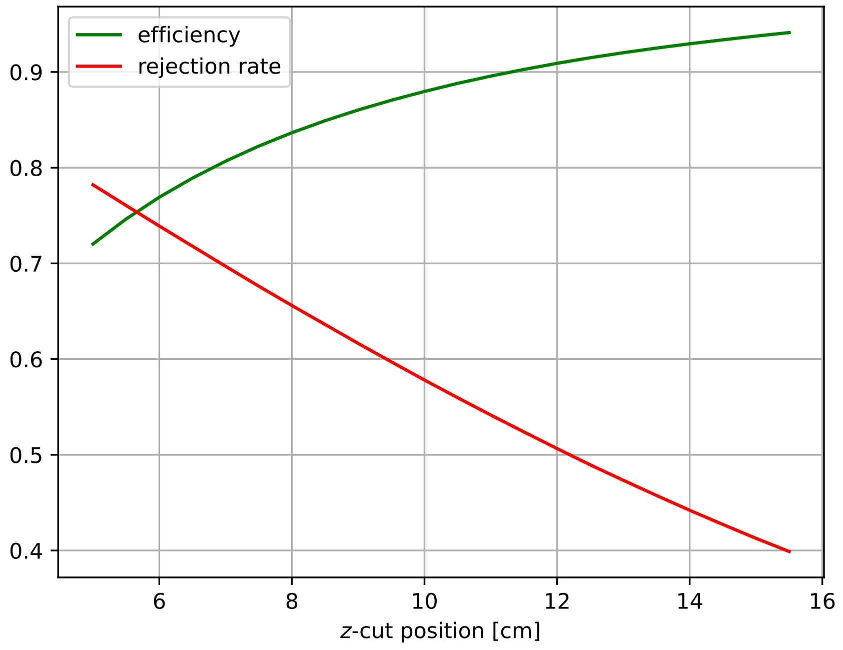


Figure 6.5: Efficiency and rejection for different z-cuts.

Chapter 7

Retraining and Deep Neural Networks

7.1 Retraining

As indicated in chapter 2, a key motivation for the improvement of the Neural Track Trigger is the strong increase in background and noise during Exp. 26. However, this problem is also an opportunity, as training on the noisy data leads to an overall improvement in network performance.

This retrained network is already implemented and working. It will therefore be called “NN24”, while the network from before LS1 will be called “NN22”. A detailed comparison of both network is shown below.

Double Gaussian

While the double Gaussian of the newly trained network looks relatively similar to the old one, a few things stand out: The new graph has a higher peak, meaning that more tracks have a very high accuracy. In addition, the function is a little slimmer, which is reflected in both σ values.

Efficiency and Rejection

When comparing rejection and efficiency between the two networks (fig. 7.2), two direct results of the retraining can be seen. The efficiency is higher for all z-cuts with NN24, because it is used to having to recognize signal tracks with high levels of background. For example, with a z-cut at $\pm 15\,\text{cm}$, NN22 has an efficiency of $93.75\,\%$, while NN24 finds $95.73\,\%$ of signal tracks. Nevertheless, the rejection is slightly worse now, with a decrease from $41.27\,\%$ to $39.53\,\%$. This is a consequence of the used training data. While old experiments contained relatively balanced amounts of signal and background tracks, around $70\,\%$ of the tracks come from the vertex in Exp. 26, which strengthens the bias of the network towards the IP (see fig. 7.3). The reason for the changed ratio is that during

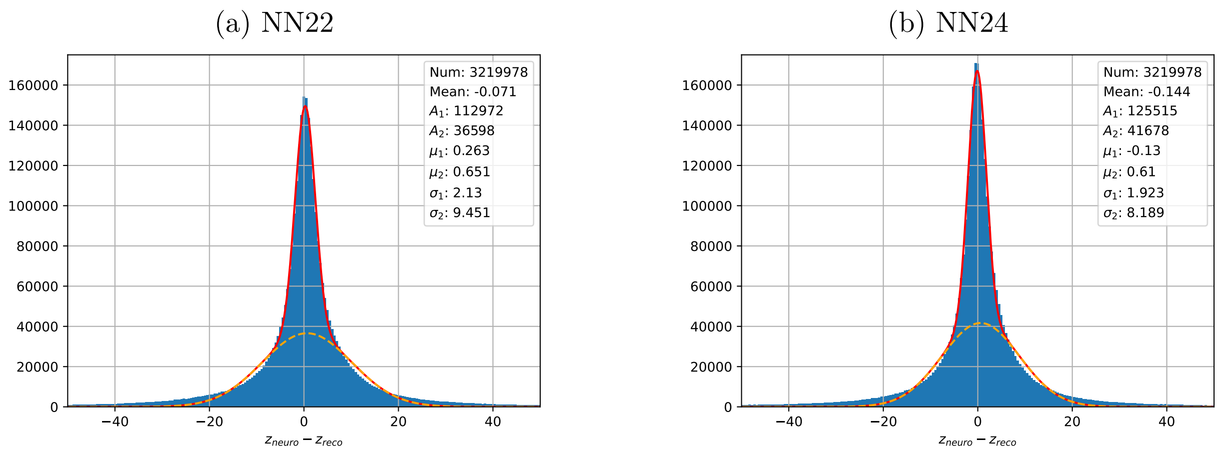


Figure 7.1: Comparison of the double Gaussian to the old network.

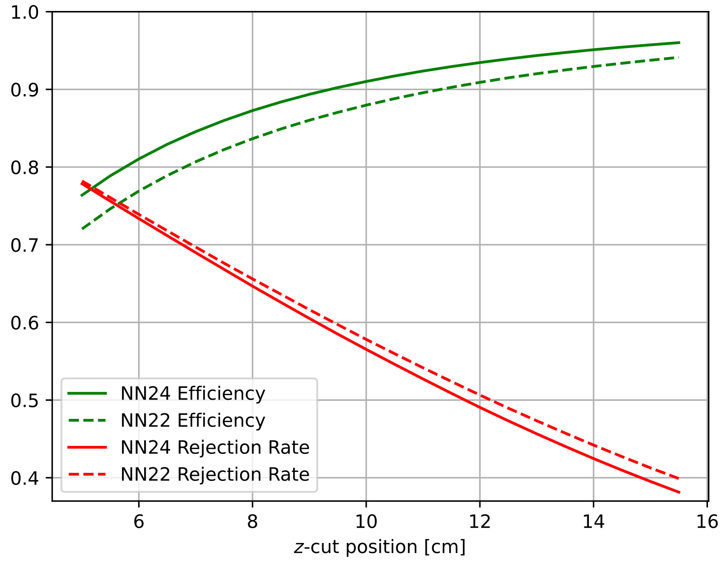
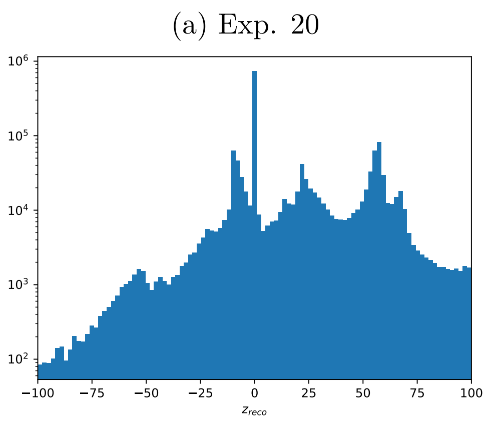


Figure 7.2: Comparison of efficiency and rejection rate for different z-cuts.

Exp. 26 the very successful current neurotrigger was already installed, meaning that more background tracks were rejected than in Exp. 20.



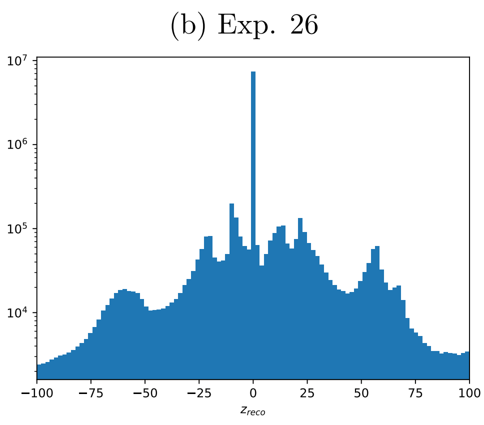


Figure 7.3: In Exp. 20, less than half of the tracks originated from the vertex. This number increased to about 70 % in Exp. 26.

The curve in fig. 7.4 shows an observation that will be seen many times during this thesis: For the final performance, the efficiency is much more important than the rejection rate. It is nice to have a good rejection rate even for loose z-cuts, but the biggest impact on rejection is the z-cut, which can only be tightened with sufficient efficiency. Therefore, the performance of the new network is better, although the comparison of efficiency and rejection for different z-cuts gave mixed results. For an efficiency of 95 %, NN22 has a rejection rate of 36.31 %, while NN24 manages to reject 42.95 % of background tracks.

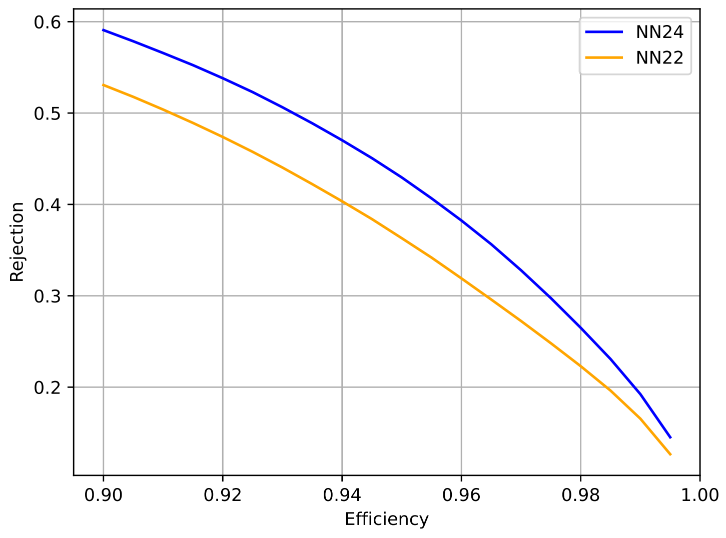


Figure 7.4: Performance comparison of the old and the retrained network.

7.2 Deep Neural Networks

As indicated in chapter 4, NN22 was kept rather simple with only one hidden layer due to hardware constraints. During Long Shutdown 1, some of those constraints were lifted, allowing for more complex networks. The UT3 boards were replaced with faster and more powerful UT4 boards, meaning that there are now more resources for the network. In addition, the track finding can be done on the same board, increasing the latency for neural computations. Calculations in a deep neural network can only be done sequentially, one layer at a time, which is reflected in the necessary calculation time. Before LS1, this restricted the network to one hidden layer. With the added latency, deep neural networks with up to four hidden layers now seem possible.

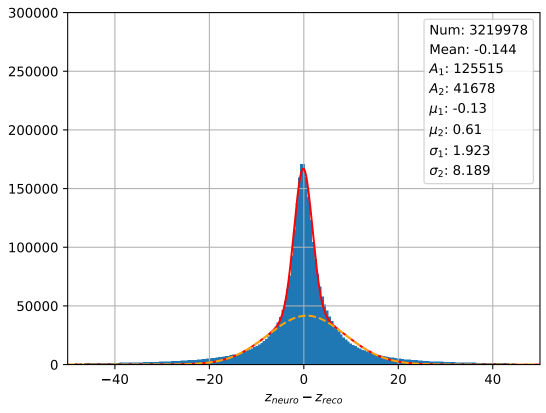
Overview of Tested Network Architectures

Many different architectures have been tested in the scope of this thesis. According to some interpretations of the Kolmogorov-Arnold representation theorem, every function can be approximated to an arbitrary degree of accuracy by a neural network with one sufficiently large hidden layer [24]. Thus, a neural network with one big hidden layer consisting of 800 nodes, i.e. an order of magnitude more than the currently installed networks, was tested. Although the number of trainable parameters is quite similar to that of a deeper and slimmer neural network, the performance is relatively disappointing (compare fig. 7.6). The most likely explanation is the following: In the hardware, but also during training, the floating-point accuracy is finite, so relevant information may be lost when adding up 800 values. Even on a smaller scale and with more hidden layers, there is a general tendency that increasing the number of hidden nodes per layer beyond a certain point does not lead to much better results. The number of hidden layers has a more favorable impact on the results: With every additional layer, the performance is improved, though the effect is strongest for two hidden layers and weakens when adding more.

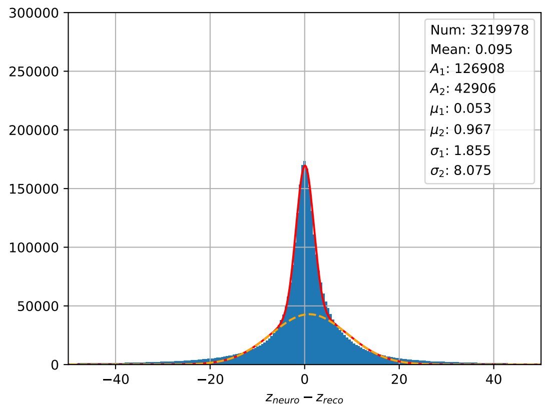
Double Gaussian

In fig. 7.5, the z-resolutions of different network architectures were compared. As in the previous sections, this is done by plotting a histogram of Δz and fitting it with a double Gaussian. One can then compare the widths of the core Gaussians (σ_1) and the wide Gaussians (σ_2). To ensure that only the network architectures are considered, from now on the newly trained network from section 7.1 will be used for comparison. While the double Gaussian remained largely unaffected by mere retraining without a change in architecture, the introduction of more hidden layers already makes the function slimmer by almost a factor of two (see table 7.1). The peaks are also much higher and the tails end earlier, promising a higher efficiency. As previously discussed, the network with one large (800 nodes) hidden layer does not perform much better than the current architecture, the double Gaussian is only marginally slimmer. However, even the seemingly small step of going to two hidden layers brings a noticeable improvement, with another small gain from the third layer. Expanding the three hidden layer network to 300 nodes per hidden layer has a very small positive impact at best, so for now the focus will be on fewer nodes per hidden layer. Similar tendencies can be observed in the z-scatterplot. It should be noted that the network with four hidden layers was trained without experts, explaining that the function is worse than the one for three hidden layers. This decision will be justified in section 7.3

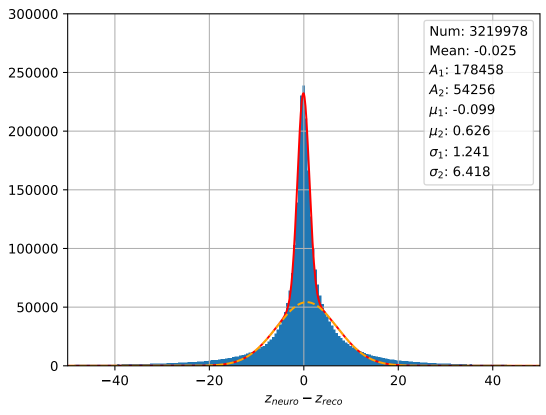
(a) 81 Hidden Nodes in One Layer



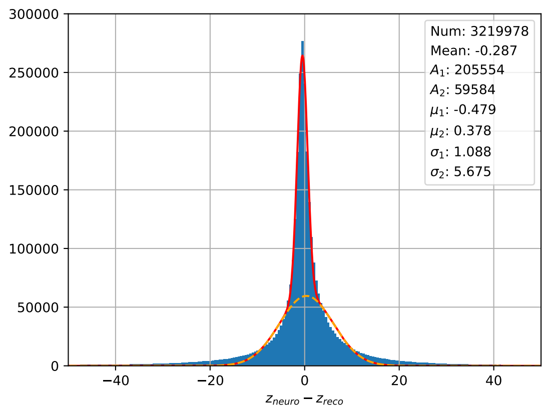
(b) 800 Hidden Nodes in One Layer



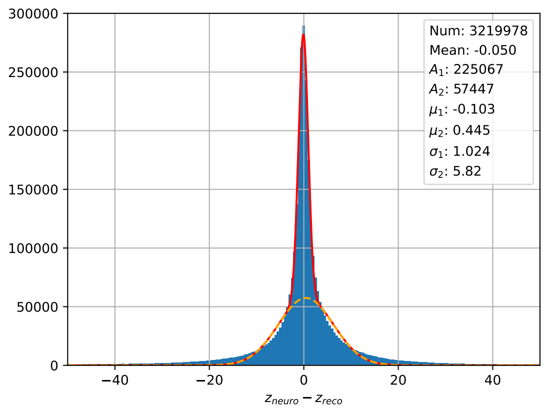
(c) Two Hidden Layers with 100 Nodes Each



(d) Three Hidden Layers with 100 Nodes Each



(e) Three Hidden Layers with 300 Nodes Each



(f) Four Hidden Layers with 100 Nodes Each

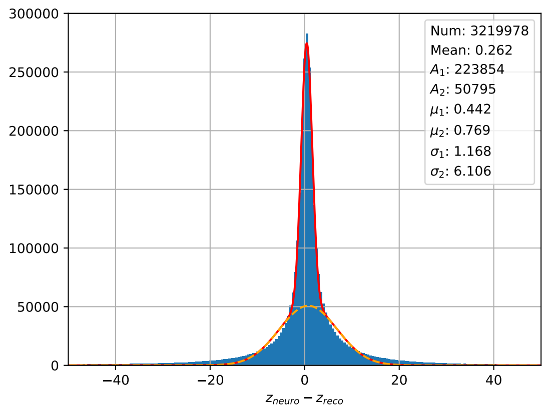


Figure 7.5: Comparison of the double Gaussian with different network architectures.

Network	σ_1 [cm]	σ_2 [cm]
NN24	1.92	8.19
1x800	1.86	8.08
2x100	1.24	6.42
3x100	1.09	5.68
3x300	1.02	5.82
4x100	1.17	6.11

Table 7.1: Parameters of the double Gaussians of different network architectures.

Efficiency and Rejection

The ROE curve confirms the general trends seen in the double Gaussian. The number of hidden nodes per layer hardly affects the performance, but adding hidden layers does. Contrary to the results of the double Gaussian however, the clear favorite according to the ROE curve is the network with four hidden layers. This is also underlined by a comparison of rejection rates at an efficiency of 95 %, seen in table 7.2. Compared to NN24, the network with four hidden layers and 100 nodes per hidden layer has a higher rejection rate by more than 10 percentage point, going from 42.95 % to 54.59 %.

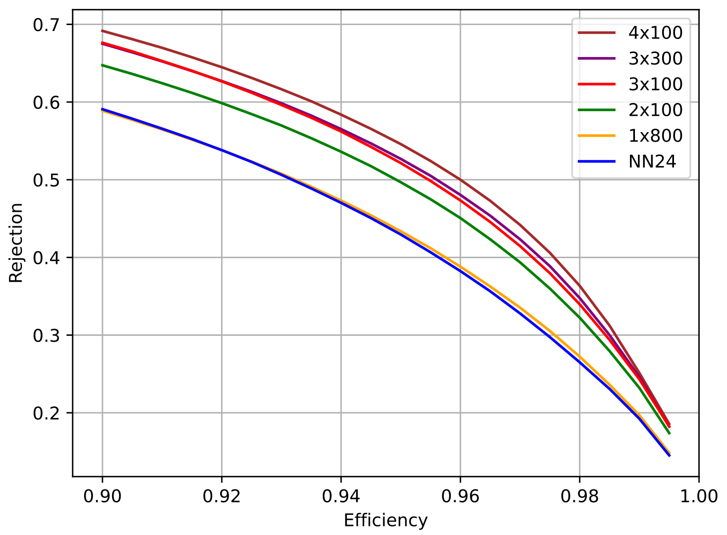


Figure 7.6: ROE Curve for different network architectures.

Once again the performance evaluation for different z-cuts can be used for a better understanding of the strengths and weaknesses of the networks (see fig. 7.7). Both efficiency and rejection rate improved when going from one to four hidden layers. When compared to three hidden layers, the four hidden layer network also excels in both aspects. For strict z-cuts under 8cm there seems to be a tendency towards weaker networks having a higher rejection rate, indicating a stronger confinement of the Feed-Down to the vertex with better networks.

Network	Rejection at 95 % Efficiency
NN24	42.95 %
1x800	43.37 %
2x100	49.68 %
3x100	52.15 %
3x300	52.68 %
4x100	54.59 %

Table 7.2: Rejection rates of different networks for an efficiency of 95 %.

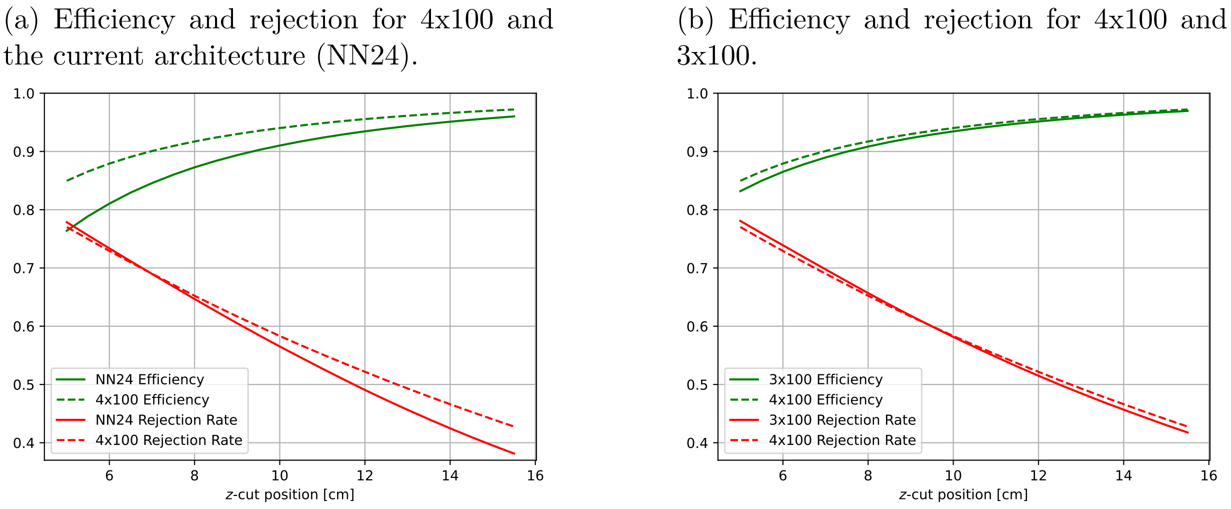


Figure 7.7: The deep neural networks beat the current architecture in both efficiency and rejection. The fourth hidden layer adds some efficiency, but the rejection rate is slightly worse.

7.3 Experts

As discussed in chapter 4, the current network uses experts for different configurations of missing stereo layers for optimizing the accuracy. Indeed, the double Gaussian of NN24 is much wider when no experts are used: The core Gaussian goes from 1.92 cm to 2.37 cm, the wide Gaussian from 8.19 cm to 8.98 cm (see fig. 7.8).

Experts are a good way to compensate missing information from the superlayers, but now there are more complex structures, so a single expert is enough (see fig. 7.9). The slight loss in accuracy and therefore efficiency is offset by the gain in rejection rate, leading to a nearly equal performance with and without experts. Later studies have shown that networks without experts even perform better than those with experts when going to four hidden layers and using the new preprocessing discussed in chapter 8. Reasons may include the smaller sets of training data for experts (see table 7.3) and the resulting decrease in robustness to rare background tracks.

For the above reasons, all networks in the following chapters will have a single expert, unless stated otherwise.

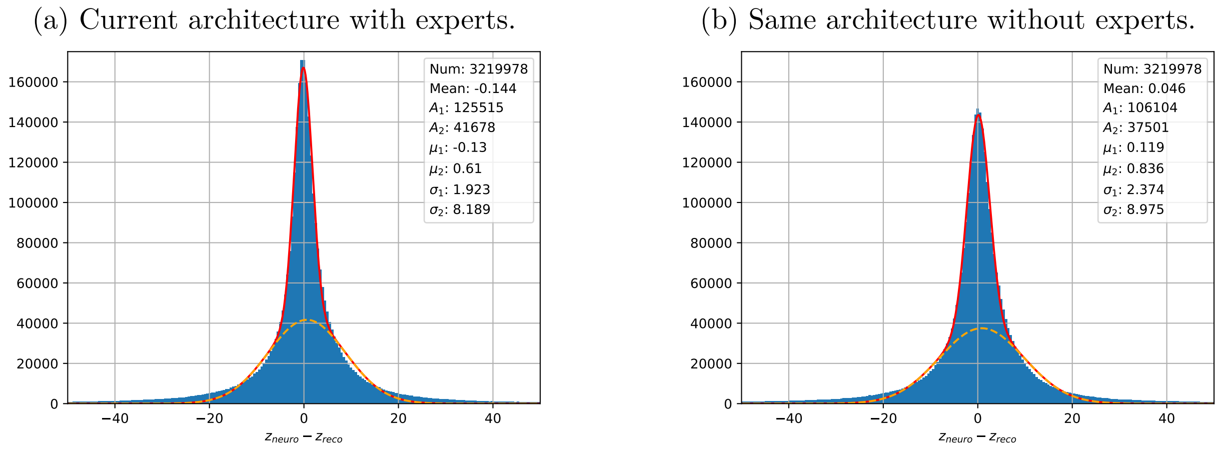


Figure 7.8: Double Gaussian with and without experts.

Expert	**Size of Data Set**
Expert 0	2443058
Expert 1	300174
Expert 2	101815
Expert 3	241796
Expert 4	133135
Total	3219978

Table 7.3: Size of the training data set for every expert.

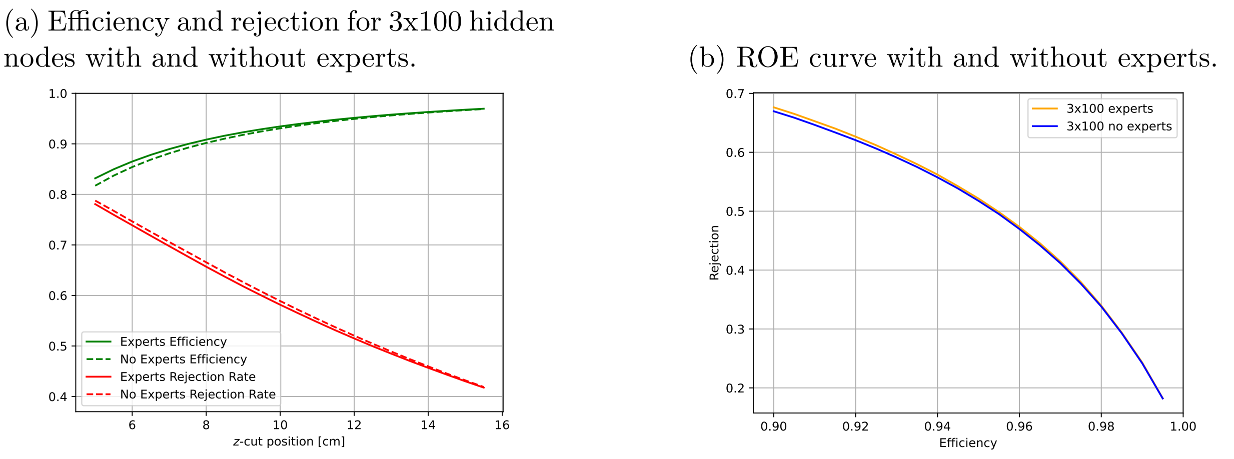


Figure 7.9: In the relevant region around 15 cm, the experts only lead to a minimal improvement.

7.4 Quantization and Quantization Aware Training

As mentioned earlier, the biggest requirement for the network is the ability to implement it on the available hardware. This includes not only the number and size of the hidden layers, but also the bitwidth of the weights and biases. There are currently two steps for limiting the network bitwidth. The first step is to restrict the weights to values between -64 and 64. Next they are rounded to multiples of $\frac{1}{2048}$, so that in total every weight takes up 18 bits. This quantization does not affect the network performance in any significant way, as table 7.4 shows.

Quantization	**Efficiency**	**Rejection**
Full Bitwidth, No QAT	93.45 %	58.16 %
18 bits, No QAT	93.44 %	58.19 %
12 bits, No QAT	93.15 %	58.57 %
Full Bitwidth, QAT	93.34 %	59.65 %
18 bits, QAT	93.34 %	59.69 %
12 bits, QAT	93.20 %	59.80 %

Table 7.4: Efficiency and rejection for a z-cut at 10 cm for different quantizations and with and without QAT. The evaluation was done on a network with 3x100 hidden nodes and experts, because at the time this was considered the most promising network.

A much stricter quantization was also tested, where the weights are reduced to 12 bits. Here, the weights are first restricted to values between -8 and 8, which does not make a difference, as all weights are already in this interval. Then they are rounded to multiples of $\frac{1}{64}$, so the resolution is decreased by a factor of 32. This does have an effect, but the impact is still rather small (see table 7.4).

Regardless, it was studied whether the Quantization Aware Training (QAT) of PyTorch could improve the performance, in particular the performance after quantization. Using this module, the network is optimized to being implemented on hardware with a quantization to 8 bits. While this is not the type of quantization that will be used on the FPGAs, it was still tested in the hope that it would make the network more robust to limited bitwidths. The effect on the efficiency does not really surpass the statistic uncertainty, but surprisingly the rejection rate increases by more than one percentage point for all quantizations. The reasons for this are not entirely clear, but as it helps the overall performance, from now on all trainings will be done using QAT.

Chapter 8

Improved Network Input

In the previous chapter it was shown that the quality of the neural network predictions can be greatly improved with better training data and more hidden layers. However, the output will always be bounded by what information the network receives. Different approaches to passing better input data to the network will be discussed in the following chapter.

8.1 Extended Input

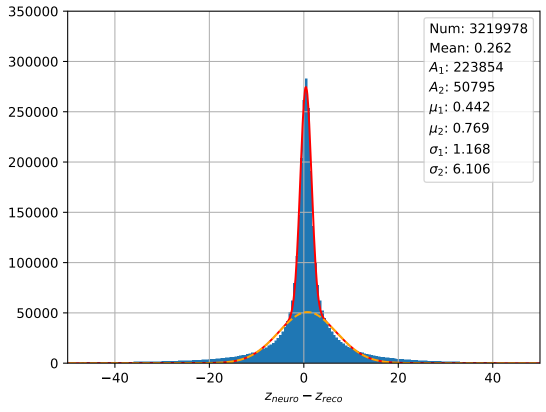
The most obvious change in trigger hardware after LS1 was the introduction of more powerful FPGA boards, but there were also more subtle but important improvements. One of them is the new ability to feed the drift times of all wires of a track segment into the network, which was first investigated in [25]. The time resolution for those wires is 32 ns, compared to 2 ns for the priority wires. In every track segment there are eleven drift times that are passed to the network in addition to the current three input parameters per superlayer, so that there are now 126 instead of 27 input nodes. As there is now no left/right information available, the drift time is always taken as positive and scaled to a value between 0 and 1. If a wire is not hit, the value defaults to -1. The arrangement of the wires in a track segment is always consistent, allowing spacial information about the particle's path to be extracted.

Double Gaussian and z-Scatterplot

As expected, the network gets a big boost in accuracy from the extended input, nicely visible in the double Gaussian and the z-scatterplot. Both sigma values of the double Gaussian are almost cut in half (see fig. 8.1).

The z-scatterplot undergoes equally significant changes (see fig. 8.2) There are much more points near the desired diagonal line and the red line at the vertex is reduced to merely a point. At first glance it may seem like the Feed-Down has weakened, but in fact it is now merely focused to values of $|z_{neuro}| < 5\,\text{cm}$. While now only about 49 % of background tracks have $|z_{neuro}| < 15\,\text{cm}$, compared to 56 % without extended input, the proportion of tracks with $|z_{neuro}| < 5\,\text{cm}$ increased from 23 % to more than 25 %.

(a) Double Gaussian with Standard Input



(b) Double Gaussian with Extended Input

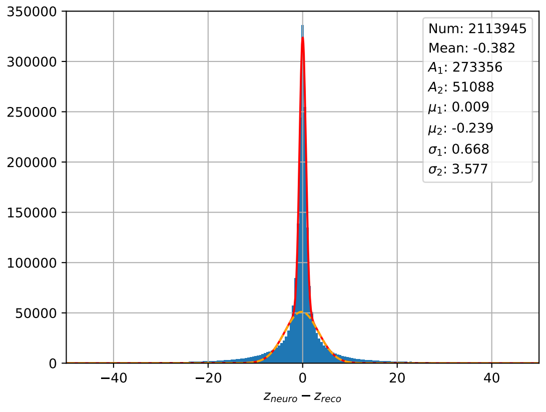
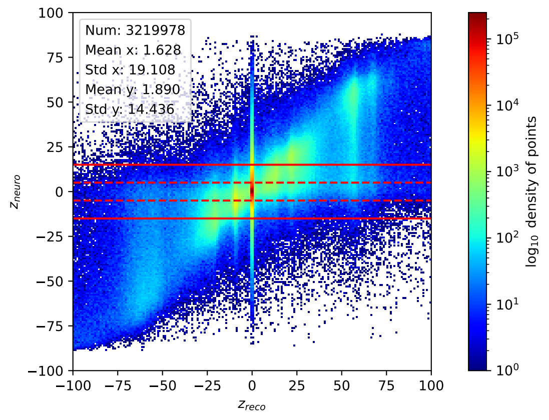


Figure 8.1: Comparison of the double Gaussians of networks with 4x100 hidden nodes with standard and extended input. The differences in the scaling of the y-axis are caused by the different amounts of tracks.

(a) z-Scatterplot with Standard Input



(b) z-Scatterplot with Extended input

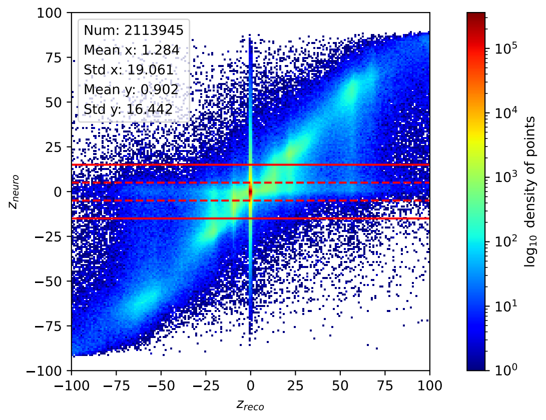
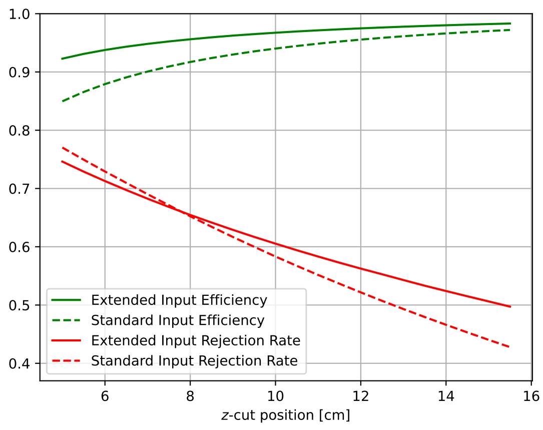


Figure 8.2: Comparison of the z-scatterplots of networks with 4x100 hidden nodes with standard and extended input. The solid lines indicate a z-cut at ± 15 cm, the dashed lines a z-cut at ± 5 cm.

Efficiency and Rejection

The impact of the z-scatterplot can be clearly seen in the efficiency and rejection (see fig. 8.3a). Smaller deviations in the prediction of signal tracks cause a much better efficiency that does not drop as quickly when lowering the z-cut. For loose z-cuts, the rejection also improves, but the strong focus of the Feed-Down onto the vertex causes smaller gains for stricter z-cuts. Nevertheless, the overall performance is very satisfactory, surpassing the effect of deep learning (see fig. 8.3b). With an efficiency of 95 %, the rejection rate goes from 54.59 % to 67.65 %.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

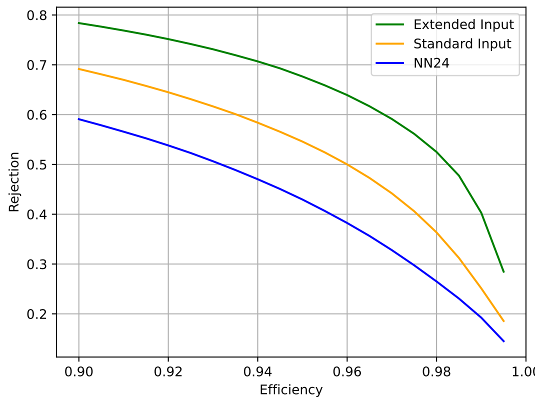


Figure 8.3: Efficiency and rejection for standard and extended input (4x100 hidden nodes).

8.2 ADC-Cut

In order to reduce noise and thereby suppress fake and background tracks, an ADC-cut can be used. The ADC-count is a measure of the charge deposition on a wire. It is generally a good indicator of how reliable a hit is. Electronic cross-talk from nearby wires or synchrotron photons, i.e., sources of background and noise, are the primary causes of low counts. Ideally, the network would be directly provided with the ADC-count for every wire in a track segment, but this is unfortunately not yet possible. However, an ADC-cut can be installed before the TSF, so that every wire with an ADC-count below a certain threshold is treated like a missed wire.

Figure 8.4 shows a distribution of ADC-counts for signal and background events. Values over 300 are cut off, because in some events the ADC-count can reach very high values. In the signal data, there is a clear peak around 50, which is caused by minimum ionizing particles. The background strongly increases for values under 10 because of the earlier discussed processes. Although the analyzed run contains around four times as much signal as background, with an ADC-cut at 10 twice as many background wires as signal wires are disregarded. In addition, signal tracks may also contain some electronic cross talk, so it can be expected that by applying the cut hardly any relevant information is lost. Therefore, an ADC-cut at 10 will be used for the following studies.

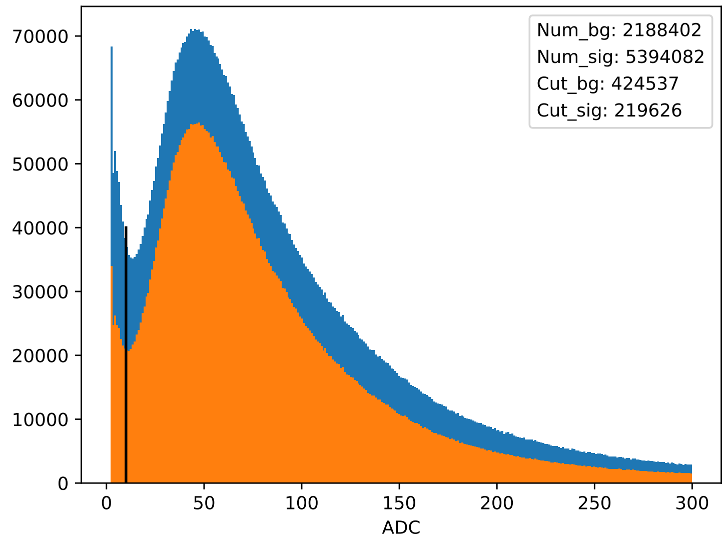


Figure 8.4: Histogram of ADC-counts for signal (orange) and background (blue). The black line indicates a cut at an ADC-count at 10.

Efficiency and Rejection

Figure 8.5 shows the effects of the ADC-cut at 10 on efficiency and rejection.[1] The most notable change can naturally be seen in the rejection rate, which improves by about five percentage points. There might even be some background tracks that are no longer found by the 2DFinder, which would further increase the effective rejection rate if they made it into the statistics. As previously noted, the efficiency also benefits from the cut, because there is less noise distracting the network. Overall the ADC-cut results in a significant performance boost, with the rejection rate at 95 % efficiency increasing from 67.18 % to 79.38 %.

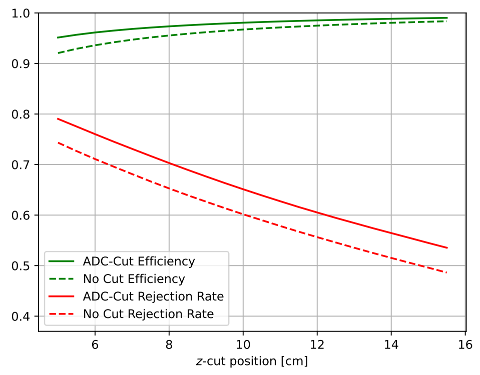
8.3 Deployment of a 3D Track Model

A further approach to improving the input of the neural network is to replace the 2DFinder with a 3DFinder. This was first proposed in [3] and optimized in [4]. The basic idea is to add the θ angle as a third dimension to the parameter space and use both axial and stereo layers. An example of the resulting Hough space for a simulated single muon can be seen in fig. 8.6

Using the 3DFinder has a number of advantages. While the 2DFinder only requires a track to originate from the transverse vertex region $(x, y) = (0, 0)$, without any knowledge of the z component, the 3DFinder works with a vertex assumption of $(x, y, z) = (0, 0, 0)$. With this track model, it is expected that the number of found tracks strongly decreases with larger values of $|z|$ (fig. 8.7).

[1]The neural networks now only have 80 instead of 100 nodes per hidden layer. This will be justified in section 9.1.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

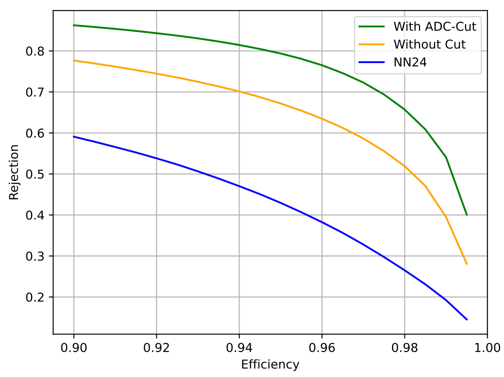


Figure 8.5: Efficiency and rejection with and without ADC-cut at 10 on identical networks with 4x80 hidden nodes.

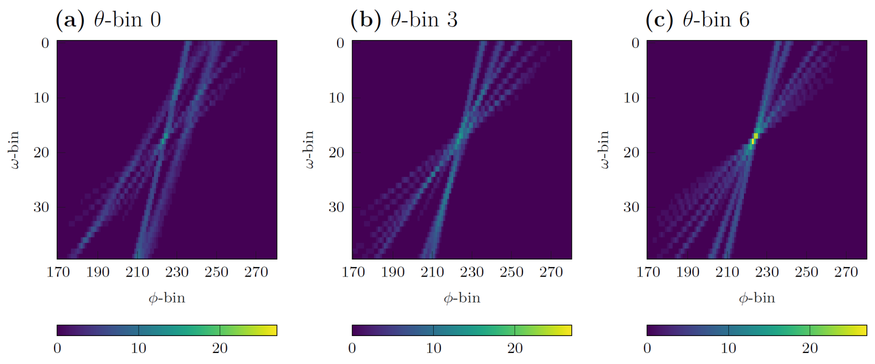


Figure 8.6: Heatmaps of the Hough space for different θ-bins of a single muon particle gun track [4].

Because it requires more information to find a track, the 3DFinder is also less susceptible to finding fake tracks, i.e. tracks that only consist of noise and do not have a corresponding reco track. The choice of Stereo Layers is also superior to the old mechanism, as they are now directly integrated into the track finding.

Efficiency and Rejection

In the previous sections, performance was estimated using all tracks found by the 2DFinder. Now only tracks found by the 3DFinder can be evaluated, meaning that some of the most important effects of the new preprocessing are not represented. In order to accurately evaluate the combined performance of the 3DFinder and the neural network, adding tracks not found by the Hough finder to the amount of rejected tracks, unbiased input data would be necessary. There is still a boost in efficiency though, because the use of more track segments and a finer binning leads to more accurate track parameters, resulting in better input for the neural network [4]. The rejection rate is slightly worse, which is likely a result of background tracks far from the vertex being rejected by the 3DFinder and not counting towards the rejection rate (see fig. 8.8a). Even ignoring those biases, the

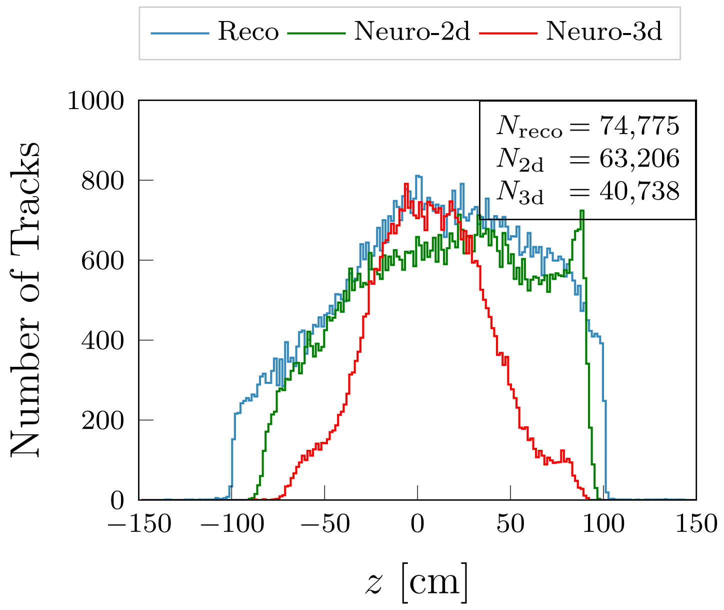
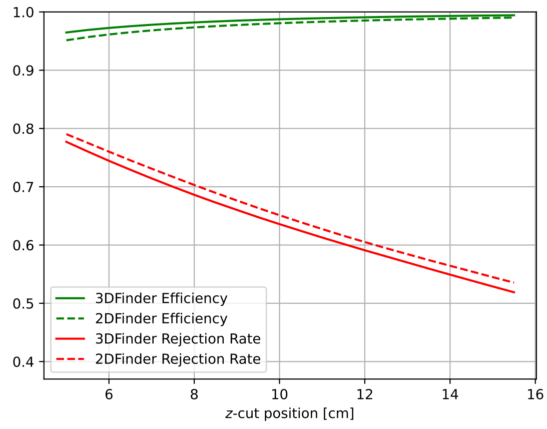


Figure 8.7: Found tracks for a distribution uniform in z within the interval [−100 cm, 100 cm]. Not all tracks are found by the reconstruction (blue curve), as shallow ones may escape the CDC without hitting enough wires. The 3DFinder (red curve) finds more tracks than the 2DFinder (green curve) near the interaction region, but much fewer further away from the vertex [4].

network trained and evaluated on input from the 3DFinder performs a little better than the previous one. (see fig. 8.8b). At an efficiency of 95 %, the rejection rate goes up from 79.38 % to 81.78 % for a network with 4x80 hidden nodes.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

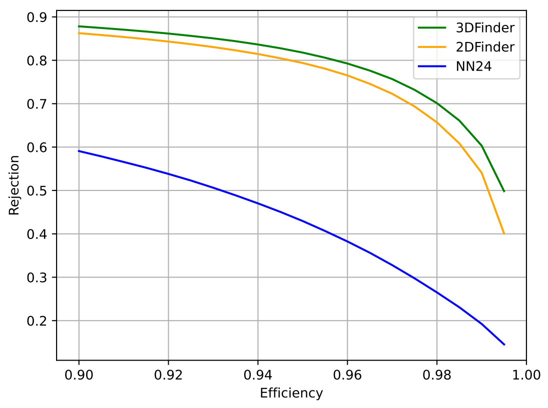


Figure 8.8: Efficiency and rejection with and without 3DFinder input.

Chapter 9

Further Optimization

9.1 Number of Nodes

While the new UT4 units are much more powerful than the UT3s, there are still hardware limits. It is therefore desirable not to make the neural networks bigger than necessary. In section 7.2 it was already pointed out that the number of nodes per hidden layer does not have a big impact on the performance, making it a good lever to minimize resource usage. A comparison of networks with four hidden layers and between 40 and 100 nodes per hidden layer can be seen in fig. 9.1. The smallest network with 40 nodes per hidden layer does have a noticeably worse performance than the other ones, but 60 nodes already seem to be enough (see fig. 9.1a and table 9.1). With more advanced preprocessing, i.e. an ADC-cut removing background hits in the track segments and input to the network based on the 3D track model, the differences are even smaller, so there is no need to have more than 60 nodes in a 4-hidden layer network architecture(see fig. 9.1b). At efficiencies between 94 % and 98 %, the network with 60 nodes per hidden layer even slightly outperforms the one with 80 nodes, with the rejection rate at 95 % efficiency increasing from 81.78 % to 82.50 % (compare table 9.1). This architecture will thus be used for the remainder of this thesis.

(a) Only Extended Input (XI) (b) XI with ADC-Cut and 3DFinder Input

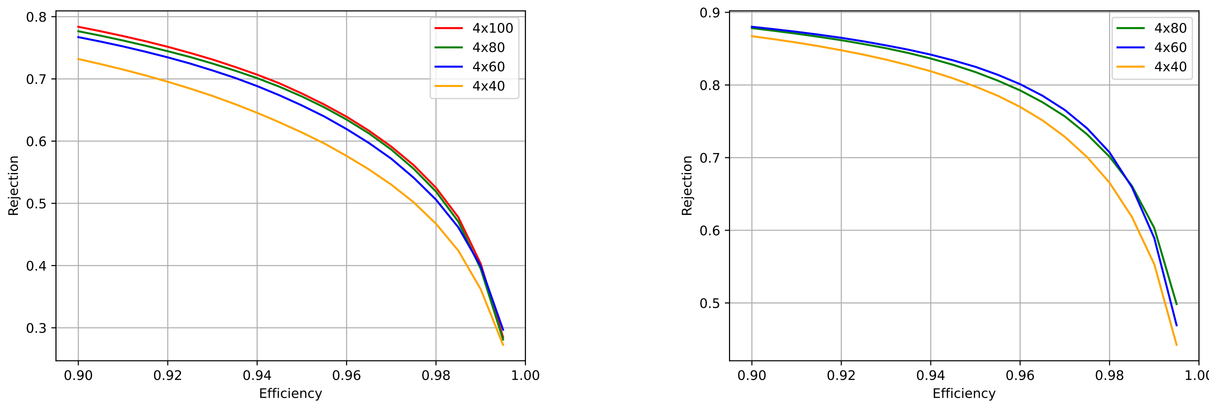


Figure 9.1: Comparison of different amounts of nodes per hidden layer.

Architecture	Rejection with XI	Rejection with XI, ADC-Cut and 3DF
4x40	61.39 %	79.78 %
4x60	65.74 %	82.50 %
4x80	67.18 %	81.78 %
4x100	67.65 %	not evaluated

Table 9.1: Comparison of the rejection rates at an efficiency of 95 % for 4-hidden layer network architectures. In the left column, networks are using extended input, but not yet an ADC-cut or input from the 3DFinder (3DF). On the right side, networks with extended input, an ADC-cut at 10 and 3DFinder input are evaluated.

9.2 Exploring Theta Experts and Direct Theta Input

Most of the potential of the 3DFinder has already been used in section 8.3, but there is one more feature that has been ignored until now. Apart from the inverse momentum and the azimuth angle ϕ, the 3DFinder also makes a prediction for the polar angle θ. While it is much less accurate than the network's final θ prediction (see fig. 9.2), there are still some ways it could be used.

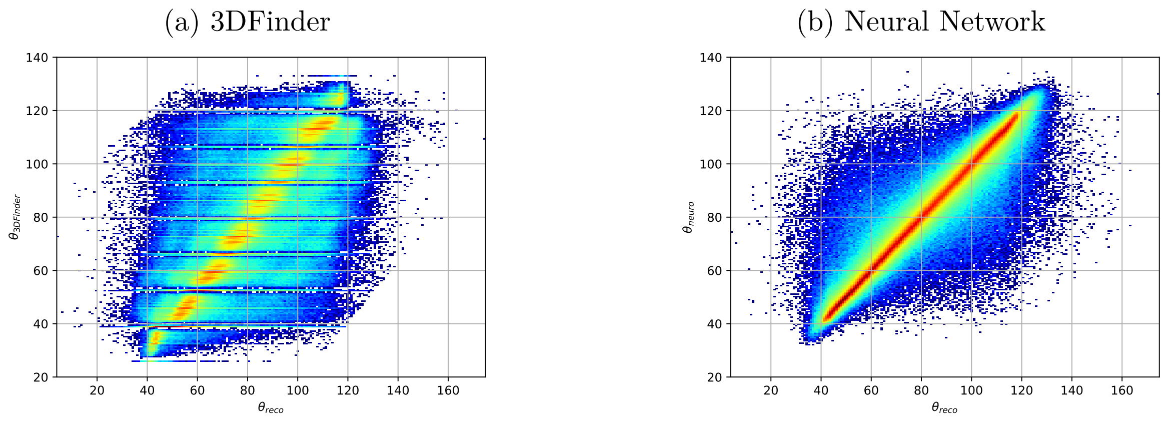


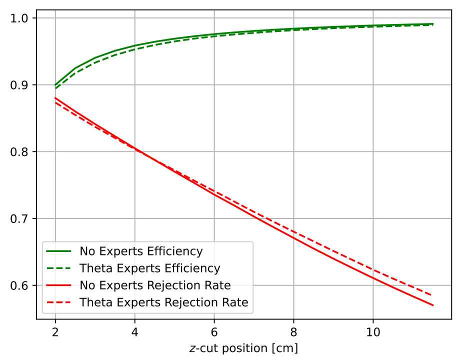
Figure 9.2: θ Predictions of the 3DFinder (left) and the neural network (right).

Theta Experts

The first option considered is to use θ experts. While in section 7.3 it was demonstrated that the network performance does not benefit from experts for missing stereo layers, θ experts could still be viable. As θ is not a discrete quantity, the boundaries between the experts can be chosen so that every expert has a sufficient amount of tracks for training. This is the case for experts from 20° to 60°, from 60° to 100°, and from 100° to 140°, where the data sets have 1,127,948, 1,086,987, and 753,978 tracks, respectively.

Using those three experts, the performance is a bit worse that the one without experts (see fig. 9.3). For an efficiency of 95 %, the rejection rate decreases from 82.50 % to 81.04 %. Again, the reasons for this are a matter of speculation. It seems that the problem with experts is not necessarily the smaller amount of input data, but rather the lack of variety in it, making the whole concept possibly obsolete.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

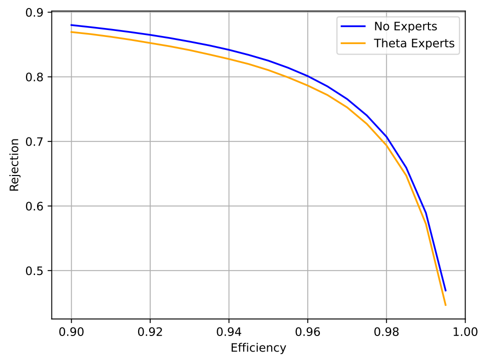
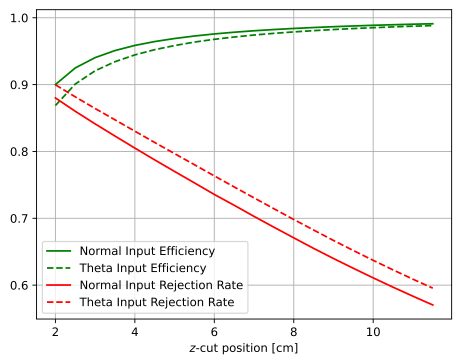


Figure 9.3: Performance with and without theta experts using the same network architecture.

Direct Theta Input

Another way to use the 3DFinder's θ prediction is to pass it directly to the network as an additional input node. This results in a promising increase in the rejection rate. Unfortunately there is also a slight efficiency loss, especially in the region of strict z-cuts below 6 cm, which will be the more important in the future. Therefore, the ROE curve lies mostly below the one without direct θ input, only surpassing it for small efficiencies. At an efficiency of 95 %, the rejection drops slightly from 82.50 % to 81.88 %. The reason for this can be found in fig. 9.2. The 3DFinder's prediction is not bad, but the network has a much higher accuracy in predicting θ, even without direct theta input. Adding information that is already included in the input data does not lead to a better performance.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

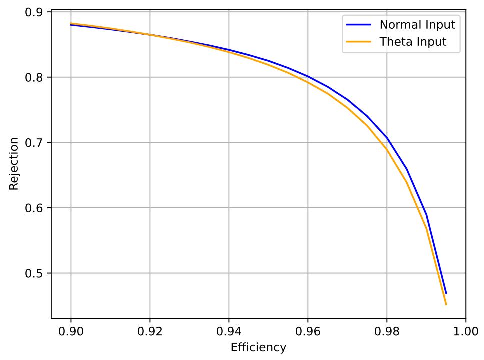


Figure 9.4: Performance with and without direct theta input.

9.3 Binary Input for Additional Drift Cells

As explained in section 8.1, the eleven additional inputs in each track segment are used to pass the drift times of these wires to the neural network. This seems similar to the priority wire, where the drift time has a big impact on spacial resolution. There is one key difference however: There is no left-right-information available for the non-priority wires. Therefore, the drift times only contain information about how far from the wire the particle passed, but not on which side. Another problem is that the time resolution for non-priority wires is limited from 2 ns to 32 ns. In order to evaluate the impact of drift times in the extended input, a new type of input was defined. Here, a wire is assigned a value of one if it was hit, i.e. exceeded the ADC-cut, and a value of zero if it was not hit.

(a) Efficiency and Rejection for Different z-Cuts

(b) ROE curve

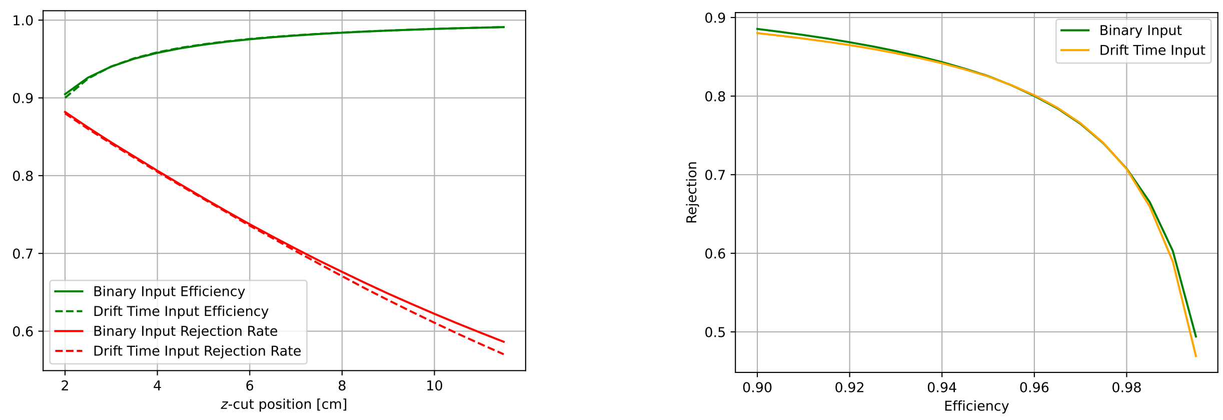


Figure 9.5: Comparison of binary input and drift time input.

The impact of this change can be seen in fig. 9.5. There is little difference between the two input types, with the rejection rate at 95 % efficiency increasing by only 0.04 %, so there is no advantage to using the drift times of non-priority wires. In addition, a binary input is easier to implement in hardware, so it will be used for the further studies below.

9.4 Additional Classification Output Node

Motivation

Up to this point, the quality and quantity of the input and the architecture of the hidden layers have been optimized. However, there are still two important issues to address, which can be done by focusing on the network output.

The first one has existed from the beginning and has only become more severe with the introduction of extended input: Many background tracks are still mapped to the vertex, with a high peak around $z = 0$ cm (see fig. 9.6). Therefore, it is almost impossible to increase the rejection rate beyond a certain point, even with z-cuts of less than 2 cm. The next problem can be seen in the signal prediction. Due to the asymmetry of the detector,

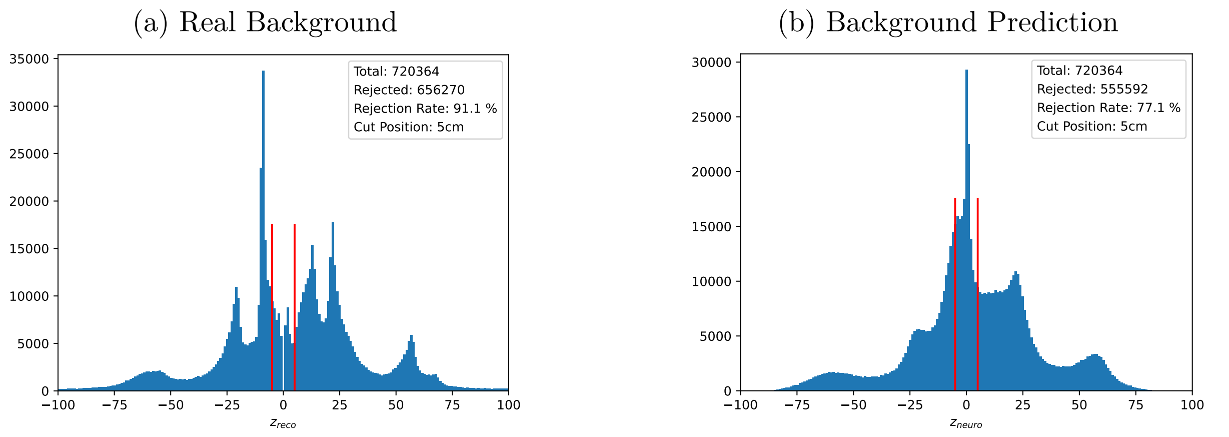


Figure 9.6: Real Background distribution and background prediction by the current best network with 4x100 hidden nodes, extended input, an ADC-cut at 10 and input from the 3DFinder.

the peak is not quite symmetric around $z = 0$ cm, raising the question of whether symmetric z-cuts cause a bias.

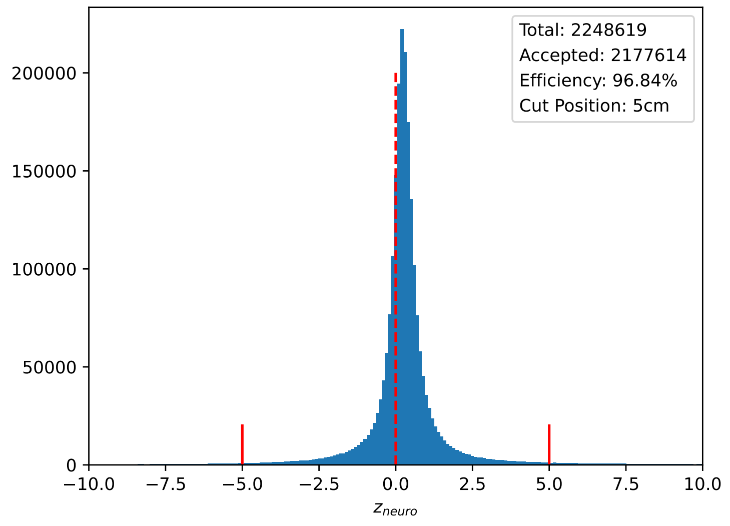


Figure 9.7: Signal prediction with z-cuts at $z = \pm 5$ cm. Of the 71,005 signal tracks rejected by the cut, 44,157 have $z_{neuro} > 5$ cm while only 26,848 have $z_{neuro} < -5$ cm.

An obvious solution to the second problem would be to choose asymmetric z-cuts. For example, the cuts could be made at about -4.1 cm and 5.9 cm, achieving the same efficiency as with cuts at $z = \pm 5$ cm. Then the same number of signal tracks would be rejected on both sides. The strong asymmetry of the middle peak in the background prediction causes the rejection rate to increase from 77.1 % to 77.9 %, so the performance benefits from the adjusted z-cuts. In the end, however, they are not necessary, because there is a more elegant solution that mitigates both the Feed-Down and the asymmetry.

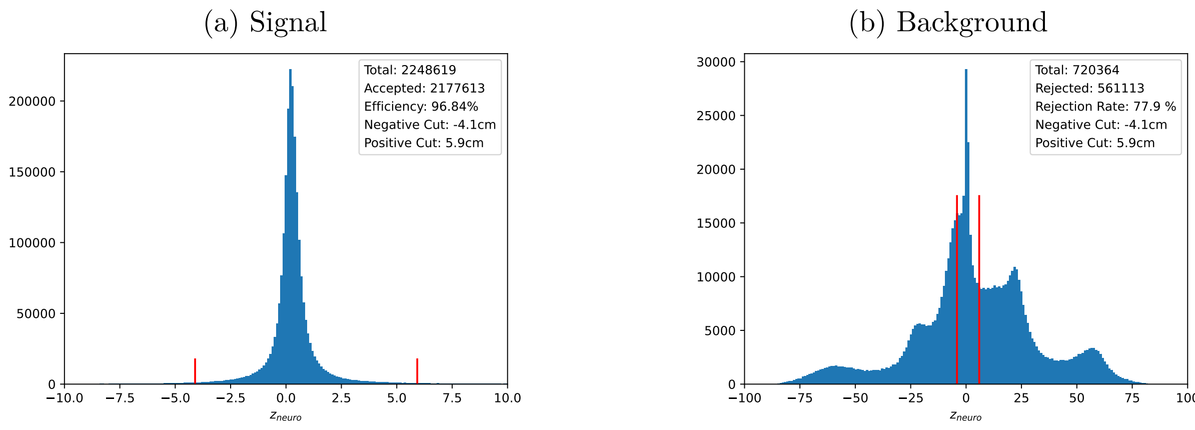


Figure 9.8: Signal and background prediction with asymmetric z-cuts.

Concept

So far the only goal of the neural network was to predict the z value as accurately as possible, for background and for signal tracks. A background track correctly predicted at 2 cm will have a very good contribution to the loss function, although it is almost impossible to reject with a z-cut. Even a prediction at $z = 0$ will not be too bad for the loss function because of the small distance to the actual position. To change this, a third output node can be introduced, which is 1 for signal and 0 for background tracks. As there is not yet a clear label for signal and background tracks in the input data, the classifications of all tracks with $|z| < 1$ cm were set to 1, while all others were set to 0, in accordance to the definition used for the ROE curve. For a discussion on how accurate this definition is, refer to chapter 5.

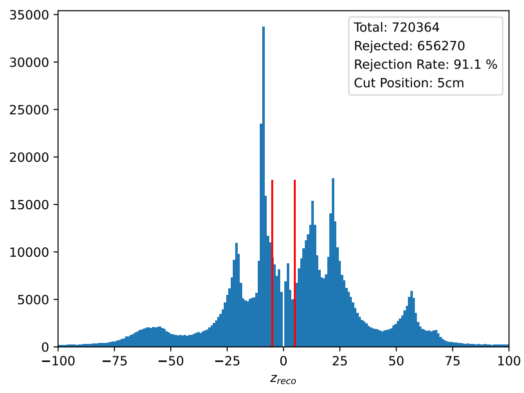
Performance Using z-Cut

Even before using the third output node for separating signal and background, the different training leads to a better accuracy in background prediction. The real peaks in the background are more clearly pronounced and the artificial peak at $z = 0$ cm is slimmer (fig. 9.9). Therefore, a much higher rejection rate can be achieved, with a smaller but relevant improvement of the efficiency for stricter z-cuts below 6 cm (fig. 9.10a). Consequently, the new network has a higher performance than an identical network without a classification node. For an efficiency of 95 %, the rejection rate increases from 82.54 % to 87.89 %.

Performance Using Classification Cut

The network's improved vertex prediction is already quite useful, but it really shines when you start applying a classification cut. As shown in fig. 9.11, signal and background are very neatly separated, making a wide range of cuts viable. For example, a cut at 0.5, indicated by the red line, results in an efficiency of 97.88 % and a rejection rate of 86.61 %. A detailed analysis of the performance for different cuts can be seen in fig. 9.12. For low efficiencies the rejection rates are already better than using a z-Cut, but the biggest improvement can

(a) Real Background



(b) Background Prediction

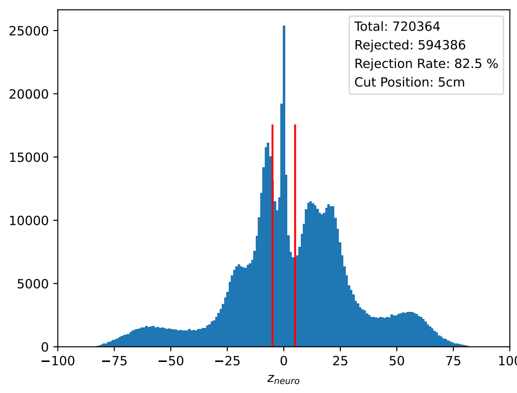
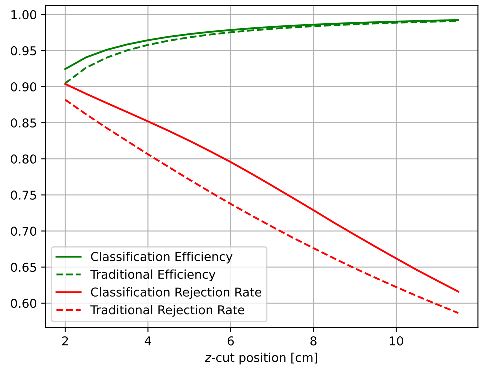


Figure 9.9: Real background distribution and background prediction by the network trained with classification node.

(a) Efficiency and Rejection for Different z-Cuts



(b) ROE curve

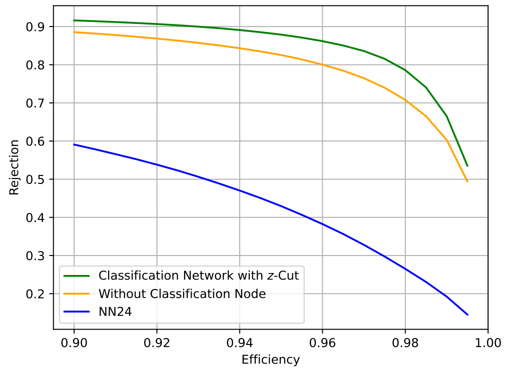


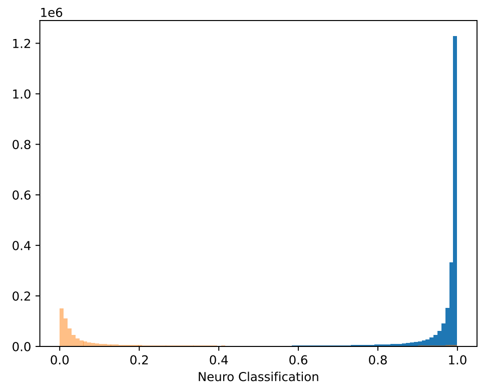
Figure 9.10: Performance with classification node and z-cut.

be observed for efficiencies over 98 %, where the rejection rate stays remarkably stable. Even with an efficiency of 99 %, a rejection rate of more than 80 % can be achieved (see table 9.2).

Performance Using Combined Cuts

There are now two very effective cuts for classifying tracks into signal and output, so a natural question is whether they can be combined. This can be done by requiring a track to be accepted by either both cuts or at least one cut. The former is displayed in fig. 9.13. In fig. 9.13a, for each of the new curves a relatively loose z-cut was chosen. On the remaining data, different classification cuts were applied, and the resulting efficiencies and rejection rates were added to the plot. For low efficiencies all curves are close to the curve with only the classification cut, meaning that the z-cut does not have a high impact. As the maximum efficiency for a given z-cut is approached, the rejection rate decreases, until the curve ends at the corresponding point on the z-cut curve, where the classification cut is at 0 and has no effect. Something similar happens with a fixed classification cut. This time

(a) Linear Scale



(b) Logarithmic Scale

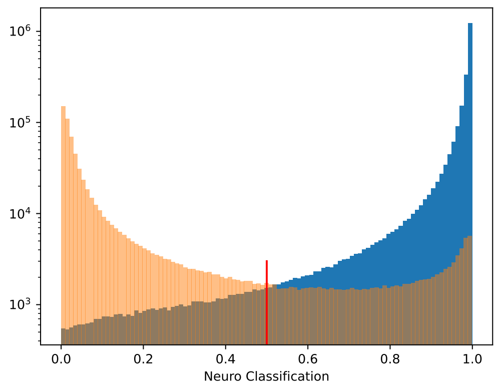
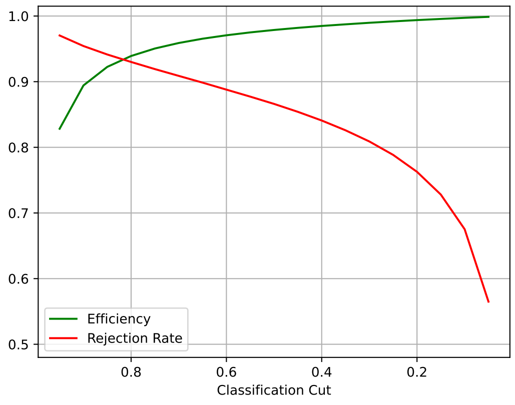


Figure 9.11: Histogram of classification values for signal (blue) and background (orange).

(a) Efficiency and Rejection for Different Classification Cuts



(b) ROE curve

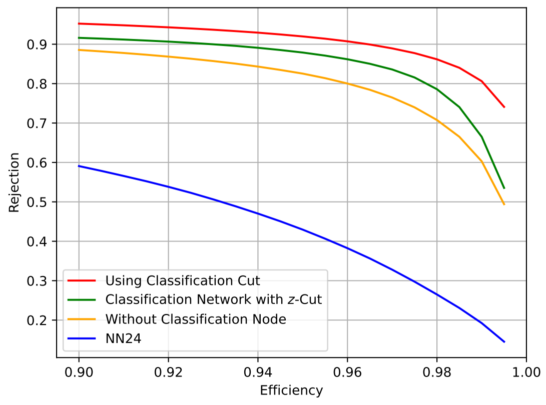


Figure 9.12: Performance using classification cuts.

Network	**Rejection at** 95 % **Efficiency**	**Rejection at** 99 % **Efficiency**
No Classification	82.54 %	60.29 %
Class with z-Cut	87.89 %	66.51 %
Classification Cut	91.98 %	80.62 %

Table 9.2: Rejection rates for different efficiencies for a normal network with 4x60 hidden nodes, a network of the same architecture with a classification node, but still using a z-cut, and the same network using a classification cut.

the curve starts on the curve for the z-cut and meets the other curve at the corresponding efficiency.

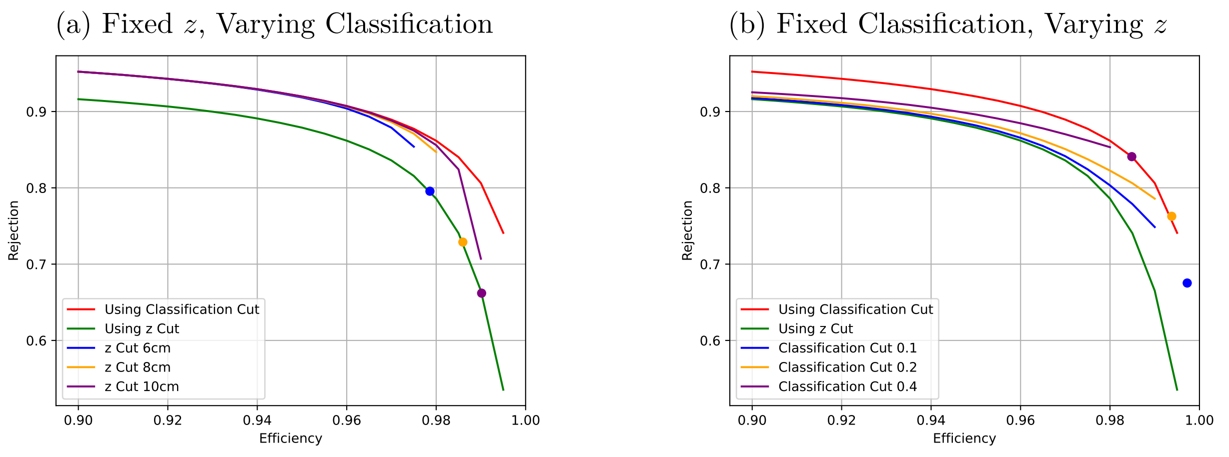


Figure 9.13: Performance when requiring both cuts. On the left side, the dots indicate the performance with only a z-cut, i.e. a classification cut at 0. When the classification cut is increased, the curve is getting closer to the one with only the classification cut, but it never surpasses it. The graph on the right side works analogously, but with z-cut and classification cut switched.

It can be concluded that adding a z-cut after a classification cut only makes the performance worse and is not worth considering.

Changing to the other approach, i.e. only requiring one of the two cuts to be fulfilled, leads to very similar plots. Now the lines go the other way, but the result is effectively the same (see fig. 9.14).

The reason for this is the strong correlation between the z prediction and the classification output, seen in fig. 9.15. There are very few tracks where there is a large discrepancy between the two metrics, so that combining the parameters does not improve accuracy. This point is further supported by the plots in fig. 9.16. They show that after a relatively reasonable classification cut of 0.4 is applied, the z prediction of the remaining background looks a lot like the signal prediction. Similarly, the signal tracks rejected by the cut are so spread out that it is not possible to further increase the efficiency by applying a z-cut without significant losses in the rejection rate.

One might then question if the output nodes for z and θ are even necessary. This was evaluated by training a new network with only a classification node. The resulting performance is almost identical, as fig. 9.17 shows, so the two extra output nodes do not improve the performance. However, they still offer a way to track the performance and at least loosely compare it to other networks. The estimate for the polar angle θ is also necessary for the STT. Therefore, networks will continue to be trained with three output nodes.

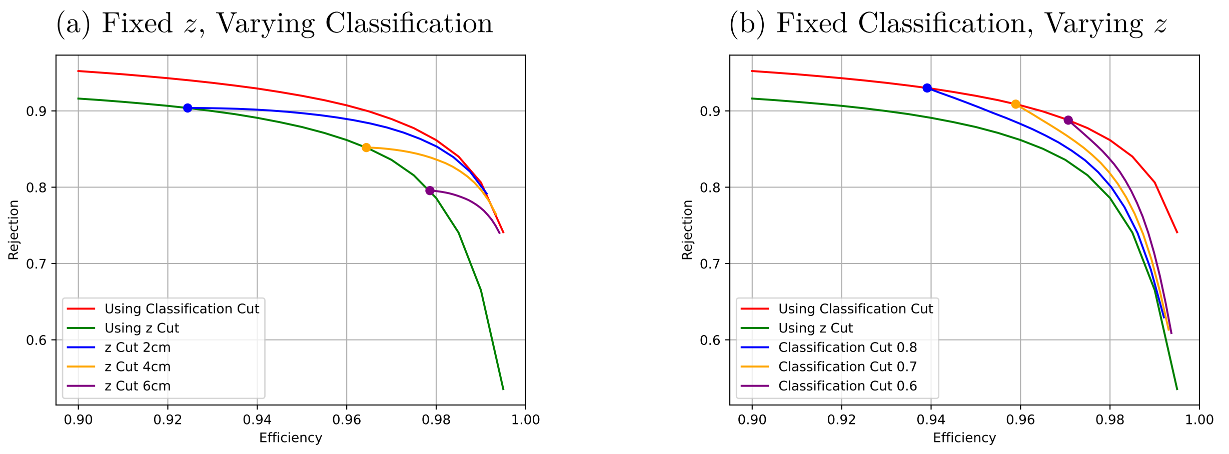


Figure 9.14: Performance when requiring only one cut. The plots work like the ones in fig. 9.13, but now a track is accepted when it passes either the z-cut or the classification cut.

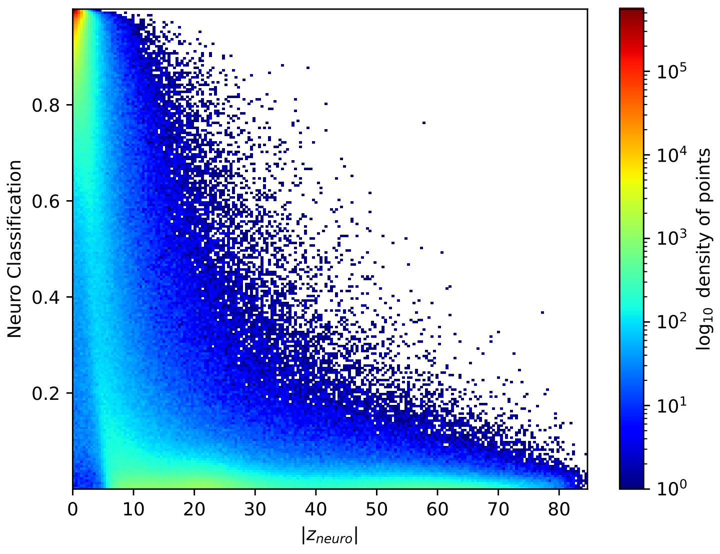


Figure 9.15: Correlation between z prediction and classification output.

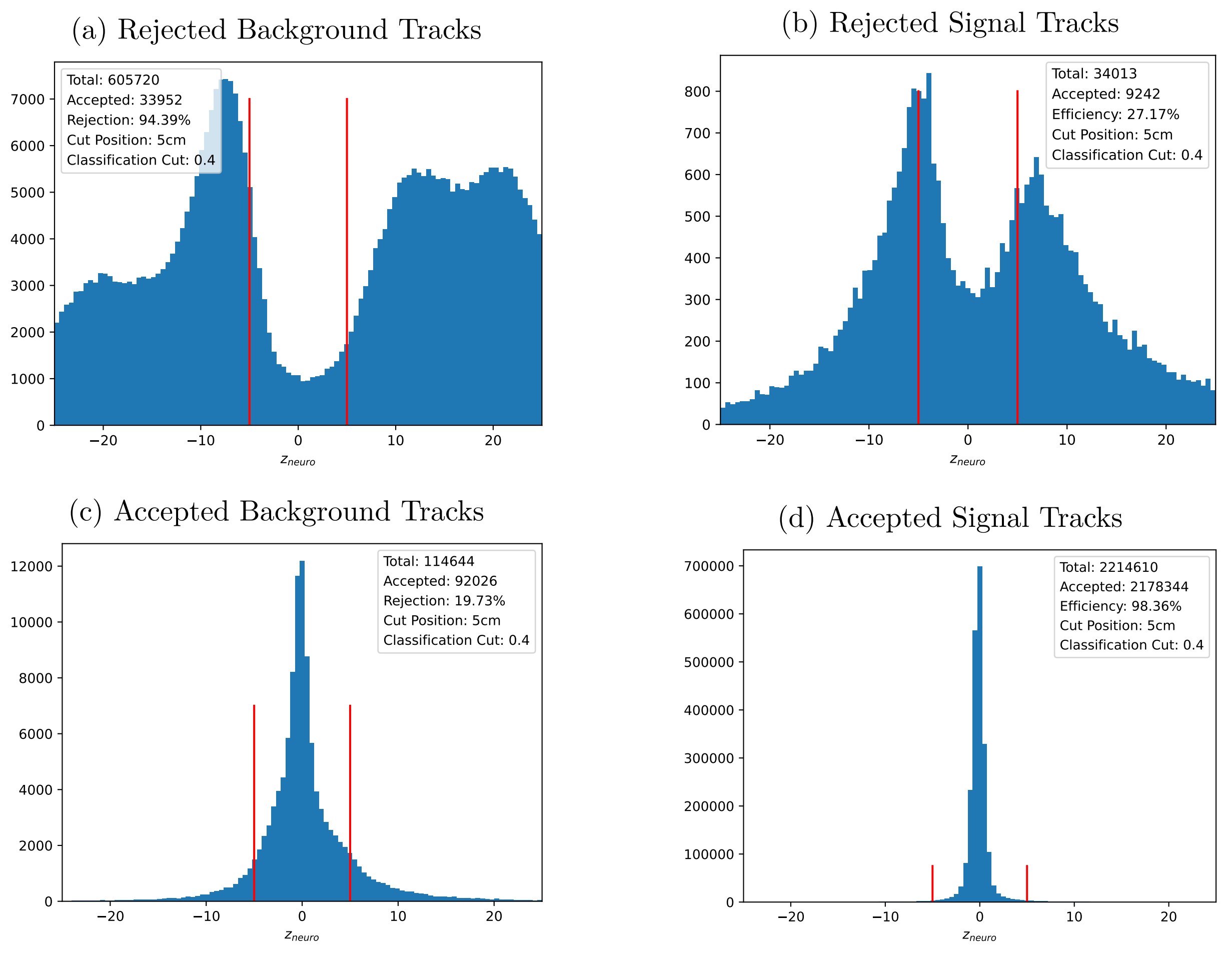


Figure 9.16: Histograms of background and signal predictions on tracks accepted and rejected by a classification cut of 0.4.

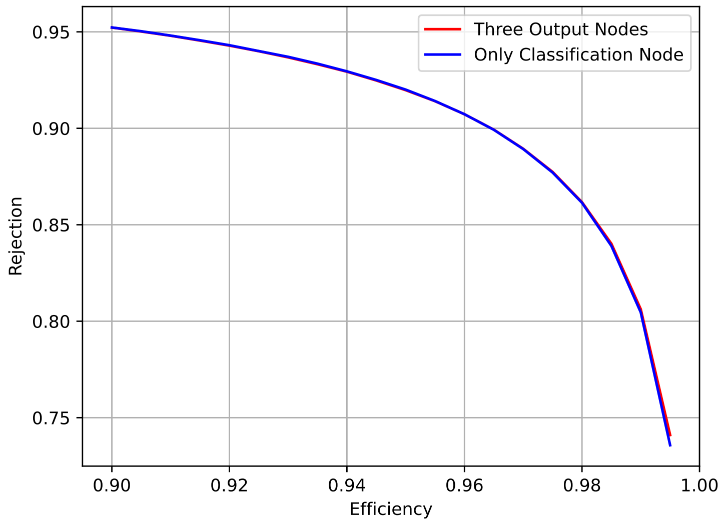


Figure 9.17: Comparison of performance with three output nodes and with only a classification node. The curves are almost identical for most efficiencies.

Chapter 10

Final Performance Evaluation

So far, when a new concept was introduced, it was only compared to the same network without the new feature. This ensured that the exact impact of the change was evaluated, but there were no direct comparisons to the more basic networks. Therefore, in this section all differences between NN22 and the final network will be summarized, and the performance will be compared.

The first change in the data pipeline is the implementation of an ADC cut at a value of 10, applied already before the Track Segment Finder to reduce noise. The 2DFinder is replaced with a 3DFinder, leading to better suppression of fake and background tracks and more accurate parameters. In addition to the three parameters for the priority wires, the hitpattern in the Track Segment is passed to the network for every superlayer. Instead of only one hidden layer with 81 nodes, four hidden layers with 60 nodes each are used. Finally, the existing output nodes that predict z and θ are supplemented by a third output node that returns 1 for signal and 0 for background tracks. The z-cut, which was previously used for separating signal and background, is replaced by a cut on the value of the classification node, further improving the performance. This network will be compared to the network from May 2022 (NN22) below.

Double Gaussian

While it has become increasingly clear throughout this thesis that the double Gaussian is not suitable for high-level performance comparison of neural networks, it is still a good measure of the z resolution. The width of the core Gaussian decreased from 2.13 cm to 0.53 cm, so the accuracy for good tracks increased by a factor of four. The wide Gaussian, which is an important indicator of the effectiveness of the z-cut, went from 9.45 cm to 3.06 cm.

z-Scatterplot

The better accuracy can also be seen in the z-scatterplot. Most points are very close to the ideal diagonal line and hardly any tracks with a displacement of more than 30 cm are mapped to the vertex. With a z-cut of 5 cm (dashed), the new network can accept about as many signal tracks as the old network did with a z-cut at 15 cm (solid line). It should

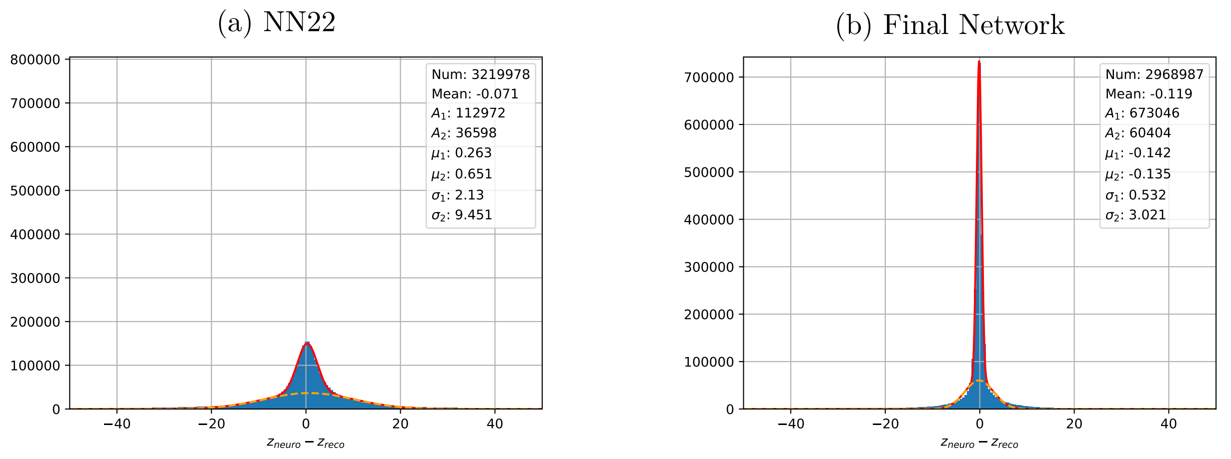


Figure 10.1: Double Gaussian for previously used network and final network.

be noted though that many tracks in the vicinity of the vertex are predicted very close to $z = 0\,\text{cm}$. The Feed-Down may not be as obvious anymore, but it still exists and is very hard to get rid of.

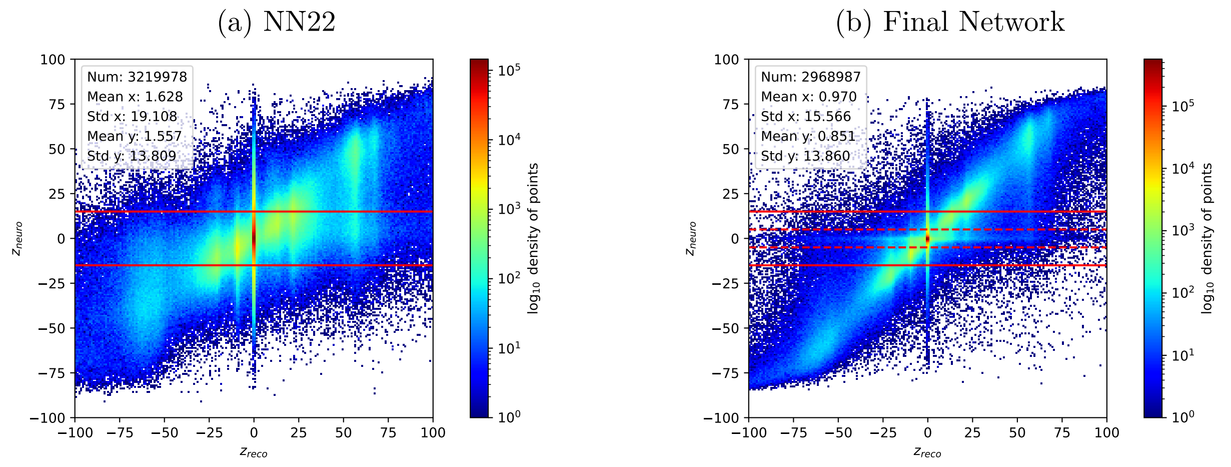


Figure 10.2: z-Scatterplot for previously used network and final network. The solid lines indicate a z-cut at $\pm 15\,\text{cm}$, the dashed lines a z-cut at $\pm 5\,\text{cm}$.

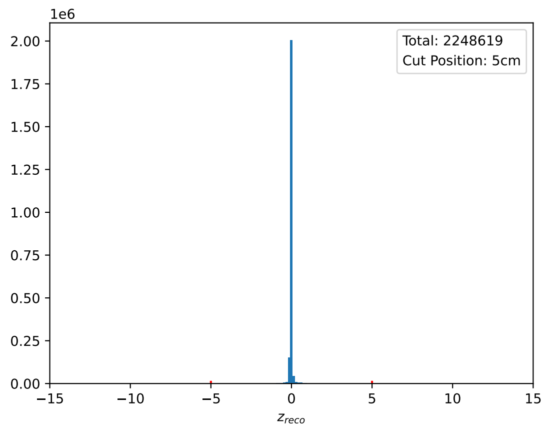
Signal and Background Prediction

As implied by the scatterplot, the z prediction has become much better with the new network. This is obvious for the signal tracks, but also very noticeable in the background, where the structure of the real tracks is now clearly visible (see fig. 10.3).

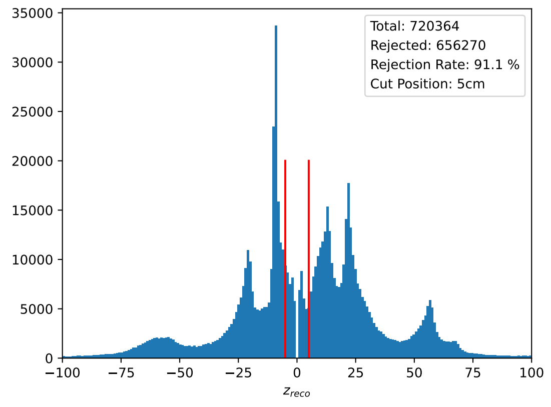
Efficiency and Rejection

It is clear from previous observations that the new network has a much higher efficiency for all z-cuts than the old one. For a z-cut of $2\,\text{cm}$, where the old network accepts less than half of the signal tracks, the new network still has an efficiency of well over $90\,\%$. What

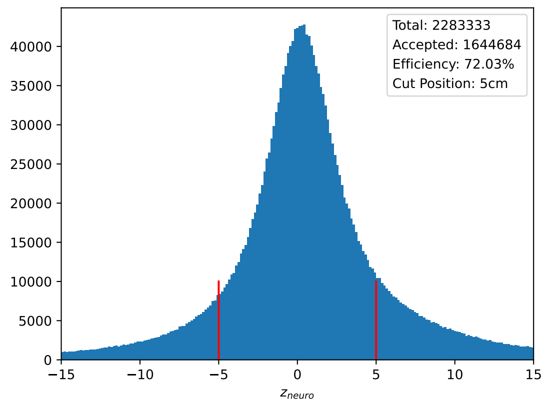
(a) Real Signal



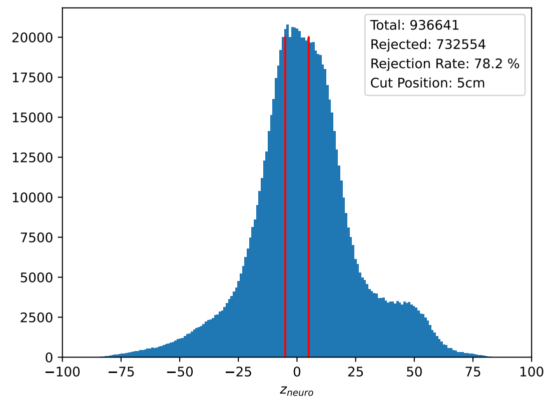
(b) Real Background



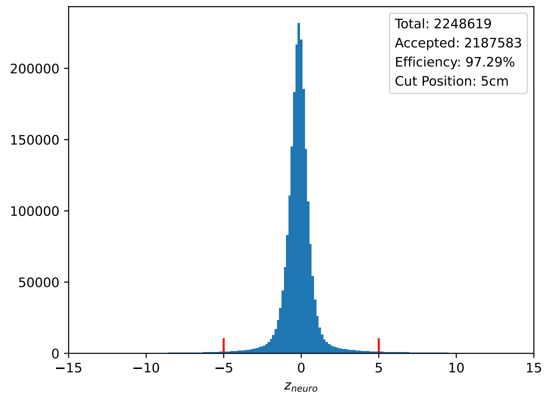
(c) NN22 Signal Prediction



(d) NN22 Background Prediction



(e) Final Network Signal Prediction



(f) Final Network Background Prediction

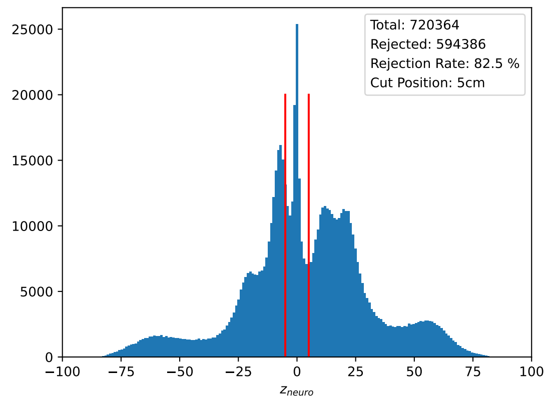


Figure 10.3: Comparison of the double Gaussian with different network architectures.

is not a given is that the rejection rate is also better for all realistic z-cuts, here the old network is only better for cuts below 3 cm. For comparison: The same network without a classification node is already surpassed at around 6 cm because of the strong Feed-Down. There is also a new line in this plot, a blue line indicating the rejection rate for perfect z-prediction. This line shows that for large z-cuts the new network is already very close to an optimal rejection rate. However, the gap grows at z-cuts under 10 cm, because much of the biggest peak in the background is still mapped to $z = 0$ cm (compare fig. 10.3).

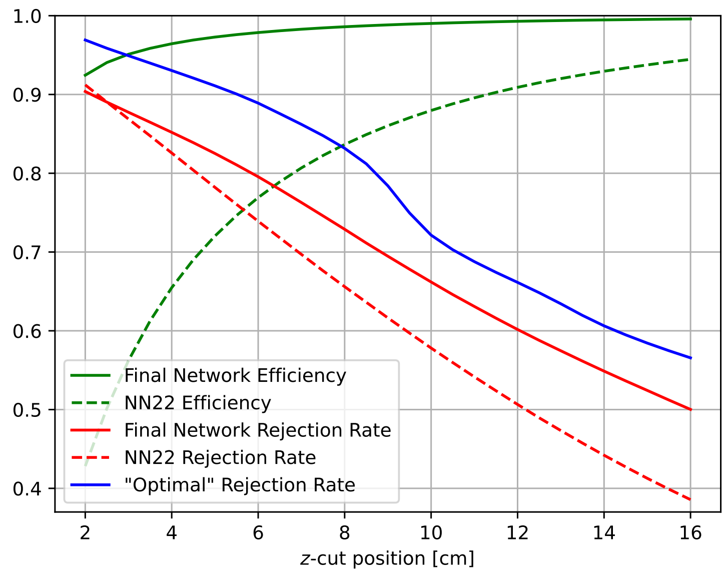


Figure 10.4: Efficiency and rejection for different z-cuts.

Finally, it is time to look at the ROE curve, the most important graph for evaluating network performance (fig. 10.5). Unlike with the other plots, here the classification cut takes effect. Nevertheless, the trends of the previous plots can also be observed here. For equal efficiencies, the rejection rates are consistently at least 40 percentage points higher than the ones of the old networks. In the most relevant region of 96 % to 99 % efficiency, the difference even surpasses 60 percentage points (see table 10.1).

Efficiency	**NN22**	**NN24**	**Final Network**
92 %	47.38 %	53.81 %	94.29 %
95 %	36.31 %	42.95 %	91.98 %
99 %	16.55 %	19.23 %	80.62 %

Table 10.1: Rejection rates of the different networks for efficiencies of 92 %, 95 % and 99 %. Compared to NN22, the rejection rate can be more than doubled while increasing the efficiency from 95 % to 99 %.

In summary, the results show that the changes to the Neural Track Trigger, once implemented, will make it more robust to high background levels. The rejection rate was increased significantly with a simultaneous increase in efficiency.

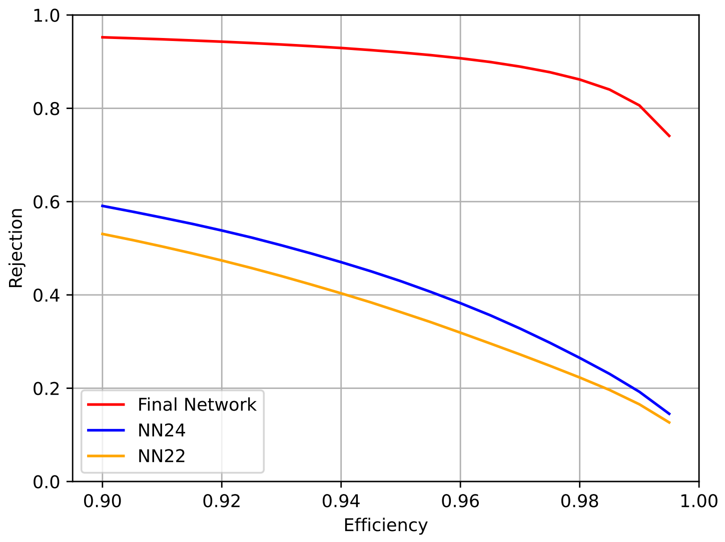


Figure 10.5: ROE curve for the old network, the currently implemented network and the final network.

Chapter 11

Conclusion and Outlook

11.1 Conclusion

Until the Long Shutdown (LS1) in June 2022, the CDC trigger at Belle II used a neural network with 27 input nodes, one hidden layer with 81 nodes and 2 output nodes for predicting the z-origin and polar emission angle θ of track candidates provided by the 2DFinder. Events where at least one track had a prediction of $|z_{neuro}| < 15\,\mathrm{cm}$ and a momentum of $p > 0.7\,\mathrm{GeV}$ were triggered. This minimum bias Single Track Trigger (STT) worked well at the luminosities that were achieved, which did not exceed about $3 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. However, when reaching luminosities beyond $4 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, a strong rise of the background was observed, resulting in an increase of the STT trigger rate by about a factor of 2, almost saturating the data acquisition system. In this thesis, it was studied how the higher capabilities of the UT4 boards installed during LS1 can be utilized to stabilize the performance of the Neural Track Trigger under severe background conditions.

The first step was retraining a network with the current architecture on high background data. The resulting network slightly improved the performance and has therefore already been implemented in hardware and is running successfully in the current data taking after LS1. For further improvement, deeper network architectures were evaluated. A network with four hidden layers, each containing 60 nodes, was chosen because it provided the best performance. Next, the network input was extended to include information on the hitpattern of each track segment. In addition, a cut on the wire signals was applied, reducing noise from electronic cross-talk and synchrotron photons. Furthermore, the 2DFinder was replaced with a track finder in three dimensions (“3DFinder”) in order to suppress fake and background tracks and provide more accurate network parameters. Finally, an additional output node was added to the network that directly classifies a track as signal or background.

Using this network and a cut on the value of the classification node, the background rejection rate for a single track efficiency of 95 % was increased from 36.3 % to 92.0 %. Even an efficiency of 99 % is now feasible, with a rejection rate of 80.6 %.

11.2 Outlook

Now that the Neural Track Trigger has been optimized, the changes need to be implemented into the hardware. This is already being worked on, starting with the 3DFinder. Recent work on the hardware also suggests that much deeper networks with fewer nodes per hidden layer may be easier to implement, so such networks are currently being tested.

Once the upgraded track trigger is online, it should work for the next few years, with retraining the network and loading the weights into the hardware whenever necessary. However, there are still some problems that need to be addressed. Originally, one hope for the classification node was that it would significantly reduce the Feed-Down. It did have a positive effect, but as fig. 11.1 shows, background tracks that are predicted close to the vertex still tend to be accepted by a classification cut. The plan is therefore to tackle the root of the problem, namely the strong bias towards the vertex in the training data. In order to obtain unbiased input data, a new so-called ”f-stream” was implemented, that still requires the full trigger information, but no longer relies on the L1 trigger to accept an event before recording it. Using this, there will soon be enough data for training better neural networks and evaluating them more accurately.

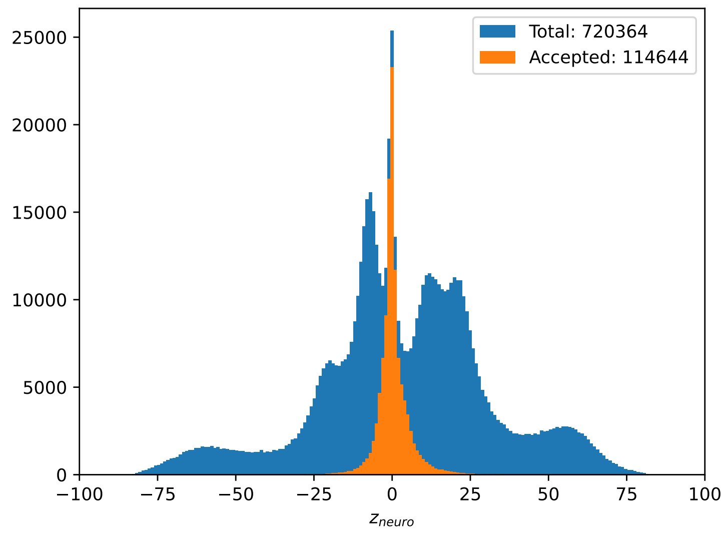


Figure 11.1: Background prediction for all tracks (blue) and for tracks accepted by a classification cut of 0.4 (orange).

Another “problem”, though introduced by construction, is that even for the single hidden layer network, the efficiency drastically dropped for large displacements from the vertex (see fig. 11.2). This effect is expected to be even larger for the upgraded track trigger. Therefore, feebly interacting particles with long lifetimes leading to a vertex far from the interaction point are unlikely to be triggered. In order to become sensitive to this very interesting new class of events (e.g. expected in certain models for dark matter production in electron positron annihilation), there are plans to launch a Displaced Vertex Trigger, first proposed in [13].

As luminosity increases further towards the design luminosity of $6 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, there might eventually come a point where even with the new 3D track finding model the Neural

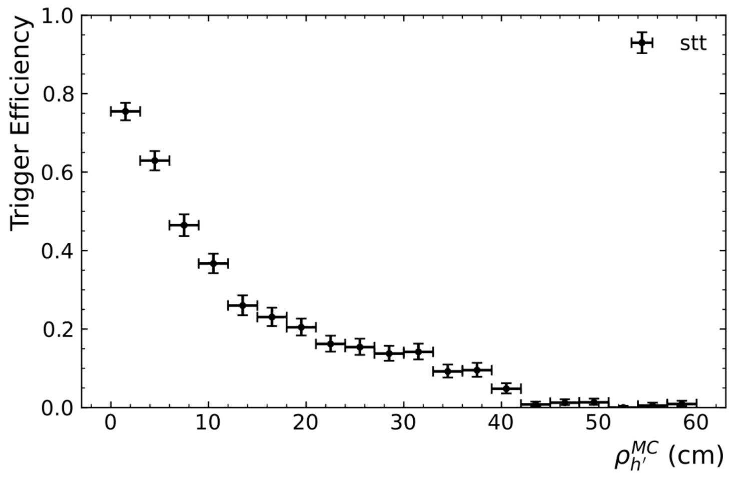


Figure 11.2: Trigger efficiency as a function of the displacement from the vertex [13].

Track Trigger will no longer be able to cope with the background. A promising approach for this scenario is to use Graph Neural Networks (GNNs) for track finding, which have shown to be highly efficient even at very high backgrounds [26]. However, those networks will require even more powerful FPGAs, so they will most likely not be implementable in the near future by our collaborators.

Acknowledgments

First and foremost, I am very grateful to my supervisor, Prof. Christian Kiesling, for providing me with the opportunity to write this thesis. He has always been very supportive and encouraging, sharing his enthusiasm for the project and physics in general. The discussions with him have taught me a lot, and I am looking forward to working with him in the future.

I also want to express my gratitude to Kai Unger for implementing the Neural Track Trigger into hardware and for the valuable discussions about the feasibility of different networks.

I want to thank Felix Meggendorfer for giving valuable input at our weekly trigger meeting and for answering my questions on the PyTorch repository and basf2.

Furthermore, I am grateful to Yuxin Liu for sharing his code for training data generation and for helping me with the implementation of it.

I also want to thank my friend and colleague Simon Hiesl for being very supportive and always open to discuss new results.

Finally, I would like to thank the head of the Belle II group at MPP, Dr. Hans-Günther Moser, who has helped me with the organizational matters.

Bibliography

[1] T. Abe et al. *Belle II Technical Design Report*. 2010. eprint: arXiv:1011.0352. URL: https://arxiv.org/abs/1011.0352.

[2] S. Bähr et al. *The Neural Network First-Level Hardware Track Trigger of the Belle II Experiment*. 2024. eprint: arXiv:2402.14962. URL: https://arxiv.org/abs/2402.14962.

[3] Sebastian Skambraks. "Efficiency Physics Signal Selectors for the First Trigger Level of the Belle II Experiment based on Machine Learning". PhD thesis. Ludwig-Maximilians-University Munich, 2020. URL: https://www.mpp.mpg.de/~cmk/Belle/Sebastian_Thesis.pdf.

[4] Simon Hiesl. "Upgrade of Belle II's Neural Network Trigger by Track Finding in Three-Dimensional Hough Space". MA thesis. Ludwig-Maximilians-University Munich, 2024. URL: https://docs.belle2.org/record/4347/files/BELLE2-MTHESIS-2024-018.pdf.

[5] M. Tanabashi et al. "Review of Particle Physics". In: *Phys. Rev. D* 98 (2018). URL: https://link.aps.org/doi/10.1103/PhysRevD.98.030001.

[6] A. Crivellin et al. *The Anomalous Magnetic Moment of the Muon: Beyond the Standard Model via Chiral Enhancement*. 2022. eprint: arXiv:2207.01912. URL: https://doi.org/10.48550/arXiv.2207.01912.

[7] Sara Pohl. "Track Reconstruction at the First Level Trigger of the Belle II Experiment". PhD thesis. Ludwig-Maximilians-University Munich, 2017.

[8] E. Kou et al. *The Belle II Physics Book*. 2018. URL: https://arxiv.org/pdf/1808.10567.

[9] Andrei Sakharov. "Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe". In: *Sov. Phys. Usp.* 392.34 (1991). URL: https://beta.iopscience.iop.org/article/10.1070/PU1991v034n05ABEH002497/pdf.

[10] Makoto Kobayashi and Toshihide Maskawa. "CP-Violation in the Renormalizable Theory of Weak Interaction". In: *Progress of Theoretical Physics* 49.2 (1973). eprint: https://academic.oup.com/ptp/article-pdf/49/2/652/5257692/49-2-652.pdf. URL: https://doi.org/10.1143/PTP.49.652.

[11] Lincoln Wolfenstein. "Parametrization of the Kobayashi-Maskawa Matrix". In: *Phys. Rev. Lett.* 51 (1983). URL: https://link.aps.org/doi/10.1103/PhysRevLett.51.1945.

[12] J. Brodzicka et al. “Physics achievements from the Belle experiment”. In: *Progress of Theoretical and Experimental Physics* 2012.1 (2012). URL: https://doi.org/10.1093/ptep/pts072.

[13] Elia Schmidt. “Developing a Displaced Vertex Trigger for Dark Matter Searches at the Belle II Experiment”. MA thesis. Ludwig-Maximilians-University Munich, 2023. URL: https://docs.belle2.org/record/3620/files/BELLE2-MTHESIS-2023-021.pdf.

[14] Deutsches Elektron-Synchrotron DESY. *SuperKEKB and Belle II.* Accessed: 2024-06-21. URL: https://belle2.desy.de/belle_ii_experiment.

[15] Kazunori Akai, Kazuro Furokawa, and Haruyo Koiso. “SuperKEKB collider”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (2018). URL: https://www.sciencedirect.com/science/article/pii/S0168900218309616?via%3Dihub.

[16] Y. Ohnishi et al. “Accelerator design at SuperKEKB”. In: *Progress of Theoretical and Experimental Physics* 3 (2013). URL: https://academic.oup.com/ptep/article/2013/3/03A011/1556583.

[17] *Belle II Detector 3D model.* URL: https://www.belle2.org/archives/.

[18] Mariangela Varela. “Slow Pion Identification using the Pixel Detector of Belle II”. MA thesis. Ludwig-Maximilians-University Munich, 2023. URL: https://docs.belle2.org/record/4101?ln=en.

[19] A. Paladino. “Beam background evaluation at SuperKEKB and Belle II”. In: *JINST* (2020). URL: https://docs.belle2.org/record/1912/files/BELLE2-CONF-PROC-2020-008_submitted.pdf.

[20] Yoshihito Iwasaki et al. “Level 1 Trigger System for the Belle II Experiment”. In: *IEEE Transactions on Nuclear Science* (2011). URL: https://www.phys.hawaii.edu/~idlab/taskAndSchedule/local_DAQ/main_update.pdf.

[21] Paul V. C. Hough. “Man - Machine Collaboration in the Analysis of Bubble Chamber Photography for High - Energy Physics”. In: *Optical Engineering* 2.3 (1964). URL: https://doi.org/10.1117/12.7971269.

[22] D. H. Ballard. “Generalizing the Hough transform to detect arbitrary shapes”. In: *Pattern Recognition* 13.2 (1981). URL: https://www.sciencedirect.com/science/article/pii/0031320381900091.

[23] *Documentation of the PyTorch NeuroTrigger Trainer for Belle2.* URL: https://gitlab.desy.de/b2nnt/nnt-pytorch.

[24] Jonhannes Schmidt-Hieber. “The Kolmogorov-Arnold representation theorem revisited”. In: *Neural Networks* 137 (2021). URL: https://doi.org/10.1016/j.neunet.2021.01.020.

[25] Yuxin Liu, Akimasa Ishikawa, and Taichiro Koga. “Study of Sudden Beam Losses of SuperKEKB and Development of 3D Track Hardware Trigger using Machine Learning at Belle II”. MA thesis. The Graduate University for Advanced Studies, Sokendai, 2023.

[26] Marc Neu et al. “Real-Time Graph Building on FPGAs for Machine Learning Trigger Applications in Particle Physics”. In: *Computing and Software for Big Science* 8.8 (2024). URL: `https://doi.org/10.1007/s41781-024-00117-0`.

Declaration:

I hereby declare that this thesis is my own work, and that I have not used any sources and aids other than those stated in the thesis.

München, den 30.06.2024

Timo Forsthofer