TECHNICAL UNIVERSITY OF MUNICH

MAX PLANCK INSTITUT FÜR PHYSIK

APPLIED AND ENGINEERING PHYSICS MASTER THESIS

A Universal Approach for Particle Identification at Belle II Using Neural Networks

Author:
Xavier Simó Romaguera

Supervisor:
Prof. Dr. Stephan Paul

October 23, 2023

Local Supervisors:
Dr.rer.nat. Stefan Wallner
Dr.rer.nat. Daniel Greenwald

Abstract

The Belle II experiment aims to study the standard model (SM) of particle physics and to search for new physics (NP) beyond the standard model with unprecedented precision. This requires to accurately identify particles produced in these decays, necessitating the development of robust methods for particle identification (PID).

This thesis presents a novel approach to charged particle identification using a neural network aiming to separate hadrons and leptons. We achieved a significant improvement in K/π separation performance. Subsequently, we developed an extended neural network for the simultaneous separation of electrons (e), muons (μ), pions (π), kaons (K), protons (p), and deuterons (d). With this extended neural network, we achieved the same performance for K/π separation as with the specialized neural network.

Furthermore, our approach outperforms the standard method for PID employed at Belle II, as well as another machine-learning based method developed to perform only lepton identification called lepton BDT. This is shown for binary classification, i.e separation of a pair of species. Furthermore, we show that multi-class classification, i.e. the separation of one species from a set of other species, comes with additional challenges. Also, for multi-class classification our neural network approach outperforms the other PID methods used at Belle II.

In summary, in this work we have developed a universal neural network. This means that it is able to perform hadron and lepton identification with a better performance than all existing methods for particle identification at Belle II.

Contents

Chapter 1

Introduction

The Standard Model (SM) [1] of particle physics stands as one of the most successful theoretical frameworks in modern physics. It elegantly describes the fundamental particles and their interactions through the electromagnetic, weak and strong forces. However, despite it is remarkably accurate in predicting and explaining a wide range of experimental observations over six decades of experiments in High Energy Physics, it provides an incomplete understanding of physics picture. There remain unexplained phenomena, such as dark matter, neutrino masses, or the asymmetric dominance of matter over antimatter. New theories have been developed to explain this discrepancies. It is crucial to also experimentally explain New Physics (NP) beyond the SM.

Experiments like the Belle II experiment are carried out to search for deviations that might hint to new physics. However, the approach employed by Belle II differs from other experiments which usually rely on high energy collisions to discover new particles, e.g LHCb. These are experiments in the high energy frontier. They use a direct method, since they are trying to directly observe the "new" particle. Instead of focusing in increasing the energy, the primary goal of the Belle II experiment is to drastically increase the luminosity (high precision frontier). They use a so-called indirect method, i.e the particles are reconstructed and we look at missing information.

Therefore, the primary mission of the Belle II experiment is to address some of these outstanding questions by studying the properties of particles and their decays, with a particular focus on phenomena such as CP violation [2]. CP violation holds crucial insights into the early universe's evolution and the matter-antimatter asymmetry that we observe today. CP violation is an small effect allowed by the SM. However, the CP violation predicted by the SM is an insufficient source, as experiments have found more CP violation that predicted.

To achieve this, Belle II uses electron-positron collisions, which produce various particle products at the $\Upsilon(4S)$ resonance [3]. The products that we want to study are: B-meson [4], D-meson and τ-lepton [5]. They serve as a laboratory to search NP beyond the SM. Ultimately, they decay mainly into 6 charged particle species: electrons (e), muons (μ), pions (π), kaons (K), protons (p), and deuterons (d), which

are the quasi-stable final-states particles produced at Belle II.

Our objective is to differentiate between these six charged particle species hypotheses. For example, this task is challenging in the case of τ decays, particularly those involving $\tau \to K\ \pi\ \pi$ or $\tau \to \pi\ \pi\ \pi$. due to the $\tau \to \pi\ \pi\ \pi$ decay has 20 times larger rate than the $\tau \to K\ \pi\ \pi$ decay. Only particle identification (PID) can separate these processes. Furthermore, PID is also important for other reactions studied at Belle II, e.g. B decays. In summary, to distinguish the six charged particle species, we need to define robust methods to do particle identification in order to ascertain the nature of the detected particles.

Belle II consists of various detectors, which produce an immense amount of data. There are six detectors that measure particle properties that are used for PID. The information of the detectors is expressed in terms of likelihood values $\mathcal{L}^{\mathrm{D}}(h)$ for each detector D and each particle species h. The standard approach at Belle II, called Pure Likelihood approach, combines directly this information. Additionally, there is another specialised tool only for lepton identification, called Boosted Decision Tree (BDT) [6, 7].

Machine Learning (ML) has proven to be an ideal approach for identifying patterns and relationships in extensive data sets. To this end, using neural network for particle identification might be an interesting approach. A first attempt was developed by Tsaklidis et al. [8], who proposed an initial method to enhance exclusively the performance of kaon-pion separation. In this work, only likelihoods from kaons and pions for the six detectors, i.e $\mathcal{L}^{\mathrm{D}}(K)$ and $\mathcal{L}^{\mathrm{D}}(\pi)$, were used. Later, Wallner extended this kaon-pion separation research by using the likelihood of the six particle species.

The aim of this thesis is to develop a novel method capable of performing charged particle identification simultaneously for hadrons and leptons, separating the six charged species. Starting with the neural network proposed by Wallner for K/π separation, the first step is to fine-tuned it and study it. Then, we extend the neural network for all the species.

This work persecutes two main goals. First, we want to develop a method that gives the best performance over all the current methods used at Belle II, i.e Pure Likelihood and BDT. Second, we want to have a universal method, i.e that can be used for all samples and can separate all particle species simultaneously.

This work is divided into eight chapters, with the introduction having already been outlined. Here's a brief overview of the subsequent chapters. Chapter 2 provides an brief introduction to the SuperKEKB accelerator and the Belle II experiment. In Chapter 3 the physics principles for particle identification, along with a description of the PID detectors are explained. Moreover, it describes the standard Belle II PID methods. At the end, the method used to evaluate the performance of the different methods is presented. Chapter 4 describes the neural networks devel-

oped for this work and introduces the neural network PID. Chapter 5 evaluates the performance of the neural network developed for K/π separation. It includes studies of the network architecture. Furthermore, it reaches a conclusion on which training data sample we should use. Chapter 6 studies the performance of the extended neural network for binary classification. It is tested in different samples against other dedicated methods. Chapter 7 evaluates the extended neural network for multi-class classification. Chapter 8 gives the conclusions and outlook of this work.

Chapter 2

Belle II at SuperKEKB

In this chapter, an introduction of the SuperKEKB facility is given in section 2.1. Next, in section 2.2, the Belle experiment is introduced, describing the main goals and challenges, providing a concise overview of its detectors, and descriving the coordinate system of the Belle II experiment.

2.1 The SuperKEKB Accelerator

The Belle II experiment is located at SuperKEKB e^+e^- collider in Tsukuba, Japan. It accelerates and collides electrons and positrons with asymmetric energies of 7 GeV and 4 GeV, respectively, resulting in a centre-of-mass energy of $\sqrt{s} = 10.58$ GeV [3] at the $\Upsilon(4S)$ resonance. Since the $\Upsilon(4S)$ predominantly decays into B-meson pairs (B- and $\bar{B}$-), with a branching fraction of above 96% [9], SuperKEKB is ideal for the studies of B-Physics. Hence, SuperKEKB aim to improve our knowledge of the flavour physics, estimate with more precision the parameters of the Standard Model and search for physics beyond the Standard Model. The design luminosity of SuperKEKB is $8 \times 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, which is an increase by a factor of 40 with respect to its predecessor, the KEKB. In its first data taking period [10] from 2018 to 2022 SuperKEKB reached the world highest instantaneous luminosity of $4.7 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and collected an integrated luminosity of $424\,\mathrm{fb}^{-1}$ [11].

Figure 2.1 gives an overview over the SuperKEKB facility. The electrons, introduced with an electron injection gun, are accelerated using a LINAC. Part of the electrons are separated from the rest and, once they hit a thick tungsten target, a shower of particles is produced including positrons, which are separated using a magnetic field. The electron beam will be stored in the 3 km long high-energy ring (HER) and accelerated to 7 GeV. Analogously, the positron beam is stored in the low-energy ring (LER) and accelerated to 4 GeV. The collision point of the electron and the positron beam is located in the Tsukuba section, where the Belle II detector is located. Further information can be found in [3].

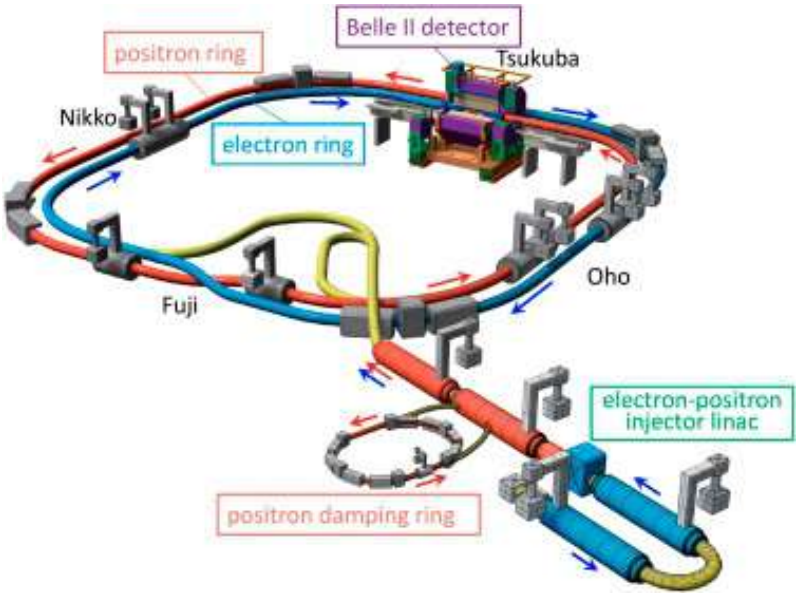


Figure 2.1: Schematic view of the SuperKEKB facility from ref. [12].

2.2 The Belle II Experiment

One of the primary objectives of the Belle II experiment is to identify particles resulting from decays. To achieve this, we have outlined four key tasks: particle identification, tracking, calorimetry, and neutral measurements. We face the intricate challenge of having a high background rate [11], leading to an increased occurrence of fake hits and radiation damage. However, the performance of the Belle II is expected to be equivalent to or better than Belle even under the higher background. To achieve this, the Belle II detector is designed as an exceedingly precise measurement system.

The detector is centred around the interaction point (IP), where electrons and positrons collide, with the aim of detecting and measuring all particles produced in the e^+e^- collisions. Due to the asymmetry of the SuperKEKB collisions, the detector is asymmetric along the beam axis. In the context of Belle II, the "forward" direction is the direction in which 7 GeV electron beam points, while "backward" is the direction in which the 4 GeV positron beam points.

Figure 2.2 shows a visual representation and schematic of the coordinate system. It is composed of three distinct components: the barrel, the forward endcap, and the backward endcap. The barrel is located at the central region of the detector, surrounding the interaction point. The forward endcap is positioned in the forward direction relative to the interaction point, whereas the backward endcap is located in the opposite direction from the interaction point.

The Belle II detection system is formed by seven individual detectors, each one dedicated to a specific task. This section provides a brief overview. Detectors for

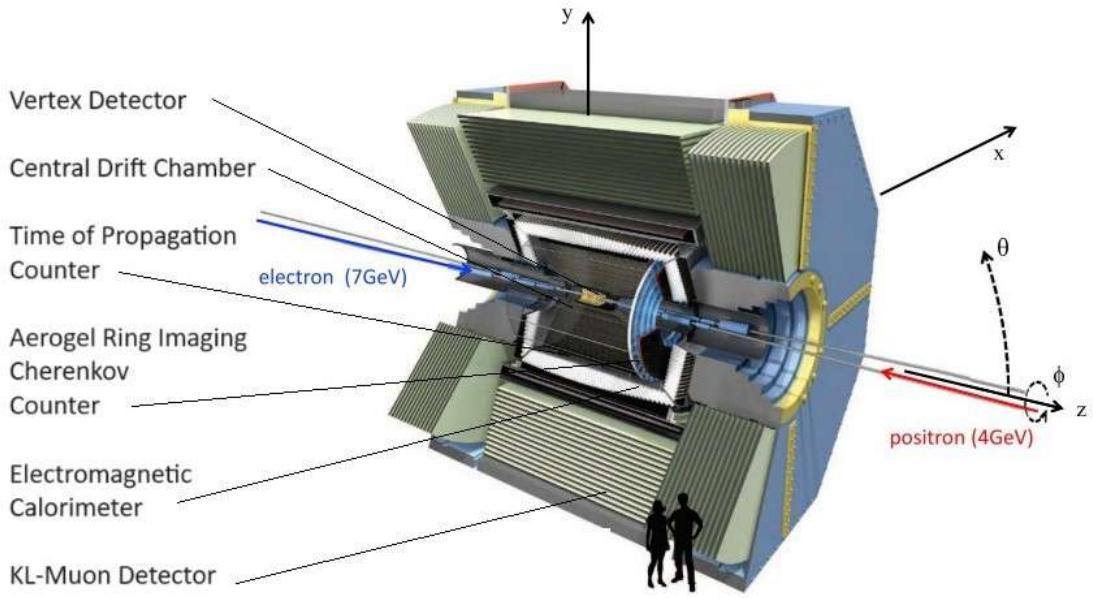


Figure 2.2: Belle II Coordinate system.

PID are described in more detail in chapter chapter 3.

The innermost detector from the IP is the Pixel Detector (PXD). Together with the Silicon Vertex Detector (SVD), they allow for the precise measurement of each particle's tracks as it passes through these detectors. This information is crucial to reconstruct the point where the particle was created (vertex) with high resolution (~ 50 μm).

After passing through the SVD, the particle enters the Central Drift Chamber (CDC). The CDC consists of layers of gas-filled cells, where charged particles ionize the gas. The ionitzation is measured, resulting in a sequence of hits that trace the particle's path. Furthermore, the magnetic field is mainly homogeneous in the CDC. Therefore, by measuring the curvature of a particle inside the magnetic field, resulting from the Lorentz force experienced by particles within it, the momentum of charged particles can be also determined.

Outside of the CDC, there are the Time-of-Propagation Counter (TOP) and the Aerogel Ring Imaging Cherenkov Detector (ARICH), covering the barrel and forward endcap regions respectively. Both use Cherenkov radiation to identify the species of charged particles, as discussed in section 3.1.1.

Photons and electrons are detected by the Electromagnetic Calorimeter (ECL). They deposit nearly all their energy in the ECL by producing electromagnetic showers. The energy deposition is used to determine the energy of the particle.

The outermost detector of Belle II is the K-Long and muon (KLM) detector. The KLM measures the energy deposited in its scintillators, allowing for the identification of muons. Additionally, it provides K-Long identification information.

Belle II uses a right-handed Cartesian coordinate system. The z-axis points along the beam-line, in the direction of the electrons. The x-axis points towards the centre of the Belle II detector,and the y-axis points vertically upwards. The origin of the coordinate system is located at the interaction point where the electrons and positrons collide.

The direction of a track is often expressed in spherical coordinates (θ, ϕ). The polar angle θ is the angle between the z-axis and the direction of the track. The azimuthal angle ϕ is the angle between the x-axis and the direction of the track. The polar angles is in the range of $[0, \pi]$ radians and azimuthal angle is in the range $[0, 2\pi]$ radians.

The detectors cover almost the full solid angle and provide excellent momentum resolution across the entire kinematic range. In summary, the Belle II experiment, with its detectors, exhibits a highly efficient particle identification system capable of distinguishing photons and charged particles i.e pions, kaons, protons, electrons, and muons, over the full kinematic range of the experiment. Furthermore, the Belle II spectrometer is equipped with a fast and efficient trigger system [13], as well as a data acquisition system [14]. The Belle II trigger system efficiently selects and records collision events. The data acquisition system manages the flow of information from the detector components and transfers the raw data to the offline storage. Then, it is converted into physical variables.

Chapter 3

PID at Belle II

This chapter starts with a description of main physical principles used for particle identification: Cherenkov radiation and energy loss (see section 3.1). In section 3.2 the different PID detectors are explained in detail. It includes a brief description of their main parts, physical functioning, operating principles, and detector likelihood definition $\mathcal{L}^{\mathrm{D}}(h)$.

In section 3.3 we explain how to combine the likelihoods obtained from the detectors to define the likelihoods for each specie $\mathcal{L}(h)$, the so-called Pure Likelihood approach. In section 3.4 we explain how combine these variables, using a normalisation process, to define the classification variables for various PID tasks. Furthermore, the lepton BDT, an existent method focused on lepton PID is explained in section 3.5. Finally, in section 3.6, the performance measures for PID are defined to assess the performance of the different methods.

3.1 Physics Principles for PID

3.1.1 Cherenkov Radiation

In order to perform particle identification, and specially for K/π separation, the Belle II experiment uses Cherenkov radiation. In principle, a massive particle can't exceed the speed of light in the vacuum. However, this does not hold when travelling through a refractive medium with $n > 1$, where n is the refractive index. This is due to the fact that velocity of the light drops to c_0/n, where c_0 is the velocity of the light in the vacuum. When a charged particle passes through a dielectric material (meaning that it can be polarised), it induces the local electromagnetic field, polarising the near molecules. Next, the molecules return to a state of equilibrium, releasing a coherent and the electromagnetic field. If the charged particle exceeds the speed light, Cherenkov photons are emitted, creating a cone due to wavefronts of electromagnetic radiation trail behind the particle. Figure 3.1 shows the release of a Cherenkov photon. As discussed in [15], one can define the emission angle of a Cherenkov photon θ_c as:

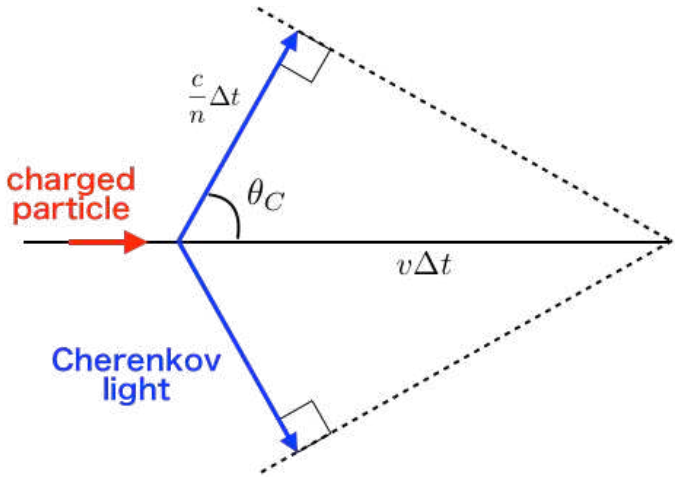


Figure 3.1: The schematic view of the Cherenkov photon. Retrieved from [12].

$$\cos\theta_c = \frac{\frac{c_0}{n}\Delta t}{v\Delta t} = \frac{1}{n\beta} \tag{3.1}$$

where β is the ratio of velocity of the particle and c_0. Using:

$$\beta = \frac{|\vec{p}|}{E} = \frac{|\vec{p}|}{\sqrt{m^2 + |\vec{p}|^2}} \tag{3.2}$$

where $|\vec{p}|$ is the magnitude of the momentum, E is the energy and m is the mass of the particle, one can write:

$$\cos\theta_c = \frac{\sqrt{\left(\frac{m}{|\vec{p}|}\right)^2 + 1}}{n} \tag{3.3}$$

At end, one can define

$$p_{th} = \frac{m}{\sqrt{n^2 - 1}} \tag{3.4}$$

where p_{th} is the momentum threshold of the particle. Below this threshold, Cherenkov photons are not produced.

The Time Of Propagation counter (TOP) and Aerogel Ring Imaging Cherenkov detector (ARICH) use the Cherenkov effect to identify particles. As explained in section 2.2, we know the momentum of the particles through their curvature in the magnetic field. By measuring the emitted Cherenkov photons, we can apply Eq. (3.3) to determine the particle's mass. Figure 3.2 shows the relation between the Cherenkov angle and the momentum. This relation is specific for a given particle species, resulting in the differently colored curves

Despite using the same physics principle, the TOP and the ARICH vary in their operating principles. In addition, their cover different θ regimes. ARICH is located

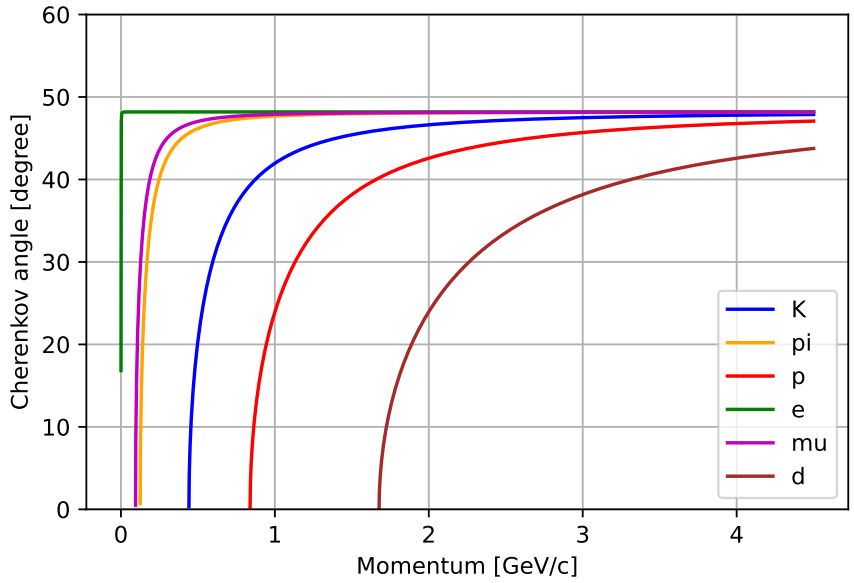


Figure 3.2: Cherenkov angle as a function of momentum for various particle species for refractive index of $n = 1.5$.

at the forward endcap; whereas the TOP detector is situated in the barrel region (see Fig. 3.4). This is indeed useful because the combination cover a wider range of θ. More details about the Cherenkov radiation and its applications can be found in [16].

3.1.2 Energy loss

In addition to the characteristic emission of Cherenkov light, the energy loss through matter dE/dx can be used to perform particle identification. This process describes how charged particles lose energy as they traverse a medium, primarily due to interactions with the electrons within the material. The energy loss, among others factors, depends on the charge and velocity of the particle and the density of the medium. It follows the Bethe Block formula.

To perform particle identification, we use as additional input the momentum of the charged particle, obtained previously. Figure 3.3 illustrates the energy loss for different charged particle species as a function of the momentum. For a given momentum, the energy loss is specific for a particle species. Therefore, by combining the momentum measurement with the measured energy loss (dE/dx), we can distinguish between different particle species.

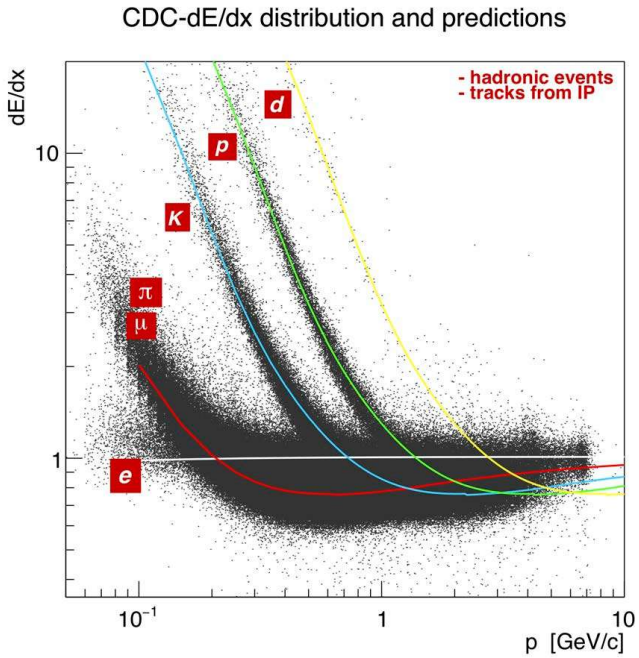


Figure 3.3: $\frac{dE}{dx}$ with respect to the momentum of various charged particle species. The points belong to experimental values. The curve are the theoretical values for the different particles. Retrieved from [17].

3.2 The Belle II Detectors for PID

As explained in the previous chapter, the Belle II detection system consists on seven individual detectors. They are build to work in different θ regions as shown in Fig. 3.4. Additionally, table 3.1 provides a summary of their components, spatial locations, and their corresponding θ coverage.

Further, some of the detectors are partly specialized to identify certain particle. For example, the KLM detector is designed to identify muons, while the ECL detector is optimized for detecting photons or electrons.

Finally, they work in different momentum regions. For example, as illustrated in Fig. 3.3, in the region $|\vec{p}| > 1.5$ GeV/c information derived from energy loss measurements in the CDC and SVD is insufficient for PID, as all species yield to a similar energy loss. On the othr hand, the TOP detector proves to be a important tool in this region, offering supplementary information.

We need to combine the information from the six PID detectors in an optimal way to effectively identify particles produced. For that purpose, different groups of detector experts propose models to integrate these measurements, simplifying the complexity and deriving more user-friendly variables. They are called PID likelihoods $\mathcal{L}^{\mathrm{D}}(h)$, where D stands for the 6 detectors (SVD, CDC, TOP, ARICH, ECL

Table 3.1: Summary of the detector components. Retrieved from [3].

Purpose	Name	Component	Configuration	θ coverage
Beam pipe		Beryllium	Cylindrical, inner radius 10 mm, 10 μm Au, 0.6 mm Be, 1 mm paraffin, 0.4 mm Be	
Tracking + Particle ID	SVD	Silicon Strip (double sided)	Rectangular and trapezoidal, strip pitch: 50(p)/160(n)-75(p)/240(n) μm, with one floating intermediate strip; four layers at radii: 39, 80, 104, 140 mm small cell, large cell, 56 layers	[17°; 150°]
Tracking + Particle ID	CDC	CDC Drift Chamber with $He - C_2H_6$ gas	14336 wires in 56 layers, inner radius of 160 mm outer radius of 1130 mm	[17°; 150°]
Particle ID	TOP	RICH with quartz radiator (DIRC)	Barrel: 16 segments in ϕ at $\sim$ 120 cm, 275 cm long, 2 cm thick quartz bars with 4 × 4 channel MCP PMTs	[31°; 128°]
	ARICH	RICH with aerogel radiator	FWD end-cap: 2 × 2 cm thick focusing radiators with different n, HAPD photodetectors	[15°; 34°]
Calorimetry	ECL	CsI(Tl)	Barrel: $r = 125 - 162$ cm, end-caps: at $z = -102$ and $z = +196$ cm,	[12.4°; 31.4°], [32.2°; 128.7°], [130.7°; 155.1°]
Muon ID	KLM	barrel: RPCs and scintillator strips	2 layers with scintillator strips and 13 layers with 2 RPCs	[40°; 129°]
	KLM	end-caps: scintillator strips	14 (12) layers of [7–10] × 40 mm^2 strips in forward (backward) region	[25°; 40°], [129°; 155°]

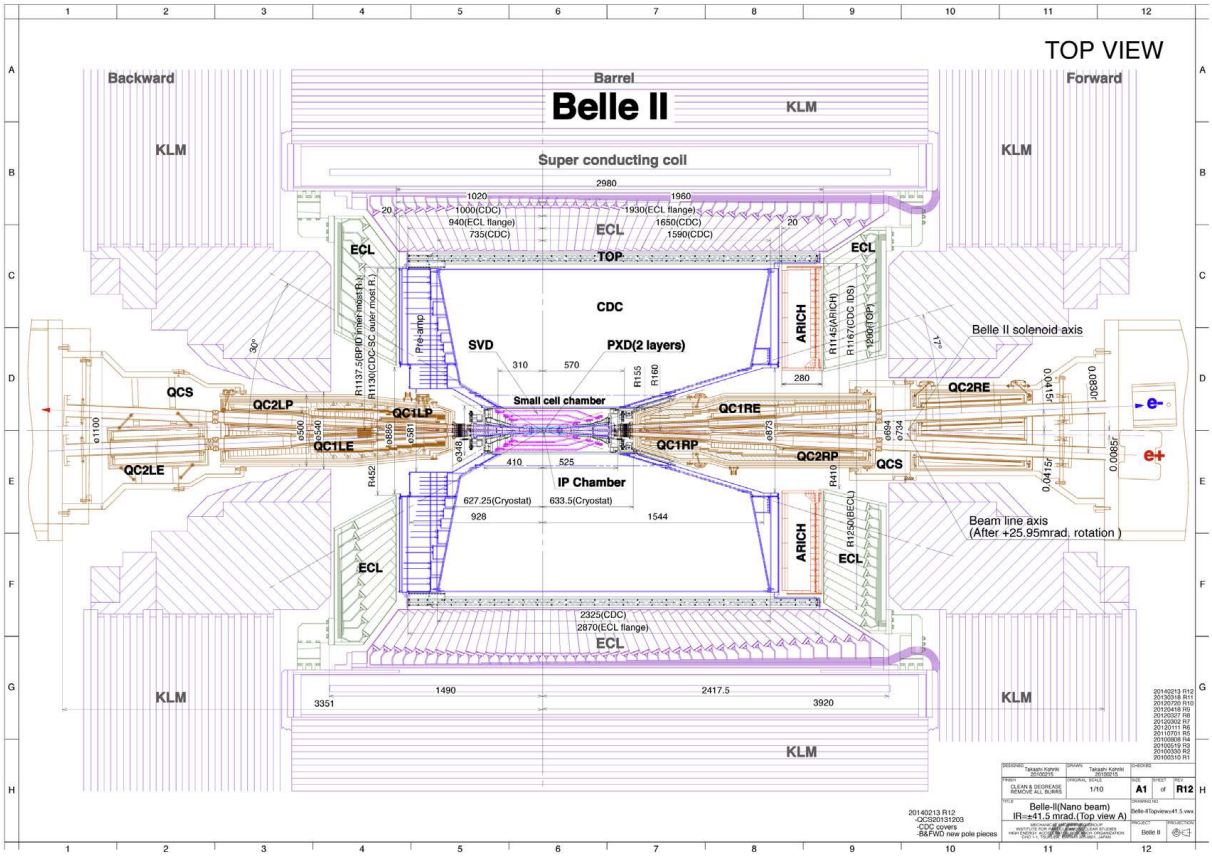


Figure 3.4: Schematic view of Belle II detectors. Retrieved from [3].

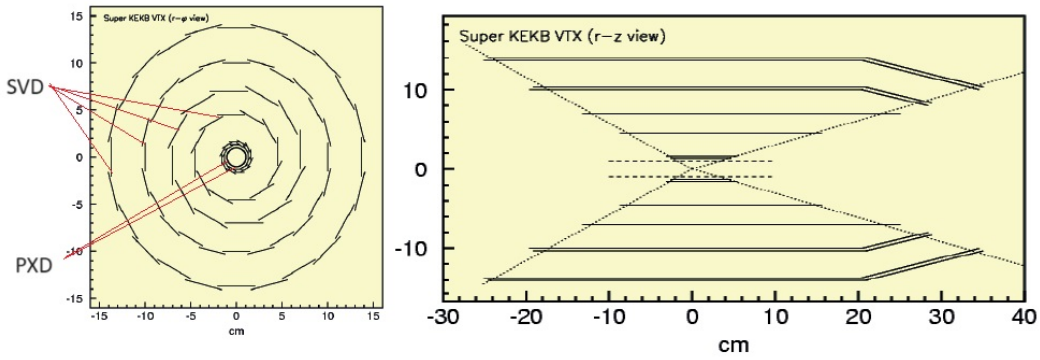


Figure 3.5: Schematic view of the Belle II vertex detector with a Be beam pipe, two pixelated layers and four layers of silicon strip sensors. Retrieved from ref [3].

and KLM)[1] and h for the possible species. The generation of these likelihoods involves employing distinct complex models, different for each detector. As a result, 36 PID likelihoods are obtained since we have 6 species as hypotheses and 6 detectors. The brief explanation of how the likelihoods of each detector are computed can be found below. Further information on how the likelihoods are formed can be found in [3].

3.2.1 Vertex Detector (VXD)

The Vertex Detector (VXD) [3] is formed by two semiconductor detectors, the Pixel Detector (PXD) and Silicon Vertex Detector (SVD). It is located in the innermost part of the Belle II experiment, around the beam pipe, and is comprised in total of six layers. The PXD, which builds the first two layers, is made of silicon pixel sensors due to high background expected close to the interaction point. The outer four layers are the SVD, which is made of silicon strips. The VXD layers distribution is shown in Fig. 3.5. Both detectors share the same polar angle coverage ($\theta \in [17°; 150°]$) and have full coverage of the azymuthal angle.

Both, have the same physics working principle. When charged particles pass through the silicon, they ionize the material, generating electron-hole pairs, which will move towards the electrodes. They will be measured as electrical signals. The position and timing of these signals are used to reconstruct the trajectory and interaction points of the particles (tracking and vertexing).

3.2.1.1 Silicon Vertex Detector (SVD)

The SVD [18,19] layers are situated at the following radii: r=39mm, 80mm, 104mm, and 135mm. SVD sensor are constructed with Double-Sided Silicon micro-strip Detectors (DSSDs), the size and shape of which depend on the layer. In total, there

[1]PXD is not used for PID in this work, only for tracking.

are 72 SVD sensors and around 220 thousand strips. The sensor are distributed perpendicular (n-strips) and parallel (p-strip) to the beam direction , providing (x,y) coordinates of the hit location. The high granularity of the SVD allows for precise tracking and vertexing, helping identify primary and secondary interaction points.

Additionally, the energy loss in the SVD is measured by measuring the deposited charge. This information is used to perform particle identification in the low momentum region that can not reach the CDC.

The likelihood of the SVD $\mathcal{L}^{\mathrm{SVD}}(h)$ is computed using information from the hits and deposited charge in the detector, and comparing it with the expected distribution based on the assumed specie hypothesis.

3.2.2 Central Drift Chamber (CDC)

One of the the main tracking detectors of the Belle II is the CDC [20]. It is made of a large volume drift chamber with small drift cells, between two semiconductor tracking detectors, with a inner radius of 160 mm and an outer radius of 1130 mm. Its main purpose is to to reconstruct the trajectory of charged particles. It is filled with $\mathrm{He} - \mathrm{C_2H_6}$ 50:50 gas mixture, to suppress multiple scattering with an average drift velocity of 3.3 cm/μs. Charged particles traversing the chamber ionize the gas. The produced charged is detected by wires.

The number of ionized electrons is roughly proportional to the particle's energy loss, allowing the determination of energy loss (dE/dx) by the CDC. As explained in section 3.1.2, this allows to perform particle identification. The CDC covers a polar angle interval of $17° < \theta < 150°$ and a full azimuthal angle.

The likelihood of the CDC $\mathcal{L}^{\mathrm{CDC}}(h)$ is computed using its momentum and energy loss to relate it with the expected particle, as shown in Fig. 3.3.

3.2.3 Time of Propagation counter (TOP)

Figure 3.6 shows a schematic of the TOP detector. The Time of Propagation detector [21,22] is located in the barrel region of the Belle II detector in between the CDC and ECL, with θ coverage of $\in [31°, 128°]$. The TOP detector comprises 16 radiator modules positioned around the CDC. Each detector module consists of a 45 cm wide and 2 cm thick quartz bar with a small expansion volume (about 10 cm long). These radiators are read out micro-channel plate photo-multipliers (MCP-PMT), located at one end of the bars, and a spherical focusing mirror, attached to the other end, to focus and direct the Cherenkov photons towards the PMTs.

In the TOP detector, the quartz radiator serves as the refractive medium, with its refractive index varying between 1.43 and 1.58 depending on the wavelength of the light. Cherenkov photons generated within the TOP undergo internal reflection as they traverse the quartz radiator until they eventually interact with a photon

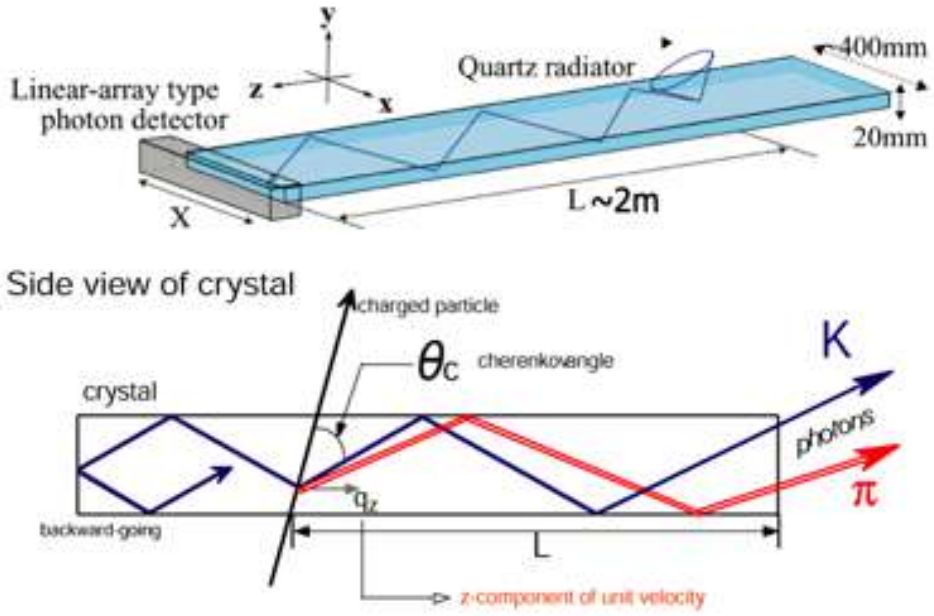


Figure 3.6: Schematic view of TOP working principle and side view with the internal reflection of the Cherenkov photons visualised. Retrieved from [24].

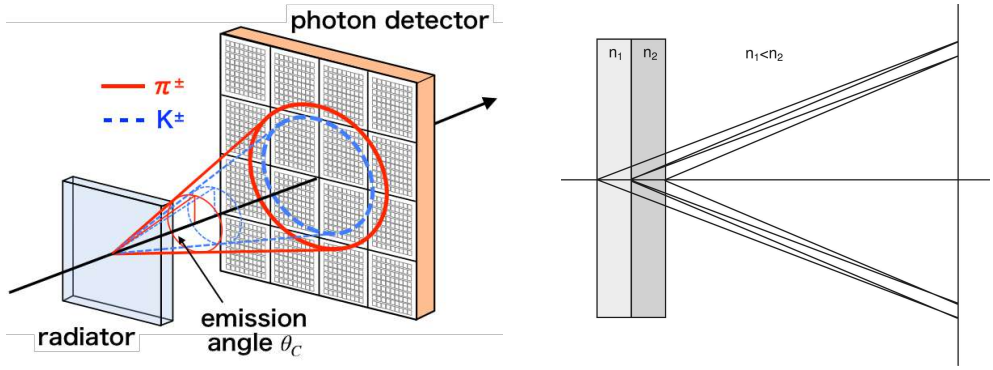
detector located at one end. The trajectory of the photon depends on the angle under which the photons is emitted with respect to the quartz bar. This angle is given the the inclination of the particle track with respect to the quartz bar and the Cherenkov angle under which the photon is emitted, i.e. the angle with respect to the direction of the track. The detector measures both the impact location and the time of propagation of these Cherenkov photons by the MCP-PMTs.

The impact position and the time of propagation are characteristic for the Cherenkov angle (illustrated in Fig. 3.6) and therefore for the particle species for a given track momentum and inclination. This characteristic pattern of impact location and time of propagation is used to formulate a likelihood $\mathcal{L}^{\mathrm{TOP}}(h)$ for each species hypothesis h for a given track. Additional information on how to compute the TOP likelihood is given in [23].

3.2.4 Aerogel Ring-Imaging Cherenkov detector (ARICH)

The ARICH also uses Cherenkov radiation as its operating principle. It is located only in the forward endcap [2] region, covering $\theta \in [14°; 34°]$. The working principle of the ARICH is based on production of the Cherenkov photons once a charged particle enters a aerogel radiator. The θ_c of the Cherenkov light-cone is measured by a

[2]Due to the boosted centre of mass energy, particles predominantly travel towards the forward endcap rather than the backward endcap. Additionally, those particles directed towards the backward endcap typically exhibit low momenta, a range already effectively covered by the CDC.



(a) (b)

Figure 3.7: ARICH, the proximity focusing RICH with a non-homogeneous aerogel radiator in the focusing configuration, principle of operation. Retrieved from a) [12] and b) [28].

photon detector situated behind the aerogel radiator. This configuration is shown in Fig. 3.7a. One can observe that different particles, K and π in this example, produce different rings as θ_c depends on the mass through Eq. (3.3). To ensure effective detection, 20cm thick expansion volume is installed between the aerogel radiator and photon detectors, in order to adequately sized Cherenkov rings for effective detection. The detectors are based on Hybrid Avalanche Photo-Detectors (HAPD) technology, which are arranged in 9 concentric rings for a total of 540 sensors. They are composed of a vacuum tube with solid state senor of avalanche diode type photo-detector (APD). Further details of Belle II HAPD distribution and optimisation can be found in [25].

ARICH uses two different aerogel radiators, placed one after the other one. They have the same thickness but different refractive index ($n = 1.046$ and $n = 1.056$) to produce Cherenkov rings that are focused at the same point at the photon detector. With this setup (shown in Fig. 3.7b), a better resolution is obtained [26], when compared to using only one medium.

The likelihood of the ARICH $\mathcal{L}^{\text{ARICH}}(h)$ is computed by evaluating the observed hits from the Cherenkov photons on each pixel in the photon detector, given the expected number of hits for a specific charged track hypothesis. Additional information on how to compute the ARICH likelihood is given in [27].

3.2.5 Electromagnetic Calorimeter (ECL)

A high resolution Electromagnetic Calorimeter (ECL) [28, 29] plays an important role in the Belle II experiment, to effectively measure neutral final state particles. The main aim of the ECL is measure photons. However, it also offers a way to efficiently identify electrons, i.e. separate electrons from muons and charged hadrons. In addition, they help the KLM in K_L^0 identification and triggering. It is composed by a highly segmented array of thallium doped caesium iodide CsI(Tl) crystals, pointing towards the interaction region of the beams (as shown in Fig. 3.8). In

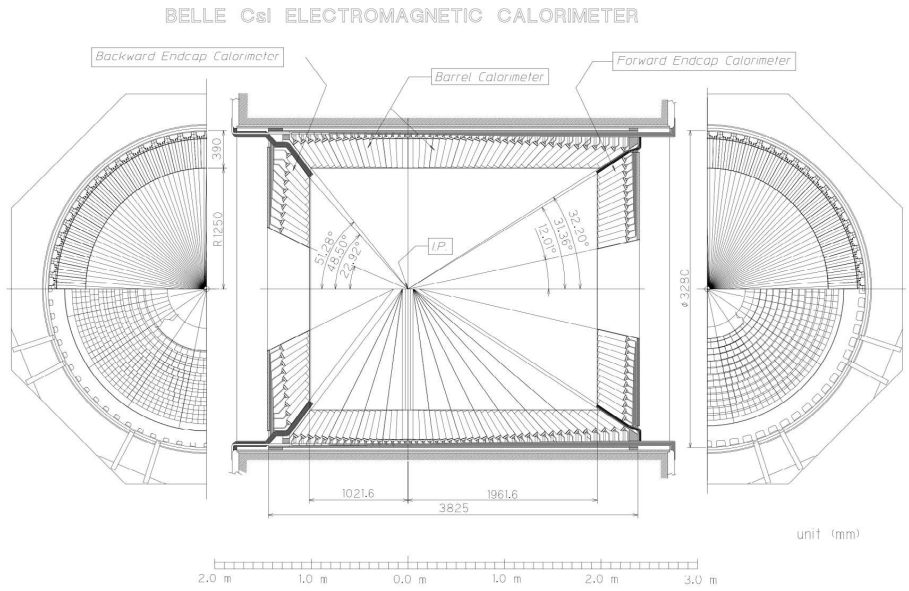


Figure 3.8: Schematic view of ECL detector. Retrieved from [28].

total, there are 8736 crystals, covering about 90% of the solid angle in the centre-of-mass system. The ECL is placed in all three detector regions: the barrel, the forward endcap and in the backward endcaps. It covers almost the full polar angle [3].

The working principle of the ECL is based on the production of electromagnetic showers by charged particles and photons entering the calorimeter and interacting with the lead tungsten crystals, depositing energy. The shower products generate scintillation light, which is subsequently detected at the end of each crystal. The detection is carried out with two sets of photodiodes, that are glued to the crystal, with a sensitive area of 10 mm^2, connected to sensitive preamplifiers.

The intensity of this scintillation light is proportional to the energy of the incident particle, allowing for precise determination of the deposited energy. Specifically, electrons and photons deposit all of their initial energy. Therefore, the deposited energy is equal to the total energy. As e is quasi massless, the electron and photon momentum equals its energy. Hence, the ratio of measured energy over measured momentum E/p, peaks at 1. Photon and an electron are separated by finding a corresponding charged track in the Central Drift Chamber (CDC) or not. The photon, being uncharged, does not leave a track, whereas the electron does, allowing us to differentiate them. Particles of other species with larger mass do not lose all of their energy in the ECL. Therefore, the ratio of E/p does not reach its peak at 1.

[3] $12.4° < \theta < 155.1°$, except for two about $\sim 1°$ wide gaps between the barrel and endcaps, where table 3.1 shows the exact θ range for all three regions.

To compute the likelihood $\mathcal{L}^{\mathrm{ECL}}(h)$, the ECL only uses the ratio between the energy deposited and the momentum, E/p.

3.2.6 K-long muon detector (KLM)

The K_L^0 and muon detector, know as KLM [3], is the outermost detector of the Belle II experiment. The aim of this detector is to detect long-lived particles, which are not absorbed in the ECL, that traverse a significant distance through the detector volume before ultimately reaching the outermost region. It consists of an alternating sandwich of 4.7 cm thick iron plates and active detectors. These iron plates act as the magnetic flux return for the solenoid, situated between the ECL and KLM. Additionally, they yield an extra 3.9 interaction lengths of material, surpassing the 0.8 interaction lengths of the calorimeter, in which K_L^0 mesons shower hadronically. In contrast, muons do not produce any showers, but are visible as curved tracks in the active part of KLM. The detection is done using layers of scintillator strips that produce scintillation light, captured by silicon photomultipliers (SiPMs). It has an angular acceptance of $20° < \theta < 155°$including both end caps and the barrel region. [4].

Muons are identified by matching the extrapolations of charged tracks from the CDC to the KLM with signals in the active part of the KLM. If a KLM cluster lacks a corresponding track, this indicates a K_L^0 particle. Extended information on muon and K_L identification is provided in [30].

The likelihood of the KLM $\mathcal{L}^{\mathrm{KLM}}(h)$ is determined based on the presence or absence of a cluster, along the extrapolation of charged tracks through the KLM.

3.3 Pure Likelihood Approach

The starting point is to combine the likelihoods from the six PID detectors in an optimal way to effectively identify particles produced. The standard approach for PID at Belle II [3, 31] uses the likelihoods from each of the six detectors for the six hypotheses: e, μ, π, K, p, and d. To define a combined PID likelihood $\mathcal{L}(h)$ for hypotheses h, the likelihoods from the subdetectors are multiplied as they are assumed to be independent:

$$\mathcal{L}(h) = \prod_{\mathrm{D}} \mathcal{L}^{\mathrm{D}}(h) = \mathcal{L}^{\mathrm{SVD}}(h)\,\mathcal{L}^{\mathrm{CDC}}(h)\,\mathcal{L}^{\mathrm{TOP}}(h)\,\mathcal{L}^{\mathrm{ARICH}}(h)\,\mathcal{L}^{\mathrm{ECL}}(h)\,\mathcal{L}^{\mathrm{KLM}}(h) \quad (3.5)$$

This method is called Pure Likelihood approach.

It presents two major drawbacks, both of which are overcame with the use of the neural networks proposed in chapter 4. The computation of likelihoods requires modelling, which require approximations. Therefore, the likelihoods might not be

[4] Table 3.1 shows the exact θ range divided in all three regions.

perfect. On the other hand, the Pure Likelihood approach uses the direct multiplication of individual likelihoods (Eq. (3.5)) and does not account for possible correlations, leaving room for improvement.

3.4 Binary Normalitzation

Each physics analysis has its specific requirements on PID. They can be grouped into two tasks: binary classification and multi-class classification. The choice between both depends on the specific characteristics of the physics process that is studied by the analyst. Binary classification is used if only two possible species are considered. Multi-class classification involves the separation of more than two species simultaneously, i.e to separate one particle from the rest.

For multi-class classification, i.e considering all six species, the likelihood defined in Eq. (3.5) can directly be used as classification variable. This concept is elaborated upon in chapter 7.

For binary classification, an essential intermediate step has to be applied to the $\mathcal{L}(h)$ to formulate classification variables from them. It is called binary normalitzation. To perform binary classification on two species of interest labelled α and β, the binary classification variables $C(\alpha\ :\ \beta)$ are defined as:

$$C(\alpha\ :\ \beta) = \frac{\mathcal{L}(\alpha)}{\mathcal{L}(\alpha) + \mathcal{L}(\beta)} = 1 - C(\beta\ :\ \alpha) \tag{3.6}$$

The classification variables can be interpreted as the "probability" to have a α particle. $C(\alpha\ :\ \beta)$ is in the range of 0 to 1 and $C(\alpha\ :\ \beta) + C(\beta\ :\ \alpha) = 1$.

In order to identify a track as being of species α, $C(\alpha\ :\ \beta)$ is required to be above a certain threshold r. A higher threshold means greater confidence, while a lower threshold allows predictions even with less confidence. Depending on the specific objective, e.g aiming for high efficiency or low misidentification rate, one has the freedom to select the threshold, which determines how confident you want a prediction. For example, if one is interested in performing K/π separation, a track is identified as a kaon if $C(K\ :\ \pi)$ is above a chosen threshold.

3.5 Introduction of the Boosted Decision Tree

To improve lepton PID, an IA based method was previously developed [6,7]. It uses a boosted decision tree (BDT) classifier, which combines likelihood information of PID detectors with an additional so-called ECL cluster-shape variables available, in order to enhance the discrimination power.

Table 3.2: Description of the input variables for the BDT [32]. The "Range" column indicates whether a variable is defined only in a particular region of the phase space. Note that the KLM info is used only for the muon classifiers. [6]

Variable	Range	Description
E/p[c]	-	Ratio of cluster energy over track momentum.
E_1/E_9	-	Ratio of the energy of the seed crystal over the energy sum of the 9 surrounding crystals.
E_9/E_{21}	-	Ratio of the energy sum of 9 crystals surrounding the seed over the energy sum of the 25 surrounding crystals (minus 4 corners).
Cluster LAT	-	Cluster lateral moment
$\|Z_{40}\|$	-	Zernike moment $n = 4, m = 0$, calculated in a plane orthogonal to the EM shower direction.
$\|Z_{51}\|$	-	Zernike moment $n = 5, m = 1$, calculated in a plane orthogonal to the EM shower direction.
Z_{MVA}	-	Score of BDT trained on 11 Zernike moments.
ΔL[cm]	-	Projection on the extrapolated track direction of the distance between the track entry point in the ECL and the cluster centroid.
PSD_{MVA}	-	Score of a BDT trained to classify clusters as originated by an EM or hadronic shower, using crystal-level info including waveform pulse shape.
$\Delta \log \mathcal{L}(\ell/\pi)_{CDC}$ (binary)	-	Log-likelihood difference between $\ell - \pi$ hypothesis is in the CDC (binary)
$\mathcal{L}_\ell^{CDC}/\sum_i \mathcal{L}_i^{CDC}$ (multi-class)	-	Global lepton likelihood ratio in the CDC (multi-class).
$\Delta \log \mathcal{L}(\ell/\pi)_{TOP}$ (binary)	ECL Barrel†	Log-likelihood difference between $\ell - \pi$ hypothesis in the TOP (binary)
$\mathcal{L}_\ell^{TOP}/\sum_i \mathcal{L}_i^{TOP}$ (multi-class)	ECL Barrel†	Global lepton likelihood ratio in the TOP (multi-class).
$\Delta \log \mathcal{L}(\ell/\pi)_{ARICH}$ (binary)	ECL FWD endcap†	Log-likelihood difference between $\ell - \pi$ hypothesis in the ARICH (binary)
$\mathcal{L}_\ell^{ARICH}/\sum_i \mathcal{L}_i^{ARICH}$ (multi-class)	ECL FWD endcap†	Global lepton likelihood ratio in the ARICH (multi-class).
$\Delta \log \mathcal{L}(\mu/\pi)_{KLM}$ (binary)	$p_{lab} > 0.6\text{GeV}/c$	Log-likelihood difference between $\ell - \pi$ hypothesis is in the KLM (binary)
$\mathcal{L}_\ell^{KLM}/\sum_i \mathcal{L}_i^{KLM}$ (multi-class)	$p_{lab} > 0.6\text{GeV}/c$	Global lepton likelihood ratio in the KLM (multi-class).

† The ECL polar angle coverage per region is the following: ECL Forward end-cap $\in [12.4°; 31.4°]$, ECL Barrel $\in [32.2°; 128.7°]$ and ECL Backward end-cap $\in [130.7°; 155.1°]$.

A particle hitting the ECL creates a signal not only in a single cell in a cluster of cells. Hence, we can measure not only the deposited energy, but also the shape. This shape is also different for different particle species. The $\mathcal{L}^{\text{ECL}}(h)$ is computed only using the energy and momentum rate, while the ECL cluster-shape variables encode the cluster shape, offering additional information for PID.

The full list of input variables is outlined in table 3.2. This BDT is trained on a simulated sample (refer to section 4.1).

3.6 Performance Evaluation

To test the performance of a neural network, a confusion matrix is usually defined [33]. The matrix contains the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). For example for K/π separation, K is the positive category and π is the negative category. True means that the prediction is correct, while false means it is incorrect. A "true positive" means that the model accurately predicts the positive category, while a "true negative" means

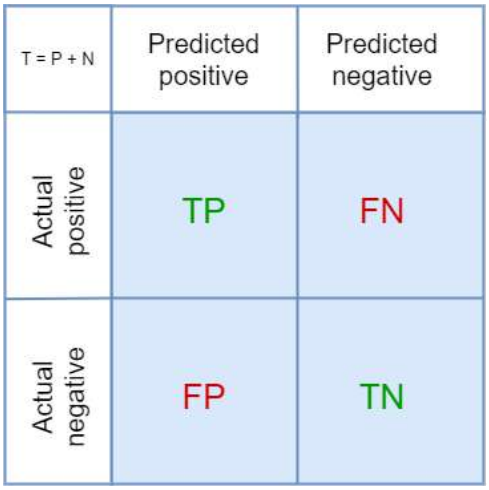


Figure 3.9: Confusion matrix. Retrieved from ref. [34]

an accurate prediction of the negative category. A "false positive" means that the model incorrectly predicts the positive category, and a "false negative" means that the model inaccurately predicts the negative category. Figure 3.9 illustrates a confusion matrix.

A ROC curve, which stands for Receiver Operating Characteristic curve, is a graphical representation that illustrates the performance of a classification method independently of the chosen threshold, by scanning across various classification thresholds. It displays two essential parameters against each other:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} \tag{3.7}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \tag{3.8}$$

In the ROC curve, the horizontal-axis represents the FPR, while the vertical-axis represents the TPR. In the following work, TPR is referred to as efficiency, and False Positive Rate is referred to as misidentification rate.

For example if we are performing K/π separation: TP are the kaons correctly identified as kaons, FN are the kaons incorrectly classified as pions, FP are the pions incorrectly classified as kaons and TN are the pions correctly classified as pions. Therefore, we can define:

$$\text{TPR} = K \text{ efficiency} = \frac{\text{Number of kaon tracks identified as a kaon}}{\text{Total number of kaon tracks}} \tag{3.9}$$

$$\text{FPR} = \pi \text{ misID-rate} = \frac{\text{Number of pion tracks identified as a kaon}}{\text{Total number of pion tracks}} \tag{3.10}$$

Analogously, the π efficiency and the K misidentification rate are defined as as:

$$\pi \text{ efficiency} = \frac{\text{Number of pion tracks identified as a pion}}{\text{Total number of pion tracks}} \tag{3.11}$$

$$K \text{ misID-rate} = \frac{\text{Number of kaon tracks identified as a pion}}{\text{Total number of kaon tracks}} \tag{3.12}$$

Chapter 4

Definition of the Neural Networks

The aim of this project is to propose an alternative method for particle identification in order to improve current performance. This is achieved through the development of a neural network, whose outputs $O_{\mathrm{NN}}(\alpha)$ replace $\mathcal{L}(h)$ in Eq. (3.6) to define classification variables for neural network PID. This is explained in detail in section 4.4.

In section 4.1, the different data sets utilized in this study are described with their main properties. Additionally, a balancing process to refine training sample is described. In section 4.2, a brief introduction to neural networks is presented, elucidating the functions and main components, needed for the PID using a neural network. In section 4.3 we describe the neural networks that are developed for PID at Belle II.

4.1 Data Sets

The data sets serve two main purposes: training and testing the neural network's performance. Depending on their purpose, they must have different properties and regimes.

For training, we require data sets which are large and cover the full kinematic range. Furthermore, they should be clean (have no background) on an event-by-event basis, i.e the target specie is well know. Simulated samples, generated via Monte-Carlo simulations, are the perfect sample for training, as they posses all the properties mentioned. However, as detector's simulation might not be perfect, we also require real-data samples.

Real-data samples are mainly used for testing, since training in real data can present numerous challenges. Additionally, it is difficult to obtain real-data samples for all particles without momentum or angular range limitations. For testing, we require large data sets, with no limitation in the kinematic range. Furthermore,the testing samples must be statistically clean, but are not required to be clean on an event-by-event basis. Initially, data and background are not separable but we have some variables which have different distribution for data and background. By mod-

elling and fitting we can assign weights for each event. Finally we can reduce the background using a statistical variable, called sweights, to obtain statistically clean samples.

Section 4.1.4 describes a process necessary to minimize bias from the intrinsic distribution of the training sample, referred to as the "balancing process."

4.1.1 Particle-Gun Monte Carlo Simulation Sample (pgMC)

In order to train the neural network that can separate all six particle species, we need a sample containing all six particle species that we want to be able to predict from: namely electrons (e), muons (μ), pions (π), kaons (K), protons (p), and deuterons (d). To address this, we use a so-called particle-gun Monte Carlo (pgMC) simulation sample for each one of these species. In the pgMC sample, the momentum of each specie was isotropically generated. The magnitude of the momentum was randomly drawn from a uniform distribution within the range of $0.001 < |\vec{p}| < 7$ GeV/c.

For each event, a certain number of charged particles is generated, referred to as multiplicity. The multiplicity influences the PID performance of the TOP detector as more tracks per event result in a high precision of the event time, needed as input for the TOP PID [35]. However, for multiplicities < 4, this effect is not well reproduced in the simulation. Therefore, we use subsamples with the following multiplicities: 4, 6, 8, 10, and 16. This helps in minimizing bias from this effect. For each subsample, the same number of tracks is generated.

Once the track is created, we need to simulate the detector response i.e we replicate how detectors interacts with particles. This process involves modelling the behaviour of each detector component. With that, we can generate simulated data that closely resembles what would be observed in a real experiment. The simulation is done using the basf2 [36] Belle II simulation framework[5].

Besides training, we use this sample for testing purposes, as it allows to the test the performance in any desired combination of particles. The sample is split into 80 % used for training the neural network and 20 % used for testing. Approximately 650,000 tracks are available for each particle for testing.

Figure 4.1 displays the kinematic distribution of reconstructed kaon tracks. It shows that the entire kinematic range is covered. The distribution observed is very similar for the other particle species. In total, the sample contains 4 557 037 kaons, 4 555 367 pions, 4 556 069 protons, 4 555 254 electrons, 4 552 777 muons and 4 555 055 deuterons for training, after the balancing process is applied (refer to section 4.1.4). Therefore it is a large sample (large number of tracks). It is clean on an event-by-event basis, as we know the true specie with no background. Thus, it possesses all

[5]For the simulation, release-06-00-08 was used.

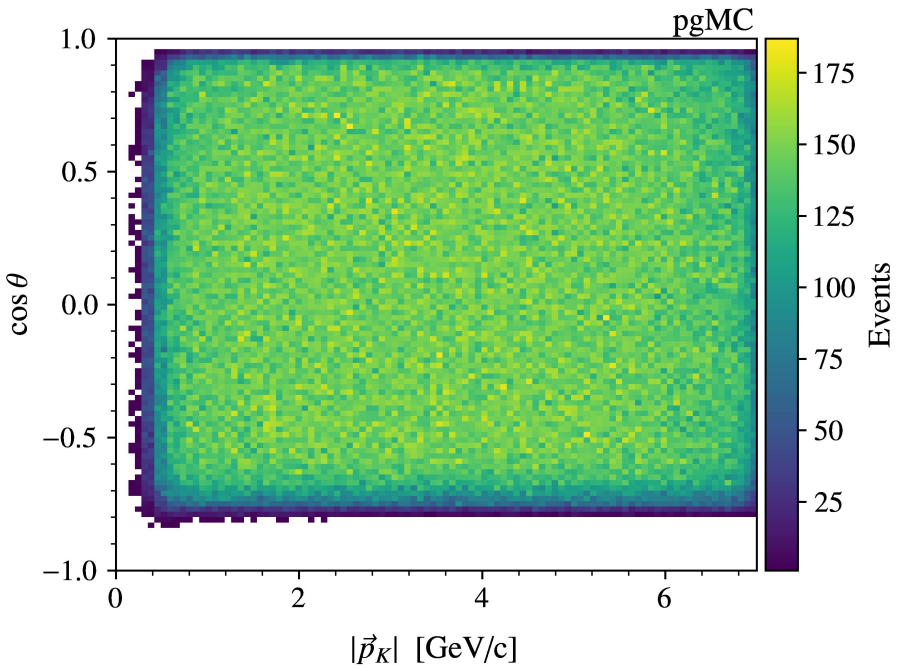


Figure 4.1: Distribution of K tracks in $\cos\theta$ and $|\vec{p}|$ from the particle-gun sample.

the features we desire in a training sample.

4.1.2 Real-Data Samples (Proc13+b)

In addition to the the simulated sample of pgMC sample, it is also necessary to have real-data samples. Three distinct types of data samples for different species are utilized, each stemming from a different decay process. To use a real-data sample, we must know the true species without using PID. For that, we use specific physics processes that produces only certain particle species as decays, so we are sure about which particles do we have. In the following sections, "Proc13+b" indicates a real-data sample.

4.1.2.1 Real-Data Sample of D^* Decays (Proc13+b D^*)

The obtain a real-data sample of pions and koans we use the D^* decay. These particles commonly decay into the following products [37]: $D^{*,+} \to D^0[\to K^-\pi^+]\ \pi^+$ and $D^{*,-} \to \bar{D}^0[\to K^+\pi^-]\ \pi^-$. The $\pi^\pm$ tags the decay, i.e defines the charge of the D^*. Then, for the two other tracks, their charge define their species. This allows to identify them without PID.

Again, we split into 80 % for training and 20 % for testing. This sample will be employed later for testing the PID performance for hadron, kaon and pion tracks. It contains 523 899 tracks for both kaons and pions corresponding to an integrated

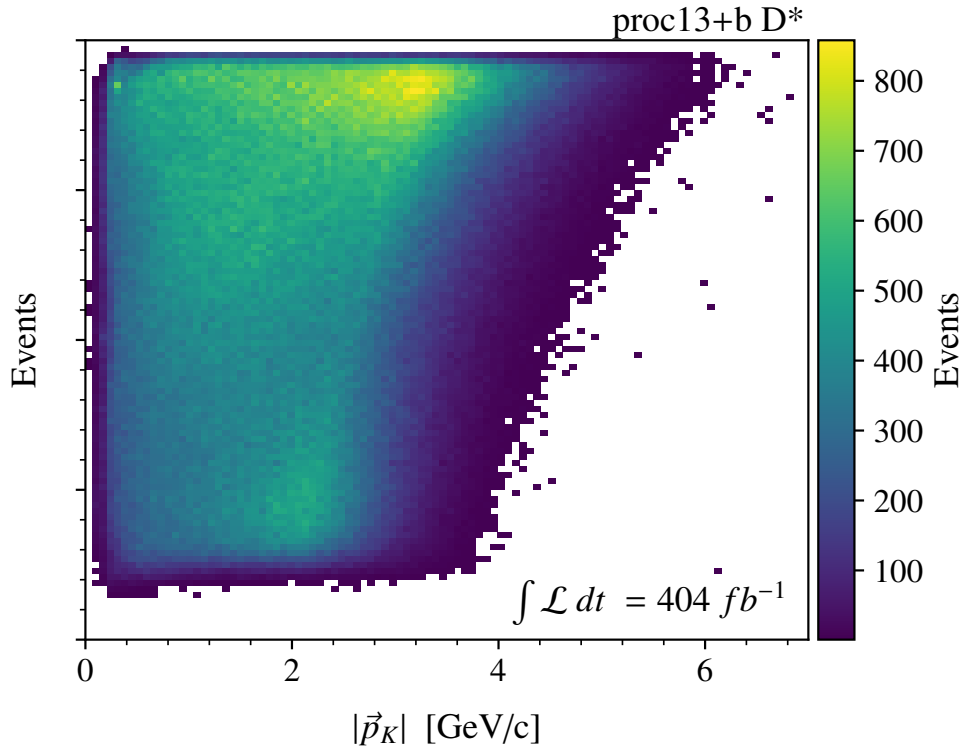


Figure 4.2: Distribution of K tracks in $\cos\theta$ and $|\vec{p}|$ from D^* decays from real data.

luminosity of $\int \mathcal{L}\mathrm{d}t = 404\,\mathrm{fb}^{-1}$. Figure 4.2 shows an example of kaon track distribution for real-data D^* sample. It highlights the main properties of the sample. It is a large sample (large number of tracks), covering a large phase-space but not the entire range. Furthermore, after we statistically remove the background using sweights variable, it is a almost background free sample.

In chapter 5, this sample is also used for training a neural network on real data. After balancing, there are approximately 1 900 000 kaon and pion tracks available for training.

4.1.2.2 Real-Data Sample of Λ_0 Decays (Proc13+b Λ_0)

To obtain a real sample of protons and pions, we use the decays of Λ_0 and $\bar{\Lambda}_0$ to $\Lambda_0 \to p\pi^-$ and $\bar{\Lambda}_0 \to \bar{p}\pi^+$ [37]. The species of the final-state particles is known from their charge and their characteristic kinematics.

Figure 4.3 (left) shows, for the Λ_0 decay, the distribution of pion tracks, which predominantly occupy the low momentum region ($0 < |\vec{p}| \lesssim 2.7$). Figure 4.3 (right) shows the distribution of protons, which covers almost the entire momentum range. The sample consists of 7 569 099 tracks which are used in the following only for testing. As explained for the D^* sample, we can use it for testing as it is a large sample,

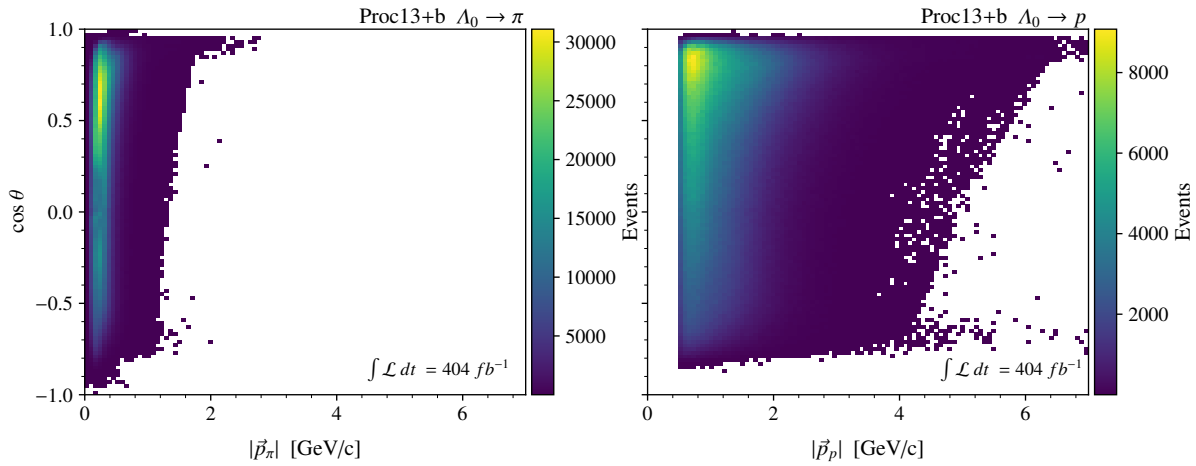


Figure 4.3: Distribution of π (left) and p (right) tracks in $\cos\theta$ and $|\vec{p}|$ of the Λ_0 decay sample.

that does not cover the full momentum range, with the background statistically removed.

4.1.2.3 Real-Data Sample of J/Ψ Decays (Proc13+b J/Ψ)

Finally, J/Ψ decays serve as a source of real-data samples for leptons, both electrons and muons. J/Ψ can decay into various final states [37, 38]. We employ the following decays: $J/\Psi \to e^+e^-$ and $J/\Psi \to \mu^+\mu^-$, which provide a real-data sample of electrons and muons. They are separated using the tag-probe approach. Figure 4.4 shows the kinematic distribution of the decay into muons and electrons. They cover a large phase-space but not the entire range. These samples consist of 1 282 804 electron tracks and 1 861 268 muon tracks, which are exclusively used for testing.

There is a substantial background in these samples. Furthermore, there exists a correlation between the J/Ψ candidate invariant mass, which is used to calculate the sweights, and the lepton momentum. This prohibits the usual background suppression method using sweights. Therefore, this sample cannot be used for momenta below 1.5 GeV/c.

4.1.3 Simulated Sample of D^* Decays (MC15rd D^*)

Comparing the performance on real data with the performance on simulated data requires a simulated sample that closely resembles the real-data sample. We use a sample which models the real-data D^* decay process, generating a simulated sample of D^* referred as MC15rd D^*.

The kinematic distribution of the simulated D^* sample is depicted in Fig. 4.5 (left). One can see that it is almost equal to the real-data D^* sample, as displayed in Fig. 4.2 (right).

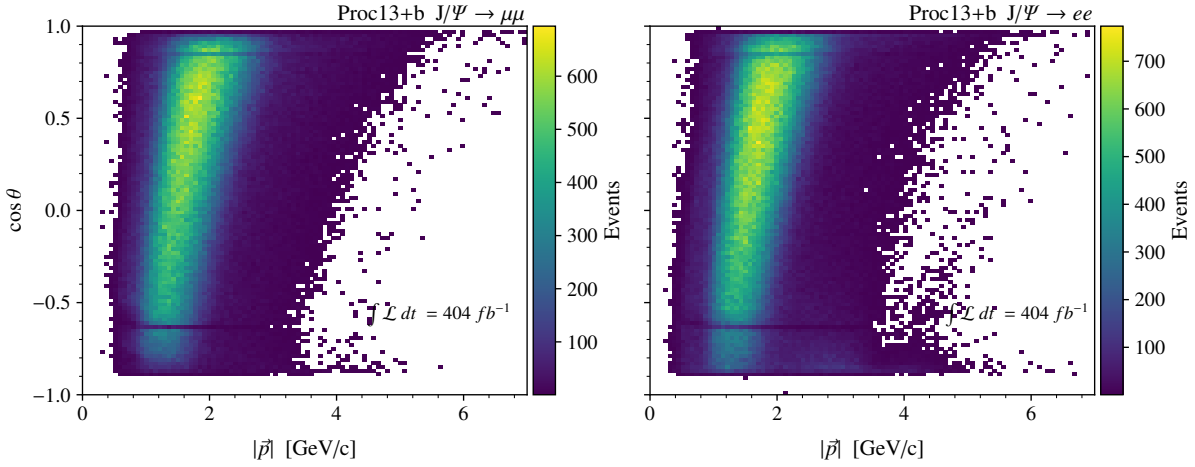


Figure 4.4: Distribution of μ (left) and e (right) tracks in $\cos\theta$ and $|\vec{p}|$ from the J/Ψ decay sample.

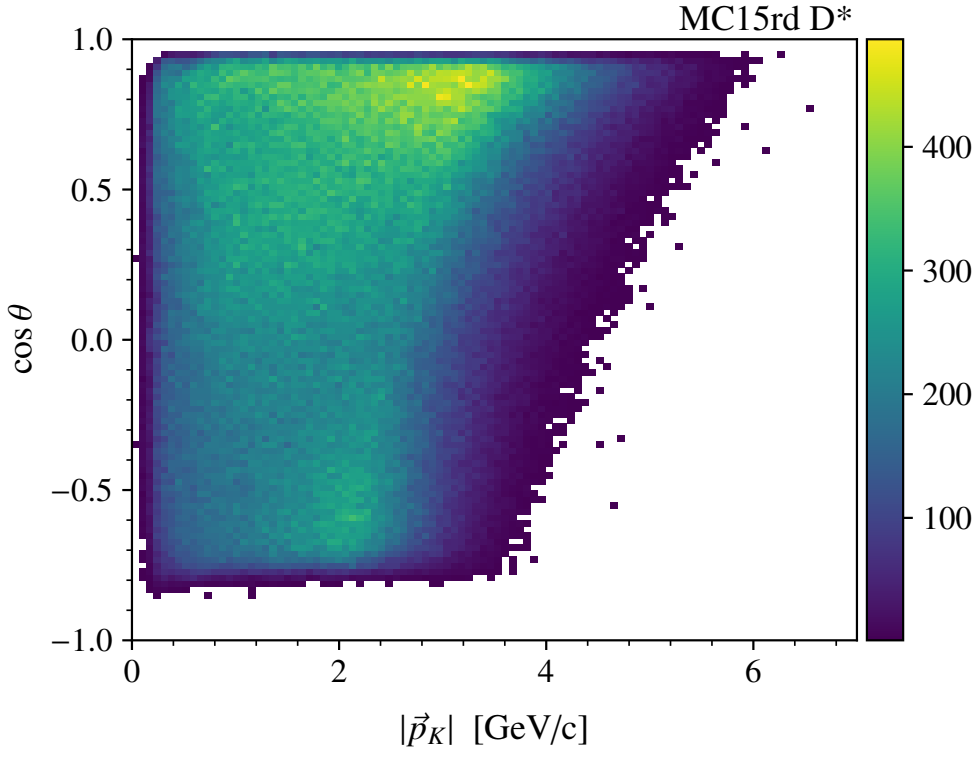


Figure 4.5: Distribution of K tracks in $\cos\theta$ and $|\vec{p}|$ from D^* decays from simulation.

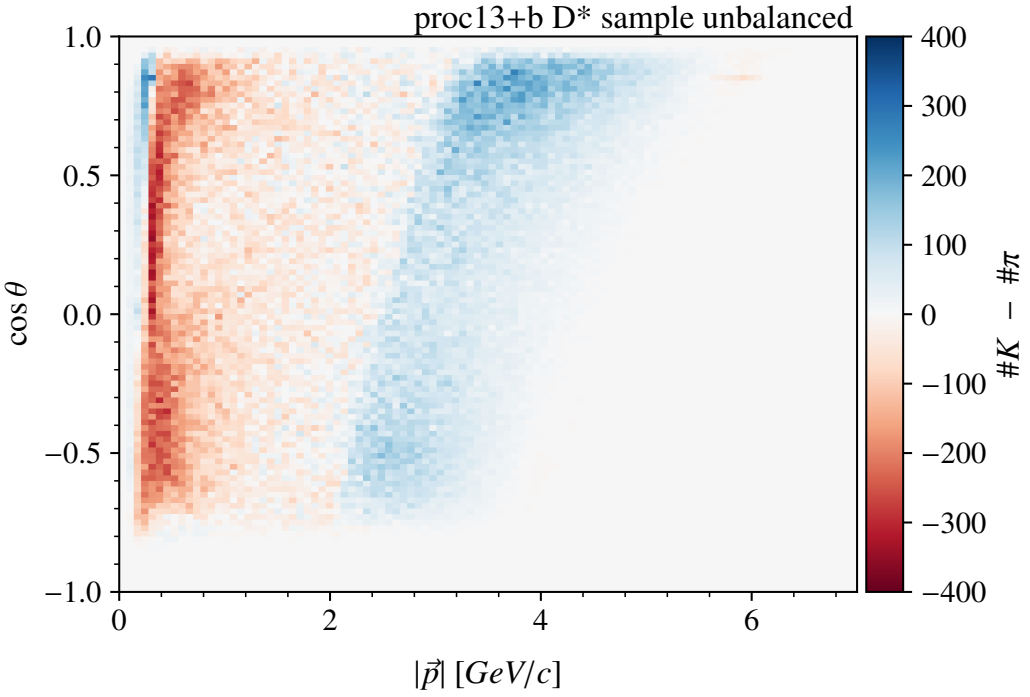


Figure 4.6: Difference between the number of K and π tracks as a function of $\cos\theta$ and $|\vec{p}|$ for the real-data D^* sample before performing the balancing process.

4.1.4 Balancing of Training Samples

The primary objective of the neural network is to capture the detector responses for a specific particle species. At the same time, it is essential to prevent the neural network from learning specific characteristics of the training data set. The major characteristic for the D^* sample is that, there are more kaons in the high-momentum region and more pions in the low momentum region (see Fig. 4.6). If the neural network were to learn this particular feature, it might exhibit a bias towards the kaon hypotheses in the high-momentum region. This potential bias could adversely affect the network's performance on data sets with another kinematic distributions.

To address this, a balancing process is implemented for each training sample. To this end, the K and π subsamples are divided into ($\cos\theta$, $|\vec{p}|$) cells [6]. Within each cell, K tracks are randomly excluded, if there are more K tracks than π tracks according to the ratio of the number of K and π tracks. If there are more π tracks, π tracks, are randomly excluded. This procedure ensures that the distribution of K and π tracks in the balanced sample is is the same in each cell, as shown in Fig. 4.7.

The particle-gun sample is balanced by construction. However, due to the momentum-dependent reconstruction efficiency, which depends on the particle specie, a slight imbalance arises in the particle-gun sample after reconstruction, as illus-

[6] The range $0 < |\vec{p}| < 7$ GeV/c in momentum and the range $-1 < \cos\theta < 1$ in $\cos\theta$ are divided into 100 equally spaced bins, which yields in total 10000 cells.

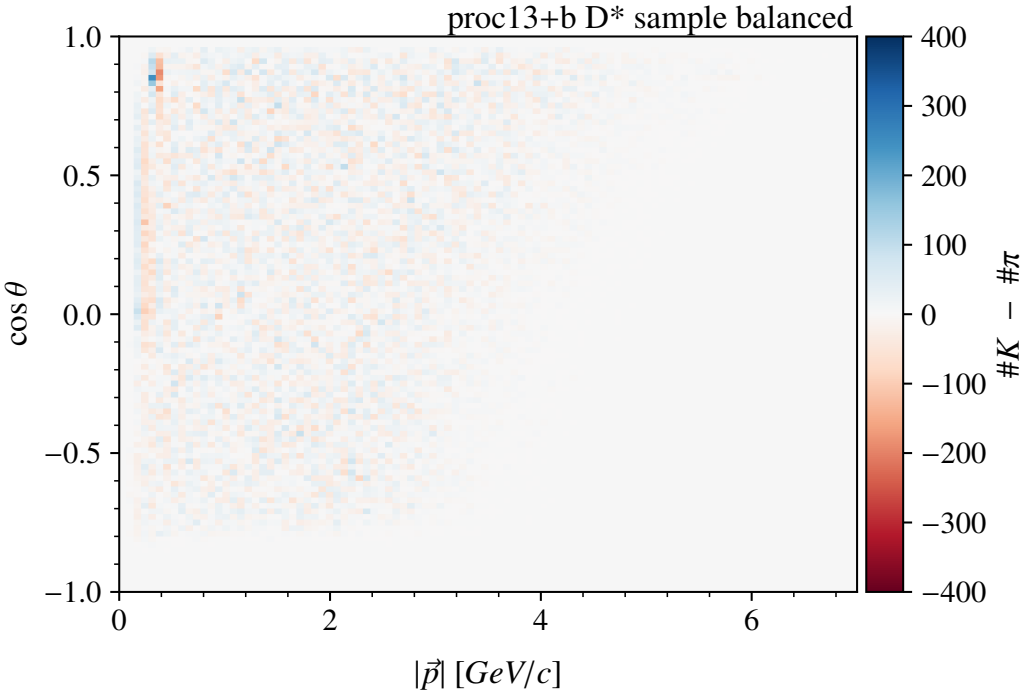


Figure 4.7: Same as Fig. 4.6, but after the balancing process. The artefact that appears close to 0 GeV/c is due to the binning chosen and does not affect the analysis.

trated in Fig. 4.8 for kaons and pions. To mitigate potential bias resulting from this effect, we applied the balancing procedure also to the particle-gun sample before utilizing it for training. For particle-gun sample we balance the six species simultaneously. The distribution of the resulting balanced sample is shown in Fig. 4.9 for kaons and pions. For the other species, the same plot is obtained.

One should note that the balancing procedure was exclusively applied to the training samples and not to the testing samples.

4.2 General Neural Network Theory

Neural networks [39,40] are a powerful paradigm in machine learning and data analysis. Their ability to learn from data and make predictions makes them invaluable tools across a wide range of applications. They are inspired by the intricate biological neural networks found in the human brain, which by adhering to simple rules, enables them to learn highly intricate relations and find complex patterns.

The simplest neural network are called multi layer perceptron (MLP). They are a feedforward artificial neural network with fully connected neurons, divided in different layers. A basic depiction of a fully connected network is provided in Fig. 4.10a.

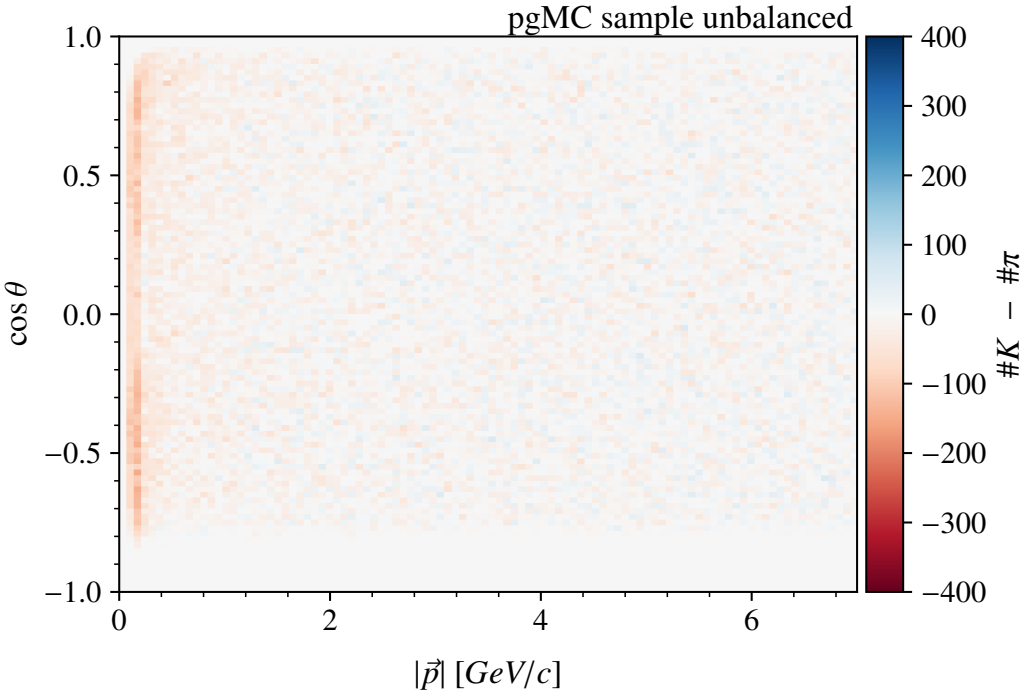


Figure 4.8: Difference between the number of K and π tracks as a function of $cos\theta$ and $|\vec{p}|$ for the particle-gun sample before performing the balancing process.

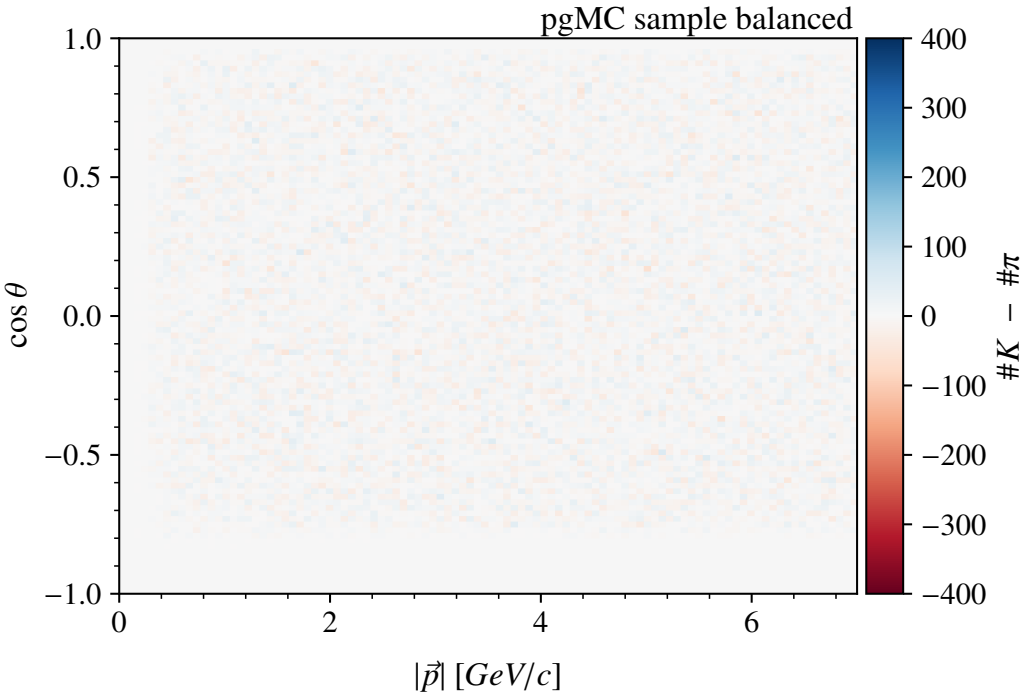
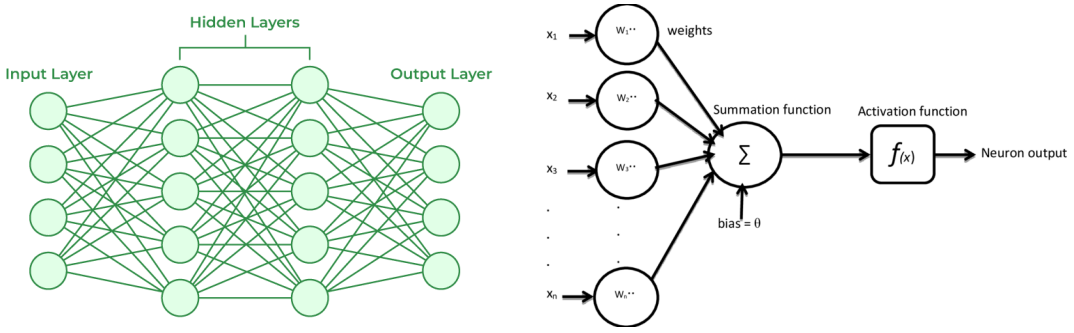


Figure 4.9: Same as Fig. 4.8, but after the balancing process.



(a) (b)

Figure 4.10: (a) Schematic drawing of a fully-connected neural network. (b) Schematic drawing of a neuron.

The neural network initiates with the first layer, called the input layer, where data that enters the network is commonly know as inputs. It is followed by a series of layers which are called hidden layers. Here, one can select the number of hidden layers according to the problem one want to solve. The neurons of one layer receive inputs originating from the previous layer. The hidden layer transform the inputs using the weights in each of their nodes. Finally, they use an activation function, which can introduce non-linearity into the network. Therefore, the outputs of a layer are the inputs to its following layer. In this manner, the information is transported from the input layer to the last layer, know as the output layer. The output layer processes the outputs of the last hidden layer of the network and its neurons gives the network's response, allowing it to learn and approximate complex relationships in the data. The choice of activation function should align with the desired task, specially for the activation function in the output layer.

Each neuron nodes works in a similar way, illustrated in Fig. 4.10b. The inputs, which are numbers, are multiplied by a weight w_i and a bias is added to the weighted inputs. Finally, the resulting value z is passed through an activation function σ. The output from one hidden layer serves as the input for the subsequent hidden layer. Mathematically, this process $f(\{x_i\})$ can be expressed as:

$$f(\{x_i\}) = \sigma(z) = \sigma(b + \sum_i x_i w_i) \tag{4.1}$$

The neural network must be trained to make accurate predictions. This means that the weights and biases have to be tuned to have an optimal performance of the neural network. In our case, we are working with supervised learning. This requires a so-called labeled data set for training, which not only includes the input variables, but also have its target values. During the training the neural network is evaluated on the training samples and its predictions (using Eq. (4.1)) are compared with the target values. To quantify the difference between the prediction and the true target, a loss function is used. There are several possibilities, which depend on the problem. Using a method known as back-propagation [40] the loss function is

minimised by changing the weights and biases in an iterative procedure. By doing so, the neural network learns from the provided training data and becomes able to make predictions on unseen data.

The architecture of a neural network plays a crucial role in its effectiveness and it clearly depends on the performed task. We are using a fully connected neural network. Its most important hyperparameters are the number of hidden layers, the number of nodes per layer and the activation function. The specific value of these hyperparameters for our neural network are described in section 4.3. Additional hyperparameters of training process, such as the learning rate, loss function, batch size, epochs, and optimization algorithm, will be also detailed.

4.3 The Ingredients of Our Neural Networks

4.3.1 Definition of Inputs

We first define the inputs of the neural networks. Our goal is to have all the necessary inputs that contain valuable information for PID. We use the $\mathcal{L}^{\mathrm{D}}(h)$ for each specie h and each detector D (see section 3.3), as they encode the measured detector high level information. However, the log-likelihoods are used instead for numerical stability. In total, 36 log-likelihoods are used.

Additionally, we include kinematic information of the measured particle track. These are the angles in the spherical coordinate system (see section 2.2), the magnitude of the momentum $|\vec{p}|$, and the charge. Kinematic variables might give valuable information for PID as the modelling of $\mathcal{L}^{\mathrm{D}}(h)$ is imperfect.

Furthermore, we aim for lepton PID. As explained before, the likelihood of the ECL only uses E/p information. To improve it we consider other variables, called ECL cluster-shape variables, which offer more information for lepton PID. They have already proven to give valuable information for PID in BDT method, as explained in section 3.5.

The entire list of input variables is shown in table 4.1.

4.3.2 Architecture

Throughout this work, three different neural networks will be compared. The first one aims for K/π separation. The second one aims for six species separation, focusing on hadron PID. The third aims for six species separation, focusing on hadron PID and lepton PID. For all the networks, we use a fully connected layer architecture. Nevertheless, they present some minor difference in structure. The networks and their are summarised in table 4.2. The neural network are the following:

1. **Neural network for 2 species**: The architecture of this neural network is illustrated in Fig. 4.11. In total, it has 40 input variables (red nodes in

Table 4.1: List of input variables of the neural network. A detail description of the ECL cluster-shape variables for the BDT is given in ref. [32]

Inputs Set 1

Variable	Description
$\log \mathcal{L}^{\mathrm{ARICH}}(\mu)$	PID log-likelihood value for μ from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(\mu)$	PID log-likelihood value for μ from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(\mu)$	PID log-likelihood value for μ from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(\mu)$	PID log-likelihood value for μ from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(\mu)$	PID log-likelihood value for μ from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(\mu)$	PID log-likelihood value for μ from TOP
$\log \mathcal{L}^{\mathrm{ARICH}}(p)$	PID log-likelihood value for p from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(p)$	PID log-likelihood value for p from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(p)$	PID log-likelihood value for p from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(p)$	PID log-likelihood value for p from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(p)$	PID log-likelihood value for p from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(p)$	PID log-likelihood value for p from TOP
$\log \mathcal{L}^{\mathrm{ARICH}}(\pi)$	PID log-likelihood value for π from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(\pi)$	PID log-likelihood value for π from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(\pi)$	PID log-likelihood value for π from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(\pi)$	PID log-likelihood value for π from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(\pi)$	PID log-likelihood value for π from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(\pi)$	PID log-likelihood value for π from TOP
$\log \mathcal{L}^{\mathrm{ARICH}}(K)$	PID log-likelihood value for K from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(K)$	PID log-likelihood value for K from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(K)$	PID log-likelihood value for K from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(K)$	PID log-likelihood value for K from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(K)$	PID log-likelihood value for K from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(K)$	PID log-likelihood value for K from TOP
$\log \mathcal{L}^{\mathrm{ARICH}}(d)$	PID log-likelihood value for d from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(d)$	PID log-likelihood value for d from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(d)$	PID log-likelihood value for d from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(d)$	PID log-likelihood value for d from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(d)$	PID log-likelihood value for d from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(d)$	PID log-likelihood value for d from TOP
$\log \mathcal{L}^{\mathrm{ARICH}}(e)$	PID log-likelihood value for e from ARICH
$\log \mathcal{L}^{\mathrm{CDC}}(e)$	PID log-likelihood value for e from CDC
$\log \mathcal{L}^{\mathrm{ECL}}(e)$	PID log-likelihood value for e from ECL
$\log \mathcal{L}^{\mathrm{KLM}}(e)$	PID log-likelihood value for e from KLM
$\log \mathcal{L}^{\mathrm{SVD}}(e)$	PID log-likelihood value for e from SVD
$\log \mathcal{L}^{\mathrm{TOP}}(e)$	PID log-likelihood value for e from TOP
$\cos\theta$	Cosine of polar angle of momentum (in lab frame)
ϕ	Azimuthal angle of momentum (in lab frame)
p	Magnitude of momentum (in lab frame)
charge	Electric charge of particle in units of e_0

Inputs Set 2

Variable	Description
E/p[c]	Ratio of cluster energy over track momentum.
E_1/E_9	Ratio of the energy of the seed crystal over the energy sum of the 9 surrounding crystals.
E_9/E_{21}	Ratio of the energy sum of 9 crystals surrounding the seed over the energy sum of the 25 surrounding crystals (minus 4 corners).
Cluster LAT	Cluster lateral moment
$\lvert Z_{40}\rvert$	Zernike moment $n = 4, m = 0$, calculated in a plane orthogonal to the EM shower direction.
$\lvert Z_{51}\rvert$	Zernike moment $n = 5, m = 1$, calculated in a plane orthogonal to the EM shower direction.
Z_{MVA}	Score of BDT trained on 11 Zernike moments.
ΔL[cm]	Projection on the extrapolated track direction of the distance between the track entry point in the ECL and the cluster centroid.
PSD_{MVA}	Score of a BDT trained to classify clusters as originated by an EM or hadronic shower, using crystal-level info including waveform pulse shape.

Table 4.2: Comparison of the neural networks used including its code, name, inputs (see table 4.1), and outputs.

Code	Neural Network	Inputs	Direct Outputs
001	Neural network for 2 species	Inputs Set 1	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$
002	Neural network for 6 species without cluster-shape	Inputs Set 1	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$
003	Neural network for 6 species with cluster-shape	Inputs Set 1 + Inputs Set 2	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$

Fig. 4.11). The complete list of these 40 input variables is summarised in table 4.1 under the name of Inputs Set 1. The determination of network's hyperparameters is discussed in section 5.1. Finally, the selected set of hyperparameters consists of 512 nodes per layer and 2 hidden layers represented by the blue nodes in Fig. 4.11.

This neural network aims to only distinguish between kaons and pions. For this purpose, this neural network has two outputs, denoted as $O_{\mathrm{NN}}(K)$ and $O_{\mathrm{NN}}(\pi)$, which are indicated by the green nodes in Fig. 4.11. It is identified as neural network 001 in table 4.2.

2. **Neural network for 6 species without cluster-shape**: The second neural network has the same 40 inputs (Inputs Set 1) as the previous neural network. Furthermore, it has the same hidden layer structure.

   However, this neural network aims to simultaneously predict for all six species. For that, the number of outputs is different. This neural network has six outputs which are $O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$, $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$ and $O_{\mathrm{NN}}(d)$. The architecture is illustrated in Fig. 4.12. This neural network is labeled as 002 in table 4.2.

3. **Neural network for 6 species with cluster-shape**: This neural networks aims to simultaneously predict for all six species. Hence, this neural network has the same structure and outputs as the previous one. The only difference is that 9 extra inputs are added (Inputs Set 2), to extend the network for lepton PID. The reason of adding this extra inputs is discussed in detail in section 6.2. Thus, in total the input of this network are: Inputs Set 1 + Inputs Set 2, giving a total of 49 inputs. It is illustrated with Fig. 4.12. It is coded as 003 in table 4.2.

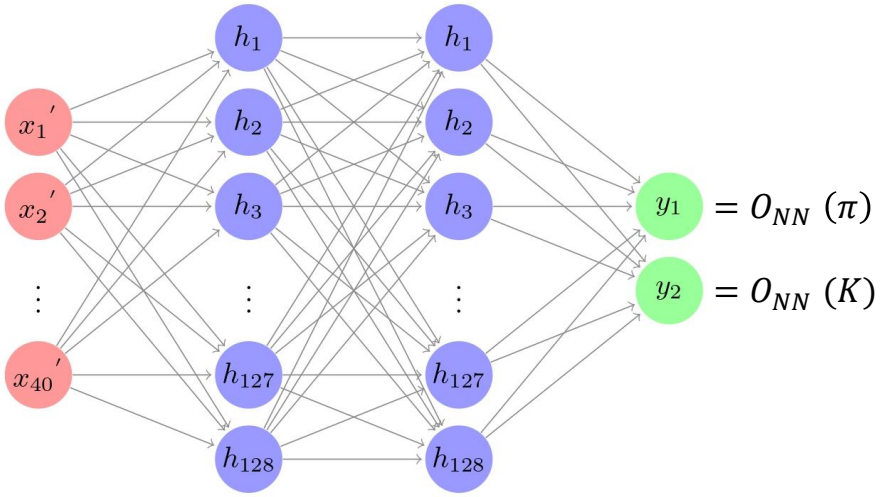


Figure 4.11: Schematic representation of the neural network for 2 species. The red nodes represent the input variables. The blue nodes represent the nodes of the two hidden layers. The green nodes represent the output variables.

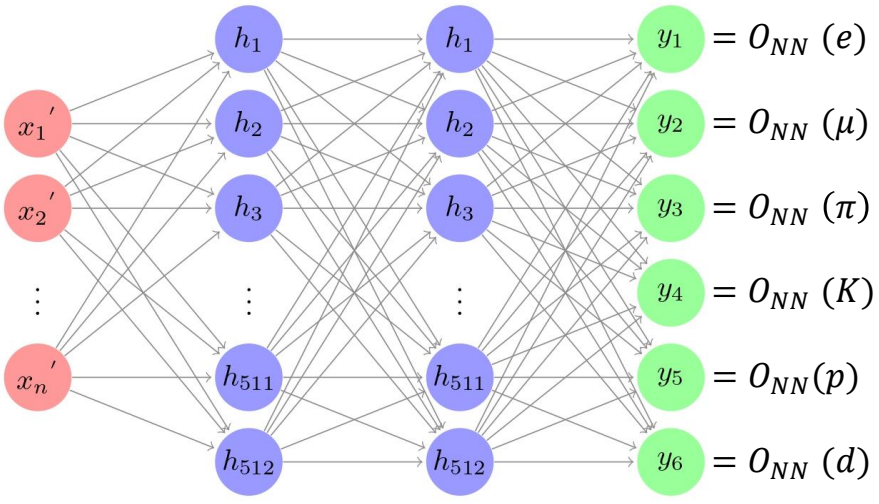


Figure 4.12: Show the same as Fig. 4.11 network for 6 species without cluster-shape ($n = 40$) and with cluster-shape ($n = 49$).

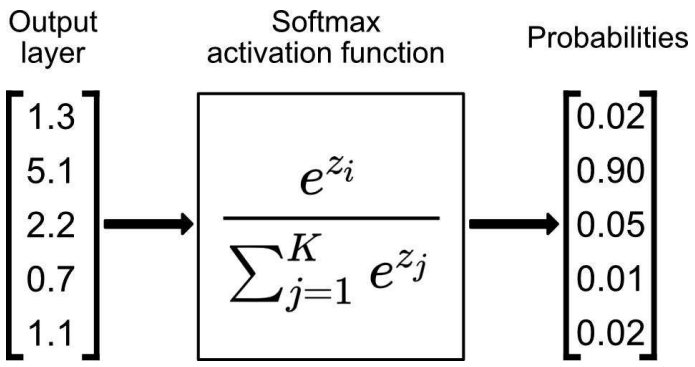


Figure 4.13: Example of a Softmax activation function.

All three neural networks share the following properties. Each of the hidden layers consists of a linear layer, with a PReLU activation function :

$$\text{PReLU}(x) = max(0, x) + a \times min(0, x), \tag{4.2}$$

with a single free parameter a. Furthermore, the PReLU is followed by a dropout layer with a dropout rate of 0.4. This means setting randomly neurons to zero during training, which helps to prevent overfitting.

The activation function of the output layer is different from the one at hidden layers. The choice of activation function in the output layer depends on the problem to solve. For multi-class classification tasks, the Softmax function is widely used for the output layer, as it ensures normalitzation. With this activation function, each output of the neural network is in the range $(0, 1)$ and all the outputs sum up to 1, as shown in Fig. 4.13 with a random example.

Despite all neural networks will be used and compared, the neural network for 6 species with cluster-shape represents the final result of this work. Its architecture is summarized in table 4.3, detailing its layers. In total, the neural network consists of 291336 free parameters, which are determined during the training process.

4.3.3 Input Normalization

To improve the neural network learning process it is a common procedure to normalize the input variables such that are centered around zero with a standard deviation of one [41].To this end, the input variables x_i: $\cos\theta$, ϕ, p, and charge; are normalized according to:

$$x_i' = \frac{x_i - \text{mean}(x_i)}{\text{std}(x_i)}. \tag{4.3}$$

Table 4.3: TOP: Neural network architecture from input layer (top) to output layer (bottom). The first column gives the layer type, the second and third columns the input and output shapes, respectively. The last column gives the number of free parameters of each layer. BOTTOM: Number of total, trainable and non-trainable parameters of the neural network.

Layer	**Input Shape**	**Output Shape**	**Parameters #**
Neural Network	[1, 49]	[1, 6]	–
Linear	[1, 49]	[1, 512]	25,600
PReLU	[1, 512]	[1, 512]	1
Dropout	[1, 512]	[1, 512]	–
Linear	[1, 512]	[1, 512]	262,656
PReLU	[1, 512]	[1, 512]	1
Dropout	[1, 512]	[1, 512]	–
Linear	[1, 512]	[1, 6]	3,078
Dropout	[1, 6]	[1, 6]	–
LogSoftmax	[1, 6]	[1, 6]	–
Type of Parameters			**Parameters #**
Total parameters			291,336
Trainable parameters			291,336
Non-trainable parameters			0

Here, mean(x_i) is the mean value of the input variable x_i, and std(x_i) its standard deviation.

Not all six PID detectors contribute information for a given track, due to the fact that detectors cover disjoint $\cos\theta$ ranges (see table 3.1). Handling cases where no PID information is available in the neural network is done by assigning them a specific value to differentiate them from events with information. In our case, we set them to 1, which is the value chosen to encode missing information. The mean value is not subtracted for the other non-missing log-likelihood values. Hence, they primarily remain below 1. Consequently, the value 1 is reserved for missing information. This same treatment is applied for any variable in Inputs Set 2 exhibiting a missing value. Overall, we implement the following normalization procedure for the log-likelihood input variables and Inputs Set 2:

$$x_i' = \begin{cases} 1 & \text{if } x_i \text{ is missing} \\ \frac{x_i}{\text{std}(x_i)} & \text{else} \end{cases} \tag{4.4}$$

Furthermore, there are some log-likelihood input variables which require a special treatment. We found out that the log-likelihood value for the electron hypothesis from the SVD, $\log \mathcal{L}^{\text{SVD}}(e)$, is not correctly implemented for simulated data. As shown in Fig. 4.14, a discrepancy is evident between simulated and real data, where for momenta above 2 GeV/c, the $\log \mathcal{L}^{\text{SVD}}(e)$ for simulated data is set at a fixed value of -6.907755374908447. This discrepancy can be treated in the neural net-

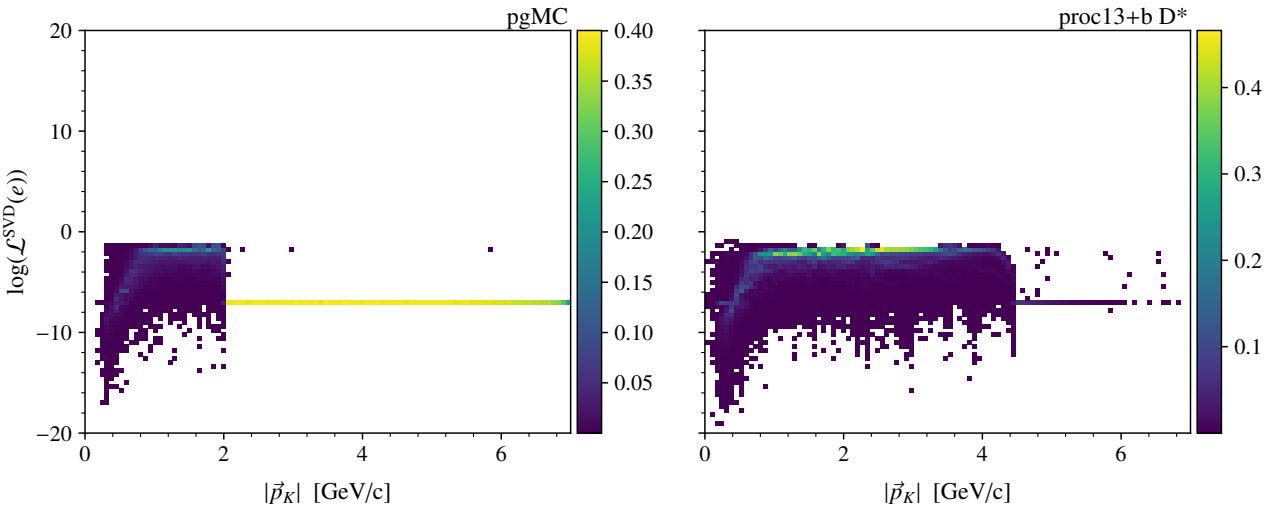


Figure 4.14: SVD log-likelihood for electron hypothesis for K tracks from (left) the particle-gun simulation and from (right) real-data D^* decays. The z-axis shows the density of events.

work by fixing the value of $\log \mathcal{L}^{\mathrm{SVD}}(e)$ to -6.907755374908447 for $|\vec{p}| > 2$ GeV/c. As the SVD does not give valuable PID information in this momentum region, no information is lost by this approach.

Second, we observe real-data simulation discrepancies for the TOP detector log-likelihood values in the region $-0.55 < \cos\theta < -0.50$, as discussed in section 6.4.2. This problem is resolved by not using the TOP information according to Eq. (4.4) in the range $-0.55 < \cos\theta < -0.50$.

4.3.4 Training of the Neural Network

The neural networks were trained on the particle-gun sample (see section 4.1.1). The training sample is split into a subsample used for training (90 %) and a subsample used for validation (10 %).

The PyTorch library [42] served as the framework for building and training the neural network. The selected loss function is the the negative log-likelihood loss function (NLLLoss in PyTorch) which is recommended for classification problems with multiple classes.

The optimization algorithm determines how the weights and biases are updated during training. Thus, the optimizer's goal is to minimize the loss function and thereby optimize the network's performance at each epoch. The Adam optimiser was chosen for this project. It is an extended version of stochastic gradient descent. ADAM considers both first and second moments of the gradient, leading to faster convergence compared to alternative strategies. That is to say, it combines the advantages of the Gradient Descent with Momentum and the Root Mean Square

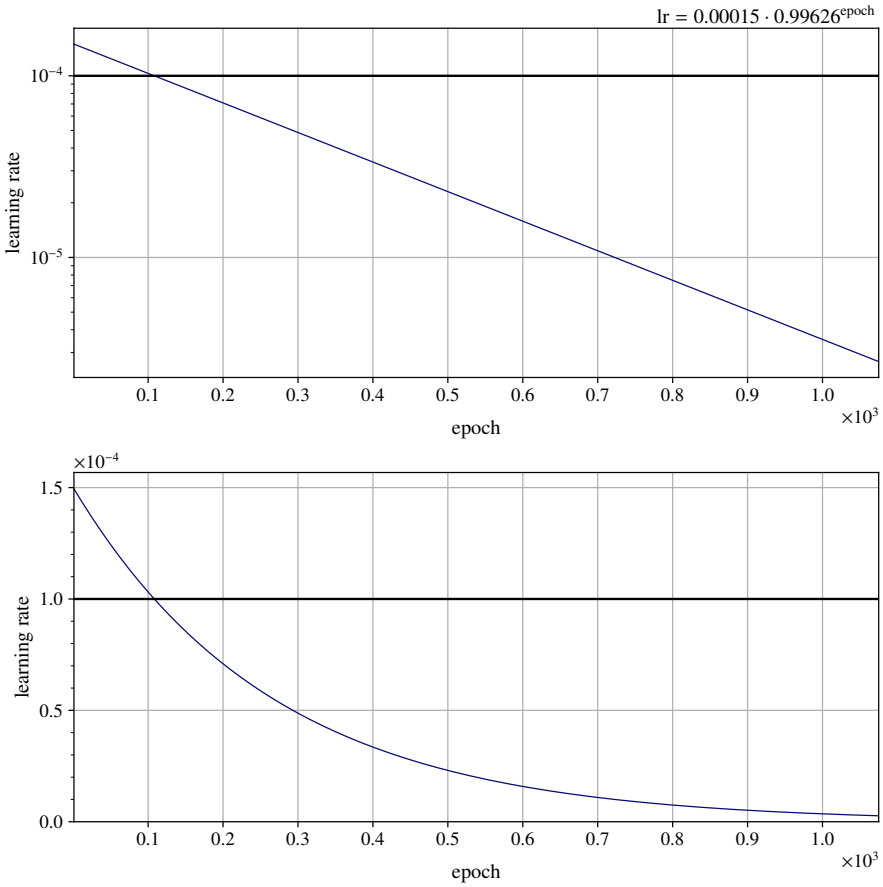


Figure 4.15: Learning rate as a function of the epoch according to the employed exponential learning-rate scheduler in (top) log scale and (bottom) linear scale.

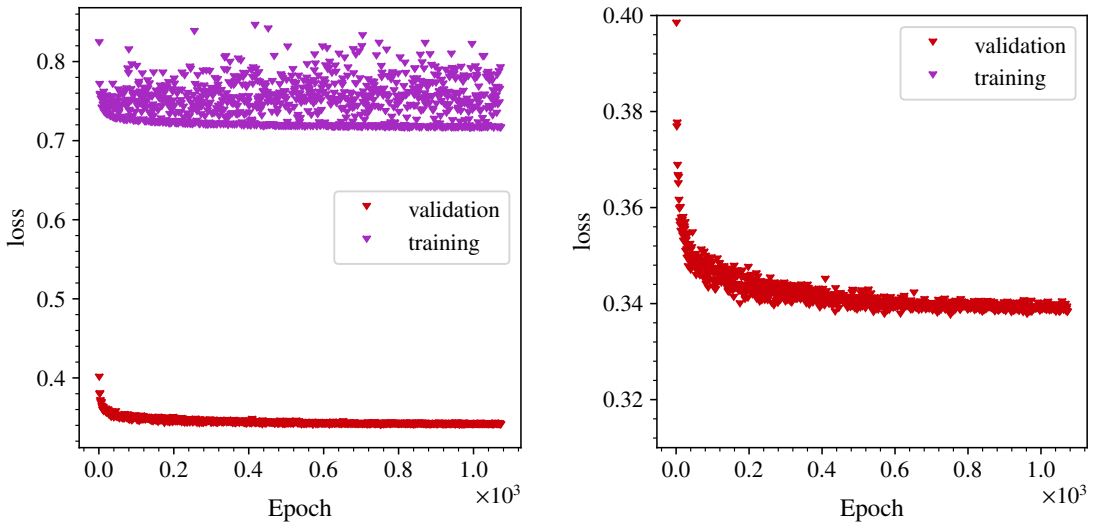
Propagation (RMSP) [40, 43].

The learning rate is a hyperparameter that controls how quickly the network parameters values are adjusted in each training step. An ideal learning rate is balanced between being low enough for the network to converge on a good solution but should be high enough to complete training in a reasonable time. In our case, the initial learning is 1.5×10^{-4}, obtained by trial and error. Then, we use an exponential learning-rate scheduler which reduces the learning rate by a factor of 0.99626 at each epoch

$$\text{Learning Rate (epoch)} = \text{Learning rate (0)}\ \gamma^{\text{epoch}} \tag{4.5}$$

In this way, when we are closer to the solution we have more precision. This process is shown in Fig. 4.15.

Finally, we have to select the better epoch among all the ones used for training. As an example, we show how it is done for the neural network for 6 species with cluster-shape. We train the neural network until we observe convergence in the loss function. The neural network was trained with a batch size of 128, and after 1075 epochs convergence was observed. Figure 4.16 illustrates the loss values



(a) (b)

Figure 4.16: Loss function for the training (purple) and the validation (red) sample. The dip of the arrow indicates the loss value. (a) shows the full loss range. (b) shows a zoom in the region of the loss function of the validation sample.

obtained during training for both the training sample (purple) and the validation sample (red). The training loss function is noisy due to the high dropout rate in the layers. The noisiness introduced by dropout is outweighed by the benefits of avoiding overfitting. Furthermore, the validation loss function converges to a stable value, without showing signs of overfitting. Further, Fig. 4.17 shows the area under the ROC curve (AUROC) for the validation sample per epoch, which also illustrates good convergence. Finally, we select the epoch, in this case 1073, which gives the largest AUROC for the validation sample, among the 1075 epochs.

4.4 Binary Classification Variables of the Neural Network

The neural network for six species, both with and without cluster-shape, can predict among six different species simultaneously. They have six outputs denoted as $O_{\mathrm{NN}}(h)$; h representing six possible species. To perform binary classification, we have to apply the same normalization procedure as described in section 3.4 for the Pure Likelihood. We replace $\mathcal{L}(h)$ in Eq. (3.6) by the outputs $O_{\mathrm{NN}}(h)$:

$$C(\alpha \ : \ \beta) = \frac{O_{\mathrm{NN}}(\alpha)}{O_{\mathrm{NN}}(\alpha) + O_{\mathrm{NN}}(\beta)} \tag{4.6}$$

It's worth noting that binary normalization is not necessary for the neural network work for 2 species, as it exclusively predicts only K and π species, and it is already normalized.

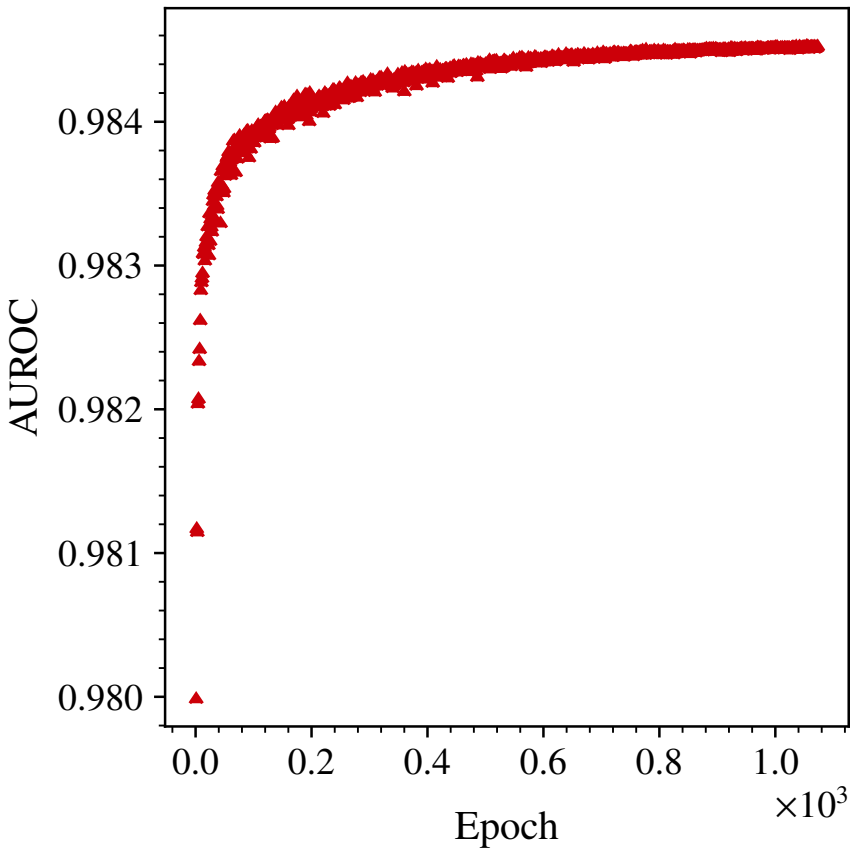


Figure 4.17: Evolution of the area under the ROC curve (AUROC) for the validation sample with the training. The dip of the arrow indicates the AUROC value.

Chapter 5

Neural Network for K/π Separation

Although the ultimate objective of this work is to develop a novel method capable of simultaneously separating multiple particles species, our basis is the neural network for K/π proposed by Wallner. Therefore, the first step is to implement some improvements for the neural network for K/π separation, before extending it to six species.

In this work, we studied the network architecture and its impact on performance is presented in section 5.1. Also, we did the final performance evaluation on real and simulated data, as discussed in section 5.2. In section 5.3 we compare a neural network trained on simulated data with a neural network trained on real data. Finally, in section 5.4, we conduct an analysis of the neural network's input effects, commonly referred to as 'feature importance'.

5.1 Neural Network Hyperparameters Optimization

The first step is to find the optimal architecture of the neural network, since the architecture affects its performance. We designed the hidden layers of network with a diamond shape, using the formula $N_i = N_0/\alpha^i$. The architecture is characterized by three key hyperparameters: the number of hidden layers, the number of nodes in each of the two central hidden layers N_0, and the exponential function slope α. The simplest neural network has an input layer and the output layer; where the number of nodes are the number of inputs and outputs respectively. Further, it has two central layers of N_0 nodes. Furthermore, we can add more layers in each side of the two central layers. The number of nodes of each extra layer depends on the α parameter and the distance to the center i. The hidden layers of the neural network are symmetrical with respect to the center.

To identify the optimal combination of these hyperparameters, we employed the

Optuna library [44] for hyperparameter exploration. As an initial analysis, the number of hidden layers is chosen from the set (2, 4, 6, 8), $N_0 \in (16, 1008)$, and α, from the set (1, 2, 3, and 4). Optuna scans the hyperparameter space, by creating and evaluating models with different parameters sets. Optuna does not try all parameter combinations but chooses the next set of parameters to try according to the performance of the previous models. Models with different parameter sets were tested by training a neural network for 600 epochs. The validation performance was assessed using the AUROC for the validation sample.

The results of this hyperparameter optimization process are shown in table 5.1. The first conclusion is that more parameters do not necessarily mean better performance, but there is a tendency that large models perform better. However, there are models with fewer parameters (intermediate models), which have similar performance; for example the one with 2 hidden layers of 80 nodes. Finally, very small models, e.g 2 hidden layers of 80 nodes, exhibit a clear decrease in performance.

Overall, the performance of the network, as indicated by the AUROC score, did not exhibit a strong sensitivity to the choice of hyperparameters in the tested range. In addition to the PID performance, other factors such as computational time and network size have to be considered. This is crucial to ensure that the neural network remains practical for implementation. Due to the similar performance observed in the tests, and considering the computational time, we can define three different neural networks with their hyperparameters described in table 5.2. The large network is chosen among the largest models with best performance. It has 2 hidden layers of 640 nodes. The medium network, with a moderate number of parameters, has 2 hidden layers of 128 nodes, but still with a similar AUROC as the large network. The small network has 2 hidden layers of 80 nodes, at the expense of a significant drop in AUROC.

For each of these selected network hyperparameter sets, we conducted a full training over 4000 epochs. The resulting ROC curve outcomes, tested on the real-data D^* sample, are illustrated in Fig. 5.1. The observed results can be summarized as follows: (i) the large network exhibits the highest performance; (ii) the medium network exhibits performance comparable to the large networks; and (iii) the small network exhibits a notable drop in performance compared to the others.

As shown, the medium network exhibits performance similar to the large network, despite having significantly fewer nodes, which translates into fast computation[7]. Therefore, with computational efficiency in mind, we chose the medium network architecture, with 2 layers of 128 nodes, for the neural network for K/π separation. This neural network is implemented into the Belle II software. Detailed studies with this neural network can be found in Wallner et al. [45].

[7]In basf2, the Belle II software, the difference is 0.07 ms/call vs. 0.20 ms/call, between the medium and large neural network. Furthermore, the computational costs of the small network are about the same as those of the medium network.

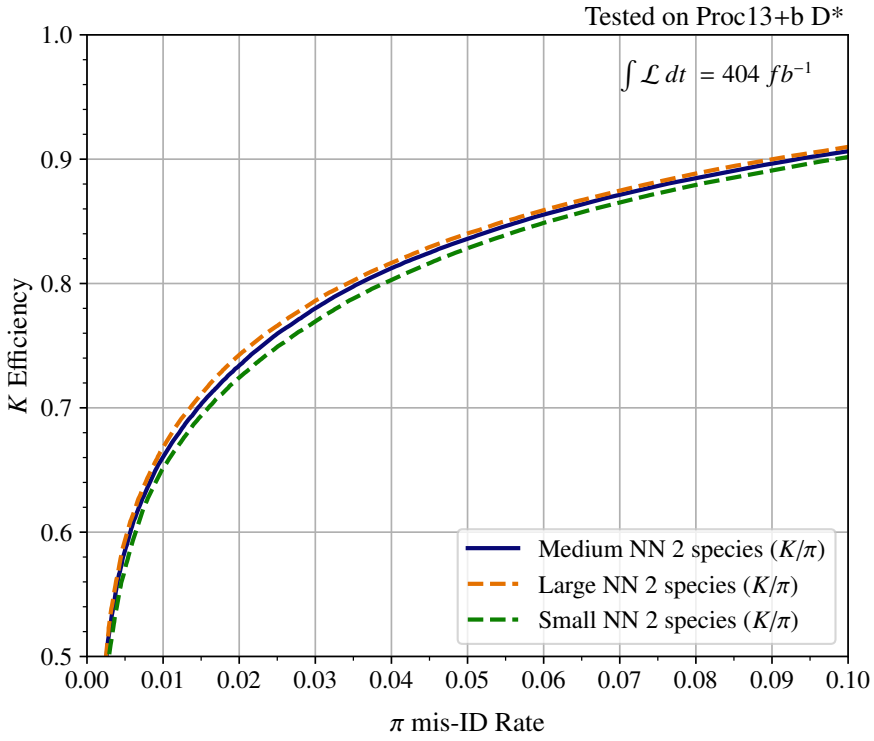


Figure 5.1: Particle-identification performance for K identification evaluated on the real-data sample from D^* decays (see section 4.1.2.1) for the three neural networks listed in table 5.2 trained on the particle-gun sample. The receiver operating characteristic (ROC) curve for kaon identification is shown, i.e. the efficiency to identify a kaon versus the rate with which pions are misidentified as kaons. Higher kaon efficiencies for the same pion misidentification rate mean better performance.

Table 5.1: Results of the hyperparameter optimization using Optuna ordered from highest to lowest AUROC validation.

AUROC validation	Number of layers	Nodes per layer	Slope α
0.962542	8	784	1
0.962419	6	944	1
0.962293	6	752	1
0.962268	2	624	1
0.962254	2	1008	1
0.962229	6	816	2
0.962203	4	560	3
0.962200	6	912	2
0.962197	6	848	2
0.962191	2	752	1
0.962171	4	464	3
0.962147	2	512	1
0.962135	2	624	1
0.962079	6	688	2
0.962069	2	464	1
0.962060	4	560	1
0.962050	6	816	1
0.961980	4	496	4
0.961952	6	816	1
0.961865	6	624	1
0.961815	4	208	1
0.961793	2	272	1
0.961647	4	272	1
0.961567	8	848	2
0.961558	8	848	2
0.961492	4	688	3
0.961380	2	176	1
0.960704	6	464	3
0.960371	2	80	1
0.959406	6	528	4
0.958971	8	784	3
0.954637	8	944	4
0.953955	6	176	4
0.952424	8	688	4
0.952421	8	80	2
0.951568	8	272	3
0.951318	8	528	4
0.940321	8	16	2
0.935178	8	16	2
0.500000	8	16	3

Table 5.2: Selected models from the hyperparameter optimization with Optuna.

Model	Number of layers	Nodes per layer	Slope α	Performance
Large Network	2	640	1	best
Medium Network	2	128	1	similar to best
Small Network	2	64	1	worst

Nevertheless, in following chapter, we are expanding the neural network for 2 species to identify all 6 species. Section 6.5 justifies the need to increase the complexity of the neural network to perform the new task. Since we will compare the performance of both networks, we want them to have the same architecture to have a fair comparison. To this end, instead of the mentioned network of 2 hidden layers of 128 nodes; we trained another neural network with 2 hidden layers of 512.

This network with 2 hidden layers of 512 will be used in the following and is called neural network for 2 species (coded as 001 in table 4.2).

5.2 Neural Networks 2 Species: Performance Evaluation

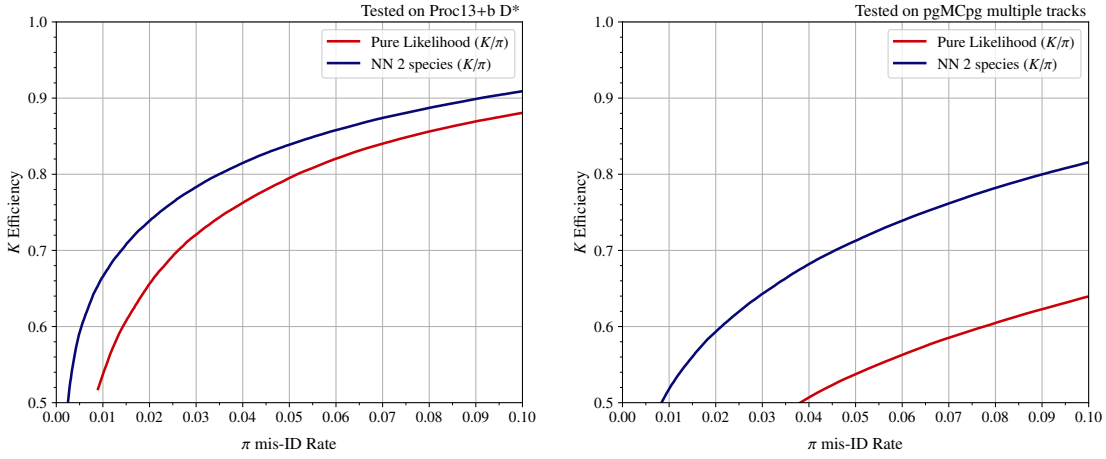
In this section, we analyze performance in K/π separation of the neural network for 2 species. As explained before, it is trained on simulated data. Figure 5.3b shows the performance on a real-data D^* sample. The neural network for 2 species performs better than the Pure Likelihood approach, up to 21.77 %.

Similarly, Fig. 5.3a illustrates the performance on pgMC, yielding identical results. Furthermore, the difference in performance of the neural network and Pure Likelihood approach is larger.

In summary, the neural network for 2 species outperforms the Pure Likelihood method, i.e standard method in the Belle II experiment, when applied to both simulated and real data; which was one of the main goals of this work. Detailed studies can be found in Wallner et al. [45], where for example, the real-data and simulated data agreement is shown.

5.3 Training on Real vs Simulation Data

So far, the neural network for 2 species was trained on simulated data (pgMC). Nevertheless, one should evaluate models trained with real data to compare the performance, as we can not take for granted than training on simulated data is better. To address this, we have at our disposal two identical neural networks: one trained on simulated data (depicted in blue) and the other on real data (depicted in black),



(a) (b)

Figure 5.2: Particle-identification performance for K identification for the neural network for 2 species (in blue) and the Pure likelihood approach (in red). (a) The performance is evaluated on the real-data sample of D^* decays (see section 4.1.2.1). (b) The performance is evaluated on a pgMC simulated sample (see section 4.1.1).

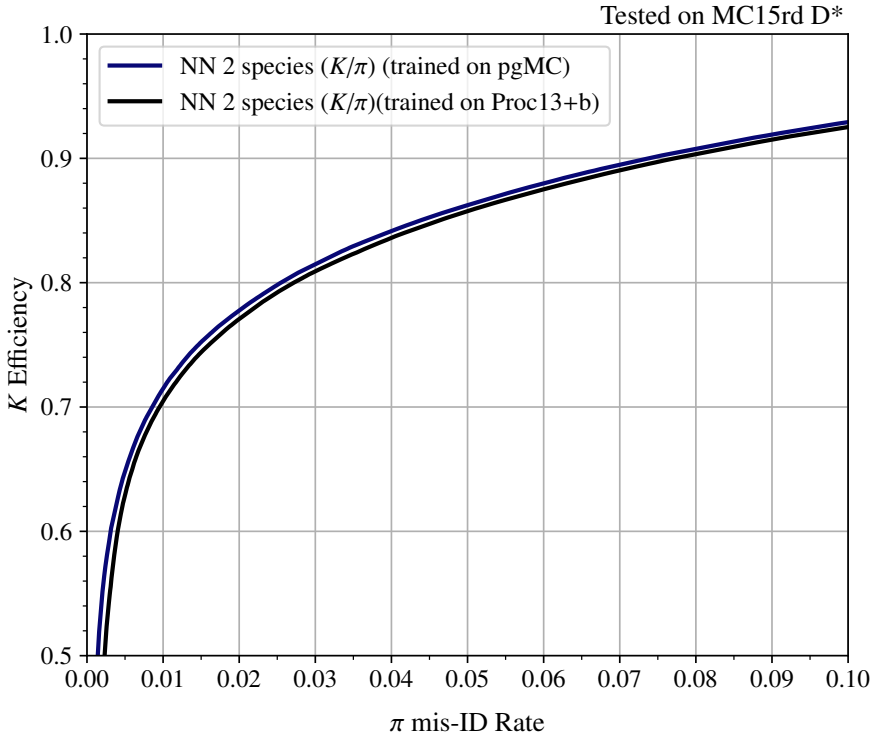
as illustrated in Fig. 5.3.

Figure 5.3a shows the performance tested on simulated data. Both neural networks exhibit similar levels of performance. Figure 5.3b shows the performance tested on real data. The neural network trained on real data shows better performance than the neural network trained in simulated data.

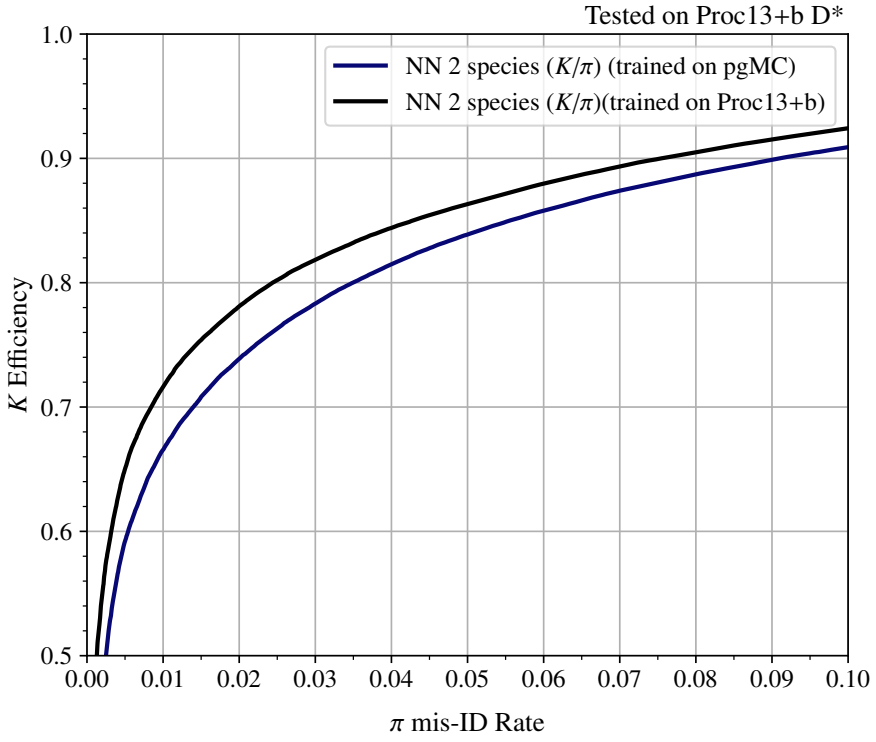
The clear observation is that neural networks performs equal or better when trained on real data than with simulated data. This may be attributed to the fact that training on real data overcomes imperfections in simulation, due to the fact that there are details in real data that are not reproduced simulation.

However, training on real D^* data has some significant drawbacks: it does not cover the full kinematic range, and may introduce potential biases due to the training sample. This problems are described and illustrated in detail in section 4.1.2.1. Similar challenges are encountered with other real samples. For example, if we want to reproduce everything for leptons, we have to use J/Ψ sample, which has the same problems, as explained in section 4.1.2.3. Overall, training on real data might work for specialized problems. In our case, we want to have a generalised neural network so the "small" gain in performance is not worth all the problems.

The ultimate goal of separating for all six species simultaneously requires samples from all six species. It becomes not feasible to obtain real-data samples for all particles without the mentioned problems. As a result, all the neural networks employed in the subsequent sections are trained with pgMC.



(a)



(b)

Figure 5.3: Particle-identification performance for K identification for the neural network for 2 species trained in pgMC (in blue), and for the neural network for 2 species trained on the real-data D^* (in black). (a) shows the performance evaluated on the simulated-data D^* (see section 4.1.3). (b) shows the performance evaluated on the real-data sample of D^* decays (see section 4.1.2.1).

5.4 Feature Importance

It is very challenging to directly understand the inner workings of a neural network by studying the trained neural network parameters. However, we can try to understand the behaviour of a neural network by measuring the feature importance. It is a techniques that quantifies the influence that an input variable has on the output. This is done by permutation of inputs. To this end, we systematically shuffle the values of individual input variables and observe the resulting impact on the network's performance, i.e we quantify the change in predictive accuracy compared to the original performance (without any permutation).

Figure 5.4 shows the feature importance of the neural network for 2 species. The charge and azimuthal angle are not important parameters for the neural network. We observe that the TOP, CDC and ARICH give the highest feature importance. This is an expected behaviour, due to the TOP is the most important detector for K/π separation, followed by the CDC. The ARICH plays also an important role[8]. One can note that among the log-likelihoods for the six species, the K and π log-likelihoods are the most important ones for most of the detectors, as expected. Furthermore, the other likelihoods have a non-zero importance.

Here, we only use kaon and pion tracks. Ideally, if the $\mathcal{L}^{\mathrm{D}}(h)$ were perfect, all the information necessary for K/π separation would be contained in the $\mathcal{L}^{\mathrm{D}}(K)$ and $\mathcal{L}^{\mathrm{D}}(\pi)$. Then, likelihoods for other species $\mathcal{L}^{\mathrm{D}}(h \neq K, \pi)$ would be not give additional information. Taking this into consideration, one possible explanation to the improvement in performance is that the $\mathcal{L}^{\mathrm{D}}(h)$ are imperfect and some addition information is stored in $\mathcal{L}^{\mathrm{D}}(h \neq K, \pi)$. The Pure Likelihood approach only uses $\mathcal{L}^{\mathrm{D}}(K)$ and $\mathcal{L}^{\mathrm{D}}(\pi)$, while the neural network alos uses information from the other $\mathcal{L}^{\mathrm{D}}(h \neq K, \pi)$, which provides extra information.

The imperfection in $\mathcal{L}^{\mathrm{D}}(h)$ gives a hint on why the neural network outperforms the Pure Likelihood (as shown in section 5.2). Another possible explanation is that the neural network finds correlations between the likelihoods and uses them to improve the performance. This would require an additional analysis.

[8]This is attributed to the cases where, the particle goes to the forward endcap and does not cross the TOP detector.

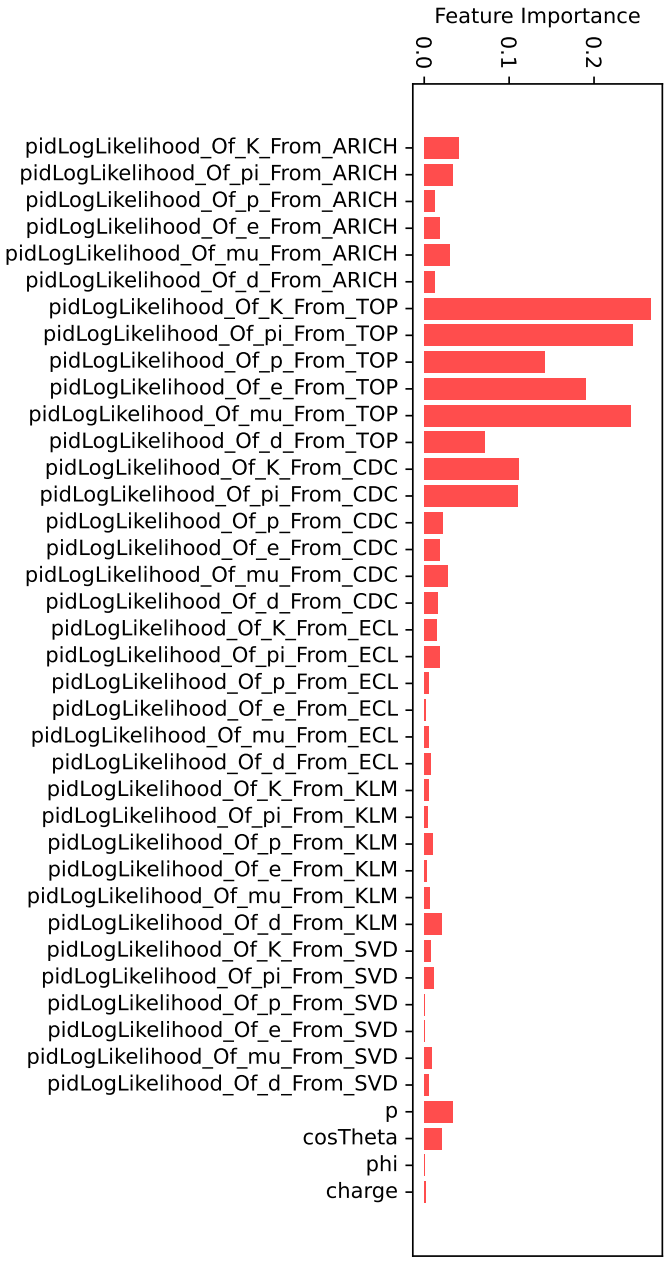


Figure 5.4: Shows the feature importance on the pgMC sample (see section 4.1.1) of the neural network for 2 species (2 layers of 512 nodes). In the vertical axis the inputs of the neural network are listed. In the horizontal axis, the importance of each variable is shown. A larger value means that this parameter is more important for the PID performance. A low bar indicates that the parameters are not important for the neural network

Chapter 6

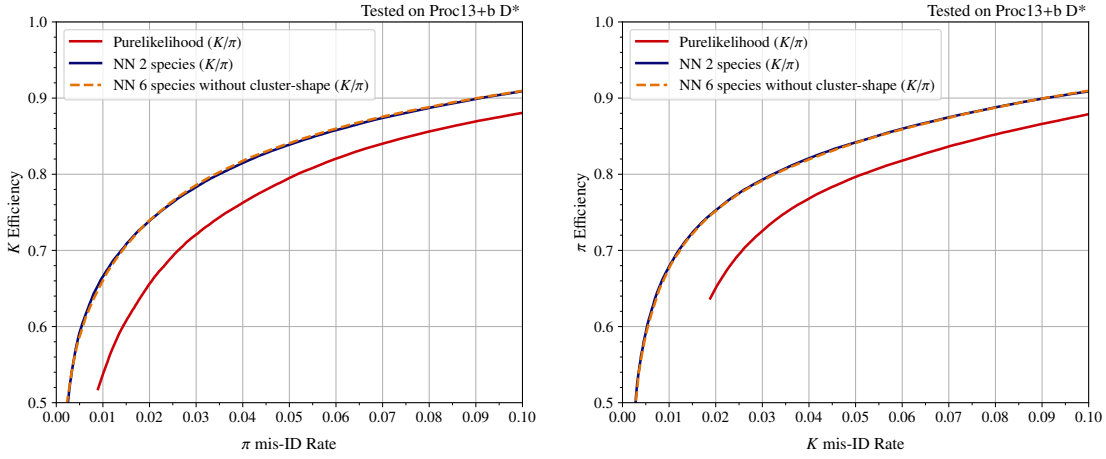
Neural Network for Six Species: Binary Classification

In this chapter, we test the performance of the neural network for 6 species with different data sets to asses whether it is a good alternative for PID. The first step is to compare the performance of both of the neural networks, for 2 and 6 species; to asses if there is any decrease in performance by incorporating more species. This is done in section 6.1. Next, in section 6.2 we test the method specifically used for lepton PID against the neural network. Additionally, it shows the importance of slightly modifying our neural network to cope with lepton PID, which is a more complicated task. Furthermore, we present performance tests on real data in section 6.3. In section 6.4, we will analyze the dependence of the neural network performance on kinematics ranges and bins. Finally, in section 6.5 we discuss the optimal architecture for the neural network for 6 species.

The three networks, detailed in section 4.3.2, are used in the following. They will be used using different normalisation. To give the reader an easier understanding of the classification variables obtained after the normalization, tables are given in appendix A.

6.1 Comparing Neural Network Performance: Two Species vs Six Species Prediction

In chapter 5 we discussed the performance of the neural network with 2 species. Now, we aim to test the neural network for 6 species for the same particles. To assess whether there is any decrease in the performance for K/π separation when increasing complexity of the task, i.e when training a network to distinguish among all 6 species, we measure the performance for K/π separation of the neural network for 6 species without cluster-shape and compare it with the performance form neural network for 2 species. To ensure a fair comparison, we apply the binary normalisation procedure explained in section 4.4 to the neural network for 6 species without cluster-shape. In other words, we place both sets of predictions on a common scale, enabling a fair evaluation of their performance. For instance, when performing K/π



(a) (b)

Figure 6.1: (a) K/π separation performance for the neural network for 6 species without cluster-shape (binary normalitzation) in orange, the neural network for 2 species in blue and the Pure Likelihood approach in red. The ROC curve for (a) kaon identification and (b) π misidentification are shown. The performance is evaluated on the real-data sample from D^* decays (see section 4.1.2.1). Higher efficiencies for the same misidentification rate mean better performance.

separation, Eq. (4.6) is used, where α is replaced by a kaon and β is replaced by a pion i.e $C_{\mathrm{NN}}(K\ :\ \pi)$.

Figure 6.1a show the ROC curve for K efficiency vs π misidentification rate for the real-data D^* sample. We use the neural network for 6 species without cluster-shape (binary normalitzation) (coded 002 in table 4.2) to have a fair comparison, since it has the same inputs as the neural network for 2 species (coded 001 in table 4.2). Both neural networks (blue and orange curves) exhibit equal performance. Furthermore, Fig. 6.1b shows the same for π efficiency against K misidentification rate, where the same agreement is observed.

In addition, similar tests have been conducted on other data samples, yielding consistent results. From this we conclude that the neural networks can be trained for multi-class classification without compromising their performance for binary classification.

6.2 Extension of the Neural Network for Lepton Identification

The ultimate goal of this work is to propose a novel method that can be used for both hadron and lepton identification. In previous section 6.1, the neural network for 6 species without cluster-shape has already proven to outperform the Pure Likelihood approach for hadron PID.

As explained in section 3.5, a boosted decision tree (BDT) is used at Belle II to improve lepton PID. To include lepton PID, an extended version of the neural network is proposed, which uses the same strategy as the BDT. This extended neural network, combines the inputs of the neural network for 6 species without cluster-shape, i.e Inputs Set 1 in table 4.1, but adds the ECL observables from the BDT, i.e Inputs Set 2 in table 4.1, to optimize for lepton PID. This extended neural network is called neural network for 6 species with cluster-shape, coded 003 in table 4.2.

In this chapter, despite not being explicitly written, the binary normalitzation is always for used in the neural networks, for the particles we are comparing.

6.2.1 Influence of ECL Cluster-Shape Variables as Inputs

The first step is to compare the neural network for 6 species without cluster-shape against the neural network for 6 species with cluster-shape, aiming to analyze the impact of the additional ECL cluster-shape variables as inputs. For this purpose, tests have been conducted with two different types of data samples.

Figure 6.2 shows the performance of the neural network for 6 species without cluster-shape (orange) and the neural network for 6 species with cluster-shape (purple), evaluated on a real data sample of D^* (see section 4.1.2.1). The performance for both neural networks is very similar. The cluster-shape variant slightly outperforms the other neural network. This illustrates that, for hadrons, including cluster-shape variables does not cause any detriment in performance.

Tests have also been performed to assess the effect for lepton identification on pgMC samples (section 4.1.1). Figure 6.3a shows the performance for μ/π separation. Both neural networks, with and without cluster-shape variables, exhibit superior performance compared to the BDT (depicted in green). Additionally, the two neural networks perform similarly, with neural network with the cluster-shape exhibiting a slightly better performance.

When evaluating the performance in e/π separation (Fig. 6.3b) the neural network for 6 species with cluster-shape performs better than the BDT. Consequently, with the neural network with cluster-shape we can improve electron identification. However, the BDT outperforms the neural network for 6 species without cluster-shape. Consequently, this demonstrates the necessity of incorporating the cluster-shape variables as inputs in the neural network, showing that the ECL cluster-shape variables introduced as inputs enable highly effective electron discrimination.

The neural network for 6 species with cluster-shape has been identified as the optimal approach, both for lepton PID but also for hadron PID, improving the performance compared to the network for 6 species without cluster-shape. Furthermore, the neural network for 6 species with cluster-shape achieves a better performance than the BDT in all considered cases. Consequently, this neural network will be

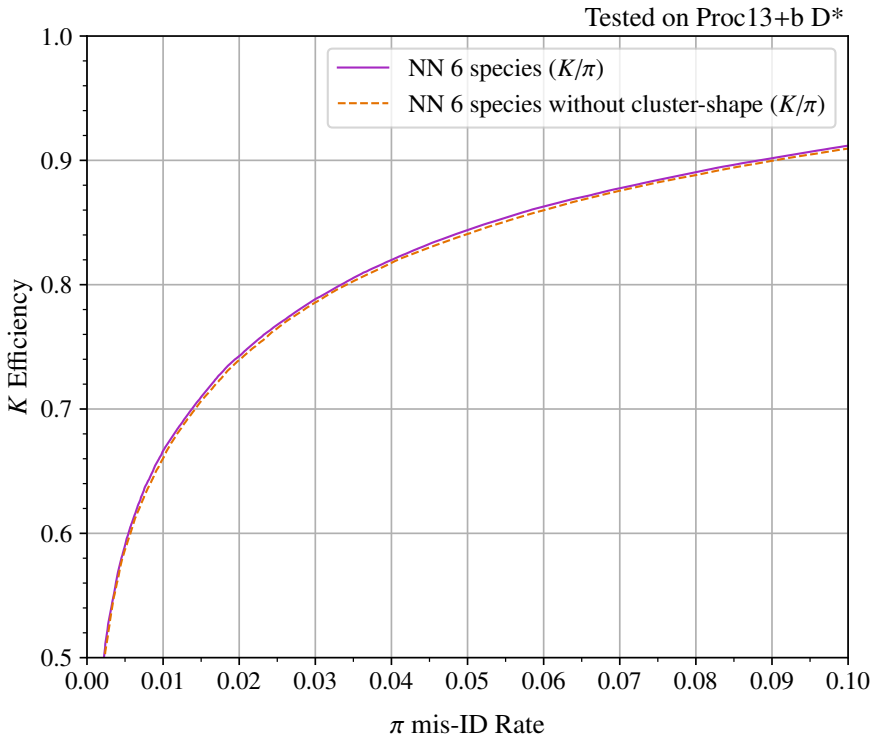
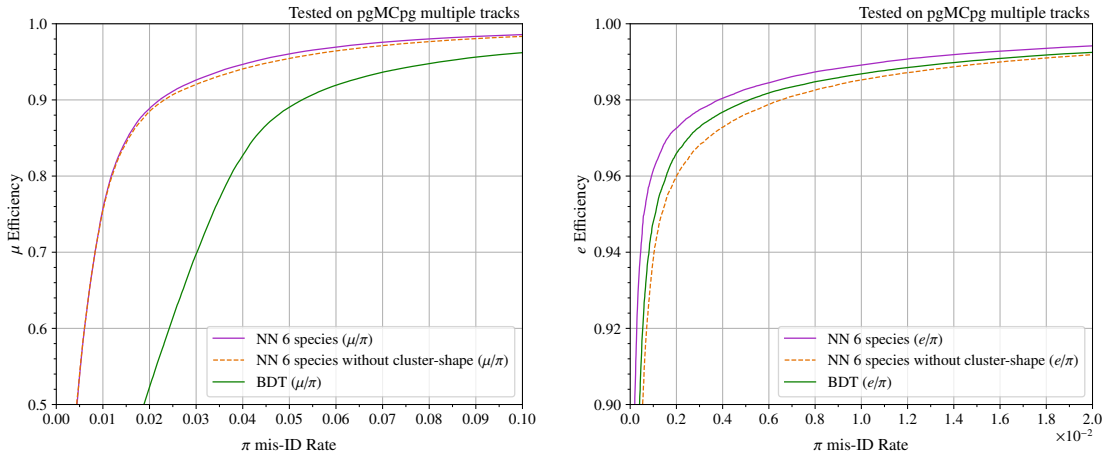


Figure 6.2: Performance tested on real-data D^* sample (see section 4.1.2.1) for K/π separation of the neural network for 6 species with cluster-shape (binary normalitzation) (purple) and for the neural network for 6 species without cluster-shape (binary normalitzation) (orange).



(a) (b)

Figure 6.3: Performance tested on the pgMC sample (see section 4.1.1) of the neural network for 6 species with cluster-shape (binary normalitzation) (purple), for the neural network for 6 species without cluster-shape (binary normalitzation) (orange) and for the BDT (green). (a) Shows μ/π separation and (b) shows e/π separation but with a zoomed π misidentification rate.

used until the end of the work.

6.2.2 Performance Evaluation on pgMC Lepton Samples

Having defined the optimal neural network, the neural network with cluster-shape, we aim to test it on other samples. We compare the new extended neural network with the BDT and the Pure Likelihood approach on a pgMC sample (section 4.1.1) of electrons, muons, and pions. Figure 6.4 shows, for e/μ, e/π and μ/π combinations, the neural network for 6 species with cluster-shape (purple), the BDT (green), and the Pure Likelihood approach(red).

Initially one can preform a fully lepton separation, e/μ separation, which is shown in Figs. 6.4a and 6.4b. The BDT performs better than the Pure Likelihood approach. Hence, the BDT proposed for leptons is accomplishing its purpose, helping to improve the lepton PID. Furthermore, the neural network with cluster-shape performs slightly better than BDT.

Figure 6.4c shows the performance for e/π separation. The neural network and the BDT exhibit similar performance. Moreover, both methods outperform the Pure Likelihood approach significantly.

In contrast, the performance for μ/π separation (see Fig. 6.4d), exhibit different feature. The BDT performs similar or even worse than Pure Likelihood approach. This can be attributed to the fact that the BDT was designed for electron identification, but as one consider other particles, its performance decreases. The neural network gives clearly the best performance among all methods considered.

Overall, the neural network with cluster-shape exhibits performance levels at least on par with, if not superior to the BDT. Therefore, one can conclude that the neural network with cluster-shape has overall the best performance (compared to the BDT and Pure Likelihood approach) when tested on pgMC samples from electrons, muons and pions. Furthermore, the versatility of the neural network is evident as it can be applied for both hadron and lepton PID, whereas the BDT is primarily designed for lepton PID, in particular for electrons. The last statement will become evident once we deal only with hadron separation i.e K/π (section 6.3).

6.3 Real-Data Performance Evaluation

The neural network with cluster-shape has already demonstrated high efficiency in lepton PID in the pgMC sample (sections 6.2.1 and 6.2.2). However, one of the main goals of the work is to have a universal neural network that can work both on simulated and real-data sample, to use in with experimental data from the Belle II experiment. Thus, final tests are required to validate its performance on real data. The results are presented in Fig. 6.5 where, the neural network is depicted in purple,

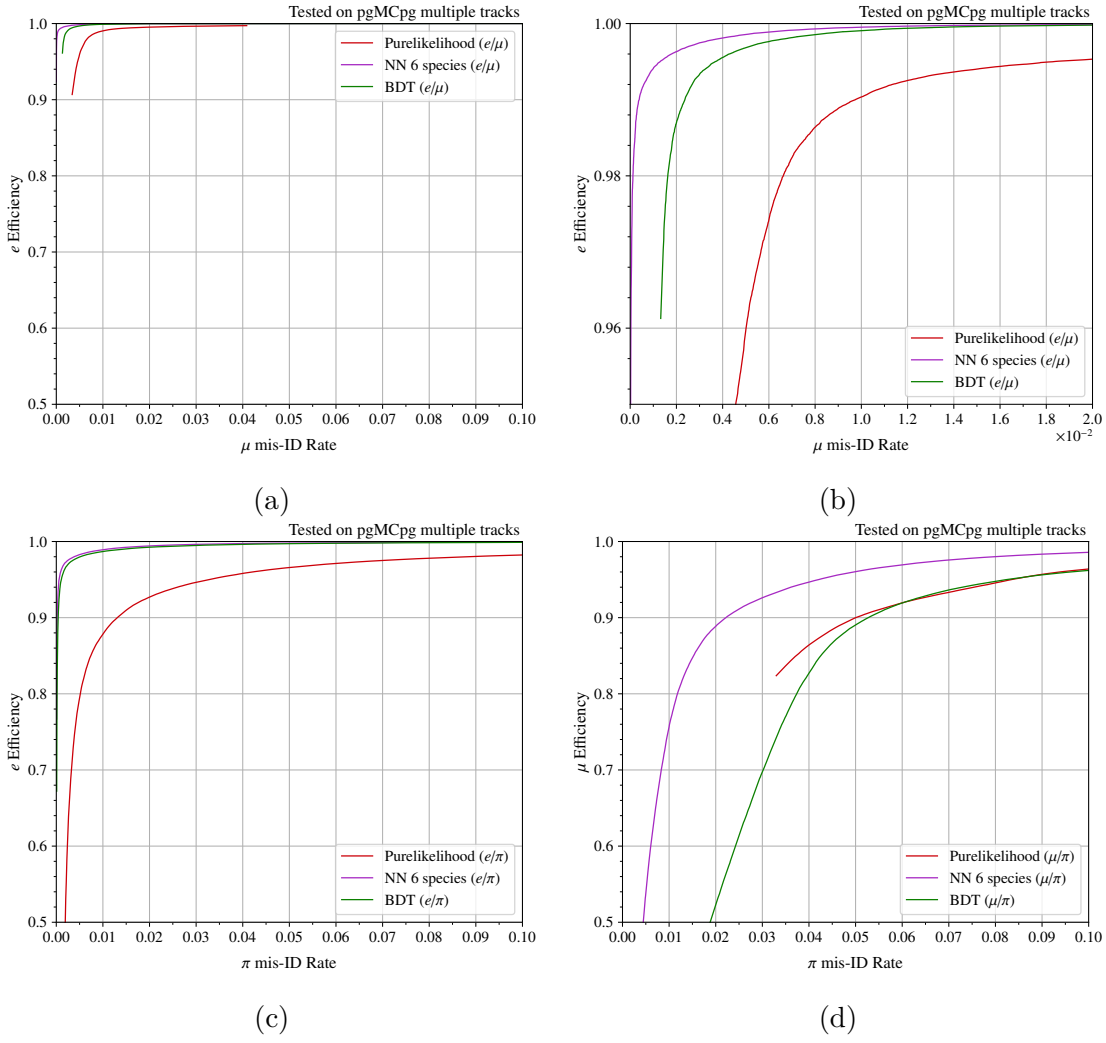


Figure 6.4: Performance for lepton identification tested on the pgMC sample (see section 4.1.1) for the neural network for 6 species with cluster-shape (binary normalitzation) in purple, the BDT in green and the Pure Likelihood approach in red. (a) shows e efficiency against μ misidentification rate, (b) provides the same plot as (a) but with a zoomed μ misidentification rate, (c) shows e efficiency against π misidentification rate and (d) shows μ efficiency against π misidentification rate.

the BDT in green and the Pure Likelihood approach in red.

First, we focus on hadron separation. Figure 6.5a shows the performance on a D^* sample (section 4.1.2.1), and therefore performing K/π separation. The BDT gives the worst performance. This demonstrates the limited capability of the BDT for hadron separation. The neural network outperforms all methods. Figure 6.5b shows the p/π separation performance tested on the real-data Λ_0 sample. Once again, the BDT performs the worst in a hadron sample. The neural network exhibits the best performance of all methods, improving the performance considerably.

Second, additional tests in hadron/lepton separation are performed. The J/Ψ decays sample (section 4.1.2.3) provides the e and μ particles while the D^* sample provides the π particles. However, as explained in section 4.1.2.3, using real data samples of J/Ψ decays is not feasible for $|\vec{p}| < 1.5$ GeV/c. Furthermore, an upper limit of $|\vec{p}| < 4.5$ is applied to ensure a similar number of hadrons and leptons for a fair comparison. Figure 6.5c, shows the e/π separation performance tested on real-data sample. As for the simulated sample, the BDT gives a better performance than the Pure Likelihood approach. Additionally, the performance of the neural network is practically identical to that of the BDT. Figure 6.5d shows μ/π separation performance. The BDT performs similarly to the Pure Likelihood approach. The neural network gives the best performance.

Overall, as for the simulated sample, the BDT is efficient for e identification in real samples. However, the good BDT performance is limited to e identification. In contrast, the neural network has proven to perform as well as the BDT for e, and significantly better for other species.

The results presented in this section, combined with section 6.2.2, demonstrate a crucial characteristic of the neural network, which we refer to as "universality". The neural network for 6 species with cluster-shape has proven to systematically outperform the Pure Likelihood approach for all combinations of hadrons, leptons and hadron/leptons. Furthermore, it exhibits superior performance compared to the BDT in all cases examined, with the exception of e separation, where the performance is comparable but not worse. This observation holds for both real data samples and pgMC.

Consequently the neural network has proven to be an exceptionally effective method for PID. So far, specific tools, like the BDT, were developed for specific separation tasks. However, the neural network allows to make binary separation with any combination. Furthermore, the neural network possesses additional capabilities. Section 7.3 shows it can also be employed for multi-class classification.

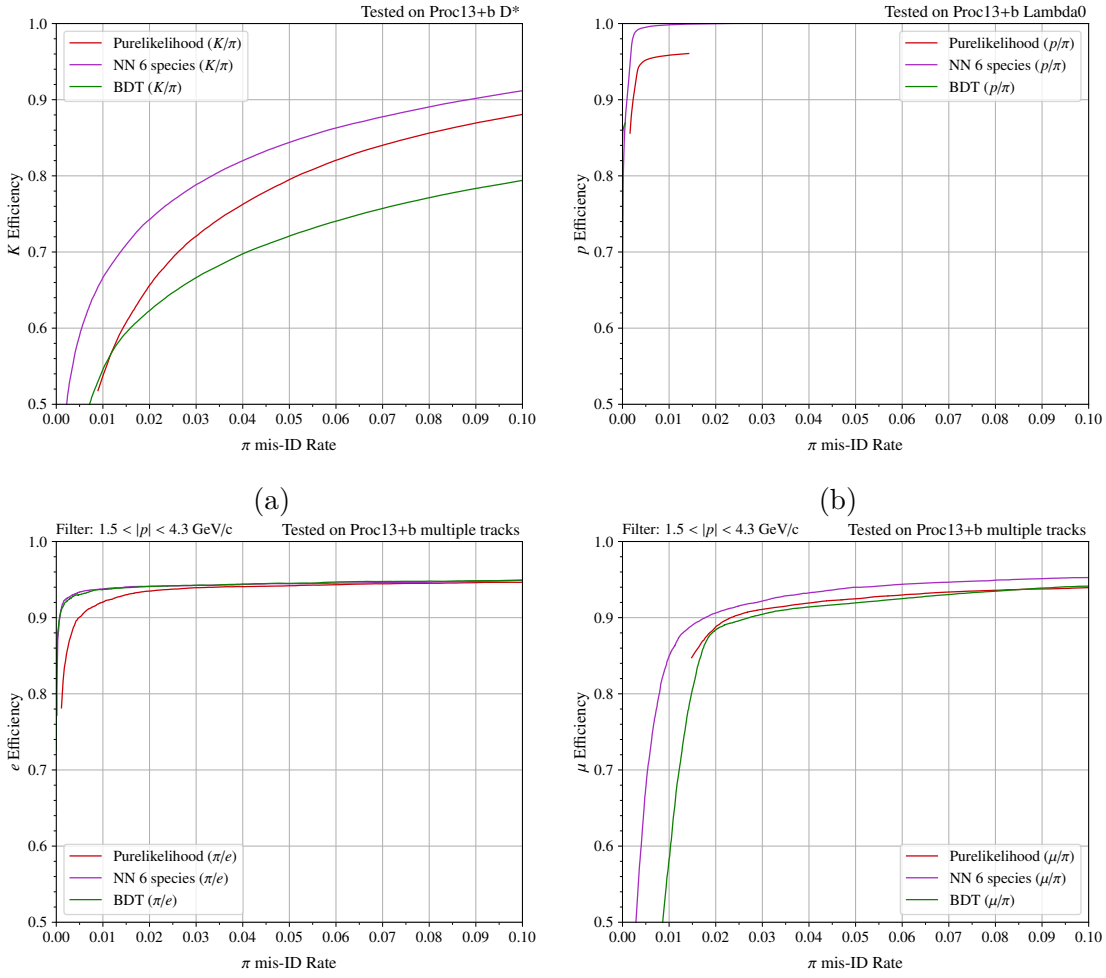


Figure 6.5: Performance for the neural network for 6 species with cluster-shape (binary normalitzation) (purple), the BDT (green) and the Pure Likelihood approach (red); tested on (a) the real-data D^* sample (see section 4.1.2.1) for K/π separation, (b) the real-data Λ_0 sample see section 4.1.2.2) for p/π separation, (c) the real-data D^* and J/Ψ sample (see sections 4.1.2.1 and 4.1.2.3) for e/π separation, and the real-data D^* and J/Ψ sample (see sections 4.1.2.1 and 4.1.2.3) for μ/π separation.

6.4 Performance Dependence on Kinematics

6.4.1 Performance in Kinematic Ranges

The PID performance, discussed in thesections 6.2.1, 6.2.2 and 6.3, represents an average performance across the kinematic distribution of the sample used for testing. In order to study the performance in kinematic regions, we show in the following the PID performance as a function of the track momentum and the track $\cos\theta$. To this end, Fig. 6.6 presents the ROC curves in three momentum regions and in three regions of $\cos\theta$, tested on the real-data D^* sample.

First, the aim is not to directly compare the two methods, but rather to analyse the regions where the PID achieves superior performance. This is crucial because the PID performance varies depending on the region. In the low momentum region ($|\vec{p}| < 0.7\,\mathrm{GeV/c}$)(see to Fig. 6.6a), both methods exhibit remarkably high performance. In contrast, in the the backward region ($\cos\theta < -0.625$) (see Fig. 6.6d), both methods give relatively low performance. The low performance is caused by the fact that in the backward region the TOP and ARICH, which are detectors very important for K/π separation, are not present as shown in table 3.1. In the remaining regions (Figs. 6.6b to 6.6d and 6.6f), an intermediate level of performance is observed.

Second, we analyse the performance disparity between both methods. In the low momentum region (refer to Fig. 6.6a) and high momentum region ($|\vec{p}| > 2.0\,\mathrm{GeV/c}$) (see Fig. 6.6c), the neural networks clearly outperform the Pure Likelihood approach. In the intermediate momentum region ($0.7 < |\vec{p}| < 2.0\,\mathrm{GeV/c}$), both methods exhibit a similar PID performance. In the backward region and the barrel region ($-0.625 < \cos\theta < 0.846$), shown in Figs. 6.6d and 6.6e, respectively, the neural network outperforms the Pure Likelihood approach. However, in the forward region ($\cos\theta > 0.846$), shown in Fig. 6.6f, the neural network performs similarly to the Pure Likelihood approach. Only for low misidentification rates, the neural network slightly outperforms the Pure Likelihood approach.

In summary, the neural network performs equally well or even better than the Pure Likelihood approach in all regions. There is no region where the performance of the Pure Likelihood approach surpasses that of the neural network. Consequently, our neural network can be applied across all momentum and angular ranges without compromising performance.

6.4.2 Performance in Kinematic Bins

To conduct a more detailed study of the momentum and $\cos\theta$ dependency in PID performance, it is necessary to assess the PID performance within specific $(|\vec{p}|, \cos\theta)$ bins. To this end, a threshold for the classification variable of the desired hypothesis must be determined, as explained in section 3.4. This threshold is not universal, i.e. the same threshold may have a different meaning for two different PID methods.

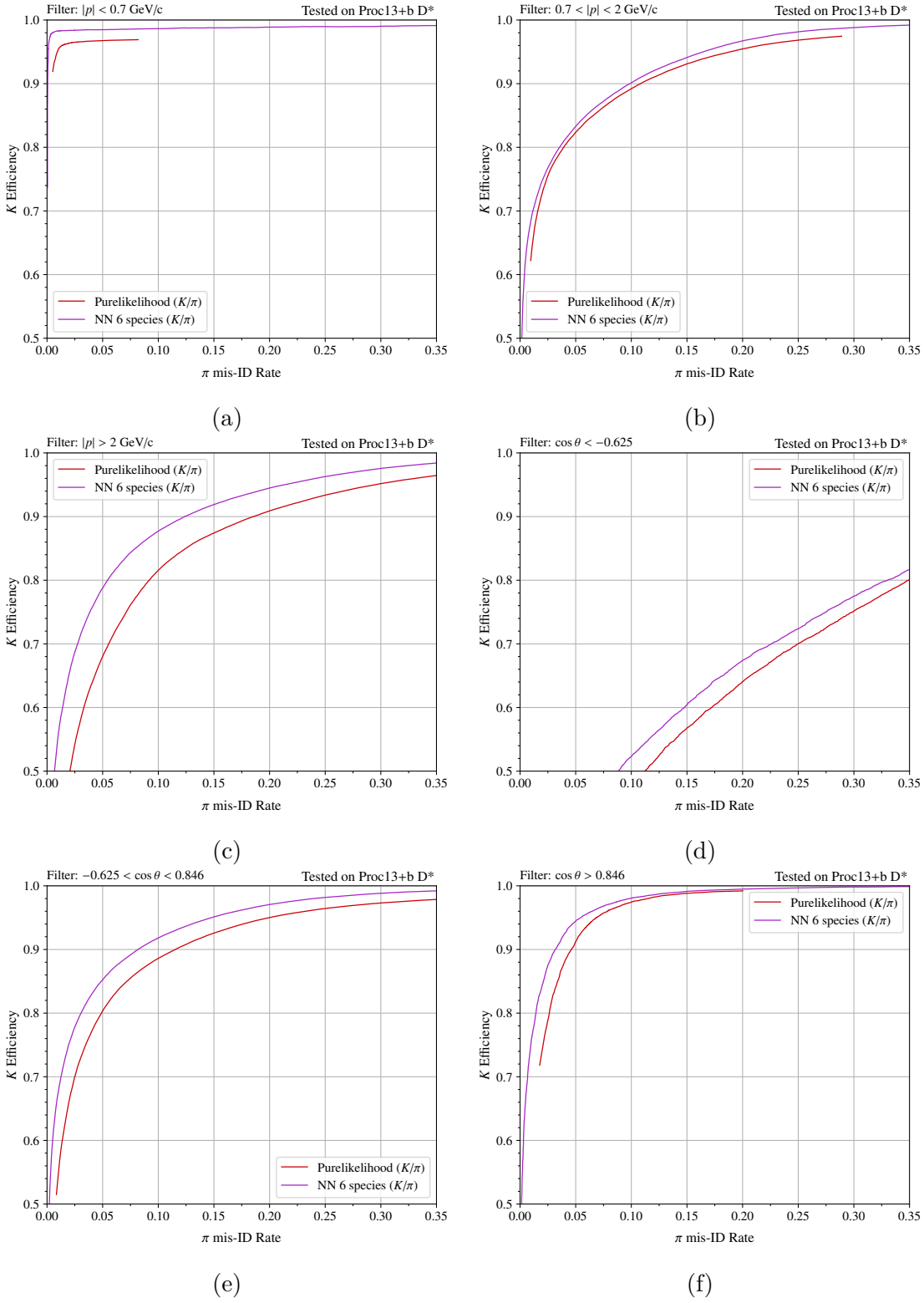


Figure 6.6: ROC curve for K/π separation, evaluated on the real-data D^* sample (see section 4.1.2.1) for the neural network for 6 species with cluster-shape (binary normalitzation) (purple) and the Pure Likelihood approach (red); in six kinematic regions: (a) the low momentum region ($|\vec{p}| < 0.7\,\mathrm{GeV/c}$), (b) the intermediate momentum region ($0.7 < |\vec{p}| < 2.0\,\mathrm{GeV/c}$), (c) the high momentum region ($|\vec{p}| > 2.0\,\mathrm{GeV/c}$), (d) the backward region $\cos\theta < -0.625$, (e) the barrel region $-0.625 < \cos\theta < 0.846$, and (f) the forward region $\cos\theta > 0.846$.

Consequently, we individually selected a threshold for each PID method to ensure that the misidentification rate for π is the same for both methods. This approach ensures a fair comparison of the efficiencies of the tested methods.

For a target average π misidentification rate of 2 %, this results in a threshold of 0.9887977595519104 for the Pure Likelihood approach and a threshold of 0.7469493898779755 for the neural network PID method. In Fig. 6.7 (top left), the K efficiency on real data is displayed as a function of track momentum. Below approximately 1 GeV/c, the neural network slightly outperforms the Pure Likelihood. In the range $1 \lesssim |\vec{p}| \lesssim 1.5$ GeV/c, both methods yield a similar K efficiencies. Beyond about 1.5 GeV/c, the neural network surpasses clearly the Pure Likelihood approach. In this region, the performance is better by about a factor of about 1.5. In Fig. 6.7 (top right), the π misidentification rate is shown, where the neural network exhibits a similar momentum dependence as the Pure Likelihood approach. A minor difference can be found in two small peaks around in 1 GeV/c and around 3 GeV/c, where the neural network show a slightly higher misidentification rate compared to the Pure Likelihood approach.

In Fig. 6.7 (bottom left), the K efficiency on real data is presented as a function of $\cos\theta$. The neural network displays a weaker dependency on $\cos\theta$ while maintaining an overall higher efficiency, particularly for $\cos\theta > 0$. However, in this range, the π misidentification rate is also higher for the neural network (bottom right plot). Further, in the forward region ($\cos\theta > 0.84$) the K efficiency for the neural network is lower. However, this does not indicate a worse performance of the neural network since the misidentification rate of the Pure Likelihood approach also increases in this region. Further, there is a clear decay in performance in both methods for $\cos\theta < -0.5$, due to the explanation given above for the backward region.

It is interesting to point out that the Pure Likelihood approach exhibits a large spike of the π misidentification rate at $\cos\theta \approx -0.5$. We avoid this spike in the neural network by disregarding the TOP information in the region $-0.55 < \cos\theta < -0.50$ (as seen in section 4.3.3) without losing performance[9] in this region for the neural network.

Additionally, in Fig. 6.8, the same performance measures are presented for more relaxed thresholds. The threshold is set at 0.4370874174834967 for the Pure Likelihood approach and 0.47729545909181836 for the neural network, which both correspond to a π misidentification rate of 10.6 %.

The neural networks has a higher K efficiency (displayed in top left plot) in all the momentum range, but for a small region ($1 \lesssim |\vec{p}| \lesssim 1.5$ GeV/c), where the the efficiency is similar to the Pure Likelihood approach. On the top right, the π misidentification rate is again depicted. The neural network exhibits lower misiden-

[9]The only bin in this region where the Pure Likelihood approach has higher K efficiency is at $\cos\theta \approx -0.5$. However, the π misidentification rate explodes in this bin for the Pure Likelihood approach.

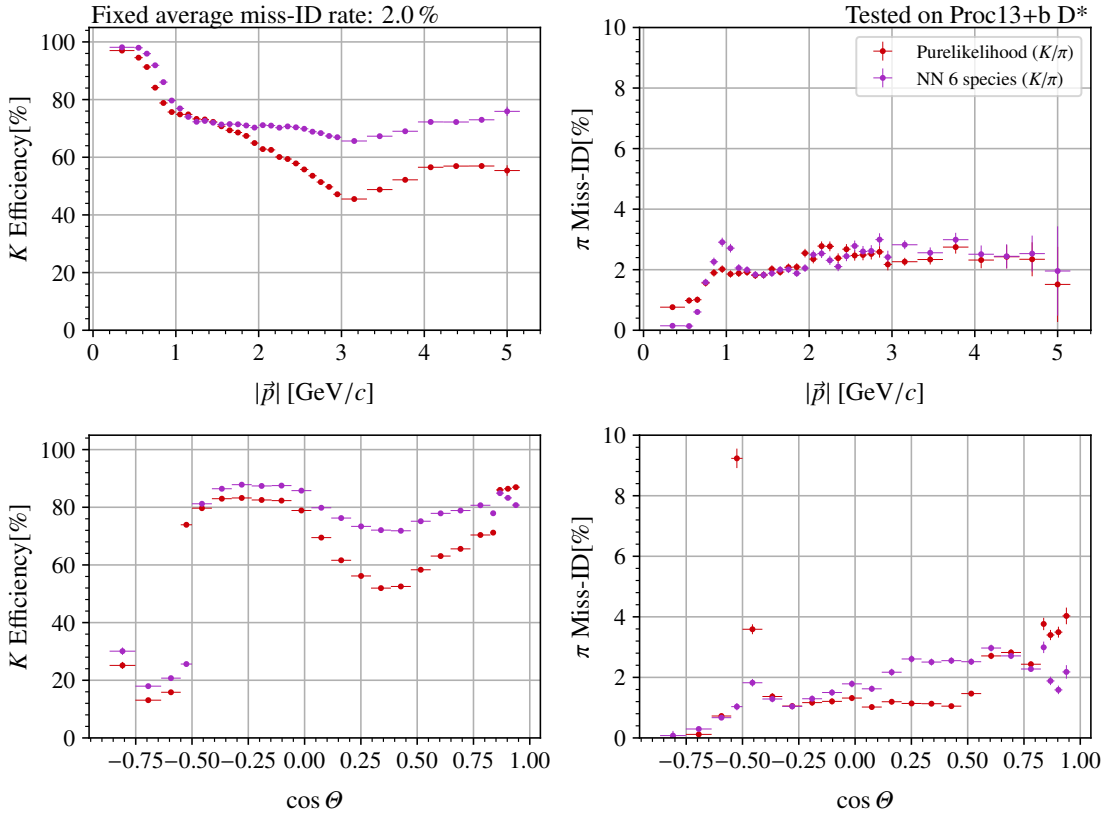


Figure 6.7: Performance for K/π separation for a target average π misidentification rate of 2 %, on the real-data D^* sample as a function of the track momentum (top row) and as a function of $\cos\theta$ (bottom row). The left column shows the K efficiency. The right column shows the π misidentification rate. The red data points represent the performance of the Pure Likelihood approach. The purple data points represent the performance of the neural network for 6 species with cluster-shape (binary normalitzation).

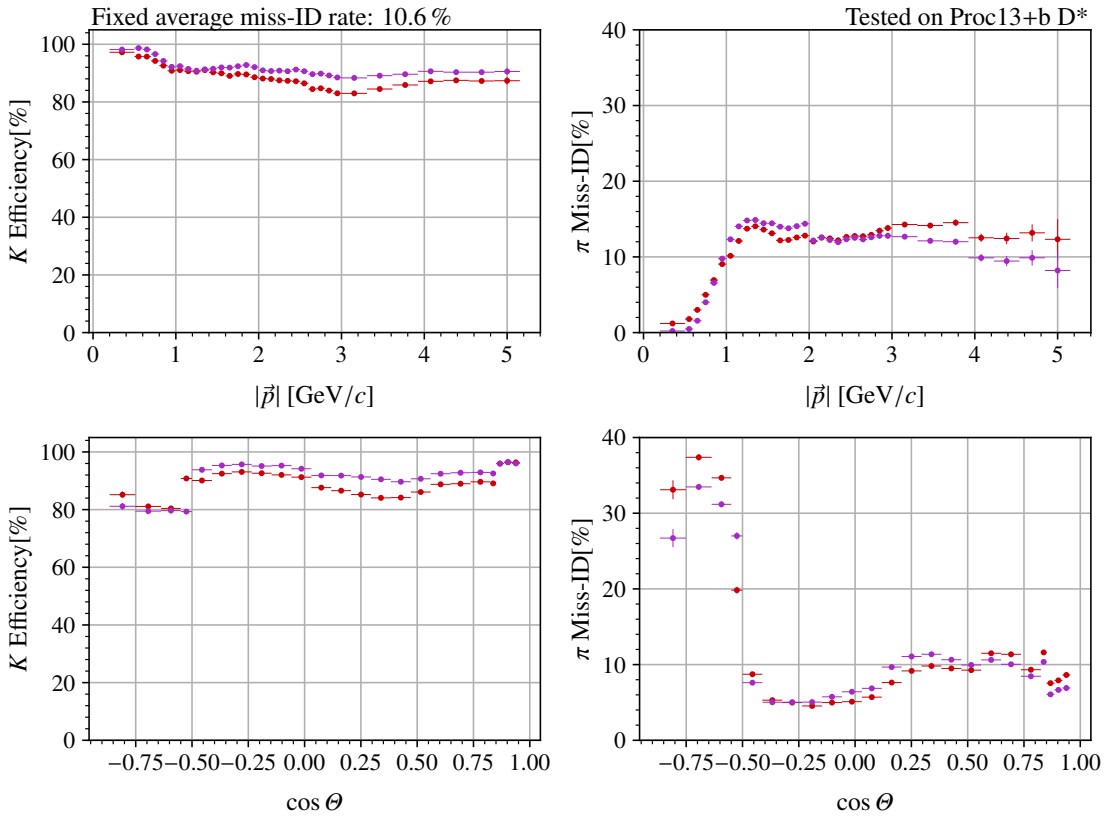


Figure 6.8: Same as Fig. 6.7, but for a target average π misidentification rate of 10.6 %.

tification rates in the ranges of $0 \lesssim |\vec{p}| \lesssim 1$ GeV/c and $|\vec{p}| \gtrsim 2.3$ GeV/c.

The neural network exhibits a higher K efficiency (bottom left plot) in the range $\cos\theta > -0.5$. In this same range, the misidentification rates for both methods are comparable. Specifically, for $-0.5 < \cos\theta < 0.5$, the neural network has a slightly higher misidentification rate, whereas for $\cos\theta > 0.5$, the Pure Likelihood approach has a higher misidentification rate. For the range $\cos\theta < -0.5$, the neural network shows a similar or slightly lower K efficiency (bottom left plot). However, it yields a much lower misidentification rate compared to the Pure Likelihood approach. Hence, one can conclude that in this region the neural networks still outperforms the Pure Likelihood approach.

Overall, when considering the information from both Fig. 6.7 and Fig. 6.8, it aligns with the findings observed in Fig. 6.6: since there is no region where the neural network performs worse than the Pure Likelihood approach, the neural network can be applied across all momentum and angular ranges. Nevertheless, one should take into account that there may not be a significant improvement when applied in intermediate momentum range or forward angular region.

For completeness we also studied the dependence on the azimuthal angle (ϕ), as shown in Fig. 6.9. There is not a strong dependence of the K efficiency (left) or in the π misidentification rate (right) on ϕ. The neural network has higher efficiencies for all ϕ bins than the Pure Likelihood approach. The neural network exhibits a similar ϕ dependence as the Pure Likelihood approach in the misidentification rate.

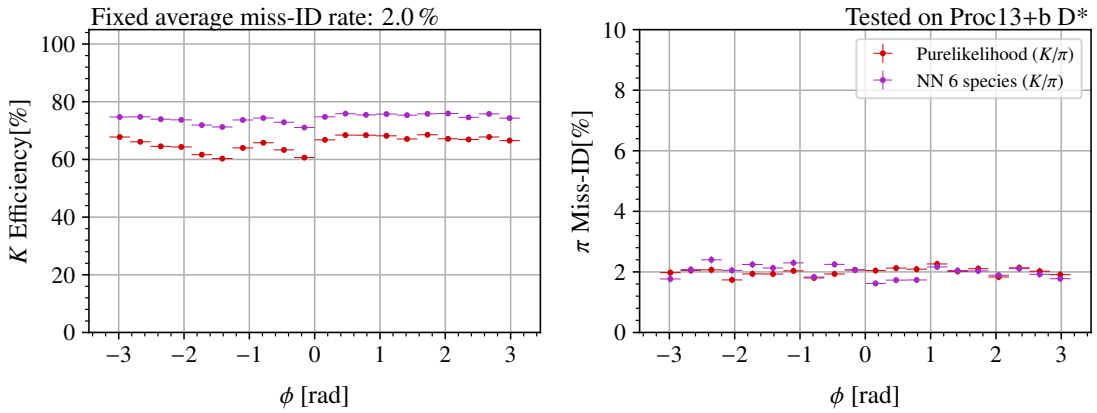


Figure 6.9: Performance for K/π separation for a target average π misidentification rate of 2 %, on the real-data D^* sample as a function of ϕ. The left column shows the K efficiency. The right column shows the π misidentification rate. The red data points represent the performance of the Pure Likelihood approach. The purple data points represent the performance of the neural network for 6 species with clustershape (binary normalitzation).

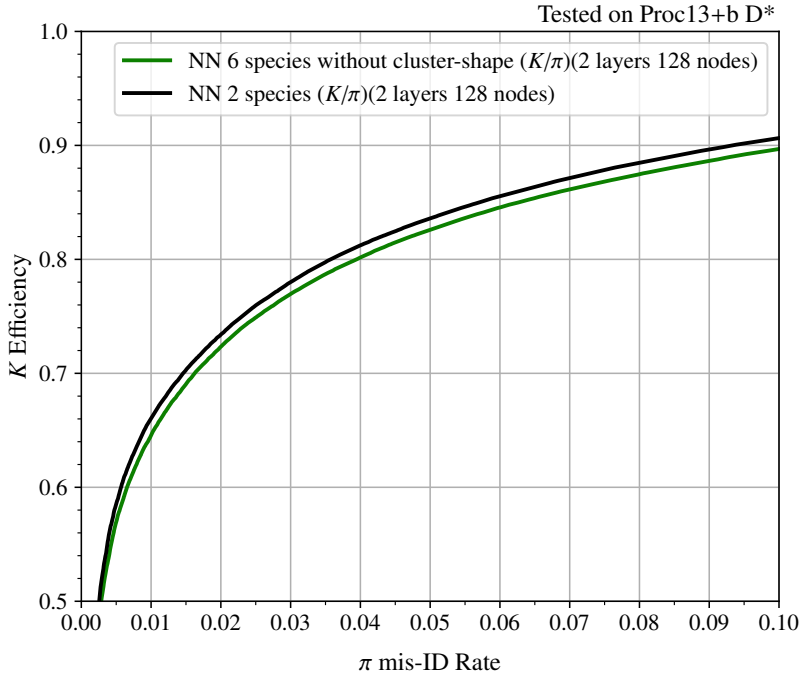


Figure 6.10: Performance tested on real-data D^* sample (see section 4.1.2.1) for K/π separation of the neural network for 2 species with 2 hidden layers of 128 nodes (black) and of the neural network for 6 species (binary) with 2 hidden layers of 128 nodes (green).

In summary, ϕ does not play a significant role for the PID performance, and hence for comparing PID methods. The weak ϕ dependence can be explained by the fact that the PID detectors distribution are designed symmetric in ϕ.

6.5 Architecture of Neural Network for Six Species

Section 5.1 shown that for a neural network for 2 species, it is enough to use an architecture with 2 hidden layers of 128 nodes. However, in this chapter, the two neural networks for 6 species, which are the ones detailed in table 4.2, have 2 hidden layers of 512 nodes. The reason for this is that an increased complexity of the task usually requires to increment the nodes of the neural network.

Figure 6.10 shows that a neural network for 2 species with 2 hidden layers of 128 nodes) has better performance than a neural network for 6 species without cluster-shape (binary) with 2 hidden layers of 128 nodes. This concludes that 2 hidden layers of 128 nodes are not enough to cope with the extension to separate all six species.

To find the optimal architecture for the six species separation, we always use 2 hidden layers, and we incrementally increase the number of nodes in the hidden layers until signs of overfitting are observed. Figure 6.11 shows validation loss func-

tion for the number of nodes set to 1000. It is very noisy it is especially that the loss of the validation sample increases at some point. This is a clear indication of overfitting. Slight indication of overfitting were also found around 800 nodes.

Consequently, to avoid overfitting we use 2 hidden layers of 512 nodes in the neural networks for 6 species, both with and without cluster-shape. Figure 4.16b shown that we do not have overfitting with this architecture.

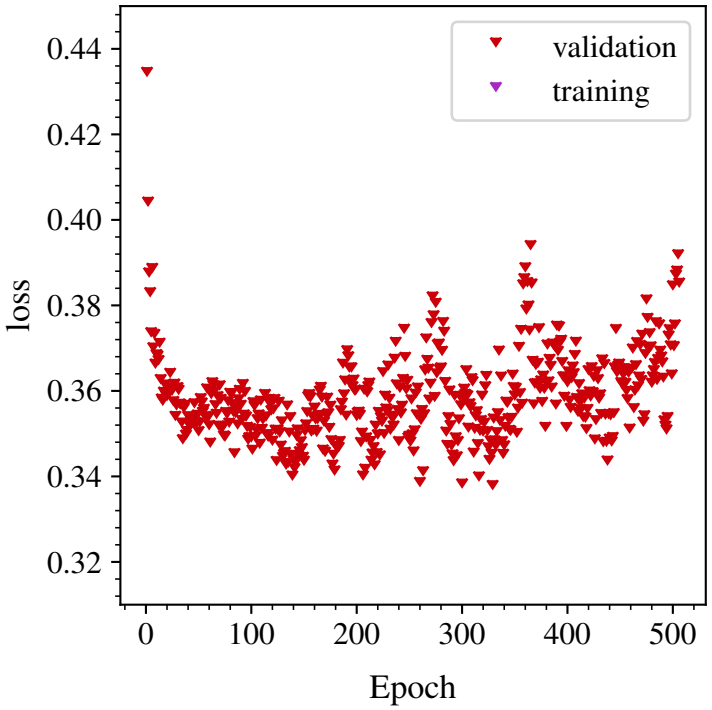


Figure 6.11: Loss function for the validation sample. The dip of the arrow indicates the loss value.

Chapter 7

Neural Network for Six Species: Multi-Class Classification

So far, our focus has been on evaluating the performance of various methods for binary classification, where the objective is to distinguish between only two species. However, there are scenarios where it is crucial to simultaneously separate more than two species, which falls under the domain of multi-class classification.

Initially, we include an examination of the neural network's outputs in section 7.1. This also provides information about species similarities. For multi-class classification, the same methods as before can be used. However, the normalization process needs to be adjusted for multi-class classification, as outlined in section 7.2. In section 7.3, we discuss the performance of the various methods for multi-class classification and we compare it to one from binary classification. Next, in section 7.4 we will analyze the dependence on kinematics bins for the multi-class methods. A study of the global PID performance, i.e all studying the six particles simultaneously, is done in section 7.5. At the end, section 7.6 we evaluate the multi-class performance for various species in the background.

Naturally, the neural network for 2 species, capable of separating only K and π, is not suitable for multi-class classification. Therefore, in the realm of multi-class classification, we compare the Pure Likelihood approach with the neural network for 6 species with cluster-shape. Again, to give the reader an easier understanding of the classification variables obtained after the normalization, tables are given in appendix A.

7.1 Evaluation of the Neural Network's Output

It is interesting to see the direct outputs of the neural network. This yield insights into which particle species are relatively easier or more challenging to distinguish. Figure 7.1 shows the output of the neural network $O_{\mathrm{NN}}(h)$ for the different species, for particle-gun samples of various species.

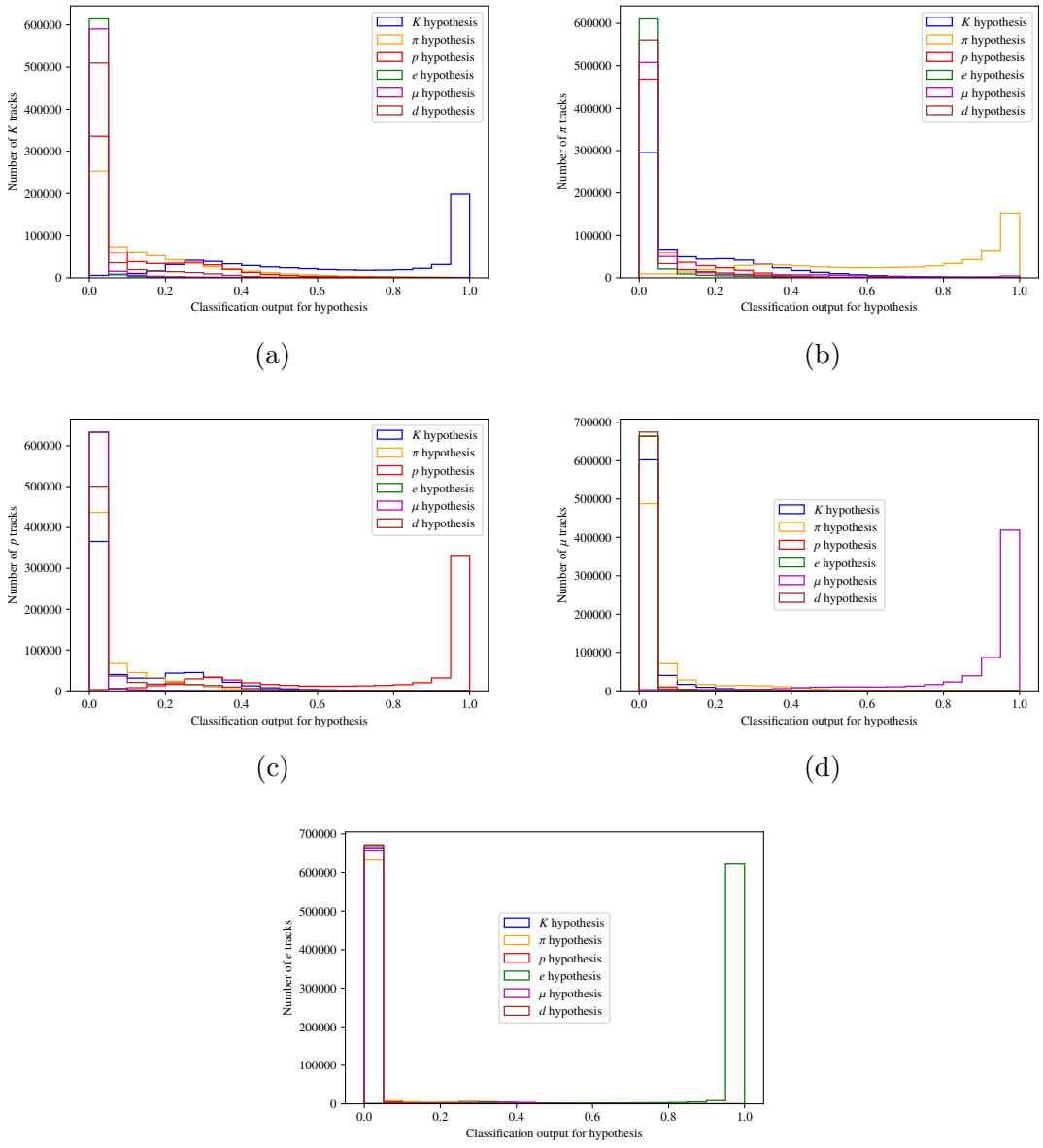


Figure 7.1: Distribution of the output of the neural network for hypothesis h, $O_{\mathrm{NN}}(h)$, for a particle-gun sample of true tracks of (a) K, (b) π, (c) p, (d)μ and (e) e. Here $O_{\mathrm{NN}}(K)$ is shown in blue, $O_{\mathrm{NN}}(\pi)$ is shown in yellow, $O_{\mathrm{NN}}(p)$ is shown in red, $O_{\mathrm{NN}}(e)$ is shown in green, $O_{\mathrm{NN}}(\mu)$ is shown in purple and $O_{\mathrm{NN}}(d)$ is shown in brown.

As expected, $O_{\mathrm{NN}}(K)$ peaks at 1 for the kaon sample, whereas the other $O_{\mathrm{NN}}(h)$ for $h \neq K$ peak at 0 (see Fig. 7.1a). However, both the $O_{\mathrm{NN}}(\pi)$ and $O_{\mathrm{NN}}(p)$ distributions exhibit a tail for kaon sample. This indicates that protons and pions are the species most difficult to differentiate from kaons.

Similarly, $O_{\mathrm{NN}}(\pi)$ exhibit a peak at 1 for the pion sample. And the other $O_{\mathrm{NN}}(h)$ for $h \neq \pi$ peak at 0 (see Fig. 7.1b). Once again, tails are observed for both $O_{\mathrm{NN}}(p)$ and $O_{\mathrm{NN}}(K)$. This indicates that protons and kaons are the species most difficult to differentiate from pions.

The same is shown for proton tracks (Fig. 7.1c), where one can observe that $O_{\mathrm{NN}}(p)$ peaks at 1, with other peaking at 0. Additionally, $O_{\mathrm{NN}}(K)$ exhibits a tail, reaffirming that kaons are the particles most closely to protons. This suggests that the signature of protons in the PID detectors is similar, though not identical, to that of kaons. Therefore, separating protons from kaons will be more challenging than distinguishing them from any other particle. As a result, it is evident that the neural network possesses sufficient separation power to discriminate hadrons.

Figure 7.1d shows the same for muon tracks. It can be seen that $O_{\mathrm{NN}}(\mu)$ peaks at 1, whereas the other $O_{\mathrm{NN}}(h)$ values for $h \neq \mu$ peak at 0. Further, a tail can be observed for $O_{\mathrm{NN}}(\pi)$. This can be explained by the fact that the muon mass is close the pion mass, which leads to a similar signature of these two species in the PID detectors, which makes their differentiation more challenging.

In contrast, for electron tracks, the outcomes are notably distinct. As shown in Fig. 7.1e, a clear peak at 1 is observed for electrons, with a corresponding clear peak at 0 for other hypotheses, with no tail existent. Further, $O_{\mathrm{NN}}(e)$ for the other particles (Figs. 7.1a to 7.1d) is peaking at 0 clearly, with no tail. The neural network has thus demonstrated its proficiency in distinguishing between leptons, specially performing for electrons.

In summary, we conclude that the network has successfully captured the distinctive features of each species, offering an good separation capability across all hypotheses. Additionally, this information regarding similarities among particles will prove valuable in later stages in this chapter.

7.2 Multi-class Normalitzation

For the neural network for 6 species in multi-class classification, we still use the same output variables. However, we perform a different normalitzation. We aim to perform α/J separation, where α represents one particle specie and J represents a selected set of particle species, denoted as $J = \{\beta,\ \gamma,\ \delta,\ \epsilon\ ...\}$. Hence, we aim to separate α from all species in J. For the the neural network for 6 species (multi-

class), classification variables are defined as

$$C(\alpha \ : \ j) = \frac{O_{\mathrm{NN}}(\alpha)}{O_{\mathrm{NN}}(\alpha) + \sum\limits_{j_i=\beta,\gamma...} O_{\mathrm{NN}}(j_i)} \tag{7.1}$$

The values of $C(\alpha \ : \ j)$ range from 0 to 1. This approach is referred to as the the neural network for 6 species (multi-class: α,β,γ,δ,...). For example, if one aims to distinguish K from three other particle types $\{\pi, \ \mu, \ e\}$, the classification variables read as follows:

$$C(K \ : \ \pi, \mu, e) = \frac{O_{\mathrm{NN}}(K)}{O_{\mathrm{NN}}(K) + O_{\mathrm{NN}}(\pi) + O_{\mathrm{NN}}(\mu) + O_{\mathrm{NN}}(e)} \tag{7.2}$$

For the Pure Likelihood in the multi-class case, we use the same variables as defined in Eq. (3.5). We can use analogously Eq. (7.1) replacing the outputs $O_{\mathrm{NN}}(h)$ by the likelihoods $\mathcal{L}(h)$ to obtain the Pure Likelihood multi-class classification variables. They read as:

$$C(\alpha \ : \ j) = \frac{\mathcal{L}(\alpha)}{\mathcal{L}(\alpha) + \sum\limits_{j_i=\beta,\gamma...} \mathcal{L}(j_i)} \tag{7.3}$$

This gives the Pure Likelihood approach (multi-class: α,β,γ,δ,...). It shares the same limitations discussed in section 3.4. We can do analogously the same for Eq. (7.2):

$$C(K \ : \ \pi, \mu, e) = \frac{\mathcal{L}(K)}{\mathcal{L}(K) + \mathcal{L}(\pi) + \mathcal{L}(\mu) + \mathcal{L}(e)} \tag{7.4}$$

In the following we use the Pure Likelihood approach and neural network for 6 species with cluster-shape. Despite being able to predict for 6 species, by using different normalizations we normalize it to 2 species in the case of binary; we normalize it to 4 species as an example (sections 7.3, 7.4 and 7.6); and we normalize it to 6 species in global PID (section 7.5).

7.3 Comparison of Multi-Class and Binary Classification

The first step is to compare the performance of binary classification against multi-class classification to asses if there is an overall decrease in performance by increasing the possible species as hypotheses. To this end, we are considering K and π as tracks. However, our aim extends beyond separating kaons from pions; we want to distinguish kaons from a predefined set of other potential hypotheses, i.e separating K from $\{\pi, \mu, e\}$. As an example, in this section we select the following potential species hypotheses: K, π, e, and μ. We choose only these four particles as constitute

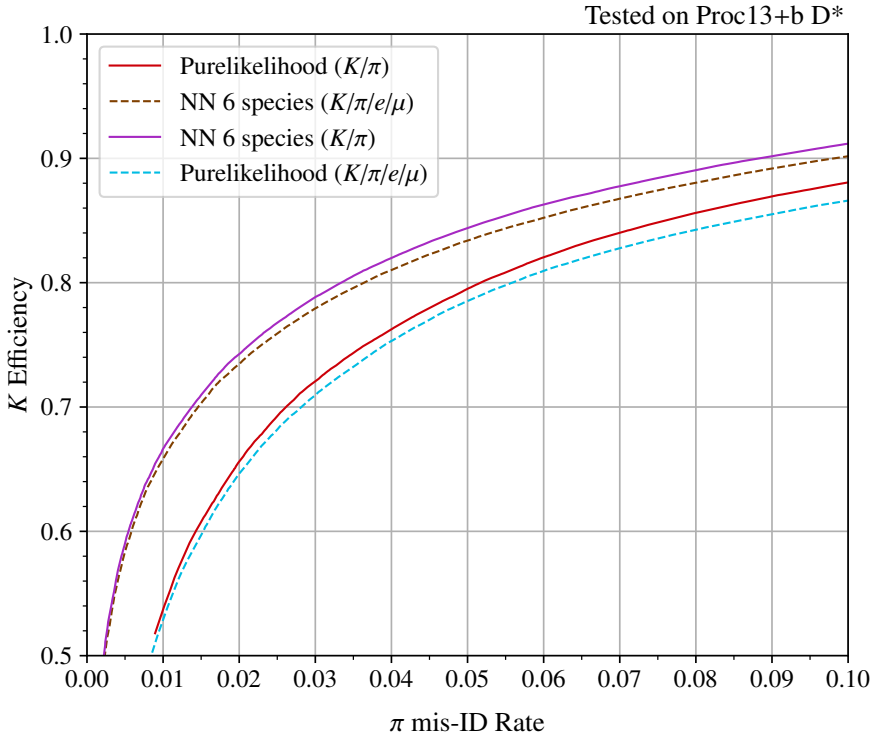


Figure 7.2: K/π separation performance on the D^* sample (see section 4.1.2.1). The methods normalized for K, π are displayed in solid lines: the neural network for 6 species (binary normalitzation) in purple and the Pure likelihood approach (binary) in red. The methods normalized for K,π,e,μ are displayed in dashed lines: the neural network for 6 species (multi-class: K,π,e,μ) in brown and the Pure likelihood approach(multi-class: K,π,e,μ) in blue.

the main product decays observed in Belle II. Protons and deuterons are comparatively less frequent.

Here, one can define two kinds of tasks. In the first category, we have methods capable of predicting between two hypotheses (K/π): the Pure Likelihood approach (binary) and the neural network (with binary normalization), represented by solid red and purple lines, respectively. These methods are the same in chapter 6.

In our example, the methods designed for multi-class classification are normalized to the four species mentioned. They are represented by dashed lines. They are the Pure Likelihood approach (multi-class: K,π,e,μ) in blue (see Eq. (7.4)), and the neural network (multi-class: K,π,e,μ) in brown (see Eq. (7.2)). In the following, the species inside the brackets indicate for which particles are the multi-class methods normalized to.

Figure 7.2 shows the multi-class classification performance on the real-data D^* sample for K/π separation. The neural network (binary) exhibits better performance than the neural network (multi-class: K,π,e,μ). Similarly, the Pure Likelihood ap-

proach(binary) outperforms the Pure Likelihood approach (multi-class: K,π,e,μ). This suggests that methods separating between two hypotheses tend to perform better than those separating for four hypotheses. Furthermore, we observe that the neural network also enhances the performance for multi-class classification.

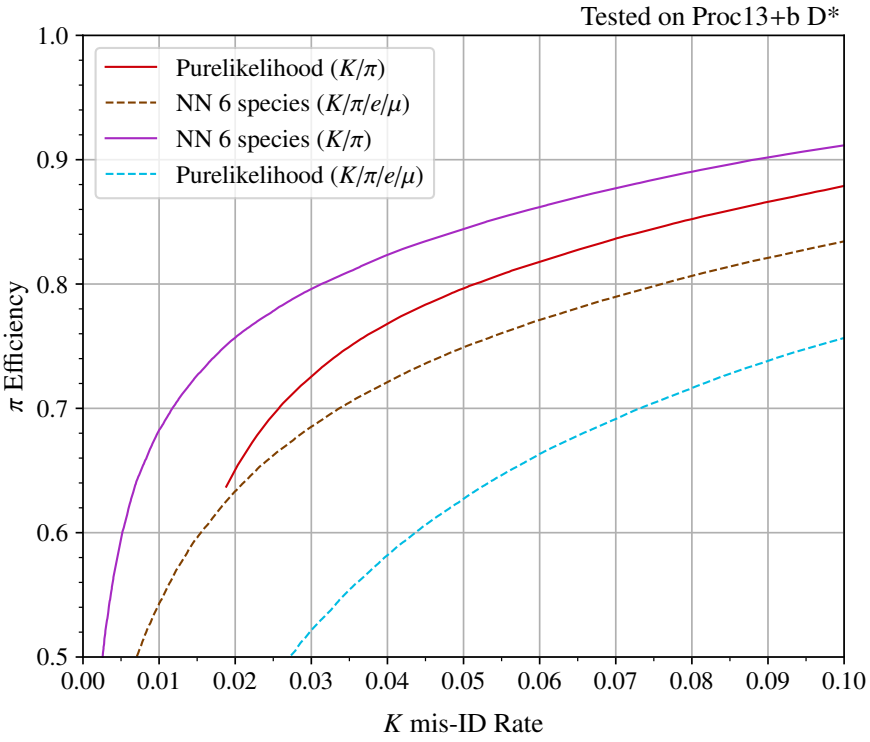
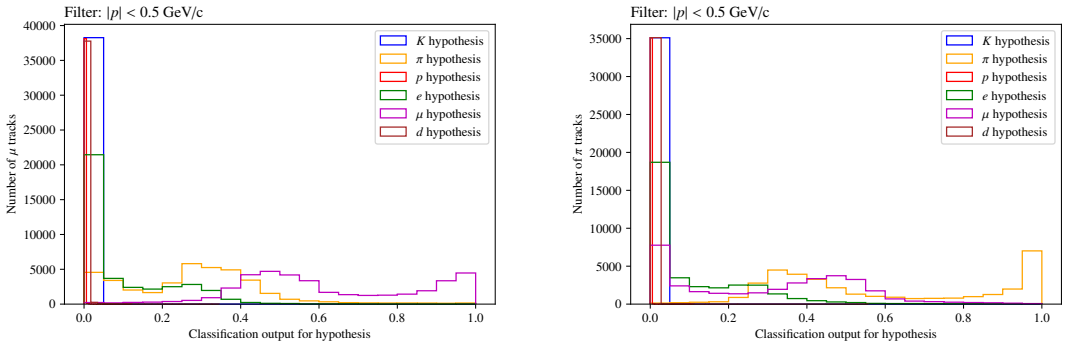


Figure 7.3: Same as Fig. 7.2, but for π efficiency against K misidentification rate.

When we separate π from $\{K\}$ or $\{K, \mu, e\}$ (see to Fig. 7.3), the performance changes drastically. The neural network (binary) and neural network (multi-class: K,π,e,μ) exhibit a substantial difference between each other, with the binary outperforming. Similarly, the Pure Likelihood approach(binary) and Pure Likelihood approach (multi-class: K,π,e,μ) have the same behaviour. Additionally, in π/K separation, the neural network (multi-class: K,π,e,μ) consistently outperforms the Pure Likelihood approach (multi-class: K,π,e,μ), underlining its capability for multi-class classification tasks.

Figure 7.2, i.e K efficiency and π misidentification rate, shows a minimal difference in performance between the neural network (binary) and neural network (multi-class: K,π,e,μ); and then a minimal difference for the Pure Likelihood approach (binary) and Pure Likelihood approach (multi-class: K,π,e,μ). Hence, incorporating e and μ in the normalization process has a minor effect in K/π separation due to the clear distinction between K and other particles. On the other hand, including e and μ in the normalization process completely alters the performance in π/K separation (see Fig. 7.3), as the pions tend to be more frequently misidentified with muons. This difference is caused by the similar masses of π and μ particles,



(a) (b)

Figure 7.4: Distribution of the output of the neural network $O_{\mathrm{NN}}(h)$ for h hypothesis for a particle-gun sample of true (a) μ and (b) π tracks in the low momentum region. Here $O_{\mathrm{NN}}(K)$ is shown in blue, $O_{\mathrm{NN}}(\pi)$ is shown in yellow, $O_{\mathrm{NN}}(p)$ is shown in red, $O_{\mathrm{NN}}(e)$ is shown in green, $O_{\mathrm{NN}}(\mu)$ is shown in purple and $O_{\mathrm{NN}}(d)$ is shown in brown.

making them difficult to differentiate. The limited power of discrimination between π and μ arises from the constraints of current detectors, particularly in the low momentum range $|\vec{p}| < 0.5$ GeV/c. dE/dX and Cherenkov radiation measurements effectively measure the particles mass and hence fail to distinguish between π and μ, as $m_\pi \approx m_\mu$. The KLM is the only detector capable of performing π/μ separation, which does not perform well in low momenta region. This is demonstrated in Fig. 7.4a in μ tracks within the low momenta range. It is evident that $O_{\mathrm{NN}}(\mu)$ (purple) does not exhibit a distinct peak at 1, as it ideally should. Similarly, $O_{\mathrm{NN}}(\pi)$ (yellow) does not peak at 0, but rather at approximately 0.4. This discrepancy is also observed in Fig. 7.4b, which illustrates the same scenario for π tracks. This highlights the importance of careful consideration when comparing more than two particles, as the performance depends on which species are considered for the separation.

Overall, one observe an intrinsic problem with multi-class classification which is an overall decrease in performance. This observation aligns with the expected behaviour, as the inclusion of additional particles in the prediction process increases the likelihood of misidentification of K or π as other species. Furthermore, we observed that there is a consistent superiority of the neural network (multi-class: K,π,e,μ) over the Pure Likelihood approach (multi-class: K,π,e,μ).

7.4 Performance in Kinematic Bins

Further, a performance analysis in kinematic bins $(|\vec{p}|, \cos\theta)$ has been conducted for multi-class classification using the real-data D^* sample. The results for K efficiency and π misidentification rate are presented in Fig. 7.5, for a target average π misidentification rate of 2 %. Both the neural network (binary) shown in purple, and the neural network (multi-class: K,π,e,μ) shown in brown, exhibit similar

performance. The same hold for Pure likelihood approach (binary) (red) and Pure likelihood approach (multi-class: K,π,e,μ) (blue). We observe similar dependence as for the binary classification (see section 6.4.2).

However, this does not apply to π efficiency and K misidentification rate (see Fig. 7.6). Figure 7.6 (top left) shows the π efficiency on real data as a function of the track momentum. The neural network (multi-class: K,π,e,μ) has higher efficiency than the Pure Likelihood approach (multi-class: K,π,e,μ) across all momentum ranges. In the top right, the K misidentification rate is shown. Below 1 GeV/c, both methods exhibit a similar misidentification rate. In the range of $1 \lesssim |\vec{p}| \lesssim 2.5$ GeV/c, the neural network (multi-class: K,π,e,μ) has a significantly lower misidentification rate compared to the Pure Likelihood approach (multi-class: K,π,e,μ). Beyond approximately 2.5 GeV/c, the neural network (multi-class: K,π,e,μ) shows a slightly higher misidentification rate compared to the Pure Likelihood approach (multi-class: K,π,e,μ). Nevertheless, within this momentum range, the π efficiency is approximately 2.5 times higher for the neural network (multi-class: K,π,e,μ), indicating that it performs better than the Pure Likelihood approach (multi-class: K,π,e,μ). Additionally, it is evident that below 1 GeV/c in top-left plot, the discrepancy between both neural network methods (purple against brown) and between both Pure Likelihood approach methods (red against blue) is exceptionally large when compared to other regions. This is attributed to the low μ/π separation power, as explained above. For that, the binary normalized methods perform much better than the multi-class normalized ones.

Figure 7.6 (bottom left) shows the π efficiency on real data as a function of $\cos\theta$. The neural network (multi-class: K,π,e,μ) exhibits higher efficiency than the Pure likelihood approach (multi-class: K,π,e,μ) across the entire range of $\cos\theta$, with a particularly notable improvement for $\cos\theta > -0.5$. However, for $\cos\theta < -0.5$, both methods show a clear decline in performance, as elaborated in the previous section for the backward region. Furthermore, beyond $\cos\theta \sim 0.85$, there is a reduction in efficiency for both methods. Moving to the K misidentification rate (bottom right plot), its behaviour depends on the considered region. For $-0.5 < \cos\theta < 0.25$, the K misidentification rate is lower for the neural network (multi-class: K,π,e,μ); whereas in the other region, it is lower for the Pure Likelihood approach (multi-class: K,π,e,μ). Additionally, a spike similar to the one observed previously for Pure likelihood approach (binary) can be seen for Pure likelihood approach (multi-class: K,π,e,μ) in the K misidentification rate plot at $\cos\theta \approx -0.5$. This spike is once again circumvented for the neural network (multi-class: K,π,e,μ) by excluding the TOP information in the region $-0.55 < \cos\theta < -0.50$ (as detailed in section 4.3.3), without compromising performance in this region.

Up to this point, we compared both multi-class methods, i.e blue versus brown. Furthermore, one can compare both of the neural network methods, i.e brown vs purple. For any kinematic bin, the efficiency of the neural network (binary) is either similar or superior to that of the neural network (multi-class: K,π,e,μ) with comparable misidentification rates. This finding underscores the consistency of the

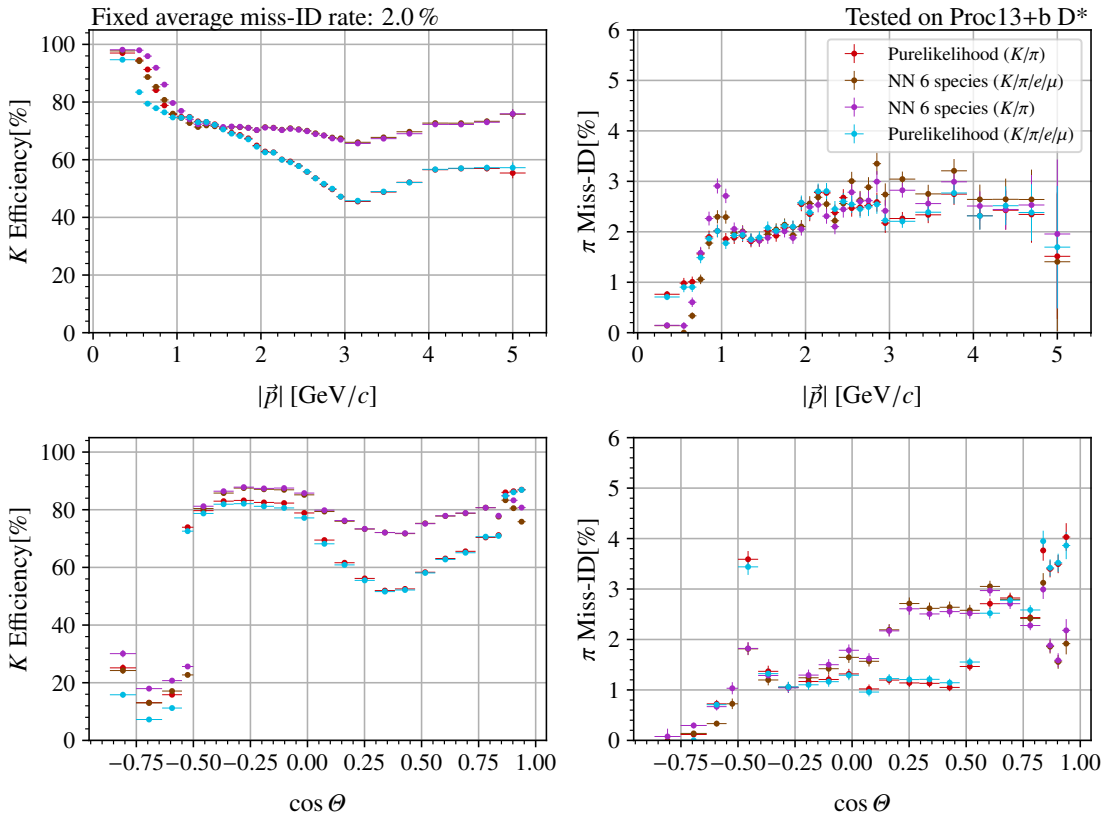


Figure 7.5: Performance for K identification for a target average π misidentification rate of 2 %, on the real-data D^* sample as a function of the track momentum (top row) and as a function of $\cos\theta$ (bottom row). The left column shows the K efficiency. The right column shows the π misidentification rate. The red data points represent the performance of the Pure likelihood approach (binary). The purple data points represent the performance of the neural network for 6 species with cluster-shape (binary). The blue data points represent the performance of the Pure likelihood approach (multi-class: K,π,e,μ). The brown data points represent the performance of the neural network for 6 species with cluster-shape (multi-class: K,π,e,μ).

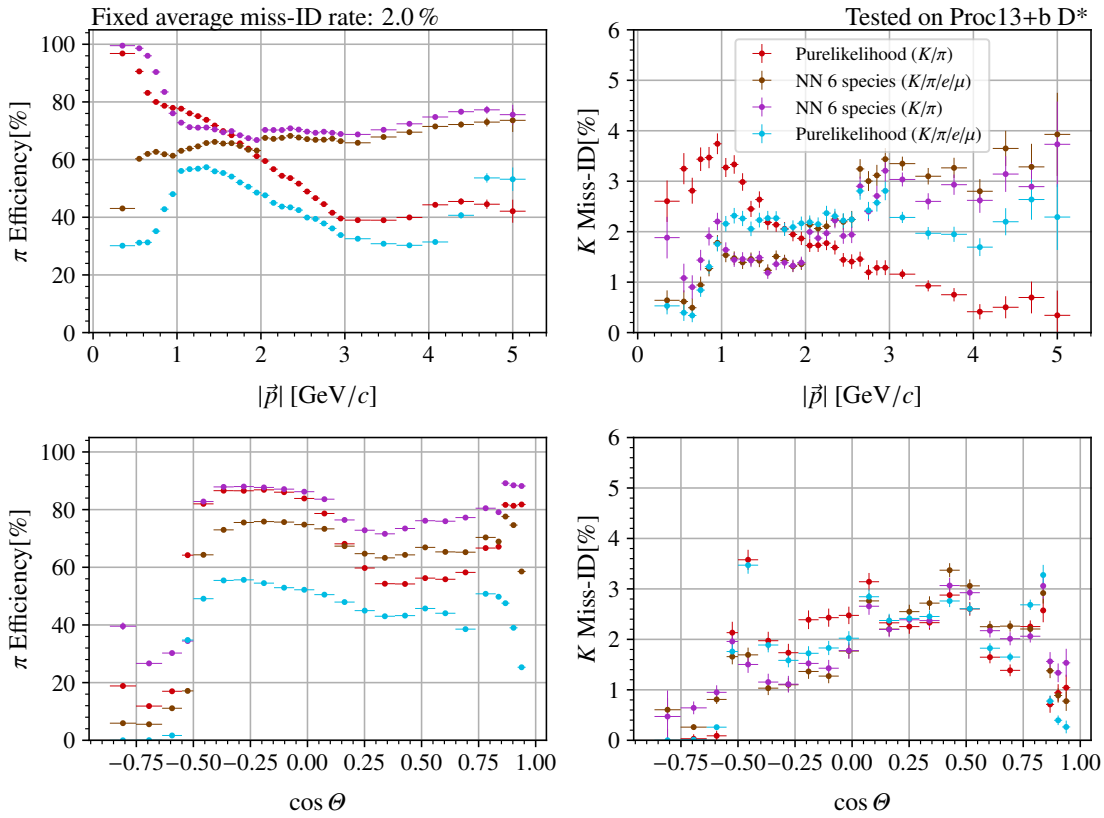


Figure 7.6: Same as Fig. 7.5 but for π identification for a target average K misidentification rate of 2 %.

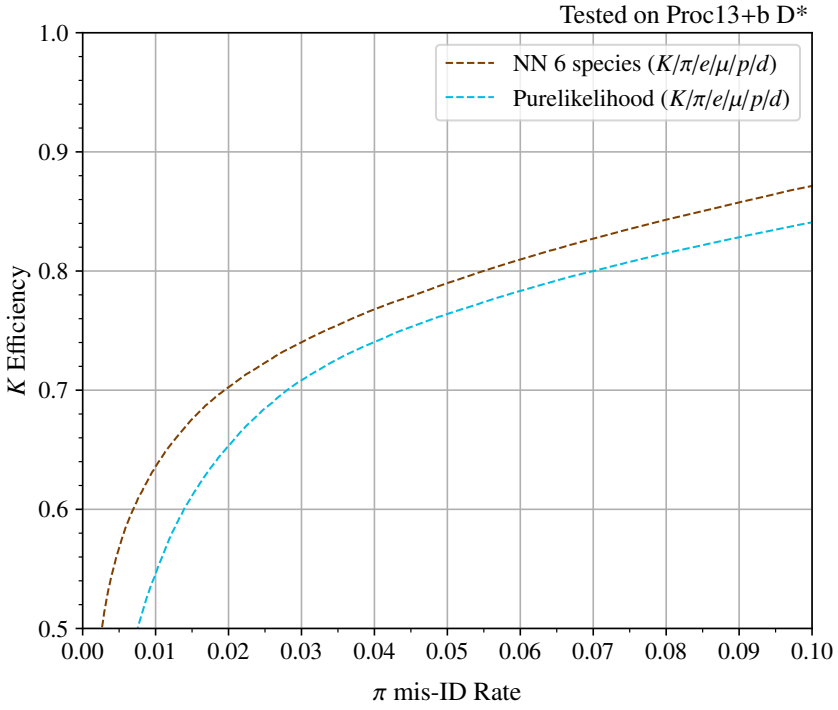


Figure 7.7: K/π separation performance on the D^* sample (see section 4.1.2.1)for the neural network for 6 species (multi-class: K,π,e,μ,p,d) in brown, and the Pure likelihood approach (multi-class: K,π,e,μ,p,d) in blue.

neural network.

7.5 Global PID

In this section, we extend the concept of multi-class classification to encompass all six particle species, including not only K, π, e, and μ, as done above, but also d and p. This is called usually global PID. In this context, we aim to evaluate the performance of the neural network (multi-class: K,π,e,μ,p,d) against the Pure Likelihood approach (multi-class: K,π,e,μ,p,d).

Figure 7.7 shows the K/π separation performance for global classification on the real-data D^* sample, thus considering K and π as tracks. It shows that the neural network (multi-class: K,π,e,μ,p,d) (brown) consistently outperforms the Pure Likelihood approach (multi-class: K,π,e,μ,p,d) (blue). The similar behaviour is observed in π/K separation (see Fig. 7.8). This superior performance underscores the efficacy of the neural network approach in handling the complexities introduced by more particle species.

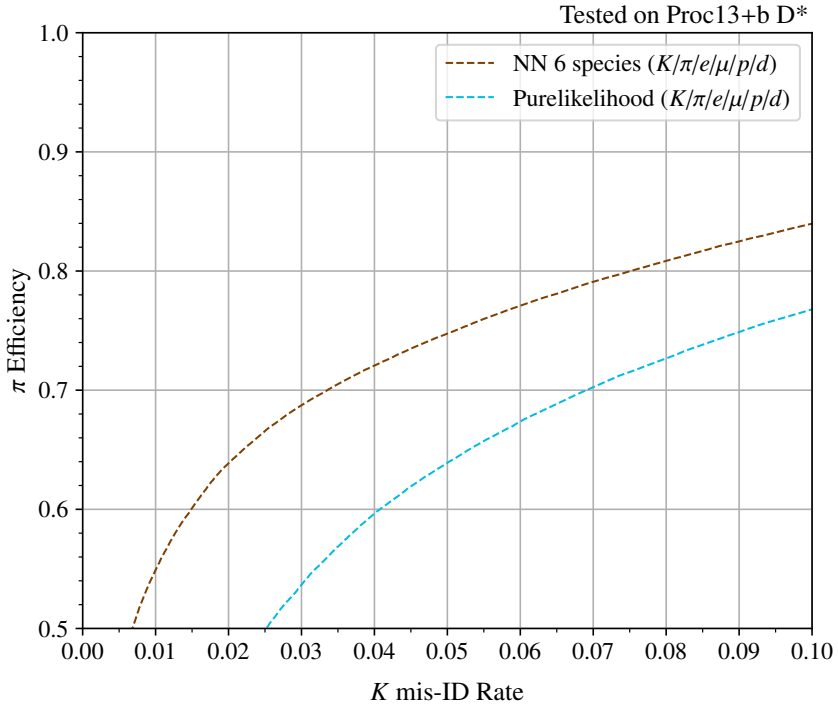


Figure 7.8: Same as Fig. 7.7 but for π efficiency against K misidentification rate.

7.6 Sample with Various Species in the Background

So far, multi-class classification has only been tested for K or π tracks. However, we also aim to evaluate performance when dealing with more than two species tracks. This is useful for samples with various species in background, which happens in experimental data.

Therefore, we use a pgMC sample for K, π, e, and μ tracks simultaneously, with equal track quantities. The species chosen for PID significantly impacts the ROC curve, as shown above. For instance, to analyse the efficiency of K, one needs corresponding plots for misidentification rate of π, e, and μ. Thus, it is necessary to define a variable that allows the creation of one ROC curve per particle efficiency. For that purpose, we define a ROC curve for a particle h, where we represent its efficiency against the total impurity in a selected-h sample. The total impurity is defined as:

$$\text{Impurity } (h) = \frac{\#Fake}{\#Total} = \frac{\#Fake}{\#True + \#Fake} = \frac{\sum_{i \neq h} \text{misID}_i N_i}{\text{eff}_h N_h + \sum_{i \neq h} \text{misID}_i N_i} \tag{7.5}$$

Here, h represents the particle specie we want to select, eff_h is the efficiency of h particle, i represents the other species, misID_i is the miss-identification of a particle of species i being identified as h, and N_j is the number of tracks for J species in the pgMC sample.

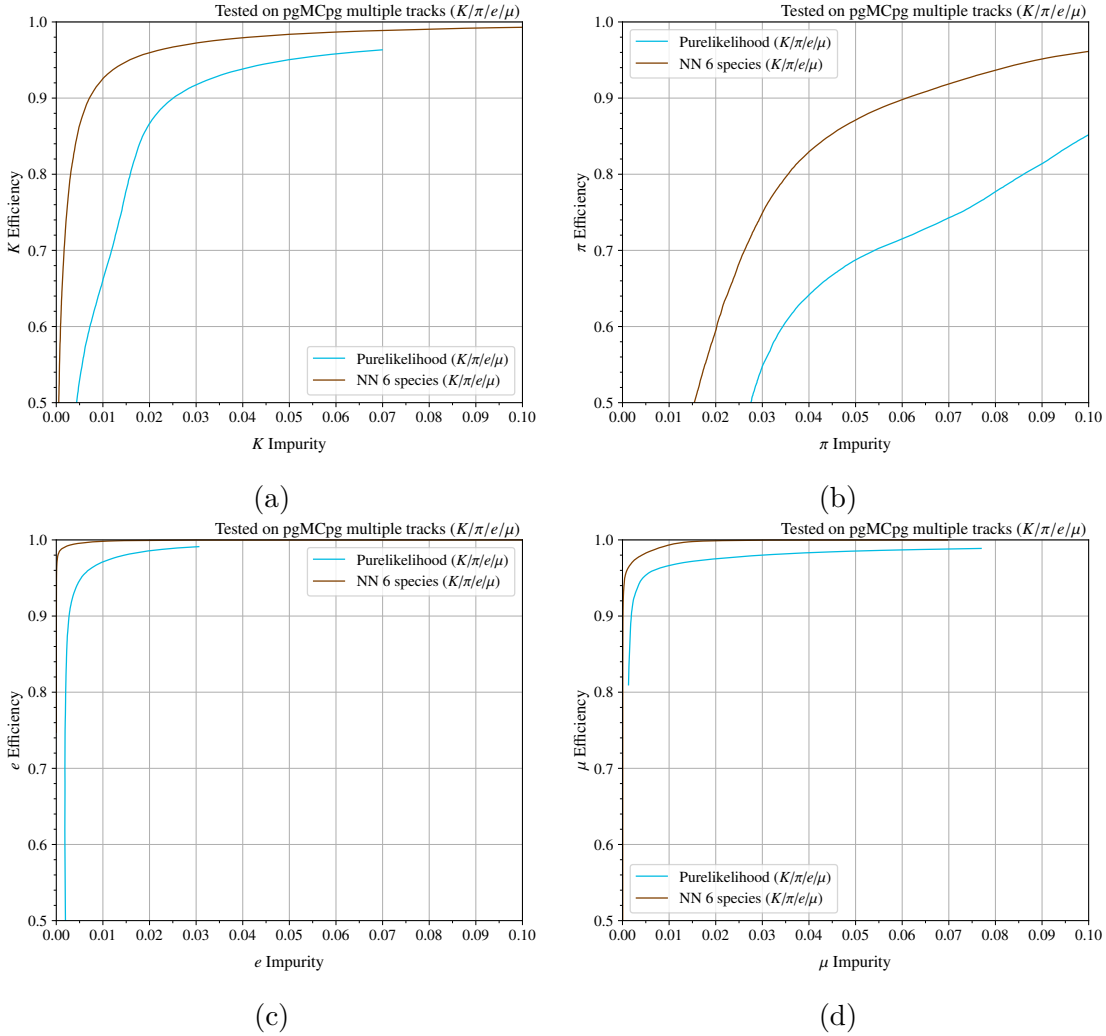


Figure 7.9: ROC curve evaluated on the pgMC sample decays (see section 4.1.1) for the neural network for 6 species with cluster-shape (multi-class: K,π,e,μ) (brown) and for the Pure likelihood approach (multi-class: K,π,e,μ) (blue). (a) shows K efficiency against its impurity, (b) shows π efficiency against its impurity, (c) shows e efficiency against its impurity, and (d) shows μ efficiency against its impurity.

Figure 7.9a illustrates the K efficiency versus impurity of a selected-K for the K, π, μ, and e pgMC sample. One can clearly see that the neural network (multi-class: K,π,e,μ) (brown) clearly outperforms the Pure likelihood approach (multi-class: K,π,e,μ) (blue). Furthermore, the same type of analysis can be done for the other species: Fig. 7.9b shows the same for π, Fig. 7.9d for μ, and Fig. 7.9c for e. The results observed are consistent.

In summary, this section demonstrates that the neural network is well-suited for multi-class classification, in all cases outperforming the Pure likelihood approach, with more than two differnt species tracks.

Chapter 8

Conclusions and Outlook

8.1 Conclusions

Beginning with the initial works of Tsaklidis et al. [8] and Wallner et al. , we have demonstrated a substantial improvement in K/π separation performance through the introduction of a specialized neural network with two outputs. The performance is affected, despite not being crucial, by the neural network architecture. It was optimized using hyperparameter tuning. Furthermore, by using feature importance, we found were this improvement comes from. It is due to the fact that in the calculation of likelihoods, approximations were made. This advancement represented an initial significant leap forward in the methodologies employed so far in the Belle II experiment.

This motivated us to extend this approach to the identification of all species with a single neural network, i.e. to develop a universal PID method. Remarkably, this more complex task did not result in a loss in performance in K/π separation when compared to the specialized neural network. On the contrary, it comes with a multitude of advantages. For instance, the neural network with two outputs is limited to K/π separation, while the neural network with 6 is not.

The neural network with six outputs not only allows for K/π separation, but also allows binary classification of any combination of two charged particle species, including both hadrons and leptons. The new neural network with six outputs outperformed the Pure Likelihood approach in all studied cases. Also, neural network with six outputs performs better than the Boosted Decision Tree (BDT) for lepton PID, in all cases. Moreover, its adaptability to a diverse range of sample, both real and simulated data, show its versatility. Therefore, the neural network with six outputs surpasses its neural network with two outputs.

Furthermore, the neural network with six outputs allows for multi-class classification. We shown that multi-class classification task comes with a reduction of overall performance for all methods. The neural network for six outputs shown the best performance also in multi-class classification compared to the Pure Likelihood approach.

We have successfully achieved our initial objectives by creating a PID method that demonstrates superior performance across all mentioned cases, surpassing existing methods used in the Belle II experiment.

8.2 Outlook

While we achieved encouraging results, there is room for enhancements and future developments. The advancements presented in this master thesis not only represent a substantial leap forward in classification methodologies for the Belle II experiment, with the improved performance. They also open up a multitude of possibilities for future research and applications in this field.

One area with significant potential for improvement lies in the selection of inputs. For instance, one can integrate more detailed information obtained from the detectors as we did it with the ECL cluster-shape variables. This has the potential to overcome limitations of the likelihood calculation.

Furthermore, we can add information from other detectors, like the PXD detector, which has not been implemented so far.

Most of the inputs are related with the log-likelihood information. As previously explained, these requires modelling, which might be imperfect. More accurate likelihood can be developed in the future. They could easily be incorporated in the neural network approach by retraining the neural network.

Appendix A

Neural Networks With Their Normalizations

Table 4.2 shows the main three neural networks presented in this work. However, the neural networks can be used in different ways depending on the normalitzation process employed. Sections 3.4 and 4.4 shows the binary normalitzation, whereas section 7.2 shows the multi-class normalitzation. Furthermore, different multi-class classification cases are used.

Tables A.1 and A.2 give the classification variables, obtained from normalizing the neural networks, of the different cases shown along the work.

Table A.1: Comparison of the classification variables used along the work. It includes code the neural network name used (see table 4.2), its outputs and how they are normalized to convert them into classification variables. The ones without indentation represent the general case. The ones with indentation and the symbol ↳ represent specific cases.

Neural Network	**Direct Outputs**	**Normalitzation**	**Classification variables**
Neural network for 2 species	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$		$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$ where $O_{\mathrm{NN}}(\pi) + O_{\mathrm{NN}}(K) = 1$
Neural network for 6 species without cluster shape (binary normalitzation)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	$\rightarrow$ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(h_1 : h_2)$, $C_{\mathrm{NN}}(h_2 : h_1)$ for h_1, h_2 chosen, where $C_{\mathrm{NN}}(h_1 : h_2) + C_{\mathrm{NN}}(h_2 : h_1) = 1$
↳ Neural network for 6 species without cluster shape (binary: K,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	$\rightarrow$ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(K : \pi)$, $C_{\mathrm{NN}}(\pi : K)$
↳ Neural network for 6 species without cluster shape (binary: e,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	$\rightarrow$ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(e : \pi)$, $C_{\mathrm{NN}}(\pi : e)$
↳ Neural network for 6 species without cluster shape (binary: μ,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	$\rightarrow$ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(\mu : \pi)$, $C_{\mathrm{NN}}(\pi : \mu)$

Table A.2: Continuation of table A.1

Neural Network	**Direct Outputs**	**Normalitzation**	**Classification variables**
†Neural network for 6 species with cluster shape (binary normalitzation)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(h_1 : h_2)$, $C_{\mathrm{NN}}(h_2 : h_1)$ for h_1, h_2 chosen, where $C_{\mathrm{NN}}(h_1 : h_2) + C_{\mathrm{NN}}(h_2 : h_1) = 1$
↳ †Neural network for 6 species with cluster shape (binary: K,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(K : \pi)$, $C_{\mathrm{NN}}(\pi : K)$
↳ †Neural network for 6 species with cluster shape (binary: e,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(e : \pi)$, $C_{\mathrm{NN}}(\pi : e)$
↳ †Neural network for 6 species with cluster shape (binary: μ,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(\mu : \pi)$, $C_{\mathrm{NN}}(\pi : \mu)$
↳ †Neural network for 6 species with cluster shape (binary: e,μ)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(e : \mu)$, $C_{\mathrm{NN}}(\mu : e)$
↳ †Neural network for 6 species with cluster shape (binary: p,π)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ binary norm. (Eq. (4.6))	$C_{\mathrm{NN}}(p : \pi)$, $C_{\mathrm{NN}}(\pi : p)$
† Neural network for 6 species with cluster shape (multi-class: K,π,e,μ)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ multi-class norm. Eq. (7.1)	$C_{\mathrm{NN}}(K)$, $C_{\mathrm{NN}}(\pi)$ $C_{\mathrm{NN}}(e)$, $C_{\mathrm{NN}}(\mu)$ $\sum_{h_i} C_{\mathrm{NN}}(h_i) = 1$
† Neural network for 6 species with cluster shape (multi-class: K,π,e,μ,p,d)	$O_{\mathrm{NN}}(K)$, $O_{\mathrm{NN}}(\pi)$, $O_{\mathrm{NN}}(p)$ $O_{\mathrm{NN}}(\mu)$, $O_{\mathrm{NN}}(e)$, $O_{\mathrm{NN}}(d)$	→ multi-class norm. Eq. (7.1)	$C_{\mathrm{NN}}(K)$, $C_{\mathrm{NN}}(\pi)$, $C_{\mathrm{NN}}(p)$ $C_{\mathrm{NN}}(\mu)$, $C_{\mathrm{NN}}(e)$, $C_{\mathrm{NN}}(d)$ $\sum_{h_i} C_{\mathrm{NN}}(h_i) = 1$

† These neural networks are exactly the same. The only thing it changes is the normalization process

Bibliography

[1] M. K. Gaillard, P. D. Grannis, and F. J. Sciulli, "The standard model of particle physics," *Reviews of Modern Physics*, vol. 71, no. 2, p. S96, 1999.

[2] H.-Y. Cheng, "The strong cp problem revisited," *Physics Reports*, vol. 158, no. 1, pp. 1–89, 1988.

[3] E. Kou, P. Urquijo, *et al.*, "The Belle II Physics Book," *Progress of Theoretical and Experimental Physics*, vol. 2019, p. 123C01, 12 2019.

[4] N. Toutounji and K. Varvell, *Reconstruction Methods for Semi-leptonic Decays of B-mesons with the Belle II Experiment*. PhD thesis, Sydney, The University of Sydney, Sydney, 2019. Presented on 21 01 2019.

[5] L. Zani, "Studies on τ decays at belle ii," 2023.

[6] B. I. L. I. Group, "Muon and electron identification performance with 189 fb^1 of belle ii data," Apr 2021.

[7] L. group and B. I. Collaboration, "Muon and electron identification efficiencies and hadron-lepton mis-identification rates at belle ii for moriond 2021," Mar 2021. For Moriond 2021.

[8] I. Tsaklidis and F. Bernlochner, "Improving kaon-pion separation with neural networks," Nov 2021.

[9] P. D. Group *et al.*, "Review of Particle Physics," *Progress of Theoretical and Experimental Physics*, vol. 2022, p. 083C01, 08 2022.

[10] Kuhr, Thomas, "Belle ii at the start of data taking," *EPJ Web Conf.*, vol. 214, p. 09004, 2019.

[11] T. Abe *et al.*, "Belle ii technical design report," 2010.

[12] M. Yonenaga and H. Kakuno, *Particle Identification using the Aerogel RICH Counter at the Belle II Experiment*. PhD thesis, Hachioji, Tokyo Metropolitan University, Hachioji, 2020. Presented on 31 08 2020.

[13] C.-H. Kim, Y. Unno, H. Cho, B. Cheon, S. Kim, I. Lee, E.-J. Jang, S.-K. Choi, Y. Kim, J. Ahn, M. Remnev, A. Kuzmin, T. Koga, Y.-T. Lai, Y. Iwasaki, H. Nakazawa, D. Liventsev, M. Nakao, S. Yamada, R. Itoh, T. Konno, S.-H. Park, Y.-J. Kwon, O. Hartbrich, and M. Ritzert, "Trigger slow control system of

the belle ii experiment," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1014, p. 165748, 2021.

[14] M. Nakao, T. Higuchi, R. Itoh, and S. Y. Suzuki, "Data acquisition system for belle ii," *Journal of Instrumentation*, vol. 5, p. C12004, dec 2010.

[15] J. V. Jelley, "Cerenkov radiation and its applications," *British Journal of Applied Physics*, vol. 6, p. 227, jul 1955.

[16] K. Kleinknecht, *Detektoren fuer teilchenstrahlung*. Springer-Verlag, 2015.

[17] L. Vitale, "Belle ii experiment: Status and prospects," Aug 2019. 30 min.

[18] K. Adamczyk *et al.*, "The design, construction, operation and performance of the belle II silicon vertex detector," *Journal of Instrumentation*, vol. 17, p. P11042, nov 2022.

[19] L. Zani *et al.*, "The silicon vertex detector of the belle II experiment," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1038, p. 166952, sep 2022.

[20] N. Taniguchi, "Central Drift Chamber for Belle-II," *Journal of Instrumentation*, vol. 12, p. C06014, June 2017.

[21] Y. Horii, "TOP Detector for Particle Identification at the Belle II Experiment," *PoS*, vol. EPS-HEP2013, p. 500, 2013.

[22] J. Fast, "The belle ii imaging time-of-propagation (itop) detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 876, pp. 145–148, 2017. The 9th international workshop on Ring Imaging Cherenkov Detectors (RICH2016).

[23] M. Starič, "Pattern recognition for the time-of-propagation counter," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 639, no. 1, pp. 252–255, 2011. Proceedings of the Seventh International Workshop on Ring Imaging Cherenkov Detectors.

[24] KEK, "The TOP counter: a new method for particle identification." `https://www2.kek.jp/proffice/archives/feature/2010/BelleIIBPID.html`.

[25] S. Korpar, I. Adachi, N. Hamada, M. Higuchi, T. Iijima, S. Iwata, H. Kakuno, H. Kawai, P. Križan, S. Nishida, *et al.*, "A 144-channel hapd for the aerogel rich at belle ii," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 766, pp. 145–147, 2014.

[26] T. Iijima, S. Korpar, I. Adachi, S. Fratina, T. Fukushima, A. Gorišek, H. Kawai, H. Konishi, Y. Kozakai, P. Križan, T. Matsumoto, Y. Mazuka, S. Nishida, S. Ogawa, S. Ohtake, R. Pestotnik, S. Saitoh, T. Seki, T. Sumiyoshi, Y. Uchida, Y. Unno, and S. Yamamoto, "A novel type of proximity focusing rich counter with multiple refractive index aerogel radiator," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 548, no. 3, pp. 383–390, 2005.

[27] M. Yonenaga *et al.*, "Performance evaluation of the aerogel RICH counter for the Belle II spectrometer using early beam collision data," *Progress of Theoretical and Experimental Physics*, vol. 2020, p. 093H01, 08 2020.

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

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

[30] M. A. Palaia, F. Forti, and F. Tenchini, *Optimizing the sensitivity to the $\tau \to l\gamma$ decay with a novel tag approach at Belle II*. PhD thesis, Pisa, University of Pisa, Pisa, 2022. Presented on 12 12 2022.

[31] G. Edenhofer, *Optimization of Particle Identification*. PhD thesis, 06 2018.

[32] J.-F. Krohn, C. Hagner, and A. Glazov, "K 0 l identification studies for belle ii," 2016.

[33] D. Berrar, *Performance Measures for Binary Classification*, pp. 546–560. 2019.

[34] M. Varela, "Slow pion identification using the pixel detector of belle ii," Master's thesis, Faculty of Physics, Ludwig-Maximilians-Universität Munich, Munich, Germany, September 7 2023.

[35] S. Wallner, "Update of k-pi separation using neural networks," 2023.

[36] The Belle II Collaboration, "Belle ii analysis software framework (basf2)," Aug. 2022.

[37] K. Nakamura *et al.*, "Particle physics booklet," 01 2010.

[38] L. Kopke and N. Wermes, "J/psi Decays," *Phys. Rept.*, vol. 174, p. 67, 1989.

[39] J. Bilk and J. S. Lange, *Employing Deep Learning to Find Slow Pions in the Pixel Detector in the Belle II Experiment*. PhD thesis, Giessen, Justus-Liebig-University, Giessen, 2021. Presented on 25 11 2021.

[40] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. `http://www.deeplearningbook.org`.

[41] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *Nuclear Science, IEEE Transactions on*, vol. 44, pp. 1464 – 1468, 07 1997.

[42] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[43] H. Yao, D.-l. Zhu, B. Jiang, and P. Yu, "Negative log likelihood ratio loss for deep neural network classification," in *Proceedings of the Future Technologies Conference (FTC) 2019: Volume 1*, pp. 276–282, Springer, 2020.

[44] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," *CoRR*, vol. abs/1907.10902, 2019.

[45] S. Wallner, X. Simo, H.-G. Moser, D. Greenwald, and S. Paul, "Particle identification of kaons and pions using a neural network," Jul 2023.

[46] T. Hara, T. Kuhr, and Y. Ushiroda, "Belle II coordinate system and guideline of belle II numbering scheme," 8 2011.

[47] H. Ye *et al.*, "Commissioning and performance of the belle ii pixel detector," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 987, p. 164875, 2021.

[48] U. Tamponi, "The top counter of belle ii: status and first results," 2018.

[49] A. Abashian *et al.*, "The Belle Detector," *Nucl. Instrum. Meth. A*, vol. 479, pp. 117–232, 2002.

[50] L. Schinnerl, T. Kuhr, and N. Hartmann, *Analysis Specific Filters for Smart Background Simulations at Belle II*. PhD thesis, Munich, LMU, Munich, 2022. Presented on 08 08 2022.

[51] P. Urquijo, "Physics prospects at the belle ii experiment," *Nuclear and Particle Physics Proceedings*, vol. 263-264, pp. 15–23, 2015. Capri 2014 – Fifth workshop on Theory, Phenomenology and Experiments in Flavour Physics.

Acknowledgements

First of all, I would like to thank Stefan in particular. Your work and previous research was an important starting point for the development of this work. I have really enjoyed the last few months and learned a lot. Not forgetting all the support and help throughout the project and in the weekly meetings. For all that and more, thank you.

I would also like to thank Daniel. When I was looking for a Master's thesis, you recommended particle physics to me and the truth is that I couldn't be happier with my choice.

Thanks also to all the people in both the E18 group and the MPP. You have helped me a lot from day one, with guidance at the beginning of my project, as well as questions and discussions in general. Not only on a professional level but also on a personal level.

Last but not least, I would like to thank my family and friends for their support whenever I needed it.