Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

BACHELOR'S THESIS

Application of Machine Learning Methods for the Identification of Proton Decay in Liquid Argon Detector

Anna Gul'a Gartman

Supervised by
Ing. Jiří Franc, Ph.D.

2024

Acknowledgments

I would first like to thank my supervisor, Jiří Franc, for leading (and sometimes pushing) me along the path I thought I would never venture.

I am very thankful to my advisor, Viktor Pěč who provided us with the data and was always on the line to help us disentangle the physics behind the numbers and the lines.

Furthermore, I would like to thank Pavel Strachota for providing technical support related to working on the HELIOS cluster and for his advice on writing this thesis.

Declaration

I declare that this Bachelor's Thesis is entirely my own work and I have listed all the used sources in the bibliography.

Anna Gul'a Gartman

This bachelor's thesis was conducted under the supervision of Ing. Jiří Franc, Ph.D. and Mgr. Viktor Pěč, Ph.D. as the advisor.

České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská

Katedra: matematiky

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Anna Guľa Gartman
Studijní program:	Aplikované matematicko-stochastické metody
Název práce (česky):	Aplikace metod strojového učení pro identifikaci rozpadu protonu v detektoru s kapalným argonem
Název práce (anglicky):	Application of machine learning methods for the identification of proton decay in liquid argon detector

Pokyny pro vypracování:

1) Seznamte se s experimentem DUNE a detektory na bázi tekutého argonu.
2) Nastudujte možnosti detekce rozpadu protonu v experimentu DUNE a zaměřte se na kanál p -> K + nu.
3) Vyberte vhodné metody strojového učení pro identifikaci signálu a jeho separaci od pozadí tvořeného interakcí atmosferického neutrina s jádrem argonu.
4) Natrénujte vybrané metody pomocí nasimulovaných vzorků a porovnejte s dosud dosaženými výsledky.
5) Použijte ensemble techniky pro vytvoření finálního modelu.

Doporučená literatura:

1) B. Abi, R. Acciarri, M. A. Acero, et al., Prospects for beyond the Standard Model physics searches at the Deep Underground Neutrino Experiment. Eur. Phys. J. C 81, 322, 2021.

2) Ch. Alt, Sensitivity study for proton decay via p->vK using a 10 kiloton dual phase liquid argon time projection chamber at the Deep Underground Neutrino Experiment. PhD Thesis, ETH, Zurich, 2020.

3) A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019.

4) I. Goodfellow, Y. Bengio, A. Courville, Deep Learning: Adaptive Computation and Machine Learning series. The MIT Press, 2016.

5) L. Joshua, Towards Neutron Transformation Searches. PhD Thsis, Barrow U., Tennessee, Knoxville, 2021.

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Jiří Franc, Ph.D.
Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, ČVUT v Praze, Trojanova 339/13, 120 00 Praha 2

Jméno a pracoviště konzultanta:

Mgr. Viktor Pěč, PhD.
Fyzikální ústav AV ČR, v. v. i., Na Slovance 1999/2, 182 21 Praha 8

Datum zadání bakalářské práce: 31.10.2022

Datum odevzdání bakalářské práce: 2.8.2023

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 31.10.2022

.....
garant oboru

.....
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta jaderná a fyzikálně inženýrská -3-

.....
děkan

Abstract

The search for proton decay has long been a cornerstone of particle physics, as its observation could offer evidence for the Grand unification theories (GUTs). Despite decades of experimental efforts, no evidence of proton decay has been found. Nevertheless, updated proton lifetime limits have helped to rule out some Grand unification models and constrain others, making the ongoing search highly relevant. To further the search, the new underground detectors with longer exposure time, improved sensitivity to certain proton decay channels, and stronger background suppression are needed. One of the leading projects in the field is the the Deep Underground Neutrino Experiment (DUNE), an international scientific collaboration and a future neutrino observatory. The DUNE's Liquid Argon Time Projection Chamber (LArTPC) far detector will facilitate the search.

In recent years, machine learning has emerged as a valuable tool at different stages of high-energy particle physics research. Deep neural networks, in particular, possess significant potential to improve detection sensitivity.

Several proton decay channels are predicted by GUTs with $p \to e^+\pi^0$ and $p \to K^+\bar{\nu}$ being dominant. In this work, we focus on the latter. We employ two ensemble convolutional neural network models with a transfer learning approach to distinguish between the simulated proton decay and background interactions of atmospheric neutrinos on argon. Our late fusion model, combining the three modified ResNet18 model outputs through a gating mechanism, demonstrates excellent performance in terms of the ROC AUC and the PRC AUC. Conversely, our early fusion model using EfficientNet B2 with spatial inputs from LArTPC readout planes only slightly improves the modified ResNet18 model performance.

Key words: proton decay, convolutional neural network, residual neural network, ensemble technique, the Deep Underground Neutrino Experiment, LArTPC.

Abstrakt

Pozorování rozpadu protonu by mohlo naznačovat platnost teorií velkého sjednocení (GUTs). Navzdory desetiletím experimentálního úsilí, žádné známky rozpadu protonu nebyly pozorovány. Nicméně aktualizované limity životnosti protonu pomohly vyloučit některé modely velkého sjednocení a omezit jiné. Pro další výzkum jsou zapotřebí nové podzemní detektory s delší dobou expozice, zlepšenou citlivostí na určité kanály rozpadu protonu a silnějším potlačením pozadí. Jedním z předních projektů v této oblasti je DUNE, mezinárodní vědecká kolaborace a budoucí neutrinová observatoř. Vzdálený detektor s kapalným argonem (DUNE LArTPC far detector (FD)) poslouží k detekci rozpadu protonu.

V posledních letech se strojové učení stalo cenným nástrojem ve různých fázích výzkumu ve fyzice vysokých energií. Zejména hluboké neuronové sítě mají významný potenciál pro zlepšení citlivosti detekce.

GUTs předpovídají několik kanálů rozpadu protonu, přičemž dominují $p \to e^+\pi^0$ a $p \to K^+\bar{\nu}$. V této práci se zaměřujeme na druhý zmíněný. Využíváme dvou architektur konvolučních neuronových sítí a přístupu *transfer learning* pro klasifikaci simulovaných vzorků rozpadu protonu a pozadí tvořeného interakcemi atmosférických neutrin s atomy argonu. Náš *late fusion* model, kombinující výstupy tří modelů založených na známé architektuře ResNet18 prostřednictvím tzv. *gate* metody, dosahuje z hlediska ROC a PRC charakteristik vynikajících výsledků. Naopak, naše *early fusion* architektura používající EfficientNet B2 s prostorovými vstupy z vyčítacích rovin LArTPC poskytuje pouze mírné zlepšení výsledků ve srovnání s modifikovanou ResNet18.

Contents

Abbreviations

APA anode plane assembly.

BDTs boosted decision trees.

BSM beyond the Standard Model.

CNNs convolutional neural networks.

DUNE the Deep Underground Neutrino Experiment.

FD far detector.

FSIs final state interactions.

GENIE Generates Events for Neutrino Interaction Experiments.

GUTs Grand unification theories.

HEP high-energy particle physics.

LAr liquid argon.

LArTPC Liquid Argon Time Projection Chamber.

MC Monte Carlo method.

ML machine learning.

MLPs multilayer perceptrons.

ND near detector.

PRC precision-recall curve.

PRC AUC area under the PRC curve.

ROC receiver operating characteristic.

ROC AUC area under the ROC curve.

ROI region of interest.

SGD stochastic gradient descent.

SM Standard Model of particle physics.

SP single-phase.

SURF Sanford Underground Research Facility.

TL transfer learning.

TPC time projection chamber.

Chapter 1

Introduction

It is believed that in the early universe, mere moments after the Big Bang, there was a balance between the masses and charges of particles and their respective antiparticles. Contemporary observations indicate a dominance of matter over antimatter by nine orders of magnitude, as highlighted in [1]. Understanding this asymmetry could hold important clues to beyond the Standard Model physics.

The conservation of baryon number in interactions of elementary particles is a convenient and natural symmetry accounting for the stability of ordinary matter. Under the baryon number conservation principle, the proton, the lightest baryon, would be inherently stable as long as baryon number conservation holds. However, no explicit constraint to baryon number non-conservation is known.

The concept of proton decay, forbidden in the Standard Model, was first proposed by Andrei Sakharov in 1967 [2], introducing the notion of baryon number violation. Since then, the theory received considerable attention within the high-energy particle physics (HEP) domain.

1.1 Standard Model of Particle Physics and Beyond

The Standard Model of particle physics (SM) describes all known elementary particles and their interactions through electromagnetic, weak, and strong forces. The elementary particles can be divided into fermions and bosons, with every fermion having a corresponding antifermion with the same properties except for the opposite electric charge. Fermions can be subdivided into quarks and leptons, constituting all known matter. Quarks are only found in bound states within the composite particles, hadrons. There are three types of hadrons: baryons consisting of three quarks, antibaryons consisting of three antiquarks, and mesons consisting of one quark and one antiquark. The baryon number $\mathscr{B}$ is defined by expression

$$\mathscr{B} = \frac{1}{3}(n_q - n_{\overline{q}}) \tag{1.1}$$

where n_q is the number of quarks and $n_{\overline{q}}$ is the number of antiquarks. According to Weyl, Stuckenberg, and Wigner, the baryon number is conserved in all interactions in the Standard Model [3]. In the early 1970s, various beyond the Standard Model (BSM) theories, such as Grand unification, gained great research interest.

According to Grand unification theories (GUTs), the weak, the strong, and electro-

magnetic forces are merged into a single unified force at $\approx 10^{16}$ GeV. However, particle accelerators cannot directly produce particles with masses at this energy scale. Nonetheless, there is a more feasible approach. In GUTs, the baryon number is not conserved, and protons are unstable, with finite yet extremely long lifetimes of $10^{30} - 10^{36}$ years [3]. This range is directly accessible for future experiments [4]. As such, the search for proton decay is a crucial test for various GUTs models and will provide a better understanding of the nature of matter, regardless of whether the decay itself is observed.

1.2 Brief Overview of Proton Decay Search

Initial efforts to detect proton decay with first-generation underground experiments, including IMB, Soudan, Kamiokande, and Fréjus, did not yield the anticipated results; nevertheless, lower limits on proton lifetime for different decay channels were determined. The need for larger detectors with longer exposure time was evident.

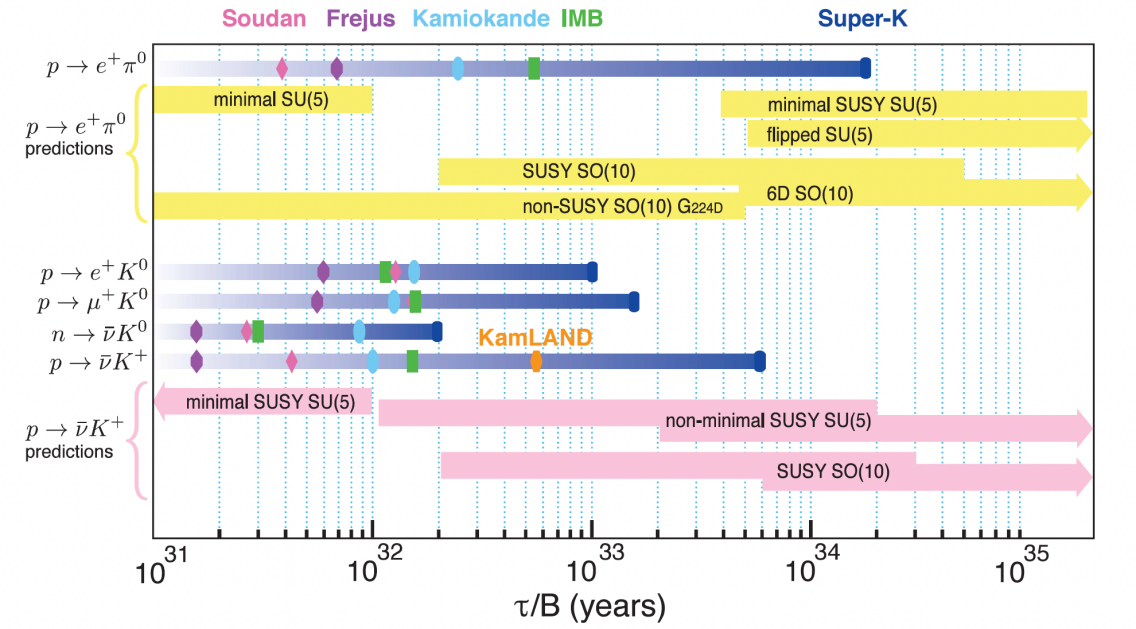


Figure 1.1: Summary of nucleon decay experimental lifetime limits from past and ongoing experiments for several nucleon decay modes, and the model predictions for the lifetimes in the domaninat proton decay modes, $p \to e^+\pi^0$ and $p \to K^+\bar{\nu}$. The limits displayed are 90% confidence level lower limits on the partial lifetimes, here denoted τ/B [5].

Consequently, second-generation detectors, such as Super-Kamiokande and Soudan 2, were launched. While these also failed to find evidence for proton decay, the Super-Kamiokande has put the most stringent limits to date on the proton partial lifetimes.

The next-generation detectors at DUNE, Hyper-Kamiokande, and JUNO will enable probing proton lifetimes up to $10^{33} - 10^{34}$ [4], which is well within the range predicted by GUTs. With its exceptional event imaging, particle identification, and calorimetric capabilities, the DUNE LArTPC FD is poised to be a powerful instrument in the search for rare processes. Many nucleon decay modes are accessible to DUNE. Among them is proton decay via $p \to \bar{\nu}K^+$, favored by many GUTs models [6].

Chapter 2

The Deep Underground Neutrino Experiment

The Deep Underground Neutrino Experiment, hosted by the Fermi National Accelerator Laboratory (Fermilab), is a state-of-the-art neutrino observatory and nucleon decay experiment under construction. The experiment will comprise a neutrino beam and two particle detectors, as shown in Figure 2.1. DUNE unites an effort of more than 1000 scientists from over 30 countries striving to explore the enigma surrounding neutrinos.

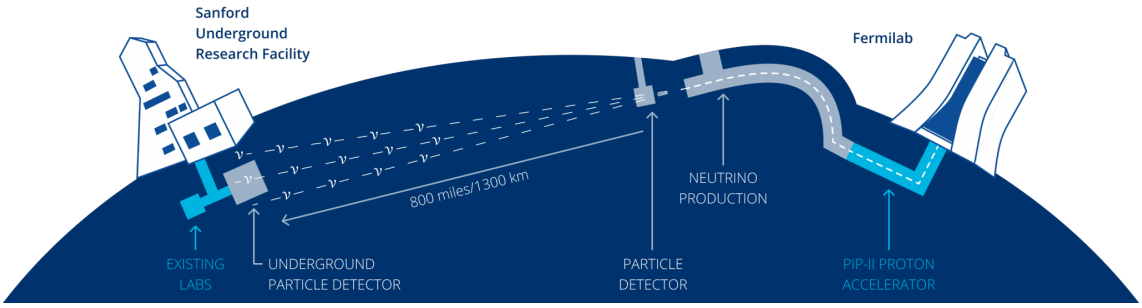


Figure 2.1: Schematic of the Deep Underground Neutrino Experiment. The accelerator complex and near detector are hosted at Fermilab, and the far detector is hosted at the Sanford Underground Research Facility [7].

The experiment will use the world's most powerful neutrino beam to send neutrinos over 1300 km from Fermilab in Illinois to the Sanford Underground Research Facility (SURF) in South Dakota to examine interactions between matter and neutrinos. Understanding the changes particles undergo on their journey through earth will give physicists substantial insights into the history of the universe.

2.1 Key Goals of the DUNE Science Program

The main scientific goals of DUNE are precision measurements of neutrino oscillations to determine the violation of charge-parity symmetry and the ordering of neutrino masses, as well as the detection of core-collapse supernova neutrinos and nucleon decay searches to advance BSM physics [5].

2.1.1 Nucleon Decay Searches

Baryon number conservation implies proton stability. However, the conservation of baryon number, while a feature of the SM, lacks a fundamental symmetry to mandate it. Thus, phenomena like nucleon decay or neutron-antineutron oscillation, violating baryon number conservation, could unveil new physics. These ideas led to the development of large-scale underground detectors (for a brief review, see Section 1.1).

The DUNE FD, with the largest active volume of liquid argon (LAr), will be highly sensitive to several possible nucleon decay modes. The LArTPC technology is particularly advantageous in tracking and identifying kaons. The entire decay chain for nucleon decays into charged kaons, e.g., $p \to K^+\overline{\nu}$, can be observed in LArTPC provided that the kaon is reconstructed within the appropriate energy range and the kaon decay mode is known.

Moreover, DUNE FD scientific program is not limited to kaon-related decay modes. The studies also include proton decay via $p \to e^+\pi^0$ and neutron decay into a charged lepton and a meson.

The Monte Carlo method (MC) simulations of nucleon decays are being conducted. In this thesis, we focus on proton decay via $p \to K^+\overline{\nu}$. For more details on proton decay simulations, see Section 2.3.

2.2 The DUNE Detectors

The DUNE near detector (ND), located 574 m downstream from the neutrino source, will serve as an experiment control system that will measure the energy spectra ν_μ and ν_e before any oscillation takes place. The ND will conduct neutrino on argon interaction measurements to minimize systematic uncertainties of the FD observations. The FD will be installed approximately 1.5 km underground. It will comprise four 10 kt independent LArTPC detector modules, each contained within a cryostat. Currently, the single-phase (SP) technology is considered.

2.2.1 Single-Phase Far Detector Technology

In a SP LArTPC, a volume of LAr medium is subject to a drift field of 500 V/cm [5]. As charged particles traverse the detector, they ionize the argon atoms; the ionization electrons drift horizontally toward the wall of anode plane assembly (APA) units. The APAs comprise three layers of wires, with two of them strung at 37.5° angles (with respect to the vertical) to form a readout grid. The topmost APA wire layer is strung vertically. As drifting electrons cross the grid, they induce a bipolar signal. Eventually, the final layer collects the electrons, resulting in a unipolar signal. The process is illustrated in Figure 2.2.

At the same time, the charged particles emit the scintillation light, which arrives at photon detectors nanoseconds later. Position in the drift direction can be reconstructed by comparing the time it takes for the ionization charge to be collected on the anode and the scintillation light detection time.

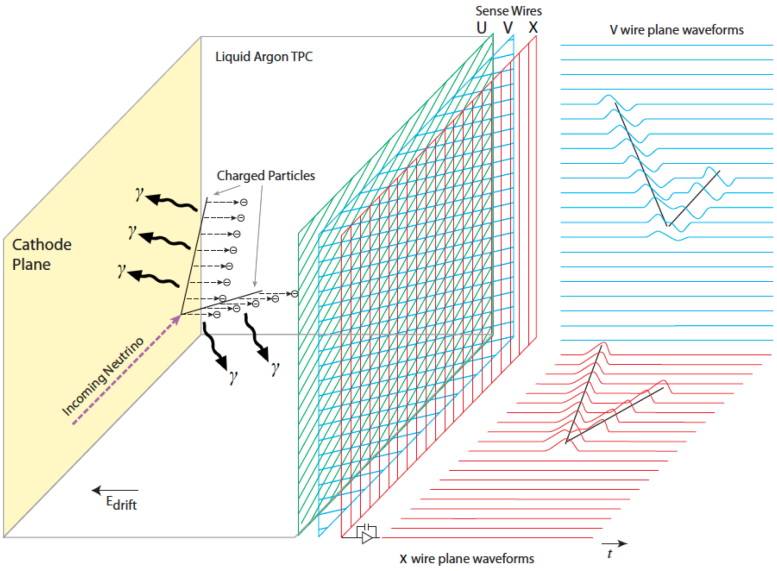


Figure 2.2: The general operating principle of the SP LArTPC. Negatively charged ionization electrons from the neutrino interaction drift horizontally, opposite to the electric field E_{drift} in the LAr and are collected on the anode made of the U, V and X sense wires. The right-hand side represents the time projections in the two dimensions as the event occurs [5]

2.2.2 The DUNE Near Detector

Located 574 meters downstream from the neutrino source, the DUNE ND will measure the initial composition and energy of the neutrino beam, thus improving the calibration and interpretation of observations at the FD. The ND will also be instrumental in studying neutrino-argon interactions in both liquid and gaseous states.

The ND's independent physics program complements that of the FD. It will focus on electroweak physics, quantum chromodynamics, and the investigation of rare BSM processes and exotic particles.

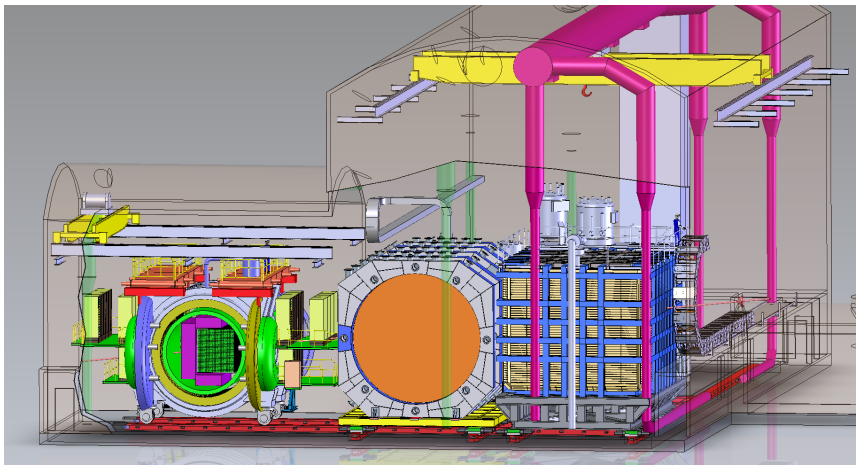


Figure 2.3: Illustration of the ND components. Neutrino beam enters the hall from right to left. The ND LArTPC (right) is the most upstream component; immediately downstream of it (center) is the MPD, and beyond (left) is the SAND [8].

Figure 2.3 illustrates the ND's layout, featuring the ND LArTPC, as its most upstream component. Following it is the Multi-Purpose Detector (MPD) and the System for on-Axis Neutrino Detection (SAND) positioned further downstream. SAND, always on-axis, acts as a neutrino spectrum monitor. A unique aspect of the ND is the ability of the ND LArTPC and a High-Pressure Gaseous Argon time projection chamber (TPC) (HPgTPC) to shift off-axis relative to the beam. This allows access to varied neutrino energy spectra, offering an additional degree of freedom to the measurement. The design and functionality of the ND LArTPC overlap with that of the FD, which reduces the impact of nuclear effects and detector-driven systematic uncertainties in oscillation signal extraction at the FD.

2.3 Signal and Background Simulation Methodology

In DUNE FD, several nucleon decay modes will be possible to observe. Nucleon decays involving kaons are particularly advantageous since the entire decay chain can be observed for them. Proton decay, the signal, via $p \to K^+\bar{\nu}$ can be tagged in a LArTPC if the kinematics and the decay of a single kaon with a sufficient energy can be reconstructed [4]. The background events are primarily interactions of cosmic neutrinos on argon nuclei.

ID	Decay channel(s)	ID	Decay channel(s)	ID	Decay channel(s)
Antilepton + Meson		Antilepton + Mesons		Antilepton + Photon(s)	
1	$p \to e^+\pi^0$, $n \to e^+\pi^-$	23	$p \to e^+\pi^+\pi^-$	42	$p \to e^+\gamma$
2	$p \to \mu^+\pi^0$, $n \to \mu^+\pi^-$	24	$p \to e^+\pi^0\pi^0$	43	$p \to \mu^+\gamma$
3	$p \to \bar{\nu}\pi^+$, $n \to \bar{\nu}\pi^0$	25	$n \to e^+\pi^-\pi^0$	44	$n \to \bar{\nu}\gamma$
4	$p \to e^+\eta$	26	$p \to \mu^+\pi^+\pi^-$	45	$p \to e^+\gamma\gamma$
5	$p \to \mu^+\eta$	27	$p \to \mu^+\pi^0\pi^0$	46	$n \to \bar{\nu}\gamma\gamma$
6	$n \to \bar{\nu}\eta$	28	$n \to \mu^+\pi^-\pi^0$	Three or more leptons	
7	$p \to e^+\rho^0$, $n \to e^+\rho^-$	29	$n \to e^+\pi^- K^0$	49	$p \to e^+e^+e^-$
8	$p \to \mu^+\rho^0$, $n \to \mu^+\rho^-$	Lepton + Meson		50	$p \to e^+\mu^+\mu^-$
9	$p \to \bar{\nu}\rho^+$, $n \to \bar{\nu}\rho^0$	30	$n \to e^-\pi^+$	51	$p \to e^+\bar{\nu}\nu$
10	$p \to e^+\omega$	31	$n \to \mu^-\pi^+$	52	$n \to e^+e^-\bar{\nu}$
11	$p \to \mu^+\omega$	32	$n \to e^-\rho^+$	53	$n \to \mu^+e^-\bar{\nu}$
12	$n \to \bar{\nu}\omega$	33	$n \to \mu^-\rho^+$	54	$n \to \mu^+\mu^-\bar{\nu}$
13	$p \to e^+K^0$, $n \to e^+K^-$	34	$n \to e^-K^+$	55	$n \to \mu^+e^+e^-$
14	$p \to e^+K^0_S$	35	$n \to \mu^-K^+$	56	$n \to \mu^+\mu^+\mu^-$
15	$p \to e^+K^0_L$	Lepton + Mesons		57	$p \to \mu^+\bar{\nu}\nu$
16	$p \to \mu^+K^0$, $n \to \mu^+ + K^-$	36	$p \to e^-\pi^+\pi^+$	58	$p \to e^-\mu^+\mu^+$
17	$p \to \mu^+K^0_S$	37	$n \to e^-\pi^+\pi^0$	59	$n \to \bar{\nu}\bar{\nu}\nu$
18	$p \to \mu^+K^0_L$	38	$p \to \mu^-\pi^+\pi^+$	60	$n \to \bar{\nu}\bar{\nu}\bar{\nu}\nu\nu$
19	$p \to \bar{\nu}K^+$, $n \to \bar{\nu}K^0$	39	$n \to \mu^-\pi^+\pi^0$		
20	$n \to \bar{\nu}K^0_S$	40	$p \to e^-\pi^+K^+$		
21	$p \to e^+K^{\star 0}$	41	$p \to \mu^-\pi^+K^+$		
22	$p \to \bar{\nu}K^{\star +}$, $n \to \bar{\nu}K^{\star 0}$				

Figure 2.4: Nucleon decay modes available in Generates Events for Neutrino Interaction Experiments (GENIE) simulations [4, 9].

2.3.1 Background Simulations

For an accurate simulation of the background in the DUNE FD, it is essential to understand the source and behavior of atmospheric neutrinos forming a substantial part of the background in nucleon decay searches. Atmospheric neutrinos are produced when cosmic rays, mainly protons and heavier nuclei, collide with atoms in the Earth's atmosphere. This interaction triggers a cascade of secondary particles, particularly pions and kaons, which decay into neutrinos due to their short-lived nature. In DUNE, the Bartol model is used to model the neutrino flux. The Bartol model calculates the yield of neutrinos from the decay of charged pions and kaons, factoring in the energy and type of the primary cosmic ray, atmospheric composition, and density, as well as geomagnetic effects that impact the trajectory of cosmic ray particles. A critical aspect of the Bartol model lies in its ability to forecast the energy spectrum and angular distribution of neutrinos. It differentiates between various types of neutrinos (muon neutrinos, electron neutrinos, and their antiparticles) across a wide range of energies.

Atmospheric neutrino interactions on argon are modeled using the GENIE simulation framework [9]. The interaction cross-section, measured in area units, expresses the probability of neutrino-argon interaction. To estimate the total event rate in the detector, the expected number of interactions per unit of time and volume is obtained by multiplying the neutrino flux by the interaction cross-section. This product is then integrated over the relevant energy range and volume.

2.3.2 Signal Simulations

The GENIE toolkit is employed for nucleon decay signal simulation (for proton decay modes available, see Figure 2.4), albeit with adaptations and extensions beyond its primary use for neutrino interaction simulations [4]. This involves incorporating theoretical models that predict different nucleon decay modes, each characterized by specific final-state particles and branching ratios. GENIE employs MC techniques to create statistically significant samples of hypothetical decay events.

Considering final state interactions (FSIs) becomes crucial in nucleon decay event simulations. Those are the interactions occurring between the production and detection of the initial nucleon decay products. These interactions can significantly alter the observable properties of the decay products, such as their energy and momentum distributions (see Figure 2.5). For instance, a kaon produced from a nucleon decay, say $p \rightarrow K^+\bar{\nu}$, may interact with other nucleons or particles in the surrounding medium, losing a part of its kinetic energy in each interaction. In GENIE, FSIs are modeled to account for the various processes accompanying decays.

According to [4], the tracking efficiency for kaons is 58%, which implies that only 58% of the simulated kaons can be reconstructed as a track in the detector. The tracking efficiency loss is mainly attributed to low energy (< 40 MeV) kaons, resulting in tracks of < 4 cm.

2.3.3 Discriminating the Signal from the Background in Particle Identification

In particle physics experiments, particularly those involving neutrinos, the challenge of accurately identifying particles is paramount. As particles traverse the detector medium,

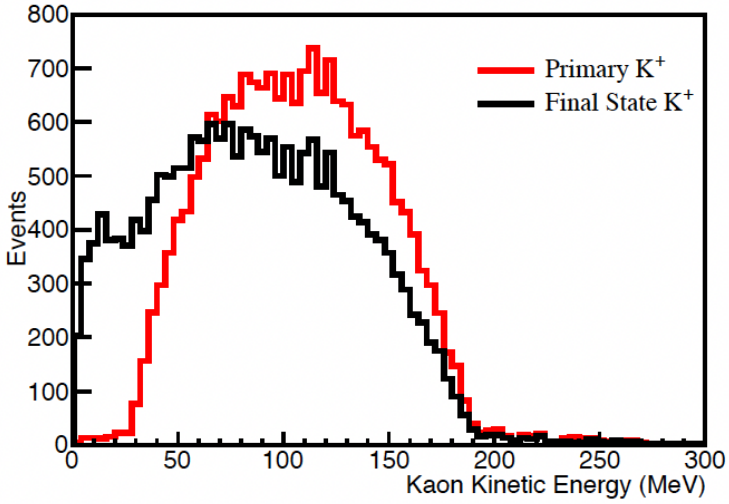
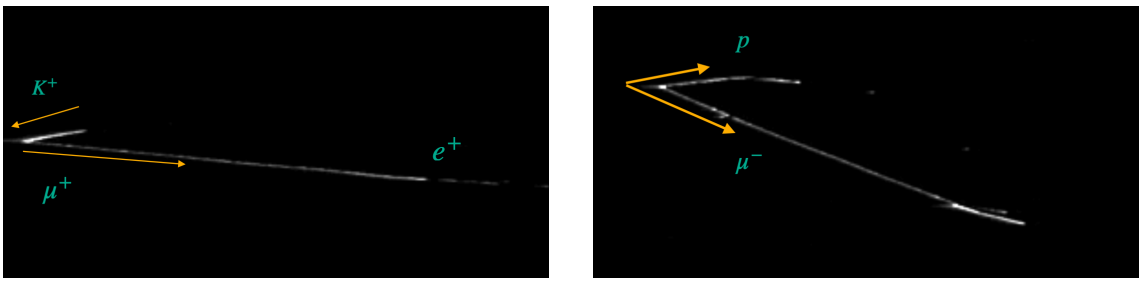


Figure 2.5: A comparison of the kinetic energy of kaons in simulated proton decay via $p \to K^+\bar{\nu}$ before and after FSIs occur [4].

they lose energy primarily through ionization and scintillation. An intuitive method for particle identification relies on analyzing the energy loss, i.e., the stopping power, denoted as $\mathrm{d}E/\mathrm{d}x$, which describes the energy loss of a particle per unit distance. Different particles exhibit distinct $\mathrm{d}E/\mathrm{d}x$ profiles. For instance, heavier particles like protons have a higher stopping power compared to lighter particles like muons or electrons. The Particle IDentification Algorithm (PIDA) [4] leverages this information by combining the $\mathrm{d}E/\mathrm{d}x$ data and particle track length to improve the accuracy of particle identification.

A typical background in $p \to K^+\bar{\nu}$ simulations stems from atmospheric neutrino interactions with argon, namely, $\nu_\mu n \to \mu^- p$. Challenges in discrimination arise when the muon's momentum mimics that from a $K^+ \to \mu^+\nu_\mu$ decay at rest, compounded by potential misreconstruction of the proton as a kaon (for an illustration, see Figure 2.6) [6].



(a) (b)

Figure 2.6: An example event displays for two commonly misidentified events: 2.6a the proton decay via $p \to K^+\bar{\nu}$, with the kaon decaying into a positively charged muon μ^+ and a muon neutrino ν_μ and 2.6b a charged current quasi-elastic scatter of muon neutrino on a neutron, i.e., $\nu_\mu n \to \mu^- p$, the prevalent background process.

To address this, a log-likelihood ratio is utilized as a probabilistic discriminator. The method takes advantage of spatial characteristics and energy deposition patterns unique to each reaction. For instance, near the kaon-muon vertex, where the K^+ decays,

the kaon exhibits a higher ionization density due to its residual kinetic energy deposition. The local increase in ionization is a key attribute used to distinguish between signal and background events.

the method involves calculating two log-likelihood ratios along each particle track: a backward ratio, commencing at the hadron-muon vertex, and a forward ratio, starting from the opposite end. These are defined as follows

$$\mathcal{L}_{\text{direction}} = \sum_i \log \frac{p_i^{\text{sig}}}{p_i^{\text{bkg}}}, \tag{2.1}$$

where the direction is either forward or backward, and p_i^{sig} and p_i^{bkg} denote the probability densities of the i-th event being a signal or background, respectively. The combined sum, $\mathcal{L} = \mathcal{L}_{\text{fwd}} + \mathcal{L}_{\text{bkwd}}$, yields a more robust discrimination between signal and background [4].

2.4 The Prospects of Machine Learning in HEP

Two machine learning (ML) approaches are typically used in HEP: the classical methods, such as boosted decision trees (BDTs) or multilayer perceptrons (MLPs), and image-based algorithms. The former are widely used for the event reconstruction and feature engineering, while the latter are used for classification [10, 11].

The challenges in ML and HEP are largely similar. Particularly, when it comes to feature engineering, the core challenge lies in effectively extracting and interpreting complex patterns from large datasets in both fields. This similarity highlights the potential of machine learning methods, particularly convolutional neural networks (CNNs), in HEP.

HEP experiments conducted at particle accelerators and neutrino observatories generate vast datasets, with the events of interest being extremely rare compared to the background interactions. Similarly, in ML, one deals with extracting relevant patterns from large, often unstructured datasets.

CNNs excel in this domain due to their hierarchical structure, which allows for automatic and adaptive feature extraction. In image processing, CNNs have shown remarkable success. They identify patterns and structures at various levels of complexity, from simple edges and dots to complex, composite shapes. This is applicable and potentially advantageous in HEP experiments, where particle data collected by detectors can often be represented as images or image-like structures.

By applying CNNs to grid-like data, one can automate the feature extraction process, identifying complex patterns that may not be detectable with traditional methods. In nucleon decay searches, where the signature of decay events is subtle, traditional methods like BDTs have been effective [4]. However, they rely heavily on predefined features and might only capture some relevant information. CNNs, on the other hand, can learn to identify features directly from the data, potentially revealing new and significant patterns that could improve event classification.

Chapter 3

An Overview of Machine Learning Techniques

3.1 A Conceptual Introduction to CNNs

Deep neural networks have achieved outstanding performance. The CNNs emerged during the second wave of neural network popularity in the 1980s when researchers began experimenting with networks for computer vision and speech recognition. However, due to memory and computational resource limitations, fully connected were not feasible. An innovative solution was required to address these challenges. Like conventional MLPs, CNNs draw inspiration from neuroscience. By utilizing convolutional layers, CNNs can model the characteristics of the mammalian visual cortex. Thus, they excel at capturing hierarchical and spatial dependencies within data that exhibits a grid-like topology.

3.1.1 Convolution Operation

the convolution operation is defined for real-valued functions. Intuitively, it describes the response of a linear time-invariant system to an input stimulus. In the continuous case, it is defined for the real-valued function f and g as follows

$$(f * g)(t) = \int_{\mathbf{R}} f(u)g(t-u)\mathrm{d}u \tag{3.1}$$

if the integral exists. For each t, it can be described as the area under the function $f(u)$ weighted by the function $g(-u)$. As the t changes, $g(t-u)$ emphasizes different parts of the input function $f(u)$. The discrete convolution can be defined similarly.

It is worth noting that the convolution operation used in the context of neural networks differs from its mathematical definition. Moreover, a single convolution kernel can only extract one feature type across different spatial locations. To extract multiple feature types per layer, convolution is applied repeatedly. Although the inputs exhibit a grid topology, they are not exactly a grid of scalars but a grid of vector values. The inputs to the first layer of the network are usually multichannel, such as RGB images, fed into the network in batches. The inputs to the intermediate network layer are the outputs of the preceding one, resulting in four-dimensional data tensors. Three dimensions correspond to the RGB values, and the fourth dimension is used to index the inputs

in batches. In neural networks that use multichannel convolutions, commutativity is guaranteed only if each operation has the same number of input and output channels. Consider a 4D kernel tensor $\mathbf{K}$ with element $K_{i,j,k,l}$ and input data $\mathbf{V}$ with element $V_{i,j,k}$. Let the output $\mathbf{Z}$ be the result of convolving the input with the kernel. To sample every s pixels in each direction in the output, where s is the stride (or a step size of the convolutional filter), a downsampled convolution is performed:

$$Z_{i,j,k} = c(\mathbf{K}, \mathbf{V}, s)_{i,j,k} = \sum_{l,m,n} V_{l,(j-1)s+m,(k-1)s+n} K_{i,l,m,n}. \tag{3.2}$$

3.1.2 Motivation

In traditional neural network layers, all input and output neurons are interconnected. That is represented by a parameter matrix, where each element corresponds to the weight of the connection between specific input and output neurons. The fully connected structure implies that every output unit is influenced by every input unit, resulting in a dense matrix of weights.

In contrast, CNNs exhibit sparse interactions. In the context of CNNs, the term *sparse* denotes that each output unit is linked to a limited number of input units rather than all of them. The sparse connectivity is accomplished using a kernel smaller than the input.

As a filter is convolved across the input, the basic features, such as edges and dots, are detected. This results in a reduced number of operations required to obtain the output, thereby leading to lower requirements for parameter storage and memory usage of the model.

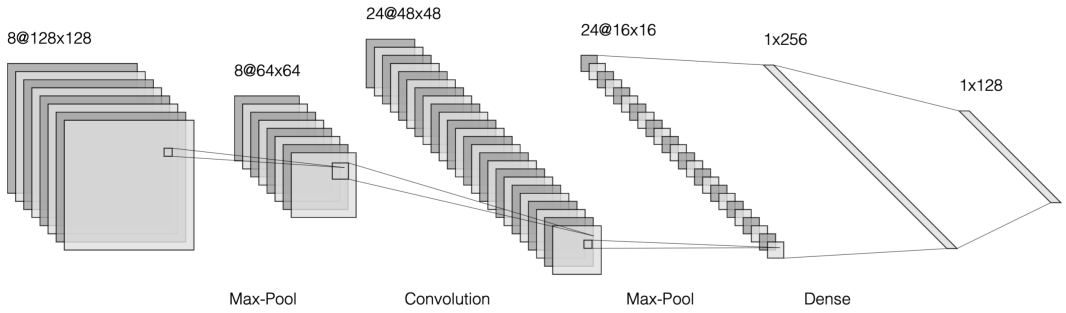


Figure 3.1: An illustration of a CNN architecture.

In deep CNNs, units in deeper layers may indirectly interact with a larger portion of the input. This allows the network to capture local patterns and spatial hierarchies efficiently [12]. Unlike in traditional neural networks, in CNNs, the parameters can be used for multiple functions in a model. Each element of the kernel is used at every position of the input. The parameter sharing used by the convolution operation means that instead of learning unique parameters for each location, a single set of parameters is learned. Moreover, parameter sharing implies translational equivariance of the layer.

3.1.3 Training CNNs

The CNNs are trained by backpropagation with gradient descent. During the backpropagation process in a neural network, the gradient of the loss function with respect to

the kernel is computed using the gradient with respect to outputs. This allows the network to learn and improve its performance over time. Consider a convolutional network $c(\mathbf{K}, \mathbf{V}, s)$ with a kernel tensor $\mathbf{K}$, the input tensor $\mathbf{V}$ and a kernel stride s. The objective is to minimize an arbitrary loss function $J(\mathbf{V}, \mathbf{K})$. In the forward pass, the input $\mathbf{V}$ is fed into the neural network to produce an output $\mathbf{Z}$, which is used to compute the loss J. During the backpropagation, a tensor $\mathbf{G}$ with elements $G_{i,j,k} = \frac{\partial}{\partial Z_{i,j,k}} J(\mathbf{V}, \mathbf{K})$ is obtained. To train the network, the derivatives of J with respect to each kernel weight are computed as follows [12]

$$\frac{\partial J(\mathbf{V}, \mathbf{K})}{\partial K_{i,j,k,l}} = \sum_{p,q,r} \frac{\partial J}{\partial Z_{p,q,r}} \frac{\partial Z_{p,q,r}}{\partial K_{i,j,k,l}} = \sum_{m,n} G_{i,m,n} V_{j,(m-1)s+k,(n-1)s+l}. \tag{3.3}$$

To continue propagating the error through the network's intermediate layers, the gradient of the loss with respect to $\mathbf{V}$ is computed. Since

$$\frac{\partial J(\mathbf{V}, \mathbf{K})}{\partial V_{i,j,k}} = \sum_{p,q,r} \frac{\partial J}{\partial Z_{p,q,r}} \frac{\partial Z_{p,q,r}}{\partial V_{i,j,k}}, \tag{3.4}$$

and the output for a given convolutional layer is computed as in Equation 3.2,

$$\frac{\partial J(\mathbf{V}, \mathbf{K})}{\partial V_{i,j,k}} = \sum_{\{l,m:(l-1)s+m=j\}} \sum_{\{n,p:(n-1)s+p=k\}} \sum_{q} K_{q,i,m,p} G_{q,l,n}. \tag{3.5}$$

3.2 Residual Networks

The concept of residual learning was proposed to address the challenge of network depth and its influence on performance [13]. The issue is the vanishing, exploding, or shattering (instability) of gradients [14], which hinder the convergence of the network in the beginning. Deep residual networks (ResNets) introduced in [13] are modular architectures comprising many blocks, commonly known as residual units. The original residual unit is defined as

$$\mathbf{y}_i = \mathcal{H}(\mathbf{x}_i) + \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i), \quad \mathbf{x}_{i+1} = f(\mathbf{y}_i). \tag{3.6}$$

Here, $\mathbf{x}_i$ denotes the input feature to the i-th residual unit, $\mathcal{W}_i = \{W_{i,j} : 1 \leq j \leq K\}$ is a set of weights and biases associated with the i-th residual unit and K is the number of layers in residual unit. The residual function $\mathcal{F}$ may take various forms; for instance, in the original ResNet model [13], a stack of two 3×3 convolutional layers is used. The function f is an activation function, e.g., the rectified linear unit (ReLU) $f(\mathbf{y}_i) = \max(0, \mathbf{y}_i)$, or the hyperbolic tangent $f(\mathbf{y}_i) = \tanh(\mathbf{y}_i)$. The function $\mathcal{H}$ is originally an identity mapping, e.i., $\mathcal{H}(\mathbf{x}_i) = \mathbf{x}_i$.

To better understand the ResNets concept, let us consider a simplified scenario where $f = \mathrm{id}$, which implies that $\mathbf{x}_{i+1} = \mathbf{y}_i$ in Equation 3.6. Then, for an arbitrary unit L that is deeper than the unit i

$$\mathbf{x}_L = \mathbf{x}_i + \sum_{k=1}^{L-1} \mathcal{F}(\mathbf{x}_k, \mathcal{W}_k). \tag{3.7}$$

That states the feature $\mathbf{x}_L$ of a deeper unit can be represented as a feature $\mathbf{x}_i$ of a shallower unit and a residual function $\sum_{k=1}^{L-1} \mathcal{F}$. That means the feature of a deeper unit is roughly the sum of all preceding residual functions, unlike the plain networks, where a feature is obtained by multiplying the weight matrix and a feature vector. It is worth noting that the shortcut connections do not introduce extra parameters or alter the computational time, assuming that an element-wise addition is negligible in computation [13]. The residual networks are trained via backpropagation with gradient descent. We will illustrate it in the case of the identity activation $f = \mathrm{id}$; the chain rule applied to a loss function J holds

$$\frac{\partial J}{\partial \mathbf{x}_i} = \frac{\partial J}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_i} = \frac{\partial J}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_i}\sum_{k=1}^{L-1}\mathcal{F}(\mathbf{x}_k, \mathcal{W}_k)\right). \tag{3.8}$$

Here, the term $\frac{\partial J}{\partial \mathbf{x}_L}$ ensures the information is propagated directly to shallower units. The term $\frac{\partial J}{\partial \mathbf{x}_L}\frac{\partial}{\partial \mathbf{x}_i}\sum_{k=1}^{L-1}\mathcal{F}(\mathbf{x}_k, \mathcal{W}_k)$ propagates through the weight layers. It is worth noting that the expression $\frac{\partial}{\partial \mathbf{x}_i}\sum_{k=1}^{L-1}\mathcal{F}(\mathbf{x}_k, \mathcal{W}_k)$ cannot be equal to -1 for all instances in a mini-batch. As a result, the gradient $\frac{\partial J}{\partial \mathbf{x}_i}$ is highly unlikely to vanish, even if the weights are arbitrarily small. In the case of a non-trivial activation f, the backpropagation formula can be derived similarly, as in Equation 3.8.

3.3 Transfer Learning

Obtaining sufficient data for a particular task can be challenging, if not impossible, due to limited accessibility or the high cost of obtaining and labeling data. Consequently, one often relies on extrapolating knowledge across domains. This philosophy has been the basis for transfer learning (TL), a machine learning approach that seeks to improve the performance of a model on a specific problem by leveraging knowledge from previously solved tasks. TL facilitates learning by establishing connections between past and target tasks, resulting in faster and potentially more precise outcomes.

The increasing availability of large-scale data repositories has made TL an appealing solution for tackling problems in domains where limited data is available. In particular, using existing datasets related to the target domain of interest, though different, can facilitate the development of effective machine learning models.

In data analysis, a domain $\mathcal{D}$ is characterized by a feature space $\mathcal{X}$ and a marginal probability distribution $p(X)$. Here, $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. For a given domain $\mathcal{D}$, a task $\mathcal{T}$ is defined by a label space $\mathcal{Y}$, and a predictive function f that is learned from a set of feature vectors and their corresponding labels $\{(x_i, y_i)\}$. The domain data is then defined as $D = \{(x_i, y_i) \,:\, x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$.

Similarly, we define the source domain data D_S and the target domain data D_T as $D_S = \{(x_{S,i}, y_{S,i}) \,:\, x_{S,i} \in \mathcal{X}_S, y_{S,i} \in \mathcal{Y}_S\}$ and $D_T = \{(x_{T,i}, y_{T,i}) \,:\, x_{T,i} \in \mathcal{X}_T, y_{T,i} \in \mathcal{Y}_T\}$, respectively. Further, we denote the source task as $\mathcal{T}_S$ and the target task as $\mathcal{T}_T$, each with their corresponding predictive functions, f_S and f_T.

TL is a technique that can enhance the accuracy of a target predictive function f_T by utilizing information from a source domain $\mathcal{D}_S$ and corresponding task $\mathcal{T}_S$, which may differ from the target domain $\mathcal{D}_T$ and task $\mathcal{T}_T$. This approach can be used with multiple source domains, in contrast to conventional machine learning where $\mathcal{D}_S$ and $\mathcal{D}_T$, as well

as $\mathcal{T}_S$ and $\mathcal{T}_T$, are identical.

Several cases are possible. Since the source domain $\mathcal{D}_S = \{\mathcal{X}_S, p(X_S)\}$ and the target domain $\mathcal{D}_T = \{\mathcal{X}_T, p(X_T)\}$ are distinct, $\mathcal{D}_S \neq \mathcal{D}_T$ implies unequal feature spaces $\mathcal{X}_S \neq \mathcal{X}_T$ and/or marginal distributions of the inputs $p(X_S) \neq p(X_T)$. In the latter case, a learner trained on a specific source domain may not perform optimally on a target domain.

Another possible scenario is when the label space of the source and target domains, denoted by $\mathcal{Y}_S$ and $\mathcal{Y}_T$, respectively, do not match. The class space mismatch is accompanied by a difference in the conditional probability distribution, represented by $p(Y|X)$. Another possible case is an imbalance in the dataset labels between the source and target domains, leading to $p(Y_S|X_S) \neq p(Y_T|X_T)$. In the upcoming section, we will elaborate on the learning challenges posed by dataset imbalance.

3.4 Learning on an Imbalanced Dataset

As previously mentioned, discrepancies in distribution may arise between the source and target domains. Imbalanced learning poses a significant challenge as it can compromise the performance of most conventional learning algorithms. This is because these typically assume that the class distributions are balanced, which is often not the case in real-world scenarios. Consequently, the algorithms may fail to accurately represent the distributive characteristics of complex datasets, leading to poor generalization.

Technically, any dataset demonstrating unequal class distribution can be considered imbalanced. However, the common understanding is that the term *imbalanced data* corresponds to datasets exhibiting significant or severe class imbalance. The problem is present in both binary and multiclass classification tasks. The choice of a descriptive, suitable metric is therefore crucial.

A well-performing classifier should provide an equal level of predictive accuracy for both the minority and majority classes in a given dataset. On the contrary, it is seen that the classifier tends to provide a severely imbalanced degree of accuracy, excelling on the majority class at the expense of the minority class, which is often the one desired or sought. Therefore, accuracy is not a sufficient measure of goodness, and more informative assessment metrics, such as receiver operating characteristics, precision-recall, and loss curves, are necessary for conclusive evaluation of the learner's performance.

3.4.1 Assessment Metrics for Imbalanced Learning

Assuming a binary classification problem, the performance of a classifier can be represented by a confusion matrix (see Table 3.1).

Table 3.1: Confusion matrix for binary classification.

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

One of the most commonly used metrics to assess the performance of a classifier is

accuracy A, which is defined, using the notation from Table 3.1, as

$$a = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}. \tag{3.9}$$

Accuracy is computed as the ratio of the number of correct predictions (true positives and true negatives) to the total number of predictions made by the classifier. It provides a simple and intuitive way to describe a classifier's performance on a given dataset. Nevertheless, it can be misleading when the class distribution is uneven. In imbalanced datasets, where one class significantly outnumbers the other, relying solely on accuracy may result in an inadequate representation of a model's performance. To better understand the root problem, we can examine a confusion matrix in Table 3.1. The left column displays negative data instances, while the right column represents positive ones. By comparing the counts in both columns, we can determine the class distribution in the dataset. Therefore, metrics that rely on both columns are particularly susceptible to imbalances and shifts in data distribution. This implies that accuracy, as a performance measure, will fluctuate despite the classifier's underlying fundamental performance remaining constant, depending on variations in class distribution.

When assessing the performance of a model on different datasets, inconsistencies can arise, leading to difficulties in analyzing the model's performance. This is especially true when the assessment metrics are sensitive to the data distribution and when data imbalance is present.

Other evaluation metrics are precision P (Equation 3.10), recall R (Equation 3.11), and F-measure F_β (Equation 3.12).

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.10}$$

Precision measures the accuracy of positive classifications. Conversely, recall measures completeness, i.e., how many examples of the positive class are labeled correctly.

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.11}$$

Even though these two metrics, much like accuracy and error rate $ER = 1 - A$, share an inverse relationship, they are not both sensitive to changes in data distributions. Precision is distribution sensitive, while recall is not. However, neither provides information on the number of incorrectly labeled positive or misclassified examples. One can use an F-measure that combines precision and recall to address this.

$$F_\beta = \frac{(1 + \beta)^2 \cdot R \cdot P}{\beta^2 \cdot R + P} \tag{3.12}$$

The F-measure can be interpreted as a measure of classification effectiveness in terms of the ratio of the importance of either recall or precision weighted by the β coefficient. Typically, $\beta = 1$ is used. However, it is important to note that it can be influenced by data distribution, making it less reliable when comparing the performance of models on different datasets. Nonetheless, in most cases, it still proves to be a superior choice compared to accuracy, precision, and recall.

The receiver operating characteristic (ROC) is more optimal in this case. The ROC utilizes two assessment metrics: true positive rate TPR (same as recall) and false positive rate FPR defined as follows

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3.13}$$

By plotting the TPR against the FPR, the ROC curve is generated, and each point in the ROC space represents the effectiveness of a particular classifier on a given data distribution. The ROC curve provides a visual representation of the trade-offs between the benefit and cost of classification with respect to data distributions.

Chapter 4

The Aspects of Data

4.1 On Data Origin and Interpretation

The data comprises MC simulated signal and background samples. The signal is proton decay via $p \to K^+\overline{\nu}$, and neutrino-argon interactions represent the background.

For the signal sample, the simulation includes the modeling of the initial state of the argon nucleus, the decay kinematics of the proton, and the intranuclear propagation of the decay products. The simulation of the atmospheric neutrino interactions encompasses modeling the neutrino flux, the nuclear model, and the propagation of the particles emerging inside the nucleus. The signal and background data was generated during the GENIE [9] runs `54474279` and `54053565`, using the `dune10kt_v4_1x2x6` detector geometry model. The Bodek-Ritchie extension of global relativistic Fermi gas was used. The model extends the global relativistic Fermi gas, which assumes the nucleus to be non-interacting particles, by nucleon-nucleon correlations [15].

The data was subject to a simple preselection filter to discard the atmospheric neutrino interactions whose signatures differ significantly from those of the signal. The filter cut is described as follows:

1. maximum reconstructed track length $\leqslant 100$;
2. $1 \leqslant$ number of reconstructed tracks $\leqslant 6$;
3. number of reconstructed showers $\leqslant 4$;
4. $5 <$ number of reconstructed clusters of hits < 80;
5. $100 <$ number of reconstructed hits < 1200.

Here, hits represent parts of wire signals exceeding a certain threshold, fitted with a Gaussian. They hold the time corresponding to the peak of the Gaussian curve, the amplitude representing the peak's height and thus the charge amount deposited, and the width of the Gaussian, providing insights into the duration of the signal. The hits are grouped into clusters formed by associating hits that are adjacent in both their physical location on the individual readout plane and their timing, as determined by the drift time.

The cut reduced the number of background and signal events by 65% and 10%, respectively. The dataset size was further reduced by a total of 10% by the image processing-based event selection described in this chapter.

It is essential to consider the detector readout system geometry for proper interpretation and processing of the data.

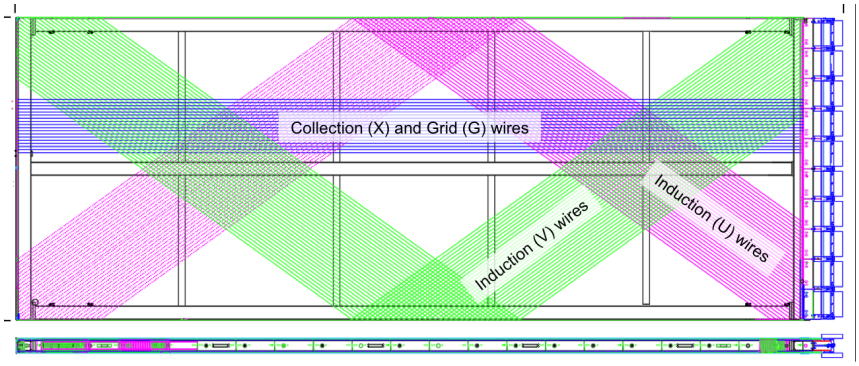


Figure 4.1: Illustration of the DUNE APA wire wrapping scheme showing small portions of the wires from three readout planes (*U, V, X*). The TPC electronics, shown in blue to the right, mount directly to the frame and process signals from both the collection and induction channels [5].

As charged particles ionize the argon atoms, the ionization electrons drift in the detector medium towards the APA wall. The APAs are large, rectangular frames made of stainless steel with a slim profile. Each frame is strung with wires in several layers. The topmost layer consists of shielding or grid wires that shape the electric field and reduce the impact of external electronic noise. Beneath are induction wire layers, placed at angles relative to the vertical to capture drifting electrons from particle interactions. The final layer is the collection plane, where electrons are gathered to create a detectable signal. The DUNE APA wire wrapping scheme is illustrated in Figure 4.1.

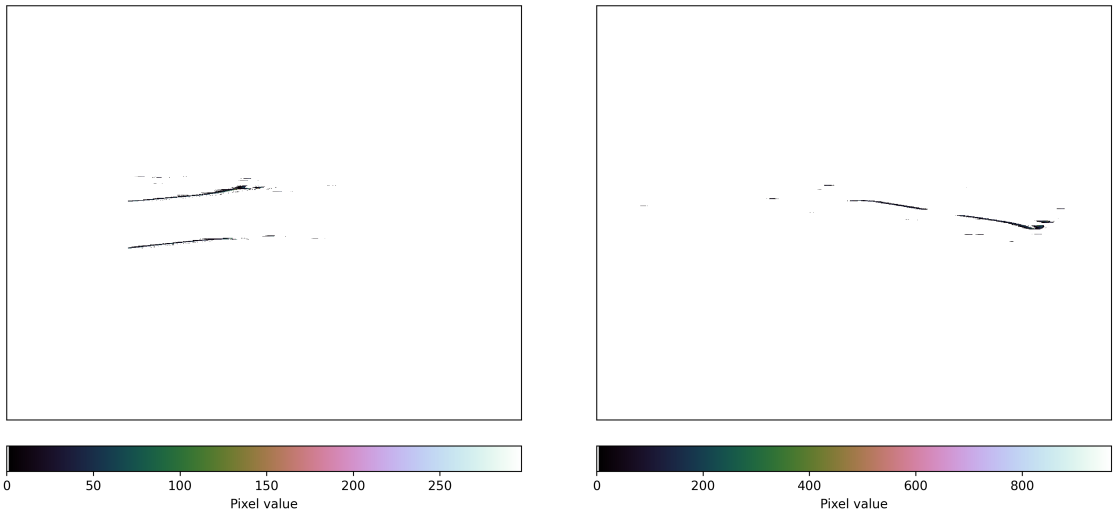
4.1.1 The Data Challenges

The wrapping of wires around the APA frame can lead to the scenario illustrated in Figure 4.2. In this case, a single particle track may be partially captured by the wires on one side of the wire plane and partially on the opposite side. This results in a *double-track* image, where one continuous particle track appears as if it were two separate tracks. When interpreting data from the APA, these instances are crucial to consider since they may deplete the classifier's performance.

4.2 Data Preprocessing

The dataset utilized in this study is formatted as comma-separated value (CSV) files. Each file corresponds to a single event, categorized as either a signal or background. The values in the file are so-called ADC counts. The ADC count is a quantized representation of the amplitude of the analog input signal.

Initially, the data has a linear array format, where the values of individual pixels are sequentially enumerated, spanning a total of N elements. The structure of the CSV files is such that they represent images with dimensions $n \times m$, where n denotes the number of rows, and m is the number of columns in an image.



(a) Induction plane.

(b) Collection plane.

Figure 4.2: An example event display where the track continuation is observed.

Table 4.1: Initial linear data structure. Pixel values are denoted as v_i for $i \in \{0, 1, \ldots, N-1\}$.

Index	Pixel Value
0	v_0
1	v_1
2	v_2
$\vdots$	$\vdots$
$N-1$	v_{N-1}

Table 4.2: Data organized as a matrix corresponding to the image structure; r_i and c_j denote rows and columns of the resulting image, respectively. The total number of pixels is $N = n \times m$.

r \ c	c_0	c_1	c_2	$\ldots$	c_{m-1}
r_0	v_0	v_1	v_2	$\ldots$	v_{m-1}
r_1	v_m	v_{m+1}	v_{m+2}	$\ldots$	v_{2m-1}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
r_{n-1}	$\ldots$	$\ldots$	$\ldots$	$\ldots$	v_{N-1}

A tabular representation of the initial data format is provided for clarity. In Table 4.1, each row corresponds to a pixel index, ranging from 0 to $N-1$, and the associated pixel value is denoted as v_i for each index i within the set $0, 1, \ldots, N-1$. Table 4.2 provides an image interpretation of the initial linear data.

The images are produced in (wire, time) coordinates, where the wire is the number of the wire where the reconstructed hit was detected, and time indicates the duration between the interaction and the detection of the hit on the wire. Each pixel corresponds to approximately 5 mm in the wire coordinate, owing to the spatial separation of the wires in the readout plane. Time is measured in ticks, with each tick representing 0.5 μs corresponding to approximately 0.8 mm of electron drift.

It is important to note that no data augmentations and transformations other than conversion to grayscale, normalization, and resizing the images were employed in either phase of the training process due to the coordinate choice. The detector and event geometry would be violated in the opposite case.

For the late fusion model, the data loaders corresponding to the branch ResNet18-based models utilized the fixed seed to feed the data into the networks. The data was

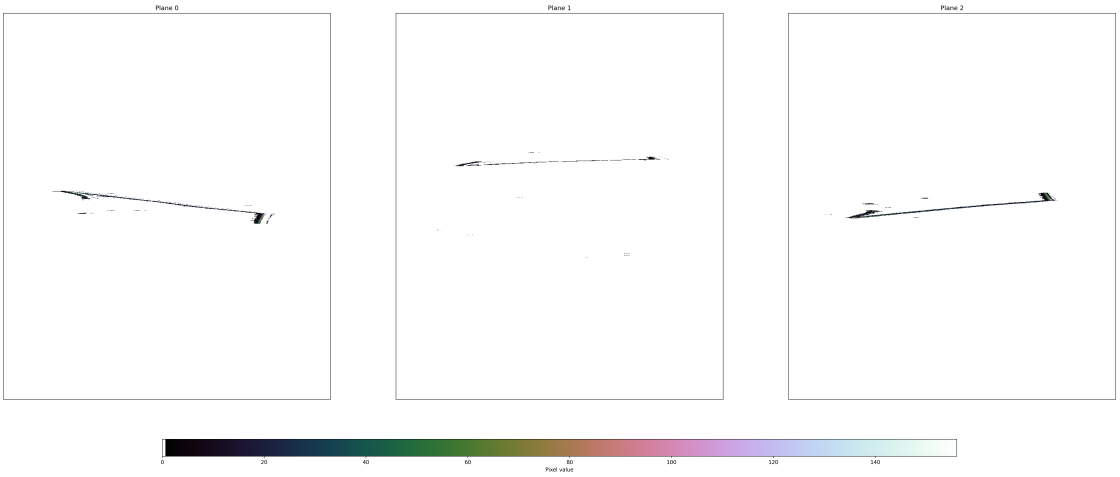


Figure 4.3: An example signal event projections. *Plane 0* and *Plane 1* correspond to *U* and *V* induction wire planes, and *Plane 2* is the collection plane denoted as *X* in Figure 4.1.

not shuffled to ensure the event projections from the readout planes align. The same applies to the early fusion model using the spatial channels (the readout plane views) instead of the traditional RGB representation.

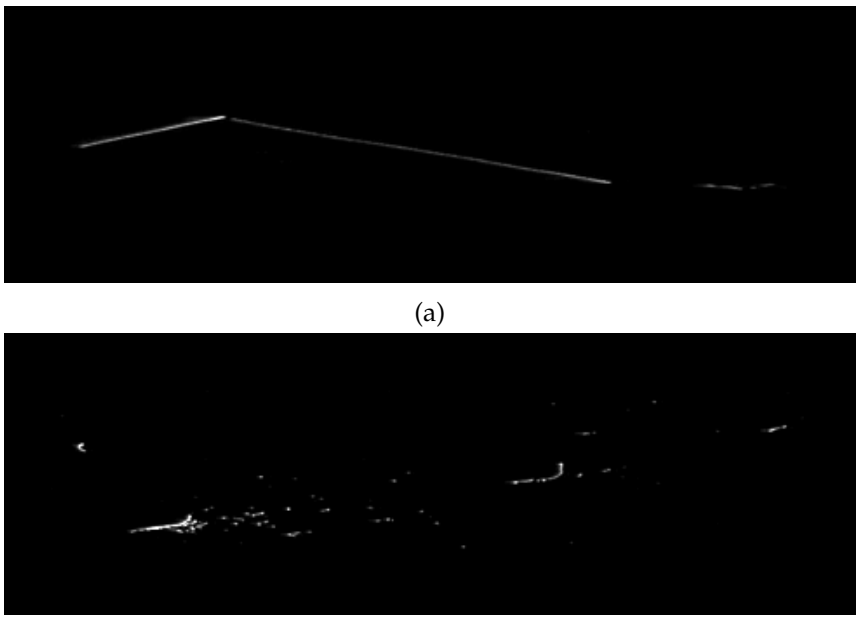
Figure 4.3 shows a signal $p \to K^+\overline{\nu}$ event as seen in the detector three readout views. Figure 4.4 shows signal and background interactions.

Each CSV file contains information about the region of interest (ROI). For each file, the ROI coordinates were extracted. The values inside the ROIs were checked to ensure high data quality, and events with empty or almost empty ROIs were discarded. Since the event is translation invariant, the ROIs were then zero-padded and centered to match the chosen image dimensions of 1000×1000 pixels. Any interactions spanning beyond the chosen dimensions were centered, cropped to fit within the 1000×1000 pixel region, and adjusted to ensure the most significant portion of the interaction was within the chosen ROI. Since the ROI arrays are mostly zero, they were saved as compressed sparse row (CSR) matrices to optimize the storage.

Table 4.3: Dominant K^+ decay modes. The statistics in the second column are taken from reference [16].

K^+ decay mode	Expected fraction	Simulated dataset
$K^+ \to \mu^+\overline{\nu}_\mu$	$(63.560 \pm 0.110)\%$	63.94%
$K^+ \to \pi^+\pi^0$	$(20.670 \pm 0.080)\%$	18.34%
$K^+ \to \pi^+\pi^+\pi^-$	$(5.583 \pm 0.024)\%$	4.76%
$K^+ \to \pi^0 e^+ \nu_e$	$(5.070 \pm 0.040)\%$	4.13%
$K^+ \to \pi^0\mu^+\nu_\mu$	$(3.352 \pm 0.033)\%$	3.04%
$K^+ \to \pi^+\pi^0\pi^0$	$(1.760 \pm 0.023)\%$	1.27%

The files were sorted according to kaon decay mode based on logs from the event generator, with statistics presented in Table 4.3.



(a)

(b)

Figure 4.4: A demonstration of a signal 4.4a and a background 4.4b simulated events.

The constrained flood fill algorithm was deployed to address the track continuation problem discussed in Subsection 4.1.1. The details regarding this approach will be described in the subsequent section.

4.2.1 The Constrained Flood Fill Algorithm

Flood fill is a classic algorithm used in computer graphics for filling a connected region with a single color or pattern. The flood fill algorithm starts from a given *seed* pixel and spreads to adjacent pixels. It continues to spread until it reaches the boundaries of the area, which are defined by pixels of a different color or a predefined limit.

The conventional flood fill algorithm does not consider additional constraints; it simply fills all connected pixels that match the seed pixel's properties (such as color or intensity). Several variations of flood fill algorithms exist, such as the four-way or eight-way flood fill, which differ in how they define connectivity between pixels (directly or diagonally adjacent).

On the other hand, the constrained flood fill adds certain conditions or limitations to the filling process. Here, the constraints are the minimum area of a region (track) to be considered and the maximum vertical distance between two regions to be considered a single track. These constraints restrict the fill to certain areas and characteristics, making it more controlled and selective. Instead of filling all connected regions with similar properties, constrained flood fill will only fill regions that meet specific criteria, such as size or proximity to another region.

The process starts by iterating through each image in the provided dictionary. The image is converted to grayscale, and then a binary mask is created using a thresholding technique. Nonzero pixels in this mask are identified, and if there are none, the loop continues to the following image.

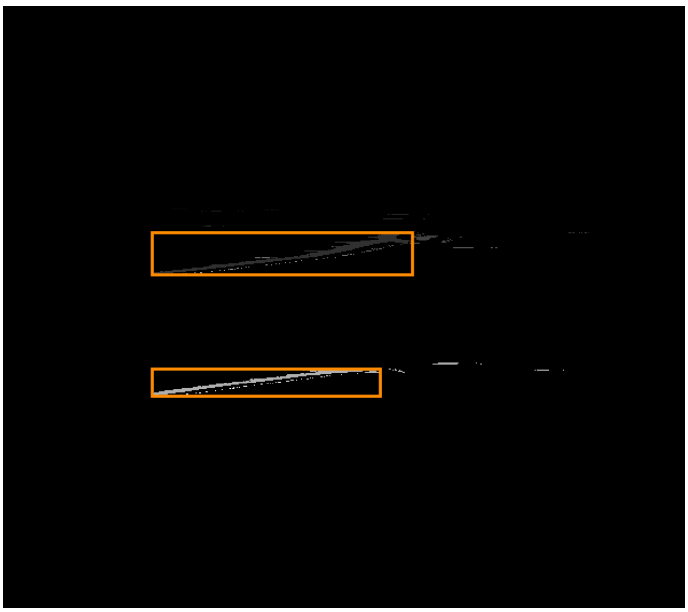


Figure 4.5: Demonstration of the constrained flood fill algorithm applied to an image of the signal. The sufficiently large regions (track parts) are marked with rectangles.

Algorithm 1 Constrained Flood Fill for Track Continuation Detection

Require: $I, A_{\text{min}}, D_{\text{max}}, T$
Ensure: L_{paths}
$L_{\text{paths}} \leftarrow []$
for each *img* in I **do**
    Convert *img* to grayscale and create binary mask M
    Identify nonzero pixels in M
    if M has no nonzero pixels **then**
        Continue to next *img*
    end if
    Choose a random seed pixel in M
    Apply flood fill to M; label and identify regions
    Initialize mask for large regions, M_{large}, and $found \leftarrow$ False
    for each *region* in identified regions **do**
        if $region.area \geq A_{\text{min}}$ **then**
            Update M_{large} and centroids
        end if
    end for
    for each pair of different regions (R_A, R_B) **do**
        if min vertical distance between R_A and $R_B < D_{\text{max}}$ **then**
            $found \leftarrow$ True; Break
        end if
    end for
    if $found$ **then**
        Append path of *img* to L_{paths}
    end if
end for
return L_{paths}

A random pixel is chosen as the seed for flood filling. The binary mask is flood-filled from this seed point, and the filled area is labeled. Regions in the labeled image are analyzed, and only those with an area larger than the minimum region area are considered. For each of these large regions, centroids and pixel coordinates are stored.

The core of the algorithm involves comparing each pair of regions. The vertical distance between any two pixels (one from each region) is calculated for each pair. If the minimum of these distances is less than the specified maximum vertical distance, it is determined that the continued track has been found. In the positive case, the process for the current image terminates and continues with the next image in the dictionary. A list of image paths where the continued tracks have been identified is returned for further processing.

About 10% of the signal and background data was impacted by the *double-track* issue. The images containing track continuation were removed from the dataset. This resulted in a reduction in the active volume of the LArTPC detector, which is not addressed in this work.

Chapter 5

The Methodology

This chapter overviews two data fusion methodologies, employing networks trained using Python 3.10.9, Torch 2.0.1+cu117, and Torchvision 0.15.2+cu117 [17] on four NVIDIA A100-SXM4-80GB GPUs within the HELIOS cluster [18] at the FNSPE Department of Mathematics. Bayesian hyperparameter search was conducted using the Optuna optimization framework and the Weights and Biases Sweeps tool [19, 20]. The model assessment metrics, such as ROC, precision, recall, and the F1 score, were calculated using the Scikit-learn library [21]. The code is available in our GitHub repository.

5.1 Ensemble Approaches

In machine learning, multimodal approaches primarily enhance system robustness by leveraging the unique information from individual data sources to clarify ambiguities and refine the quality of noisy data.

Multimodal machine learning techniques can be divided into early and late fusion, depending on their integration stage in the data processing pipeline. Although these methods are standard for data with multiple modalities, e.g., images, sound, or video, they are equally pertinent for the multi-view dataset in our context. Each readout plane in the detector captures distinct spatial representations of events. Thus, when combined, more relevant information about the event is obtained.

5.1.1 Late Fusion

Late fusion, or decision-level fusion, processes each spatial projection independently using separate neural network channels merged at the decision point. In this approach, each spatial representation is treated individually, allowing the network to capture and analyze the characteristics inherent to each view.

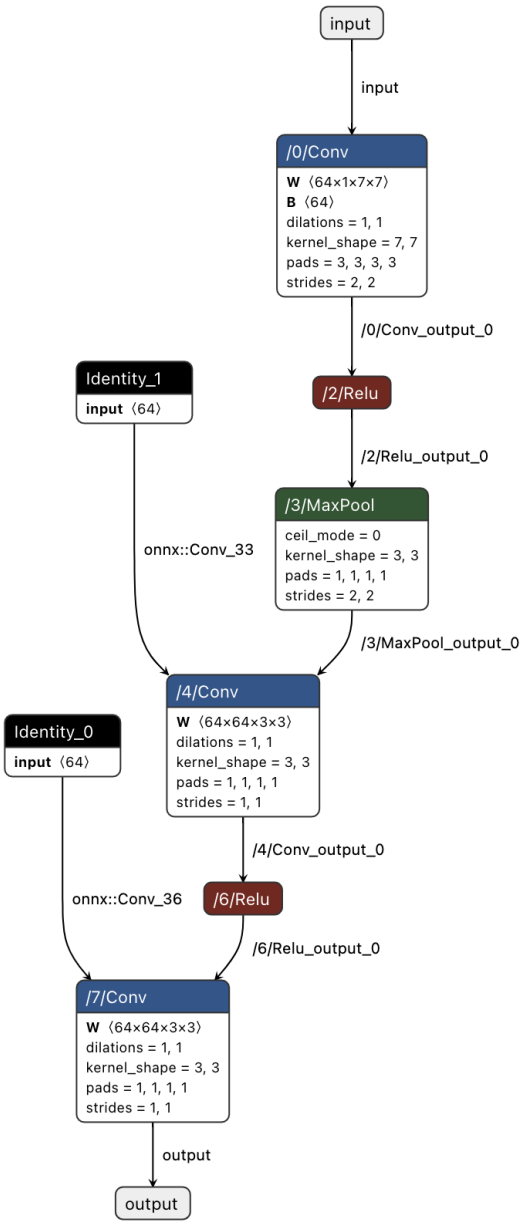


Figure 5.1: A schematic of the ResNet's *basic block* [13].

For late fusion, we utilized three ResNet18-based models. The first convolutional layer of each model is modified to accept one-channel (grayscale) images. Right after the ResNet backbone, two additional linear layers are added. Those are followed by the classifier layer with one output neuron producing the score representing the likeliness of the particular event projection being the signal.

During the training process, we used an early stopping technique with a five-epoch patience threshold, saving the model's best weights upon achieving minimal loss on the validation set.

The final classification layer of each model is removed, and all submodel parameters are frozen to ensure the integration of their high-level features rather than modifying their feature extraction functions. The architecture is designed to handle three inputs

corresponding to three readout plane projections, each processed separately by its respective model. Outputs from these models are flattened and concatenated, forming a unified feature vector. This vector is subject to a gated fusion process, which modulates the combined features. Subsequently, a linear classifier processes the modulated vector to produce a single output score.

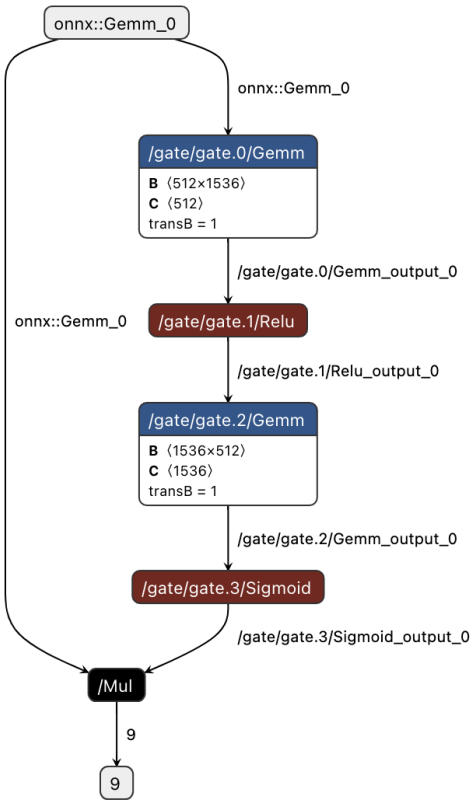


Figure 5.2: The architecture of the gating network utilized in the late fusion model.

Through a series of linear transformations and nonlinear activations (ReLU and sigmoid), the gating network dynamically regulates the concatenated feature vector from the branch models. Initially, it reduces the high-dimensional input to a lower dimension; then, a ReLU activation is applied. The ReLU is followed by another linear transformation and a sigmoid activation to scale the output to a range between zero and one.

The gating allows for priority feature extraction. Certain features are selectively amplified or attenuated by scaling the original feature vector with the gating network's output. The model then focuses on features more pertinent to the task; on the contrary, less informative or noisy features are sifted out to potentially increase the model's performance. Moreover, the gating mechanism bolsters model resilience: by learning different feature weightings, the model adapts to diverse input scenarios.

The architecture of the gating network is illustrated in Figure 5.2. The *Gemm*, or *general matrix multiplication*, unit denotes the fully connected layer. Here, **B** and **C** denote the weight matrix and a bias vector of the layer. The *transB = 1* attribute signals that the matrix **B** is transposed. The *Mul* unit stands for element-wise multiplication when applying the gating mechanism to the input. The network outputs a scaled feature vector.

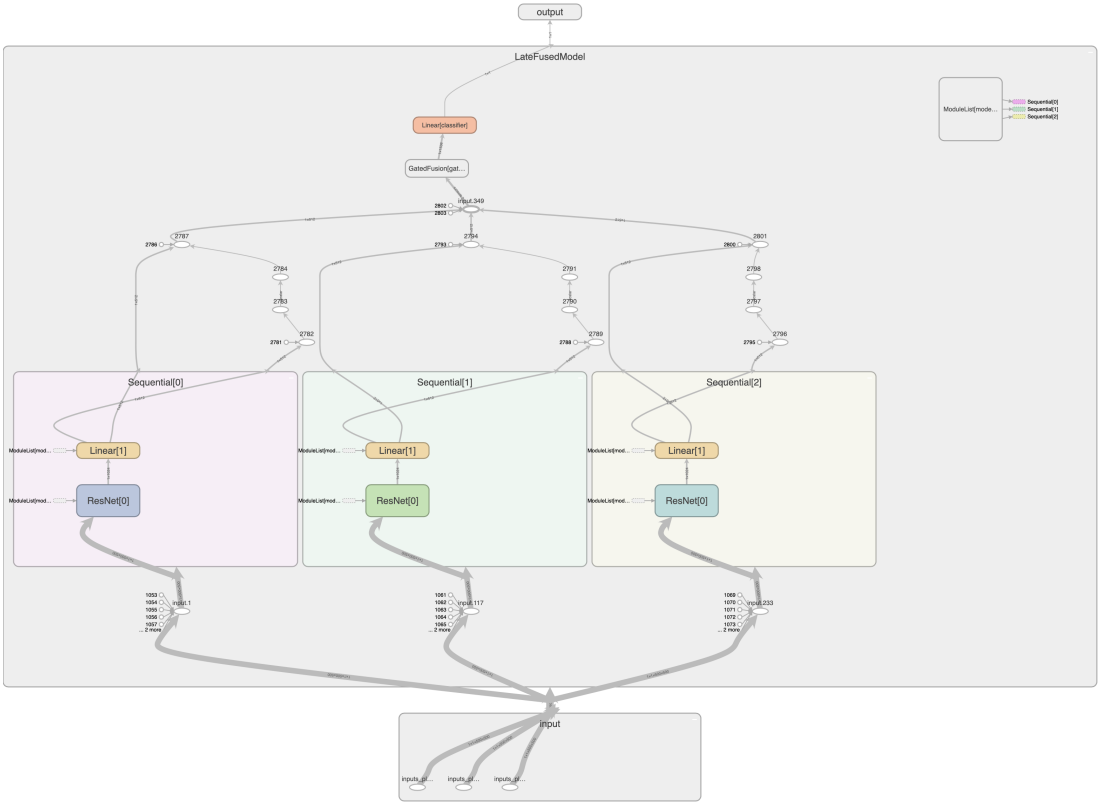


Figure 5.3: The simplified architecture of the late fusion model. The inputs corresponding to the three signal projections in the LArTPC are fed into the three ResNet18-based models. Subsequently, the outputs of the fourth layer of the ResNets18 pass through the average pooling, flattening, and two added linear layers. The outputs are concatenated and passed through the gating mechanism illustrated in Figure 5.2 to produce a single score (logit) representing the likeliness of the particular event being the signal.

5.1.2 Early Fusion

Early fusion, or feature-level fusion, combines the information from different sources at the outset before any extensive data processing takes place. The approach allows for the exploitation of potentially complementary information in the raw data. It is beneficial when the correlation between the individual modalities is high and when the joint distribution of features across the data is crucial for the task [22].

An EfficientNet B2 architecture [23] with pre-trained ImageNet [24] weights was used for early fusion. The single-channel images of the events corresponding to the three LArTPC readout planes were stacked along the channel dimension to create a single three-channel representation of the event in a way similar to RGB. Instead of the colors, the spatial channels are created.

5.1.3 Miscellaneous Ensemble Trials

In the earlier phases of our work, we employed simple, voting-style techniques, such as the *max-*, *average-*, and *weighted average*-ensemble. These processed the ResNet18-based submodel outputs in a way similar to the late fusion model. However, those were

applied in the final stage of the data processing pipeline. The parameters of the trained ResNet18-based models were frozen, the submodels were left to decide, and the final decision level layer implementing the particular voting method was added.

These models did not improve the classification quality in terms of our preferred metric, recall. On the contrary, these seemed to inherit the instability of the ResNet18-based submodels. Therefore, we shifted our focus toward more flexible and complex data-fusion models offering a more complete spatial representation of the events. The simple ensemble techniques are not considered in further discussions.

5.2 Training Details

The dataset employed in this study consisted of 31 731 and 83 957 signal and background events per readout plane, partitioned with 90% allocated for training and the remaining 10% equally split to validation and testing datasets. The signal-to-background ratio in the dataset yielded approximately 0.37 and was ensured to be preserved in all subsets.

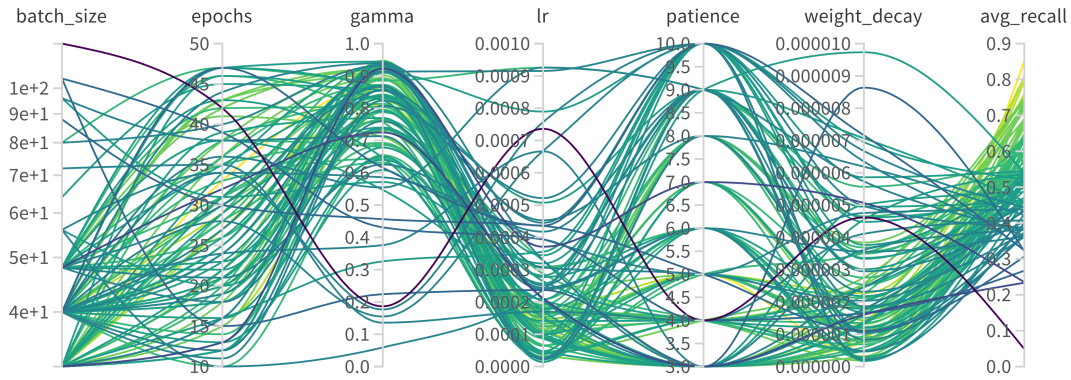


Figure 5.4: A parallel coordinate plot illustrating the hyperparameter search for the late fusion model. The individual lines pass through the coordinates corresponding to respective hyperparameters chosen by the Sweeps tool based on the Bayesian method. On the right is the scale of the optimization objective, in this particular case, the average recall on the validation dataset. The individual runs, or, *sweeps* with different hyperparameters lead to different objective values.

Table 5.1: The hyperparameter search description table.

Hyperparameter	Search space	Mod. ResNet18 choice	Late fusion choice
Batch size	32 – 128, step = 8	128	128
Patience	3 – 10	5	5
Learning rate	$10^{-6} - 10^{-3}$	2×10^{-4}	1.34×10^{-4}
Weight decay	$0 - 10^{-5}$	1.59×10^{-4}	4.93×10^{-4}
Gamma	$0.05 - 0.95$	0.71	0.63

We conducted a hyperparameter optimization for a modified ResNet18 and the late fusion architectures using the Optuna framework and the Weights and Biases Sweeps

tool. The objective of the Optuna *study* was to maximize the average recall on the validation dataset, paralleling sensitivity requirements in actual proton decay searches. The hyperparameter search spaces and the values chosen for training are listed in Table 5.1.

The Optuna optimization process for the modified ResNet18 model took 100 trials and utilized a median pruner as a callback. The median pruner compares the performance of running trials against the median performance of completed trials at similar stages and discontinues the trials with performance below the median.

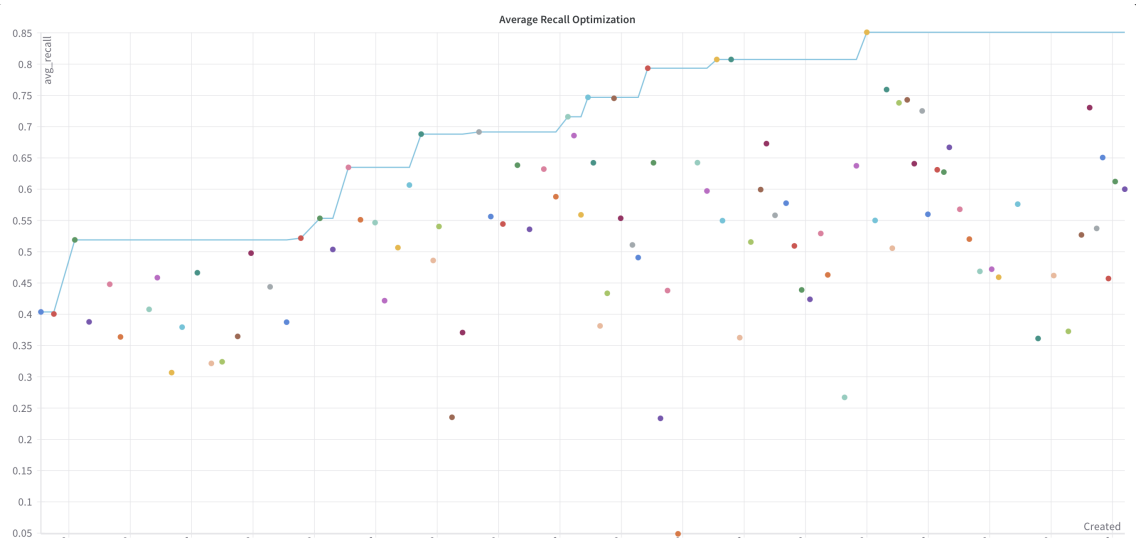


Figure 5.5: A scatterplot of the average recall values obtained during the Weight and Biases Sweeps hyperparameter search in time. The running maximum, indicated by a light-blue line, gradually increases over time, resulting in a better performance in terms of the average recall.

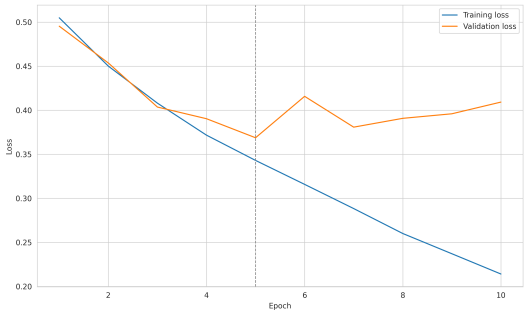
For the late fusion model hyperparameter search, the Sweeps tool by Weights and Biases was used. The pruners are not explicitly available in the Weights and Biases Sweeps tool. However, the tool is capable of predicting the success probability and expected improvement based on the chosen parameters. The Sweeps utilizes the Bayesian search and, based on the success of the preceding runs, terminates or accepts the current run. In Figure 5.4, a parallel coordinate plot illustrating the Sweeps optimization process is displayed.

The optimal hyperparameters (see Table 5.1) were applied to train individual branch models for late fusion. The branch models did not utilize the TL approach and were trained from scratch with randomly initialized weights. The training utilized the Adam optimizer, a variant of stochastic gradient descent (SGD), and incorporated an exponential learning rate scheduler. For all models, the batch size was set to 128 images. For the models trained on the *Plane 0* and *Plane 1* datasets, the training took 10 epochs; the model trained on the *Plane 2* dataset was trained for 12 epochs. Each epoch averaged approximately 7 minutes on four NVIDIA A100-SXM4-80GB GPU units. The trainable parameter count totaled 12 220 865 for each ResNet18-based branch model.

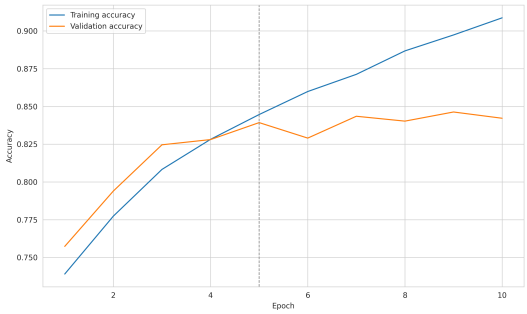
Learning curves for the three individual models are illustrated in Figure 5.6, alongside ROC and precision-recall curve (PRC) plots in Figure 6.1.

In the figures, the training loss consistently decreases, indicating that the model is

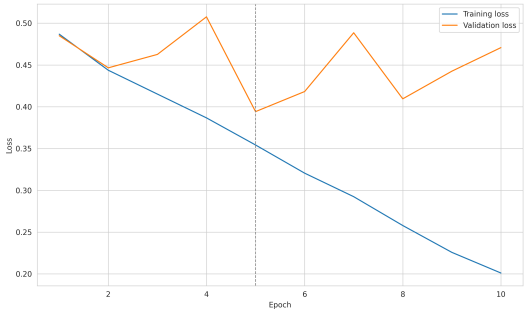
learning and improving its performance on the training data over time. This is expected as the optimization algorithm iteratively adjusts the model parameters to minimize the loss function. The validation loss provides insight into the model's ability to generalize to unseen data. In an ideal learning scenario, the validation loss should decrease alongside the training loss. However, here, the validation loss exhibits volatility throughout the epochs. Initially, as the model begins to learn from the training data, the validation loss decreases. This is indicative of the model's improving generalization capabilities. Nevertheless, as training progresses, the validation loss begins to fluctuate, which is symptomatic of overfitting.



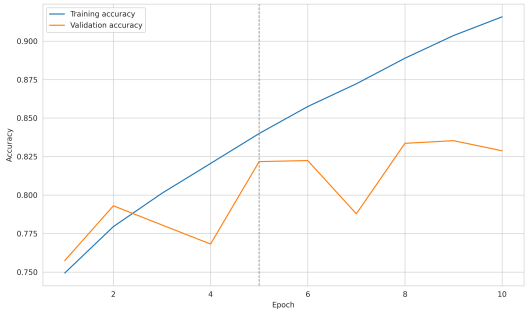
(a) *Plane 0* loss.



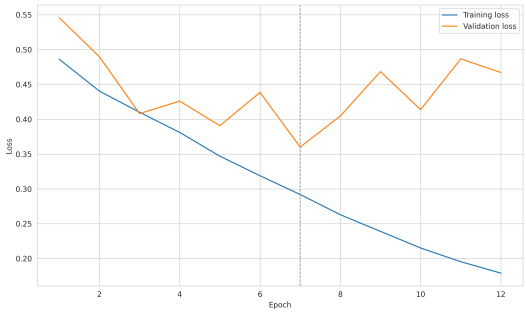
(b) *Plane 0* accuracy.



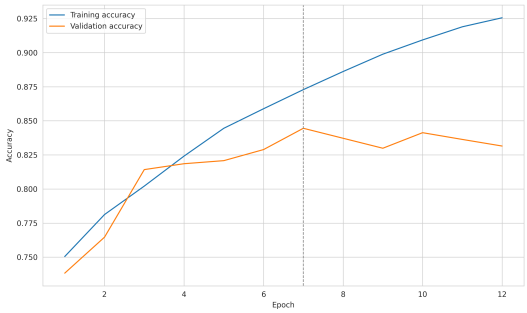
(c) *Plane 1* loss.



(d) *Plane 1* accuracy.



(e) *Plane 2* loss.

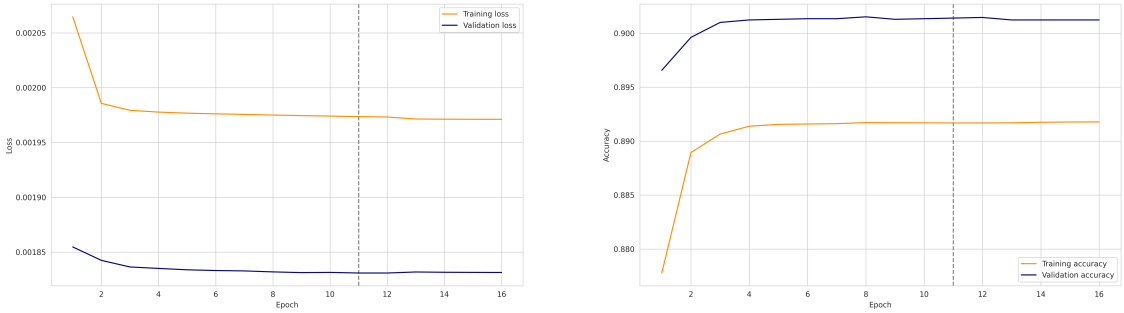


(f) *Plane 2* accuracy.

Figure 5.6: Loss and accuracy curves for the modified ResNet18 submodels used to construct the late fusion model. The dashed gray line marks the epoch with the best validation loss when the model weights were saved.

The loss on the validation set stopped decreasing, reaching the minimum value of approximately 0.4, followed by a slight increase. At the same time, the training loss continued decreasing to reach the lowest value of less than 0.2 at the last epoch. The trend of the training loss curve suggests the further decrease. Complement to loss plots are

the accuracy plots with accuracy drops corresponding to the loss raises and vice versa. For all three submodels, the validation accuracy stagnated at approximately 0.83. On the contrary, the accuracy on the train set continued to grow until reaching the score of approximately 0.93 by the last epoch.

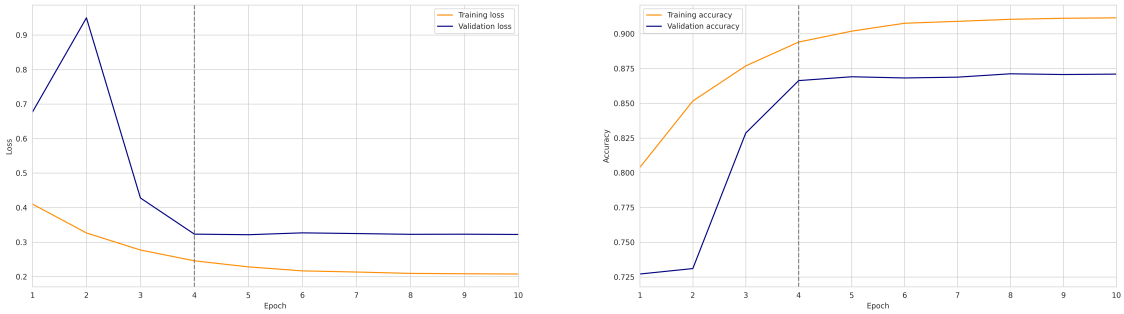


(a) (b)

Figure 5.7: Loss (5.7a) and accuracy (5.7b) plots for the late fusion model. The dashed gray line marks the epoch with the best validation loss when the model weights were saved.

The data-fusion techniques were employed to refine the performance. The trained submodels were used to construct the late fusion model. That was further trained for 16 epochs, with a batch size of 128 images, a learning rate of approximately 1.3×10^{-4}, modulated by the exponential learning rate scheduler with the gamma of 0.63. Each epoch in this phase averaged approximately 12 minutes. The training process utilized the early stopping technique with a patience of 5 epochs. The trainable parameter count yielded 1 576 449. The weights of the model were saved at the eleventh epoch when the validation loss reached its minimum value. This model's loss and accuracy curves are displayed in Figure 5.7. In contrast to the branch model learning curves, those of the late fusion models are smooth with evident trends. It is worth noting that the minimum loss, compared to that of the submodels, is substantially lower (for the same loss function, the binary cross-entropy), meaning less discrepancy between the ground truth and the predicted labels. The validation loss in the case of the late fusion model is lower than the training loss, yielding approximately 1.84×10^{-3} and 1.97×10^{-3}. Generally, the consistent decrease in loss combined with the very low loss scores mean a good level of model generalization. The validation accuracy score reached the maximum of approximately 0.9, which is slightly higher than the training accuracy, peaking at 0.89.

EfficientNet B2 architecture [23] with the pre-trained ImageNet weights [24] was used for early fusion. The number of trainable parameters in the model is 7 794 184. Instead of the standard RGB channels, the projections from the three readout planes were used. The hyperparameters for the early fusion were chosen manually during several runs with the different hyperparameter values. The model was trained over 10 epochs, with each epoch taking approximately 14 minutes. The loss and accuracy curves for the model are in Figure 5.8.



(a) (b)

Figure 5.8: Loss (5.8a) and accuracy (5.8b) curves on the training and validation set for the early fusion model. The dashed gray line marks the epoch with the best validation loss when the model is saved.

As discussed in section 3.4.1, loss and accuracy themselves are not sufficient for model assessment. Thus, additional metrics, such as F1 score, ROC, and PRC curves, are utilized to interpret the training results. The objective of the training is to optimize the recall on the validation set, which is similar to that of DUNE. In the next chapter, we present and discuss our results for individual branch models, the late fusion model, and the early fusion model in terms of proton decay sensitivity or recall.

Chapter 6

The Results

The objective is to accurately distinguish between the signal and the background simulated events represented by proton decay via $p \rightarrow K^{+}\overline{\nu}$ and atmospheric neutrino interactions on argon, respectively. The modified ResNet18 models were first utilized to learn the feature representations of the signal and background events on each respective readout plane view. The learned features were then combined using the late fusion approach with a gating mechanism to scale them based on their relevance to the classification, enhancing the model's flexibility and performance.

The signal-to-background ratio in the dataset yielded approximately 0.37. The signal is the minority class and has a higher importance in the context of the proton decay search; the right choice of model assessment metric is, therefore, crucial. In proton decay searches, the sensitivity, or recall, is the metric of choice. In this chapter, we present and discuss our results in terms of the sensitivity to proton decay.

The modified ResNet18 models, each trained from scratch on a corresponding view dataset, were utilized in the late fusion approach. The ROC and PRC curves for these models are in Figure 6.1.

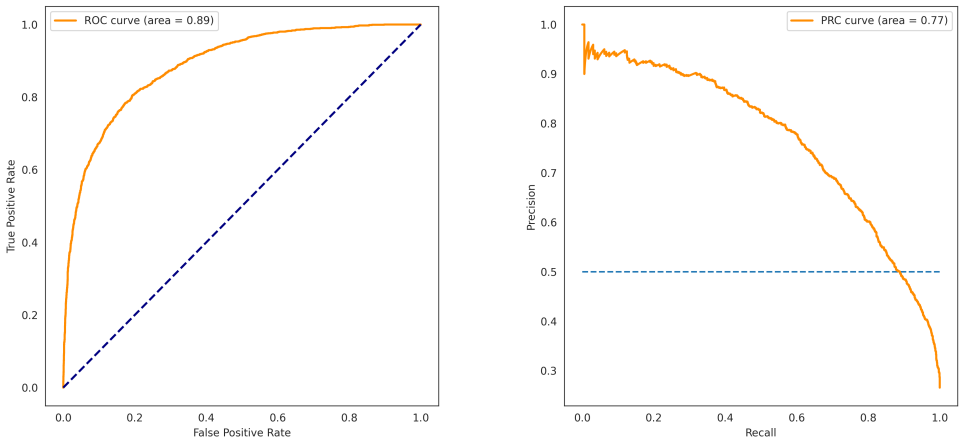
6.1 The Modified ResNet Results

The PRC plot displays the trade-off between precision and recall for different threshold settings. The area under the PRC curve (PRC AUC) is 0.77 and 0.72 for the induction plane projections and 0.79 for the collection plane projections. This suggests moderate performance, especially in a context where the positive class is rare or when the false positives are more costly. The performance is acceptable since precision consistently exceeds the baseline random classifier across most recall levels.

The ROC curve visualizes the trade-off between the true and false positive rates for various thresholds. The true positive rate is equivalent to recall, while the false positive rate is the proportion of negative instances incorrectly classified as positive. The area under the ROC curve (ROC AUC) yielded 0.89 and 0.87 for the models trained on *Plane 0* and *Plane 1* projections and 0.9 for the model trained on *Plane 2* projections. This indicates a solid discriminative ability compared to a no-skill classifier.

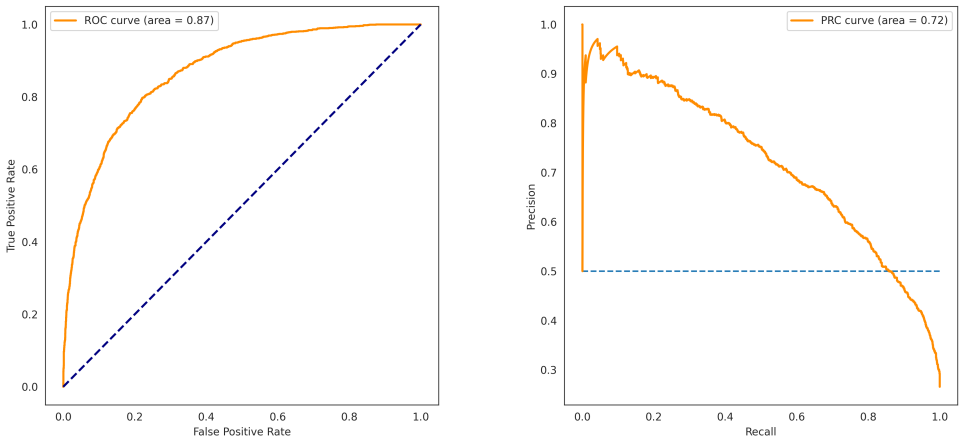
The results suggest that the individual branch models show a decent discriminative performance despite the loss and accuracy curves showing signs of early overfitting and instability.

The histograms in Figure 6.2 illustrate the distribution of the signal and background



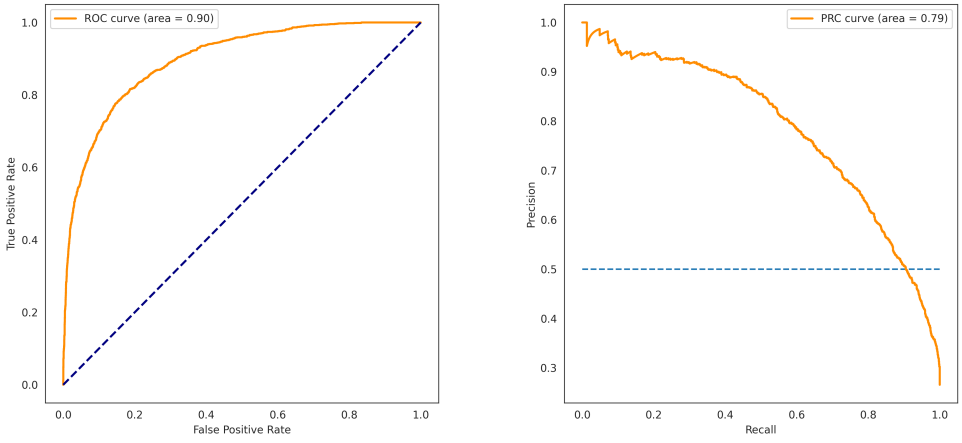
(a)

(b)



(c)

(d)



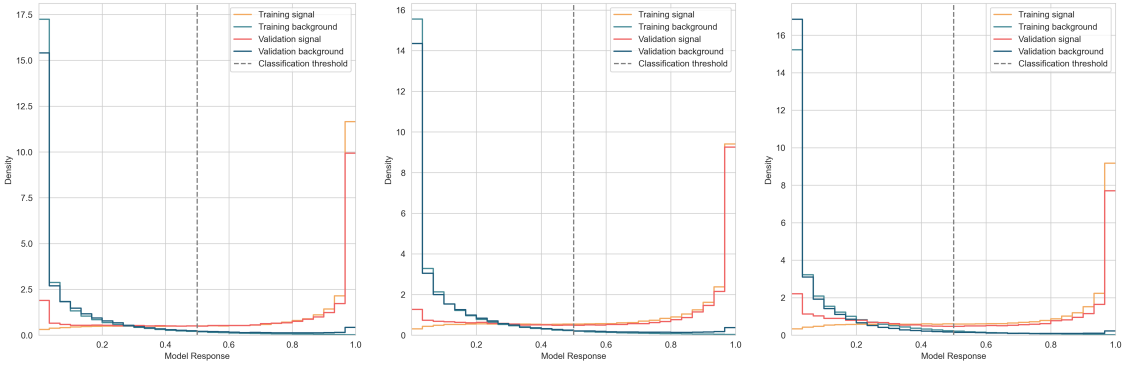
(e)

(f)

Figure 6.1: ROC and PRC curves for the modified ResNet18 submodels used to construct the late fusion model. Figure pairs 6.1a – 6.1b, 6.1c – 6.1d and 6.1e – 6.1f correspond to branch models trained on the *Plane 0*, *Plane 1* and *Plane 2* datasets, respectively. The dashed line represents the performance of the no-skill, or random guess, classifier. The curves are evaluated on the respective test datasets.

events based on the sub-model response scaled using sigmoid to the range of zero to unity.

On the validation set, the background density peaks near zero, and the signal density peaks near one, while having another minor mode around zero. On the training set, we observe a signal density with a rather heavy tail and one mode around unity. In contrast, the background density is light-tailed, with a prominent peak around zero. This pattern holds across all three branch models, with no significant improvement on the collection plane (*Plane 2*) dataset.



(a) (b) (c)

Figure 6.2: Distribution of the signal and background samples based on the response of the modified ResNet sub-models. From left to right, individual figures correspond to models trained on *Plane 0*, *Plane 1*, and *Plane 2* datasets. The dashed gray line represents the classification threshold of 0.5.

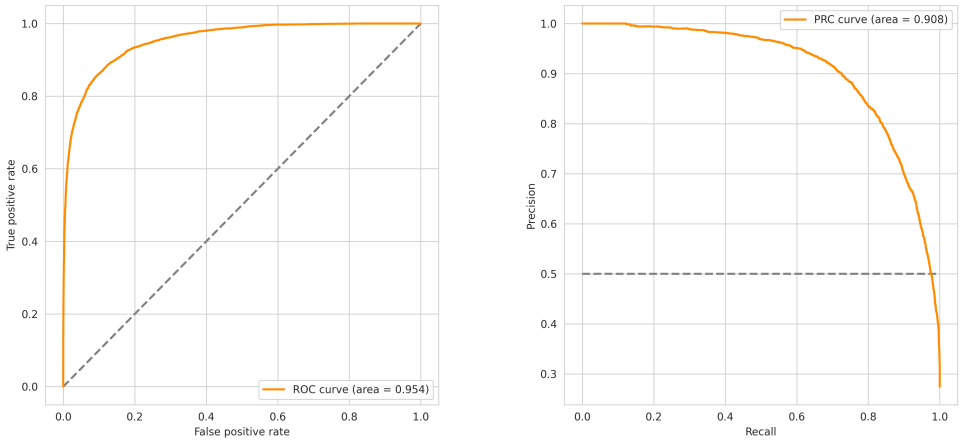
Notably, the model response at around 0.3 marks the intersection of the signal and background distributions, which falls below the classification threshold used for metric calculation. It is essential to clarify that this behavior does not signal poor performance since evaluation is based on ROC AUC and PRC AUC. However, in the future works, the calibration of the classification threshold with respect to the metric of choice will be performed.

The track continuation detection procedure may have induced differences (see Figure 5.6 and Figure 6.1) in the performance of the induction and collection plane datasets. The track continuation problem occurs when the readout plane wires partly detect the particle track on one side of the APA frame, and the rest of the track is recorded by the wires on the other side of the APA frame, due to the wire wrapping scheme. The *double-track* cases were identified using the constrained flood fill algorithm described in Section 4.2.1. The algorithm parameters are the tolerance, the minimum area of the region containing the track (the circumscribed rectangle), and the maximum distance between each two regions to consider the track parts as a single track. However, the performance of the algorithm was only assessed by the visual inspection of a small subset of the data. That revealed that some track continuation instances are still present. This indicated that the algorithm is not sufficiently universal since the region separation presents a hard, non-adaptive constraint and improvements to the constrained flood fill algorithm are needed.

An algorithm more reliable and reasonable than a simple visual inspection is required to assess the performance of the constrained flood fill and identify potential algorithm drawbacks.

6.2 The Late Fusion Results

The fusion of the three modified ResNet18 models improved the classification performance on the test set, resulting in the ROC AUC of 0.954 and the PRC AUC of 0.908, which is an expected behavior due to the gating mechanism that enables the priority training by multiplying the more significant feature vectors by higher sigmoid outputs.

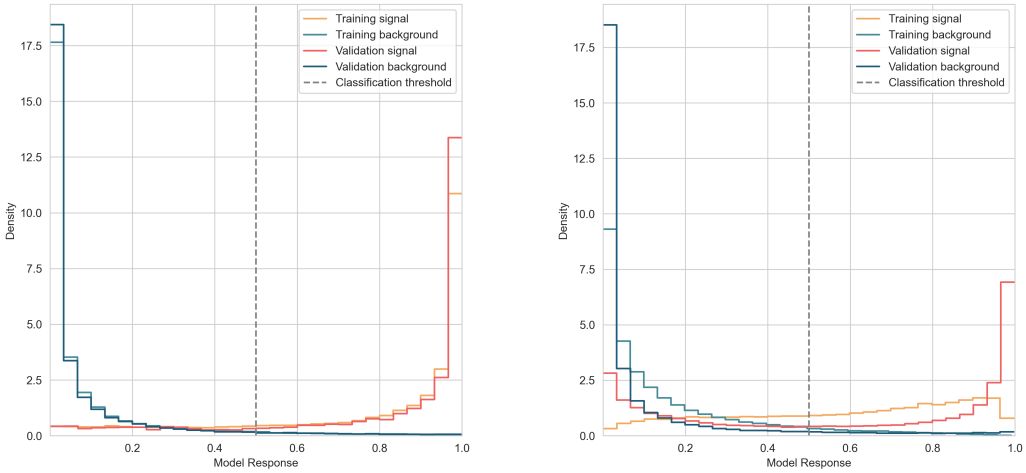


(a) (b)

Figure 6.3: The ROC (6.3a) and PRC (6.3b) curves for the late fusion model. The curves are evaluated on the test dataset.

The discriminability improvement is illustrated in Figure 6.4a. Indeed, the distribution tails of signal and background densities on the validation set are lighter, resulting in more prominent peaks. The small peak of the signal density observed for the ResNet18-based submodels is no longer present in the late fusion model histogram.

Even though the signal-to-background ratio was ensured to be approximately the same on both sets, the training data seemed to be slightly harder to predict (see accuracy and loss plots in Figure 5.7 for the late fusion model). That is possibly due to the gating mechanism (Figure 5.2) serving as a regularization and on-the-flight feature selection technique. Dropout, the most common regularization approach, temporarily removes a portion of network units with all their input and output connections. The choice of which units to drop is random. The dropout is applied during the training phase to prevent overfitting [25]. The gating mechanism is conceptually very similar to dropout. Instead of the probabilistic approach, the gate scales the features based on their importance, which is represented by the sigmoid output score, to result in the effective feature space dimension reduction. Nevertheless, in contrast to the dropout technique, our gating mechanism is applied in all phases of model training.



(a) (b)

Figure 6.4: The distribution of the signal and background events based on the response of the late (6.4a) and the early (6.4b) fusion models. The histograms are evaluated on the training and validation datasets.

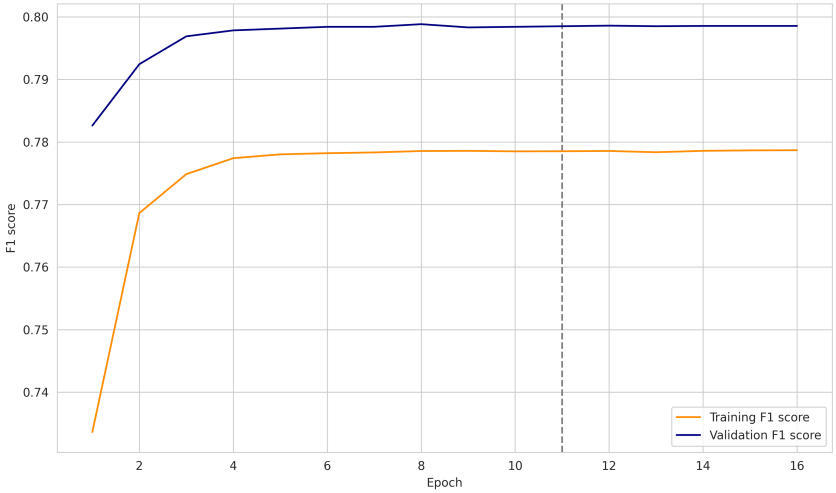


Figure 6.5: The F1 score curve on training and validation set for the late fusion model.

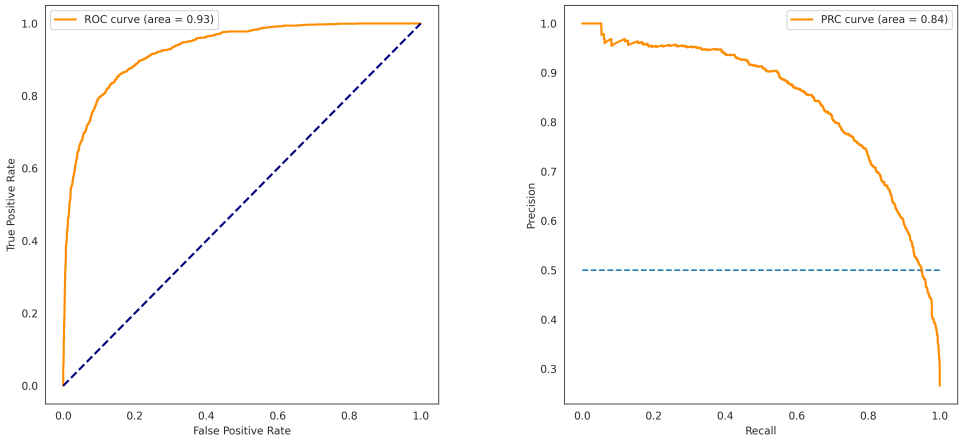
For completeness, an F1 score plot for the late fusion model is displayed in Figure 6.5. The curves are smooth with a clear increasing trend with a maximum of approximately 0.8, which indicates a decent model performance. The late fusion model improved the performance of the branch models which demonstrated the F1 scores of merely 0.7.

6.3 The Early Fusion Results

The early fusion model does not rely on the modified ResNet18 model results. Instead, it combines the information contained in the simulated LArTPC readout plane projections. Compared to the performance of the modified ResNet18 model on the *Plane 2* test dataset, the ROC AUC and the PRC AUC increased by approximately 0.03 and 0.05, respectively. Figure 6.4b displays the distribution of signal and background events based

on the model response.

The distributions of the signal and the background are mostly distinct, on both datasets, with heavy-tailed training signal density with a rather small peak around the unity. Overall, the histogram signals a decent discriminative ability of the model.



(a) (b)

Figure 6.6: The ROC (6.6a) curve and the PRC (6.6b) curve for the early fusion model. The curves were evaluated on the test dataset.

The ROC and PRC curves for the early fusion model are displayed in Figure 6.6. The complications hindering the model improvement may arise in the early fusion applied to the signal and background time projections stacked akin to the RGB channels. The challenge lies in the spatial positioning of the signal patterns, which differs from the conventional RGB representation where color components align.

6.4 Comparative Analysis

Few studies with similar methodology and objective are available in the field, while none are for proton decay search. The proper comparison is, therefore, impossible. However, we would like to highlight the works presented in [11] and [26].

In [11], the neutrino interactions are classified by the DUNE Convolutional Visual Network (CVN) illustrated in Figure 6.7. The architecture of the CVN is based on the Squeeze-and-Excitation ResNet34 (SE-ResNet34) variant [13, 27, 28] and shares conceptual similarities with our late fusion model. The CVN receives three inputs that correspond to the three readout views of the LArTPC. The inputs are 500×500 pixel images of simulated neutrino interactions generated in (wire, time) coordinates. The dataset used in the study consisted of a total of 3 212 351 events from a single MC sample, with each event represented by three LArTPC readout plane projections.

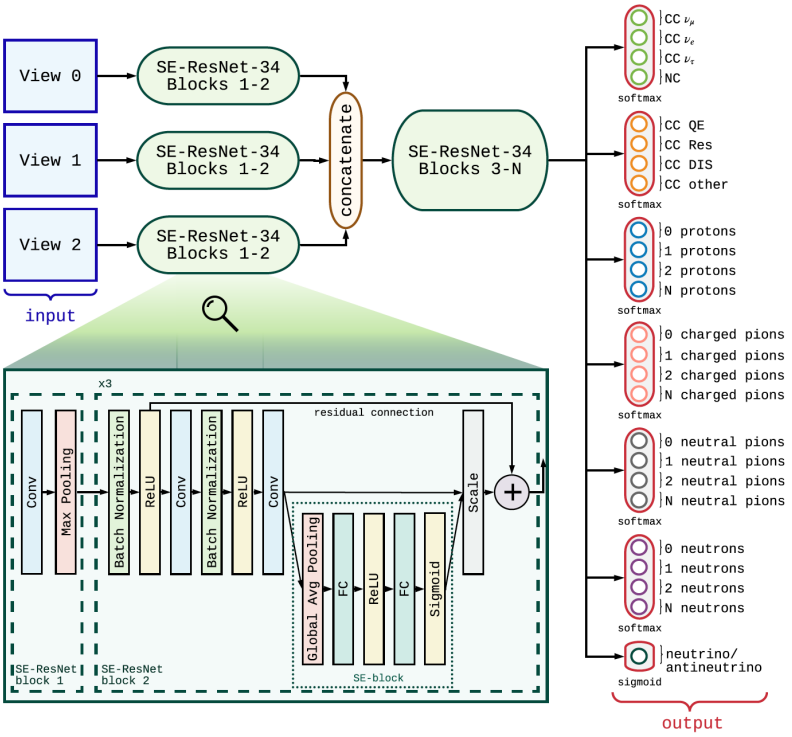


Figure 6.7: Simplified diagram of the DUNE CVN architecture [11].

The model was trained for 15 epochs utilizing the early stopping technique. The model achieved a maximum training accuracy of approximately 0.92 and a validation accuracy of approximately 0.91 for neutrino flavor classification. Figure 6.8 displays the distribution of events based on the CVN charged current (CC) ν_μ classification score for the RHC beam mode [4]. The original paper presents results for various neutrino flavors, including a maximum selection efficiency of 97% for the CC ν_μ signal in the RHC neutrino beam mode. This outcome corresponds to a maximum recall of 0.97 and applies to a dataset with 27% of the CC ν_μ signal and 40% neutral current (NC) background. Further results can be found in [11].

In [26], a CNN-based algorithm for the separation of particle tracks and showers and Michel electron identification is proposed. The algorithm was tested on the ProtoDUNE detector data. The inputs to the network illustrated in Figure 6.10 are 48×48 pixel images of the small detector regions, or *patches* centered around the pixel corresponding to the highest energy deposit. The dataset consisted of 30 million images, with roughly 15 million containing tracks, 11 million containing showers, 3 million empty, and 1 million in the Michel sample. The classification threshold is optimized based on the F1 score.

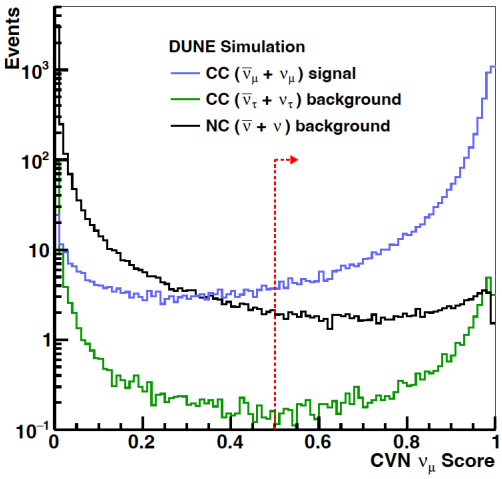
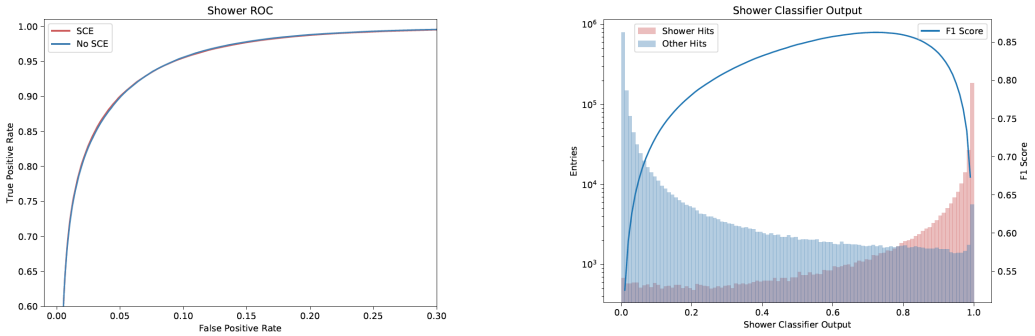


Figure 6.8: The number of events as a function of the CVN CC ν_μ classification scare for RHC beam mode. Neutrino and antineutrino interactions have been combined within each histogram category [11]



(a) (b)

Figure 6.9: The ROC curve (6.9a), histogram and the F1-score as a function of the classifier threshold (6.9b) [26].

The results are presented in terms of the ROC curve, efficiency (recall), and the F1 score. Figure 6.9 illustrates the ROC curve and the histogram combined with an F1 score curve for the track-vs-shower classification. The ROC AUC value is not discussed in the paper [26].

Our proton decay identification study achieved a maximum recall of 0.71, while the late fusion model's F1 score reached approximately 0.8. The findings in [26] were obtained for a shower-to-track ratio of around 0.73, while our signal-to-background ratio was roughly 0.37. In [11], the signal-to-background ratio in CC ν_μ classification was 0.67. Due to the distribution discrepancy and different nature of the interactions, the results could not be directly compared. Nevertheless, we can glean from both studies that the classification threshold should be calibrated based on the study's requirements and the preferred metric.

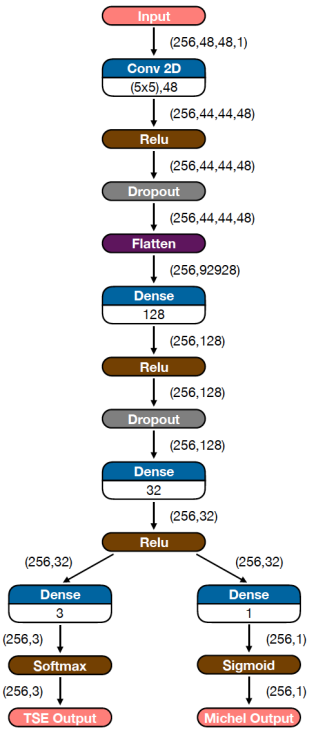


Figure 6.10: The CNN architecture utilized in the study [26]. The output of the network features two branches. The first branch returns the track, shower, or empty (TSE) classification scores. The second branch returns the score for a Michel electron classification.

Chapter 7

Conclusions and Further Research

Proton decay, proposed by A. Sakharov in 1967 [2], caused quite a stir in the particle physics community and prompted extensive research and experimental efforts. The Deep Underground Neutrino Experiment, with its cutting-edge detectors and a comprehensive BSM physics program, has a great potential to unveil new and exotic physics to improve our understanding of matter and the universe.

In this thesis, we explored the opportunities of deep learning in HEP research. The CNNs are advantageous in feature engineering, with their ability to autonomously extract meaningful information. Residual networks, particularly, are of great value: the shortcut connections allow the training of much deeper networks with a substantially lower risk of overfitting [13].

We trained three modified ResNet18 models from scratch on the dataset consisting of the simulated proton decay via $p \to K^+\overline{\nu}$ in LAr and background interactions of atmospheric neutrinos and argon. Despite the subtle nature of the data, the models showed a decent discriminative ability. They performed well in terms of the ROC, PRC on the test set (Figure 6.1), and event distributions based on scores evaluated on the training and validation datasets (Figure 6.2). The models were then used to develop a late fusion architecture featuring the so-called gate (Figure 5.3, Figure 5.2). The resulting late fusion model improved the performance of the networks substantially.

A separate branch of our ongoing work is the development of the early fusion architecture, which combines information from the three readout planes to create a comprehensive spatial representation of events. Our current architecture choice involves replacing the traditional RGB image representation in the EfficientNet B2 model with the spatial channels, i.e., the projections from the LArTPC readout planes. While the model only slightly improved accuracy compared to modified ResNet18 models trained separately on corresponding readout view datasets, the signal and background distribution plot indicates that the model is potent enough to effectively distinguish between signal and background (Figure 6.4b).

Overall, the individual branch models and the early fusion model display learning curve patterns that suggest overfitting and mild instability, but they also demonstrate commendable performance with regard to ROC and PRC. However, among the classifiers employed, the late fusion model stands out as the most effective.

7.1 Future Research

In our further work, we plan to develop flexible and resilient late and early fusion architectures as well as run extensive hyperparameter optimization for the models. Our objective will remain to maximize the sensitivity of the models to proton decay. Furthermore, we have plans to implement a multiclass classification problem for proton decay via $p \rightarrow K^{+}\overline{\nu}$ based on different kaon decay modes (Table 4.3). For that, substantially more data will be needed, implying more stringent requirements on the data processing pipeline. The computational challenges we have encountered thus far in data preprocessing include converting the original CSV file containing the image data and analyzing event images, which involves extracting ROIs and detecting track-continued instances. Rather than eliminating the track continuation cases from the dataset (as discussed in 4.2.1), which would reduce the active volume of the LArTPC, our proposed solution is to reconstruct the tracks. An algorithm more potent than a simple visual inspection of the processed images will be developed to assess the track reconstruction process. To evaluate the model performance, we aim to utilize conformal prediction.

It was a great opportunity to learn to train (deep neural networks) within a framework such promising as the Deep Underground Neutrino Experiment. We aspire to further contribute to the project that may be the culmination of neutrino and nucleon decay physics. However, hopefully, with that, the epilogue to the BSM discoveries will not follow.

References

[1] J. L. Barrow, "Towards Neutron Transformation Searches," January 2021. FERMILAB-THESIS-2021-37.

[2] A. D. Sakharov, "Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe," *Soviet Physics Uspekhi*, vol. 34, p. 392, may 1991.

[3] K. Babu, E. Kearns, U. Al-Binni, S. Banerjee, D. Baxter, Z. Berezhiani, M. Bergevin, S. Bhattacharya, S. Brice, R. Brock, *et al.*, "Baryon number violation," *arXiv preprint arXiv:1311.5285*, 2013.

[4] B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II: DUNE Physics," February 2020.

[5] B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I: Introduction to DUNE," *JINST*, vol. 15, no. 08, p. T08008, 2020.

[6] C. Alt, B. Radics, and A. Rubbia, "Neural-network-driven proton decay sensitivity in the $p \to \bar{\nu} K^+$ channel using large liquid argon time projection chambers," *JHEP*, vol. 04, p. 243, 2021.

[7] Fermi National Accelerator Laboratory, "Overview of the Long-Baseline Neutrino Facility and Deep Underground Neutrino Experiment." https://lbnf-dune.fnal.gov/about/overview/ Accessed: January 1, 2024.

[8] V. Hewes *et al.*, "Deep Underground Neutrino Experiment (DUNE) Near Detector Conceptual Design Report," *Instruments*, vol. 5, no. 4, p. 31, 2021.

[9] C. Andreopoulos *et al.*, "The GENIE Neutrino Monte Carlo Generator," *Nucl. Instrum. Meth. A*, vol. 614, pp. 87–104, 2010.

[10] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle, "A Convolutional Neural Network Neutrino Event Classifier," *JINST*, vol. 11, no. 09, p. P09001, 2016.

[11] B. Abi *et al.*, "Neutrino interaction classification with a convolutional neural network in the DUNE far detector," *Phys. Rev. D*, vol. 102, no. 9, p. 092003, 2020.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[14] D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W.-D. Ma, and B. McWilliams, "The Shattered Gradients Problem: If ResNets are the answer, then what is the question?," *ArXiv*, vol. abs/1702.08591, 2017.

[15] C. Alt, *Sensitivity study for proton decay via $p \to \bar{\nu}K^+$ using a 10 kiloton dual phase liquid argon time projection chamber at the Deep Underground Neutrino Experiment*. PhD thesis, Zurich, ETH, 2020.

[16] Particle Data Group, "Strange Mesons: $K^{\pm}$ Infromation." https://pdglive.lbl.gov/Particle.action?init=0&node=S010&home=MXXX020 Accessed: January 1, 2024.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[18] Strachota, Pavel, "HELIOS Cluster Documentation." http://helios.fjfi.cvut.cz Accessed: January 3, 2024.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[20] Biewald, Lukas, "Experiment Tracking and Hyperparameter Tuning with Weights and Biases," 2020. https://www.wandb.com/ Accessed: January 3, 2024.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, "Early vs Late Fusion in Multimodal Convolutional Neural Networks," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–6, 2020.

[23] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[26] A. Abed Abud *et al.*, "Separation of track- and shower-like energy deposits in ProtoDUNE-SP using a convolutional neural network," *Eur. Phys. J. C*, vol. 82, no. 10, p. 903, 2022.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *European Conference on Computer Vision*, 2016.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.