Slow Pion Identification using the Pixel Detector of Belle II

Identifizierung langsamer Pionen mit dem Pixeldetektor von Belle II

Master Thesis
at the Faculty of Physics
of the Ludwig-Maximilians-Universität
Munich

Submitted by
Mariangela Varela
from Panama

September 7, 2023

Supervisor: Prof. Dr. Christian Kiesling
Institute: Max Planck Institute for Physics (Werner Heisenberg Institute)

Abstract

Charged pions coming from D^* decays are very useful to tag the flavor of neutral B mesons. At the Belle II experiment, these pions have low transversal momentum and therefore, a sizeable fraction does not reach all layers of the Silicon Strip Detector (SVD). As a consequence, the tracks of these ”slow” pions are not reconstructed. This will become problematic when the Region Of Interest (ROI) algorithm is implemented online, which extrapolates the reconstructed tracks to the Pixel Vertex Detector (PXD) to select only regions of interests that are then sent to storage, deleting the rest.

This thesis provides a method to recover slow pions, otherwise lost due to ROI, using only the clusters from the PXD. This PXD stand-alone cluster rescue method uses artificial neural networks (NN) to discriminate slow pions from the dominating QED electron background by analysing PXD cluster variables. The neural networks were trained using two approaches, one general and one specific: the former refers to training the NN using data from all PXD layers and all pixel multiplicities. The latter, on the other hand, refers to NNs trained with data separated according to PXD layer number and pixel multiplicity. Monte Carlo (MC) generated data was used to train and to test all NNs. Significant discrimination between slow pions and electrons has been achieved for both approaches. With the general approach, an efficiency of ∼89% was obtained. In addition, efficiencies between 84% and 99% were achieved for the specific approach, with the highest efficiencies corresponding to the cases with large pixel multiplicities (4 or more). Moreover, the algorithm was also tested with both approaches on Early Phase 3 (EP3) MC events and proved to be resilient to a more generalised background, with similar efficiencies achieved as in the case of QED electrons as background. As a last step, real slow pion data was also used for testing, giving results with a decreased performance (∼59%), mainly caused by differences between MC and real data. By considering only clusters with a charge higher than 30 ADU, MC was brought closer to real data and an efficiency of ∼73% was achieved.

Contents

1 Introduction

The observed matter-antimatter asymmetry in the Universe remains one of the biggest unanswered questions in physics today. One of the conditions needed to explain the differences between matter and antimatter is the violation of charge conjugation and parity (CP) symmetry. The Standard Model of particle physics predicts the existence of this phenomenon for interactions mediated by the weak force. However, the prediction is orders of magnitude too small to account for the observed levels of matter-antimatter asymmetry [1]. Hence, additional sources of CP violation must be found.

The Belle II experiment, located in Tsukuba, Japan, aims at finding new sources of CP violation by doing high-precision measurements of B meson decays [2]. To achieve this, large numbers of B mesons are produced in pairs, more precisely in a B meson and an anti B meson, at the SuperKEKB accelerator from electron-positron collisions at the $\Upsilon(4S)$ resonance [3]. About half of the time, the produced pairs are neutral B mesons. To study CP violation in B^0-$\bar{B}^0$ systems, knowledge about their flavor is necessary, which is done using the flavor tagging algorithm [4]. This method requires the full reconstruction of a B meson (signal side), which has decayed in a flavor unspecific CP eigenstate. Once a particular neutral B meson has been reconstructed, the decay modes of the other B meson (tag side) are used to determine the quark flavor content of the signal B. In this way, important parameters can be measured, which allow to investigate CP violating processes. The final objective is to test the Standard Model (SM), as well as to search for New Physics (NP) in rare decays [5].

Charged pions ($\pi^{\pm}$) coming from charged D^* are one of the targets used for identifying $\bar{B}^0$ or B^0 on the tag side. The D^* decays dominantly into $D^0\pi^{\pm}$ and since D^* and D^0 have a small mass difference, the momentum spectrum of the pion peaks at low values around 100 MeV [4]. For this reason, these pions are known as slow pions. With a transverse momentum < 250 MeV, most slow pions do not make it to the Central Drift Chamber (CDC) of the Belle II detector, with a considerable fraction not even traversing all 4 layers of the Silicon Vertex Detector (SVD), the main sub-detector for track reconstruction for such low momentum particles. As a consequence, the tracks of those pions are not reconstructed.

Since the Belle II experiment is expected to reach a peak luminosity of 6×10^{35} cm^{-2}s^{-1}, preparations are taking place to deal with the significant increase of irreducible low energetic electrons from QED processes in the Pixel vertex Detector (PXD). Since the signals from these QED electrons in the PXD would exceed the data from all other sub-detectors by about an order of magnitude, a Region of Interest (ROI) algorithm has been developed to reduce the size of the data stored for the PXD to a reasonable level. It is important to note that the PXD data can not be used during the online track reconstruction. Therefore, a possible reduction of the PXD data must be achieved by the other tracking detectors (SVD and CDC), which reconstruct the tracks online and provide the ROIs for the PXD data: the algorithm extrapolates the reconstructed tracks to the PXD and defines regions of interest to be stored. Everything outside those regions is then deleted. This is a good way to differentiate signal from background in the inner-most detector. However, valuable data such as slow pions which are not reconstructed in the other tracking detectors would be lost.

Being the PXD the only sub-detector which all slow pions reach, this thesis focuses on developing a method to exploit this fact, using only the PXD to identify and rescue those slow pions that would otherwise be lost due to ROI. In this way, the rescued PXD clusters can be used in addition to the SVD information to achieve an efficient reconstruction, obtaining an all-silicon, 6 layer tracking. The main challenge lies in distinguishing hits made by slow pions from those made by background tracks, dominated by QED two photon processes. The method presented in this thesis uses artificial neural networks to discriminate slow pions from background particles, analysing only PXD cluster parameters.

In Chapter 2, the relevant physics for the Belle II experiment is presented, discussing discrete C, P, CPT symmetries, neutral meson mixing, CP violation and new physics. Chapter 3 introduces the Belle II experiment, describing the SuperKEKB collider, B meson production, flavor tagging and the Belle II detector in detail. Special emphasis is put on the PXD, which is the main detector in this thesis. In Chapter 4, the ROI algorithm and slow pions are discussed. In Chapter 5, the method for slow pion rescue is laid out, the PXD cluster variables are defined and compared between slow pions and background electrons. Also, the concept of neural networks is presented, showing how it was implemented to develop the cluster rescue algorithm. The obtained results and their analysis are presented in Chapter 6. Finally, Chapter 7 concludes this thesis and provides an outlook for the future.

2 Physics at Belle II

In this chapter, the relevant physics of the Belle II experiment is presented. This chapter is mainly based on the book ”Introduction to Elementary Particles” by Griffiths [6], ”Modern Particle Physics” by Thomson [7] and ”Flavor Physics at the Tevatron” by Kuhr [8].

2.1 Discrete Symmetries

In particle physics, the symmetry of a physical system refers to a characteristic of the system that is preserved when subjected to a specific transformation. Symmetries in nature are important because they provide insight into the underlying physical laws. This was proven by Emmy Noether in 1918 with her famous theorem in which she stated that every continuous symmetry of a physical system corresponds to a conservation law in that system [9]. Relevant discrete symmetries in particle physics include charge conjugation C, parity conjugation P and time conjugation T.

Charge Conjugation

The charge conjugation operator $\hat{C}$ acts on the charge of a particle while leaving the momentum and spin invariant. It inverts the particle's electromagnetic charge, baryon number, lepton number and flavor. The $\hat{C}$ operator transforms a particle in its anti-particle:

$$\hat{C}\,|p\rangle = |\bar{p}\rangle \tag{2.1}$$

Applying the charge conjugation operator twice restores the original state. Hence, its eigenvalues are ±1. The $+1$ is obtained when all inverted quantum numbers are zero and corresponds to a particle that is its own anti-particle, for example the photon. The -1 corresponds to particles that have anti-particles, for example the electron and the positron.

Parity Conjugation

Parity conjugation operator $\hat{P}$ acts on the spatial coordinates $\vec{r} = (x, y, z)$ by flipping the sign of the spatial vector. That is, it mirrors the system in space at the origin:

$$\hat{P}\vec{r} \rightarrow -\vec{r} \tag{2.2}$$

Just like in the case of charge conjugation, applying the $\hat{P}$ operator returns the initial system which results in eigenvalues ±1. The $+1$ corresponds to axial vectors, which are invariant under parity transformation. The -1 corresponds to polar vectors, which change their sign under parity transformation.

Time Reversal and CPT Theorem

The time conjugation operator $\hat{T}$ inverts t into $-t$, where t is the time of the system:

$$\hat{T}t \rightarrow -t \tag{2.3}$$

$\hat{T}$ reverses all vectors such as velocity $\vec{v}$, momentum $\vec{p}$, angular momentum $\vec{l}$, etc.

Microscopic systems can be invariant under time reversal, unlike macroscopic systems. However, due to the CPT theorem, it is expected for microscopic systems to violate time reversal symmetry.

The CPT theorem is one of the most important results of quantum field theory and it states that the combined operation of charge conjugation, parity and time reversal is an exact symmetry of any interaction. The CPT theorem concludes that every particle must have the same mass and lifetime as its anti-particle. Moreover, it also infers that if there is evidence for CP violation, there has to be a compensating violation of T.

2.2 Neutral Meson Mixing

Neutral mesons such as B^0, B_s^0, K^0 and D^0 can oscillate into their anti-particles and vice versa through the weak interaction, as shown in Figure 2.1. The mesons are composed of particular quark flavors, so they can be described in terms of flavor eigenstates. In the following, B mesons are used as example. The flavor eigenstates of B^0 and $\bar{B}^0$ are denoted as $|B\rangle$ and $|\bar{B}\rangle$ respectively. $|\bar{B}\rangle$ is obtained by applying a C conjugation to $|B\rangle$. Due to the mixing phenomenon, these two flavor eigenstates need to be described as a whole in order to properly represent a physical state. Such physical states are given by mass eigenstates, which are linear combinations of the flavor eigenstates. In other words, a meson state can be described as a superposition of flavor eigenstates

$$\begin{aligned} |B_L\rangle &= p\,|B\rangle + q\,|\bar{B}\rangle \\ |B_H\rangle &= p\,|B\rangle - q\,|\bar{B}\rangle\,, \end{aligned} \tag{2.4}$$

where p and q are complex numbers quantifying the degree of mixing. $|B_L\rangle$ and $|B_H\rangle$ are mass eigenstates with definite masses and lifetimes. Their difference in mass is used as an indication of the respective eigenstates: L and H stand for light and heavy. In a detector, these are the states being measured.

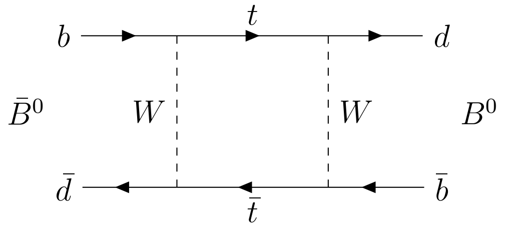


Figure 2.1: Box diagram for B^0-$\bar{B}^0$ mixing.

2.3 CP Violation and New Physics

The combination of the charge conjugation (C) and parity (P) symmetry gives rise to the charge conjugation parity (CP) symmetry. According to this symmetry, the laws of physics should not vary if a particle is exchanged with its anti-particle and its spatial coordinates are inverted. The violation of this symmetry is known as CP violation and it is defined as a difference between a process and the same process that has undergone a charge conjugation and parity transformation [8].

CP violation has only been observed in decays mediated by the weak interaction. No evidence of CP violation in strong nor electromagnetic interactions has been found.

2.3.1 Types of CP Violation

Three sources of CP violation are currently known. To explain them, a general decay of an X initial particle with final state f and decay amplitude A is considered. The anti-particle is referred to as $\bar{X}$ with final state $\bar{f}$ and decay amplitude $\bar{A}$.

Direct CP Violation

Direct CP violation is caused when a process decays at a different rate than its parity conjugated anti-process. In other words, their decay amplitudes are not symmetric and therefore their decay rates Γ, are different:

$$\Gamma(X \to f) \neq \Gamma(\bar{X} \to \bar{f}) \tag{2.5}$$

$$\left|\frac{A}{\bar{A}}\right| \neq 1 \tag{2.6}$$

Indirect CP Violation

Indirect CP violation is found for neutral, oscillating mesons if the rates of $B^0 \to \bar{B}^0$ do not equal $\bar{B}^0 \to B^0$. That is, if there is an asymmetry in the flavor oscillation observed experimentally in flavor specific decays, only accessible via mixing

$$\Gamma(X \to \bar{X} \to \bar{f}) \neq \Gamma(\bar{X} \to X \to f). \tag{2.7}$$

In theory, CP violation in mixing occurs when the mass eigenstates are not an equal mixture of their flavor eigenstates. Recalling Equation 2.4, indirect CP violation happens when the mixing terms are not equal

$$\left|\frac{q}{p}\right| \neq 1 \tag{2.8}$$

CP Violation in the Interference Between Mixing and Decay

This type of CP violation arises from the interference between decays with and without mixing going to the same final state f. The processes $X \to f$ and $X \to \bar{X} \to f$ are not distinguishable and therefore an interference of the amplitude of both processes occurs. The condition to observe CP violation in this case is given by

$$\mathrm{Im}\left\{\frac{\bar{A}}{A}\frac{q}{p}\right\} \neq 0 \tag{2.9}$$

2.3.2 CKM Matrix

CP violation has only been observed in interactions where quarks undergo flavor changing transitions, which happens only via the weak force. This has been integrated in the Standard Model by the Cabibbo-Kobayashi-Maskawa framework, formulated in 1973 [10].

The quark mass and flavor eigenstates are different and both bases can be related through the Cabibbo–Kobayashi–Maskawa (CKM) matrix. This is a 3 × 3 unitary matrix that describes the probability of quark flavor changing interactions and it is defined as:

$$V_{CKM} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}, \tag{2.10}$$

where each element of the CKM matrix is a complex number which gives the transition probability between quarks [7]. The weak eigenstates (d', s', b') are related to the mass eigenstates (d, s, b) by:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}. \tag{2.11}$$

Therefore, the matrix has 18 degrees of freedom. Due to the unitary condition and absorbing physically irrelevant phases in the quark fields, the matrix can be reduced to 4 free parameters: three mixing angles between the three quark generations θ_{12}, θ_{23}, θ_{13} and one complex phase δ. When the complex phase is non-vanishing, CP violation is observed in quark transitions [11]. These parameters are not predicted by theory and therefore must be measured experimentally.

The CKM matrix can be written in the Wolfenstein parametrization [12]

$$V_{CKM} = \begin{pmatrix} 1 - \frac{\lambda^2}{2} & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{\lambda^2}{2} & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} + \mathcal{O}(\lambda^4). \tag{2.12}$$

A, ρ and η are real parameters of order unity. λ is defined as $|V_{us}| \approx 0.22$. Diagonal terms are of the order one, corresponding to transitions of quarks within quark generation. Transitions between first and second generation (V_{td} and V_{us}) are of order λ, transitions between second and third generation (V_{cb} and V_{ts}) are of the order λ^2 and transitions between first and third generation (V_{ub} and V_{td}) are of the order λ^3.

The Unitary Triangle

The unitary condition of the CKM matrix infers a six constrains, including the following:

$$V_{ud}\ V^*_{us} + V_{cd}\ V^*_{cs} + V_{td}\ V^*_{ts} = 0 \tag{2.13}$$

$$V_{ud}\ V^*_{ub} + V_{cd}\ V^*_{cb} + V_{td}\ V^*_{tb} = 0 \tag{2.14}$$

$$V_{us}\ V^*_{ub} + V_{cs}\ V^*_{cb} + V_{ts}\ V^*_{tb} = 0. \tag{2.15}$$

These relations can be represented by the so-called unitary triangles in the complex plane. Unlike the terms in Equation 2.13 and 2.15, the terms in Equation 2.14 are of the same order ($\mathcal{O}(\lambda^3)$). Consequently, the corresponding triangle will have sides of similar length, as shown in Figure 2.2, and the angles will differ from zero. Note that the displayed triangle has been normalized by $V_{cd}V^*_{cb}$ to fix the lower side to the real axis with a length of 1. The new coordinates are given by

$$\bar{\rho} = \rho\left(1 - \frac{\lambda^2}{2}\right) \qquad \bar{\eta} = \eta\left(1 - \frac{\lambda^2}{2}\right). \tag{2.16}$$

The angles of the triangle are

$$\alpha = arg\left(-\frac{V_{td}V^*_{tb}}{V_{ud}V^*_{ub}}\right) \qquad \beta = arg\left(-\frac{V_{cd}V^*_{cb}}{V_{td}V^*_{tb}}\right) \qquad \gamma = arg\left(-\frac{V_{ud}V^*_{ub}}{V_{cd}V^*_{cb}}\right). \tag{2.17}$$

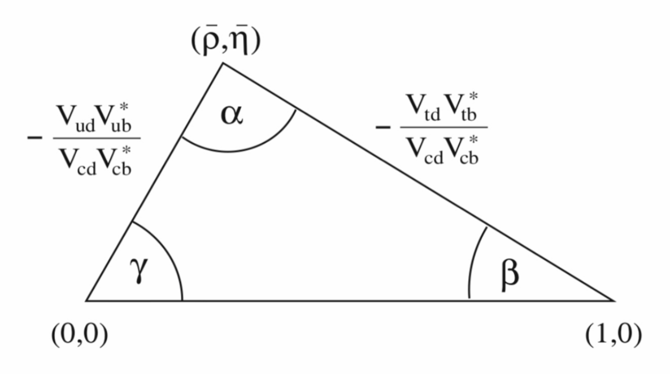

$(\bar{\rho},\bar{\eta})$
$-\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}$
$-\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}$
α
γ
β
(0,0)
(1,0)

Figure 2.2: Unitary triangle for Equation 2.14 normalized to $V_{cd}V_{cb}^*$. The presence of large angles in the unitary triangle means large phase differences between the involved CKM matrix elements which correspond to large CP violating effects. Taken from [8].

Measuring the angles of the unitary triangle is important for verifying the SM, constraining the CKM matrix elements and searching for new physics. If the sides and angles are measured to be such that the triangle is not closed, then this would point into the direction of new physics. Moreover, this triangle is particularly interesting because it is built from matrix elements describing $b \to d$ transitions, which occur in the mixing of $B^0 - \bar{B}^0$ mesons. Therefore, studying B decays are useful for constraining the CKM matrix elements and thus, the search for new physics.

2.3.3 New Physics

With the CKM matrix, CP violation was incorporated into the SM. This is an important mechanism that explains the evolution of a matter-dominated universe. However, what the SM predicts does not account for the observed matter-antimatter asymmetry in the universe. The difference is orders of magnitude too small. Therefore, there must be another source of CP violation beyond the SM still to be discovered.

Neutral B mesons are the neutral meson system investigated at the Belle II experiment with the purpose of making precise measurements to test the quark mixing and CP violation described by the CKM framework and find new physics beyond the SM.

3 The Belle II Experiment

The Belle II experiment is a particle physics experiment located at the High Energy Accelerator Research Organization (KEK) in Tsukuba, Japan. Through the production of B mesons, the Belle II aims at measuring with high precision weak interaction parameters and to look for New Physics (NP). B and $\bar{B}$ decays are of interest for several reasons. First, they allow for measurements of CKM matrix elements. In addition, physics processes beyond the SM are sensitive to rare B meson decays. Lastly, mixing-induced CP violation can be studied [13].

3.1 SuperKEKB

The SuperKEKB is an asymmetry-energy and double-ring particle collider with the purpose of producing B meson pairs from electron-positron collisions. Specifically, electrons and positrons are accelerated to 7 GeV and 4 GeV respectively and stored in separate rings. They collide at the Interaction Point (IP), where the Belle II experiment is located. The electron and positron bunches collide at an angle, known as crossing angle. The asymmetric beam energies produce a boost along the beam axis which makes the separate measurement of decaying B meson vertices possible [2]. This is critical for time-dependent CP violation measurements. The beam energies can be varied to values around a center-of mass energy of 10.58 GeV, which is the Υ(4S) resonance, as shown in Figure 3.1. This is a flavorless meson state which decays 96% of the time into an $B\bar{B}$ pair, since the mass of one B meson is 5.28 GeV [14]. In addition, Υ(4S) decays almost in equal parts into an entangled neutral or a charged B meson pair. Due to the high rates of B meson production, the Belle II experiment is referred to as a "super B-Factory experiment".

At SuperKEKB, e^+ and e^- are accelerated in the linear accelerator (linac). The electrons are produced through the photoelectric effect using a Ytterbium-doped laser to hit an Ir_5Ce photo-cathode. Moreover, the positrons are produced by accelerating electrons and making them collide with a tungsten target. This generates bremsstrahlung radiation which then converts into e^+e^- pairs. The electrons are then injected in the High Energy Ring (HER) and positrons are passed through a damping ring (to reduce their emittance to the required level) [15] before storing them in the Low Enery Ring (LER). A sketch of SuperKEKB is shown in Figure 3.2.

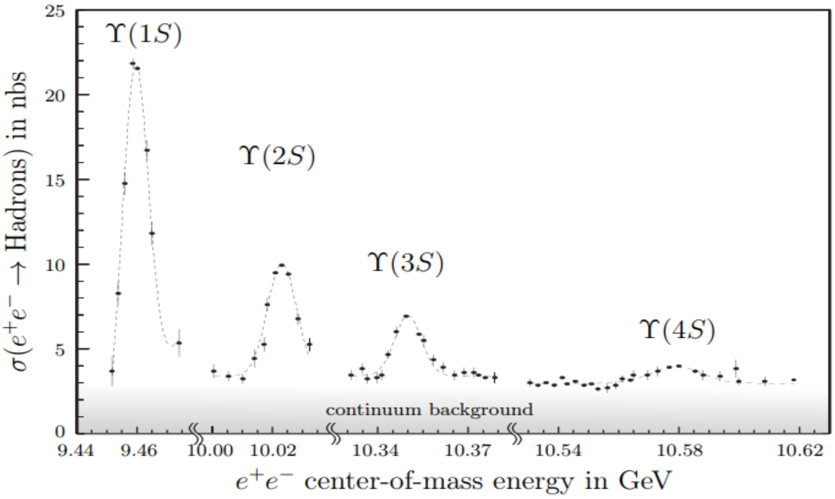


Figure 3.1: e^+e^- hadronic cross section as a function of e^+e^- center-of-mass. Taken from [11].

.

This accelerator is the upgraded version of its predecessor, KEKB. SuperKEKB aims to reach a peak luminosity of $6{\times}10^{35}$ cm^{-2}s^{-1}, which is 30 times higher than the peak luminosity reached by KEKB ($2.11{\times}10^{34}$) [16]. To achieve the increase in luminosity, a nano-beam scheme was adopted. This refers to an increase in the beam currents by a factor of 2 compared to KEKB, as well as a beam size reduction by a factor of 20 at the IP, from 1 μm to 50 nm [15]. Applying this scheme requires the final focus magnets to be closely placed at the IP, enlarging the beam crossing angle by a factor of 4.

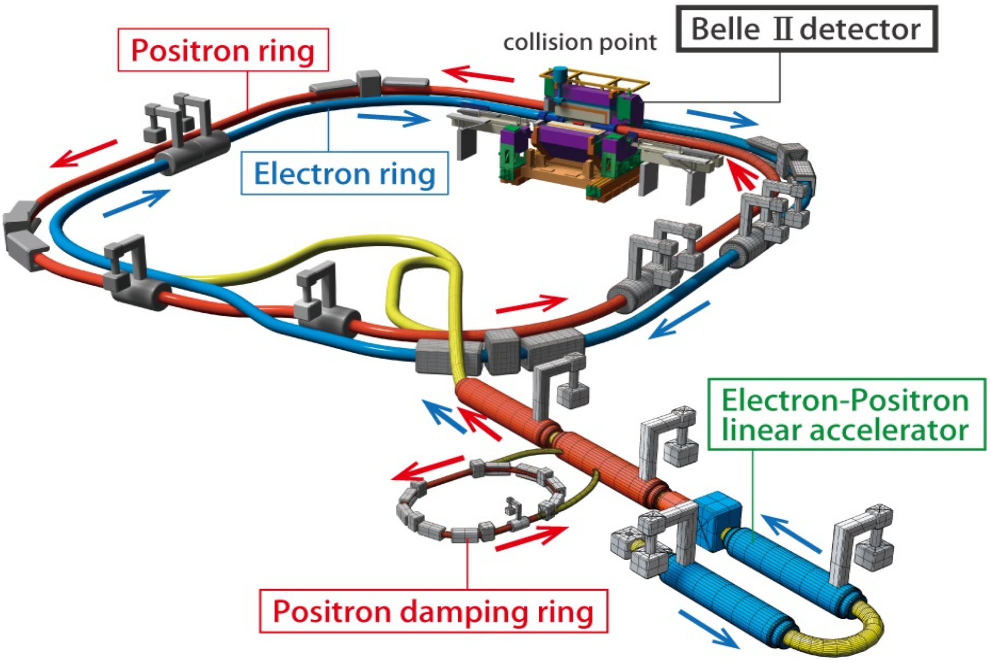


Figure 3.2: SuperKEKB electron-positron collider. It is composed of the electron ring (HER), the positron ring (LER), the linac for particle injection and the positron damping ring. Taken from [17].

3.2 The Belle II Detector

The Belle II detector is composed of a series of sub-detectors arranged in cylindrical layers around the IP, enclosing it almost hermetically [15]. Each sub-detector has a specific measurement purpose which are combined to identify the decay products coming from the particle collisions. Furthermore, the detector can be classified into an inner and outer part [18]. The inner part focuses on precise measurements of the position and momenta of charged particles to reconstruct their tracks, as well as to identify primary and secondary vertices. On the other hand, the outer part deals with particle identification and energy measurement. The inner part is composed of the Vertex Detector (VXD) (including the Pixel Detector (PXD) and the Silicon Vertex Detector (SVD)), the Central Drift Chamber (CDC), the Time Of Propagation (TOP) and the Aerogel Ring-Imaging Cherenkov detector (ARICH). The outer part consists of the Electromagnetic Calorimeter (ECL) and the long-lived Kaon (K_L) and Muon detector (KLM). In between the outer and inner detector region, the solenoid magnet is found. Moreover, the sub-detectors can also be classified by their primary tasks: tracking is mainly done by the VXD and CDC, particle identification is done by TOP and ARICH and calorimetry is performed by the ECL and TOP. Figure 3.3 shows a schematic view of the Belle II detector.

The origin of the Belle II coordinate system is centered at the IP. The $x-$axis is horizontal and points away from the accelerator, the $y-$axis is vertical and points upwards and the $z-$axis is the Belle II solenoid axis and points roughly along the electron beam. Moreover, ϕ is the azimuthal angle and θ is the zenith angle [19].

In the following sections, each sub-detector is briefly discussed. These sections are based on the "Belle II Technical Design Report" [18].

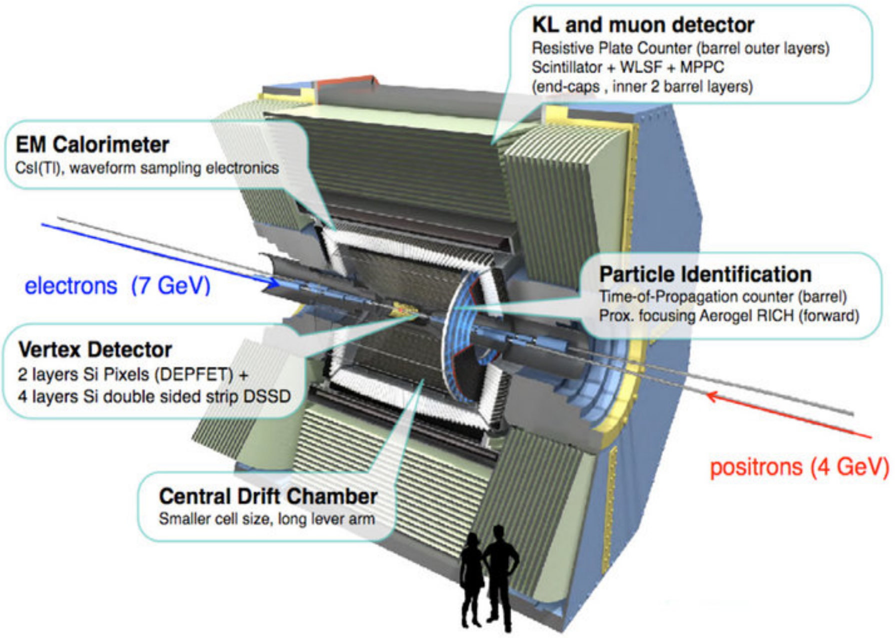
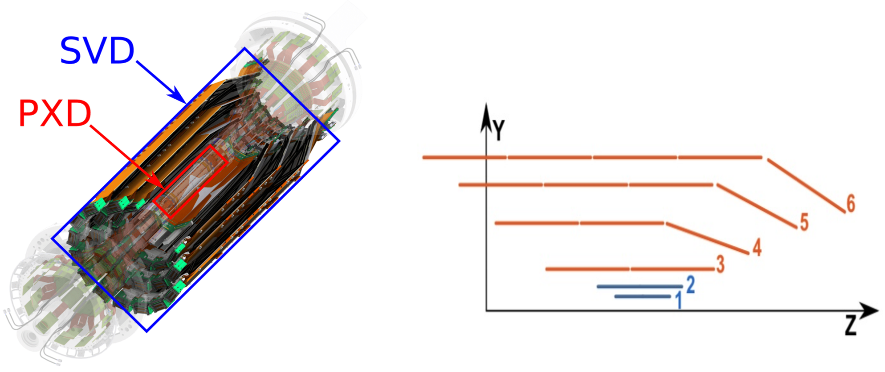

KL and muon detector
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC
(end-caps , inner 2 barrel layers)
EM Calorimeter
CsI(Tl), waveform sampling electronics
electrons (7 GeV)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (forward)
Vertex Detector
2 layers Si Pixels (DEPFET) +
4 layers Si double sided strip DSSD
positrons (4 GeV)
Central Drift Chamber
Smaller cell size, long lever arm

Figure 3.3: The Belle II detector. Taken from [20].

3.2.1 Vertex Detector

The innermost sub-detector of Belle II is the VXD. It has six layers surrounding the beam pipe, two of pixel sensors and four of silicon strip detectors, known as the PXD and SVD respectively. Pixel sensors are needed at the innermost layers to cope with the high background. The polar acceptance region is $17° < \theta < 150°$ and it has a 2π

azimuthal coverage. The purpose of the VXD is to measure the position of charged particles traveling through the detector layers in order to precisely reconstruct their tracks. It is capable of reconstructing low transverse momentum tracks in the order of 30 MeV. Combining the tracking information with the CDC, the origin or decay vertex position are determined through track extrapolation. To minimize extrapolation errors, the first layer of the PXD is positioned as close as possible to the IP. Additionally, the VXD is able to reconstruct the momenta of charged particles, which trajectories are curved in the solenoid field. The VXD is illustrated in Figure 3.4. The work developed in this thesis is based on the PXD and therefore, this sub-detector will be described in more detail.



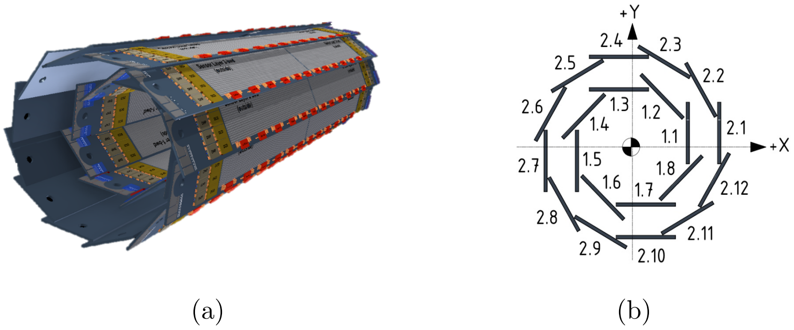
(a) (b)

Figure 3.4: (a) The Belle II VXD sub-detector composed of the PXD and SVD surrounding the beam pipe. Taken from [21]. (b) Structure of the VXD along the z-direction. The blue layers correspond to the PXD whereas the red layers signal the SVD. Taken from [22].

PXD

The PXD consists of 2 layers surrounding the IP, as shown in Figures 3.5. The layers are made of ladders, which are in turn made of two pixel modules glued together based on Depleted Field Effect Transistor (DEPFET) technology. The first (or inner) layer has 8 ladders, whereas the second (or outer) layer consists of 12 ladders [11]. In total, the PXD is equipped with 40 modules. The inner and outer layer is 14 mm and 22 mm away from the IP, respectively.

A module has a sensitive area containing 250×768 pixels surrounded by 14 Application-Specific Integrated Circuits (ASICs), which perform the DEPFET read-outs. To meet both high resolution and fast read-out requirements, the sensitive part has 2 pixel regions with different pixel sizes: smaller pixels close to the IP and bigger pixels in the outer regions. In addition, the dimensions of the modules in the inner and outer layer differ. The respective size and thickness of the pixels were chosen to optimize the reconstruction of decay vertices and position resolution. Figure 3.6 illustrates the module dimensions for layer 1 and layer 2 as well as the pixel regions. In total, the PXD counts with 7,680,000 pixels with a read-out time of 20 μs.



(a)

(b)

Figure 3.5: (a) Arrangement of the PXD around the beam pipe. Taken from [23]. (b) Naming scheme of the detector layers and ladders. Inner and outer layer are referred to as 1 and 2 respectively. The ladder number goes from 1-8 in layer 1 and from 1-12 in layer 2. Taken from [22].

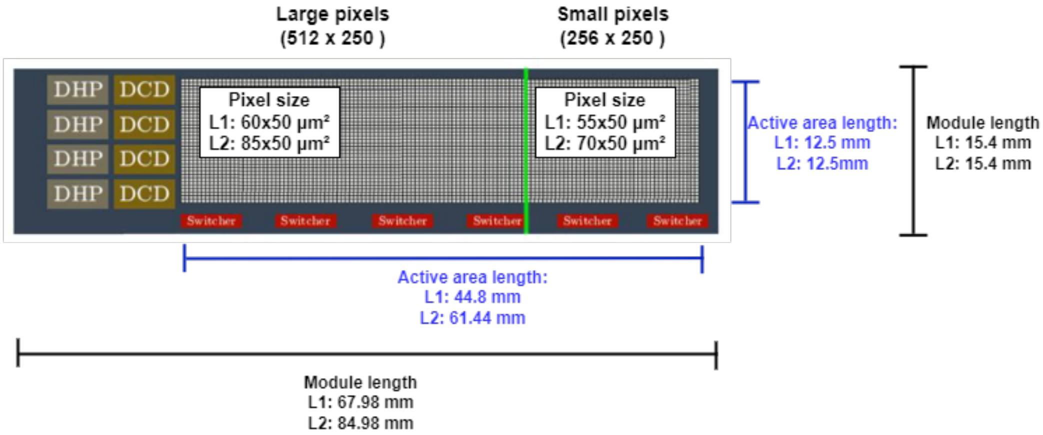


Figure 3.6: PXD module with its respective dimensions for layer 1 (L1) and layer 2 (L2). The modules in layer 2 are slightly bigger along the V direction.

A charged particle traversing the PXD hits one or more pixels in a module, producing electrons in the sensor that can be read as a signal. Only the pixels with a signal 3 times bigger than the accepted noise level (200 electrons) are considered for analysis [24]. Such pixels are grouped into pixel clusters by a so-called clusteriser algorithm [25]. The clustering is performed in the following way: the pixels are checked in a row-wise manner. When a pixel above the defined threshold is found, the pixel to the left and the direct neighbours above (in the previous row) are checked. If those pixels already belong to a cluster, the pixel is added to it. If nothing is found, then a new cluster is created, being this pixel its first member. This method is repeated until all pixels in a module have been processed [24]. Figure 3.7 gives a simplified view of the pixel clusters in a module.

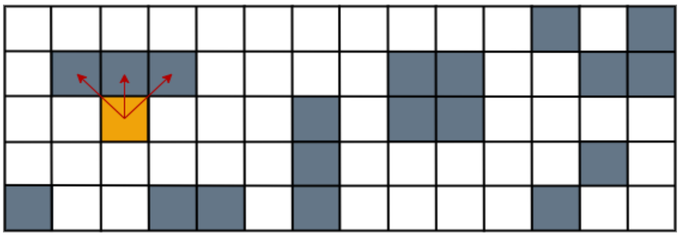


Figure 3.7: Sketch of PXD clusters in a module. Clusters can consist of one or more pixels. The red arrows indicate the direct neighbours of the highlighted cluster.

To quantify and study the PXD outputs, a local coordinate system (U, V) has been defined. V runs along the long side of a module, parallel to the detector's $z-$direction and U runs along the short side of a module. In this way, properties of the clusters such as position, number of pixels etc. can be defined. This is explained in detail in Section 5.1, where the specific cluster parameters are discussed.

The purpose of the PXD is to measure the position of passing particles with high precision. In doing so, the vertex of the produced particles is reconstructed, which is important for the measurement of mixing-induced CP asymmetry. In addition, the proximity of the inner-most layers of the Belle II detector to the IP allows for a higher position measurement resolution but will consequently detect more background hits. This is because background hits are inversely proportional to the square of the detector distance to the IP. For this reason, the inner-most layers have been chosen to be made of pixel sensors: they have a large number of channels which allow for a higher hit rate due to smaller occupancy.

SVD

The SVD is composed of four layers of Double-Sided Silicon micro-strip Detectors (DSSDs) with 38 mm, 80 mm, 104 mm and 135 mm radii. The inner-most layer of the SVD (number 3 in Figure 3.4) is composed of rectangular sensors, whereas the rest (4 , 5 and 6 in Figure 3.4) are composed of rectangular and slanted sensors. Each sensor is based on a silicon bulk with $N-$ and $P-$doped strips at the top and bottom side. The strips are positioned orthogonal to each other. A passing particle produces electron-hole pairs in the sensor, with the electrons and holes drifting to the $n-$side and $p-$side strips respectively, so that, when the strips are read-out, the x and y coordinates of the hit are found and a 2D track reconstruction can be made [26].

As mentioned before, its main task is to measure the decay vertex position and track reconstruction along with the CDC as well as momenta measurement. The reconstructed tracks are extrapolated to the PXD. In combination with the PXD it is also able to reconstruct low transverse momentum tracks, down to about 10 MeV.

3.2.2 Central Drift Chamber

The CDC is a cylinder of inner radius of 160 mm and outer radius of 1130 mm filled with an equal mixture of helium and ethane gas. The chamber is composed of 14,336 sense wires and 42,240 field wires arranged in 56 cylindrical wire layers. The layers alternate between an axial (aligned with the magnetic field) and stereo (tilted relative to the axial wires) orientation. This design allows for a full 3D helix track reconstruction. In addition, 8 field wires surround a sense wire, forming a so-called drift cell. The field and sense wires have opposite voltages, creating an electric field that makes the electrons drift in the cell. In order to withstand high background levels and high event rates, the size of the cells is about 1.8 cm × 1.8 cm.

This detector is an essential part of Belle II. It contributes to particle identification by measuring energy loss in the gas volume as they pass through the detector. Passing particles ionize the gas, letting electrons drift towards the sense wires, further ionizing the gas as they travel. From the pulse height of the measured signal, the energy loss can be obtained. Moreover, the CDC reconstructs the tracks of charged particles and measures their momenta with high precision. This is done by measuring the drift time to get information about the track position with a spatial point resolution of $\sim$100 μm. From this, the tracks are reconstructed by fitting a helix through the sense wires that were triggered. Lastly, the CDC also serves as an efficient trigger for charged particles.

3.2.3 Particle Identification (PID) System

The PID system consists of two components: a Time of Propagation (TOP) detector and an Aerogel Ring Imaging Cerenkov (ARICH) detector.

The TOP detector is used at the barrel region, which is the cylindrical section surrounding the IP, and the ARICH at the forward end-cap region, which is the section at an end of the barrel region perpendicular to the beam-pipe. Both detectors work using the fact that particles traveling inside a medium with refractive index n with speed greater than the speed of light in that medium will emit Cherenkov light. However, due to their different locations, their working principle differs. The working principle of the TOP and ARICH detectors are illustrated in Figures 3.8 and 3.9 respectively.

The TOP is made of 16 modules surrounding the CDC. Each is made of two 2.7 m long quartz bars with photo-detectors attached at the end. When passing through radiator material, charged particles emit Cherenkov photons at a specific angle θ_c, which depends on the particle's velocity. Depending on θ_c, the emitted photons will travel different paths through internal reflections in the quartz until the photo-detectors are reached. Here the time of flight is measured with a resolution of 100 ps. Combining this information with momentum measurements, the mass of the particle is extracted and thus, the particle is identified. In this way, charged pions can be distinguished from charged kaons.

On the other hand, the ARICH is a proximity focusing Cherenkov ring imaging detector. It consists of two layers of aerogel with different refractive indices and separated by an expansion volume. Charged particles crossing the aerogel medium emit Cherenkov photons, producing a Cherenkov cone which is detected by photon-detectors. The radius of the ring is measured to extract PID information. ARICH is able to distinguish pions from kaons in a momentum range of up to 4 GeV. It is also able to distinguish between pions, muons and electrons up to 1 GeV.

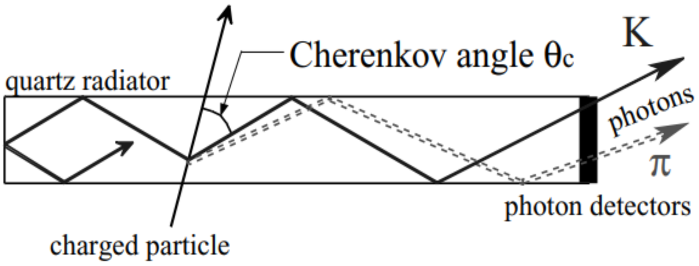


Figure 3.8: Schematic view of the TOP detector. Taken from [18].

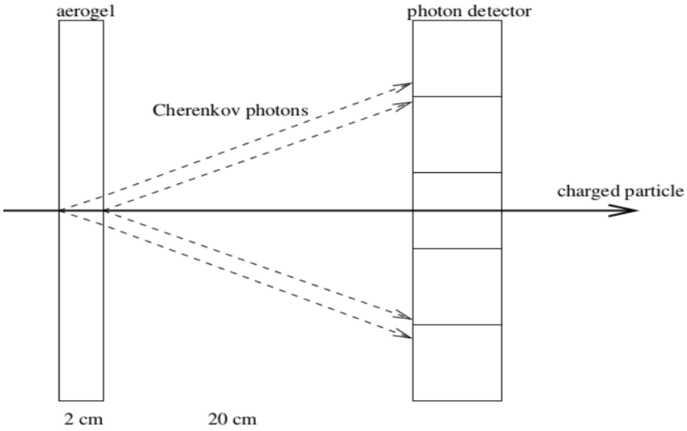


Figure 3.9: Schematic view of the ARICH detector. Taken from [18].

3.2.4 Electromagnetic Calorimeter

The ECL is used for the detection of photons and electrons. This sub-detector is able to detect photons over a wide energy range (20 MeV to 4 GeV) with a high efficiency and measure their energy and position. This measurements are indispensable to efficiently reconstruct neutral particles coming from B mesons which have further decayed into photons. In addition, the ECL allows for the separation of electrons from hadrons, particularly useful with pions. Other tasks include generation of signals for triggering, online and offline luminosity measurements and detection of K_L along with the KLM [27].

The calorimeter consists of a cylindrical container with a length of 3 m and a radius of 1.25 m. It contains a highly segmented array of thallium-doped caesium iodide CsI(Tl) crystals. The number of crystals in the barrel section is 6,624. As particles pass through the ECL, they will interact with the electrically charged particles in the material and produce a shower. As a result, the particles emitted by the shower generate scintillation photons in the CsI(Tl) crystals. Silicon photodiodes are attached to the backside of the crystals for scintillation light read-out.

3.2.5 Solenoid Magnet

A superconducting solenoid with a diameter of 3.4 m and a length of 4.4 m surrounds the ECL. It creates a magnetic field of 1.5 T which causes the charged particles to move in bent trajectories [15]. This allows for momentum measurements using the tracking information of the sub-detectors.

3.2.6 Kaon and Muon Detector

The most external sub-detector is the KLM, with the objective of identifying long-lived kaon mesons, K_L, and muons. The detector is formed by a barrel region, aligned parallel to the beam, and a forward and backward end-cap regions, aligned normal to the beam. These regions are made of alternating layers of 4.7 cm-thick iron plates and active detector elements. Specifically, 15 detector layers and 14 iron plates in the barrel region and 14 detector layers and 14 iron plates in the end-cap regions. The metal plates serve simultaneously as the magnetic flux return yoke for the magnetic field and as material to decelerate particles and generate hadronic showers. In total, the iron plates give 3.9 interaction lengths of material for a traveling particle. Moreover, the active detector components consist of scintillator strips in the end-caps and in the two inner-most layers of the barrel region and glass-electrode resistive plate chambers (RPCs) in the outer layers of the barrel.

A track in the KLM can be attributed to a passing muon if it is associated with a reconstructed charged track in the CDC. Compared to strongly interacting hadrons, muons on average exhibit significantly greater travel distances through the detector with smaller deflections. This allows for separation between muons and charged hadrons. In addition, since K_L^0 produces hadronic showers when passing through the KLM and the ECL, it can be identified by matching these two features. From this, the direction and energy of the neutral kaons can be determined.

3.3 Flavor Tagging

Neutral B meson systems are important for the search of mixing-induced CP violation (indirect CP violation). The study of $B^0 - \bar{B}^0$ processes allows to constrain the CKM angles, as well as to measure anomalies that could lead to new physics beyond the Standard Model [4]. To measure CP violation in neutral B decays, the flavor of the decaying B mesons must be known. In general, the flavor of a B meson can not be deduced from its final products. Instead, the entanglement of the neutral meson system can be exploited using the flavor tagging algorithm.

A $B^0 - \bar{B}^0$ pair is produced in an entangled state and as they propagate, they oscillate until one of them decays into a flavor-specific final state. It is not possible for the two B mesons to have the same flavor simultaneously. Thus, if the flavor of one of the B mesons is determined at the time of decay, the other meson's flavor can be inferred. This is done using the flavor tagging algorithm, which uses the final decay products of "the other" B meson to determine the meson's quark-flavor content at the time of decay. This requires a full event reconstruction of one B meson, referred to as the signal side, and the determination of the quark-flavor content of the other meson, referred to as the tag side. In summary, the signal B meson is reconstructed in a flavor unspecific CP eigenstate. This means that the flavor of the B meson can not be inferred from the decay. An example of this would be $B^0 \to \pi^+ \pi^-$. The remaining final states of an event are then fed into the flavor tagging algorithm to determine the flavor of the tag side.

As an example, let us assume a neutral B meson decays at time t_1 into a particular flavor-specific decay mode. To determine the quark-flavor of such meson, the charge of the final decay products are determined (tagging). Lets us assume the B meson is identified to be a $\bar{B}^0$. This means that at time t_1 the other$B$ meson must be a B^0. This meson then decays at time t_2 into a flavor unspecific CP eigenstate. This is illustrated in Figure 3.10.

In addition to the flavor tagging information, the decay time difference between the two mesons $\Delta t = t_1 - t_2$ needs to be determined in order to find signatures of indirect CP violation. The decay time difference can be related to the difference in decay lengths along the z-axis, Δz, which can be measured in the Belle II detector:

$$\Delta t = \frac{\Delta z}{\beta \gamma c}, \tag{3.1}$$

where β is the system's boost, γ is the Lorentz factor and c is the speed of light. Hence, by measuring Δz, Δt can be determined, which is a parameter that describes CP asymmetries.

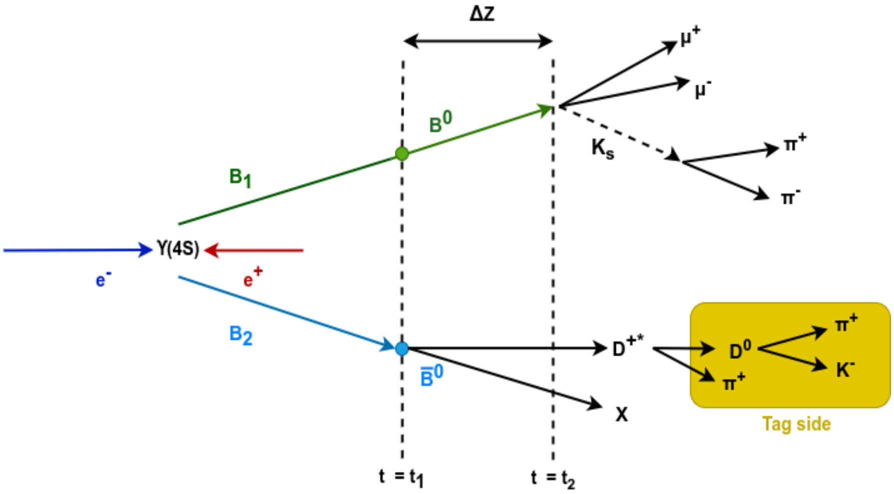


Figure 3.10: B meson production at the $\Upsilon(4S)$ resonance. B_1 and B_2 oscillate as they travel until B_2 decays at $t = t_1$ into a D^* and an unspecified state X. The D^* decays further into a slow pion and a D^0 which further decays into a charged π and charged K. The charge of the D^* can be determined from the charge of the slow pion and the decay products of D^0. In turn, the flavor of B_2 is determined at the time of decay to be $\bar{B}^0$. On the signal side, B_1 decays at $t = t_2$ into final states that give no information about its flavor. With the tag side information, it is inferred that B_1 must be a B^0. The difference in decay lengths Δz is then used to find the decay time difference, which is a parameter needed to study CP violation in the neutral meson system.

4 The PXD and Slow Pions

4.1 Region Of Interest (ROI)

With the planned increase in luminosity, the generated data set is expected to increase by a factor of 50 [28]. However, some background processes are luminosity dependent and will increase by the same amount, such as two-photon QED processes [29], shown in Figure 4.1. The fermions generated by this process are mainly low momentum electrons, affecting the PXD due to its proximity to the IP. At full luminosity, the PXD data size will exceed the rest of Belle II detectors by a factor of 10 and the PXD occupancy is estimated to be 3% [30]. For efficiency and offline storage reasons, the PXD data must be reduced online. A method to achieve this is the so-called Region Of Interest (ROI) algorithm. ROI uses hits from the SVD and other sub-detectors to perform an online event reconstruction. The reconstructed track is then extrapolated to the PXD to define "regions of interest". The fraction of pixels selected by ROI are permanently stored [21] and the rest is discarded.

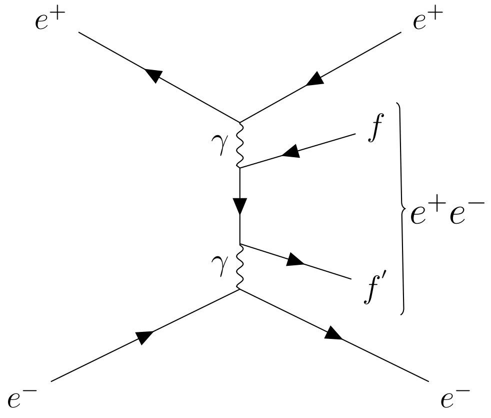


Figure 4.1: Two-photon QED process, the main source of background in the PXD. A positron-electron pair have a high probability to be produced due to a high cross-section. The produced positron-electron pairs have low momentum and therefore, they spiral through the detector.

The PXD read-out scheme is presented in Figure 4.2 and works in the following way: PXD data stored in the ASICs modules is read-out by a Data Handling Hybrid (DHH) after a hardware trigger. A DHH Controller (DHHC) is in charge of cluster reconstruction and classification. The data is then sent to the Online Selection Nodes (ONSEN) system, where it gets reduced. (ONSEN) saves only the clusters inside the region of interest, neglecting detector noise and background hits. Finally, ONSEN sends the data to the event builder system. Additionally, the High-Level Trigger (HLT) and the Data Concentrator (DATCON) find and select the ROIs by reconstructing and extrapolating tracks [30]. To perform an online track reconstruction, the HLT uses sub-events created with data from the sub-detectors, whereas DATCON uses data from the SVD [21].

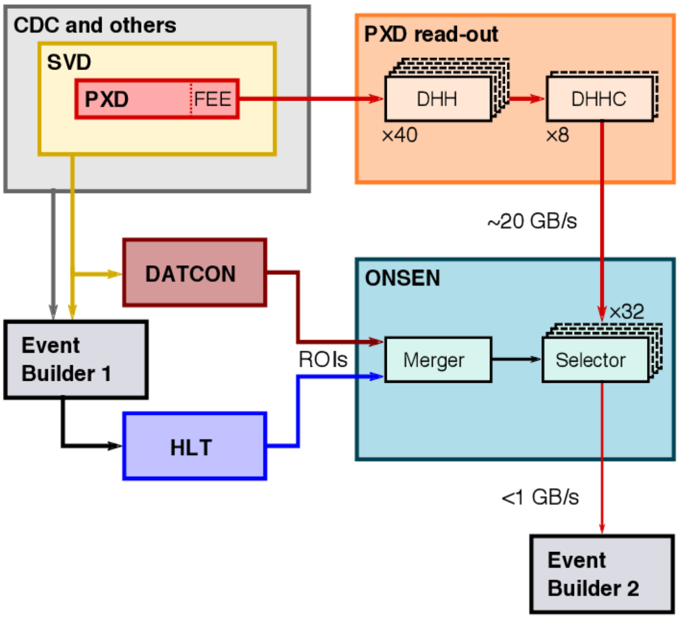


Figure 4.2: PXD read-out scheme. ROIs are selected by DATCON and HLT and then passed to ONSEN, where the PXD data is reduced. Taken from [30].

4.2 Slow Pions

A key process observed at Belle II is the semi-leptonic decay of B mesons into orbitally excited D mesons, D^*. The full decay chains are:

$$B^0 \to D^{*-} + X \tag{4.1}$$
$$\hookrightarrow \bar{D}^0\pi^-$$

$$\bar{B}^0 \to D^{*+} + X \tag{4.2}$$
$$\hookrightarrow D^0\pi^+,$$

where X is any possible final state, including lepton pairs and/or hadrons. The pions coming from the D^* are known as ”slow pions” due to their low transverse momentum, shown in Figure 4.3. They are generated nearly at rest in the D^* frame and therefore, the slow pion and the D^0 follow the D^* direction. This decay is important because the charge of the slow pion can be used for flavor tagging and determining the flavor of B^0 mesons.

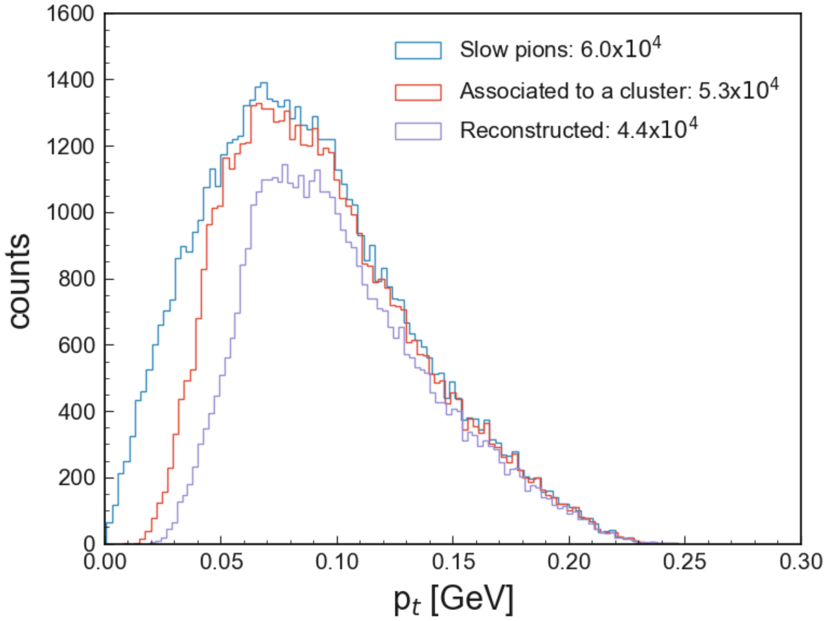


Figure 4.3: Transverse momentum distribution of slow pions: all slow pions in the event, those reconstructed and those which the reconstructed tracks are associated to a PXD cluster. The spectra shown corresponds to the reactions from 4.1 and 4.2 with X going into $\bar{l}\nu$ and $l\bar{\nu}$ respectively.

Due to their low momentum, slow pions tend to not reach the CDC [28]. A fraction of them do not even get to traverse the four layers of the SVD, which is problematic since track reconstruction is done online mainly by this detector. The SVD efficiently reconstructs the tracks of particles with momentum greater than 100 MeV [15]. Below this threshold, reconstruction becomes inefficient or no track reconstruction is done. From Figure 4.3, it can be seen that a considerable fraction of slow pions have $p_t < 100$ MeV, meaning that a fraction of slow pions will not be reconstructed. This poses a problem when the ROI algorithm is integrated, as the PXD clusters from those slow pions would be discarded.

To estimate the fraction of lost slow pions, a specific Monte-Carlo (MC) sample is considered. This sample was generated according to the reactions shown in Equations 4.1 and 4.2, having a lepton and a neutrino as X. That is

$$\begin{aligned} B^0 &\rightarrow D^{*-}\bar{l}\nu \\ \bar{B}^0 &\rightarrow D^{*+}l\bar{\nu}. \end{aligned} \tag{4.3}$$

Hence, all plots shown are from this reaction. 88.5 % of the events are associated to a PXD cluster. This is because some slow pions have an emission angle outside of the PXD acceptance region or due to PXD inefficiencies (see Figure 4.4). Also, 26.7% of the total sample is not reconstructed and would be lost due to ROI. Considering only the slow pions associated to a cluster, around 83% are reconstructed, meaning that 17% of slow pions would be lost. It is important to note that this is a lower limit since there are many B decays with D^* and additional hadrons. Therefore, the D^* and π spectra are even broader.

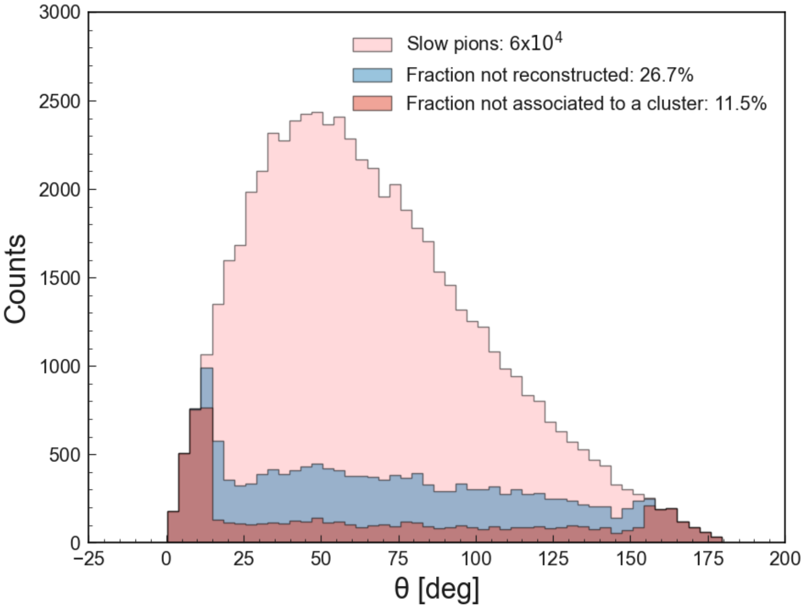


Figure 4.4: Emission angle distribution of all slow pions, the fraction that is reconstructed and the fraction that is not associated to a cluster. Most lost slow pions are outside the PXD acceptance region ($17° < \theta < 150°$). In addition, there is a constant fraction throughout the distribution that is lost due to detector inefficiencies.

5 Slow Pion Rescue Procedure

An alternative method is necessary to save the slow pions that otherwise would be lost with the implementation of ROI. This is a challenging task because, as mentioned before, PXD hits are dominated by QED background. More specifically, it is estimated that at full design luminosity (6×10^{35} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$), there will be 2000 electrons in the sensor per 1 slow pion [31]. To reduce this background significantly, we propose a PXD stand-alone cluster rescue mechanism to identify slow pion clusters against background without help from other detectors. In this way, the rescued PXD clusters can be used offline later on with SVD clusters to achieve an efficient reconstruction. To do this, a specific electron background MC sample is also considered. The produced reaction is:

$$e^+e^- \to e^+e^-e^+e^-. \tag{5.1}$$

We developed a neural network pattern recognition algorithm which uses the so-called PXD cluster variables as input parameters and estimates the probability of the cluster coming from a slow pion or electron. The details of the development of the algorithm are shown in Section 5.3.

The neural network cluster rescue is planned to be implemented in hardware for online data reduction. The algorithm would be executed on DHH, which then sends those clusters identified as slow pions to ONSEN. The recovered data can then be analysed offline along with SVD hits to help with low momentum track finding.

5.1 PXD Cluster Variables

The PXD clusters can be studied by defining variables which describe properties of the cluster such as size, position, charge, etc. The full list of considered variables is shown in Table 5.1. Figures 5.1 and 5.2 show the distributions for the slow pion and QED electrons cluster variables in layer 1 and layer 2 respectively. The MC generated data of $B\bar{B}$ events and electron background were used.

Table 5.1: List of cluster variables.

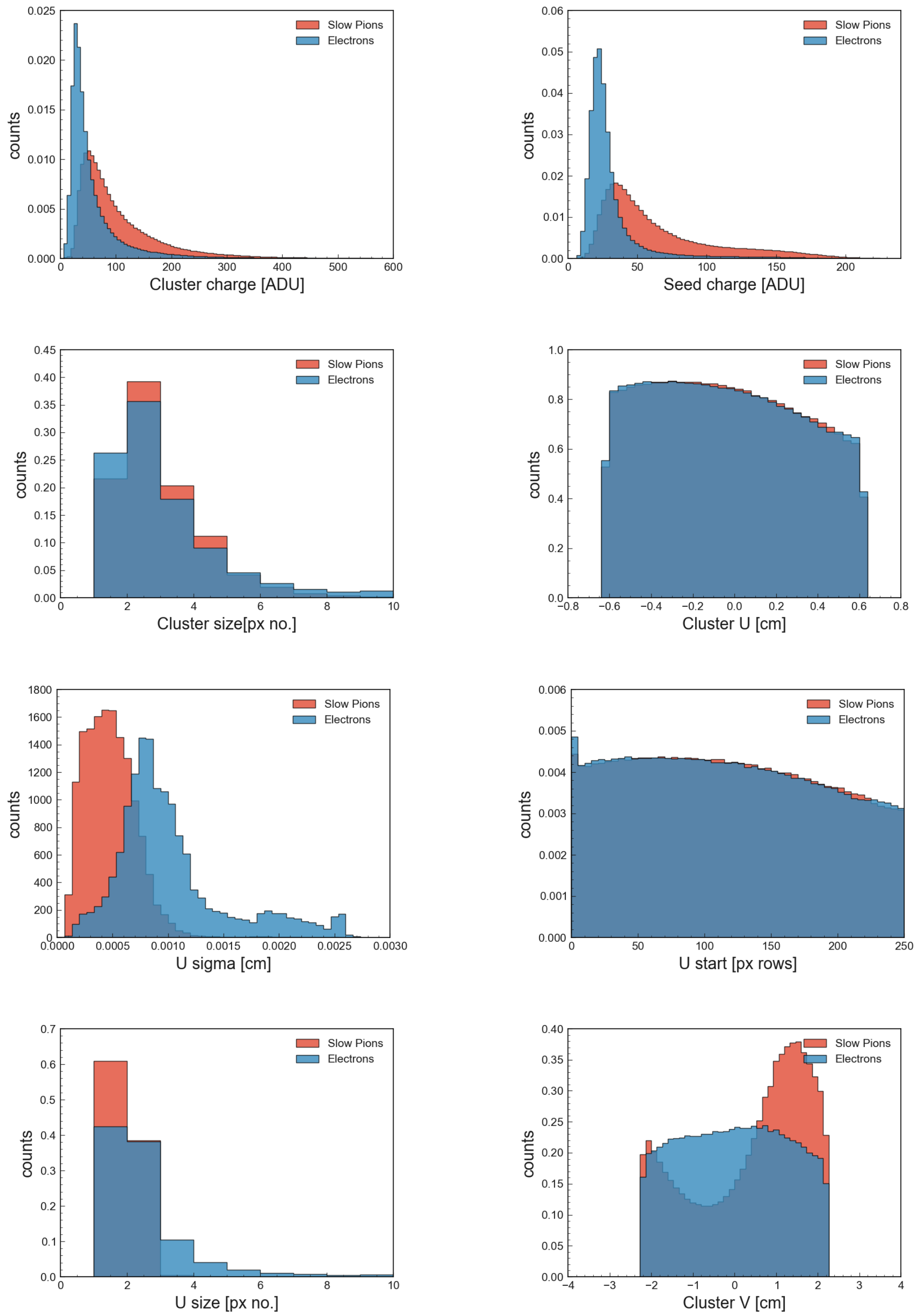
Cluster Variable	Description
Cl_charge	Charge deposited by a passing particle on the cluster
Cl_seed	Value of the pixel with the highest charge in the cluster
Cl_layer	Layer of the PXD in which the cluster is found
Cl_size	No. of pixels contained in the cluster
Cl_U	Cluster position in the sensor along the U direction
U_sigma	Error in the U position
U_start	Pixel along the U direction in which the cluster begins
U_size	No. of pixels along the U direction
Cl_V	Cluster position in a sensor along the V direction
V_sigma	Error in the V position
V_start	Pixel along the V direction in which the cluster begins
V_size	No. of pixels along the V direction
Rho	Cluster shape-correlation coefficient

Variables such as cluster position and their uncertainty are determined with a weighted sum with respect to the pixel charge and are defined as follows:

$$Cl\_U = \frac{\sum_i U_i \cdot Q_i}{\sum_i Q_i} \qquad Cl\_V = \frac{\sum_i V_i \cdot Q_i}{\sum_i Q_i} \tag{5.2}$$

$$U\_sigma^2 = \frac{\sum_i (U_i - U_m)^2 \cdot Q_i}{\sum_i Q_i} \qquad V\_sigma^2 = \frac{\sum_i (V_i - V_m)^2 \cdot Q_i}{\sum_i Q_i}, \tag{5.3}$$

where Q_i, V_i and U_i are the charge, V and U position of the i^{th} pixel respectively. V_m and U_m come from the MC track point of impact relative to the cluster center, calculated with Equation 5.2.


Slow Pions
Electrons
counts
0.000
0.005
0.010
0.015
0.020
0.025
0
100
200
300
400
500
600
Cluster charge [ADU]
Slow Pions
Electrons
counts
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
50
100
150
200
Seed charge [ADU]
Slow Pions
Electrons
counts
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0
2
4
6
8
10
Cluster size[px no.]
Slow Pions
Electrons
counts
0.0
0.2
0.4
0.6
0.8
1.0
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Cluster U [cm]
Slow Pions
Electrons
counts
0
200
400
600
800
1000
1200
1400
1600
1800
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
U sigma [cm]
Slow Pions
Electrons
counts
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0
50
100
150
200
250
U start [px rows]
Slow Pions
Electrons
counts
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2
4
6
8
10
U size [px no.]
Slow Pions
Electrons
counts
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
−4
−3
−2
−1
0
1
2
3
4
Cluster V [cm]

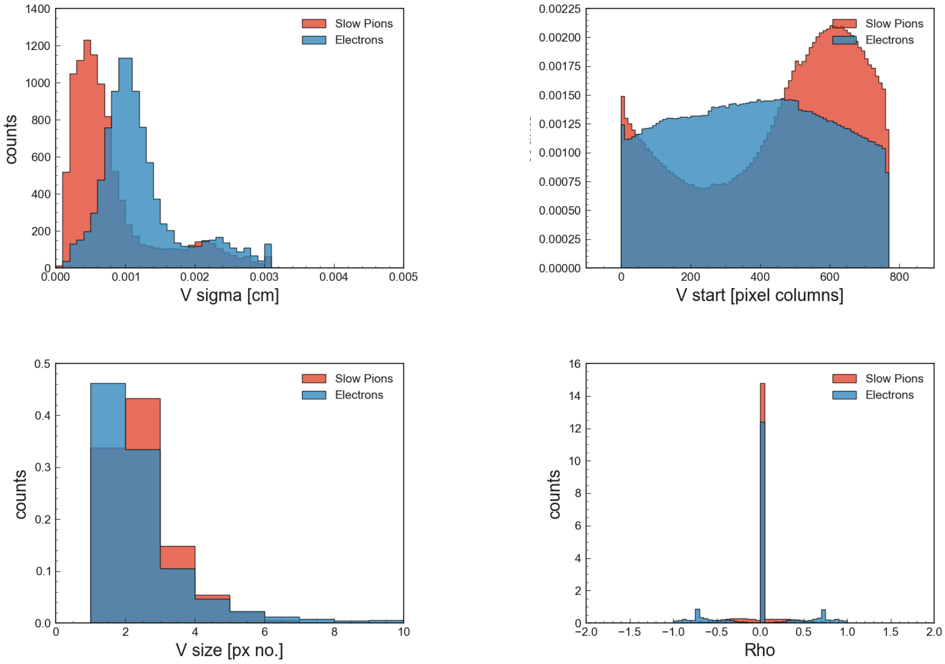
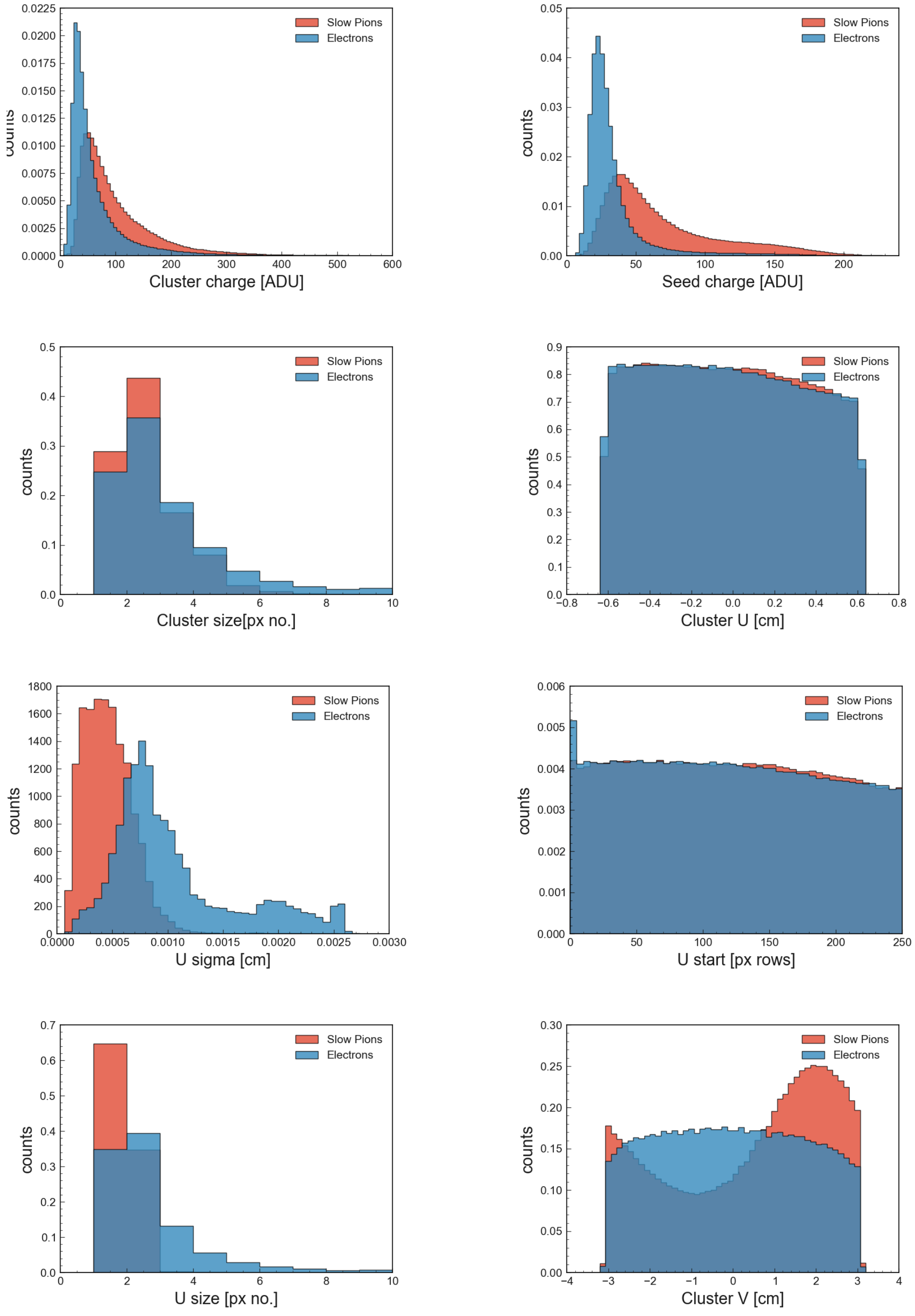


Figure 5.1: Cluster variable distributions for PXD layer 1. Data sample contains $6.38{\times}10^6$ events.


Slow Pions
Electrons
counts
Cluster charge [ADU]
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
0.0225
0
100
200
300
400
500
600
Seed charge [ADU]
0.00
0.01
0.02
0.03
0.04
0.05
0
50
100
150
200
Cluster size[px no.]
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
Cluster U [cm]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
U sigma [cm]
0
200
400
600
800
1000
1200
1400
1600
1800
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
U start [px rows]
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0
50
100
150
200
250
U size [px no.]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2
4
6
8
10
Cluster V [cm]
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−4
−3
−2
−1
0
1
2
3
4

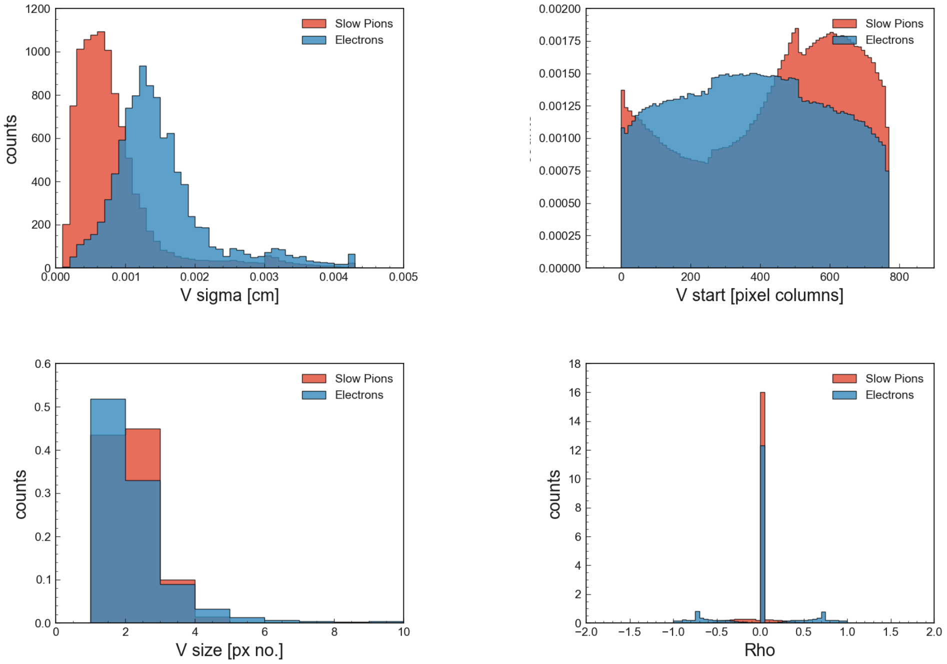


Figure 5.2: Cluster variable distributions for PXD layer 2. Data sample contains 4.60×10^6 events.

From the distributions, differences can be seen between slow pions and electrons, as well as between PXD layer 1 and 2. The variables with the strongest contrast are Cl_charge, Cl_seed, U_sigma, V_sigma, Cl_V and V_start. Their differences can be attributed to a number of factors. In the case of Cl_charge and Cl_seed, pions deposit on average a higher charge than electrons on the detector due to Bethe-Bloch and different dominant energy loss mechanisms. Since particles with low speed have a greater energy loss when traversing the same amount of material, slow pions will deposit more energy in the PXD compared to other particles with higher speed. Furthermore, slow pions show a smaller U_sigma and V_sigma compared to electrons due to electrons having lower momentum and therefore curling in the PXD, generating on average bigger clusters. Lastly, the differences in Cl_V and V_start between the two particle types can be explained by the boost happening during a collision in the Belle II detector. Since slow pions follow the D^* direction, they will also be boosted. This explains the uneven shape seen on the distributions. On the other hand, the QED electrons distribution is uniform in general.

It is important to recall that Cl_U and Cl_V describe the local coordinate system of the individual PXD sensors. Therefore, Cl_V and Cl_U variables show the position along a sensor, rather than along a ladder. Since V is the coordinate along a sensor, the distribution of Cl_V does not represent what is physically happening at the PXD. This is not the case along U because there is no boost along this direction. To obtain the physical distribution of Cl_V, a transformation from local coordinate V to absolute coordinate z is performed. Moreover, one can also transform to emission angle θ. This is done in detail in Section 5.2.

The cluster variables can be further studied by classifying them according to their cluster size. As shown in Figures 5.1 and 5.2, a cluster can have up to 10 pixels (sometimes even more, but since it does not often happen, it is enough to consider up to 10). 1 and 2 pixel clusters dominate and clusters become rather seldom after 4 pixels. Therefore, the cluster variables were grouped into 1, 2, 3 and 4 or more pixel clusters for a more detailed analysis. The respective plots can be found in Appendix A. This will come useful in Section 5.3, where the training methods for the neural networks are specified.

5.2 Physical V-Distributions

Differences in the distribution of Cl_V have been observed for different pixel multiplicities. More specifically, the distribution of slow pions and electrons become more similar with increasing pixel multiplicity. This is illustrated in Figure 5.5. To understand the reason for these changes, we look into the physical distribution of the variable Cl_V. This variable gives the position of the pixel cluster along the local coordinate V in a sensor. This means that the distribution is along a module, not a ladder. However, cluster position along a ladder gives a physical distribution showing how the slow pions and QED electrons behave along the detector ($z-$axis in the Belle II coordinate system). For this reason, a transformation from local V position to absolute z position is performed. This requires an understanding of the PXD geometry and position with respect to $z = 0$, which is the position of the IP in the detector. The dimensions are sketched in Figure 5.3. From this diagram, the following transformation formulas can be derived:

For inner layer (layer 1):

$$z_{fwd} = \frac{4.5}{2} + v_{fwd} + 1 \tag{5.4}$$

$$z_{bwd} = -\frac{4.5}{2} + v_{bwd} + 1 \tag{5.5}$$

For outer layer (layer 2):

$$z_{fwd} = \frac{6.144}{2} + v_{fwd} + 1.7 \tag{5.6}$$

$$z_{bwd} = -\frac{6.144}{2} + v_{bwd} + 1.7 \tag{5.7}$$

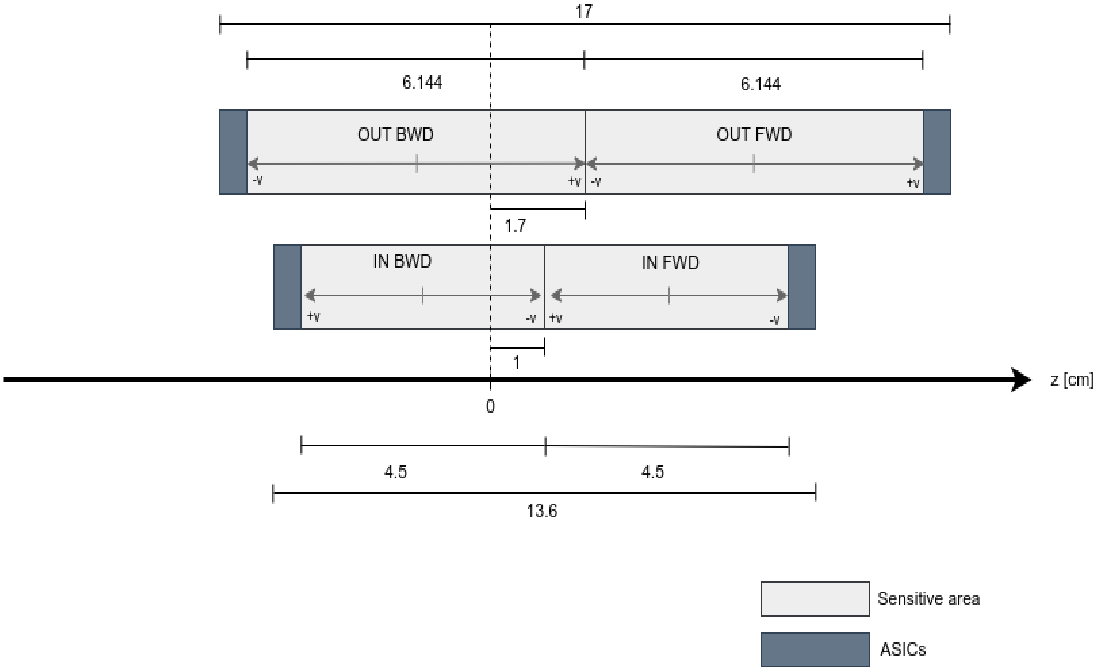
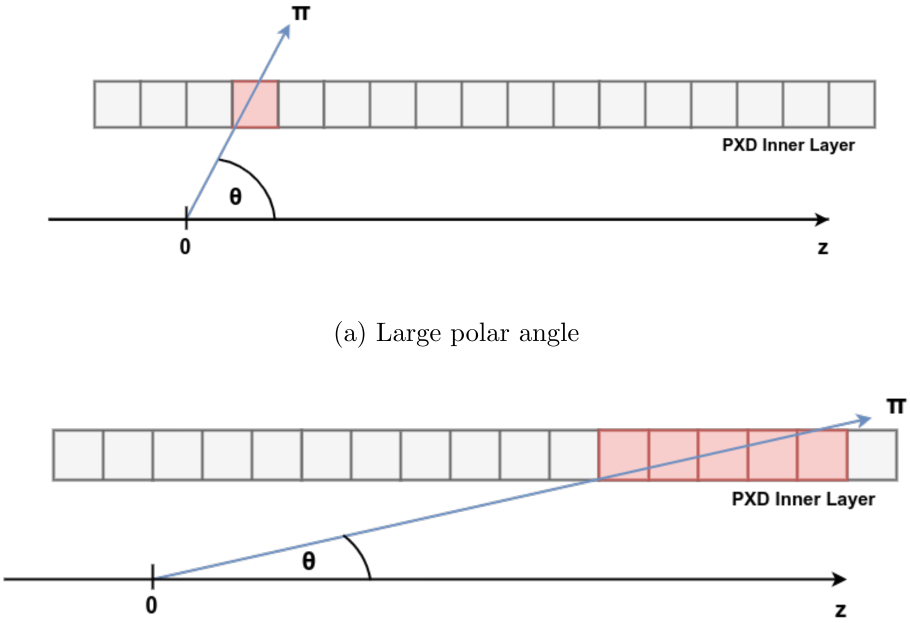


Figure 5.3: PXD measured with respect to the z coordinate system of Belle II. All measures are in cm. IN and OUT refer to the inner and outer layer respectively. FWD shows the forward modules and BWD indicates the backward modules.

Applying these transformations results in the distributions shown in Figure 5.6. Slow pion clusters with small pixel multiplicity tend to be around the IP, whereas electron clusters are spread along the ladder. As pixel multiplicity increases, slow pion clusters tend to spread out and the peak is shifted to the $+z$ direction due to the boost they experience. Furthermore, if the absolute position of a cluster is known (x, y, z), the emission angle θ can be calculated using Equation 5.8. This quantity is also plotted and shown in Figure 5.7. It is expected that a particle traversing the PXD at shallow angles will produce a cluster with more pixels, since it has more contact with the detector. Similarly, a particle traversing the PXD at angles around 90° will produce clusters with small number of pixels. Figure 5.4 illustrates this. This effect is visible in Figure 5.7: for large pixel multiplicity, slow pion and electron clusters are mostly found at shallow angles, around 30° and 150°. For small pixel number, slow pion clusters are found mainly around 50° to 120°, whereas electron clusters are again spread through the ladder.

$$\theta = \arctan \frac{\sqrt{x^2 + y^2}}{z} \tag{5.8}$$



(a) Large polar angle

(b) Small polar angle

Figure 5.4: Illustration showing how a passing particle with polar angle θ creates a cluster in the PXD. A shallow θ will hit more pixels in the detector, creating larger clusters.

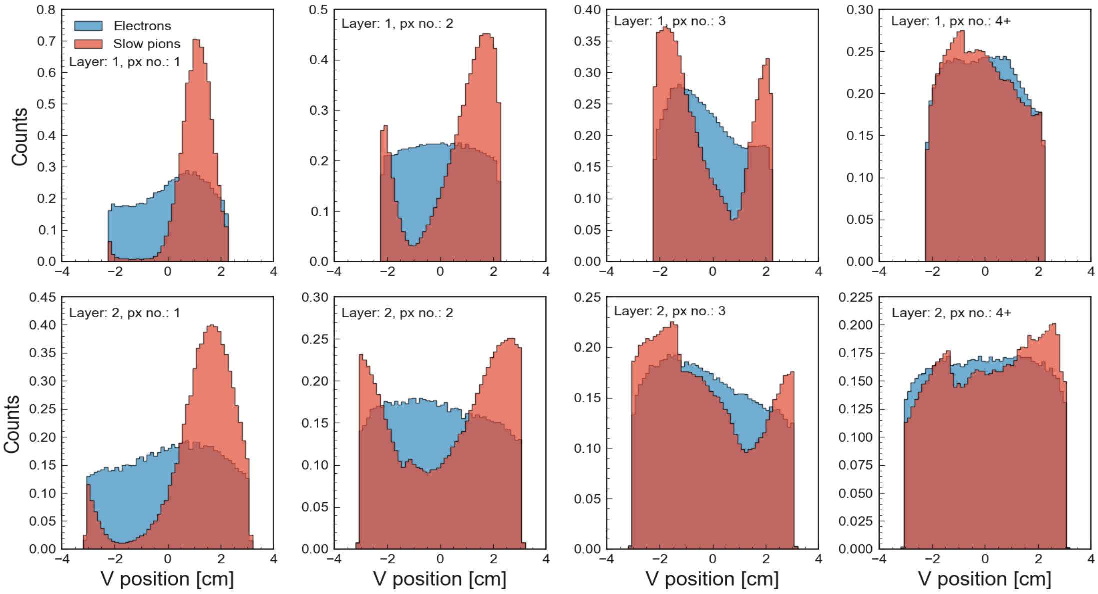


Figure 5.5: Cl_V distribution for the different specific cases. Pixel multiplicities 1, 2, 3 and 4+ are shown for layer 1 in the first row and for layer 2 in the second row.

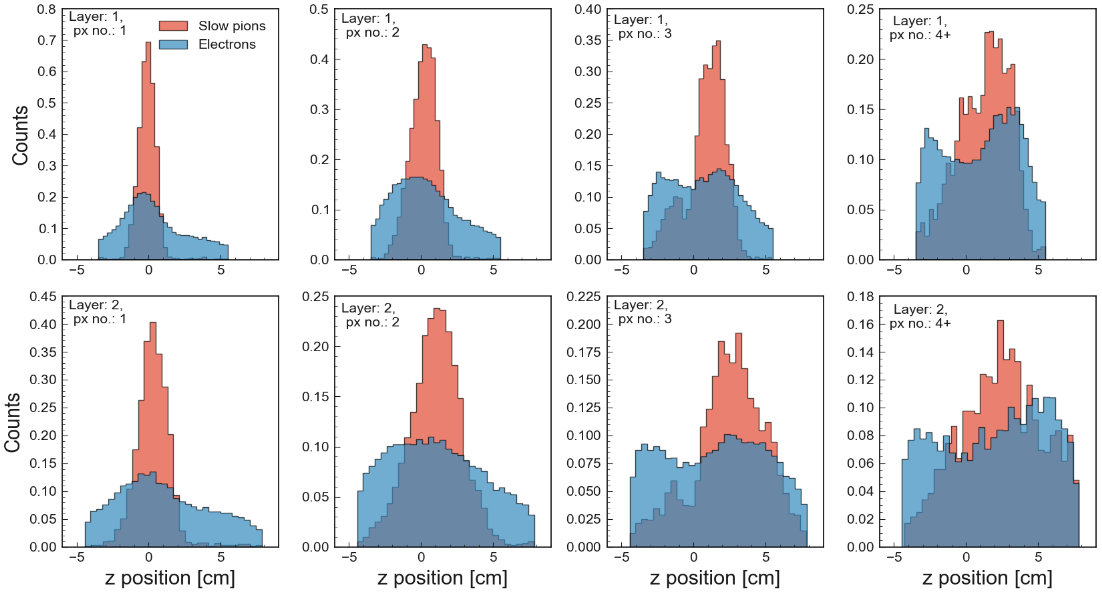


Figure 5.6: Cluster z position for the different specific cases. Pixel multiplicities 1, 2, 3 and 4+ are shown for layer 1 in the first row and for layer 2 in the second row.

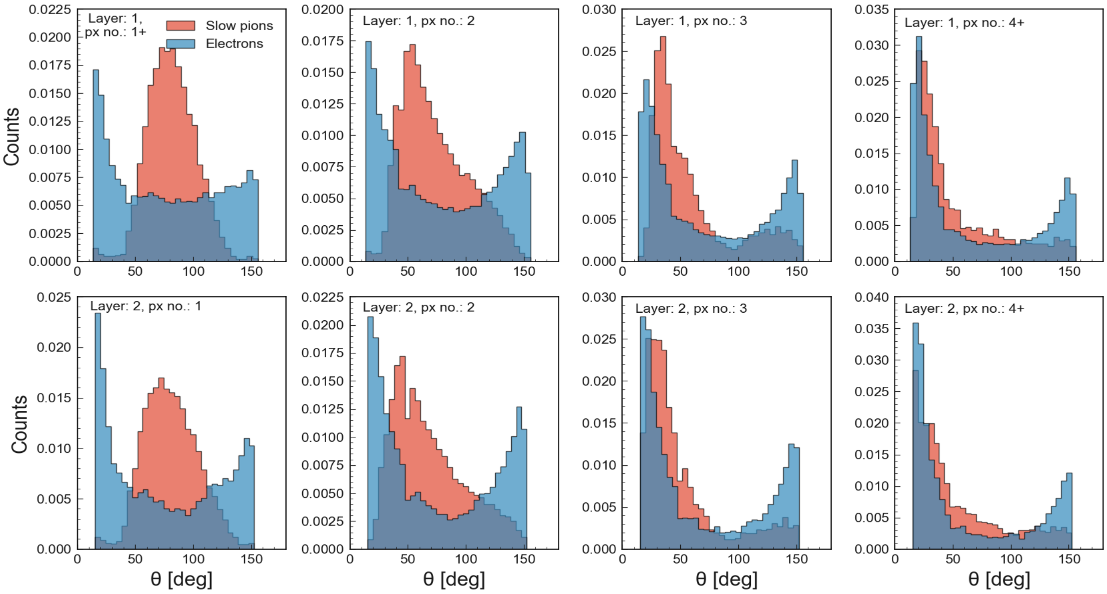


Figure 5.7: Polar angle θ distribution for the different specific cases. Pixel multiplicities 1, 2, 3 and 4+ are shown for layer 1 in the first row and for layer 2 in the second row.

5.3 Artificial Neural Networks

An artificial neural network (NN) is a powerful class of models capable of learning a particular task by identifying patterns in a provided data sample. As the name suggests, artificial neural networks imitate the functioning of biological neurons in the human brain, allowing them to learn from data and make predictions. This is particularly useful when dealing with data with complex patterns, as it provides an alternative and effective way of extracting the underlying pattern which would otherwise not be feasible with traditional methods. For this reason, this tool has and continues to revolutionize the way data analysis can be carried out.

Fundamentally, neural networks consist of interconnected computing units called neurons or nodes. Layers of neurons are formed to process input data in order to obtain output results. The working principle is the following: An input x_i is fed to each neuron i in the input layer of the NN, where the number of neurons equals n, the number of inputs. The input neurons are fed into a fully connected second layer of neurons, as shown in Figure 5.8. In this case, the neural network has one hidden layer, but this need not be the case. In general, a neural network can have an arbitrary number of hidden layers. Moreover, each connection between nodes is depicted by a specific numerical weight (w_1, w_2,...,w_n), as shown in Figure 5.9. Then, a weighted sum is done at each node and the result is passed through an activation function (f_{act}) to obtain an output. The mathematical operation performed at each node is presented in Equation 5.9:

$$O = f_{act}\left(\sum_{i}^{n} w_i x_i\right) \tag{5.9}$$

Neural networks have hyperparameters, which are parameters set before the training process, and control the behavior and performance of the model. These hyperparameters include the learning rate, number of layers, number of neurons per layer, activation functions, etc. The choice of these hyperparameters is important to reach a neural network's optimal performance. Their descriptions are summarized in Table 5.2.

Table 5.2: List of neural network hyperparameters.

Hyperparameter	Description
Learning rate	The step size at which the weights of the model are updated during training
No. of hidden layers	No. of layers between the input and output layer
No. of neurons per layer	It defines how capable the model is to represent different data patterns
Activation function	Function that introduces non-linearities in the NN to model complex relationships
Batch size	Number of data batches to be fed to the NN. The weights are updated based on the computations done on each batch
Training epoch	How many times the entire data set is used to update the model's weights
Optimization algorithm	Defines how the weights are updated during training. This affects convergence speed and performance

A neural network is then subjected to a learning phase, known as training. During this phase, a large set of example inputs are passed through the NN along with their corresponding target values. The NN then compares its predictions with the target values in order to optimize itself. A loss function is used to quantify the difference between the predicted output of the network and the actual output. In this way, the performance of the neural network can be measured. The choice of the loss function depends on the task being solved. For classification tasks, which is the type discussed in this work, binary cross-entropy is widely used. The formula is shown in Equation 5.10, where y is the true label and x is the predicted probability output of the network.

$$L = -[y \cdot log(x) + (1 - y) \cdot log(1 - x)] \tag{5.10}$$

The aim is to minimize this loss during the training phase, which is done through a method known as back-propagation, in which the weights between neurons are updated to improve predictions. The weights must be trained in supervised mode. In this manner, the NN learns from the example data and is capable of making predictions on unseen data.

In this study, neural networks were used to develop an algorithm that takes PXD cluster variables data as input and gives as output the probability of the cluster being generated by a slow pion or an electron. As mentioned in Section 5.1, cluster variable data can be sub-divided according to PXD layer (layer number) and pixel multiplicity (cluster size). Therefore, we investigated two approaches in the development of the algorithm:

- **General approach:** In this case, a ”general” data sample was used as input data for the NN. This means there were no discrimination between layer number nor pixel multiplicity. Here, one neural network is trained to separate slow pions from electrons.
- **Specific approach:** In this case, ”specific” data samples are used as input data for specific NNs. This means that cluster variable data was separated according to their layer number (1 or 2) and pixel multiplicity (1, 2, 3 or 4+) so that in total there were 8 data sets being used to train 8 different NNs. Each one specialises on a specific group of clusters, e.g clusters in layer 2 containing 3 pixels.

We compare the two approaches to see which method has a higher efficiency identifying slow pions and a higher electron rejection rate. In this way, it can be known whether implementing one neural network is enough or whether specific neural networks are needed. The details on the architecture, data preparation and training for both approaches can be found in Sections 5.3.1 and 5.3.2 respectively.

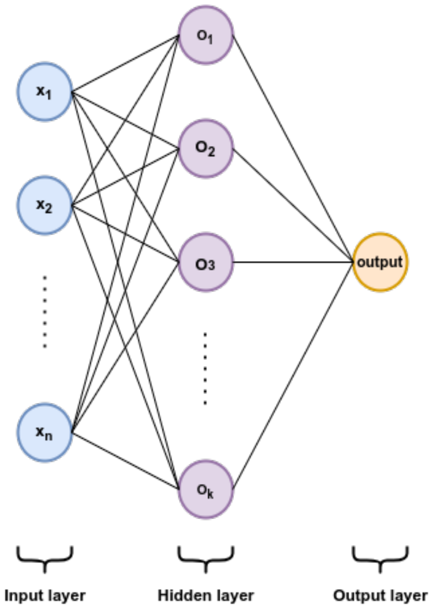


Figure 5.8: Structure of a neural network with one hidden layer. n is the number of input neurons, corresponding to n input parameters and k is the number of neurons in the hidden layers. O_i corresponds to the operation performed at each neuron, described in Equation 5.9.

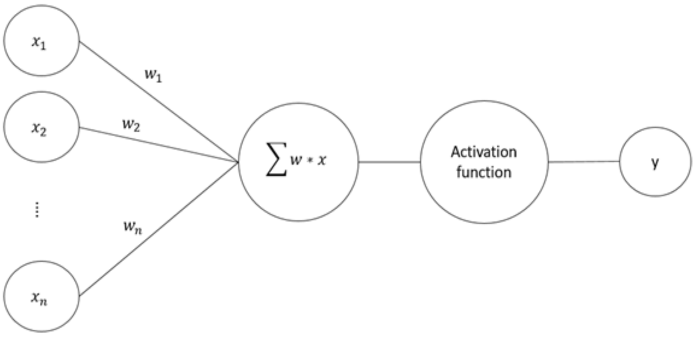


Figure 5.9: Detailed view of the action of a single neuron. The input parameters are passed to a neuron with their corresponding weights. The neuron performs a weighted sum and the result is given to an activation function, which gives the neuron output.

5.3.1 Network Architecture

Neural networks can be structured in various ways, with arbitrary number of layers and nodes. However, there is a point where augmenting these quantities does not improve accuracy. Therefore, the aim is to have a NN architecture that allows for both efficient computation and accurate predictions.

To find the most suitable architecture, a grid search was performed. The number of hidden layers and nodes were varied, as well as other factors such as learning rate, batch size and number of epochs. From this study, it was concluded that a NN of one hidden layer with 100 nodes performs with optimal efficiency. In other words, increasing these numbers did not add a significant improvement. The grid search details can be found in Appendix D.

The chosen architecture is therefore a fully-connected neural network with an input layer of cluster variables, one hidden layer with 100 nodes and an output layer with a single node which determines the type of particle associated to the corresponding cluster variables. That is, either an electron or a slow pion. The decision is made by applying a threshold to the output value. This architecture applies for both the general and specific approach (details on Section 5.3.2). It is worth noting that deep learning (more than two hidden layers) was also tried but no significant improvement was found. The results are shown in Appendix C.

The framework used for the network building and training is the PyTorch library [32]. In addition, due to its suitability for binary classification, the binary cross-entropy function is used as the loss function. As an optimizer to update the model's parameters, ADAM is chosen because it offers the advantage of taking into account the first and second moment of the gradient when calculating the direction of steepest descent, which leads to a fast convergence compared to other strategies. Moreover, the ReLU function is used as the activation function for the hidden nodes and the Sigmoid function is used for the output layer. The choice of this functions was due to ReLU being very quick in terms of training and Sigmoid providing non-linearity and a suitable output range for probability interpretation.

5.3.2 Data Preparation and Training

The variables discussed in Section 5.1 coming from MC-generated data are used as input parameters for the neural network training phase. Since the data comes from simulation, it is known whether the cluster variables belong to a slow pion or electron (represented with a 1 for slow pion and 0 for electron). Information about the cluster variables is extracted using the Belle II Analysis Software Framework

(BASF2). The input data is randomized to make sure information about slow pions and electrons is evenly distributed. In addition, to obtain a balanced training and improve convergence, all the input variables are normalized to a [-1,1] interval.

The specifications made up to now apply for both of the approaches studied. However, their training procedure is different and therefore are presented separately. In addition, the data samples used for the specific approach are subsets of the data used for the general approach (DATA0), as shown in Figure 5.10. These subsets are based on layer number and pixel multiplicity.

General Approach

In this case, DATA0 sample is used and it contains more than 11×10^6 samples composed of 50% slow pions and 50% electrons. This sample was further split into 3 equal parts for training, validation and testing. A neural network (referred to as NN0) was then trained to distinguish between slow pions and electrons for all layers and pixel multiplicities together. The training was done with the hyperparameters described in Table 5.3. The learning rate was set to 1×10^{-4}. All cluster variables in Table 5.1 are used as input parameters as they all show differences in their distributions (see Figures 5.1 and 5.2).

Specific Approach

This time, DATA11, DATA12, DATA13, DATA14+, DATA21, DATA22, DATA23 and DATA24+ were subdivided into 3 equal parts for training, validation and testing. 8 neural networks (reffered to as NN11, NN12, NN13, NN14+, NN21, NN22, NN23 and NN24+) were then trained using the 8 data samples. In this way, NNs specialized on clusters with specific layer number and pixel multiplicity are obtained. The hyperparameters for the neural networks are shown in Table 5.3. For all cases, a learning rate of 1×10^{-4} was used. Moreover, as seen in the Cluster size distributions in Figures 5.1 and 5.2, the number of pixels in a cluster varies between slow pions and electrons and clusters with one and two pixels dominate. For this reason, the data subsets do not contain an equal number of electrons and slow pions. This is not ideal since training with an uneven sample can create a bias in the neural network. To overcome this problem, the subset files were modified to have even number of slow pions and electrons. In addition, since the data is grouped by layer and pixel size, these cluster variables become irrelevant and therefore are not used during training. In addition, the distributions of DATA11 and DATA21, shown in Appendix A, suggest that U_size, V_size and Rho have no differences and therefore are also dropped for these cases. Lastly, epoch timer and batch size varies slightly depending on what sample we use. These numbers were chosen after trying different combinations and deciding for the combination which yielded a lower loss function while avoiding over-fitting. These numbers can be found in Appendix D

Table 5.3: Training hyperparameters for the NNs used in the general and specific approach. In all cases a single-hidden-layer network with 100 nodes was used.

Data sample	Data size ($\times 10^6$)	No. of input parameters	Variables dropped	Epoch timer	Batch size
DATA0	11	13	None	3000	10,000
DATA11	1.3	8	Cl_layer, Cl_size, Cl_U, Cl_V, Rho	3000	1000
DATA12	2.2	11	Cl_layer, Cl_size	3000	1000
DATA13	1.2	11	Cl_layer, Cl_size	3000	1000
DATA14+	1.1	11	Cl_layer, Cl_size	1000	1000
DATA21	1.0	8	Cl_layer, Cl_size, Cl_U, Cl_V, Rho	3000	600
DATA22	1.4	11	Cl_layer, Cl_size	3000	2000
DATA23	0.7	11	Cl_layer, Cl_size	3000	600
DATA24+	0.6	11	Cl_layer, Cl_size	2000	1000

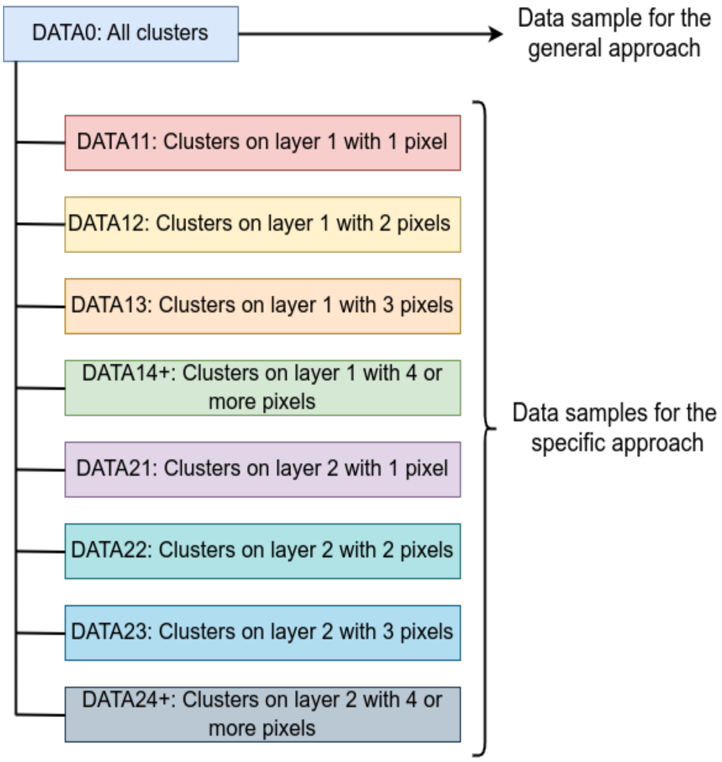

DATA0: All clusters
Data sample for the general approach
DATA11: Clusters on layer 1 with 1 pixel
DATA12: Clusters on layer 1 with 2 pixels
DATA13: Clusters on layer 1 with 3 pixels
DATA14+: Clusters on layer 1 with 4 or more pixels
DATA21: Clusters on layer 2 with 1 pixel
DATA22: Clusters on layer 2 with 2 pixels
DATA23: Clusters on layer 2 with 3 pixels
DATA24+: Clusters on layer 2 with 4 or more pixels
Data samples for the specific approach

Figure 5.10: Conceptual map illustrating which data sets are used in the general and specific approach. DATA0 represents a data file containing all information about the cluster variables. DATAXY represents a data file containing the variables of clusters in layer X with pixel multiplicity Y ($X = 1, 2$ and $Y = 1, 2, 3, 4+$).

6 Results

To evaluate the performance of a neural network, test data is given as input and a probability distribution is returned by the neural network. For slow pions as input, the distribution is expected to peak close to 1. For electrons, this distribution should peak at 0. To decide what is considered a slow pion, a threshold is set. This value can be freely chosen based on the required background rejection. To begin with, the cut was set at 0.5 to evaluate the network's output without a strict threshold. Everything above this value is considered to be a slow pion. However, since the PXD requires a data reduction by a factor of 10, a cut value giving the required rejection rate of 90% is also considered. This cut provides the results for a ”worst case scenario” data reduction requirement. Note that this cut can be relaxed if the data-rate of other sub-detectors increases. Given a threshold, the predicted values of a model can be compared with the actual values and classified in order to quantify the model's performance. The classification is made using a confusion matrix (CM) [33], illustrated in Figure 6.1. The entries of the CM are defined as true negative (TN), false positive (FP), false negative (FN) and true positive (TP). Quantities to measure the performance of the NN output are defined from the entries of the CM. The ones we focus on are accuracy, efficiency, purity and rejection. These quantities are defined as in Equation 6.1, 6.2, 6.3, and 6.4 respectively. Note that P is the number of slow pions and N the number of background events in Equation 6.1. These parameters are dependent on the threshold value. Moreover, accuracy describes the probability of getting a correct prediction for the slow pions and the electrons, efficiency measures the probability of a slow pion being correctly predicted, purity gives how pure the sample predicted as slow pions is and rejection measures the probability of an electron being correctly predicted. See Figure 6.2 for illustration.

$$Accuracy = \frac{TP + TN}{P + N} \tag{6.1}$$

$$Efficiency = \frac{TP}{TP + FN} \tag{6.2}$$

$$Purity = \frac{TP}{TP + FP} \tag{6.3}$$

$$Rejection = \frac{TN}{TN + FP} \tag{6.4}$$

It is ideal to have P $\approx$ N in a testing sample. However, there are data samples that will be dominated by background. In these cases, the quantities that determine the neural network performance are normalised such that P $\approx$ N is obtained. To do so, we define a normalisation variable f as the ratio between N and P,

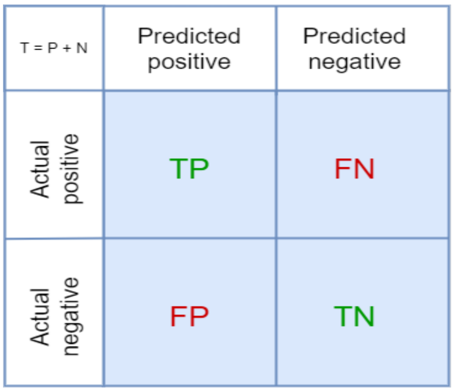
$$f = \frac{N(Background)}{N(SlowPions)}$$

Since background hits dominate, it follows that N $>$ P so that $f > 1$. f is then used to normalise the quantities dependent to N in the following way:

$$N(Background)_{norm} = N(Background) \times \frac{1}{f}$$

$$FP_{norm} = FP \times \frac{1}{f}$$

$$TN_{norm} = TN \times \frac{1}{f}$$



T = P + N	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

Figure 6.1: Confusion matrix. In this case, P is the number of slow pions, N is the number of electrons and T is the sample size.

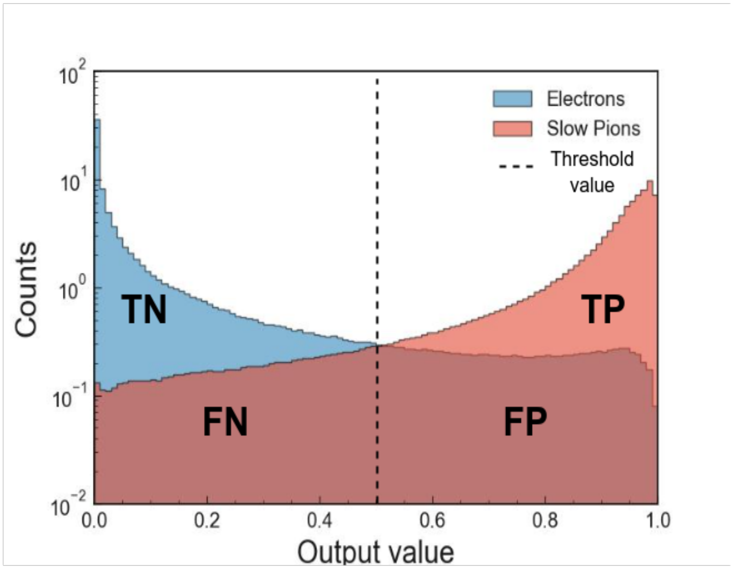


Figure 6.2: Histogram separation into TP, TN, FP and FN for a given threshold value (here: 0.5).

A set of unseen data samples were used to test the neural networks. The first data sample contains MC-generated slow pion and QED electron events. Since the NNs were trained with the same type of data, this test gives a direct evaluation of the neural network performance. The second data sample contains MC-generated slow pion with Early Phase 3 (EP3) background expected in the PXD modules. This is an intermediate step to see if the trained models are able to reject not only electrons, but also beam background from Belle II early phase 3 period. The last sample contains real slow pion data. The results for each data sample are presented in Sections 6.1, 6.2 and 6.3. The first two data samples were studied using the general and specific approach. For real data, only the general approach was studied. Lastly, the errors presented come from statistical uncertainties.

6.1 Monte Carlo - Electrons as Background

6.1.1 General Approach

In this section, the results of NN0 (see Section 5.3.2 for the nomenclature of this and the other following networks) are presented. Table 6.1 summarizes the values obtained for accuracy, efficiency, purity and rejection for a given threshold. The value for the average loss function is also given. Figure 6.3 shows the neural network output value distribution.

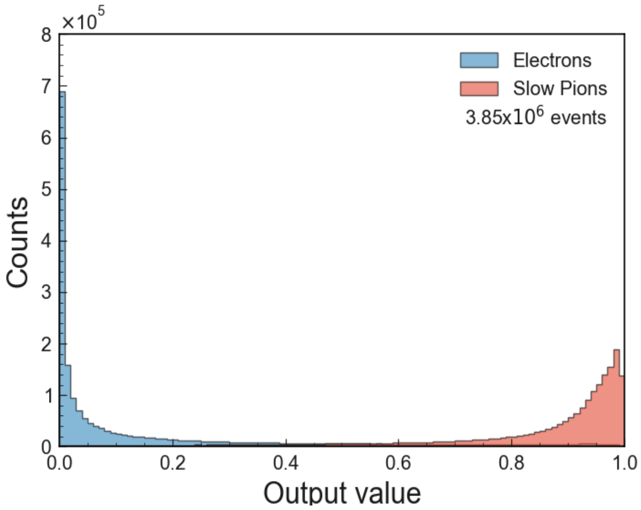


Figure 6.3: NN0 output value distribution when tested on DATA0.

Table 6.1: Performance of NN0 for threshold values of 0.5 and 0.572. The latter is the value corresponding to an electron rejection rate of 90%. Training and testing done with data samples from DATA0.

Case	**Cut**	**Accuracy [%]**	**Efficiency [%]**	**Purity [%]**	**Rejection [%]**	**Loss**
DATA0	0.5	89.2 ± 0.05	90.73 ± 0.02	88.04 ± 0.02	87.67 ± 0.02	0.263
DATA0	0.582	89.04 ± 0.05	88.10 ± 0.02	89.78 ± 0.02	89.98 ± 0.02	0.263

6.1.2 Specific Approach

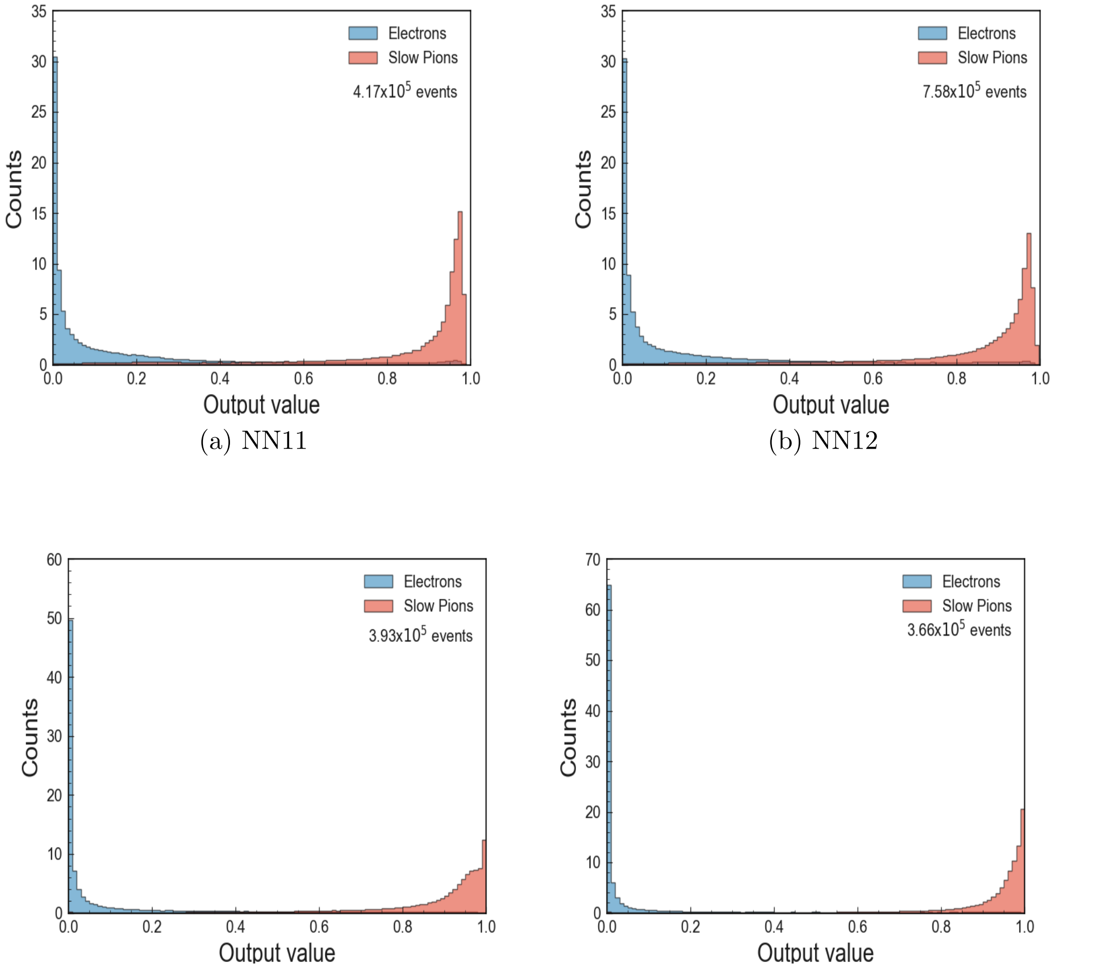
In this section, the results of the neural networks trained with the subset data based on PXD layer number and pixel multiplicity are presented. Tables 6.2 and 6.3 summarize the values obtained for accuracy, efficiency, purity and rejection for a cut of 0.5 and of a value corresponding to an electron rejection rate of 90% respectively. The values for the average loss function are also given. In addition, Figures 6.4 and 6.5 show the neural network output value distribution for specific pixel multiplicity at layer 1 and layer 2 respectively.

Table 6.2: Specific NN performances. Cut at 0.5.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]	**Loss**
DATA11	0.5	88.67 ± 0.15	88.82 ± 0.07	88.56 ± 0.07	88.52 ± 0.07	0.277
DATA12	0.5	87.91 ± 0.11	88.87 ± 0.05	87.20 ± 0.5	86.96 ± 0.05	0.292
DATA13	0.5	90.51 ± 0.15	92.62 ± 0.06	88.86 ± 0.07	88.39 ± 0.07	0.231
DATA14+	0.5	93.74 ± 0.16	95.81 ± 0.05	92.00 ± 0.06	91.67 ± 0.06	0.164
DATA21	0.5	88.20 ± 0.17	88.63 ± 0.08	87.87 ± 0.08	87.77 ± 0.08	0.287
DATA22	0.5	87.36 ± 0.14	89.22 ± 0.06	86.02 ± 0.07	85.50 ± 0.07	0.302
DATA23	0.5	90.74 ± 0.19	93.59 ± 0.07	88.55 ± 0.09	87.89 ± 0.09	0.223
DATA24+	0.5	96.07 ± 0.21	97.56 ± 0.05	94.75 ± 0.07	94.59 ± 0.07	0.114

Table 6.3: Specific NN performances. Cut chosen to reach a 90% QED electron rejection rate.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]	**Loss**
DATA11	0.558	88.58 ± 0.15	87.10 ± 0.07	89.75 ± 0.07	90.05 ± 0.07	0.277
DATA12	0.608	87.51 ± 0.11	84.97 ± 0.06	89.52 ± 0.05	90.05 ± 0.05	0.292
DATA13	0.565	90.44 ± 0.15	90.82 ± 0.07	90.13 ± 0.07	90.05 ± 0.07	0.231
DATA14+	0.395	93.56 ± 0.16	97.11 ± 0.04	90.68 ± 0.07	90.02 ± 0.07	0.164
DATA21	0.582	88.04 ± 0.17	86.08 ± 0.09	89.60 ± 0.08	90.01 ± 0.07	0.287
DATA22	0.650	86.81 ± 0.14	83.58 ± 0.08	89.34 ± 0.07	90.03 ± 0.06	0.302
DATA23	0.587	90.54 ± 0.19	91.06 ± 0.08	90.13 ± 0.09	90.03 ± 0.09	0.223
DATA24+	0.133	94.65 ± 0.21	99.24 ± 0.03	90.89 ± 0.09	90.05 ± 0.09	0.114



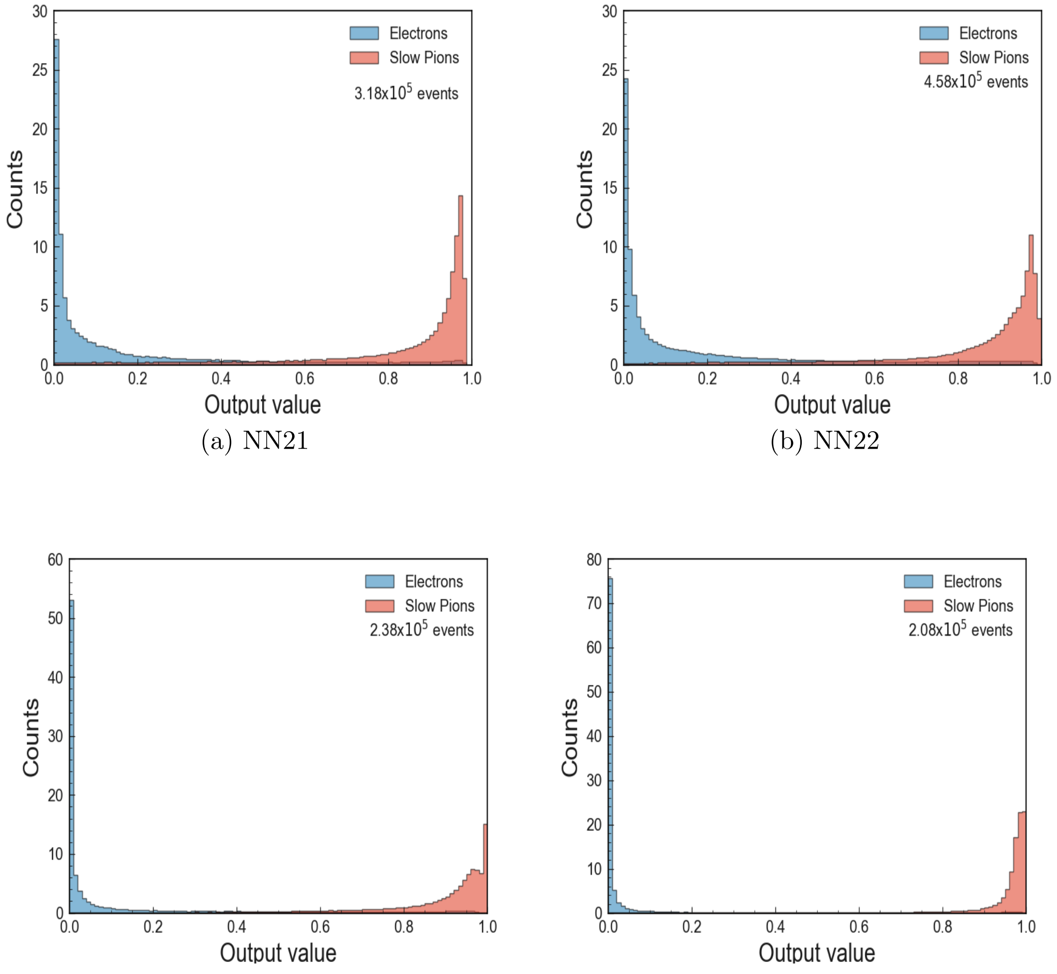
(a) NN11

(b) NN12

(c) NN13

(d) NN14+

Figure 6.4: Output value distributions for neural networks trained on data samples with PXD layer number = 1 and pixel multiplicities = 1, 2, 3 and 4+. (a) NN11 tested on DATA11 (b)NN12 tested on DATA12 (c)NN13 tested on DATA13 (4)NN14 tested on DATA14+.



(a) NN21

(b) NN22

(c) NN23

(d) NN24+

Figure 6.5: Output value distributions for neural networks trained on data samples with PXD layer number = 2 and pixel multiplicities = 1, 2, 3 and 4+. (a) NN21 tested on DATA21 (b)NN22 tested on DATA22 (c)NN23 tested on DATA23 (4)NN24 tested on DATA24+.

The specialized neural networks can be compared among each other using Receiver Operating Characteristic (ROC) curves. A ROC curve illustrates the ability of a binary classifier to discriminate as the threshold value is varied. This is done to see how pixel multiplicity can affect the performance of the neural networks. For example, it is expected that clusters with one pixel will be more challenging to classify than clusters with e.g. 4 or more pixels. The comparison is shown in Figure 6.6.

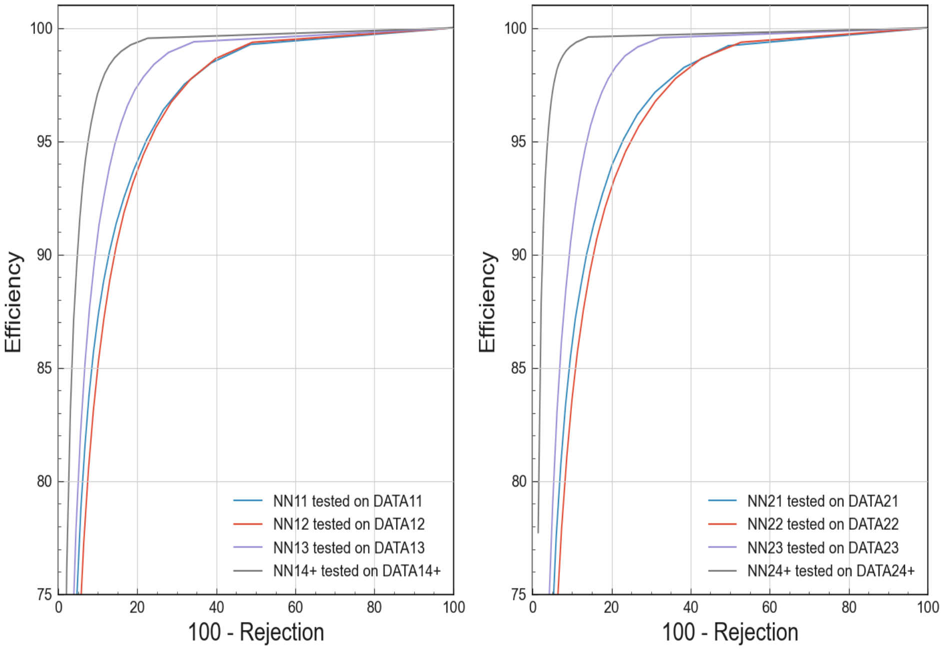


Figure 6.6: ROC curves showing the specialized neural networks performance. The left subplot compares the neural networks trained with clusters of different pixel multiplicities = 1, 2, 3, 4+ on layer 1. The right subplot compares the neural networks trained with clusters of pixel multiplicities = 1, 2, 3, 4+ on layer 2.

Figure 6.6 shows that NN12 and NN22 (clusters with two pixels) have a lower performance than NN11 and NN21 (clusters with only one pixel) respectively. This behavior is counter-intuitive as one would expect the performance to improve with pixel multiplicity. However, it can be explained when considering the cluster variables per pixel multiplicity, shown in Appendix A. The variables Cl_V and V_start show a worse separation between slow pions and electrons as the pixel multiplicity increases, affecting the performance of the neural networks accordingly. This is discussed in detail in Section 5.2.

6.1.3 Comparing Approaches

To compare the performance of NN0 with each of the specifically trained neural networks (NNXY), NN0 is tested with the data subsets. This way it can be seen if the specialized neural networks help reach a significantly higher efficiency. The results are shown in Table 6.4. In addition, the comparison is presented in Figure 6.7. The efficiency values of the general neural network is compared to those of the specific neural networks in the ROC curves to determine if the specialized neural networks have a significant performance improvement.

We choose the efficiency values to be compared to be those obtained when the respective neural networks have a rejection rate of 90%. The reason for this choice is because when the neural networks are implemented, their cut will be chosen purely based on rejection rate. Table 6.5 shows the comparison of the general and specific neural networks efficiency and their statistical significance.

Since the two neural networks were tested with different samples, the statistical significance, t, is calculated as follows:

$$t = \frac{A - B}{\sqrt{(\delta A)^2 + (\delta B)^2}}. \tag{6.5}$$

Here, A and B are the efficiencies measured for NN0 and NNXY and δA and δB are their respective uncertainties.

Table 6.4: Performance of NN0 tested on specific samples. Cut chosen to reach a 90% QED electrons rejection rate.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]
DATA11	0.457	88.15 ± 0.15	86.36 ± 0.08	89.57 ± 0.07	89.94 ± 0.07
DATA12	0.589	86.38 ± 0.11	82.76 ± 0.06	89.22 ± 0.05	90.00 ± 0.05
DATA13	0.570	88.77 ± 0.15	87.51 ± 0.07	89.78 ± 0.07	90.04 ± 0.07
DATA14+	0.339	93.16 ± 0.16	96.25 ± 0.04	90.64 ± 0.07	90.06 ± 0.07
DATA21	0.726	87.91 ± 0.17	85.74 ± 0.09	89.62 ± 0.08	90.07 ± 0.07
DATA22	0.765	85.98 ± 0.14	81.97 ± 0.08	89.11 ± 0.07	98.99 ± 0.06
DATA23	0.680	88.60 ± 0.19	87.13 ± 0.10	89.78 ± 0.09	90.09 ± 0.09
DATA24+	0.138	94.44 ± 0.21	98.80 ± 0.03	90.88 ± 0.09	90.08 ± 0.09

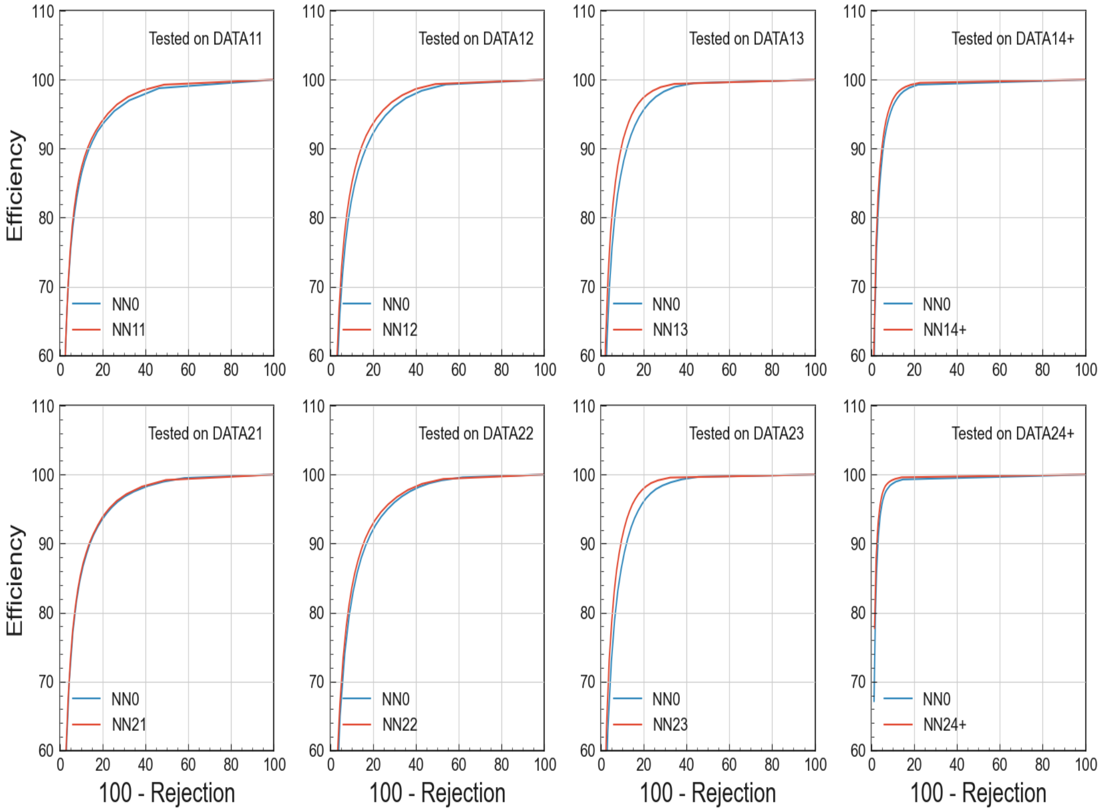


Figure 6.7: ROC curves comparing the performance of NN0 with that of each specialized neural network (trained on clusters with specific layer number and pixel multiplicity). The neural networks were tested on the data sub-samples specified in Figure 5.10.

Since almost all cases in Table 6.5 have a significance $> 3\sigma$, we conclude that the efficiency obtained with a set of neural networks trained on specific cases is statistically different than the efficiencies obtained with one neural network trained for all cases. In addition, the improvement obtained by having specialized neural networks is significant, typically of order 1 to 2%, and should be considered when implementing the algorithm into hardware.

Table 6.5: Comparing the efficiency of NN0 and specific neural networks when tested on specific data samples. The statistical significance is calculated.

Data sample	Neural network	Efficiency [%]	Statistical significance [σ]
DATA11	NN0	86.36 ± 0.08	6.96
	NN11	87.10 ± 0.07	
DATA12	NN0	82.76 ± 0.06	26.05
	NN12	84.97 ± 0.06	
DATA13	NN0	87.51 ± 0.07	33.44
	NN13	90.82 ± 0.07	
DATA14+	NN0	96.25 ± 0.04	15.20
	NN14+	97.11 ± 0.04	
DATA21	NN0	85.74 ± 0.09	2.67
	NN21	86.08 ± 0.09	
DATA22	NN0	81.97 ± 0.08	14.23
	NN22	83.58 ± 0.08	
DATA23	NN0	87.13 ± 0.10	34.74
	NN23	91.06 ± 0.08	
DATA24+	NN0	98.80 ± 0.03	10.37
	NN24+	99.24 ± 0.03	

6.2 Monte Carlo - Early Phase 3 Background

The neural networks were also tested with data containing slow pions and so-called early phase 3 (EP3) background. In this way, the performance of the neural networks is evaluated on a more realistic scenario in where slow pions must be distinguished not only from QED electrons but also from beam background. Early phase 3 background refers to the beam background during the Belle II early phase 3 running period, considered to be light background conditions. This time, the average number of clusters per event in the entire PXD is 550. The data sample used (referred to DATA-BG) contains about 5 million clusters produced by background and almost 10,000 slow pion clusters. The PXD cluster variable distributions are shown in Appendix B.

6.2.1 General Approach

NN0 performance was tested using DATA-BG and the results and neural network output value distribution are shown in Table 6.6 and Figure 6.8 respectively.

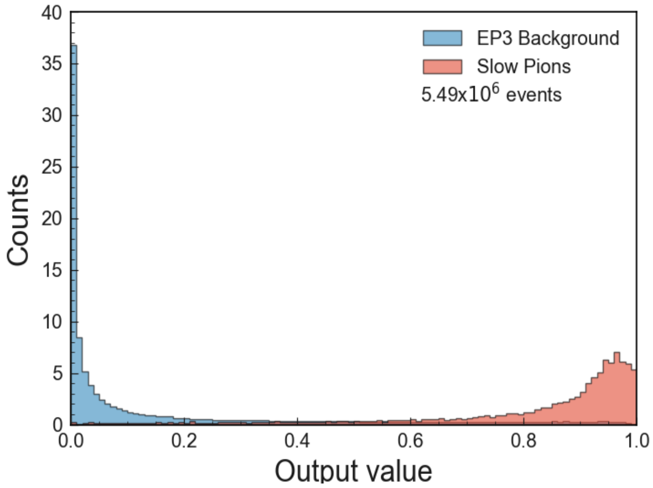


Figure 6.8: NN0 output value distribution when tested on DATA-BG.

Table 6.6: Performance of NN0 for threshold values of 0.5 and 0.605. The latter is the value corresponding to a background rejection rate of 90%. Testing done with DATA-BG.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]	**Loss**
DATA-BG	0.5	88.22 ± 0.48	89.34 ± 0.31	87.39 ± 0.12	87.11 ± 0.02	0.263
DATA-BG	0.605	87.69 ± 0.47	85.37 ± 0.36	89.52 ± 0.10	90.01 ± 0.01	0.263

6.2.2 Specific Approach

The data sample DATA-BG was subdivided according to layer number and pixel multiplicity. For clearness, the sub-samples are referred to as DATA-BGXY, where X is the layer number and Y is the pixel multiplicity ($X = 1, 2$ and $Y = 1, 2, 3, 4+$). Since there are around 500 background events per slow pion event, the sub-samples contains mostly background events, as shown in Table 6.7. Therefore, the results have to be normalized such that the number of slow pions and background events is the same, as explained in Chapter 6.

Table 6.7: No. of slow pion and background events for each of the DATABG sub-samples.

Data sample	Background events	Slow pion events
DATA-BG11	993,142	1,738
DATA-BG12	1,406,806	3,254
DATA-BG13	753,351	1,779
DATA-BG14+	1,759,155	1,645
DATA-BG21	156,818	412
DATA-BG22	193,197	633
DATA-BG23	94,111	249
DATA-BG24+	130,854	156

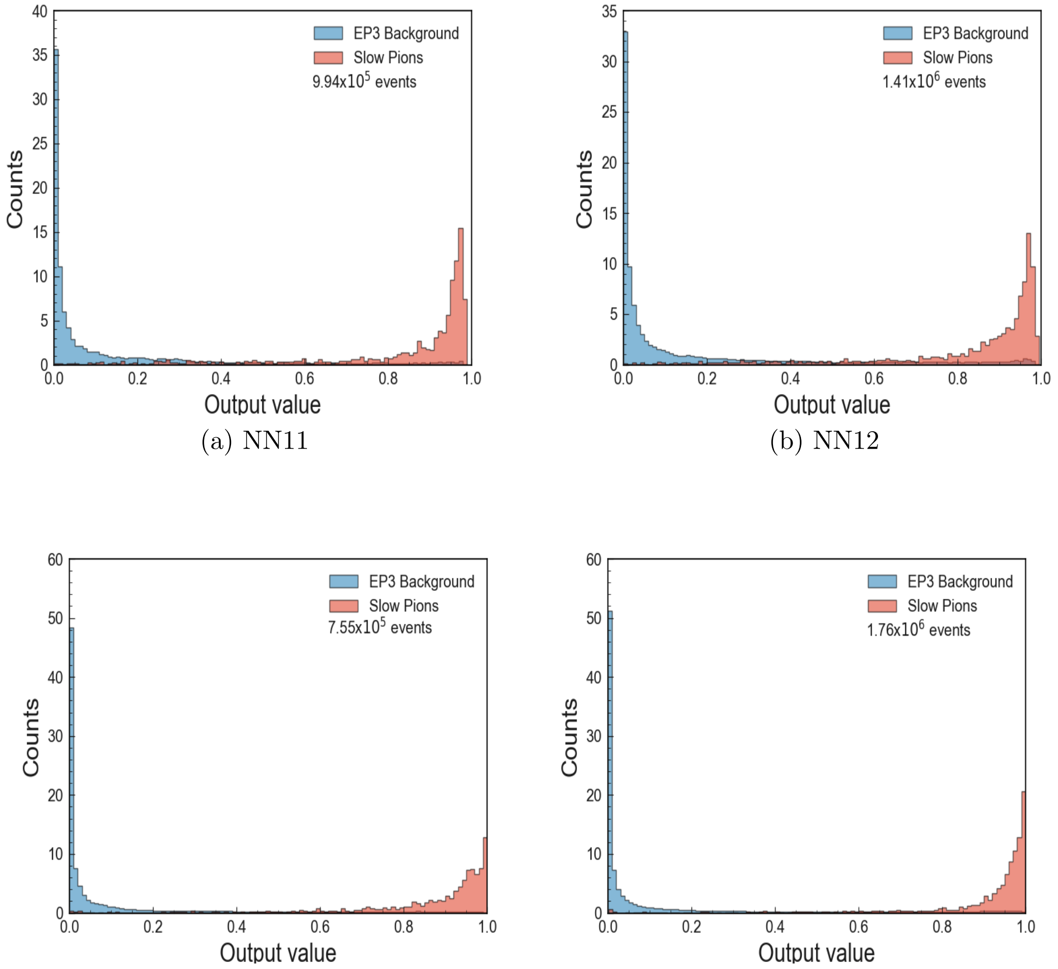
Tables 6.8 and 6.9 thus show the normalized performance values of the specialized neural networks for a 0.5 cut and for a 90% rejection rate respectively. Also, Figures 6.9 and 6.10 show the neural network output value distributions for specific pixel multiplicity at layer 1 and layer 2 respectively.

Table 6.8: Specific NN performances for a 0.5 cut. Performance at 90% rejection of EP3 background. Results are normalized.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]	**Loss**
DATA-BG11	0.5	89.57 ± 1.13	88.72 ± 0.76	90.26 ± 0.23	90.42 ± 0.03	0.277
DATA-BG12	0.5	87.51 ± 0.83	88.81 ± 0.55	86.56 ± 0.22	86.21 ± 0.03	0.292
DATA-BG13	0.5	90.92 ± 1.14	92.86 ± 0.61	89.40 ± 0.24	88.99 ± 0.04	0.231
DATA-BG14+	0.5	91.88 ± 1.20	94.65 ± 0.55	89.69 ± 0.24	89.11 ± 0.02	0.164
DATA-BG21	0.5	88.92 ± 2.35	91.02 ± 1.41	87.35 ± 0.58	86.82 ± 0.09	0.287
DATA-BG22	0.5	85.63 ± 1.91	91.94 ± 1.08	81.63 ± 0.63	79.31 ± 0.09	0.302
DATA-BG23	0.5	86.84 ± 3.02	90.36 ± 1.87	84.41 ± 0.88	83.31 ± 0.12	0.223
DATA-BG24+	0.5	96.23 ± 3.98	98.72 ± 0.90	94.04 ± 0.46	93.74 ± 0.07	0.114

Table 6.9: Specific NN performances. Performance at 90% rejection of EP3 background. Results are normalized.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]	**Loss**
DATA-BG11	0.480	89.71 ± 1.14	89.41 ± 0.74	89.96 ± 0.23	90.01 ± 0.03	0.277
DATA-BG12	0.655	86.42 ± 0.80	82.82 ± 0.66	89.25 ± 0.19	90.03 ± 0.03	0.292
DATA-BG13	0.545	90.94 ± 1.14	91.85 ± 0.65	90.21 ± 0.22	90.04 ± 0.04	0.231
DATA-BG14+	0.545	92.00 ± 1.20	93.98 ± 0.59	90.40 ± 0.22	90.02 ± 0.02	0.164
DATA-BG21	0.620	88.84 ± 2.31	87.62 ± 1.62	89.81 ± 0.49	90.06 ± 0.08	0.287
DATA-BG22	0.814	83.09 ± 1.74	76.15 ± 1.69	88.42 ± 0.47	90.03 ± 0.07	0.302
DATA-BG23	0.734	86.17 ± 2.88	82.33 ± 2.42	89.18 ± 0.68	90.02 ± 0.10	0.223
DATA-BG24+	0.238	94.68 ± 3.99	99.36 ± 0.64	90.85 ± 0.67	90.00 ± 0.08	0.114



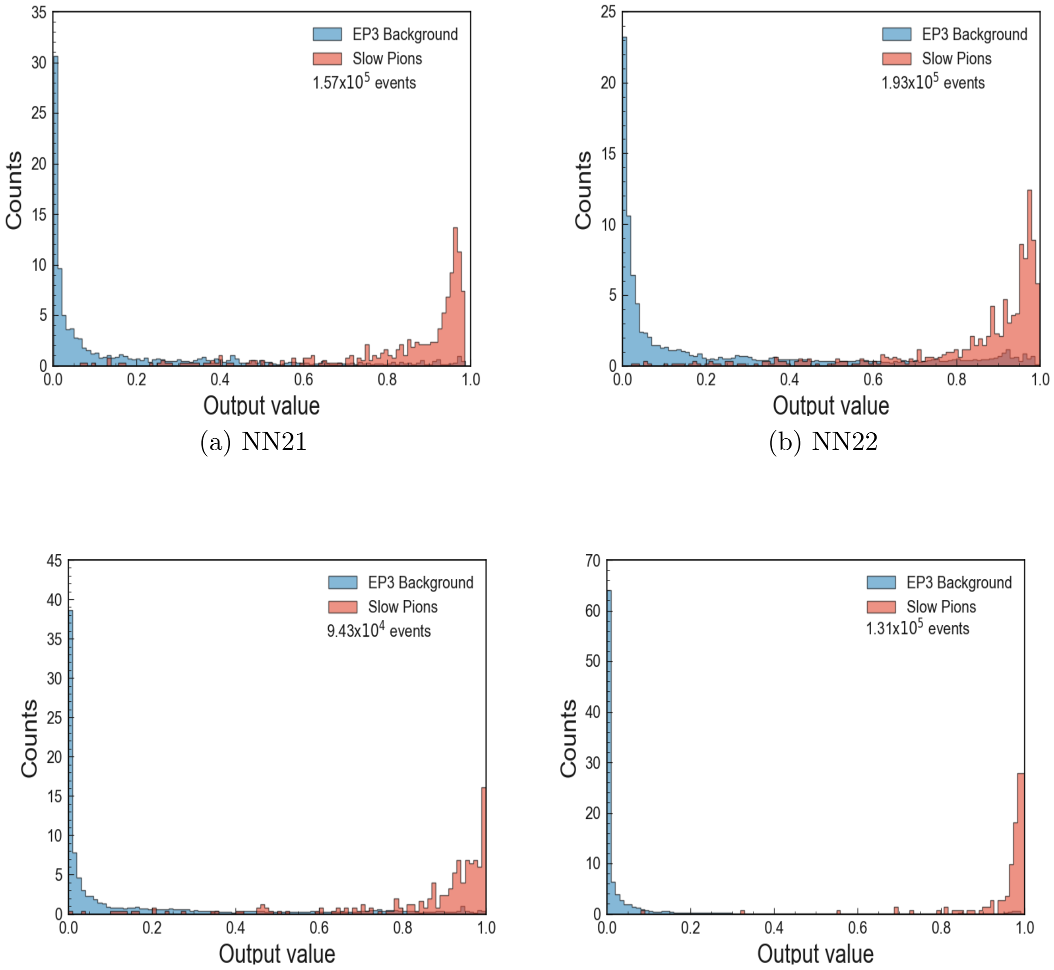
(a) NN11

(b) NN12

(c) NN13

(d) NN14+

Figure 6.9: Output value distributions for neural networks trained on data samples with PXD layer number = 1 and pixel multiplicities = 1, 2, 3 and 4+. (a) NN11 tested on DATA-BG11 (b)NN12 tested on DATA-BG12 (c)NN13 tested on DATA-BG13 (4)NN14 tested on DATA-BG14+.



(a) NN21

(b) NN22

(c) NN23

(d) NN24+

Figure 6.10: Output value distributions for neural networks trained on data samples with PXD layer number = 2 and pixel multiplicities = 1, 2, 3 and 4+. (a) NN21 tested on DATA-BG21 (b)NN22 tested on DATA-BG22 (c)NN23 tested on DATA-BG23 (4)NN24 tested on DATA-BG24+.

6.2.3 Comparing EP3 Background and QED Electron Background

This time, we compare the behavior of the neural networks when tested on data containing only electrons as background (results from Section 6.1) and on EP3 background data. This gives a better idea about how much a different background affects the performance of the neural networks. Figures 6.11 and 6.12 show the ROC curves of both data samples when tested with NN0 and each specialized neural network respectively.

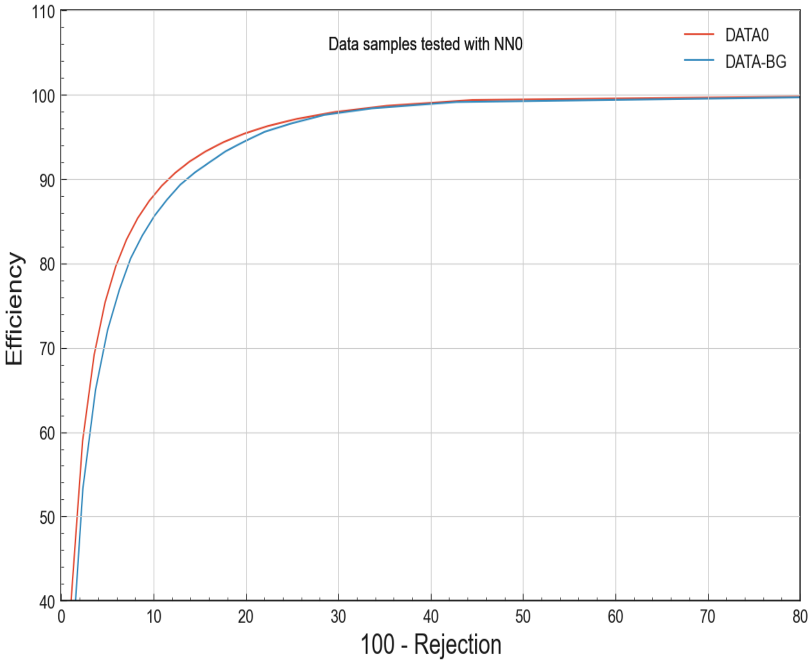


Figure 6.11: ROC curves comparing the performance of the general neural networks when tested on DATA0 (QED electrons as background) and DATA-BG (EP3 background).

Figure 6.11 shows that the performance of NN0 decreases only slightly when discriminating slow pions from EP3 background. This behavior is expected as the background sample also includes kaons ($K^{\pm}$) and pions ($\pi^{\pm}$) which have similar distributions as the slow pions. This is because these particles are decay products from $B\bar{B}$ events and therefore will be boosted, resulting in similar cluster variables distributions to that of the slow pions. Consequently, the neural network will tend to classify these as slow pions, apparently decreasing the efficiency. Moreover, the same conclusion can be drawn when looking at Figure 6.12 with the exception of the data samples tested with NN11, NN12 and NN21. In these cases, the data samples containing EP3 background have a slightly better efficiency for a rejection close to 80%.

Overall, it can be concluded that the neural networks trained to discriminate between slow pions and QED electrons also perform well when tested with additional background. The reason for this can be attributed to the fact that differences in the cluster variables can also be found between these 2 classes (see Appendix B).

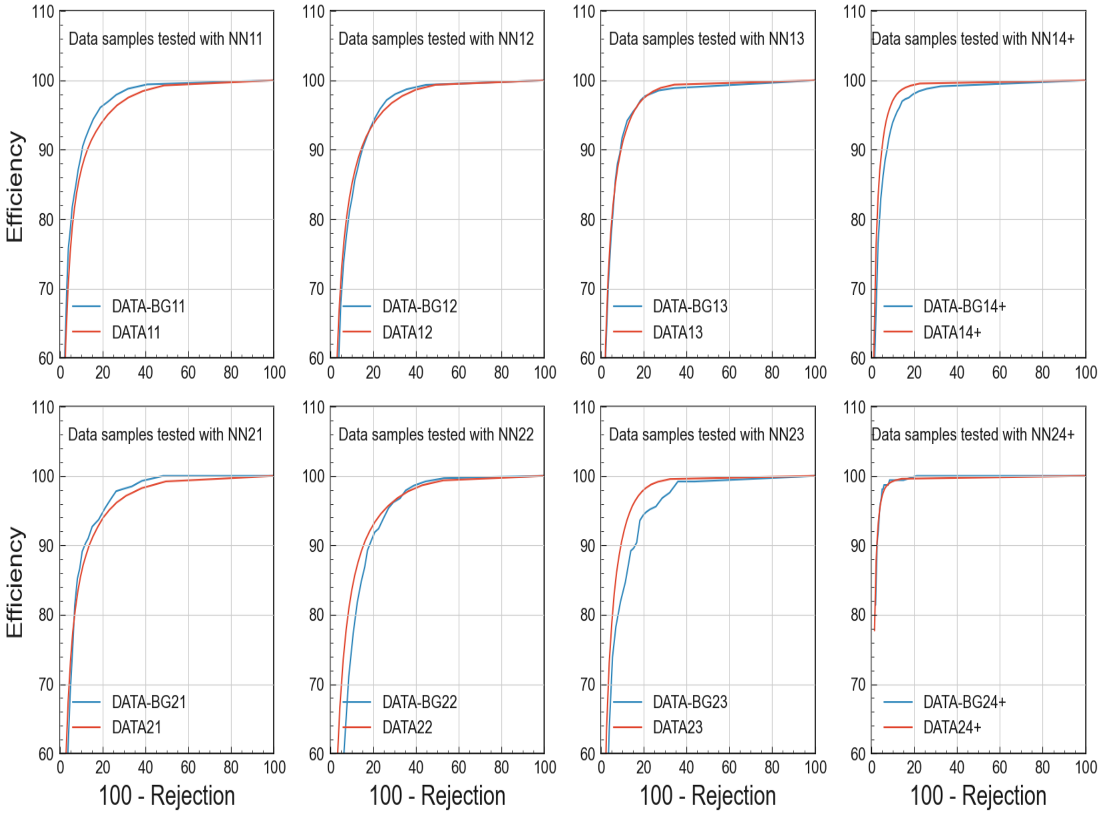


Figure 6.12: ROC curves comparing the performance of the specialized neural networks when tested with QED electrons as background and with EP3 background.

6.3 Real Data

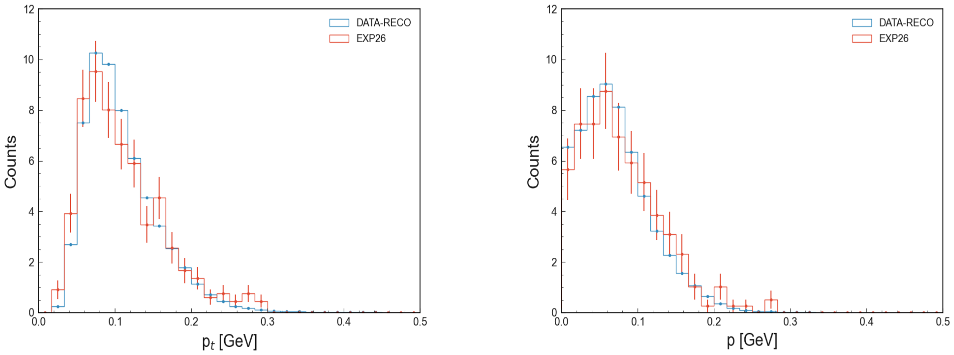
Slow pion samples can be classified into two categories: not reconstructed and reconstructed, or equivalently, into low and high momentum regime. Low momentum slow pions tend to not reach all (or any) layers of the SVD and therefore their tracks are not reconstructed. These are the slow pions that would be lost once the ROI algorithm comes into play and therefore, it is these pions that we aim to recover with the developed algorithm. On the other hand, slow pions with momentum high enough traverse all four layers of the SVD and therefore their tracks should be reconstructed.

When dealing with MC-generated data, it is known whether a cluster belongs to a slow pion or not, independently of whether the particle's track has been reconstructed or not. This is not the case when dealing with real data. Only the reconstructed tracks (and therefore the corresponding PXD cluster) can be associated to a particle. At the moment, there is no way to know which of the clusters in the PXD was generated by a slow pion without a track reconstruction. Therefore, looking into real data means looking exclusively at slow pions with reconstructed tracks. What is more, the required data needs to have information about both track reconstruction and PXD variables in order to select D^{*}, find the reconstructed slow pions and get the PXD cluster information. This type of data is known as Calibration Data Summary Table (CDST) format. This type of data, however, is used for calibration and therefore has a very limited sample of events. Neuro CDSTs or Neuro skim data, on the other hand, contain all runs, pre-scaled by a factor of 1000 [31]. For this reason, this Neuro CDST files were used to test the algorithm on real data. Specifically, a data sample referred to as EXP26, which was running in April 2022 until the end of June, was selected. The integrated luminosity is quite small due to data pre-scaling and therefore, only a small sample with around 400 slow pion events was obtained.

To accurately compare the performance of the developed algorithm when tested on real data in comparison with tested on MC samples, the MC-sample must be a subset containing only reconstructed slow pions. This subset comes from DATA0 and it is referred to as DATA-RECO. Since the previous results did not take into account whether the slow pion was reconstructed or not, new results must be produced exclusively for DATA-RECO. Moreover, due to low statistics, only the general approach will be studied in the following sections.

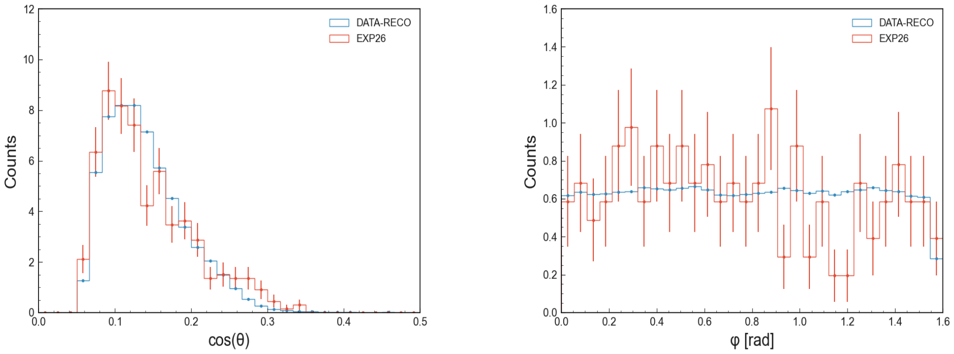
6.3.1 Kinematic and Cluster Variables - Real Data

The kinematic variables as well as PXD cluster variables of the data sample EXP26 are plotted with those of DATA-RECO to see whether they are similar. The results for the kinematic variables are shown in Figures 6.13 and 6.14. The cluster variables are shown in figure 6.15. No statistical differences can be seen in the p_t, $|\vec{p}|$, $\cos\theta$ and ϕ distributions of both data samples.



(a) (b)

Figure 6.13: MC reconstructed (5.5×10^5 events) and EXP26 (397 events) slow pions (a) Transverse momentum (b) Absolute momentum.



(a) (b)

Figure 6.14: MC reconstructed (5.5×10^5 events) and EXP26 (397 events) slow pions (a) cosine of the polar angle θ (b) azimuth angle ϕ.

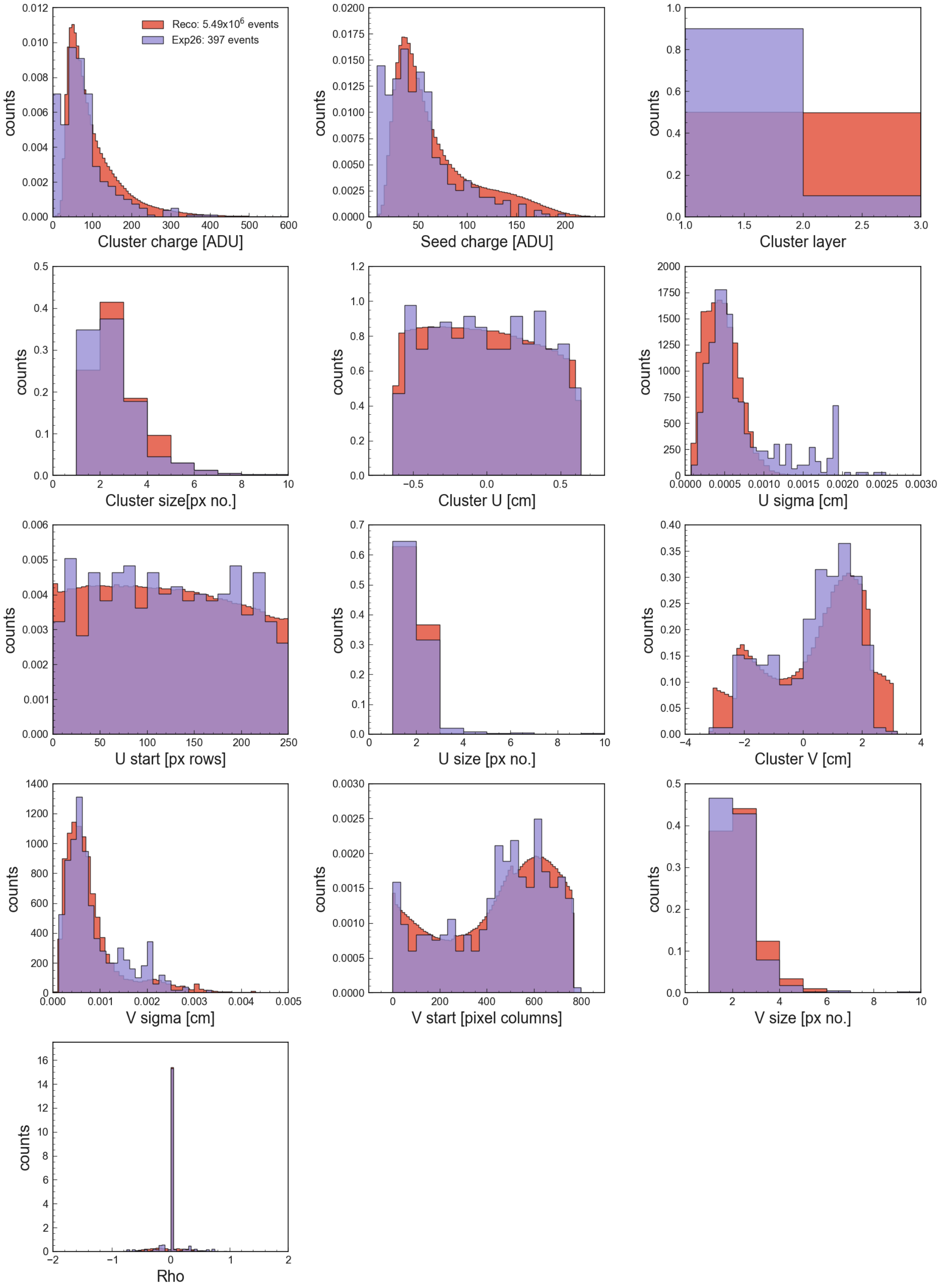


Figure 6.15: PXD cluster variables of slow pions coming from EXP26 and reconstructed, MC-generated slow pions.

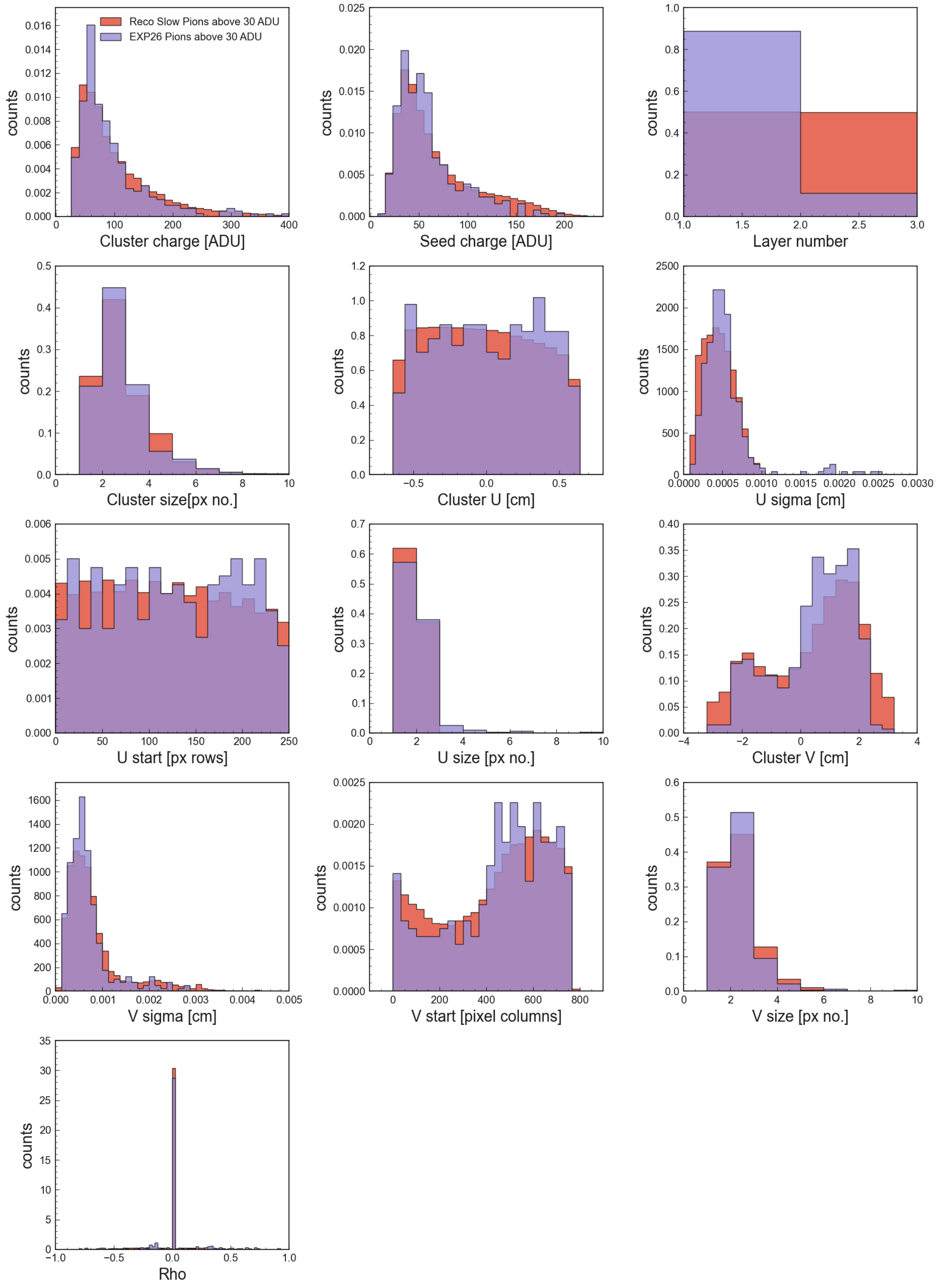


Figure 6.16: PXD cluster variables of slow pions coming from EXP26 and reconstructed, MC-generated slow pions. Showing slow pions with a charge greater than 30 ADU.

The cluster variables of EXP26 and DATA-RECO slow pions overall look very similar. Differences can be found on the layer number, in which clusters from EXP26 slow are mostly found in the first layer. This is because the real data was taken with a partly equipped PXD (only 2 of the 12 outer sensors). In addition, EXP26 shows a significant number of PXD clusters with very small charge. This is very likely to be due to noise in the PXD. To clean the real data, a cut on clusters with small charge (below 30 ADU) was applied. A comparison of EXP26 and DATA-RECO for charge > 30 ADU is shown in figure 6.16. A better agreement between real data and MC cluster variables can be seen.

6.3.2 NN Performance with Reconstructed Slow Pions

The neural network NN0 was tested on DATA-RECO and the output distribution for the slow pions is shown in Figure 6.17. A summary of the NN performance is given in Table 6.10.

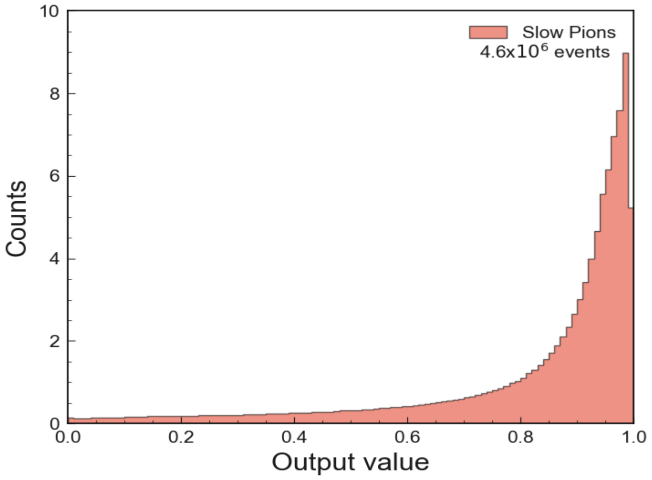


Figure 6.17: NN0 output value distribution of the reconstructed slow pions from the testing sample DATA-RECO.

Table 6.10: Performance of NN0 for threshold value of 0.5.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Loss**
DATA-RECO	0.5	88.72 ± 0.03	89.98 ± 0.01	0.263

6.3.3 NN Performance with Real Data

The EXP26 file, which contains 397 events, was used to test how the NNs perform under real data. In addition, a sample with only slow pions with charge above 30 ADU, referred to as EXP26-CLEAN (with 320 events), was also used for testing. The results are presented in Figures 6.18, 6.19 and Table 6.11.

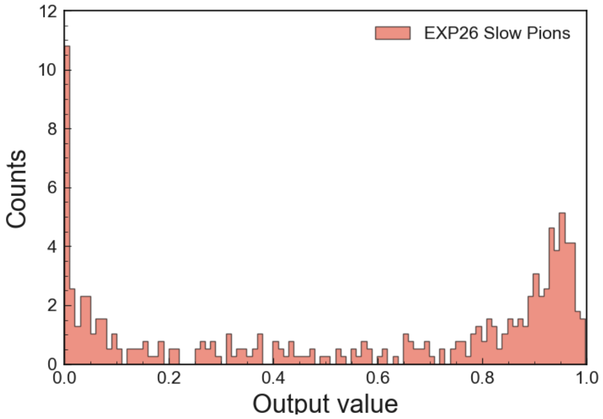


Figure 6.18: NN0 output value distribution when tested on EXP26.

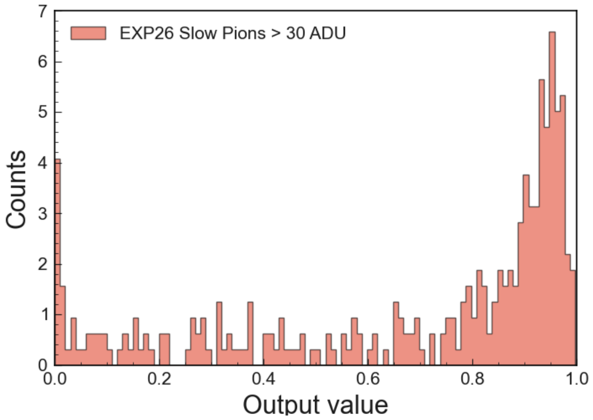


Figure 6.19: NN0 output value distribution when tested on EXP26-CLEAN.

Table 6.11: Performance of NN0 for threshold value of 0.5. Testing done with data samples from EXP26 and EXP26-CLEAN (slow pions with charge above 30 ADU).

Case	**Cut**	**Accuracy [%]**	**Efficiency [%]**	**Loss**
EXP26	0.5	59.23 ± 3.90	59.23 ± 2.47	0.263
EXP26-CLEAN	0.5	73.13 ± 4.78	73.13 ± 2.48	0.263

As it can be seen, the efficiency decreases for real data compared to reconstructed MC-slow pions. A decrease in efficiency is due to a difference between MC and real data, which is always expected. The differences in the cluster variables can be seen in Figure 6.15. Nonetheless, a significant improvement in efficiency is achieved when dropping those clusters with very small charge, attributed to noise (shown in Figure 6.16).

A way to further improve the efficiency is to train neural networks on real data. In this manner, the distributions of real slow pions can be studied by the NN and therefore, a better performance can be achieved for real data. This is one of the main outlooks of this project.

7 Conclusion and Outlook

This thesis presents the development of a PXD stand-alone slow pion rescue algorithm, providing a way of recovering valuable data for flavor tagging and thus, contributing to the search of CP violation measurements. The development of this algorithm is an important step, allowing to exploit the PXD not only to measure the position of particles, but also to perform particle identification

Using Monte-Carlo data, information on the PXD clusters parameters was extracted to train a set of neural networks with the purpose of discriminating between slow pions and the expected dominating background from low energy electron-positron pairs. Two approaches were studied: a general one, where one neural network was trained with data containing all pixel clusters (NN0) and a specific one, where 8 neural networks were trained, each specialized on a data-subset based on PXD layer number X and pixel multiplicity Y (NNXY, $X = 1, 2$ and $Y = 1, 2, 3, 4+$). This was done for several reasons: first, it was found that the distributions of certain cluster variables change with pixel multiplicity in a way that can affect the performance of the NNs. Also, specialized neural networks allow for a more careful study of each case which can result in a higher performance. For the construction of the neural networks, different architectures as well as different hyperparameters were tried during the training phase. The presented neural networks in this thesis are the optimized ones, having a compact architecture, required by executing the networks online in hardware, while maintaining a high performance. It is worth mentioning that deep learning was tried, but somewhat surprisingly, no improvement in the neural network outcomes were found. Therefore, neural networks with one hidden layer were enough.

Using both approaches, we presented the results based on 2 threshold values: the first one at 0.5, which splits the neural network outcome probability in equal parts. This cut value provides and idea on how permissive can the threshold be set. The second one is based on the requirement that the PXD data should be reduced by a factor of 10 when the design luminosity at SuperKEKB is reached. In other words, the cut is selected such that 90% of the background is rejected. This cut shows the neural network performance under conservative conditions.

For all cases studied, promising results were found. For NN0, slow pions are found with a (90.73 ± 0.02)% efficiency for a 0.5 threshold and with a (89.13 ± 0.05)% efficiency for a 0.572 cut, which is the threshold corresponding to a background rejection of 90%. For the NNs trained using the specific approach, efficiencies of (87.03 ± 0.07)%, (84.97 ± 0.06)%, (90.82 ± 0.07)% and (97.11 ± 0.04)% for pixel multiplicities 1, 2, 3 and 4+ in the first layer of the PXD were found. Similarly, for the second layer of the PXD, the efficiencies found for the same pixel multiplicities are (86.08 ± 0.09)%, (83.58 ± 0.08)%, (91.06 ± 0.08)% and (99.24 ± 0.03)%.

The same procedure was done with Early Phase 3 background data, which included also "machine-related" beam background. The neural networks proved capable of separating not only slow pions from QED electrons, but also from a more general background. Similar efficiencies as the ones stated above were obtained, proving the developed algorithm to be resilient to a more varied background. Finally, the algorithm was also tested on a small sample of real slow pions where a lower efficiency of (59.23 ± 3.90)% was observed. This can be explained by some differences between the real data and MC cluster distributions. It was also proven that such efficiency improves to (73.13 ± 4.78)% if a cut is made on clusters with sufficiently small charge, getting rid of PXD noise and bringing MC closer to real data.

Further development of this project includes obtaining a real data sample of significant size to test the trained NNs with more statistics. In addition, if the sample is large enough, training on real data can be done as a next step. Finally, the last step would be the implementation of the algorithm on hardware, which is is planned in the coming funding period 2024-2027. To avoid losing valuable data, it is crucial that this algorithm has been implemented by the time the luminosity reaches a critical value and the PXD will dominate the entire data volume of the Belle II experiment.

A Cluster Variable Distributions per Layer No. and Pixel Multiplicity

In the following pages, the PXD cluster variables distributions of MC data are presented for cluster the specific cases: layer 1 and 2, with pixel multiplicity 1, 2, 3 and 4+.

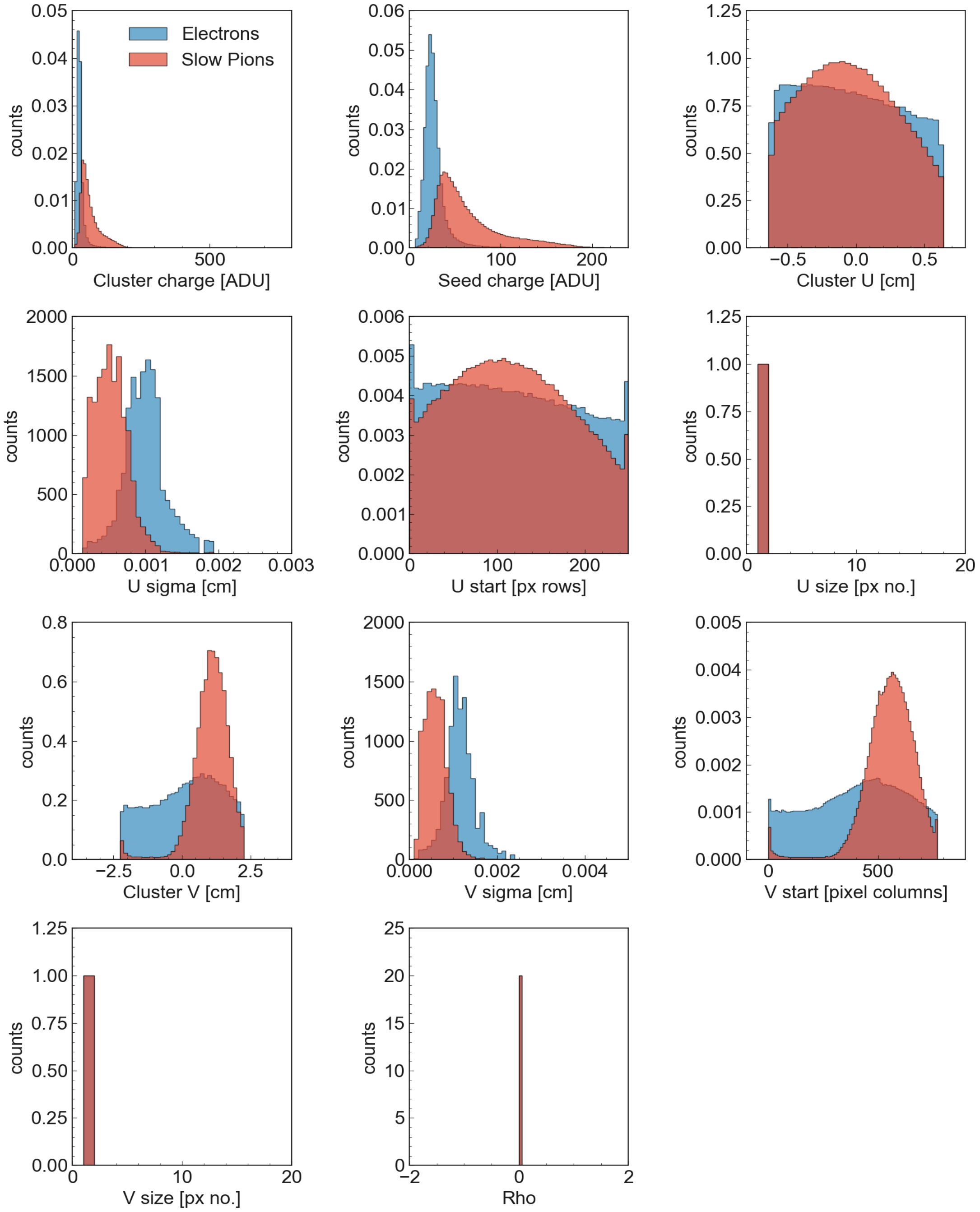


Figure A.1: Cluster variable distributions for clusters with pixel multiplicity 1 in the first PXD layer.

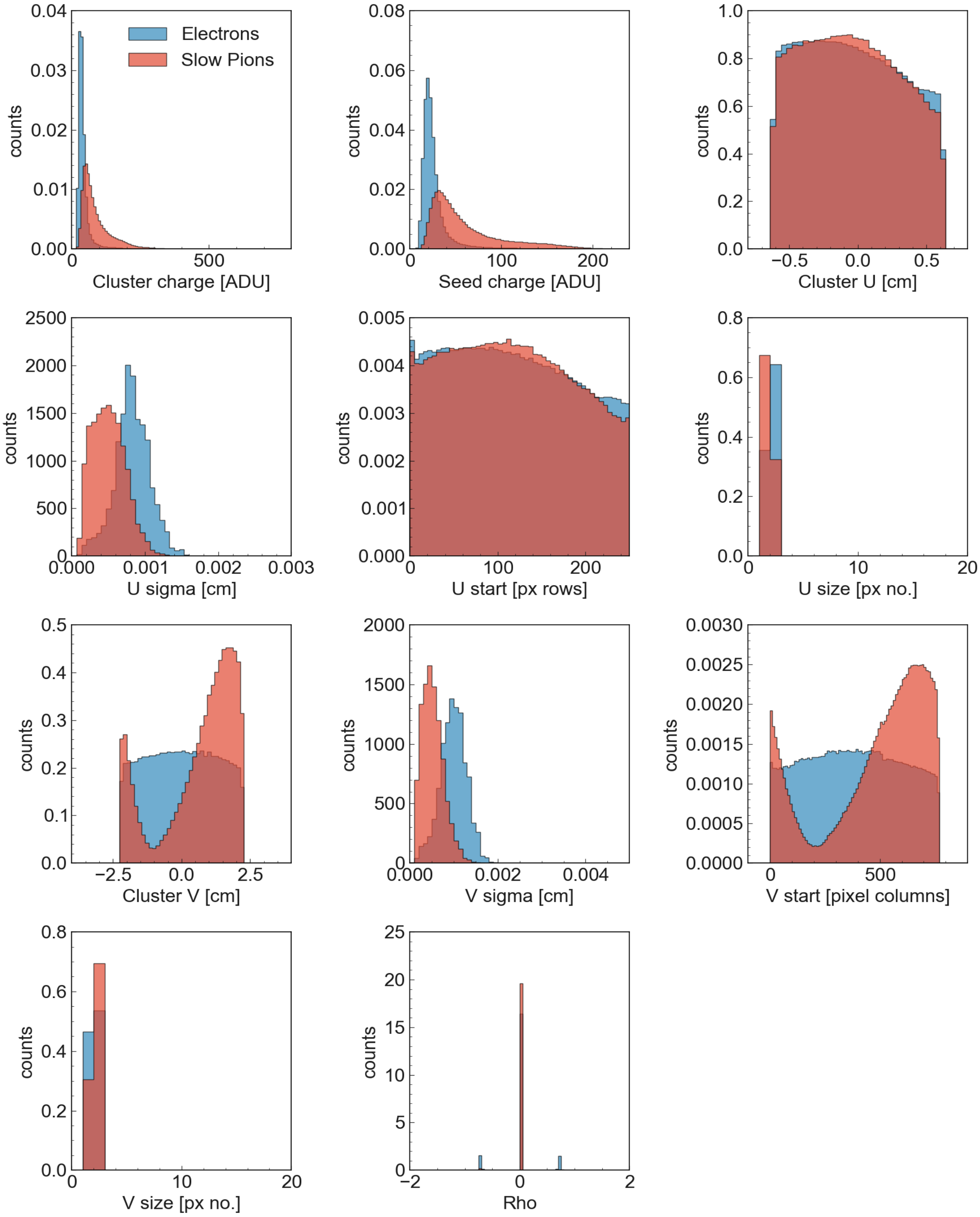


Figure A.2: Cluster variable distributions for clusters with pixel multiplicity 2 in the first PXD layer.

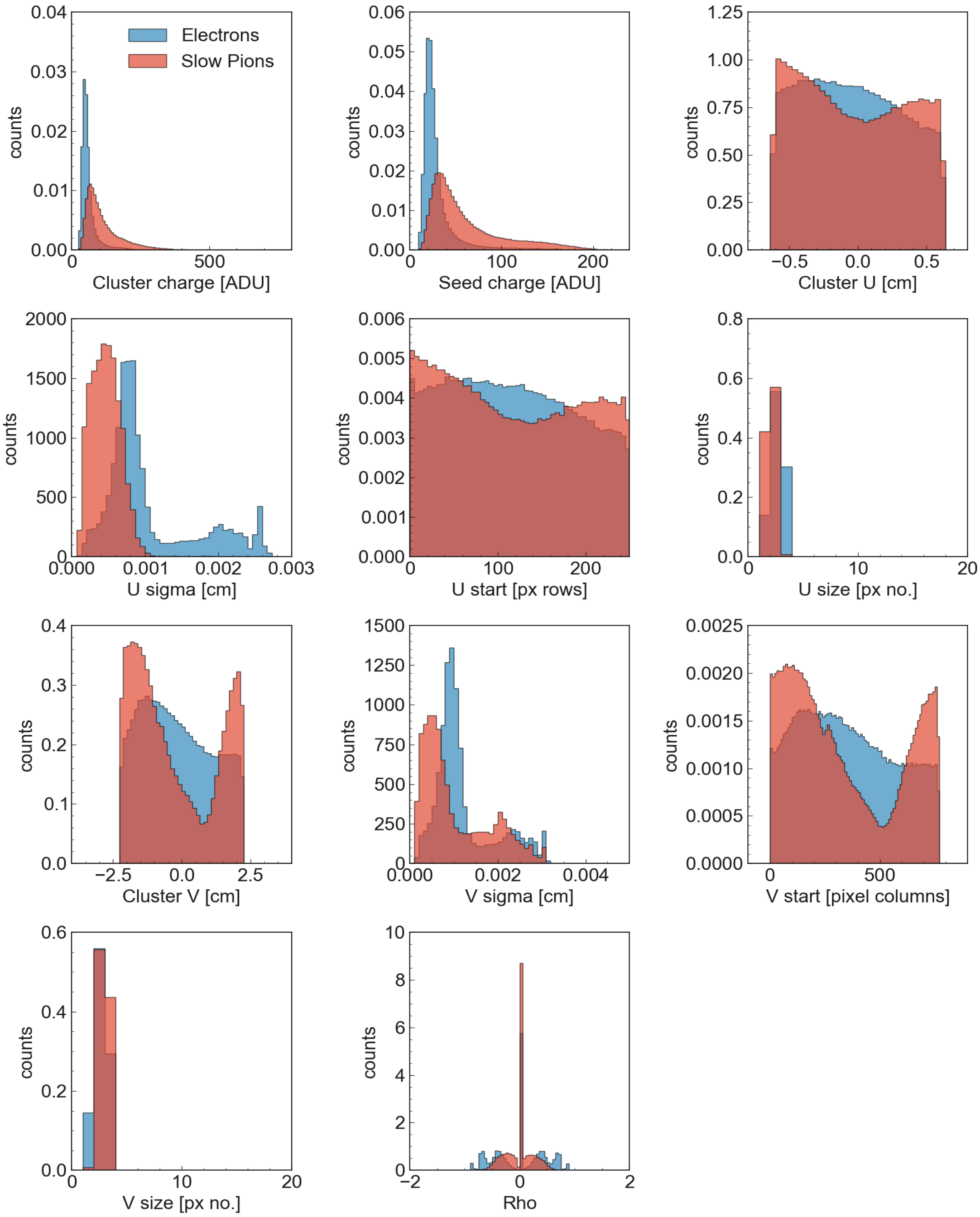


Figure A.3: Cluster variable distributions for clusters with pixel multiplicity 3 in the first PXD layer.

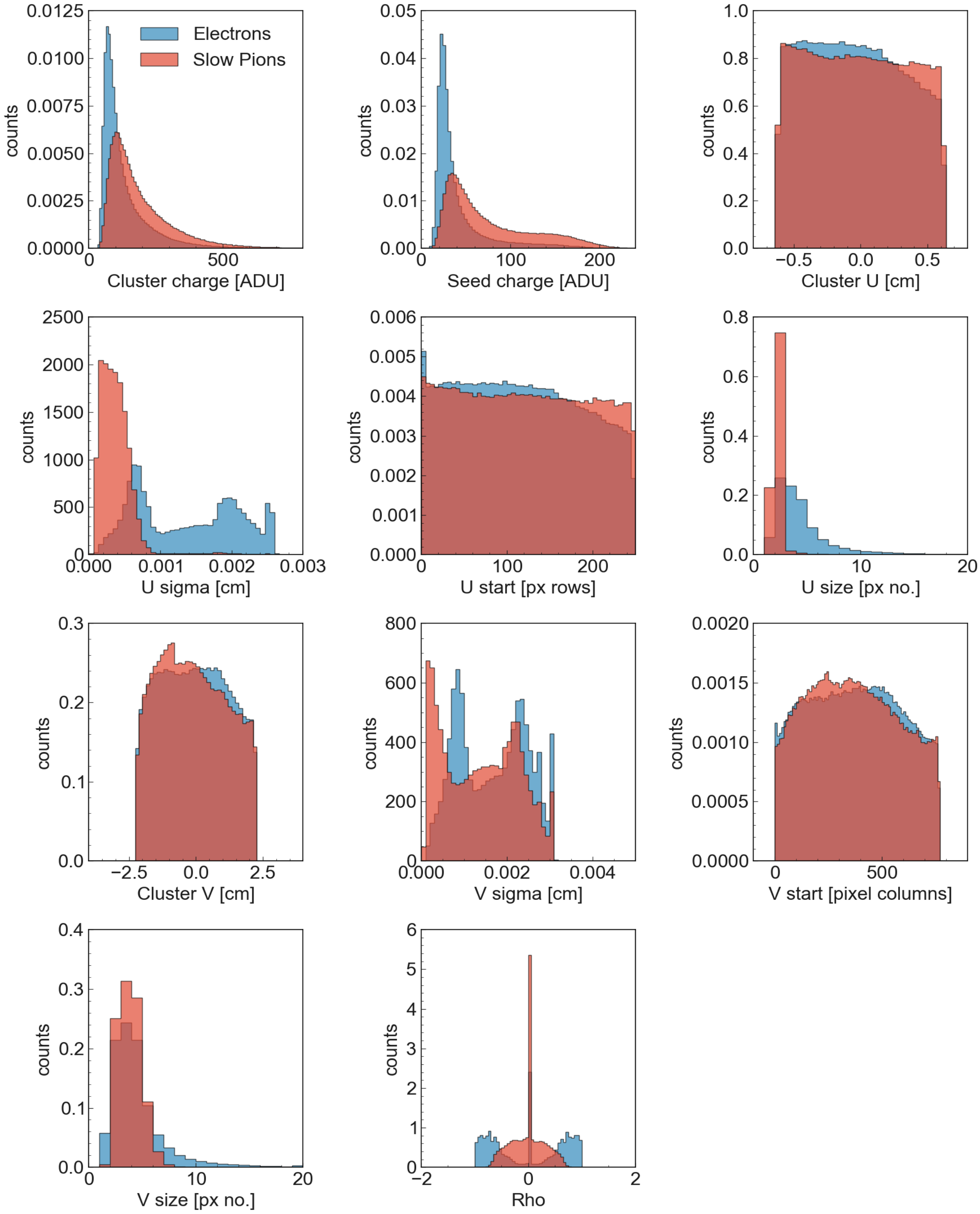


Figure A.4: Cluster variable distributions for clusters with pixel multiplicity 4 or more in the first PXD layer.

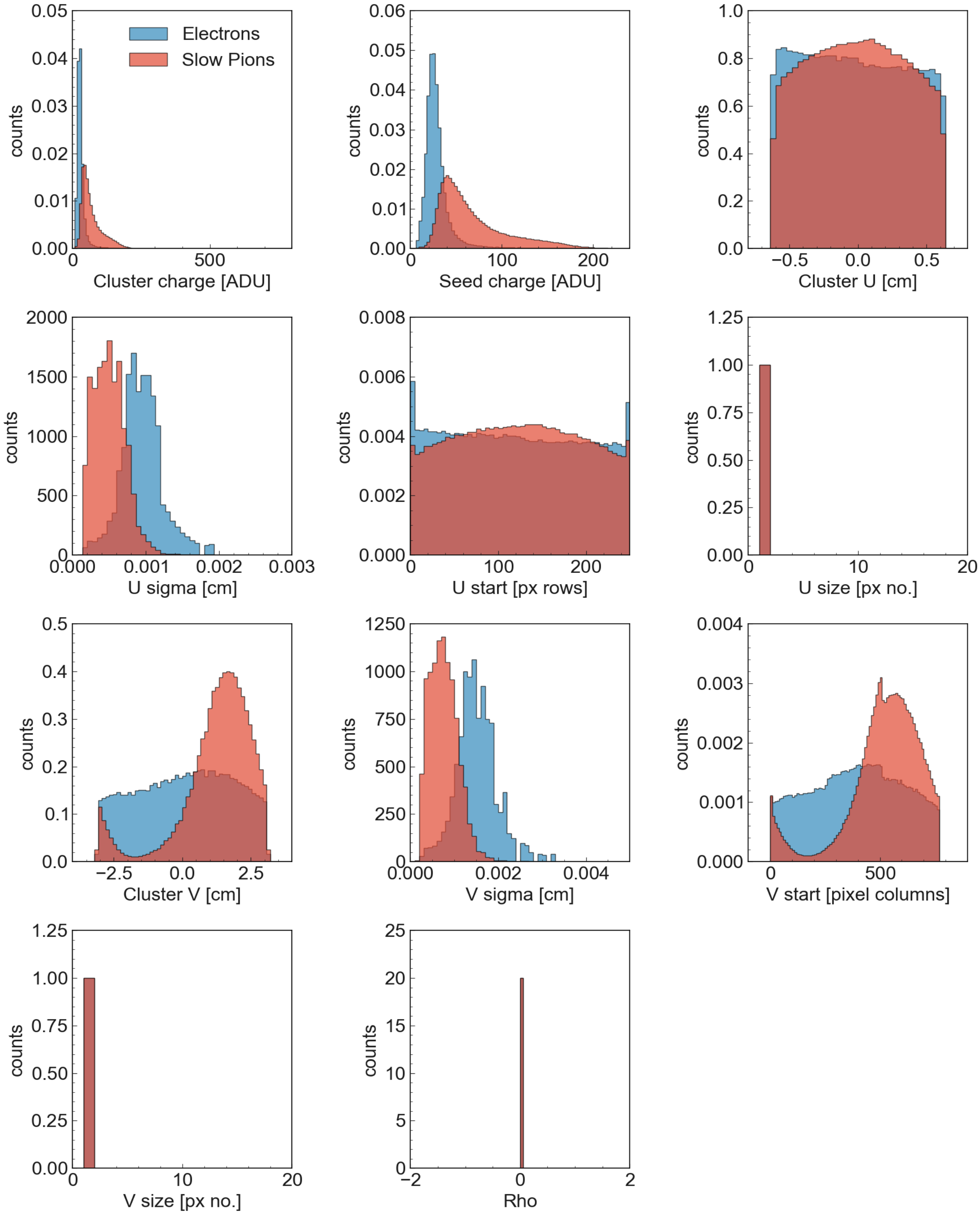


Figure A.5: Cluster variable distributions for clusters with pixel multiplicity 1 in the second PXD layer.

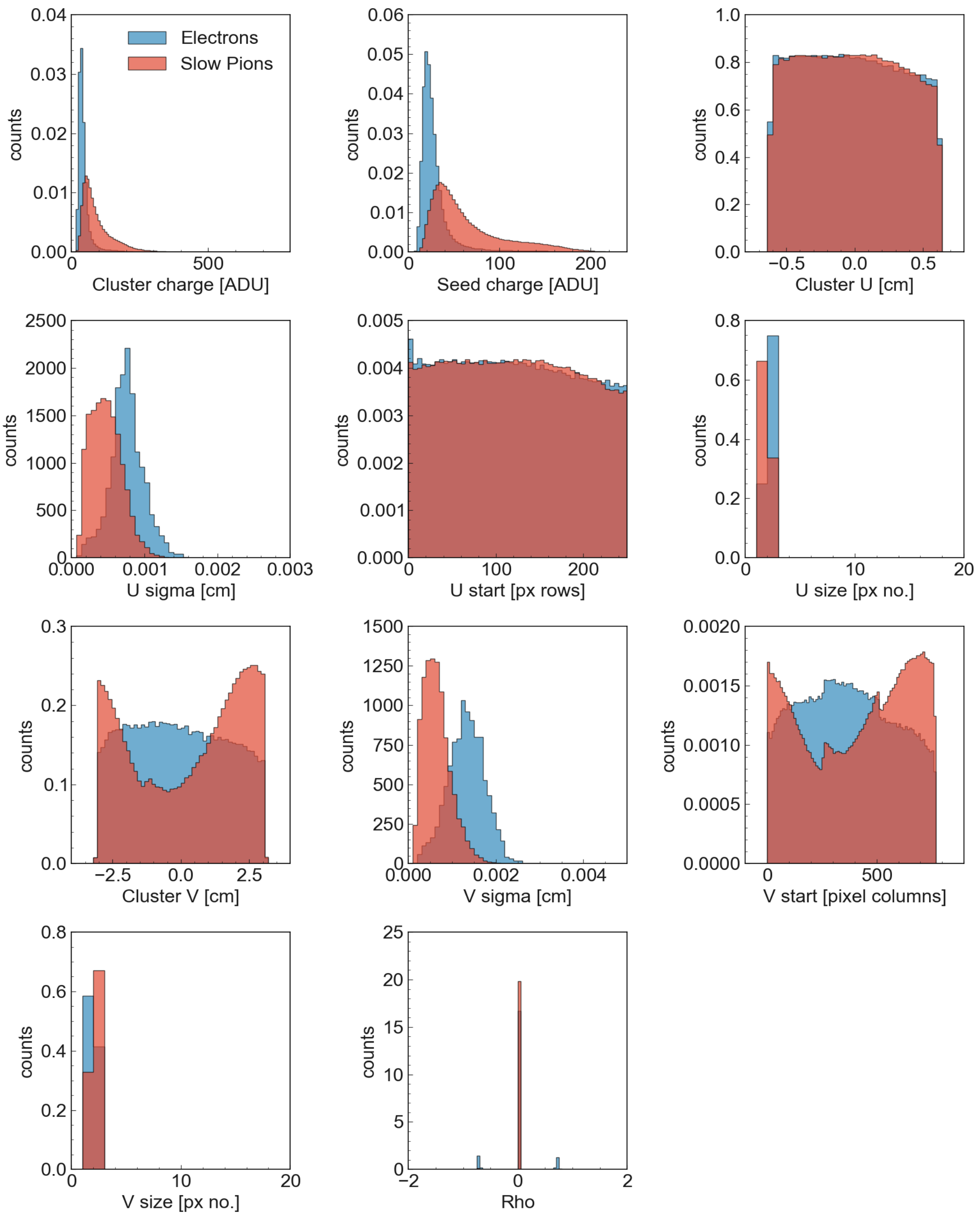


Figure A.6: Cluster variable distributions for clusters with pixel multiplicity 2 in the second PXD layer.

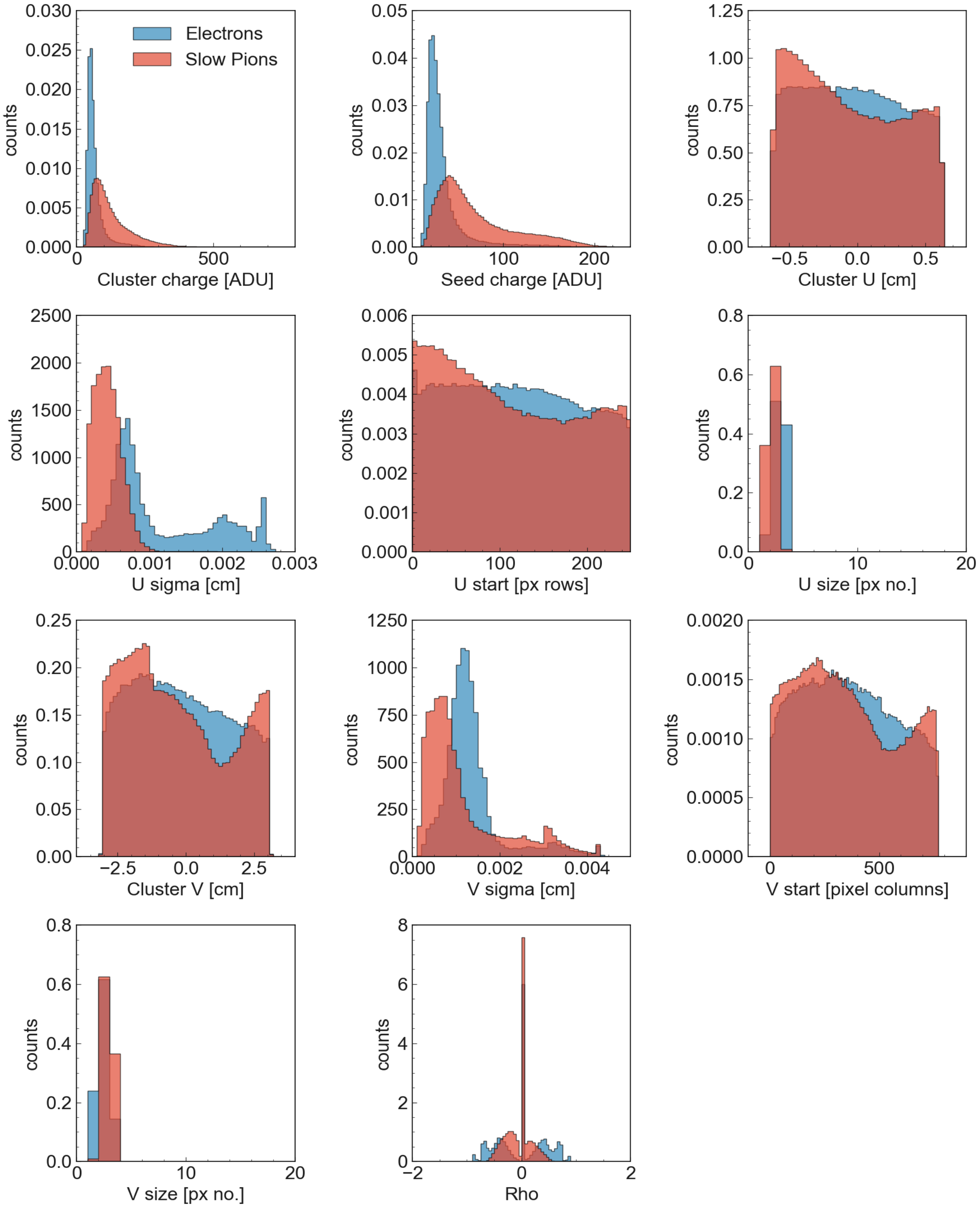


Figure A.7: Cluster variable distributions for clusters with pixel multiplicity 3 in the second PXD layer.

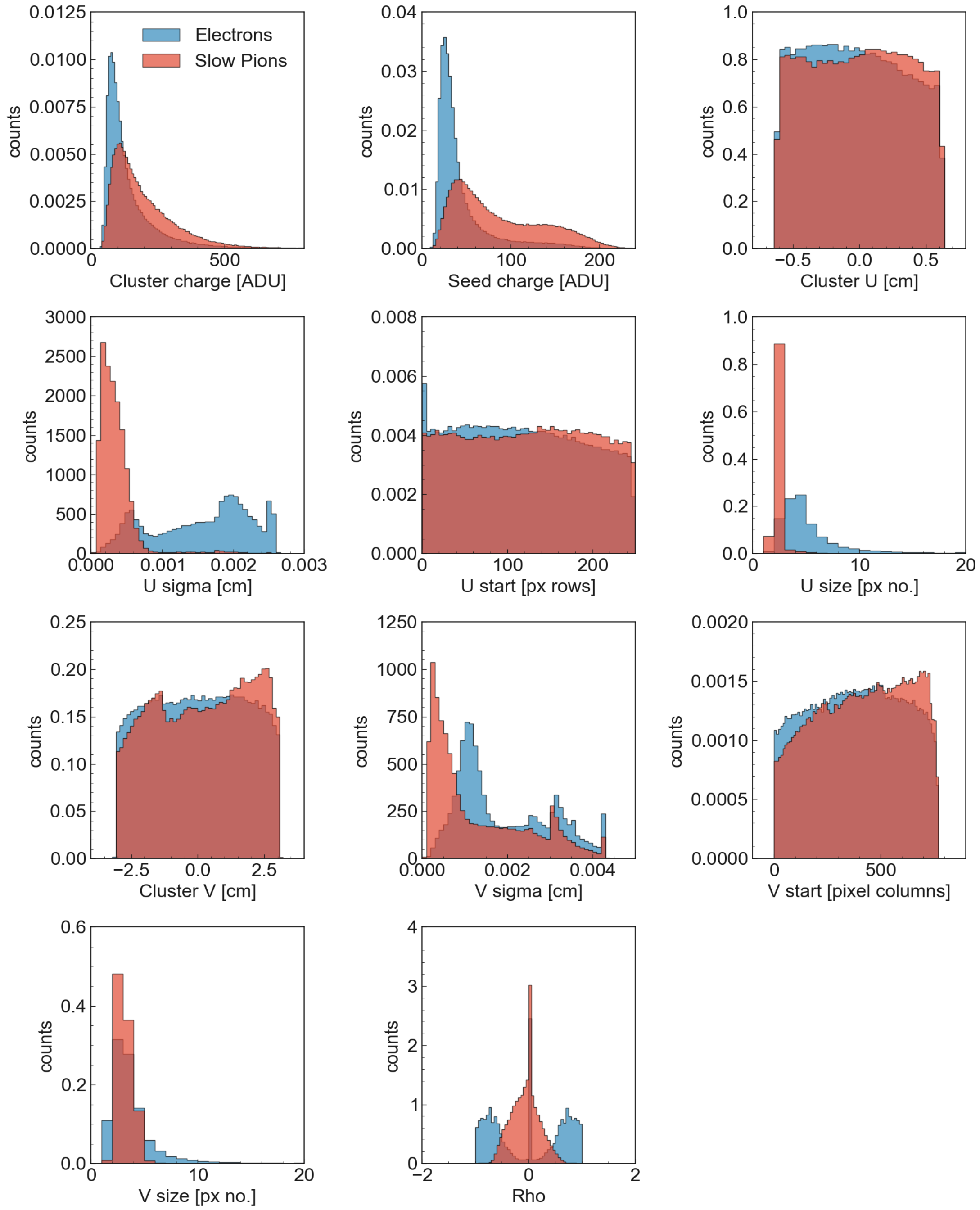


Figure A.8: Cluster variable distributions for clusters with pixel multiplicity 4 or more in the first PXD layer.

B Distribution of Cluster Variables - EP3 background

In the following pages, thePXD cluster variables distributions of MC data with early phase 3 background are presented for general and specific cases. Figure B.1 shows the distributions for all PXD layers and all pixel multiplicities. The other Figures show the cluster variables for cluster layer 1 and 2, with pixel multiplicity 1, 2, 3 and 4+.

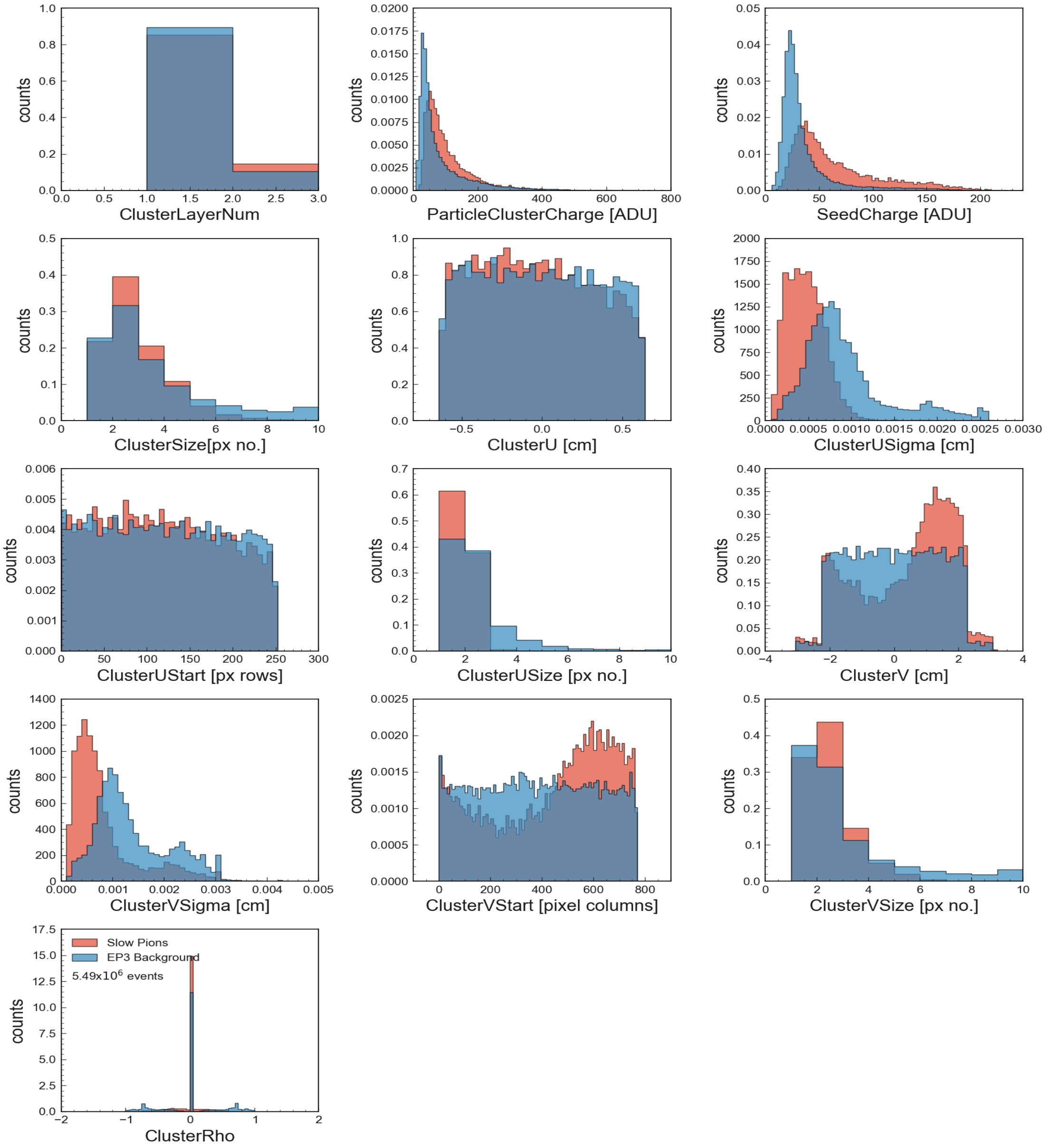


Figure B.1: Cluster variable distributions for slow pions and EP3 background.

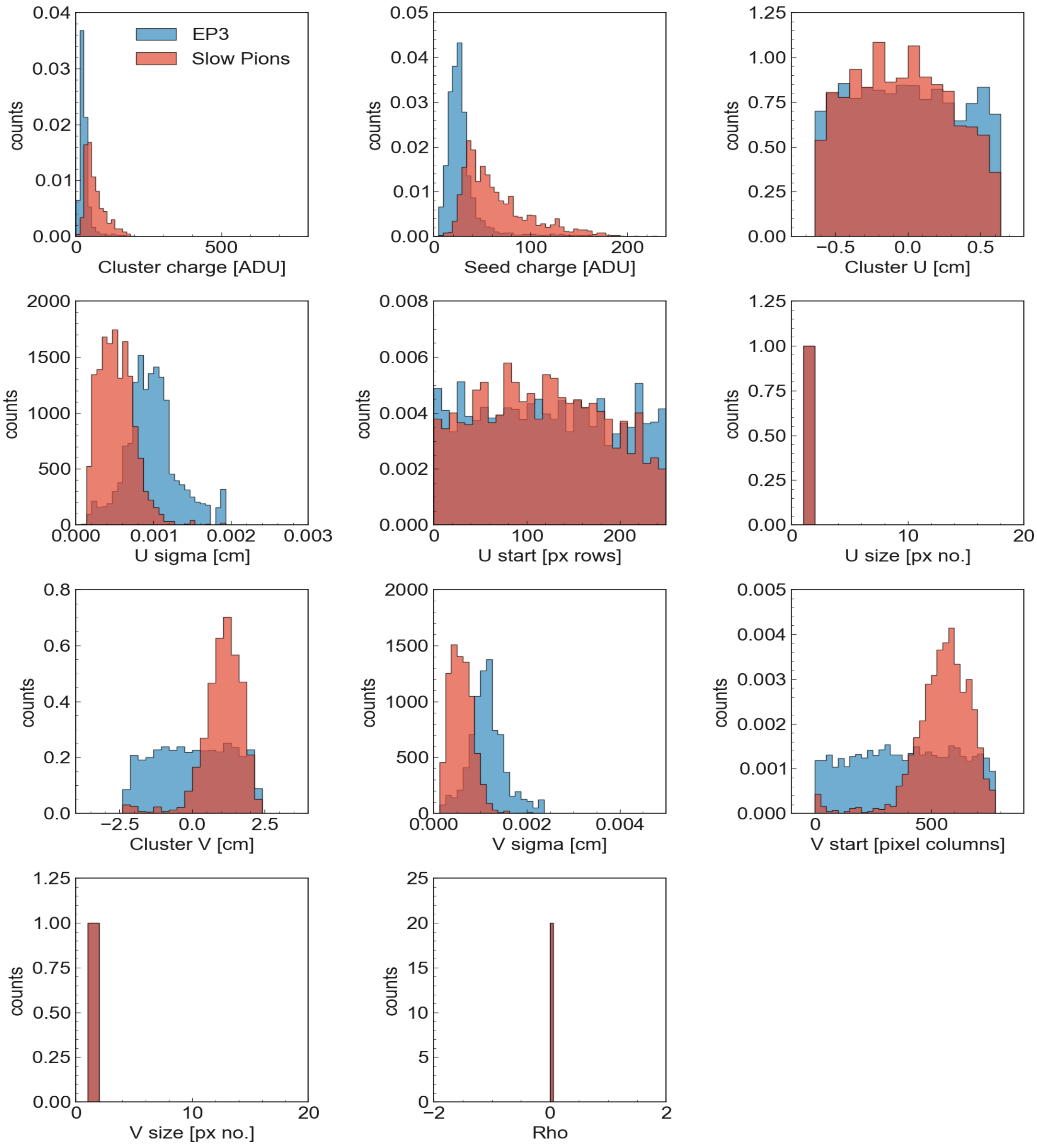


Figure B.2: Cluster variable distributions with pixel multiplicity 1 in the first PXD layer for slow pions and EP3 background.

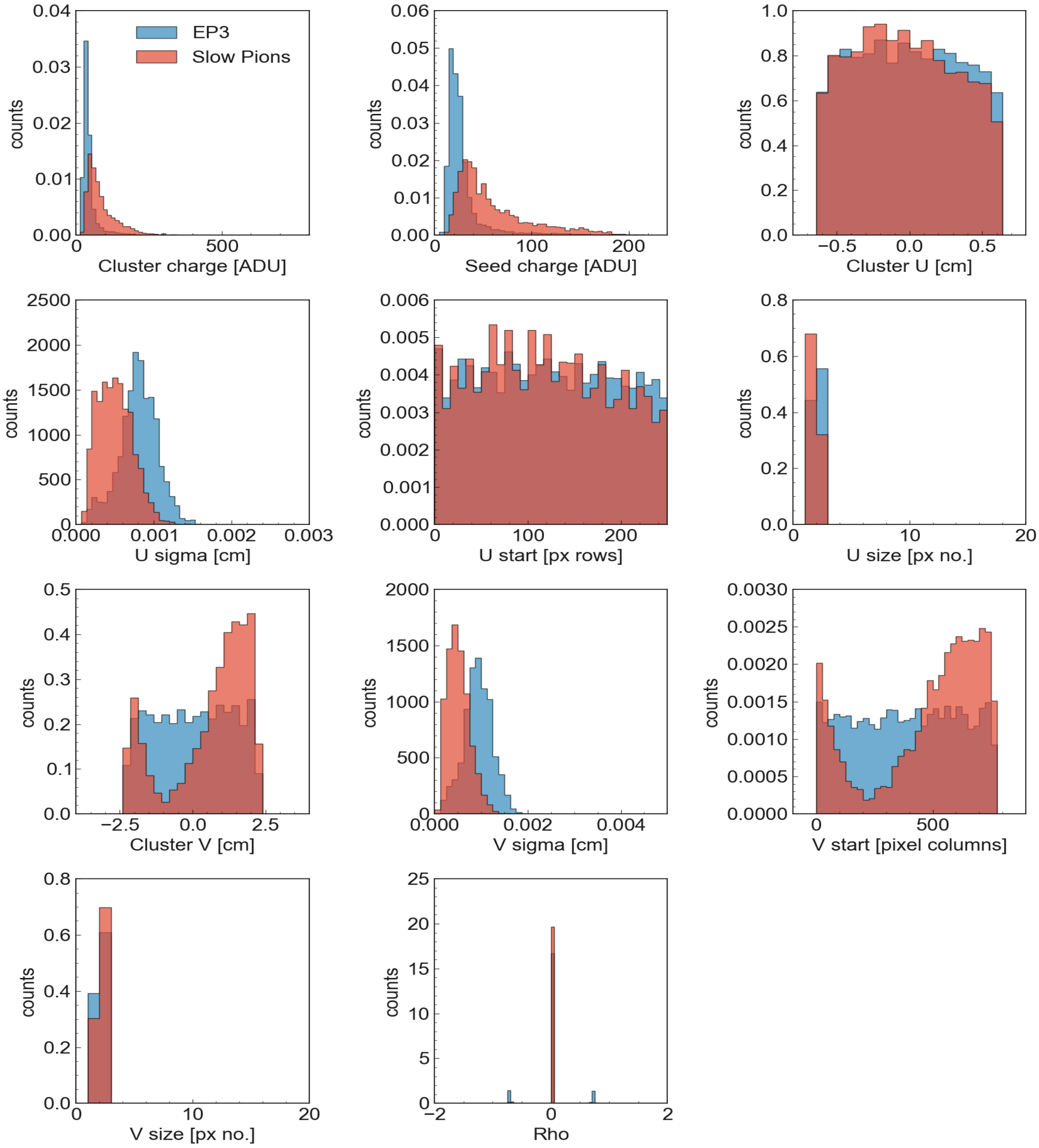


Figure B.3: Cluster variable distributions with pixel multiplicity 2 in the first PXD layer for slow pions and EP3 background.

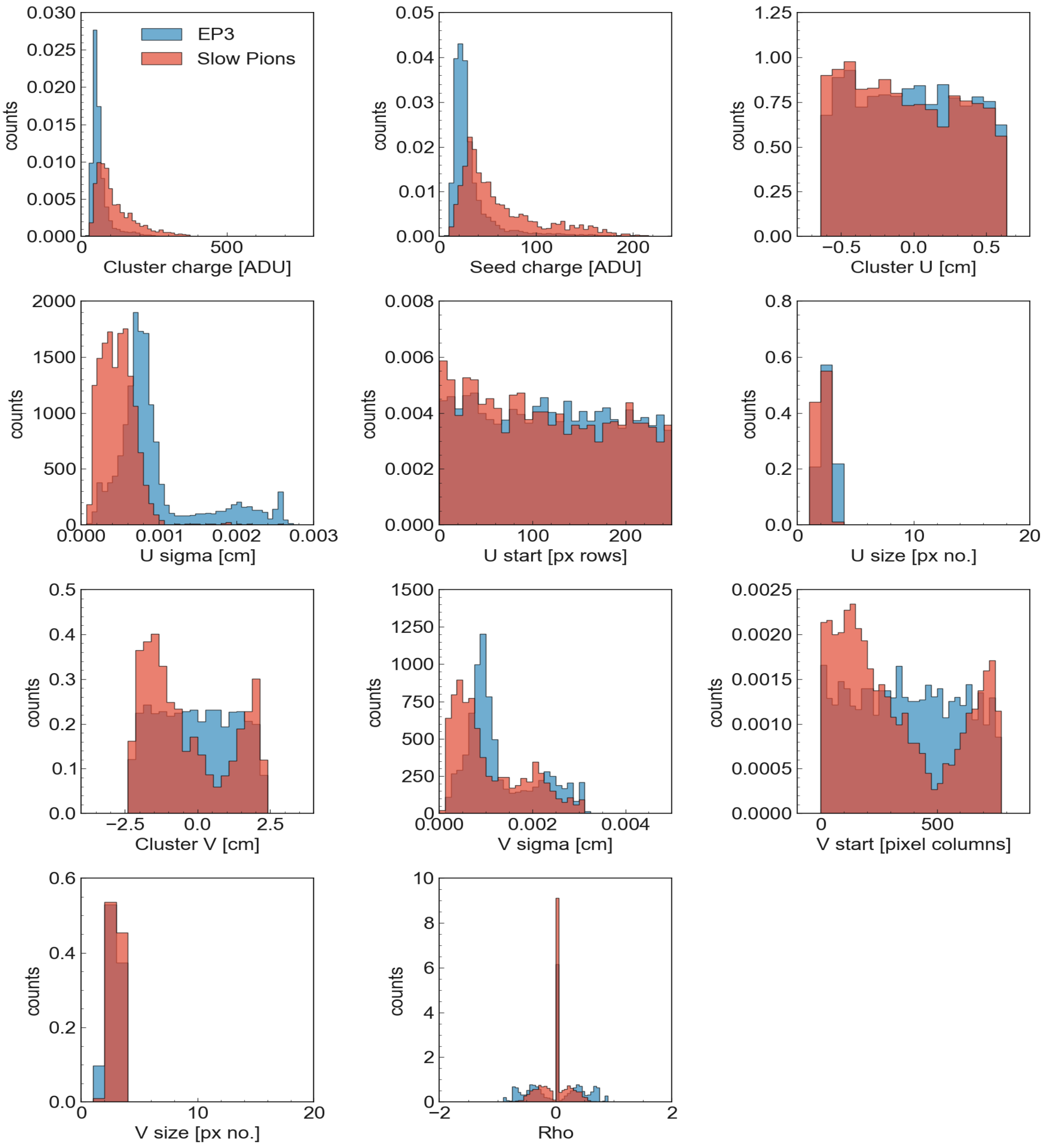


Figure B.4: Cluster variable distributions with pixel multiplicity 3 in the first PXD layer for slow pions and EP3 background.

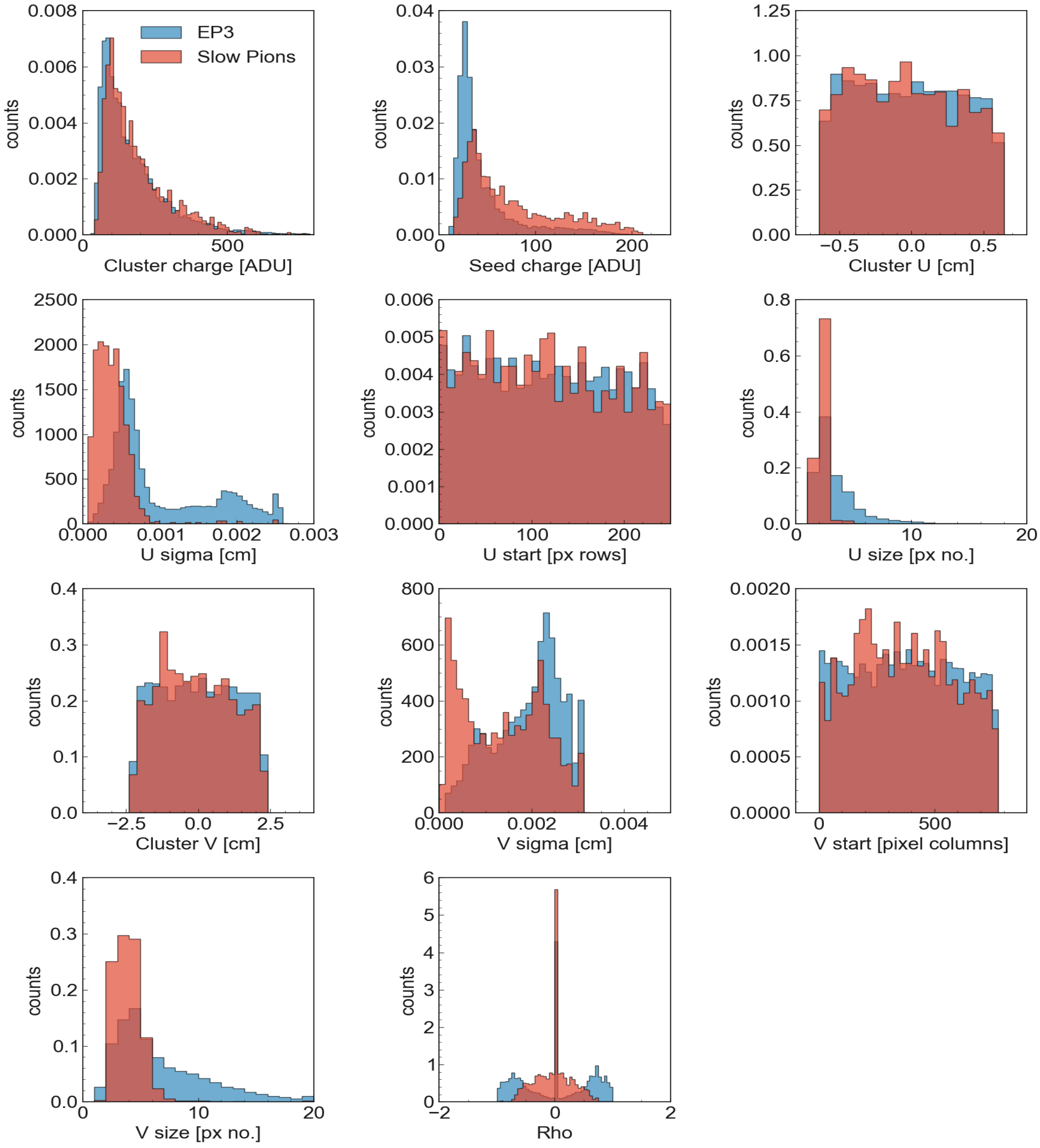


Figure B.5: Cluster variable distributions with pixel multiplicity 4 or more in the first PXD layer for slow pions and EP3 background.

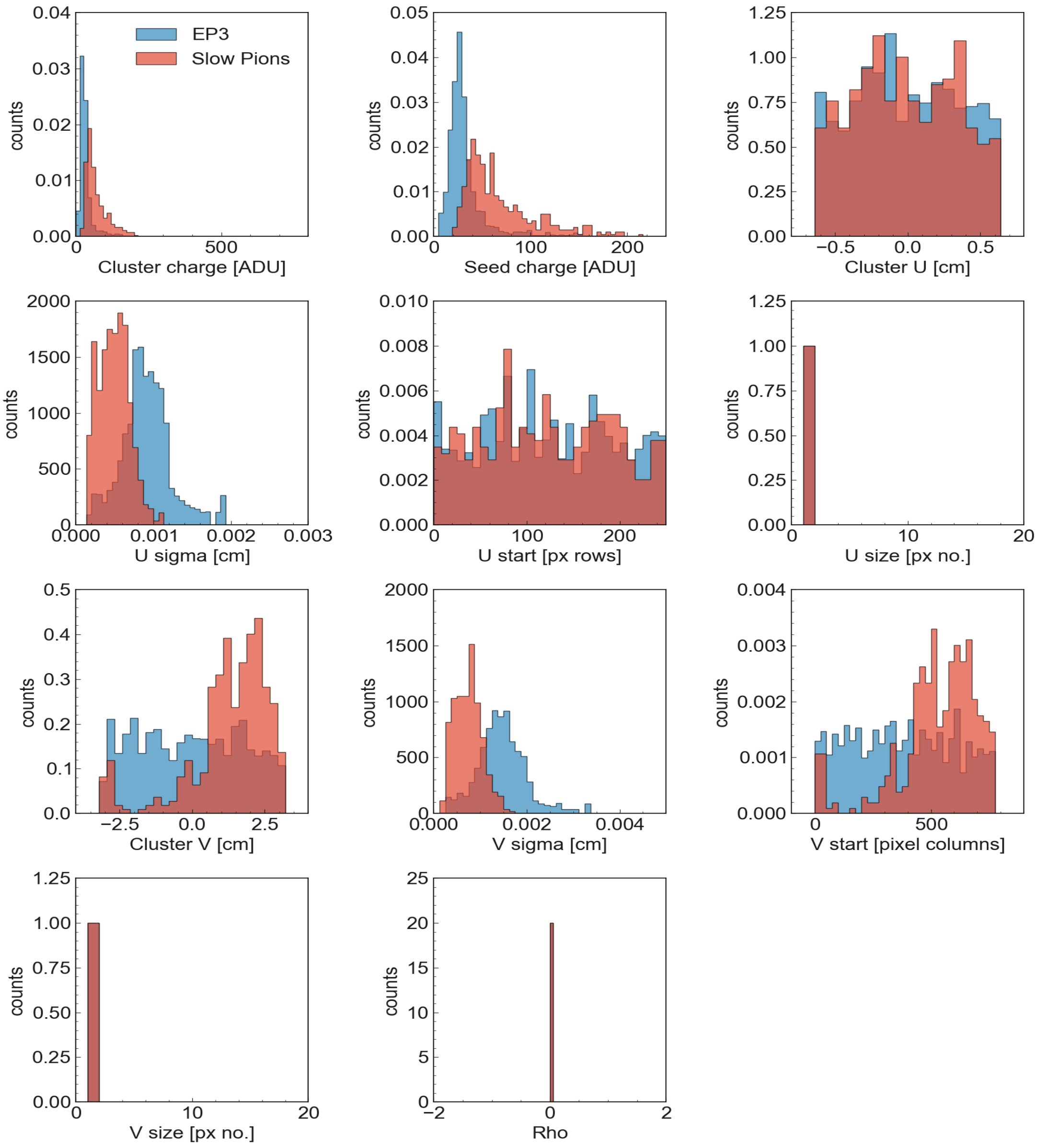


Figure B.6: Cluster variable distributions with pixel multiplicity 1 in the second PXD layer for slow pions and EP3 background.

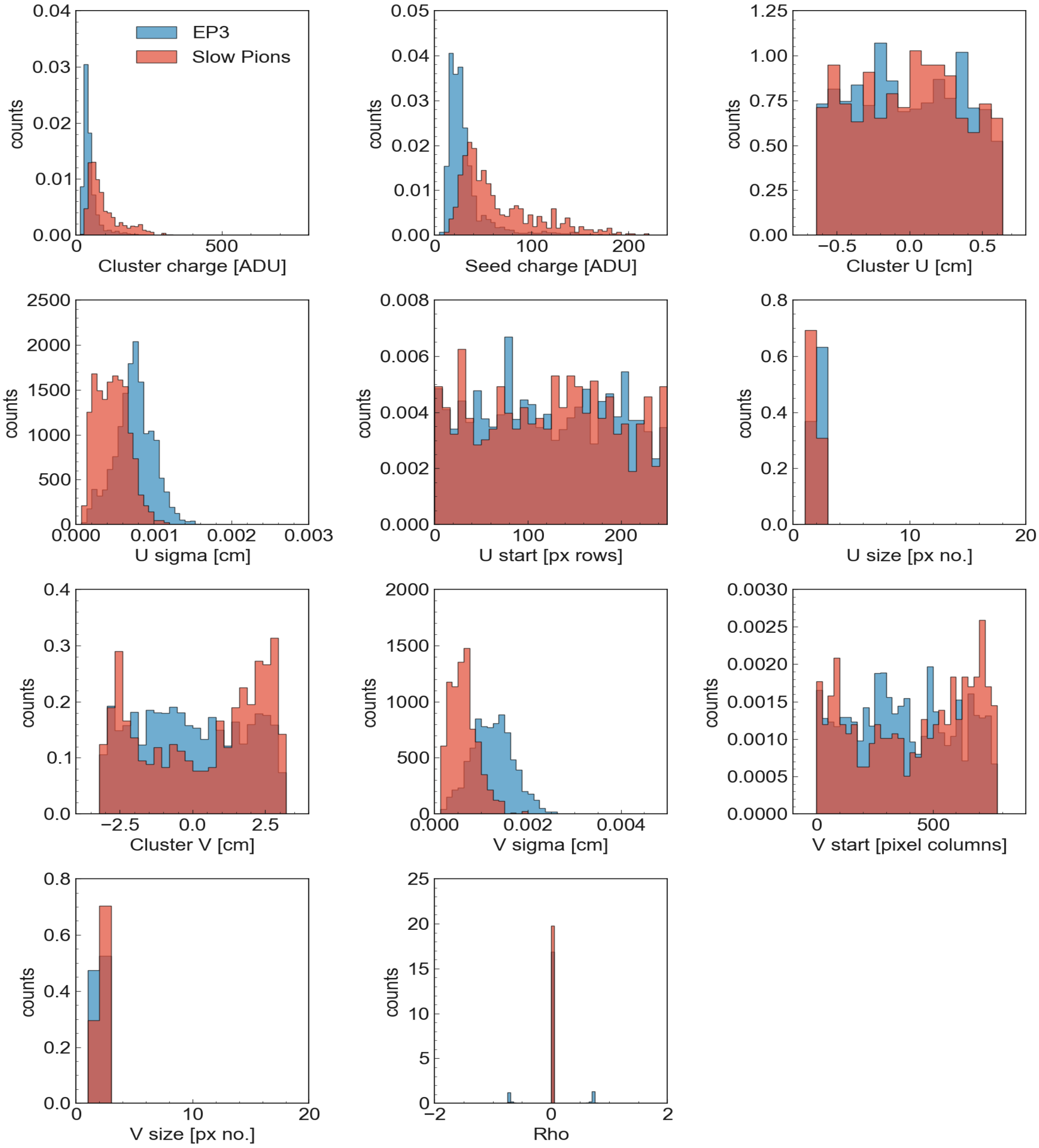


Figure B.7: Cluster variable distributions with pixel multiplicity 2 in the second PXD layer for slow pions and EP3 background.

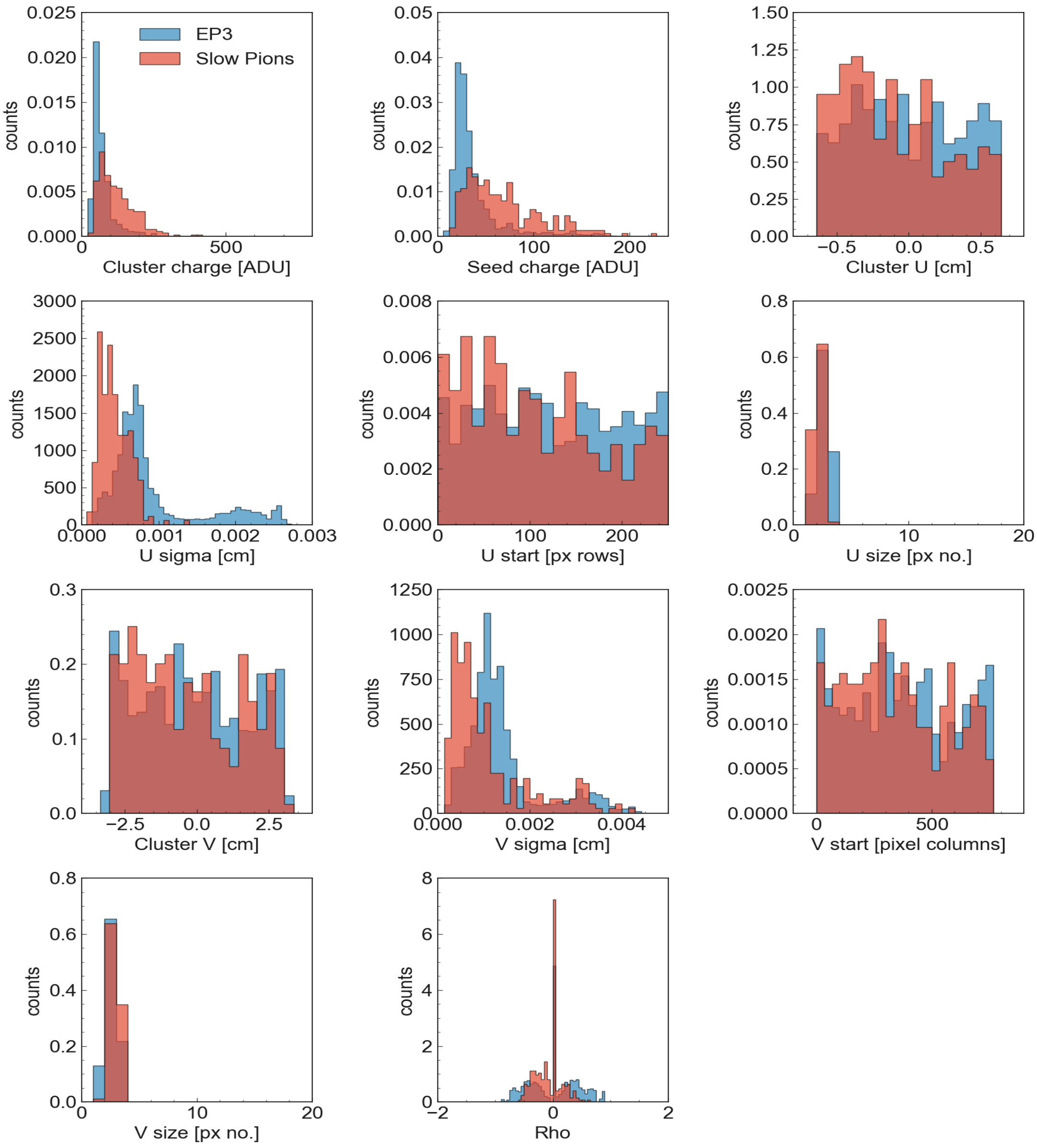


Figure B.8: Cluster variable distributions with pixel multiplicity 3 in the second PXD layer for slow pions and EP3 background.

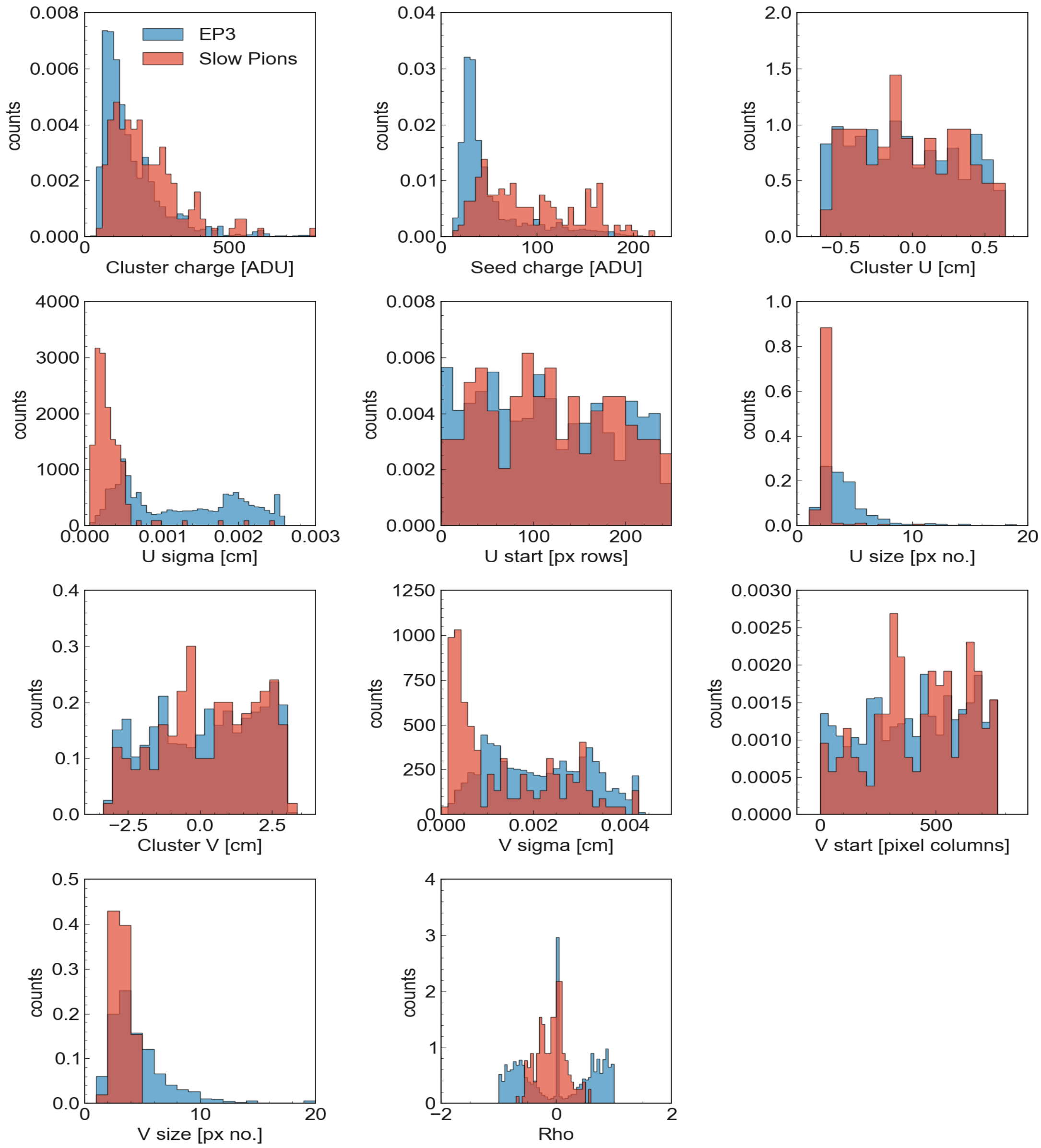


Figure B.9: Cluster variable distributions with pixel multiplicity 4 or more in the second PXD layer for slow pions and EP3 background.

C Deep Learning Results

Deep learning models were trained using the general and specific approach. For the former, DATA0 sample was used. As for the latter, DATA11, DATA12, DATA21 and DATA22 samples were used. Only these cases are studied because they are the dominating pixel multiplicities. The objective is to see if a higher performance can be achieved with an increment of hidden layers. For clearness, the deep-trained neural networks with data samples DATA0, DATA11, DATA12, DATA21 and DATA22 are referred to as DNN0, DNN11, DNN12, DNN21 and DNN22 respectively. The hyperparameters chosen are summarized in Table C.1.

Table C.1: Training hyperparameters for the deep-trained NN.

Data sample	Data size ($\times 10^6$)	No. of input parameters	Variables dropped	Epoch timer	Batch size
DATA0	11	13	None	3000	10,000
DATA11	1.3	8	Cl_layer, Cl_size, U_size, V_size, Rho	3000	1000
DATA12	2.2	11	Cl_layer, Cl_size	3000	1000
DATA21	1.2	11	Cl_layer, Cl_size	3000	1000
DATA22	1.1	11	Cl_layer, Cl_size	3000	1000

The architecture used for the deep-trained NNs is shown in Figure C.1. Table C.2 summarizes the results found for the deep-training performance.

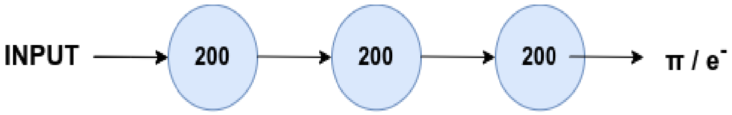


Figure C.1: Neural network architecture for deep learning.

Table C.2: Performance of the deep-trained NNs. Cut chosen such that a rejection rate of 90% is achieved.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]
DATA0	0.555	90.16 ± 0.05	90.35 ± 0.02	90.01 ± 0.02	89.98 ± 0.02
DATA11	0.572	88.64 ± 0.15	87.30 ± 0.07	89.70 ± 0.07	89.98 ± 0.07
DATA12	0.630	87.64 ± 0.11	85.31 ± 0.06	89.48 ± 0.05	89.97 ± 0.05
DATA21	0.610	88.02 ± 0.17	86.98 ± 0.09	89.64 ± 0.08	90.06 ± 0.08
DATA22	0.615	88.94 ± 0.14	83.92 ± 0.07	89.32 ± 0.07	89.96 ± 0.06

Table C.3 compares the above results with those obtained for the corresponding neural networks specified in Table 6.3. From this table, it can be seen that almost all cases are statistically significant. However, due to the improvement obtained being marginal, we conclude that using one layer neural network is enough for this task.

Table C.3: Comparing the efficiency of deep-trained neural networks with one hidden-layer neural network. The statistical significance is calculated.

Data sample	Neural network	Efficiency [%]	Statistical significance [σ]
DATA0	NN0	88.10 ± 0.02	79.55
	DNN0	90.35 ± 0.02	
DATA11	NN11	87.10 ± 0.07	2.02
	DNN11	87.30 ± 0.07	
DATA12	NN12	84.97 ± 0.06	4.00
	DNN12	85.31 ± 0.06	
DATA21	NN21	86.08 ± 0.09	7.07
	DNN21	86.98 ± 0.09	
DATA22	NN22	83.58 ± 0.08	3.32
	DNN22	83.92 ± 0.07	

Since DNN0 and NN0 are significantly different, we can also compare how DNN0 behaves when seeing specific data with the specific neural networks NN11, NN12, NN21 and NN22. The results are presented in Table C.4 and compared in Table C.5. From this table we conclude that having a deep neural network is not significantly different than having the specific neural networks.

Table C.4: Performance of DNN0 when tested on specific data. Cut chosen such that a rejection rate of 90% is achieved.

Case	**Cut**	**Accuracy** [%]	**Efficiency** [%]	**Purity** [%]	**Rejection** [%]
DATA11	0.455	88.69 ± 0.15	87.32 ± 0.07	89.78 ± 0.07	90.06 ± 0.07
DATA12	0.574	87.68 ± 0.11	85.33 ± 0.06	89.53 ± 0.05	90.03 ± 0.05
DATA21	0.725	88.04 ± 0.17	86.02 ± 0.09	89.65 ± 0.08	90.07 ± 0.08
DATA22	0.765	86.92 ± 0.14	83.82 ± 0.08	89.35 ± 0.07	90.01 ± 0.06

Table C.5: Comparing the efficiency of DNN0 and specific neural networks when tested on specific data samples. The statistical significance is calculated.

Data sample	Neural network	Efficiency [%]	Statistical significance [σ]
DATA11	DNN0	87.32 ± 0.07	2.22
	NN1	87.10 ± 0.07	
DATA12	DNN0	85.33 ± 0.06	4.24
	NN12	84.97 ± 0.06	
DATA21	DNN0	86.02 ± 0.09	0.47
	DNN21	86.08 ± 0.09	
DATA22	DNN0	83.82 ± 0.08	2.12
	DNN22	83.58 ± 0.08	

D Grid Search for Hyperparameter Tuning

In order to find the optimal neural network model, a grid search for hyperparameter tuning was done. In this way, a compromise between efficiency and quick computation time can be made. For example, a large number of nodes is beneficial to a certain extend. There is a point where increasing them does not improve performance. Another example is the training epoch. A small number can result in a model that has not finished training and a large number can also make the process take considerably longer to train and can result in an over-fitted model.

This process was done using both the general and the specific approach. Information about data sized used for each case is shown in Table D.1. The data sample used for the general approach is referred to as "Total" and the data samples used for the specific approach are referred to as "LyX, PxY" with X = layer number (1, 2) and Y = pixel multiplicity (1, 2, 3, 4+).

Note that this training was done using 60% of the data sample as training and 20% for validation and testing each. This study was done during the early phase of this thesis and therefore, the data available had a size of 1.3×10^6. As this project progressed, more data was made available (to $\sim11\times10^6$) and the training strategy changed (from 60%/20%/20% training/validation/testing data to 1/3 of the data sample for each sub-set). In addition, all cluster variables in Table 5.1 were used as input parameters, so that the input layer of all models has 13 nodes. This is something that was also changed during the optimization phase, where only the relevant cluster variables were considered.

Table D.1: Size of the data used for training and validation of the different models.

Data	Total size	Train size	Validation size	Pion events [%]
Total	1,327,973	796.784	265,595	50
Ly1, Px 1	184,305	110,583	36,861	38
Ly1, Px 2	283,600	170,160	56,720	46
Ly1, Px 3	144,774	86,864	28,955	47
Ly1, Px 4+	158,318	94,991	31,664	39
Ly2, Px 1	150,042	90,025	30,008	63
Ly2, Px 2	223,649	134,189	44,730	64
Ly2, Px 3	95,849	57,509	19,170	57
Ly2, Px 4+	87,436	52,462	17,487	41

D.1 General approach

Table D.2 summarizes the different trained models. A learning rate of 1x10^{-4} was used for all models.

Table D.2: Models trained using the general approach with their respective hyperparameters.

Model	Nodes in hidden layer	Epoch timer	Batch size	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	100	4000	10,000	0.263	89.23	90.73	88.14
2	100	6000	10,000	0.260	89.41	91.07	88.17
3	100	10,000	10,000	0.256	89.56	91.14	88.40
4	150	4000	10,000	0.259	89.44	91.10	88.18

D.2 Specific approach

From Table D.1 it can be seen that the data samples used for the specific approach do not contain an equal number of slow pions and electrons. Therefore, the results shown in the tables below are normalized according to Section 6.

Table D.3: Different models for Layer number 1 and pixel multiplicity 1.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	150	10,000	10,000	1x10^{-4}	0.280	88.18	85.68	90.20
2	150	4000	10,000	1x10^{-6}	0.525	71.00	44.71	94.33
3	100	6000	2000	1x10^{-5}	0.295	87.50	84.31	90.01
4	100	3000	2000	1x10^{-4}	0.281	88.12	85.32	90.39

Table D.4: Different models for Layer number 1 and pixel multiplicity 2.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	100	6000	2000	1x10^{-4}	0.298	87.68	87.80	87.58
2	150	4000	1000	1x10^{-4}	0.297	87.69	88.09	87.39

Table D.5: Different models for Layer number 1 and pixel multiplicity 3.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	150	4000	1000	1x10^{-4}	0.252	89.79	91.91	88.17
2	200	4000	1000	1x10^{-4}	0.251	89.91	92.05	88.27
3	150	4000	600	1x10^{-4}	0.249	89.93	92.20	88.20

Table D.6: Different models for Layer number 1 and pixel multiplicity 4+.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	100	2000	500	1x10^{-4}	0.152	93.76	93.56	93.95
2	150	2000	1000	1x10^{-4}	0.157	93.38	91.95	94.65
3	200	2000	1000	1x10^{-4}	0.158	94.04	94.88	93.31

Table D.7: Different models for Layer number 2 and pixel multiplicity 1.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	100	4000	1000	1x10^{-4}	0.270	87.56	92.79	83.99
2	100	4000	1000	1x10^{-5}	0.273	87.50	92.79	83.91
3	150	3000	1000	1x10^{-4}	0.270	87.55	92.81	83.98
4	200	3000	600	1x10^{-4}	0.272	87.45	92.85	83.81

Table D.8: Different models for Layer number 2 and pixel multiplicity 2.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	80	3000	1000	1x10^{-4}	0.285	86.54	92.22	82.82
2	150	6000	1000	1x10^{-4}	0.282	86.62	92.79	82.60
3	100	4000	2000	1x10^{-4}	0.283	86.58	92.87	82.49

Table D.9: Different models for Layer number 2 and pixel multiplicity 3.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	150	3000	1000	1x10^{-4}	0.241	89.90	94.27	86.70
2	150	4000	600	1x10^{-4}	0.233	90.17	94.38	87.05
3	200	4000	600	1x10^{-4}	0.232	90.14	94.85	86.68

Table D.10: Different models for Layer number 2 and pixel multiplicity 4+.

Model	Nodes in hidden layer	Epoch timer	Batch size	Learning rate	Loss	Accuracy [%]	Efficiency [%]	Purity [%]
1	100	3000	600	1x10^{-4}	0.123	95.96	96.89	95.12
2	150	4000	600	1x10^{-4}	0.120	96.05	96.98	95.21
3	150	6000	1000	1x10^{-4}	0.119	96.07	96.91	95.32

List of Figures

List of Tables

Abbreviations

ARICH	Aerogel Ring-Imaging CHerenkov Detector
ASIC	Application-Specific Integrated Circuits
BASF2	Belle II Analysis Software Framework
CDC	Central Drift Chamber
CDST	Calibration Data Summary Table
CKM	Cabibbo-Kobayashi-Maskawa
CM	Confusion Matrix
CP	Charge conjugation and Parity
DATCON	Data Concentrator
DEPFET	Depleted Field Effect Transistor
DHH	Data Handling Hybrid
DHHC	Data Handling Hybrid Controller
ECL	Electromagnetic Calorimeter
EP3	Early Phase 3
FN	False Negative
FP	False Positive
HER	High Energy Ring
HLT	High-Level Trigger
IP	Interaction Point
KEK	High Energy Physics Research Centre
KLM	K_L and μ detector
LER	Low Energy Ring
linac	Linear accelerator
MC	Monte Carlo
NN	Neural Network
NP	New Physics
ONSEN	Online Selection Nodes
PID	Particle Identification
PXD	Pixel Vertex Detector
QED	Quantum Electrodynamics

ROC	Receiver Operating Characteristic
ROI	Region of interest
SVD	Silicon Vertex Detector
SM	Standard Model
TN	True Negative
TOP	Time of Propagation detector
TP	True Positive
VXD	Vertex detector

Bibliography

[1] Pavel Pakhlov and on behalf of theBelleII Collaboration. "Determination of the CP violation parameters at Belle II". In: *J. Phys.: Conf. Ser* 1137 (2019). DOI: `10.1088/1742-6596/1137/1/012058`.

[2] Steven H. Robertson. "The Belle II Experiment". In: *Journal of Physics: Conference Series* 1271 (2019), p. 012011. DOI: `10.1088/1742-6596/1271/1/012011`.

[3] *Y(4S)-Particle Data Group*. URL: `https://pdg.lbl.gov/2014/listings/rpp2014-list-upsilon-4S.pdf`. (accessed: 26.07.2023).

[4] F. Abudinén et al. "B-flavor tagging at Belle II". In: *Eur. Phys. J.* 283 (2022), p. 092001. DOI: `10.1140/epjc/s10052-022-10180-9`.

[5] Phillip Urquijo. "Physics prospects at the Belle II experiment". In: *Nuclear and Particle Physics Proceedings* 263-264 (2015). Capri 2014 – Fifth workshop on Theory, Phenomenology and Experiments in Flavour Physics, pp. 15–23. ISSN: 2405-6014. DOI: `https://doi.org/10.1016/j.nuclphysbps.2015.04.004`. URL: `https://www.sciencedirect.com/science/article/pii/S2405601415002953`.

[6] David Griffiths. *Introduction to Elementary Particles*. WILEY-VCH Verlag GmbH & Co. KGaA, 2008.

[7] Mark Thomson. *Modern Particle Physics*. Cambridge University Press, 2013. DOI: `10.1017/CBO9781139525367`.

[8] Thomas Kuhr. *Flavor Physics at the Tevatron: Decay, Mixing and CP-Violation Measurements in pp-Collisions*. 1st ed. Springer Tracts in Modern Physics. Springer Berlin, Heidelberg, 2012, pp. VIII, 161. DOI: `10.1007/978-3-642-10300-1`.

[9] E. Noether. "Invariante Variationsprobleme". ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1918 (1918), pp. 235–257. URL: `http://eudml.org/doc/59024`.

[10] Kobayashi et al. “CP Violation in the Renormalizable Theory of Weak Interaction”. In: *Prog. Theor. Phys.* 49 (1973), pp. 652–657. DOI: `10.1143/PTP.49.652`.

[11] Felix Benjamin Müller. “Characterization and Optimization of the Prototypes DEPFET Modules for the Belle II Pixel Vertex Detector”. PhD thesis. Fakultät für Physik der Ludwig–Maximilians-Universität München, 2017.

[12] Lincoln Wolfenstein. “Parametrization of the Kobayashi-Maskawa Matrix”. In: *Phys. Rev. Lett.* 51 (1983), p. 1945. DOI: `10.1103/PhysRevLett.51.1945`.

[13] Pavel Pakhlov. *Belle II Overview: Physics, Machine, Detector.* 17th Lomonosov Conference on Elementary Particle Physics.

[14] T.Keck, F. Abudinén, and et al. “The Full Event Interpretation”. In: *Comput Softw Big Sci* 3.6 (Nov. 2019). DOI: `10.1007/s41781-019-0021-8`.

[15] W. Altmannshofer et al. “The Belle II Physics Book”. In: *PTEP* 2019.12 (2019). Ed. by E. Kou and P. Urquijo. [Erratum: PTEP 2020, 029201 (2020)], p. 123C01. DOI: `10.1093/ptep/ptz106`. arXiv: `1808.10567 [hep-ex]`.

[16] Kazunori Akai, Kazuro Furukawa, and Haruyo Koiso. “SuperKEKB collider”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 907 (Nov. 2018), pp. 188–199. DOI: `10.1016/j.nima.2018.08.017`. URL: `/10.1016%5C%2Fj.nima.2018.08.017`.

[17] *First particles circulate in SuperKEKB accelerator.* URL: `https://phys.org/news/2016-04-particles-circulate-superkekb.html`. (accessed: 02.07.2023).

[18] T. Abe et al. “Belle II Technical Design Report”. In: (Nov. 2010). arXiv: `1011.0352 [physics.ins-det]`.

[19] T. Kuhr T. Hara and Y. Ushiroda. “Belle II Coordinate System and Guideline of Belle II Numbering Scheme”. In: (Aug. 2011).

[20] Dmitry Matvienko. “The Belle II experiment: status and physics program”. In: *EPJ Web of Conferences* 191 (Jan. 2018), p. 02010. DOI: `10.1051/epjconf/201819102010`.

[21] F.Bernlochner and B. Deschamps and J. Dingfelder and C. Marinas and C. Wessel. “Online Data Reduction for the Belle II Experiment using DATCON”. In: *EDP Sciences* (2017).

[22] Andreas Moll et al. *The vertex detector numbering scheme.* Belle II Note Number 0010.

[23] Matej Srebre et al. "Generation of Belle II Pixel Detector Background Data with a GAN". In: *EPJ Web Conf.* 245.02010 (Nov. 2020), p. 6. DOI: doi.org/10.1051/epjconf/202024502010.

[24] Andreas Moll. "Comprehensive study of the background for the Pixel Vertex Detector at Belle II". PhD thesis. Ludwig Maximilians-Universität München Munich, 2015.

[25] P. Wieduwilt et al. "Performance of production modules of the Belle II pixel detector in a high-energy particle beam". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 991 (2021), p. 164978. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2020.164978. URL: https://www.sciencedirect.com/science/article/pii/S0168900220313759.

[26] L. Zani et al. "The Silicon Vertex Detector of the Belle II experiment". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1038 (Sept. 2022), p. 166952. DOI: 10.1016/j.nima.2022.166952. URL: https://doi.org/10.1016%2Fj.nima.2022.166952.

[27] Belle-ECL et al. "Electromagnetic calorimeter for Belle II". In: *Journal of Physics: Conference Series* 587.1 (Feb. 2015), p. 012045. DOI: 10.1088/1742-6596/587/1/012045. URL: https://dx.doi.org/10.1088/1742-6596/587/1/012045.

[28] S. Baehr et al. "Online-Analysis of Hits in the Belle-II Pixel detector for Separation of Slow Pions from Background". In: *Journal of Physics: Conference Series* 664.9 (2015), p. 092001. DOI: 10.1088/1742-6596/664/9/092001. URL: https://dx.doi.org/10.1088/1742-6596/664/9/092001.

[29] Lu Cao. *PXD Background Study.* FSP workshop: slow pion tracking. 2021. URL: https://indico.belle2.org/event/5438/contributions/28199/attachments/13787/20881/PXD_bkg_FSPslowpi_09Nov2021.pdf. (accessed: 20.07.2023).

[30] Thomas Gesler et al. "The ONSEN Data Reduction System for the Belle II Pixel Detector". In: *IEEE Transactions on Nuclear Science* 62.3 (June 2015), pp. 1149–1154. DOI: 10.1109/tns.2015.2414713. URL: https://doi.org/10.1109%5C%2Ftns.2015.2414713.

[31] Christian Kiesling. personal communication.

[32] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* 2019. arXiv: `1912.01703 [cs.LG]`.

[33] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: `10.1016/j.patrec.2005.10.010`. URL: `https://www.sciencedirect.com/science/article/pii/S016786550500303X`.

Acknowledgements

First, I want to thank my supervisor, Prof. Dr. Christian Kiesling, for giving me the opportunity of joining the Belle II group at the Max Planck Institute for Physics (MPP). Through his support and guidance, I have been able to grow and learn so much during this project. His passion for physics is contagious and I feel very lucky I was able to work with him during this time.

I also want to thank Dr. Hans-Günther Moser, the head of the Belle II group at the MPP. He has always helped me with organisational matters as well as given me the opportunity to participate in the collaboration conferences, including the incredible experience of visiting the Belle II detector in Tsukuba, Japan.

Thanks also to my colleagues and friends, Benedikt, Elia, Felix, Justin, Markus, Oskar, Rafael, Stefan and Xavi who have helped me since day one with guidance at the beginning of my project, as well as questions and discussions in general. I am grateful for our time together inside and outside the institute.

Last but not least, I want to express my gratitude to my parents and siblings, who have supported me unconditionally (despite being thousands of kilometers away) and have done everything they can to help me be where I am today.

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, September 7, 2023