ACTA UNIVERSITATIS UPSALIENSIS
Uppsala Dissertations from the Faculty of Science and Technology
150

Adeel Akram

Towards Realistic Hyperon Reconstruction in PANDA

From Tracking with Machine Learning to Interactions with Residual Gas

UPPSALA
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in Häggsalen,
Ångström, Lägerhyddsvägen 1, Uppsala, Friday, 19 May 2023 at 22:07 for the degree of
Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Dr.
Andreas Salzburger (European Organization for Nuclear Research (CERN)).

Abstract

Akram, A. 2023. Towards Realistic Hyperon Reconstruction in PANDA. From Tracking with Machine Learning to Interactions with Residual Gas. *Uppsala Dissertations from the Faculty of Science and Technology* 150. 178 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-1773-1.

The PANDA (anti-Proton ANnihilation at DArmstadt) experiment at FAIR (Facility for Antiproton and Ion Research) aims to study strong interactions in the confinement domain. In PANDA, a continuous beam of anti-protons will impinge on a fixed hydrogen target inside the High Energy Storage Ring (HESR), a feature intended to attain high interaction rates for various physics studies *e.g.* hyperon production.

This thesis addresses the challenges of running PANDA under realistic conditions. The focus is two-fold: developing deep learning methods to reconstruct particle trajectories and reconstruct hyperons using realistic target profiles. Two approaches are used: *(i)* standard deep learning model such as dense network, and *(ii)* geometric deep leaning model such as interaction graph neural networks. The deep learning methods have given promising results, especially when it comes to *(i)* reconstruction of low-momentum particles that frequently occur in hadron physics experiments and *(ii)* reconstruction of tracks originating far from the interaction point. Both points are critical in many hyperon studies. However, further studies are needed to mitigate *e.g.* high clone rate. For the realistic target profiles, these pioneering simulations address the effect of residual gas on hyperon reconstruction. The results have shown that the signal-to-background ratio becomes worse by about a factor of 2 compared to the ideal target, however, the background level is still sufficiently low for these studies to be feasible. Further improvements can be made on the target side to achieve a better vacuum in the beam pipe and on the analysis side to improve the event selection.

Finally, solutions are suggested to improve results, especially for the geometric deep learning method in handling low-momentum particles contributing to the high clone rate. In addition, a better way to build ground truth can improve the performance of our approach.

Keywords:

Adeel Akram, Department of Physics and Astronomy, Nuclear Physics, Box 516, Uppsala University, SE-751 20 Uppsala, Sweden.

ISSN 1104-2516
ISBN 978-91-513-1773-1
URN urn:nbn:se:uu:diva-499272 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-499272)

This thesis is dedicated to all physicists trying to understand the visible universe, and to those who have gone to the dark side.

Contents

Part I:
Motivation and Methods

1. Introduction

The goal of the antiProton ANihilation at DArmstadt (PANDA) experiment at the Facility for Anti-proton and Ion Research (FAIR) is to study strong interactions at the energy scale where quarks form hadrons. In PANDA, a quasi-continuous beam of antiprotons ($\bar{p}$) will impinge on a fixed hydrogen (p) target inside the High Energy Storage Ring (HESR), resulting in high interaction rates - up to 20 MHz - for hadron physics studies. The beam momentum will range from 1.5 GeV/c to 15 GeV/c.

Antiproton-proton annihilation reactions will provide opportunities to study hadronic processes in the strong interaction confinement domain. Detector signatures from these processes are complex and are very similar to other hadronic processes of less interest, which makes it difficult to distinguish signal from the background. The latter is particularly true in the high-intensity environment that PANDA provides. In feasibility studies of hadronic reactions using simulated data, it is essential to consider realistic conditions at the operational and analysis levels.

The focus of this thesis is two-fold. First, we aim to develop realistic track reconstruction algorithms. Second, we will simulate particle reactions using realistic target conditions, taking vacuum conditions in the beam pipe into account.

Several realistic track reconstruction algorithms are currently under development within PANDA. The event topology, *i.e.* how many tracks and decays an event contains, dictates which algorithm is suitable for track reconstruction, in particular the particle multiplicity, the particle momenta and the location of decay vertices. The presence of neutral particles provides an additional challenge: they leave no signal in tracking detectors but can only be observed indirectly through their decay products. This makes track reconstruction even more complicated. Many classical algorithms are specialized to specific scenarios; for example, tracks that originate in the interaction point. Other algorithms search for particles that decay a measurable distance away from it. Most algorithms, especially those designed for high-energy experiments, are less efficient for low-momentum particles with spiralling trajectories that may overlap with those of other particles. Such tracks are difficult to distinguish from one another. The frequent occurrence of low-momentum tracks with strong curvatures in mid-to-low energy hadron physics experiments, motivates the development of new track reconstruction algorithms that can accommodate these issues. In high-energy physics, the use of deep learning algorithms in particle tracking is gaining momentum. Would such approaches be useful also in the low- and intermediate domain of hadron physics? To answer this question, this thesis presents novel deep-learning methods to reconstruct particles produced in antiproton-proton annihilations in PANDA. The algorithms have been bench-marked, first using muons ($\mu^+\mu^-$) and then Λ hyperons decaying into pions and protons. In the second part of the thesis, I will outline the development and implementation of these methods and evaluate them based on the aforementioned benchmark studies.

In the third part of the thesis, I investigate how the reconstruction efficiency is affected if realistic target conditions in the beam pipe are considered. Within PANDA, two types of hydrogen targets are under development: a pellet target and a cluster-jet target. In the first case, high-density pellets will be injected to collide with the beam at the interaction point. In the second case, jets of cooled hydrogen gas will be used. Due to the beam-target interaction, gas will evaporate from the pellets and jets and may dissipate into the beam pipe, resulting in an effectively large target. This has been shown in vacuum simulations performed by a group in Muenster, Germany, where vacuum pumps and a cryogenic pump are used to mitigate the residual gas [1]. Nevertheless, all physics feasibility studies within PANDA have until now been performed assuming a point-like target. In my thesis, I will investigate the effect of residual gas on the reconstruction of hyperons using $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ as a benchmark channel using ideal track finding algorithm to minimize effects from the track reconstruction.

1.1 Thesis Disposition

The current thesis is divided into three major parts as shown in Figure 1.1. Part I consists of work done by others. It includes a physics background, experiential details, and methods applied in this thesis. Parts II and III cover the work done by myself.

PART I Motivation and Methods	PART II Realistic Track Reconstruction in the PANDA Target Spectrometer using Deep Learning Techniques	PART III Simulations of Hyperons under Realistic Target Conditions	PART VI Conclusion, Outlook, etc

Figure 1.1. Distribution of the work presented in this thesis.

In Part II, machine learning techniques are used for track reconstruction in the Straw Tube Tracker of the PANDA experiment. First, I explore the reconstruction of muons using the standard and geometric deep learning models. Next, low-momentum hadron from $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ reaction are reconstructed using geometric deep learning in the same manner as of muons. In Part III, the effects of the residual gas are studied by simulating and reconstructing hyperons from the $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ reaction.

The content of this thesis is partly based on the licentiate dissertation *"Towards a Realistic Hyperon Reconstruction with PANDA at FAIR"* that I submitted at Uppsala University in 2021. Most of the content from the licentiate has been substantially revised. From Part I, Chapters 1, 2, 3, and 4 were part of the licentiate thesis; all have been revised or extended. Chapter 5 on machine learning is presented for the first time in this thesis. Part II is completely new to this thesis, while Part III includes chapters 10, 11, and 12 from the licentiate dissertation. Chapter 10 and Chapter 11 are moderately revised, keeping the results unchanged. Chapter 12 is a modified and reduced version of the corresponding chapter in the licentiate thesis; it keeps two scenarios among four from the licentiate.

2. Hadron Physics

Hadrons are bound together by the strong interaction, and therefore they constitute the primary laboratory for strong processes. The PANDA will be the largest and most versatile hadron physics experiment ever built worldwide [2]. In the following, I will put hadron physics in the context of the Standard Model (SM). The SM is a quantum field theory (QFT) of three fundamental forces: electromagnetic, weak and strong force. In the following, a brief introduction to the SM, leading up the hyperon studies, is given.

2.1 The Standard Model

The SM is a relativistic quantum field theory (QFT) about fundamental particles and their interactions [3]. The SM is a non-Abelian gauge theory whose Lagrangian[1] is invariant under local gauge transformations. The underlying internal symmetries of the SM are represented by the gauge group $\mathscr{G}_{SM} = SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$ whose generators gives rise to twelve massless gauge bosons: eight gluons ($G^1, G^2, .., G^8$) representing $SU(3)_C$ and four *electroweak* bosons (W^+, W^-, W^0 and B) representing the gauge group $SU(2)_L \otimes U(1)_Y$, respectively.

The sub-sector of $\mathscr{G}_{SM}$ representing the strong interaction is $SU(3)_C$ that is invariant under local gauge transformations. The $SU(2)_L \otimes U(1)_Y$ represents the electroweak interactions that are spontaneously broken to the $U(1)_{EM}$ gauge group through the Higgs mechanism. As a result, the W^0 and B bosons mix together to produce in total three massive weak bosons (W^+, W^-, Z^0) and a massless electromagnetic boson (γ) representing weak and electromagnetic interactions.

In the field theoretical treatment of SM, the objects are quantum fields: *fermionic* and *bosonic* fields. The excitations of these fields give rise to the physical particles called fermions and bosons, collectively called elementary particles, that are observed in the universe. The fermions are further divided into quarks and leptons whereas bosons are divided into gauge/vector bosons ($g, W^{\pm}, Z^0, \gamma$) and a scalar boson (H). The SM particles are shown in Figure 2.1.

[1] *i.e.* containing fields and their interactions

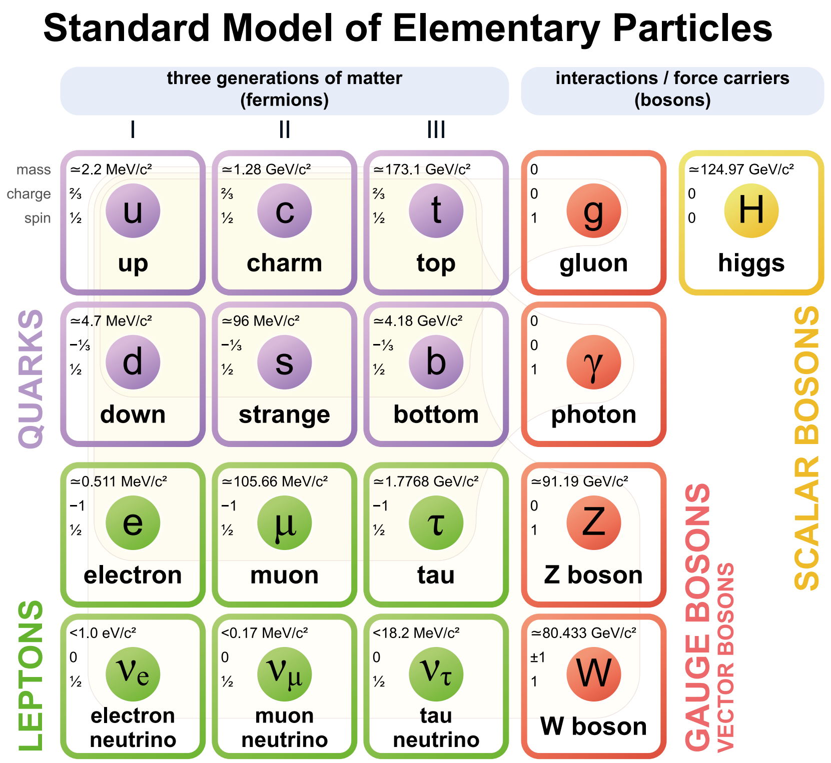


Figure 2.1. The SM of Particle Physics: Fermions (green and purple) and Bosons (orange and yellow). Image is taken from Ref. [4].

The Fermions

There are twelve fermions in the SM that are divided into two groups: *quarks* and *leptons*. All fermions have antiparticles with the same mass and spin, but with opposite electric charge and anticolor (quarks q) or lepton number (leptons l). The fermions are organized into three generations, where the higher generation particles decay into the lower generation ones through weak interactions.

The quarks come in six flavors: *up (u), down (d), charm (c), strange (s), top (t)*, and *bottom (b)* organized into the first (u,d), second (s,c) and third (t,b) generations. This means that top and bottom decay into (primarily) charm and strange, while charm and strange eventually decay into up and down. The quarks are the only particles that have fractional electric charges, and are shown in Figure 2.1.

The leptons come in both charged and neutral types. The charged leptons (l) are massive whereas their associated lepton neutrinos (ν_l) have negligible mass. Other than neutrinos, the lightest leptons are the stable electron (e^-) and its anti-particle, positron (e^+). The muon (μ^-), tau (τ^-) along with their antiparticles (μ^+, τ^+) are unstable particles and decay to electron and neutrinos[2]. The leptons are shown in Figure 2.1.

The Bosons

The bosons, or gauge bosons, are force particles responsible for interactions between fermions and bosons. For example, the electromagnetic interaction between two charged leptons occurs by the exchange of a virtual photon (γ) which is the gauge boson of the electromagnetic force. In a similar way, two quarks interact with each other through the strong interaction by the exchange of gluons (g) which are the mediators of the strong force. The weak interaction is mediated by the exchange of massive weak bosons *i.e.* W^+, W^- or Z.

Table 2.1. *The* bosons *and their properties in the SM [5].*

Interactions	Force Carrier	Mass (GeV/c^2)	Quantum Field Theory
Electromagnetic	γ	0	Quantum Electrodynamics
Electroweak	$W^\pm, Z^0$	$80.1, 91.2$	Electroweak Theory
Strong	g	0	Quantum Chromodynamics

Each force has an underlying physical theory that explains its aspects in great detail. The electromagnetic and weak forces have been unified under the Electroweak theory and together with Quantum Chromodynamics they constitute the SM.

2.1.1 Quantum Chromodynamics (QCD)

The field theory which explains the behavior of the strong interaction is called Quantum Chromodynamics (QCD). QCD is a gauge theory which is invariant under $SU(3)_C$ local phase transformations whose generators are represented by 3×3 matrices. This means the particle fields (quarks) should be a three-component field distinguish by their *color charge* with *red*, *green* and *blue* as its components. SU(3) local phase transformations means the rotation of states in color space. The local gauge invariance, however, can only be ensured by introducing eight gauge fields known as gluons. The generators of the SU(3) symmetry group do not commute, thus QCD is a non-Abelian gauge theory which allows directly for self-interactions of gluons, *i.e.* gluon-gluon interactions, as well as the interactions with the quark fields (quark-gluon interactions).

Due to color confinement, quarks can not exist freely in nature. The simplest colorless bound states can be formed either by combining a quark and an anti-quark commonly known as *meson* ($q\bar{q}$) or by three quarks commonly known as *baryons* (qqq). The lightest mesons are the pion isotriplet ($\pi^\pm, \pi^0$),

[2] *e.g.* $\mu^- \to e^- \bar{\nu}_e \nu_\mu$

two kaon isodublets (K^+,K^0 and $\bar{K}^0,K^-$) and the eta isoscalars (η,η'). Together they form a meson nonet. The lightest and most well-known baryons are the protons (p), neutrons (n), *i.e.* the nucleons, which form the atomic nucleus. They are the lightest baryons that form the bulk of visible matter in the universe. Together with $\Lambda,\Sigma^0,\Sigma^\pm,\Xi^0$, Ξ^- and the nucleons form the spin 1/2 baryon octet. The grouping of mesons and baryons as *nonet*, *octet*, etc is known as the Eightfold Way [6, 7]. The pseudoscalar meson nonet and spin 1/2 baryon octet are shown in Figure 2.2.

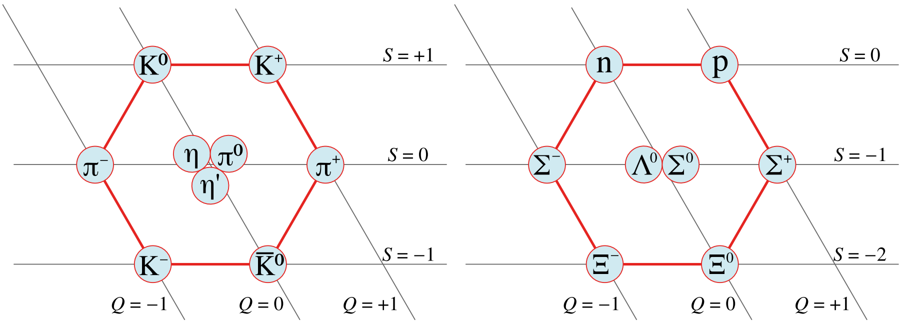


Figure 2.2. Meson Nonet and Baryon Octet, the Eightfold Way. Image credited to Wikipedia Images [8].

All baryons with strangeness different from zero are referred to as hyperons. Hyperons include the octet states $S = -1$ and $S = -2$ in the right panel of Figure 2.2. In Section 2.2, hyperons and their properties are discussed in more detail.

2.2 Hyperon Physics

Hyperons, denoted Y, are baryons where one or more up (u) or down (d) quarks are replaced by a strange (s), charm (c) and bottom (b) quarks. In this work, we restrict ourselves to ground-state strange hyperons only. In particular the lightest hyperon, the Λ, will be discussed in detail.

2.2.1 Strange Hyperons

The ground-state strange hyperons are Λ, Σ, Ξ (from the spin 1/2 baryon octet) and Ω (from the spin 3/2 baryon decuplet). The properties of strange hyperons such as mass, mean life-time, etc. are listed in Table 2.2.

The Λ hyperon has a quark composition *uds*, and is considered a heavier sibling of p and n[3]. Since the quark composition of Λ, Σ^0 and Σ^+ is very similar to that of the p and the n, they offer a tool to understand the role of

[3] $p \sim udu$, $n \sim udd$

Table 2.2. *The ground state strange hyperons and some of their properties. There are plenty of excited states of these baryons [5].*

Hyperon	m [GeV/c^2]	τ[s]	cτ [cm]	Dominant Decay, BR
$\Lambda\,(uds)$	1.115	$2.63\cdot10^{-10}$	7.89	$p\pi^-$, $(63.9\pm0.5)\%$
$\Sigma^+\,(uus)$	1.189	$8.01\cdot10^{-11}$	2.40	$p\pi^0$, $(51.57\pm0.30)\%$
$\Sigma^0\,(uds)$	1.193	$7.40\cdot10^{-20}$	$2.22\cdot10^{-9}$	$\Lambda\gamma$, $(100)\%$
$\Sigma^-\,(dds)$	1.197	$1.48\cdot10^{-10}$	4.43	$n\pi^-$, $(99.848\pm0.005)\%$
$\Xi^0\,(uss)$	1.315	$2.90\cdot10^{-10}$	8.71	$\Lambda\pi^0$, $(99.524\pm0.012)\%$
$\Xi^-\,(dss)$	1.322	$1.64\cdot10^{-10}$	4.91	$\Lambda\pi^-$, $(99.887\pm0.035)\%$
$\Omega^-\,(sss)$	1.672	$8.21\cdot10^{-11}$	2.46	ΛK^-, $(67.8\pm0.7)\%$

strangeness in electromagnetic and strong interactions. Among all hyperons, Λ is the most long-lived hyperon with a mean lifetime of $\tau = 2.6\times10^{-10}$ s. This implies that it flies a measurable distance - the decay length is $c\tau =$ 7.89 cm - after production. The Σ^0 hyperon is the only one among the ground-state hyperons that has a lighter partner with the same quark composition. This means that it can decay electromagnetically, in contrast to the others which must decay weakly. As a consequence, the lifetime of the Σ^0 is much shorter than for the other hyperons: with $\tau \approx 10^{-20}$ s its decay can be considered instantaneous on an experimental scale.

2.2.2 Hyperon Production

Hyperon-antihyperon pairs can be abundantly produced in antiproton-proton ($\bar{p}p$) annihilation reactions, a feature that will be exploited in the PANDA experiment. The production of hyperons is governed by the available energy and the mass of strange quark ($m_s \approx 93$ GeV/c^2). The mass m_s is close to the QCD scale Λ_{QCD} (~ 200 MeV) which corresponds to the confinement domain of QCD. Producing an $s\bar{s}$ pair requires energies of the same scale as Λ_{QCD}. Hence, one can say that by studying hyperon production we probe the strong interaction in the confinement domain.

Antiproton-proton annihilation with the subsequent production of an antihyperon hyperon pair, *i.e.* $\bar{p}p \to \bar{Y}Y$, provides a clear, two-body final state in contrast to production with electromagnetic, meson or proton probes, where hyperons typically needs to be produced with several recoil kaons and/or protons in order to conserve strangeness and baryon number. The final state is particle-antiparticle symmetric and the two-body final state is straight-forward to parameterize in terms of partial waves. In hyperon production studies, the variables of interest are the cross-section, scattering angle distribution and spin observables such as polarization and spin correlations [9].

In reactions like $\bar{p}p \to \bar{Y}Y$, where Y can be different hyperons such as Λ, Σ, Ξ and Ω, we learn about which degrees of freedom are relevant in this kind of processes: quarks and gluons, or hadrons [10, 11]. In Figure 2.3 three prototype reactions are shown in which $\bar{p}p$ annihilations result in the subsequent production of hyperon-antihyperon pairs with Λ, Ξ and Ω hyperons. In the figure, the most prominent decay chains are shown for each case.

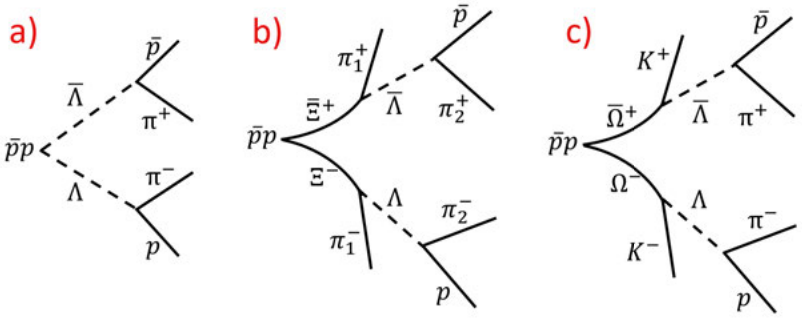


Figure 2.3. Hyperon (Λ, Ξ, Ω) production and decay channels at PANDA. Image is credited to Ref. [18].

By implanting hyperons into an atomic nucleus, hypernuclei can be formed. A hypernucleus is a state where a nucleon is replaced with a hyperon. Here, strangeness provides an additional degree of freedom which will help to understand *e.g.* neutron stars [12].

2.2.3 Hyperon Decays

All ground-state hyperons, except the Σ^0, decay through parity-violating weak decays. Hyperon decays, where the spin is experimentally traceable, offer a way to search for physics beyond the Standard Model via tests of Charge conjugation and Parity (CP) conservation. This in turn is necessary in order to understand the observed matter-antimatter asymmetry of the Universe [13].

The weakly decaying hyperons have lifetimes of the order of 10^{-10} s, which means that they travel up to several centimeters in the detector before decaying. Therefore, their decay vertices are located measurable distance from the interaction point. Many heavier hyperons, such as Σ^0, Ξ^0, Ξ^- and Ω^-, decay predominantly via intermediate neutral Λ hyperons making a reaction challenging to reconstruct from its final states. At PANDA, various strategies are under development to reconstruct hyperons. This thesis presents one of these approaches and is the first of its kind in PANDA.

2.3 Previous Hyperon Studies

In the past, experimental studies of hyperon production in $\bar{p}p$ annihilations have mainly been constrained to singly strange hyperons, *i.e.* Λ, Σ^0 and Σ^+. These studies are summarized in Figure 2.4 in terms of their production cross-sections.

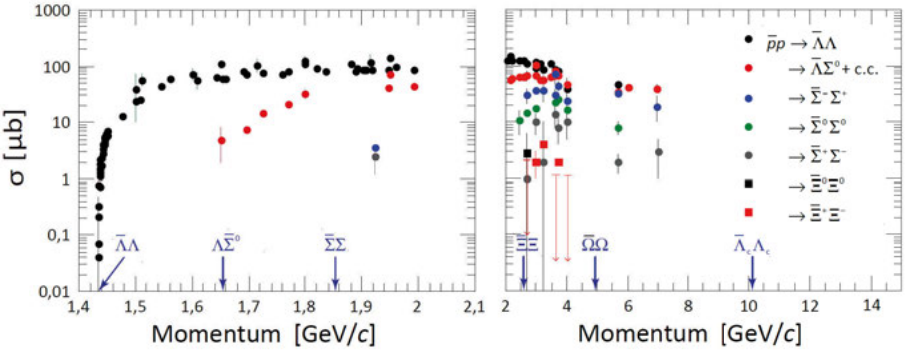


Figure 2.4. Measured cross-sections for different hyperon production channels at LEAR. Image is taken from Ref. [14].

The $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$ reaction had been intensively studied in the PS185 experiment at the Low Energy Antiproton Ring (LEAR). The PS185 was a fixed-target experiment just like the PANDA experiment where approximately 40000 exclusive events were analyzed at a beam momentum of 1.642 GeV/c [15, 16]. In these studies, it was found that the $\bar{\Lambda}$ is preferentially emitted in the beam direction in the in the center-of-mass frame *i.e.* the differential cross-section is *forward-peaking*. At $p_{beam} = 1.642$ GeV/c, the total cross-section was found to be 64.0 ± 0.4 (stat) $\pm$ 1.6 (syst) μb [16] which is of the same order as the non-resonant background. In addition, a complete spin decomposition of the reaction was performed. It was found that that the $\bar{\Lambda}\Lambda$ pair is almost exclusively produced in a spin triplet state [14].

At PANDA, $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$ reaction is a suitable benchmark channel for hyperon studies since it is well-studied at LEAR and hence well understood and parameterized. Being able to reconstruct Λ hyperons is therefore a prerequisite for all kinds of PANDA hyperon studies. In the recent past, several hyperon studies have been performed at Uppsala university. For example, Sophie Grape studied $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$, $\bar{\Lambda}\Xi^0$ reactions as part of her doctoral thesis [17], Erik Thomé studied multi-strange and charmed hyperon-antihyperon production at PANDA [9]. Both of these studies were performed with an older software package called Pandora, based on the software from the BaBar experiment. Pandora utilized idealized pattern recognition, and many of the detector features such as mechanical support structures between the active materials, were not implemented. More recently, Walter Ikegami Andersson studied spin observables of the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$, $\bar{\Xi}^+\Xi^-$ reactions using the newer and official PANDA software package PandaRoot. In addition, Ikegami Andersson developed track reconstruction and fitting tools using approaches based on *e.g.* Hough transformations and the Cellular Automaton for PANDA [10]. Jenny Regina studied detector signatures of the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$, $\bar{\Xi}^+\Xi^-$, $\bar{\Omega}^+\Omega^-$ reactions and developed a time-based track reconstruction algorithm, based on the Cellular Automaton, for PANDA [18]. However, as mentioned above, all physics simulation studies were performed using ideal pattern recognition and with an ideal point-like target. This thesis is a journey forward to understand hyperons under realistic conditions. First, reconstruction of hyperons using machine learning based track reconstruction algorithm in the Target Spectrometer of PANDA. A similar approach as the one discussed here has also been utilized in the PANDA Forward Spectrometer, in the thesis by Waleed Esmail [19]. Second, I perform simulation studies accounting for the fact that the planned cluster jet target will cause gas leakage in the beam pipe, which leads to a larger effective target than the designed one. In particular, I will investigate whether it is possible to suppress the background to hyperon-antihyperon pair production under such circumstances.

2.4 Scientific Questions for the Thesis

The scientific questions revolve around the realistic treatment of physics studies at PANDA. There are two main questions addressed in this thesis:

- How well do machine learning approaches perform in track reconstruction in low-to-mid-energy hadron physics experiments? Which kind of algorithm is suitable for this task? In particular, we are interested in tracks from low-momentum particles and tracks that do not originate in the beam-target interaction point.
- How does gas dissipation into the beam pipe affect the ability to distinguish signal from background? In particular, is it possible to treat reactions with the same final state particles, but with different point of origin?

To answer the first question, machine learning is used to reconstruct particles in the Straw Tube Tracker (STT) of PANDA: first muons and then hyperons and their decay products. This problem is handled in several steps. The process of track reconstruction is broken down into smaller tasks, i.e., edge classification and clustering. First, edge classification is performed using standard and geometric deep learning models. Second, weighted edges are clustered together to build tracks. This question is covered under Part II of this thesis.

To answer second question, first $\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$ signal reaction is reconstructed under ideal vacuum condition. To see the effectiveness of our analysis procedure, a non-resonant background, $\bar{p}p \rightarrow \bar{p}\pi^+ p\pi^-$, is also reconstructed. Using a similar procedure, signal and non-resonant background channels are reconstructed under realistic vacuum conditions. An Figure-of-Merit (FoM) is defined and calculated for both ideal and realistic cases. A comparison of FoM for ideal and realistic vacuum conditions will shed light on the effects of residual gas. This question is broadly answered in Part III of this thesis.

3. The PANDA Experiment at FAIR

The antiProton ANihilation at DArmstadt (PANDA) is a future experiment under construction at the Facility for Anti-proton and Ion Research (FAIR) in Darmstadt. PANDA will be an integrated part of the High Energy Storage Ring (HESR) at FAIR. The goal of this experiment is to study strong interaction in the confinement domain.

3.1 Facility for Anti-proton and Ion Research

The future Facility for Anti-proton and Ion Research (FAIR) is currently under construction adjacent to GSI Helmholtz Center for Heavy Ion Research (GSI) in Darmstadt. Both facilities are shown Figure 3.1.

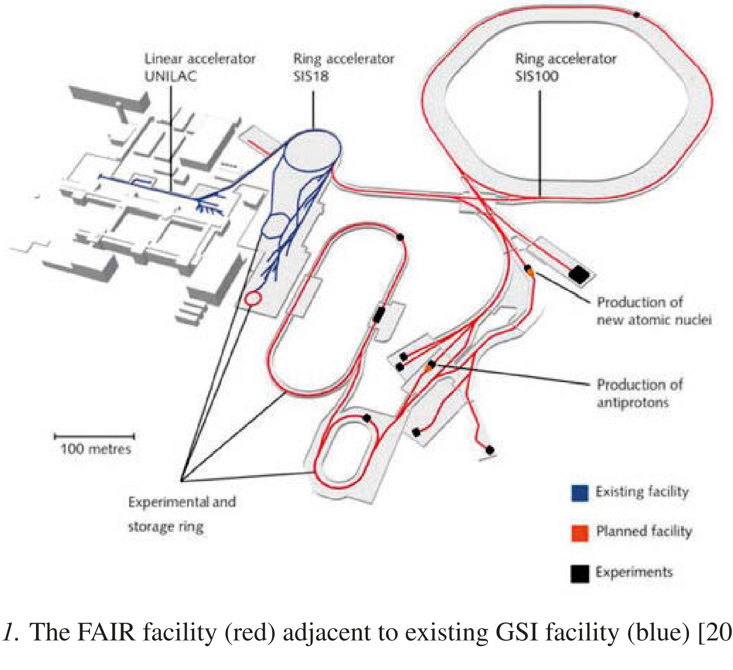


Figure 3.1. The FAIR facility (red) adjacent to existing GSI facility (blue) [20].

The GSI facility consists of a linear accelerator (UNILAC), a heavy-ion synchrotron (SIS18) with magnetic rigidity of 18 Tm and a storage ring (ESR). The UNILAC will accelerator heavy-ions to SIS18. Both the UNILAC and the SIS18 will be upgraded to serve as injectors and boosters to FAIR accelerators. FAIR consists of a proton accelerator (p-LINAC) as injector to the

SIS18, which in turn will feed the proton- and heavy-ion synchrotron SIS100. The SIS100 will have a magnetic rigidity of 100 Tm, and a circumference of $\approx$ 1100 m. To ackumulate and store the beam, the FAIR will have cooler-storage rings such as Collector Ring (CR), Recuperated Experimental Storage Ring (RESR), New Experimental Storage Ring (NESR) and the High Energy Storage Ring (HESR) for its experimental pillars such as

- PANDA - antiProton ANihilation at DArmstadt
- CBM - Compressed Baryonic Matter
- APPA - Atomic, Plasma Physics and Applications
- NUSTAR - NUclear Structure, Astrophysics and Reactions

The PANDA and APPA at FAIR will be dedicated to antiproton physics. The high energy antiprotons (up to 14.1 GeV) will be used in PANDA at HESR. A summary of FAIR can be found in Ref. [20].

3.1.1 Antiproton Target and Separator

The antiproton production will start by collecting primary protons from a proton source. The p-LINAC will accelerate these protons to the SIS18 synchrotron, where protons reach a kinetic energy of 2 GeV. The SIS100 will further accelerate these protons to a peak energy of 29 GeV. In the SIS100, protons will be compressed to a single bunch of approximately 25 ns duration ($\sim$ 7.5 m in length) with 2×10^{13} protons per bunch. Each bunch will be ejected to the *Antiproton Target and Separator* every 10 s (per cycle), where they will impinge on a stationary target, thus producing a cascade of particles through inelastic scattering.

3.1.2 Collector-Storage Rings

The antiprotons will be separated in *Antiproton Separator* and transferred to Collector Ring (CR) for pre-cooling. For a bunch size of 2×10^{13} protons at 29 GeV, approximately 1×10^{8} antiprotons are expected within the phase space acceptance (magnetic rigidity of 13 Tm corresponding to 3 GeV antiprotons) of the following antiproton separator and collector ring. Pre-cooled antiprotons will be injected to Recuperated Experimental Storage Ring (RESR) for accumulation by applying a momentum stacking scheme. Details on CR and RESR can be found in Ref. [21]. However, due to the development of global events, the CR is currently not funded while the RESR is beyond the initial phase of FAIR. Therefore, the CR and the RESR have been proposed to be replaced with the COoler SYnchrtron (COSY) and Antiproton Akumulator [22].

3.2 The High Energy Storage Ring

The High Energy Storage Ring (HESR) will be dedicated to physics studies with high energy antiprotons. The HESR will have the shape of a racetrack ring with a maximum beam rigidity of 50 Tm. It will consist of two 180° arcs and two 155 m long straight sections with a total circumference of 575 m. The schematic of HESR is shown in Figure 3.2.

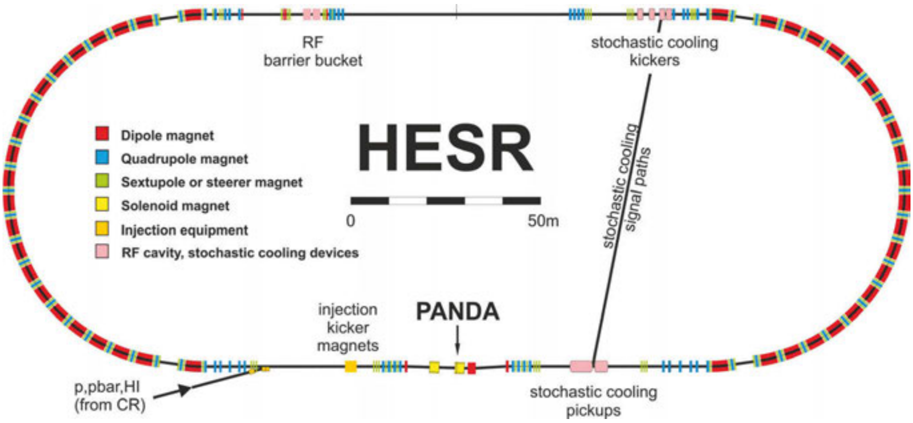


Figure 3.2. Overview of the High Energy Storage Ring (HESR) at FAIR.

The HESR will provide a quasi-continuous antiproton beam to the PANDA experiment in the momentum range of 1.5 GeV/c to 15 GeV/c (energy range: 0.83 GeV to 14.1 GeV) at an injection momentum of 3.8 GeV/c. The HESR will deliver the beam required for various physics studies planned for PANDA. The beam dynamics will be controlled by various systems, such as magnets, radio frequency (RF) cavities, and electron and stochastic cooling systems. In addition, the RF barrier bucket is intended for longitudinal beam accumulation to increase the intensity of the beam.

A typical cycle of operation includes the injection of an antiproton beam from CR/RESR into the HESR at a momentum of 3.8 GeV/c. The beam will then be pre-cooled and accelerated/decelerated to achieve desired beam momentum. The residual antiprotons will be recycled to initial injection momentum and merged with the newly injected beam from CR/RESR. Electron cooling or stochastic cooling methods will be used to attain the desired transverse emittance and momentum resolution. The beam recycling is an important feature of HESR to achieve the desired luminosity at PANDA.

The HESR will operate in two modes: the High-Resolution Mode (HR) and the High Luminosity (HL). The HR mode, accessible through the CR, will have beam momenta between 1.5 GeV/c and 9 GeV/c with a peak luminosity of $\mathscr{L} = 2 \cdot 10^{31}$ cm^{-2} s^{-1}. The beam will contain 10^{10} antiprotons with momentum resolution of $\sigma_p/p \leq 2 \cdot 10^{-4}$. Electron cooling is favored for beam momenta up to 9 GeV/c. The HL mode, accessible through the RESR, will have beam momenta of 1.5 GeV/c to 15 GeV/c with a peak luminosity of $\mathscr{L} = 2 \cdot 10^{32}$ cm^{-2} s^{-1}. The beam will contain 10^{11} antiprotons with momentum resolution of $\sigma_p/p \sim 10^{-4}$. Stochastic cooling will be available for beam momentum above 3.8 GeV/c. See details in Refs. [21, 23].

3.3 The PANDA Physics Program

During its lifetime, PANDA will explore several sub-domains of Quantum Chromodynamics (QCD) in the non-perturbative regime. As discussed in the previous sections, QCD has two special features, that corresponds to two distinct energy regions: asymptotic freedom (perturbative domain, $\alpha_s \to 0$) and confinement (non-perturbative domain, $\alpha_s \to 1$). The non-perturbative domain is a low-energy frontier that poses challenges to understand the hadronic states, since the building-blocks, *i.e.* the quarks and gluons, cannot be observed as free particles. The PANDA experiment will provide unique opportunities to study the non-perturbative QCD at the scale where quarks forms hadrons.

At PANDA, the $\bar{p}p$ and $\bar{p}A$ collisions will be used as diagnostics tools to study the hadronic states as well as their associated symmetries. PANDA is intended to run the different phases during its life-time. The physics program accessible during the Phase-I [11] is summarized in Figure 3.3.

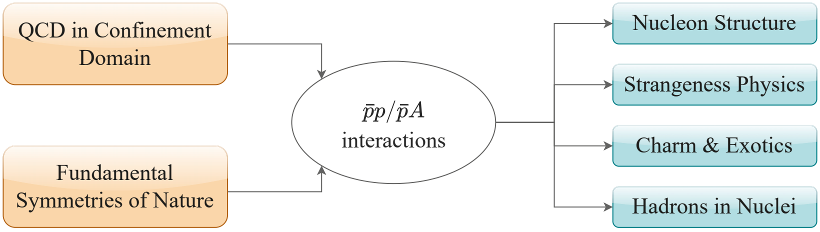

QCD in Confinement Domain
Fundamental Symmetries of Nature
$\bar{p}p/\bar{p}A$ interactions
Nucleon Structure
Strangeness Physics
Charm & Exotics
Hadrons in Nuclei

Figure 3.3. The PANDA Physics Program.

The PANDA physics program has four main pillars: *(i)* Nucleon Structure, *(ii)* Strangeness Physics, *(iii)* Charm and Exotics, and *(iv)* Hadrons in Nuclei. These areas are briefly described in the following sections.

3.3.1 Nucleon Structure

Here the main focus will be the inner structure of hadrons in terms of the distribution of quarks, as predicted by QCD and other phenomenological models. For this purpose, electromagnetic probes have been used to study electromagnetic form factors (EMFF) that quantify the inner structure of a hadron. The EMFF are functions of momentum transfer squared, *i.e.* q^2. For protons, space-like EMFF can be studied using the $e^- p \to e^- p$ scattering and reveal charge- and magnetization density [24]. The time-like form factors can give addition information about the inner structure. In many cases, time-like form factor are the only viable option. For example, in case of unstable particle one can not study form factors in the space-like region. Instead, they need to be extracted in the time-like region. The time-like region is divided into three regions: low-q^2, unphysical region and high-q^2. PANDA gives access to all time-like regions, in contrast to other experiments. The potential for measuring time-like form factor through certain hadronic reactions appears highly promising, as indicated by the findings of feasibility studies using $\bar{p}p \to e^+e^-$ [25] and $\bar{p}p \to \mu^+\mu^-$ [26] where the high-q^2 region is explored. Furthermore, in the low-q^2 region, Dalitz decays such as $B_1 \to B_2e^+e^-$ (where B denotes a baryon) can be used to study the time-like form factors.

3.3.2 Strangeness Physics

In the first phase of operation, a major area of interest will be hyperon-antihyperon production and multi-strange hyperon spectroscopy. The main goal at this stage will be to measure the cross-section, spin observables, and angular distribution of different $\bar{p}p \to \bar{Y}Y$ reactions [27]. Understanding hyperon production is crucial for deeper insights in hyperon properties, hyperon decays, CP violation, and hyperon structure [28]. In multi-strange hyperon spectroscopy, excited multi-strange hyperons can be produced in $\bar{p}p \to \bar{Y}^*Y + c.c.$ reaction, where Y and $\bar{Y}^*$ denotes hyperons and excited antihyperons [29]. One of the significant benefits is that the two-body final state makes it easier to perform partial wave analysis (PWA) and suppress background efficiently. Furthermore, it is possible to produce excited hyperons near the kinematic threshold which allows for studying final state interactions, which in turn can lead to the extraction of hyperon-antihyperon potentials [29].

3.3.3 Charm and Exotics

The quark model classify hadrons as either as mesons ($q\bar{q}$) and baryon (qqq). However, QCD allows any color singlet state. Thus, states like tetraquarks ($qq\bar{q}\bar{q}$), pentaquarks ($qqqq\bar{q}$), glueballs (gg, ggg) and hybrids ($q\bar{q}g$) should also exist. Such states are referred to as QCD-exotic states [30]. The search for exotic states and studying their properties is one of the main purposes of PANDA. During the last 20 years, several charmonium-like states have been discovered, for example, the so-called X(3872) discovered by Belle experiment in 2003 [31] followed by the discovery of other, similar states. These states are known as XYX states [30], as they do not fit into Quark Model. By measuring the cross-section line-shapes of these states, we will learn about their nature. For states which do not have $J^P = 1^-$ quantum numbers (*i.e.* the same as the photon), such line-shape studies are only possible in antiproton-proton energy scans [32].

3.3.4 Hadrons in Nuclei

The purpose of studying hadrons inside nuclei is to learn about hyperon-nuclei interactions. In nature, such interactions occur under extreme conditions in massive stellar objects like neutron stars where conversion of nucleons to hyperons becomes energetically favorable [12].

At PANDA, the antiproton beam can be used to study such interactions in several ways: (i) production of hyperon-antihyperon near threshold, (ii) antihyperon interactions with the nuclei, (iii) production of hyperatoms or hypernuclei. When the low-momentum antiproton beam interacts with the proton target, hyperons will be created with negligible momentum. The interactions of these slow-moving hyperons can be modeled by nuclear potentials. In many cases, hyperons are either captured inside the nuclear potential forming either hyperatoms or a hypernuclei. The interaction potentials in hypernuclei can be studied when an excited hypernucleus undergo γ-transitions to its ground-state. The gamma-rays from these transitions will be measured by Germanium detectors that are planned for PANDA. PANDA will provide opportunities to understand hyperon-nuclei interaction due to expected high hyperon-antihyperon production rates [33].

3.4 The PANDA Detector

The PANDA experiment is a general-purpose experiment under construction at FAIR. The design goal of PANDA is to achieve a nearly 4π acceptance, a high average interaction rate of up to 20 MHz, a high resolution in tracking and calorimetry, as well as the capability to identify various particles in a broad momentum range.

PANDA is divided into two magnetic spectrometers: the *Target Spectrometer (TS)* and the *Forward Spectrometer (FS)*. The target spectrometer will be based on a solenoid magnet surrounding the interaction point (IP). The TS is intended to perform measurements of particles emitted at large polar angles, many of them with relatively small momenta. The forward spectrometer will employ a superconducting dipole magnet optimized for particles emitted at small polar angles. These particles typically have large momenta. The complete setup is shown in Figure 3.4.

3.4.1 Target Spectrometer (TS)

The ***target spectrometer*** is hermetically sealed with sub-detectors arranged in the form of a barrel, a forward end cap, and a backward end cap. A uniform solenoid magnetic field strength of 2 T is applied. The barrel section covers angular range of $22^\circ < \theta < 140^\circ$. The forward endcap extends this range to 5° to the vertical and 10° to the horizontal planes whereas the backward endcap

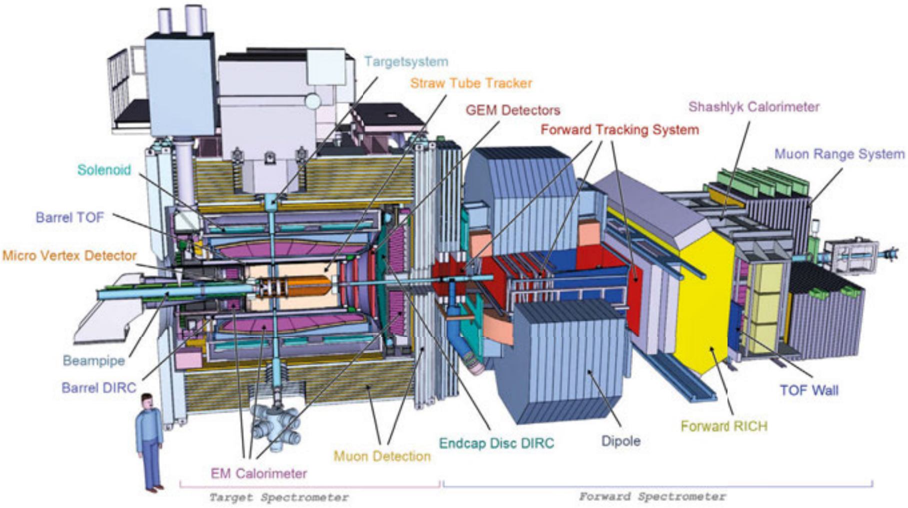


Figure 3.4. Overview of the PANDA experiment, the subsystem are labeled in both target and forward spectrometers.

covers the region in $145^\circ < \theta < 170^\circ$ range. The sub detectors in the TS are discussed in the following sections.

3.4.1.1 Solenoid Magnet

A solenoid magnet with a magnetic field of 2 T will surround the central tracking system in the target spectrometer [34]. The magnets bend charged particle trajectories subject to Lorentz force ($\vec{F}$) that is proportional to velocity (v):

$$\vec{F} = q\,(\vec{v} \times \vec{B}) \tag{3.1}$$

where q is particle charge, $\vec{v}$ is particle velocity and $\vec{B}$ is magnetic field. The force $\vec{F}$ is always parallel to the plane of $\vec{v} \times \vec{B}$. To keep a particle to circle of certain radius, one can equate Equation 3.1 to the centripetal force as follows:

$$\frac{mv_\perp^2}{r} = qv_\perp B \tag{3.2}$$

This means that the bending radius of the trajectory will be

$$r = \frac{mv_\perp}{qB} := \frac{p_\perp}{qB} \tag{3.3}$$

This means that the transverse momentum of a curve with a given radius is

$$\Rightarrow p_\perp[GeV/c] = 0.3 \times r[m] \times B[T] \tag{3.4}$$

Using Equation 3.4, one can find minimum transverse momentum (p_t^{min}) of particles needed for a particle to traverse fully traverse a detector. For example, minimum p_t for a particle to punch though the innermost and outermost layers of the STT detector, which are located at radii of 15 cm and 42 cm, respectively, is given as follows:

$$p_t^{min} > 90 - 252\,\mathrm{MeV/c} \qquad \text{(inner-outer layers)} \tag{3.5}$$

3.4.1.2 Micro Vertex Detector (MVD)

The Micro Vertex Detector (MVD) will be placed around the nominal interaction point. The maximum extension of MVD will be 150 mm in the radial direction and 230 mm in the forward direction from the interaction point. The combined polar angle coverage of both parts will be between 3° and 150°. A schematic picture of MVD is shown in Figure 3.5.

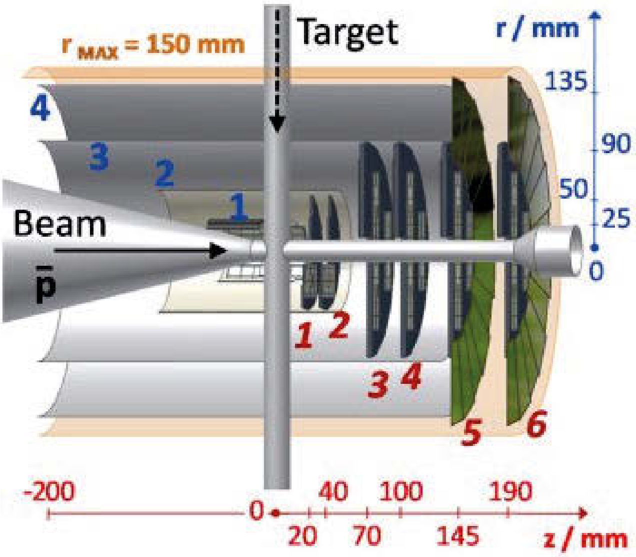


Figure 3.5. The cross-sectional view of MVD indicating barrel layers labeled in blue and the forward disks labeled in red. Image is taken from [35].

The purpose of the MVD will be to determine the origin, or vertex, of particles precisely, *e.g.* strange and charmed hadrons. Such particles have a relatively long lifetime (10^{-13}s for charm hadrons and 10^{-10}s for strange) (see Table 2.2 for lifetimes of famous hadrons) and will travel a measurable distance before decaying, thus creating a secondary vertex. In addition, the MVD will provide the time information of the event, which is necessary for track- and event reconstruction. Moreover, MVD will provide limited particle identification through energy loss (dE/dx) measurements.

The MVD will consist of semiconductor sensors in the form of pixels and strips. When a particle passes through these sensors, the electron-hole pairs are created inside the sensor due to the ionization induced by the passing particle. By applying an electric field across these sensors, the electrons drift toward the read-out electronics giving rise to a pulse. The location of a signaling sensor provides the hit position of a particle.

The MVD will be divided into four barrel layers and six forward discs. The two inner barrel layers and all forward discs are made from silicon hybrid pixel sensors of the size 100 μm × 100 μm. The other two outer barrel layers and the complementary extension to the last two discs will be equipped with Double-Sided Silicon Strip Detectors (DSSD). The DSSD will be rectangular for the barrel part and trapezoidal for the forward discs. The innermost and outermost barrel layers will be located at 25 mm and 135 mm radially. The innermost and outermost discs are housed at 20 mm and 190 mm along the beam axis, respectively. More details of MVD can be found in Ref. [35].

3.4.1.3 Gas Electron Multiplier Tracker

The Gas Electron Multiplier (GEM) tracker will be located in the forward direction along the beam pipe, covering the transition region between the target and forward spectrometers. The GEM tracker will cover the polar angle between 3° and 20°. Figure 3.6 shows schematic diagram of the GEM tracker.

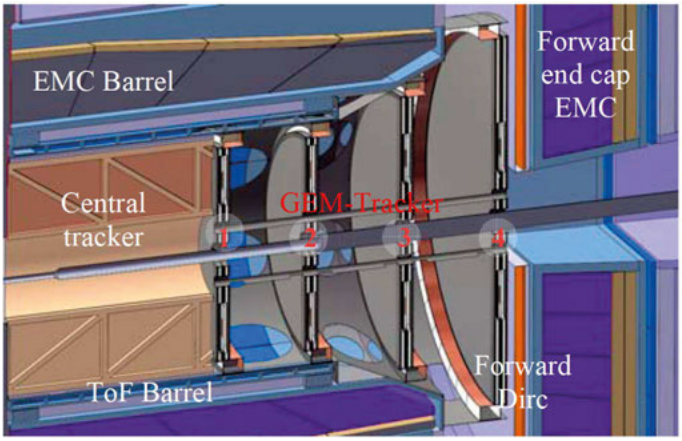


Figure 3.6. Position of the GEM stations along the beam axis. Image from Ref. [38].

The purpose of the GEM tracker will be to provide precise position information of particles emitted at small angles, escaping STT in the forward direction. The GEM tracker will consist of 3 to 4 stations. The stations will be made from gaseous micro-pattern detectors. A unique feature of GEM is the principle of electron multiplication inside an intense electric field through small perforated holes on the surface of a special substrate, a technique first introduced by F. Sauli [36]. When a particle traverses a GEM substrate, it creates electrons and ions through ionization. An electric field is applied across the substrate that intensifies surrounding the holes. The electrons pass through the holes, multiplying due to an intense electric field. These electrons reach the read-out electronics, and a pulse is generated that can be used to reconstruct particle position [37].

The GEM stations will be placed at 81 cm, 117 cm, 153 cm and 189 cm along the beam axis from the interaction point. It has been decided that only the 2^{nd}, 3^{rd}, and 4^{th} station will be kept inside the GEM. These stations have a diameter of 90 cm, 112 cm and 148 cm, respectively. The GEM planes are read out from both sides. A particle passing through a GEM station will give rise to two signals from both front and back readout channels. In general, particle tracks inside GEM will give rise to an even number of signals. However, if a track gives an odd number of GEM hits, either the particle is deflected from the surface or absorbed by the GEM plane. More details in Ref. [38].

3.4.1.4 Straw Tube Tracker (STT)

The Figure 3.7 shows the longitudinal (left) and cross-sectional (right) view of the Straw Tube Tracker. The STT covers a radial distance between 15 cm and 41.8 cm and have a length of 150 cm along the beam pipe. The geometrical polar angle coverage is $10° < \theta < 140°$ and it will be able to fully reconstruct the tracks of particles emitted at $\theta > 22°$. The purpose of the STT is to provide precise position information in x, y, and z. How this is done, will be discussed in detail in this thesis.

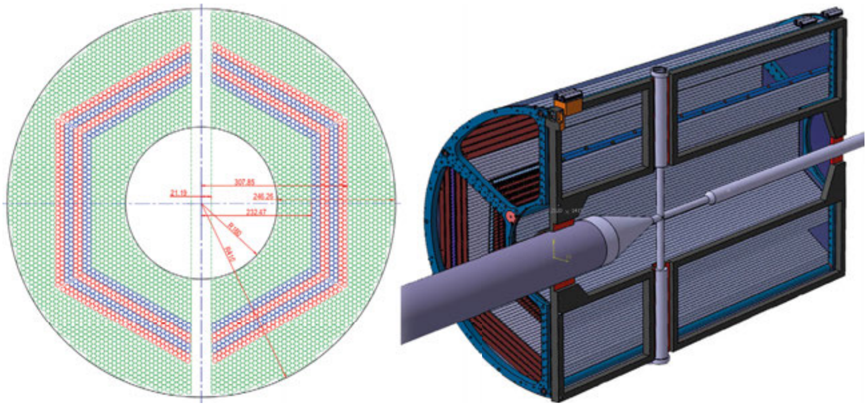


Figure 3.7. Cross-sectional (left) and Longitudinal (left) views of STT. Images are taken from Ref. [39].

The STT will be built from straw tubes, each straw is a single channel read-out drift tube consisting of aluminum (Au) coated Mylar tubes with a diameter of 10 mm and a gold (Au) plated anode wire in the center. These gas tubes will be filled with a quenching gas which is a mixture of Argon (90%) with CO_2 (10%). In total 4,636 such tubes are planned, arranged in hexagonal sectors with densely packed planar layers as illustrated in Figure 3.7 (right). There will be roughly 27 layers in the radial direction: 15 – 19 parallel layers along the beam axis (green tubes in Figure 3.7 (left)) and 8 skewed layers layers with a tilt angle of ±3° for longitudinal momentum reconstruction (red and blue tubes in Figure 3.7 (left)). The parallel tube layers aim to provide *xy*-reconstruction. The $r\phi$-plane position resolution is 150 μm and the longitudinal position resolution is 1 mm.

When a particle traverses a tube, it ionizes the gas along its path. These electrons drift towards the anode wire knocking more electrons along the way, creating an avalanche, under the influence of an applied electric field. The time to reach the anode is called drift time. The maximum drift time is $\approx$ 250 ns for an electron created near the outer wall of the tube. The collection of electrons at the anode wire create a signal/hit. The particle moving through the detector creates a trail of such hits which is called a *particle track*. The circle around the anode wire and going through particles point of closest approach to the wire is called isochrone. A signal from the anode wire indicates that a particle has traversed the tube anywhere along the isochrone. Determining which point on the isochrone crosses the true particle path requires the combination of several hits. Figure 3.8 shows a particle trajectory along with the isochrones.

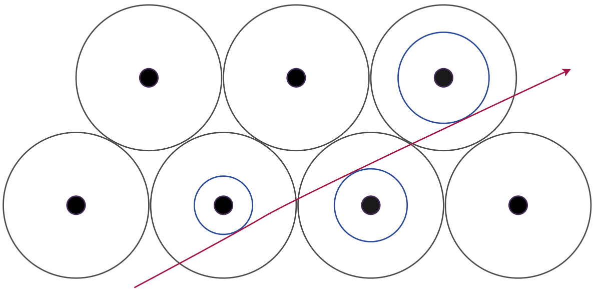


Figure 3.8. Isochrone radii in the Straw Tubes.

To make proper use of the isochrones, a reference time is needed that measures the time difference between the wire giving a signal and the particle traversing the straw tube. Such a time reference can be obtained by a faster detector such as the MVD or the Barrel Time of Flight [18]. Using isochrone can give position resolution of 150 μm in $r\phi-$plane of the STT. More details can be found in Ref. [39].

3.4.1.5 Barrel Time-of-Flight (BToF)

The Barrel Time-of-Flight (BToF) detector is one of the dedicated particle identification detectors. The BToF as well as the FToF are relative Time-of-Flight (ToF) counters *i.e.* they measure the arrival time of a charged particles rather than an absolute time difference. This is because there is no dedicated start detector in PANDA. However, complementary timing signals can be obtained from *e.g.* the MVD or by combining signals from the STT. To identify a particle, the momentum (p) and track lengths (l) of a particle are first extracted from the central tracking detectors. Second, the time-of-flight (t_i) is calculated for different mass hypothesis (m_i) according to Equation 3.6.

$$t_i = l \times \sqrt{(m_i/p)^2 + 1} \tag{3.6}$$

The time from Equation 3.6 is compared to the measured time obtained by the BToF to get the correct mass hypothesis. A normalized Gaussian is generated for each event, with the mean centered at the expected Time-of-Flight for a specific particle hypothesis. From this Gaussian, a probability density function is derived and evaluated using the measured Time-of-Flight. To determine the PID probability for each particle hypothesis, the probability density functions are normalized so that the sum of all probabilities is equal to one [40].

Figure 3.9 shows the drawing of Barrel ToF. BToF is built from scintillator tiles with dimension of $87 \times 29.4 \times 5$ mm^3 per tile. Each scintillator tile is read by four Silicon Photomultiplier (SiPMs) tubes on each side covering the 30×5 mm^2. The BToF will cover the angular acceptance of $22.5^\circ - 140^\circ$ and occupy the radial distance of 42 - 45 cm from the beam axis. To separate two different particle masses, *e.g.* π and K separation, an excellent time-resolution is needed. The BToF is designed to have a time resolution better than 100 ps. It will be placed just outside of the Barrel DIRC. See more details in Ref. [41].

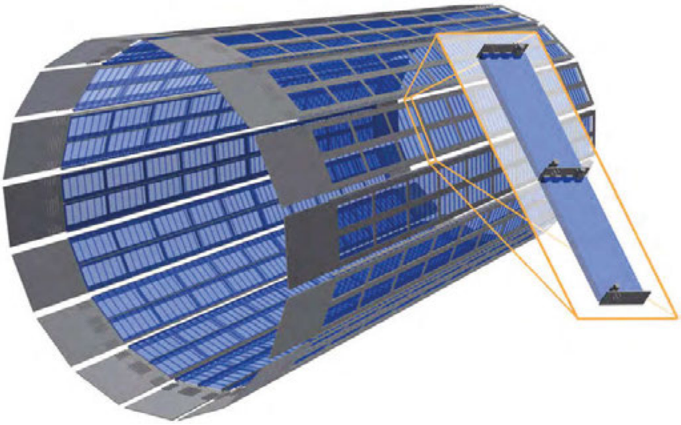


Figure 3.9. The BToF in the target spectrometer. Image is taken from Ref. [41].

3.4.1.6 Detection of Internally Reflected Cherenkov (DIRC)

The purpose of the DIRC counters are to provide particle identification. They will be based on the Detection of Internally Reflected Cherenkov (DIRC) light principle, *i.e.*, when a particle moves through a medium with a velocity greater than the velocity of light in that medium, then electromagnetic shock waves in the form of Cherenkov light are emitted. The emission angle of emitted photons and the number of photons depend on a particle's velocity. This information, together with the momentum of a particle *e.g.* from tracking detectors, can be used to calculate the particle's mass. In this way, a particle can be identified with a certain probability. Figure 3.10 shows drawing of of DIRC.

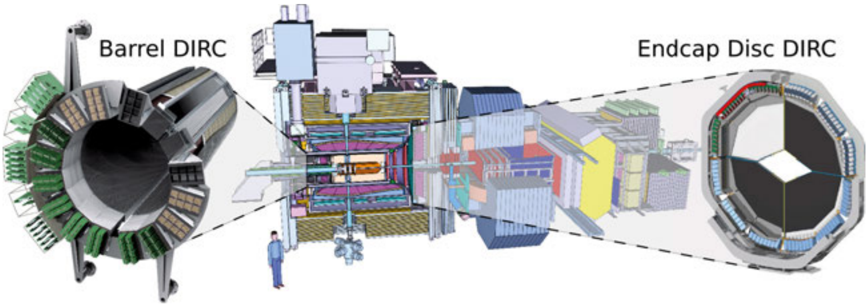


Figure 3.10. The Barrel DIRC and the forward endcap Disc DIRC in the target spectrometer. Image is taken from Ref. [42].

The DIRC, as well as the Ring Imaging Cherenkov (RICH), are Cherenkov detectors. The DIRC will be used in the barrel (Barrel DIRC) as well as in the forward endcap (Disc DIRC) of the target spectrometer. The Barrel DIRC will surround the interaction point at a radial distance of 50 cm, and have an angular acceptance of $22^\circ - 140^\circ$. It is intended to achieve clean π/K separation up to 3.5 GeV/c. The Disc DIRC will cover polar angles between $5^\circ - 22^\circ$, and will provide $\pi/K/p$ separation between $1-4$ GeV/c. It will be placed 194 cm downstream of the beam pipe. More details in Ref. [42, 43].

3.4.1.7 Muon System (MS)

The MS will be another dedicated PID detector designed explicitly for muons. Muons are almost 200 times heavier counterpart of electrons that can travel further in the detector due to their low ionization loss. The MS will detect muons in a broad momentum range together with the forward Muon Range System (MRS). The muon system will be built as alternating planar absorber (Fe) and sensor (Mini-Drift Tubes) layers. The purpose of the absorber is to absorb particles other than muons so that only muons reach the sensor layers. The mini drift tubes (MDTs) will be made of aluminum with gold-plated tungsten anode wires. The MS will be housed in barrel (2133 MDTs) and forward endcaps (618 MTDs). Details on the muon system (target spectrometer) and muon range system (forward spectrometer) can be found in Ref. [44]. The muon system as well as muon range system are shown in Figure 3.11.

3.4.1.8 Electromagnetic Calorimeter (EMC)

The EMC will be a compact calorimeter that has three sections: the backward endcap, the barrel, and the forward endcap. The EMC is shown in Figure 3.12, and its purpose will be to reconstruct photons, electrons, and hadrons with energies from a few MeV to several GeV. The EMC will measure the energy deposits of these particles through electromagnetic showers. The EMC will

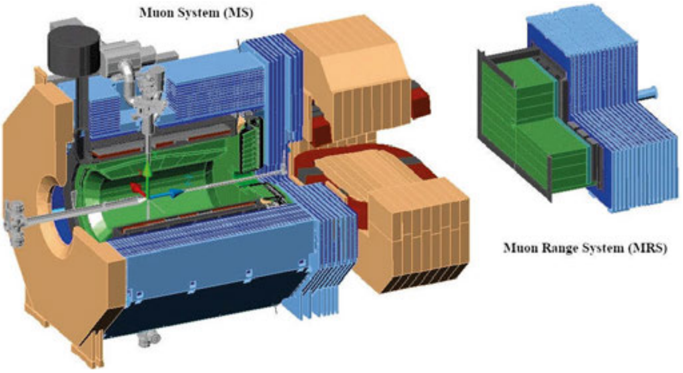


Figure 3.11. The Muon System and Muon Range System (highlighted blue) in target and forward spectrometers, receptively. Image is taken from Ref. [44].

be built from lead-tungstate ($PbWO_4$, PWO-II) crystals, an inorganic scintillator known for its high density, energy resolution, and fast repose. Using these crystals at temperature -25 °C instead of room temperature, can increase the light yield by a factor of four [17]. Each PWO-II crystal has a length of 200 mm. The total number of crystals will be 11360 in the barrel, 592 in the backward endcap, and 3600 in the forward endcap. Instead of photomultiplier tubes, special photo-detectors will be used to detect light: Avalanche Photodiode (APD) will be employed in the barrel part, whereas Vacuum Phototriodes (VPTs) will be used in the forward endcap. More details about the electromagnetic calorimeter can be found in Ref. [45].

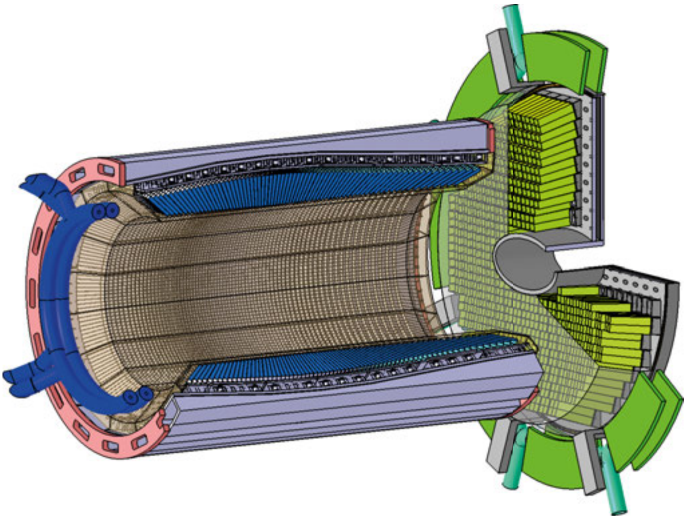


Figure 3.12. Drawing of the Barrel and Endcap EMC in the target spectrometer. Image is taken from Ref. [45].

3.4.2 Forward Spectrometer (FS)

The *forward spectrometer* consists of various planar-shaped sub-detectors arranged along the beam pipe. The dipole magnet provides the magnetic field covering the region in $\theta \leq \pm 10^\circ$ in the horizontal plane, defined from the beam pipe, and $\theta \leq \pm 5^\circ$ in the vertical plane. The dipole will provide maximum bending power of 1 Tm.

The deflected particles will be detected in the Forward Tracking Stations (FTS) for momentum measurements. For particle identification, the forward ToF (FTOF) will be used for relatively slow particles whereas the forward Ring Imaging CHerenkov (FRICH) for high momentum particles. In addition, a Shashlik-type Forward Spectrometer Electromagnetic Calorimeter (FSC) will measure energy deposits, whereas the Muon System and Muon Range System (FRS) will identify muons. The Luminosity Detector (LMD) is located upstream of the beam pipe. In the following, sub-detectors in FS will be discussed in detail.

3.4.2.1 Forward Tracking Stations (FTS)

The main purpose of FTS will be to detect particles at low polar angles escaping the target spectrometer. The FTS will consist of six planar tracking stations named *FT1, FT2,..., FT6*. These stations will be located along the beam axis at 295.4 cm, 327.4 cm, 394.5 cm, 438.5 cm, 607.5 cm, and 647.0 cm, respectively. One pair of stations (*i.e. FT1, FT2*) will be located upstream of the dipole magnet, one pair of stations (*i.e. FT3, FT4*) will be inside the dipole magnet, and one pair of stations (*i.e. FT5, FT6*) will be downstream of the dipole magnet. The stations inside the dipole magnet will bend charged particles, and the deflection will be used to measure the momentum. The acceptance of FTS will be ±10° in the horizontal plane and ±5° in the vertical plane with respect to the beam axis. The FTS is shown in Figure 3.13.

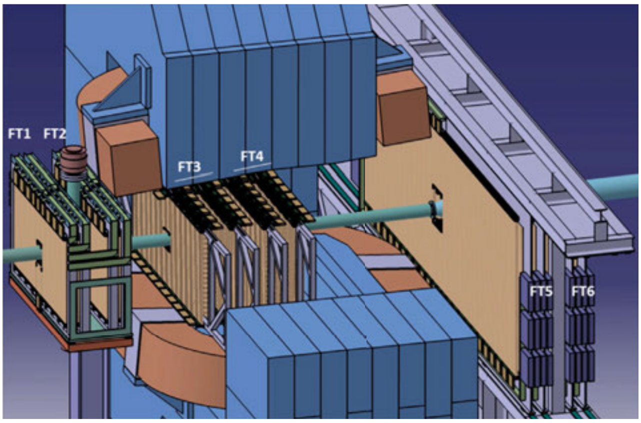


Figure 3.13. The FTS in the forward spectrometer. Image is taken from [47].

The layout of each tracking station will be based on straw tube modules. The design and detection principle of FTS will be similar to the STT with tube diameter of 10.1 mm. The straws are glued together in staggered double-layers either as 2 × 16 (32 straws) or 2 × 12 (24 straws), where the latter configuration surrounds the beam pipe. In each station, there are four detection planes made by stacking modules beside each other. For example, in FT1, each plane have 1 × 10 modules covering the 1338 × 640 mm^2. The placement of these modules in each plane will be based on their orientation with respect to the beam axis: modules in the first and fourth planes are in the vertical position, whereas the modules in the second and the third planes are inclined at $\pm 5^\circ$. The inclined modules are intended for the 3D reconstruction of particle tracks. The modules operate independently of each other which helps to repair or replace these modules whenever it will be necessary. In the first phase of operation of PANDA, the FT5 and FT6 will be replaced by the Outer Tracker from LHCb experiment [46]. More details about FTS can be found in Ref. [47].

3.4.2.2 Dipole Magnet

A large aperture dipole magnetic will be used together with the Forward Tracking Stations (FTS). The dipole magnetic will have a magnetic rigidity of 1 Tm and it will will cover the region $< \pm 10^\circ$ horizontal and $< \pm 5^\circ$ vertical to the beam axis. The dipole will be part of the accelerator lattice, hence, it will synchronously ramped up or down along with the ring magnets. More details can be found in Ref. [34].

3.4.2.3 Forward Time-of-Flight (FToF)

The FToF has the same purpose, operation principle and design as the BToF in the target spectrometer. However, it will be built as a planar detector and will be located at $z = 7.5$ m from the interaction point. The FToF will be made from the plastic scintillation slabs (*e.g.* Bicron 408) to cover a sensitive area of 5.6 m × 1.4 m. There will be 66 such slabs with variable widths: 20 innermost slabs surrounding the beamline will have a width of 10 cm, while the outermost will have a width of 20 cm. The slabs with shorter widths are intended to handle the expected high particle flux close to the beamline. As for the BToF, the design goal will be to achieve a time resolution better than 100 ps. The timing information from the FToF, combined with position information from the FTS, will provide excellent particle identification of hadrons such as π, K, and p by using the time-of-flight measurements. A good p/K and K/π separation are achievable up to momenta 4.3 GeV/c and 3 GeV/c, respectively. More details can be found in Ref. [48].

3.4.2.4 Forward Ring Imaging Cherenkov (FRICH) Detector

The FRICH will be similar in design and operation as of Barrel DIRC. It will be used to identify particles such as π, K and p with large momenta. The FRICH consists of two radiators such as silica aerogel and Freon (C_4F_{10}) gas with indices of refraction as 1.0304 and 1.00137 respectively. It will provide p/K and K/π separation in a broad momentum range of 2 GeV/c to 15 GeV/c. More details in Ref. [43].

3.4.2.5 Forward Spectrometer Electromagnetic Calorimeter (FSC)

The FSC is a sampling calorimeter indented to detect high momentum particles (photons and electrons) with high resolution and efficiency. The FSC will have a shashlik, or sandwich, design, which means that it will have alternating absorber-detector modules that sample the electromagnetic showers. The absorbers are made of lead, and the detectors of organic scintillating crystals. The incoming particles will interact with the lead plates creating an electromagnetic shower that will be detected in the scintillation crystals. Scintillation crystals will be readout with wavelength shifting fibers coupled with photomultiplier tube (PMTs). The FSC will cover polar angles of $\theta < 10^\circ$ horizontally and $< 5^\circ$ vertically with respect to the beam axis. The FSC will be placed at a distance of 7-8 m from the target just behind the FRICH. See more details can be found in Ref. [49].

3.4.2.6 Muon Range System (MRS)

The MRS is similar to the muon detector used in the target spectrometer (see Figure 3.11). The MRS will detect forward going muons. The basic principle is to absorb all particles except muons in an absorber, for example, iron (Fe), and then detect them using aluminum Mini Drift Tubes (MDT). The stopping power of the absorber will be approximately 1.5 GeV per meter of iron for relativistic muons at dE/dx = 2 MeV/g. The MRS will be built from interleaved layers of absorber material (16 Fe plates at 30 mm apart) and detector material (576 MDTs in total) layers. The MRS will be placed approximately 9 m from the target system. More details can be found in Ref. [44].

3.4.2.7 Luminosity Detector (LMD)

The LMD [50] will be placed at 11 m downstream of the target to measure the beam luminosity. The LMD, consisting of pixel sensors covering the region between 3 mrad and 8 mrad with respect to the beam, will specifically detect elastically scattered antiprotons at small angles where the Coulomb effects dominate. The cross-sections at these angles are known with high-precision and can therefore be used to calculate and monitor the luminosity.

3.4.3 Internal Targets

The target system of PANDA consists of internal targets, vacuum systems, and auxiliary systems. The choice of internal target is optimized taking into account several constraints, such as high density to meet designed high interaction rates. At the same time, the target must be thin enough to avoid interaction losses to the detector or DAQ dead time. The design goal is to achieve a target density of 4×10^{15} atoms/cm^{-2} or more. Due to these constraints, two types of internal targets are under development: the *Cluster-jet Target* and the *Pellet Target*. These targets can be made of hydrogen (H), deuterium (H_2) or any heavier (noble) gas up to the Xenon (Xe_{54}). The Figure 3.14 shows schematic of these targets.

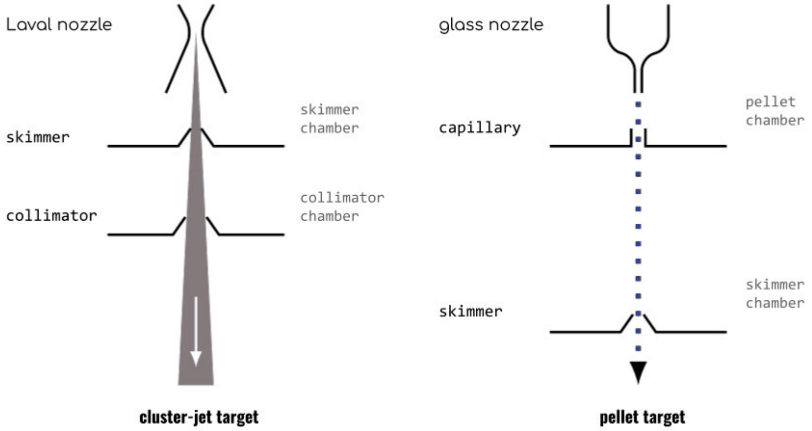


Figure 3.14. Cluster-jet target as well as pellet target at PANDA.

The design parameters of internal targets are outlined in Ref. [51]. I will also discuss the targets in the following sections.

3.4.3.1 Cluster-jet Target

In the *cluster-jet target*, a pre-cooled gas is forced to pass through a Laval-type nozzle (a convergent-divergent nozzle) (Figure 3.15. Due to adiabatic expansion, the gas cools down further and forms a stream of particle clusters due to condensation, hence the name cluster-jet target.

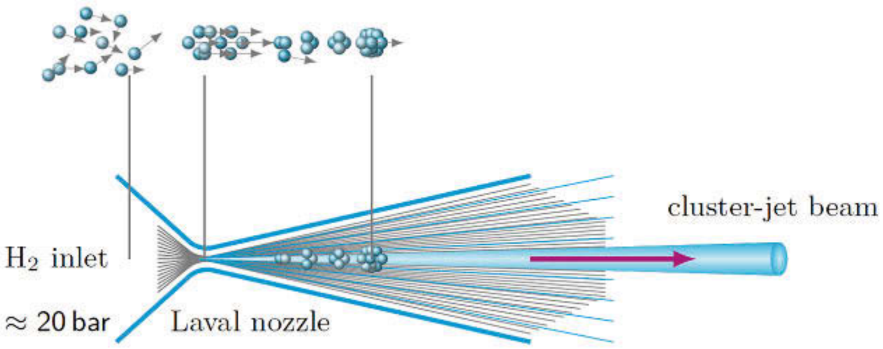


Figure 3.15. Laval-type nozzle for Cluster-jet Target. Figure is taken from Ref. [1].

The size of the clusters is strongly influenced by operating conditions such as temperature (T) and pressure (p). In addition, the nozzle diameter affects not only the size of the cluster but also its yield. For a hydrogen cluster-jet target, the typical temperature and pressure are $T_0 = 25 - 35$ K and $p_0 \leq 10$ bar, respectively, which will result in a cluster size of $10^3 - 10^5$ atoms per cluster [51]. In the Münster-type cluster-jet target, a pressure of $p_0 \leq 25$ bar results in liquefied hydrogen gas before entering the nozzle. The density of the cluster-jet then reaches 1×10^{15} atoms/cm^2 that is achieved more than 2 m behind the jet nozzle [51]. After leaving the nozzle, the jet stream is first shaped using a conical skimmer and a collimator before entering the cluster-jet vacuum pipe. The jet stream is later injected into an interaction point where interactions occur. A schematic picture of the cluster-jet target is shown in Figure 3.14.

Table 3.1. *The properties of the cluster-jet and pellet target beam for PANDA. The data is based on already achieved results, the table is taken from [51].*

	Cluster-jet Target	Pellet Target
effective target thickness	1×10^{15} atoms/cm^2	5×10^{15} atoms/cm^2
volume density distribution	homogeneous	granular
size transversal to p beam	$2 - 3$ mm	≤ 3 mm
size longitudinal to p beam	15 mm	≤ 3 mm
target particle size	nm - μm scale	20 μm
mean vertical particle distance	$\leq 10\mu$m	$2 - 20$ mm
target material	$H, H_2, ...Xe$(except He)	H, H_2, He, N_2, Ar

Table 3.1 shows the properties of both cluster-jet as well as pellet targets. The parameters listed below have already been achieved. One example is the cluster thickness of $> \times 10^{15}$ atoms/cm^2, which is better than the design parameters. Since the cluster-jet target is under constant development, one can expect improvements in various cluster-jet parameters.

3.4.3.2 Pellet Target

The *pellet target* consists of tiny frozen droplets (*i.e.* pellets) of hydrogen gas. The pellets are created by injecting cryogenic liquid into a gas, usually Helium, of the same element close to triple-point conditions through a vibrating nozzle. The vibration breaks down the liquid stream into small droplets of cryogenic gas, which freeze into solid pellets while passing through a thin tube called a vacuum injection capillary. The stream of pellets is further shaped by a skimmer leading to the interaction point. There are three primary reasons for using a pellet target: *(i)* high effective target density, *(ii)* very localized beam-target interaction point, and *(iii)* the possibility to track a pellet at the time of particle interactions. By fulfilling *(ii)* and *(iii)*, the position of pellet can be determined very precisely. The pellets have a diameter of 20 μm with an areal density of 5×10^{15} atoms/cm^2 [51] which are essential to attain required luminosities for various physics studies. A schematic picture of the pellet target is shown in Figure 3.14.

3.4.3.3 Target Beam Dump and Vacuum System

The jets or pellets are pushed to the interaction point where they collide with the antiproton beam. The remnants of the target is collected in the beam dump. However, the interaction inevitably leads to evaporation of gas from the target that dissipates to the beam pipe. There, the vacuum conditions are maintained using a differential pumping system which is equipped with hybrid *turbo molecular pumps*. The beam dump consists of three differentially pumped vacuum stages, each equipped with turbo molecular pumps and separated by orifices. These orifices are large enough to let pass the target beam, but as small as possible in order to minimize a back-flow of residual gas to the PANDA interaction point. There are several vacuum pumps to remove target gas inside the beam pipe. In Section 3.5, the effects of residual gas on the target, especially the cluster-jet target, as well as the solutions to this problem are discussed.

3.5 Effective Target Profiles

The cluster-jet target and the pellet target, have their pros and cons. In both target systems, an important issue under discussion is the accumulation of target gas inside the beam pipe which extends the effective volume of the target. That results in beam losses and beam heating, and affect the background conditions in physics studies. In the following, we will discuss how to mitigate the problem of residual gas inside the beam-pipe, with a focus on cluster-jet target.

The issue of residual gas can be mitigated by introducing a cryogenic pump located 3 m upstream of the interaction point (IP). Apart from the removal of residual gas, the main purpose of the cryogenic pump is to preserve the quality of antiproton beam that is essential for any physics studies. The standard or original geometry of the beam-pipe is denoted as the *NormalIP*.

The NormalIP is the original design of the beam pipe around the IP. Previously, a geometry called the BigIP was investigated, see *e.g.* [52]. The BigIP geometry has been discarded since it required to redesign the MVD detector. The beam pipe is narrowed down to 20 mm diameter at the IP giving a sufficiently small interaction region. This configuration is shown in Figure 3.16 with the cryogenic pump highlighted in yellow. In the next section, the vacuum simulation for NormalIP configuration is discussed.

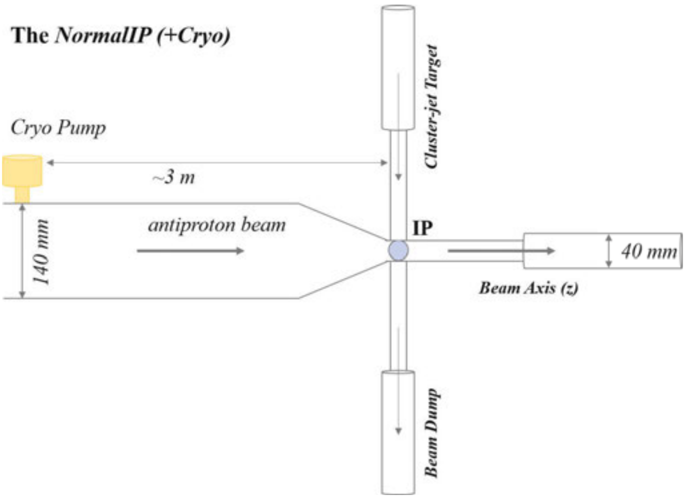


Figure 3.16. The NormalIP configuration, the original IP geometry at PANDA.

3.5.1 Vacuum Simulations

The implementation of vacuum systems are under investigation by the PANDA group at Westfälische Wilhelms-Universität Münster, Germany [1]. The NormalIP configurations with and without using a cryopump make two different scenarios. The vacuum simulation studies simulate the target and the residual gas inside the beam pipe. These scenarios are named as the following:

- NormalIP
- NormalIP+Cryo

The vacuum scenarios are based on *(i)* inclusion/exclusion of vacuum pumps to extract the residual gas, *(ii)* the corresponding density measurements of residual gas inside the beam pipe and *(iii)* the target profile based on density measurements.

The Figure 3.17 shows the density profile of both target and residual gas inside the beam pipe. One can see that effect of the cryopump downstream of the beam pipe is not significant. In order to perform realistic simulations for various physics studies, one should take into account the impact of target gas.

3.6 Data Analysis Tools

The software environment of the PANDA experiment consists of FairRoot[53] and PandaRoot [54], both based on the ROOT framework [55]. The FairRoot framework is common to all FAIR experiments *i.e.* NuSTAR, CBM, and PANDA. FairRoot contains the external software dependencies and logistical support to facilitate the operation of these experiments *e.g.* it provides detector geometries, event generation, particle transport, and event display. Fur-

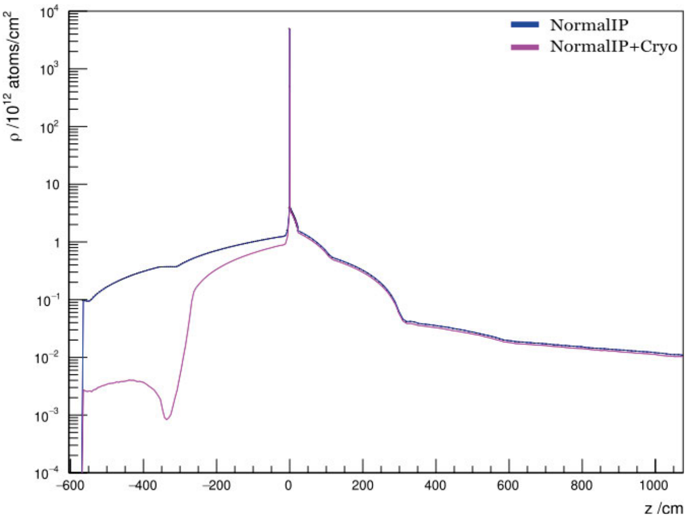


Figure 3.17. Vacuum simulation of target and residual gas with (magenta curve) and without (blue curve) cryopump, the *NormalIP*. The dip in magenta curve shows the location of cryopump. The data is provided by cluster-jet target group at Münster [1].

thermore, it provides an interface to software packages for other experiments. PandaRoot, on the other hand, contains packages and functions specific to the PANDA experiment.

3.6.1 The PandaRoot Analysis Chain

The complete data analysis chain of PandaRoot and FairRoot consists of several stages: simulation, digitization, reconstruction, and particle identification. This follows by analysis of data retrieved from previous stages. The steps are shown in Figure 3.18, along with a brief description of each step.

3.6.1.1 Simulation

In the simulation stage, the generation of particles produced in antiproton-proton annihilations is simulated, along with subsequent decays of these particles. In addition, this stage includes the propagation of particles through the detector and interactions with active and passive material. The antiproton-proton reactions are simulated using event generators, that take the dynamics of a given model and kinematics into account. The output from the event generators is in the form of particle four-vectors, decays and vertices of the particles for a particular reaction. Several event generators are available in PandaRoot such as EvtGen [56], UrQMD [57], Pythia [58] and FTF [59]. For this study, the EvtGen generator is used to generate $\bar{p}p \rightarrow \bar{\Lambda}\Lambda$ reactions. EvtGen is specialized in generating physical events based on specific decay models *e.g.* flat phase space, data parametrizations or other phenomenological mod-

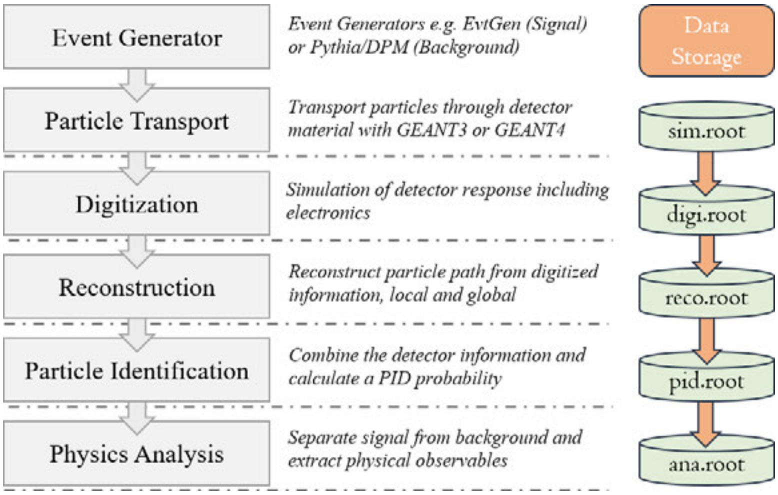

Event Generator
Event Generators e.g. EvtGen (Signal) or Pythia/DPM (Background)
Data Storage
Particle Transport
Transport particles through detector material with GEANT3 or GEANT4
sim.root
Digitization
Simulation of detector response including electronics
digi.root
Reconstruction
Reconstruct particle path from digitized information, local and global
reco.root
Particle Identification
Combine the detector information and calculate a PID probability
pid.root
Physics Analysis
Separate signal from background and extract physical observables
ana.root

Figure 3.18. The PandaRoot and FairRoot data analysis chain (top to bottom). The output of each stage is stored into a ROOT file. Image is adapted from Ref. [18].

els. Once the event kinematics is generated, the final state particles are propagated through full detector material using the GEANT4 package [60] which gives the precise information about the particle position, momentum, energy deposition, and time in various sub-detectors. This information is stored as MC truth, called *MCPoint*, with infinite precision. Detector effects such as noise are absent at this stage.

3.6.1.2 Digitization

A digitizer program is used to mimic the real detector response to construct realistic quantities in various sub-detectors. The MCPoints from the previous stage are converted into realistic hits, taking the granularity of the detector into account. Furthermore, energy deposits are converted into pulse heights. Each detector subsystem has its own specialized digitizer which coverts MCPoints in that detector as *digis* or *hits*. For example, in the case of STT, the MCPoints after digitization are stored as STTHits by the PandaRoot.

3.6.1.3 Reconstruction

At the reconstruction stage, the digitized hit positions from the previous stage are used to build tracks along the particle trajectories. In addition, hits in the calorimeter crystals are combined into clusters. In PandaRoot, the track reconstruction is a two-stage process: the *tracking finding* (often referred to as pattern recognition) and the *track fitting* (or track parameter estimation).

During track finding, hits positions from the digitizer are grouped to form clusters of hits known as track candidates by a track finding algorithm. In PandaRoot, a track candidate is stored as a *PndTrackCand* object. The choice of track finding algorithm may depend on whether the particle originates in the beam-target interaction point (primary vertex) or away from it (secondary vertex). The former puts a strong constraint on the track building process. Hence, an algorithm can be either a *primary* or a *secondary* track finding algorithm. Several track finding algorithms exist in PandaRoot that fulfill different purposes, *e.g.* tracking of particles from primary or secondary vertices, use for online or offline reconstruction, as well as idealized tracking using Monte Carlo information. In addition, different algorithms are used in the target spectrometer and forward spectrometer. In this section, the tracking algorithm used in the target spectrometer will be discussed. For a full track reconstruction in PANDA detector, a dedicated algorithm is needed in the forward spectrometer. Currently, no such algorithm has been established and therefore, the IdealTrackFinder is the default tracking algorithm in the forward spectrometer.

In the track fitting, the parameters of a track candidate are determined. In PandaRoot, a fitted track is stored as a *PndTrack* object which contains fitted parameters such as momentum, charge, and position at the first and the last hits of the track. The fitting procedure takes into account the material effects and energy loss inside the detector *e.g.* Bremsstrahlung, multiple scattering, ionization loss, etc. The most commonly used track fitting algorithm is the Kalman Filter [61], a recursive filter that predicts the current state of track, *i.e.* track parameters, based on previous measurements. This improves the efficiency of track fitting.

3.6.1.4 Particle Identification

At this stage, all available information about the detector signals within a given track is combined to identify the particle associated with the track. For this, one needs to find the mass (m) and charge (zq) of a particle. In general, the $z = 1$ for the stable and pseudo-stable particles measured in the detector. The mass can be calculated by using relativistic equation of motion $p = m_0 c\gamma\beta$ where $\beta = v/c$ and $\gamma = 1/\sqrt{1-\beta^2}$ if the momentum (p) is known.

The velocity v of charged particles can be determined by various detectors depending on the momentum range. For this purpose, information such as the time-of-flight ($\tau \propto 1/\beta$), the Cherenkov angle ($cos\theta_c = 1/\beta n$) or the energy loss (dE/dx) are utilized. Thus particle identification is subject to the type of detector used to measure the particle velocity.

A local PID probability (P) is independently calculated for each particle hypothesis e^-, μ^-, π^+, K^+ and p, including their antiparticles, in a particular sub-detector. For each hypothesis, the local probabilities are combined using a likelihood function. From Bayes' theorem, a global PID probability is calculated that can be accessed in the later analysis stage. These global PID probabilities can be further optimized during analysis [62], by using different criteria such as *Loose*: $P \geq 0$, *Tight*: $P \geq 0.5$ or *Ideal*: $P = 1$. In this thesis, the *Ideal* PID criteria are used which means that a matching is performed using Monte Carlo truth information. The correct particle is the assigned its identity by setting P=1.

3.6.1.5 Analysis

At this stage, the event selection (see, for example, Section 11.2) and parameter estimation, *e.g.* spin observables, are performed. The first task is to separate the signal, *i.e.* the reaction of interest, from the background. By using various selection criteria, the best particles are chosen to reconstruct the full event topology. The Rho analysis package, an integral part of PandaRoot, is used to perform combinatorial calculation, mass selection criteria, kinematic fits, etc. In the end, the final output is stored in ROOT files.

3.6.2 Machine Learning Tool Chain

For machine learning tasks, a comprehensive programming framework is needed. Such frameworks exist and are being developed in almost all programming languages. In C++, for example, one framework is called Toolkit for Multivariate Data Analysis in ROOT (TMVA) [63]. In Python, several frameworks exist such as PyTorch [64], TensorFlow [65], etc. In this thesis, PyTorch is used for data processing and neural network training.

4. Track Reconstruction

A particle beam is accelerated and collides either with another beam or with a target material. As a result, new particles are emitted from the interaction region in all directions. Several detectors are carefully placed around the interaction region to register their traces. Charged particles interact by ionization with the detecting material, giving position measurements along their trajectories. The first wave of particles is called primary particles, and their point of production is called the *primary vertex* or *interaction point*. The primary particles can in turn produce secondary particles, either by decaying or by interacting with the detector material. Secondary vertices from decays, which are discussed in some detail in this thesis, are denoted decay vertices. A snapshot of a collision and the subsequent interaction between particles and the detector, is called an *event* whose reconstruction is a prerequisite to understanding any physics process. The event reconstruction involves both track- and vertex reconstruction of all particles. In this thesis, however, we will focus on track reconstruction algorithms.

4.1 Track Reconstruction

Track reconstruction is a process of building particle trajectories in an event. After introducing some basics about track reconstruction in Section 3.6.1.3, I will now go into more detail.

Pattern recognition aims to reconstruct particle trajectories from the position measurements, called hits, provided by detectors. Hits are grouped into clusters or track candidates, which represent the passage of a particle.

There are two main categories of pattern recognition algorithms: *(i)* global methods, *(ii)* local methods. This categorization is based on how algorithms handle particle hits in the detector. The global methods process hits simultaneously *i.e.* independent of the order of the hits. The global methods are parallel and find track candidates all at once in an event. Examples of global methods are Conformal Mapping [66], Hough Transform [67, 68], Legendre Transform [69], Cellular Automaton [70, 71] and Artificial Neural Networks [72]. The local methods depend on predetermined *seeds*, *i.e.* a small segment of tracks taken as starting point, to find track candidates. These methods process hits sequentially to extend these seeds into track candidates. Local methods are sequential and find tracks one by one in an event. For example, Track Road, Track Following and Kalman Filter are local track-finding algorithms [73].

Global methods aim to avoid seeding bias during track finding as opposed to local methods. One can find an extensive exposition of these algorithms in Ref. [73, 74, 75].

Algorithms can also be categorized based on the origin of the particle trajectory. Let's first discuss the primary track finders. In general, particle tracks in the detector come from the beam-target interaction point, or the IP, and are called *primary tracks*. These tracks are either created in the primary interaction, or in strong or electromagnetic decays of very short-lived particles which on an experimental scale can be consider to decay instantaneously at the IP. A track originating from the IP puts a strong constraint in the track finding procedure since the combinatorics is drastically reduced. The algorithms that can only reconstruct primary tracks are called *primary track finders*.

For particles that are created in a secondary vertex, for example through a decay, a secondary track finder is required. An unstable particle with a longer life-time, for example a hyperon, flies a measurable distance from the IP before decaying, thus creating a secondary vertex. The reconstruction of such a track is much more complex, since the point of origin is unknown, hence unconstrained. Hence, the possible combination of hits that can form a track is very large compared to the case of a primary track.

4.2 Track Evaluation

Track evaluation is necessary to ensure the reconstructed tracks accurately represent particle trajectories. One way to assess the quality of track reconstruction is by calculating the overall tracking efficiency, here referred to as the physics efficiency and track purity. The physics efficiency provides information on how well the tracking algorithm can find all the particle tracks in the detector. On the other hand, the track purity provides information on how well the tracking algorithm can separate the signals produced by true particle tracks from noise or other background signals. To assess the performance of the tracking algorithm itself, a conditional tracking efficiency, here referred to as the technical efficiency, can be used in which a certain number of minimum hits are required to reconstruct a particle. These quantities will be defined using the evaluation scheme that closely follows the ATLAS community [78, 79]. Although PANDA also has its evaluation scheme, which is described in Ref. [18]. However, the PANDA scheme is not used in this thesis. Let us first define the following variables:

- $N_{\mathbf{particles}}(\mathbf{selected})$ is the number of generated particles in the detector, will be referred as *particles*. The label **selected** refers to particles generated within STT acceptance.
- $N_{\mathbf{particles}}(\mathbf{selected}, \mathbf{matched})$ is the number of particles matched to at least one reconstructed track.

- $N_{\mathbf{particles}}(\mathbf{selected}, \mathbf{reconstructable})$ is the number of generated particles that leave at least seven hits in the detector, will be referred to as the *reconstructable particles*.
- $N_{\mathbf{particles}}(\mathbf{selected}, \mathbf{reconstructable}, \mathbf{matched})$ is the number of reconstructable particles that are matched to at least one reconstructed track.
- $N_{\mathbf{tracks}}(\mathbf{selected})$ is the number of reconstructed tracks containing at least five or six hits, will be referred to as *reconstructed tracks*.
- $N_{\mathbf{tracks}}(\mathbf{selected}, \mathbf{matched})$ is the number of reconstructed tracks that are matched to a particle.

A particle is considered matched to a reconstructed track if more than *(i)* 50% of the hits in the reconstructed track belongs to the same true particle, *(ii)* 50% of the hits in the matched true particle are found in the reconstructed tracks. This is known as *two-way matching* scheme.

The physics efficiency is the fraction of particles that match at least one reconstructed track ($\varepsilon_{\text{phys}}$). The $\varepsilon_{\text{phys}}$ is given as follows:

$$\varepsilon_{\text{phys}} = \frac{N_{particles}(\text{selected, matched})}{N_{particles}(\text{selected})} \tag{4.1}$$

The technical efficiency is the fraction of reconstructable particles that match at least one reconstructed track ($\varepsilon_{\text{tech.}}$). The $\varepsilon_{\text{tech.}}$ is given as follows:

$$\varepsilon_{\text{tech.}} = \frac{N_{particles}(\text{selected, reconstructable, matched})}{N_{particles}(\text{selected, reconstructable})} \tag{4.2}$$

Finally, the track purity is defined as the fraction of reconstructed tracks that match a selected particle. The track purity is given by the following expression:

$$\text{Purity} = \frac{N_{\text{tracks}}(\text{selected, matched})}{N_{\text{tracks}}(\text{selected})} \equiv 1 - \text{Fake Rate} \tag{4.3}$$

In addition, the fake rate or ghost rate is defined as the fraction of reconstructed tracks not matching any particle tracks, while the clone rate is defined as the rate at which a particle track is matched to more than one reconstructed track.

One can choose any fraction ($> 50\%$) of hits during matching. In this thesis, we will use three different matching fractions (MF) such as 50% (loose), 75% (moderate), and 95% (tight). The 50% MF is a special case as it can result in the double matching of tracks; hence we add a small number, 0.00001, to the MF to avoid double matching.

4.3 Track Reconstruction in PANDA

Currently, various track reconstruction algorithms are under development. Only a few have been established within the standard PandaRoot software. Almost all, except the IdealTrackFinder, are realistic tracking algorithms, in the sense that they use the same information as will be available from real data. We will discuss them briefly in the following sections.

4.3.1 The IdealTrackFinder

The IdealTrackFinder is a reconstruction algorithm that utilizes the corresponding MC track information to build tracks. It takes the momentum of the first and last points (MC Points) from an MC track and applies Gaussian smearing on the momentum. By default, the Gaussian smearing corresponds to a mean of zero and a standard deviation of 5% of the MC truth momentum. The smeared momentum is used to build a reconstructed track.

The IdealTrackFinder can be employed in various parts of the PANDA detector, and different selectors are used for this purpose. In PandaRoot, each selector is implemented as a *functor*, and one can extend the functionality of a functor as needed. The predefined functors are listed below:

IdealTrackFinder – Functors

1. **Standard Track Functor** $\geq$ 4 MVD or $\geq$ 6 MVD+STT+GEM hits. It is the default functor in PandaRoot.
2. **Only STT Functor** $\geq$ 6 STT hits: To reconstruct tracks in STT only.
3. **FTS Track Functor** $\geq$ 6 FTS hits: Requires at least 6 hits in the forward tracking stations (FTS).
4. **No FTS Track Functor:** This functor excludes the forward spectrometer altogether *i.e.* no hits are included from the FTS.
5. **All Tracks Functor**: This functor is designed to accept everything for reconstruction as long as there is at least one hit available in the tracking detectors.

The IdealTrackFinder can be used for both secondary and primary tracks as it builds tracks based on the Monte Carlo truth information. Thus it represents a best-case scenario for realistic track finders.

4.3.2 The BarrelTrackFinder

The BarrelTrackFinder is a primary track finder, *i.e.* it assumes that the reconstructed track comes from the IP [80]. The BarrelTrackFinder reconstructs particle tracks only in the central tracking system (*i.e.* MVD+GEM+STT), hence the name BarrelTrackFinder.

The BarrelTrackFinder is a combinatorial track-following algorithm that takes the interaction point (IP) as its starting point and includes hits one by one by performing a circle/helix fit, thus gradually extrapolating the track.

4.3.3 The SttCellTrackFinder

The SttCellTrackFinder is a secondary track finder by design, as it does not rely on the interaction point to reconstruct tracks. The SttCellTrackFinder performs pattern recognition based on Cellular Automaton [81] and parameter estimation using Riemann mapping [82]. The SttCellTrackFinder in its earlier form builds tracks only in the STT detector [83, 84]. It has been extended to include the MVD and GEM detectors as well [85], though the extension needs further refinement before being used on a larger scale.

Currently, the SttCellTrackFinder is under development, and it can only reconstruct tracks in the *xy* projection of the detector, hence estimating the transversal component of track parameters. The longitudinal momentum reconstruction can however be done separately by the so-called PzFinder algorithm, which is explained in the coming section.

4.3.4 The PzFinder

The *PzFinder* is a set of algorithms for 3D track reconstruction. It extracts the longitudinal component of track parameters from the 2D input provided by an algorithm, for example the IdealTrackFinder or the SttCellTrackFinder. In the STT, the transversal track parameters are obtained by hits in the straight sections of the STT. However, a few layers of the STT are skewed at $\pm 2.9^\circ$ with respect to the $z-$axis. This can be exploited to extract longitudinal track parameters (z and p_z). For this purpose, the PzFinder was developed to complement the SttCellTrackFinder. It can work independently with another 2D algorithm, provided it contains sufficient hits to build a track.

The PzFinder uses the hits in the skewed straws to extract the $z-$position by isochrone alignment procedure. This procedure is explained in Ref. [10, 86]. As a result of this procedure, the PzFinder ends up with two possible values of the $z-$position that creates left/right ambiguity. To resolve this ambiguity, the PzFinder uses three different methods to choose the right solution: *(i)* Hough transformation [87], *(ii)* Combinatorial path finder [73], and *(iii)* Recursive annealing fit [73]. Among these three methods, Recursive annealing fit gives high performance in terms of efficiencies and purities greater than 95% [86].

4.4 Computing Challenges at PANDA

The computing challenge in track reconstruction mainly arises from the physics studies and the design features of an experiment. The tracks produced in PANDA, at a beam momentum of up to 15 GeV, have significantly different features from those from high energy physics experiments such as the Large Hadron Collider (LHC). In PANDA, we expect far fewer tracks per event ($N_{tracks} < 8$ in exclusive studies) as compared to LHC experiments. Moreover, PANDA tracks are often strongly curved, spiraling and overlapping. The reaction rate at PANDA is dominated by hadronic reactions where distinguishing signal from the background is complex. We will discuss two crucial features of PANDA that may pose a challenge during track reconstruction.

4.4.1 Interaction Rate

The interaction rate depends on the beam intensity and the target thickness, and is defined as the number of interactions (events) per second when a beam collides with another beam or target. The interaction rate can be calculated from the total production cross-section (σ) times the beam luminosity ($\mathscr{L}$).

$$\mathscr{R} := \sigma \cdot \mathscr{L}(t) = \frac{dN(t)}{dt}$$

For a beam-target experiment such as PANDA, the instantaneous luminosity is defined as:

$$\mathscr{L}(t) = N(t) \cdot \rho(t) \cdot f$$

where the $N(t)$ is the number of antiprotons ($\bar{p}$) in the beam, ρ is the target density and f is the beam frequency [88].

A high interaction rate often results in high track multiplicity, thus creating a dense signal environment inside the detector. At PANDA, the interaction rate will be up to 20 MHz. A reconstruction algorithm may require more computing time and resources if the number of hits per event is large.

4.4.2 Decay Signatures

The Figure 4.1 shows how the primary particles *e.g.* hyperons (Λ, Ω, Ξ) are produced in $p\bar{p}$ annihilations at PANDA experiment and further decay into secondary particles. A particular decay channel leaves a distinct signature (a cascade of trajectories) inside the detector.

These physics signatures or signals become complicated to reconstruct due to the presence of neutral particles (dashed lines), similar final states and *deep* decay topologies and multiple displaced vertices as shown in Figure 4.1: (a), (b) and (c). Moreover, the background in PANDA have features very similar

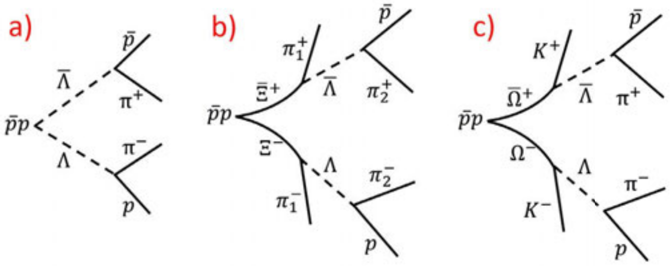


Figure 4.1. Hyperon (Λ, Ξ, Ω) production and decay channels at PANDA. The image is credited to Ref. [85].

to the reactions of interest. Thus, a straightforward discrimination of signal to background (S/B) is challenging. The combination of exclusive event reconstruction, complicated topologies, and low momentum / high curvature tracks is a special challenge for PANDA.

5. Machine Learning (ML)

Machine learning is a sub-field of artificial intelligence (AI) to study how machines, *e.g.* computers, can learn through *experience* just like humans do. The term *machine learning* is first coined by Arthur Samuel [89] in 1959. The following statement about machine learning is widely attributed to him.

"A field of study that gives computers the ability to learn without being explicitly programmed".

Machine learning consists of tools and techniques to extract meaningful information, such as patterns and relationships, by learning through data that may lead to making predictions, decision-making, and reasoning based on this information. To formulate a machine learning problem, one needs a data set and a mathematical model or a learning algorithm. Many real-world problems can be formulated in terms of machine learning *e.g.* function approximation, pattern recognition, computer vision, machine translation, object tracking, and identification are a few such examples. Each problem has distinct data that require a suitable mathematical model and a learning type.

5.1 Learning Types

The machine learning types encapsulate how a machine learning algorithm can experience or see the data. These types are named supervised, unsupervised, and semi-supervised machine learning. The latter is a combination of the first two types. We will briefly discuss the first two categories in this thesis (see Figure 5.1).

In **supervised learning**, the objective of a machine learning algorithm is to learn through a *labeled* dataset that comprises inputs and the desired outputs, or labels, hence the name labeled dataset. The learning algorithm tries to map the inputs to the output to develop a general understanding of the underlying relationship between the inputs and outputs. Over several examples, the algorithm can learn hidden patterns in the data. The algorithm will correctly predict the result when unseen data are provided. The whole process, known as model training, is repeated over several instances of the dataset until the machine learning algorithm reaches a certain performance [90]. Many algorithms exist under the supervised learning umbrella, such as support vector machines

(SVMs), decision trees (DTs), and artificial neural networks (ANNs). Problems such as regression, classification, and time-series prediction usually fall into the domain of supervised learning.

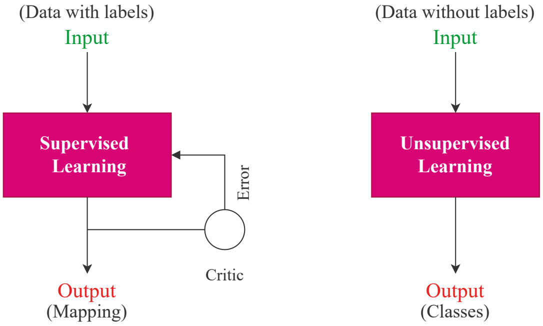


Figure 5.1. Schematic diagram of supervised and unsupervised machine learning.

In **unsupervised learning** learning, the data set is unlabeled *i.e.* it does not contain the desired output. Therefore, there is no way to supervise the learning algorithm. Instead, the learning algorithm attempt to learn patterns and relationships in the data to solve a problem. In cluster analysis, unsupervised algorithms self-organize these patterns or relationships into distinct classes based on the similarities or differences *e.g.* grouping objects based on their colors or shapes. The most commonly used algorithms for clustering are k-Means and DBSCAN [91]; for dimensionality reduction, common algorithms are principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). In detector simulation, algorithms like Auto-encoders (AEs) and Generative Adversarial Networks (GANs) are the most common models to generate new data. In addition, unsupervised learning is used in anomaly detection, dimensionality reduction, speech recognition, detector simulations, etc. Ref. [92, 93] provides more details on supervised and unsupervised learning.

5.2 Deep Learning

Deep learning is a sub-field of machine learning, comprised of deep neural networks as a base learning algorithm. A neural network consists of an input layer, one or more hidden layers, and an output layer. The architecture of a neural network can be understood in terms of the *width* and the *depth* of the network. The width defines the number of neurons arranged in a layer, whereas the depth describes the number of hidden layers in the network. In most cases, the terms *deep learning* and *deep neural networks* are used interchangeably. In the following, we will discuss deep neural networks and their applications in more detail.

5.2.1 Deep Neural Networks

The basic building block of a neural network is called an artificial neuron, a simplified computation model of a biological neuron. The first computational model, called Perceptron, was introduced in Refs. [94, 95] in the 1940s. This model mimics the functionality of a biological neuron by applying a *step* function, also known as activation function, on the aggregated weighted inputs from other neurons and generates a binary output using a particular threshold value. The weights of the inputs represent the strength of the connection between two neurons. In mathematical terms, the Perceptron model can be defined as follows:

$$\hat{y} = f\left(\mathbf{w}^T\mathbf{x} + b\right) \tag{5.1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ is the input vector, $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T$ is the weight vector, b is the perceptron bias and f is the step function. Schematic diagram of a Perceptron is shown in Figure 5.2.

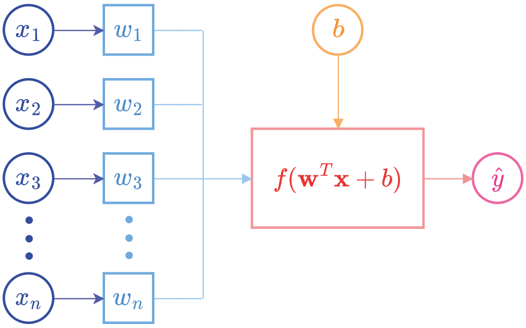


Figure 5.2. Computational model of a neuron, the Perceptron.

The modern version of artificial neurons, called sigmoid neurons, resembles Perceptrons with slight modifications. It is now possible to introduce more complex activation functions such as sigmoid, hyperbolic tangent (tanh), rectified linear unit (relu), etc. The activation functions are a non-linear mapping from inputs to outputs that yield a continuous output within an interval *e.g.* the sigmoid activation function produces a continuous output between 0 and 1, and in the case of tanh the output lies between -1 and 1.

Arranging several neurons together into layers results in a neural network. A simple feed-forward neural network has an input layer (X), several intermediate or hidden layers (HL), and an output layer (y). The networks with HL > 3 are called deep neural networks; hence the term *deep learning* is associated with these networks. The earliest examples of neural networks are Multilayer Perceptrons or Dense Neural Networks (DNNs). These networks are also called feed-forward, as the information flows from the first layer to the last layer without any feedback loops. A schematic diagram of the feed-forward network is shown in Figure 5.3.

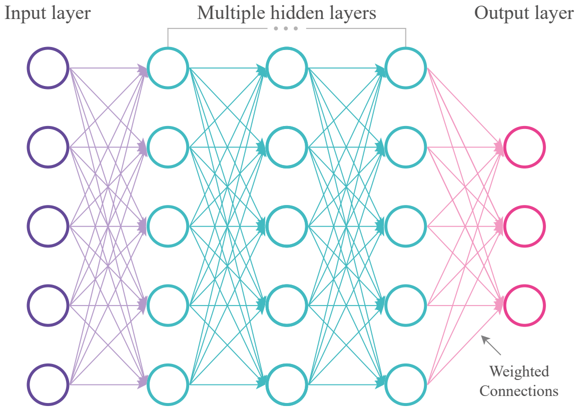


Figure 5.3. Schematic of a Dense Neural Network.

The goal of supervised DNN is to approximate some function f^* which defines a mapping $\mathbf{y} = f(\mathbf{x};\theta)$ and learns the value of the parameters θ that result in the best function approximation. Each layer in a network acts as a single mapping function chained together. For example, if functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ are chained together then $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ represents a composite mapping function. During the training process, the goal is to match then $f(x)$ to $f^*(x)$ by successfully reducing the error between the two [96]. The information flow from the input layer to the output layer is given by Equation 5.2:

$$a^{[l]} = g^{[l]}(W^{[l]}a^{[l-1]} + b^{[l]}) \tag{5.2}$$

where index l denotes layers in a network with values from $0,1,2,\ldots,L$. For $l = 0$, the $a^{[0]}$ denotes the network input (X) and $a^{[L]}$ denotes the output of final layer (L) known as network output ($\hat{y}$). $g^{[l]}$ is the activation function applied in a layer l, W is the weight matrix and $a^{[l-1]}$ is input to layer l.

The primary feed-forward networks have evolved into various classes. By adding a feedback loop, a new class of neural networks has emerged under recurrent neural networks (RNNs) [97]. The feedback loop allows RNNs to memorize the previous state of the network exhibiting a sequential or temporal behavior. Thus these networks are best suited for temporal problems such as times-series prediction, natural language processing (NLP), speech recognition, etc. Unlike feed-forward networks that process data in parallel, recurrent networks process data sequentially. However, the vanilla RNNs have a short memory *i.e.* they can not keep a record of distant states. In NLP, for example, memorizing earlier states is of particular significance where a sentence gives more sense rather than a single word. Long short-term memory (LSTM) [98] was developed to address this issue. A relatively simple version of LSTM is Gated Recurrent Unit (GRU) [99]. Although these models could memorize long sentences, they often failed to construct meaningful sentences. In encoder-decoder recurrent architectures, an attention mechanism is used to give context to the data, a mechanism that mimics cognitive attention. In computer vision, feed-forward networks have evolved into convolution neural network (CNNs) [100] that has found success in problems related to images, speech, signal processing, etc. Further models such as Transformers are developed [101] to process sequential data similar to RNNs. However, unlike attention-based RNNs, Transformers have a self-attention mechanism and avoid recurrent behavior.

Recent developments in deep learning have allowed the creation of composite models such as Auto Encoders, Variational Auto Encoders, Generative Adversarial Networks, etc., thus creating a zoo of neural networks[1]. The application of deep learning has expanded to various fields; for example, deep neural networks are widely used in bio-informatics, medicine, psychology, natural sciences, etc. Due to emerging computing technologies, more specialized models are being developed.

5.3 Geometric Deep Learning

Most of the problems discussed under deep learning (Section 5.2) fall into the Euclidean domain *i.e.* the data is structured in the form of a grid such sequences, images and so on[2]. However, many problems fall under the non-Euclidean domain, where data may or may not be structured, such as graphs and manifolds requiring specialized deep learning algorithms. As a result, a new sub-field of deep learning has emerged under the name of *geometric deep learning (GDL)*, a term first coined in Ref. [102].

The non-Euclidean data can be categorized as *grids*, *groups* (homogeneous spaces), *sets and graphs*, and *manifolds* (geodesics and gauges) [103]. For grids, different variants of RNNs and CNNs have been successfully used in various applications, as discussed in the previous section. For groups, spherical CNNs have been developed for the data from a sphere [104]. For set structured data, Deep Sets [105] and Set Transformers [106] are a few examples. For manifolds, standard CNN was used, and more specialized networks such as Geodesic CNN were successfully applied on the Riemannian manifolds [107]. In this thesis, we will focus on the second category *i.e.* graph-structured data and neural networks specialized for this data structure.

[1] See Neural Network Zoo

[2] Euclidean geometric structures: lines, planes, and other Euclidean spaces.

A graph $\mathscr{G}(V,E)$ is the combination of *nodes* or *vertices* ($V = v_1, v_2, ..$) and *edges* ($E = e_1, e_2, ...$). The graphs are either directed or undirected. In directed graphs, an edge e_{ij} is the connection or link (denoted '$\rightarrow$') from a source node v_i to a target node v_j. In undirected graphs, there is no sense of direction to establish a link between two nodes, so an edge e_k is the link (denote '$-$') between two unordered pairs (v_i, v_j) of nodes. The directed and undirected graphs are shown in Figure 5.4.

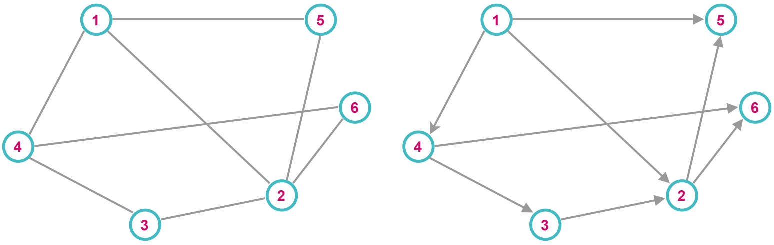


Figure 5.4. Undirected (left) and directed (right) graphs.

A conventional way to process graphs is node representation learning, where individual nodes are mapped to real-valued vectors, also called *embeddings* or *representations* [108]. However, one may lose topological dependence on the data, and the final result may depend on how these representations are created. For example, in many physics experiments, the location of the sensors inside a detector is of outmost importance as they give position measurements of a particle trajectory. Very often, these sensors are arranged in the non-Euclidean, or non-grid-like, geometry. In such cases, the data are better described as graphs, and specialized neural networks, known as graph neural networks (GNNs), have been developed that generalize the idea of node embeddings to graph representation learning. This thesis will focus on graphs and graph neural networks under GDL.

An early attempt to use a neural network directly on graphs was made in 1997 [109] leading to the works in Refs. [110, 111] and [112], the later adopted the name Graph Neural Network (GNN). Several variants of GNN have been developed so far. The most notable models are the Graph Convolutional Network (GCN) [113], Graph Attention Network (GAT) [115], and Message-Passing Neural Network (MPNN) [114].

The GCN uses convolutional operation, similar to CNN, on graphs. It updates a node's features by aggregating the neighboring nodes' features. The GAT borrows the idea of attention mechanism from Transformer models; it applies weights to give importance to certain nodes. The MPNN, on the other hand, uses a message-passing operation for node embedding.

Another model is the Interaction Graph Neural Network (IGNN), a physics inspired network that can reason about objects and their interaction in complex data [116]. In a later version, the IGNN was reformulated under MPNN framework [117]. Schematic of IGNN is shown in Figure 5.5:

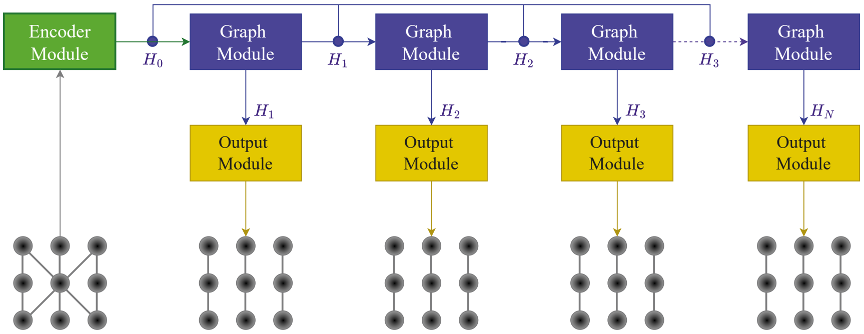


Figure 5.5. Schematic of Interaction Graph Neural Network (Interaction GNN).

The graph module is an IGNN that encapsulates a node and edge network. The encoder module transforms the input node features into a vector of hidden features and creates edge features from neighboring node features. In the graph module, aggregated neighboring edge features ($\bar{e}_i$) are passed to the node network f^v giving node output $v_i^{'}$, and neighboring node features ($v_{r_k}^{'}, v_{s_k}^{'}$, indices r, s denotes sender and receiver nodes) to the edge network (f^e) giving output $e_k^{'}$. This method is known as message-passing, which exchanges information between nodes and edges. This message-passing step is repeated several times. The output of the graph module is then passed through the output module for binary edge classification by using a certain loss function. As a result, each edge gets a score or weight. Figure 5.6 shows schematic of the graph module.

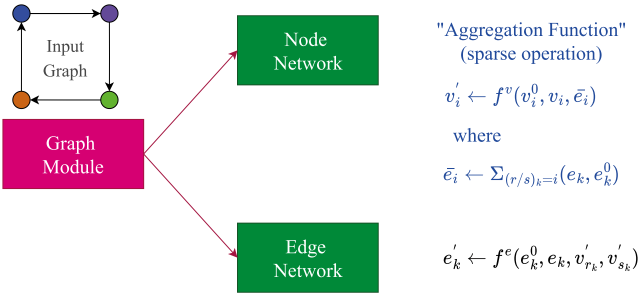

Input Graph
Graph Module
Node Network
Edge Network
"Aggregation Function"
(sparse operation)
$v_i^{'} \leftarrow f^v(v_i^0, v_i, \bar{e_i})$
where
$\bar{e_i} \leftarrow \Sigma_{(r/s)_k=i}(e_k, e_k^0)$
$e_k^{'} \leftarrow f^e(e_k^0, e_k, v_{r_k}^{'}, v_{s_k}^{'})$

Figure 5.6. Graph Module.

Previously in this section, I have attempted to give a concise and selective historical account of GNNs; however, it is intended for a partial review. The area of geometric deep learning has ballooned into hundreds of models, and covering all is beyond the scope of this thesis. The reader is referred to Ref. [118, 119, 120, 121, 122] for various GNN types and their applications. This thesis tests GCN, GAT, and IGNN under the MPNN framework.

5.4 Supervised Deep Learning

In this thesis, I will use supervised learning to train deep neural networks. In supervised learning, a model is trained on a labeled dataset in the form of features (inputs) and labels (outputs). Each example in the dataset contains inputs with their target value, and the learning algorithm maps inputs to the outputs. The data is presented to the mapping function one example at a time. By using a particular criterion, an appropriate error function, the algorithm's performance is checked, and the state (parameters) of the algorithm is updated for the next iteration. This process is called training and is repeated until the desired performance is achieved. A complete supervised training cycle is shown in Figure 5.7.

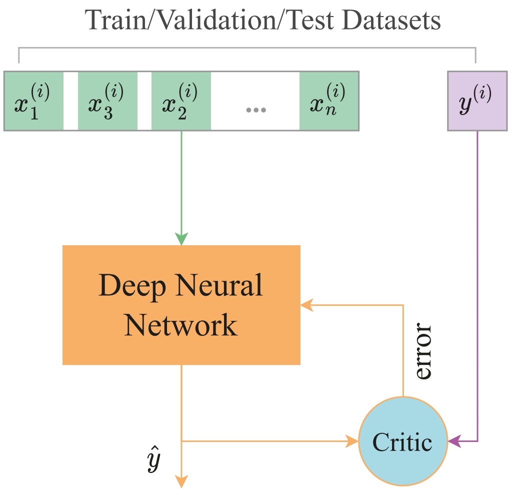


Figure 5.7. Supervised Deep Learning Cycle.

To see how well the algorithm has learned from exposure to the dataset, it is tested on an unseen dataset to see whether the algorithm has reasonably mapped the inputs to the targets. The dataset is divided into two subsets *i.e.* train dataset (X_{train}) and test dataset (X_{test}). Suppose it is desired to see whether the algorithm is generalizing (see Section 5.4.3) well during the training. In that case, a small fraction of the training dataset is reserved for validation (X_{val}). It is vital to make sure that X_{train}, X_{test} and X_{val} are drawn from the same batch or distribution of data, especially when data are generated at different times and settings, as the transient inconsistencies may induce bias in the data. For large datasets, ($\sim 10^6$) examples or more, the dataset is sliced into subsets containing 90% for training, 5% for validation and 5% for testing is regarded enough. However, other splitting ratios are equally plausible and may depend on the size of the dataset and the nature of the problem at hand.

5.4.1 Network Training

In this section, the training of neural networks is explained mathematically and highlights the important concepts as the discussion moves forward.

The dimensionality of the network is defined by the architecture (width and depth), activation functions, learning rate, regularization coefficients in loss, etc. These parameters are chosen on an ad hoc basis rather than learned, hence they are referred to as *hyperparameters*. There is no particular way to choose the hyperparameters, and it is subject to *hyperparameter tuning*.

The training process can be well understood in terms of the *forward pass*, *error calculation* and *backward pass*. We will describe the process for a single training example; for details on notations and data vectorization, see Appendix B.2.

5.4.1.1 Forward Pass

In the forward pass, input to the network is propagated through nodes of the networks. The weighted inputs are aggregated and a bias term is added. Then a non-linear activation function is applied (see working of a node in Figure 5.2) to get the output from each layer. The forward propagation equations for each training example are given as follows:

$$\begin{aligned} z^{[l]} &= W^{[l]}a^{[l-1]} + b^{[l]} \\ a^{[l]} &= g^{[l]}(z^{[l]}) \\ \hat{y} &:= a^{[L]} \end{aligned} \tag{5.3}$$

where $l = 0, 1, 2, ..., L$, $x = a^{[l=0]}$ is input vector, $a^{[l=1,2,...,L-1]}$ is output of hidden layers and $a^{[l=L]}$ denotes output of final layer ($\hat{y}$), it is the estimate of target value y. W is the weight matrix and b is the bias term, both regarded as network parameters (θ). The activation function in each layer is defined by $g^{[l]}$ applied to each layer l. Figure 5.8 shows a few activation functions used in neural network.

In general, $g^{[l]}$ can be different in each layer. Most commonly, the $relu()$ activation function is used in hidden layers to speed up the learning process while the $sigmoid()$ is applied to the output layer. In case of multi-class classification, the $sigmoid()$ is replaced with the $softmax()$ function.

5.4.1.2 Error Calculation

In supervised learning, the criterion used to assess the performance of network is defined by an error or loss function ($\mathscr{L}$). The most common form of loss function for regression or binary classification tasks is mean squared-error (MSE) function:

$$\text{MSE} := \mathscr{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2 \tag{5.4}$$

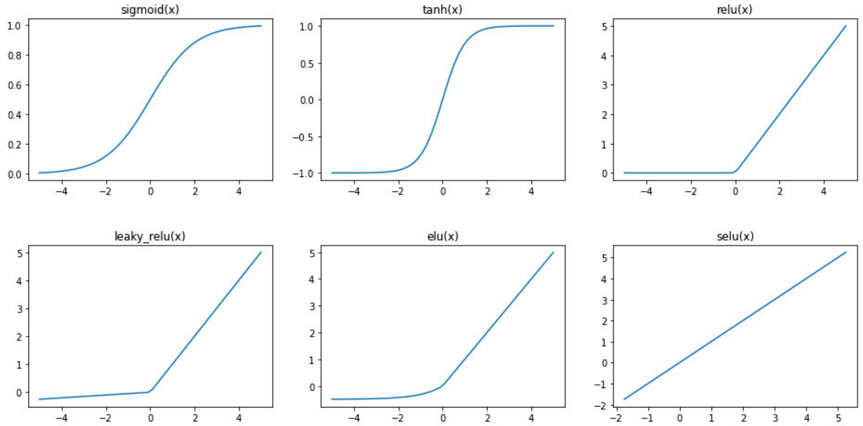


Figure 5.8. Most common activation functions used in deep learning.

Here $\mathscr{L}(\hat{y},y) \equiv \mathscr{L}(\theta)$, and $\theta = W,b$ are the network parameters. For binary classification problems, instead of squared-error function, the binary cross-entropy (BCE) function is preferred.

$$\text{BCE} \equiv \mathscr{L}(\hat{y},y) = -\hat{y}\log(y) - (1-\hat{y})\log(1-y) \tag{5.5}$$

The Equation 5.5 is a scalar valued and differentiable function which defines a convex surface with a global minimum.

The cost function (J) is the aggregation of loss over several training examples. One encounters two different cases when calculating the cost function: *(i)* batch cost (J_{batch}) over the full training dataset (m), and *(ii)* mini-batch cost ($J_{mini-batch}$) over a small fraction of training data known as mini-batches (b), where $b \ll m$. The cost functions are shown in the following equation:

$$J_{batch} = \frac{1}{m}\sum_{i=1}^{m}\mathscr{L}(\hat{y}^{(i)},y^{(i)}) \tag{5.6}$$

$$J_{mini-batch} = \frac{1}{b}\sum_{i=1}^{b}\mathscr{L}(\hat{y}^{(i)},y^{(i)}) \tag{5.7}$$

where "*(i)*" is index of an example in the dataset. The most common cost function is the mini-batch where batch size is chosen as power of 2 $(16,32,64,...)$ to benefit from the 8-bit structure of the memory, however, any number can be chosen as batch size.

5.4.1.3 Backward Pass

The backward pass is in fact a minimization or optimization stage. First, gradients of the loss function *w.r.t* the model parameters ($\theta = W,b$) are calculated using the chain rule. The gradients are calculated according to the following equation:

$$\nabla_{\theta}\mathscr{L}(\theta) = \begin{bmatrix} \frac{\partial \mathscr{L}(W,b)}{\partial W} \\ \\ \frac{\partial \mathscr{L}(W,b)}{\partial b} \end{bmatrix} \tag{5.8}$$

The chain rule essentially calculates the gradients from output layer towards the input layer. This way of numerical differentiation of the loss function is called a backpropagation algorithm [97]. The second step is the minimization of loss by updating the network parameters θ by moving a certain step in the direction of steepest descent defined by the negative of the gradient $\nabla_{\theta}\mathscr{L}(\theta)$. This procedure is known as method of gradient descent (GD) for optimization that is explained by the following equations:

$$\theta := \theta - \alpha \cdot \nabla_{\theta}\mathscr{L}(\theta) \tag{5.9}$$

where α is the step size known as the learning rate. The choice of learning rate (α) is crucial to achieve network convergence. There is no particular way to chose this parameter; one can either chose it on an ad hoc basis or use a grid-search method to find an optimal value.

In the gradient descent optimization, the gradients of earlier layers are the multiplication of gradients of later layers. That means the gradients gradually either vanish or explode during backpropagation, a problem known as vanishing or exploding gradients [123]. In case of vanishing gradients, earlier layers are hardly updated and optimization never converges to an optimum. On the other hand, an exploding gradient results in large parameter updates, leading to the divergence of optimization process. It has been shown that this issue is linked with activation functions used in a layer. A proper parameter initialization can diminish this effect [124, 125], see Appendix B.3.1 for details.

5.4.2 Network Optimization

In machine learning, optimization involves steps that lead to minimizing a cost function. The most common way to perform minimization in machine learning is through *gradient-based* optimization. However, non-gradient-based optimization exists, for example, Genetic Algorithm, Simulated Annealing [126], and Tabu Search [127]. We restrict ourselves to gradient-based optimization in this thesis.

5.4.2.1 Gradient-based Optimization

In Section 5.4.1, the gradient-base optimization is discussed for training a neural network. Based on the type of cost function, this method can be categorized as stochastic gradient descent (SGD), batch gradient descent (GD) and mini-batch gradient descent (BGD) if they use $\mathscr{L}(\theta)$, $J(\theta)_{batch}$, $J(\theta)_{mini-batch}$, respectively. We will refer to these variants collectively as vanilla gradient descent (GD).

During the optimization, one needs to choose a suitable learning rate (α) so that the network converges to a global minimum. An optimal value of learning rate can speed up the network convergence. A key challenge when optimizing a highly non-convex cost function is to avoid getting trapped into a local minimum. In that case, one needs to find a better way to choose the learning rate to get out of these minima. As argued in Ref. [128], the primary challenge is to escape a saddle point of a loss function rather than escaping a local minimum. To address this problem, new variants have been developed.

One solution is introducing a momentum term in the SGD that drives the algorithm in the right direction. The Equation 5.9 is modified as follows:

$$\begin{aligned} v_t &= \gamma v_{t-1} + \alpha \nabla_\theta \mathscr{L}(\theta) \\ \theta &= \theta - v_t \end{aligned}$$

where γv_{t-1} is the momentum term that drives the parameters θ, v_t is the velocity and γ (mass) parameter is usually chosen as 0.9. γv_{t-1}. The net effect is that the SGD escapes the saddle points and local minima due to momentum. A similar yet better algorithm is the Nesterov Accelerated Gradient (NAG) [129] that has a smart way of utilizing the momentum term mentioned above, see the following equations:

$$\begin{aligned} v_t &= \gamma v_{t-1} + \alpha \nabla_\theta \mathscr{L}(\theta - \gamma v_{t-1}) \\ \theta &= \theta - v_t \end{aligned}$$

The difference between vanilla momentum SGD and NAG is the point at which the gradient is measured. In NAG, rather than calculating the gradient at the current position before making a jump, it calculates the gradient to a *look-ahead* position where the jump will be made. This *futuristic* update prevents the optimizer from moving too fast, and instead slows it down while traversing a valley point.

Apart from the techniques mentioned earlier, there is another class of algorithms that are collectively called adaptive optimization algorithms due to their ability to adapt to the learning rate during training. Some of these algorithms are listed below:

- Adaptive Gradient (AdaGrad) [130]
- Adaleta (is an extension of AdaGrad) [131]
- Root Mean Square Propagation (RMSProp) [132]
- Adaptive Moment Estimation (Adam) (AdaGrad + RMSProp) [133]

- Nadam (Nesterov-accelerated Adaptive Moment Estimation) [134]
- AMSGrad [135]
- AdamW [136]

The study performed in Ref. [136] suggests that the most robust algorithm among these algorithms is the AdamW. It converges faster and works much better with the regularization method. The working principle of these algorithms is explained in Ref. [137]. In this thesis, the AdamW algorithm is used as an optimization algorithm.

5.4.3 Network Generalization

The purpose of machine learning is to learn hidden relationships in the data. To achieve that, we train a network on a limited dataset and hope it will perform equally well when new data is provided. The network's ability to learn the general understanding of hidden patterns in the data, and correctly predict when an unseen dataset is used, is called generalization.

During model training, the goal is to optimize the network to reach a global minimum. However, a small error doesn't translate into a generalization of the model. It means that the model is fitted closer to data rather than the underlying distribution. This phenomenon is known as *overfitting*. When new data are presented to the network, it fails to predict reliably. To identify overfitting, one needs to compare the training error and generalization error which is the expectation with respect to the underlying distribution of data. The discrepancy of these errors can be used to quantify the overfitting or generalization. Most often, the true distribution is not known. If this is the case, then we can simply estimate it by using a validation dataset during training.

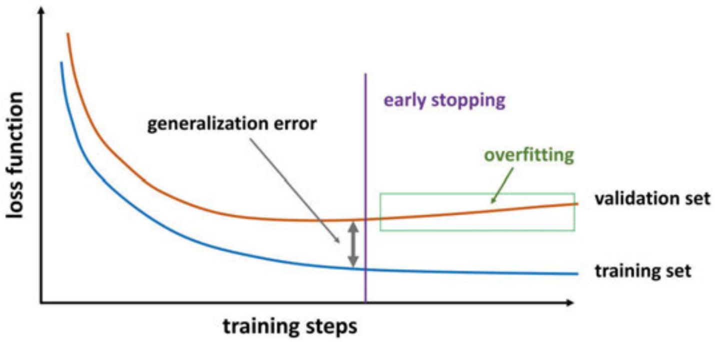


Figure 5.9. The generalization error of a network. The gradual increase in validation loss indicates overfitting (red box), though the training loss is still decreasing.

This is shown in Figure 5.9, where we see that the training training loss has substantially decreased, the gap between training and validation loss has grown over time. A large gap indicates overfitting and shows that our model has failed to generalize well after training. One way to mitigate overfitting is to collect more data; however, this is often not viable as it might be costly or time-consuming.

There are numerous techniques to combat overfitting, often called regularization methods. Some of these methods are listed below:

- Parameter norm penalties
- Activation normalization (Batch, Layer)
- Dropout and inverted dropout
- Early stopping
- Noise injection
- Data augmentation

Any modification to a model, that reduces its generalization error but not its training error, can be treated as a regularization method. Most popular among these are parameter norm penalties (l1 norm and l2 norm), dropout and early stopping. The l1-norm and l2-norm are used to penalize large weights; these norms are added to the loss function. The $l1-$norm regularization is given as follows:

$$\mathscr{L}(\theta)_{Reg.} \equiv \mathscr{L}(\theta) + \frac{\lambda}{2}\left\|\theta\right\|_2 \tag{5.10}$$

The $l2-$norm regularization is also called *weight decay* is as follows:

$$\mathscr{L}(\theta)_{Reg.} \equiv \mathscr{L}(\theta) + \frac{\lambda}{2}\left\|\theta\right\|_2^2 \tag{5.11}$$

where, λ is the regularization parameter. The layer index $[l]$ is dropped from the model parameters, θ, for the sake of simplicity. The $l2-$norm prevents the network to take large values of weights. The weight and bias terms are given as follows:

$$\left\|w^{[l]}\right\|_2^2 = \sum_{i=1}^{n^l}\sum_{j=1}^{n^{n-l}}\left[w_{i,j}^{[l]}\right]^2 \tag{5.12}$$

$$\left\|b^{[l]}\right\|_2^2 = \left[b^{[l]}\right]^2 \tag{5.13}$$

The bias term is usually optional. In case of mini-batch loss, the cost function with batch size b is given as follows:

$$J_{Reg.}(\theta) = -\frac{1}{b}\sum_{i=1}^{b}\left[\mathscr{L}_{Reg.}^{(i)}\right] \tag{5.14}$$

In the dropout method, a fraction of neurons in each training cycle is muted. The choice is based on the dropout probability (p) calculated during each cycle. Typical values are $0.2 < p < 0.5$. The dropout forces the network to redundant representations and adds noise to the process of feature-extraction. Dropout is only applied during training and switched off during validation. As a result, a large ensemble of sub-models is used during validation, thus improving network generalization. In case of early stopping, one monitors the validation loss and stops the training when the generalization error starts getting too large. Early stopping is sometimes discouraged for regularization as it violates the concept of *orthogonalization*. This means that it affects not only the validation process but also the training process. Orthogonalization is a way to independently adjust one parameter at a time. Due to orthogonalization, tools has been developed to avoid the *bias-variance trade-off* during network training.

5.4.4 Speeding up Network Training

In order to speed up the network convergence to an optimal state, several techniques are available. For example, choice of initialization of model parameters, and normalization of input and hidden features in a network. Some of these techniques are briefly discussed in Appendix B.3.

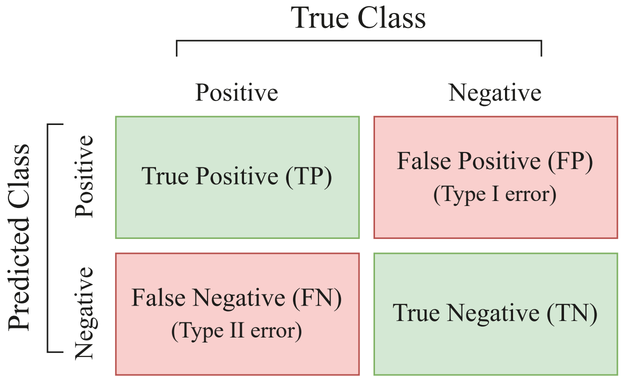
5.4.5 Hyperparameter Tuning

Hyperparameter tuning involves finding optimal values of hyperparameters. Hyperparameters differ from the model parameters (W,b) as they can not be estimated during model training but need to be carefully chosen. Hyperparameter tuning include finding optimal network architecture (nodes and layers), activation functions, optimization algorithms, regularization methods, etc. Finding optimal hyperparameters can be a tedious and time-consuming task. However, there are a few ways to choose them by, for example, performing a manual search, a grid search, a random sampling and a Bayesian search.

In manual search, one can use intuition or experience to choose hyperparameters. In grid search, one chooses a few hyperparameters to figure out what suits best, while in random sampling, hyperparameters are arbitrarily selected from prior distributions.

5.4.6 Network Evaluation

Several metrics are available to evaluate the performance of a binary classifier, for example Classification Accuracy, F1-score, Mean-Squared-Error (MSE), Receiver Operating Characteristic (ROC) curve, Area Under the ROC Curve (AUC). The choice of these metrics depends on the problem at hand. In my case, I chose AUC to evaluate my models. First, I look into the confusion matrix for a binary classifier that is shown in Figure 5.10:



		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP) (Type I error)
	Negative	False Negative (FN) (Type II error)	True Negative (TN)

Figure 5.10. A confusion matrix for a binary classification model.

where true positive (negative) means that the classifier predicted positive (negative), which is also truth (true). False positive (negative) means that the classifier predicted positive (negative), while it is false (false) in reality. The FP and FN are also known as type-I and type-II errors. The quantities from the confusion matrix can be used to calculate a variety of performance metrics *e.g.* true positive rate (TPR) and false positive rate (FPR) given as follows:

$$\begin{aligned} \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{FPR} &= \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \end{aligned} \tag{5.15}$$

TPR is called Sensitivity or Recall, and FPR is known as Positive Predicted Value or Fall-out rate. The ROC curve plots TPR *v.s.* the FPR at varying classification thresholds. The value of AUC ranges from 0 to 1 and gives an aggregate performance of the classifier at various thresholds (*i.e.* score cuts in our case). A high AUC value means high performance and vice versa. AUC has an advantage over other metrics that it is scale-invariant and threshold-invariant.

Here we used slightly modified parameters to calculate the ROC curve by using the edge classification efficiency (ε_E) and edge classification purity (p_E) defined as follows:

$$\begin{aligned} \varepsilon_E &:= \mathrm{TPR} \\ p_E &:= \mathrm{TNR} = 1 - \mathrm{FPR} \end{aligned} \tag{5.16}$$

5.5 Applications in HEP

Machine learning has been used for tackling various problems in the high energy physics (HEP) community. Most importantly, neural networks have been used for approximating complex functions (e.g. mass reconstruction), pattern recognition, signal to background suppression, triggers, jet/particle identification especially b-quark tagging, etc. More advance uses include anomaly detection, denoising of data, detector simulations, and so on. A comprehensive list of uses can be found in Ref. [138]. A list of reviews outlining the applications of neural networks in various domains of physics can be found in Ref. [139], especially in pattern recognition in [140, 141]. In this thesis, however, we will focus on the machine learning application to pattern recognition. The topic is covered broadly in Chapter 7, Chapter 8 and Chapter 9.

5.5.1 Pattern Recognition using Neural Networks

The application of ANNs as pattern recognition started in the early 1980s. One of the earliest applications of the Hopfield Network [142] for track finding was done by Denby [143], and Peterson [144], who independently worked out the procedure. Their adaptation of the Hopfield Network is nowadays known as the Denby-Peterson Network [145]. The idea was simple: every connection between two hits was attached to a Hopfield neuron that is activated if the hits forming the connection belong to the same track and deactivated otherwise. The interaction of neurons among themselves, *i.e.* the state of the network, was characterized by an energy function whose global minimum only activates the valid connections.

This network found its first realistic application in the ALEPH experiment [146], where the performance was found to be similar to that of classical track finders. Moreover, simple variants of the Hopfield networks were applied in the ARES spectrometer [147] and the ALICE experiment [148, 149]. During these applications, especially in the ARES experiment, it was found that the Denby-Peterson Networks have a limitation *e.g.* they do not scale with the density of tracks, and failed to separate tracks that are too close to each other. So new variants were searched and developed for track finding. Most notable are *elastic arms*, and *deformable templates* [150, 151].

The MLPs have found their application in high energy physics community during the same time period as the Hopfield Networks. Their use in track finding, however, was almost non-existent until recently. We will see later in Section 5.6 that how the feed-forward networks can be re-imagined for pattern recognition.

5.6 Novel Deep Learning Methods for PANDA

In Section 4.3, almost all of the reconstruction algorithms are based on the *classical* techniques except for SttCellTrackFinder, which uses cellular automaton for track reconstruction. Classical algorithms have some disadvantages; for example, many depend on the origin of tracks (*i.e.* primary track finders) and are combinatorial, thus making them relatively slow. To circumvent some of their issues and, in some instances, provide a complementary solution, machine learning algorithms, especially neural networks, are explored in this thesis. In this regard, I discussed briefly how neural networks have been used as pattern recognition in the past in Section 5.5.1.

One can re-imagine neural networks to classify the hit connection forming a track, similar to the way Hopfield Networks were used. However, instead of being a standalone solution, a probability of the hit connection can be obtained and used to build tracks. I will refer to hit connections as edges, and their classification as *Edge Classification*. For edge classification, special data will be prepared where each track is broken down into nodes and edges. The edges are either created randomly or by using a heuristic method where valid edges, *i.e.* edges belongs to a specific track, are assigned a probability of one. Otherwise, it will be assigned as zero. The naive approach is to use an MPLs as edge classifiers (Chapter 7). A rather sophisticated approach (Chapter 8, Chapter 9) is to use message-passing graph neural networks to classify edges that leverage additional information from the neighboring edges.

5.6.1 Related Work

Pattern recognition with neural networks has gained momentum in the recent years after the TrackML challenge [152]. The HEP.TrkX project [153] explored various deep learning models for track reconstruction *e.g.* RNNs and CNN on a toy dataset representing a planar detector geometry [154]. The Exa.TrkX project [155], the successor to the HEP.TrkX, used RNNs and GNNs in the space-point representation of HEP tracking data. In RNN, track seeds from true tracks are extrapolated using the LSTM. This model has two variants: *(i)* the Sequential hit predictor regression model, that achieved an efficiency of 99.93%, *(ii)* the Sequential hit predictor Gaussian model, that achieved an efficiency of 99.98%. With GNNs, Graph hit and segment classification were performed [156]—a rather serious application by Exa.TrkX used track-seed labeling with Embedded-space Graph Neural Networks [157]. Applying GNN for particle track and shower reconstruction problems [158], the same method is used for a full detector [159]. Other applications of the GNN can be found in Ref. [160, 161, 162, 163]. For PANDA, I have used GNN models for edge classification for track reconstruction in the Straw Tube Tracker, and presented preliminary results in Ref. [164]. The work presented in this thesis is a another step forward in using different models of graph neural networks for track reconstruction.

Part II:
Realistic Track Reconstruction in the PANDA Target Spectrometer using Deep Learning Techniques

6. Strategies of Deep Learning in PANDA Tracking

6.1 Problem Formulation

The goal in this part of the thesis is to develop track-building techniques based on deep learning (see Section 5.6) in the Central Tracking System (CTS) of the Target Spectrometer, where a crucial component is the STT (see Section 3.4.1.4). The track reconstruction method should have the following features:

- Be able to handle the non-Euclidean geometry of the STT
- Be agnostic of the origin of the tracks, *i.e.* be able to handle tracks from displaced vertices
- Have the capability to reconstruct particles with low momenta with reasonable efficiency

In addition, we need a framework that can be extended to other detectors, such as MVD and GEM. However, combining these detectors is a rather challenging task as these detectors comprise different geometries and detection technologies. In the past, an attempt has been made to combine STT with MVD and GEM for traditional algorithms, but it has yet to reach the operational mode [18].

In this work, I have focused on the straight sections of the STT and ignored the skewed ones. This means that we effectively consider the 2D $r\phi$-projection of the STT. The reason is that in the STT, the $z-$component is given as the center-of-tube parallel to the beam axis, as the position of the signal can be anywhere along the wire. To correctly find out the z-component, STT uses planes of skewed layers titled $\pm 3^\circ$ to the beam axis that requires an independent hit correction procedure for these layers. In general, PANDA's track reconstruction is performed in 2D and later extended to 3D by exploiting the skewed layers [10].

In addition, we will consider two different representations of the STT *i.e.* Euclidean (Chapter 7), or grid-like, geometry, and non-Euclidean (Chapter 8, Chapter 9), non-grid-like, which is a reference to non-Euclidean detector geometries. In the former case, standard deep learning models are better suited *e.g.* DNNs, RNNs, and CNNs that can process an entire event at a time. In the latter case, geometric deep learning models are suitable *e.g.* GNNs. A whole STT event is treated as a graph combining nodes and edges in both cases. Both kinds of models are used to perform edge classification on each graph. In the Euclidean case, a DNN is used for edge classification. Due to the variable size of a graph, a DNN can not process a full graph at a time, just like an RNN or CNN can do. In reality, edge classification is performed for each edge independently. Hence the notion of being *Euclidean* regarding DNNs is not strictly true. However, I will keep this notion for the sake of comparison. In non-Euclidean cases, GNNs can process a full graph regardless of size, a distinct feature of GNNs.

Apart from the detector geometry, the main challenge comes from the data we simulate. We are interested in the low-to-medium momentum, *i.e.* 50 MeV/c up to a few GeV/c, where the peculiar and non-perturbative features of strong interaction manifest. For example, hyperons often decay into pions with very small momentum (*i.e.* < 100 MeV) that are difficult to reconstruct since they may leave a curling pattern in the STT [18, 86].

In Section 4.4, I have summarized the challenges we face regarding track reconstruction in PANDA. To develop a prototype to address these challenges, I will simulate muons within the momentum range of 100 MeV to 1.5 GeV and reconstruct using the Euclidean (Chapter 7) and non-Euclidean (Chapter 8) representations. In Chapter 9, I will simulate Λ hyperons from the $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ reaction to investigate the capability of my non-Euclidean algorithm to reconstruct both tracks from displaced vertices and particles with low momenta.

6.2 Deep Learning Pipeline

In this section, I will introduce the concept of a *deep learning pipeline*, I have adapted the Exa.TrkX pipeline [155] for the PANDA experiment, especially the STT. The pipeline consists of four stages: *(i)* Data Generation, *(ii)* Edge Construction, *(iii)* Edge Classification, and *(iv)* Track Formation. These stages are shown in Figure 6.1.

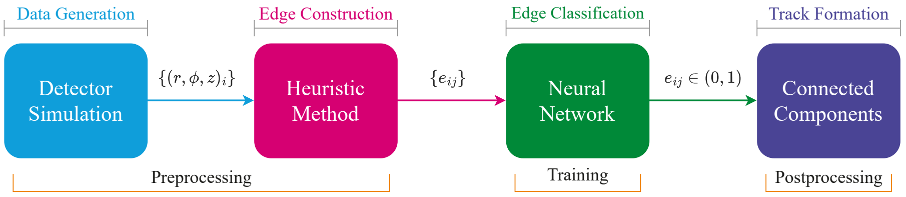


Figure 6.1. The Deep Learning Pipeline.

The pipeline runs sequentially where each stage takes input from the previous stage. The pipeline is deployed as a single instance, for example on a GPU cluster. To remain consistent in our handling of different representations of STT, I will from now on use terms from graph theory such as *nodes* for hit positions and *edges* for a link between two hits positions and use these terms for both graph based and non-graph based data. The stages of the pipeline are briefly discussed in the following sections.

6.2.1 Data Generation

The input data are produced in simulations using the PandaRoot[54]. The information is then stored into *Comma-Separated Values (CSV)* files that are later read inside a Python environment.

The signal left by a particle track inside a detector may have many distinct features, such as position, momentum, deposited charge, particle type and direction. These features can be used to define a *hit* in the detector. Further, the position of hits are transformed into cylindrical coordinates, (r, ϕ, z), new quantities are derived *e.g.* transverse momentum (p_t). Finally, the hit positions are filtered to add or remove the skewed layers in the STT. Each hit is assigned a feature vector and a corresponding particle number.

6.2.2 Edge Construction

An *edge* is a link between any two nodes. Edges are either created randomly or by some heuristic method. In our case, the heuristic method creates layer-wise edges, *i.e.* in adjacent layers, starting from the innermost layer to the outermost layer of the STT. In addition, edge construction is restricted to adjacent sectors of the STT. Each edge is labeled as either *true* or *false* depending upon whether an edge is part of a track or not. If the data contain noise, then all edges resulting from noise are marked as false. A complete *hit graph*, *i.e.* event as a hit graph, is created from the particle trajectories comprising of nodes and edges. The input features from both hits are concatenated together with a label which is then used for edge classification.

6.2.3 Edge Classification

In this stage, an *edge classification* is performed using either deep learning (Euclidean) or geometric deep learning (non-Euclidean) models. This stage has two different modes. In the first mode, a standard deep learning model, such as a DNN, is trained on the edges in a hit graph one by one. Due to the variable size of a hit graph, a DNN cannot process hit graphs as a whole hence it may lose the topological features of the data. In the second mode, a geometric deep learning model, such as a GNN, is used to classify edges in a hit graph. The GNN process the whole hit graph, *i.e.* event, as a whole, hence they are better at capturing the topological features of data.

After the training, model prediction is performed on the test dataset. As a result, a neural network assigns a probability, or score, to each edge. The edge scores are used to cluster nodes into tracks.

6.2.4 Track Formation

The track formation can be carried out using the *connected-component labeling* (CCL) algorithm from the graph theory [166, 167]. In computer vision, CCL is widely used to label connected-components in binary images. In High Energy Physics, it is often used to label tracks in the detectors [168]. To use this method, first edges are filtered out by applying a certain cut on the edge score. Then an $n \times n$ sparse or adjacency matrix in *coordinate* (COO) format, also known as the ijv or triplet format, is created, where n is equal to the number of nodes per event with matrix data taken as the edge score between nodes[1]. Finally, a depth-first-search (DFS) method scan through the adjacency matrix grouping nodes into components and assign a label to each component. In case of hit graphs, these components are known as track candidates.

Another method, equivalent to CCL, is to use density-based spatial clustering of applications with noise (DBSCAN) algorithm [91] on a predicted hit graph. The DBSCAN uses a maximum sample distance parameter $\varepsilon_{\mathrm{db}}$ that is the maximum distance between two samples to form a neighborhood *e.g.* we can chose a particular value of $\varepsilon_{\mathrm{db}}$ to mark whether two nodes are close enough to form an edge. For DBSCAN, an adjacency matrix similar to the above, is created. Here, the edge scores are converted into an inverse distance between two nodes. The DBSCAN then clusters nodes into different components based on the value of $\varepsilon_{\mathrm{db}}$ and assign a label to each cluster. These clusters are taken as track candidates.

Both CCL and DBSCAN methods are equivalent in operation as the DBSCAN method ultimately calls a connected-component algorithm such as DFS, which is similar to CCL. A higher value of edge score cut in the CCL method is equivalent to a lower value of $\varepsilon_{\mathrm{db}}$ in the DBSCAN. In this thesis, we will use DBSCAN method for track formation. Both of these method produce *connected-components* for directed graphs and *weekly-connected-components* for bidirectional graphs.

6.3 Strategy

In the following chapters, I will use the above pipeline for both Euclidean and non-Euclidean representations of the STT. In the first case, a DNN is used for edge classification for muons. In the second case, GNN will be used for edge classification for muons as well as the final state particles of the

[1] One can use either NetworkX [169] or PyG [170] libraries for this purpose.

$\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^{+}p\pi^{-}$ reaction. In this reaction, the tracks from the final state particles start in the Λ and $\bar{\Lambda}$ decay vertices, which are typically located several centimeters away from the beam-target interaction point. Hence, studying this reaction will provide a test on how well machine learning algorithms performs in the tracking of particles originating from displaced vertices. The pipeline for different cases will run on the Cori cluster of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory[2].

The work has been started and is ongoing to extend the STT to include other tracking detectors *e.g.* GEM and MVD detectors. This requires a heterogeneous deep learning setup. Not only the output data from these detectors is heterogeneous, but also the underlying detection technologies are significantly different. As a result, this is a challenging task that is not yet completed. However, the status of the work will be presented in Chapter 14.

[2] operated under Contract No. DE-AC02-05CH11231 using NERSC award ERCAP-0021226.

7. Application of Deep Learning in the Straw Tube Tracker (STT)

In this chapter, we will use a deep learning pipeline for the Euclidean representation of the STT. For simplicity, the skewed layers of STT have been removed and therefore, the tracks are reconstructed only in the $r\phi-$projection of the detector.

7.1 Deep Learning Pipeline

The deep learning pipeline designed for the Euclidean domain of our data follows the logic of Figure 6.1. In the following sections, each stage is discussed in more detail.

7.1.1 Data Generation

Five $\mu^+\mu^-$ pairs per event in the momentum range of 100 MeV/c to 1.5 GeV/c are generated using the particle gun within the geometric acceptance of STT *i.e.* $22^\circ < \theta < 140^\circ$. The choice of these relatively low momenta is motivated by the fact that the existing algorithms have difficulties reconstructing low p_t particle trajectories, as these trajectories are spiraling, strongly curved, and overlapping [18, 171].

Although muons may produce secondary particles when interacting with the detector material, these interactions have not been considered in the simulations. The muons originate at the beam-target, or $\bar{p}p$, interaction point (IP). Before reaching the STT, muons traverse the MVD detector. The GEM detector, on the other hand, falls outside the geometric acceptance of STT; hence no primary muon passes through it. In total, $100,000$ events are simulated using PandaRoot, resulting in about a million tracks.

The number of hits per event in the STT detector is shown in Figure 7.1. Tracks with relatively high transverse momentum (p_t) may traverse the full STT, leaving 26 hits per track. The peak around ~ 250 hits indicates that in a large fraction of events, all ten muons have large p_t. The region below this peak contains a small fraction of tracks that have not reached the STT or escaped the STT through the gap between the two halves of the STT detector. The events above the peak, on the other hand, are the ones that have extremely low p_t tracks ($p_t < 100$ MeV). These particles move in spiraling or curling trajectories, thus leaving more hits per event. On average, there are approximately 234 hits per event. It is also apparent from the hit distribution that the maximum number of hits in an event do not go beyond 400 hits.

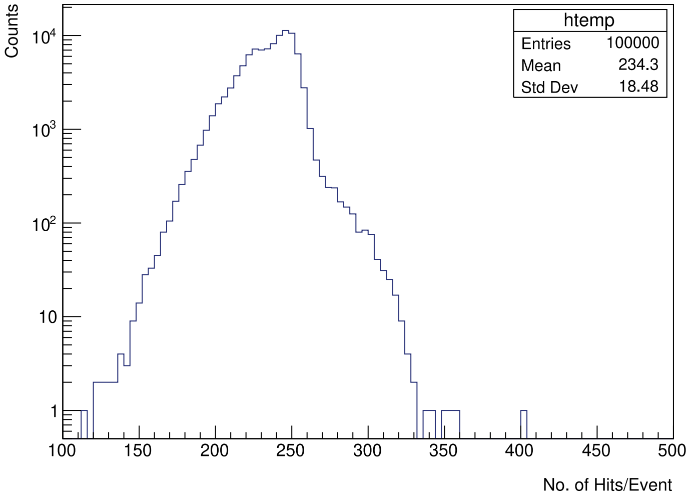


Figure 7.1. Histogram of the number of hits per event in STT. About 23 million hits are present in the dataset. The average number of hits per event are ~ 234.

7.1.2 Edge Construction

A heuristic method is used to build edges in the STT. The method creates edges in adjacent layers, from the innermost layer to the outermost layer. Moreover, edge construction is restricted to adjacent sectors of STT (see Figure 3.7). We will refer to this heuristic as the *layerwise edge construction*. The edges include both true and false edges, where true edges are those that belong to the same track, and false edges are those formed from hits from different tracks. The constraint on edge construction in the adjacent sectors is based on the assumption that it is improbable that a track traverses more than three sectors (half of the STT detector), even in the case of curly tracks. This observation is based on visually inspecting five hundred events with extremely low p_t tracks. In addition, this constraint also reduces class imbalance in the dataset to some extent, *i.e.*, between true and false edges. In our case, false edges are a factor of four more frequent than true ones. This is an unavoidable imbalance in our data. In classification tasks, the imbalanced data will force the neural network to spend more time on one class than the other, which may spoil the network's performance.

Figure 7.2 shows the total number of constructed edges as well as true and false edges. Out of the $7.8 \cdot 10^7$ total edges, the true and false edge fractions are calculated to be $\sim 20\%$ and $\sim 80\%$, respectively. To create a labeled dataset

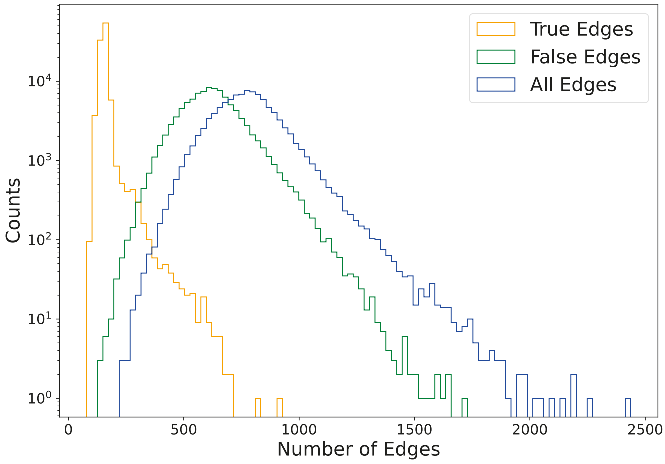


Figure 7.2. The histogram shows true (orange), false (green) and total (blue) edges constructed from the dataset.

for an edge classification task, the features of two nodes forming an edge are combined, and a label is assigned as '1' for true edges and '0' for false edges.

The node features are taken as r and ϕ positions in the polar coordinates. In addition, the isochrone radius (ρ) of a hit (see Section 3.4.1.4 regarding isochrones) is also added to the node features. As a result, the input to the neural network consists of six features from two hits (h_i, h_j) and a label.

i^{th} Node Features	j^{th} Node Features	Label
$h_i = [r_i,\ \phi_i, \rho_i]$	$h_j = [r_j,\ \phi_j, \rho_j]$	$y_{ij} = \begin{cases} 1 & \text{if True Edge} \\ 0 & \text{otherwise} \end{cases}$

The events are randomly split into three subsets for supervised learning with 90% events for training, 5% for validation, and 5% for testing. It is important to note that the split is made on the event level rather than the edge level to ensure that edges do not appear in multiple subsets.

7.1.3 Edge Classification

A six-layer DNN model with one input layer, five hidden layers, and one output layer is used for edge classification. By convention, the input layer is not counted. Hyperparameter tuning is performed using Ray Tune [172], where only the number of hidden nodes, hidden activation functions, and batch size is tuned. An example network is shown in Figure 7.3.

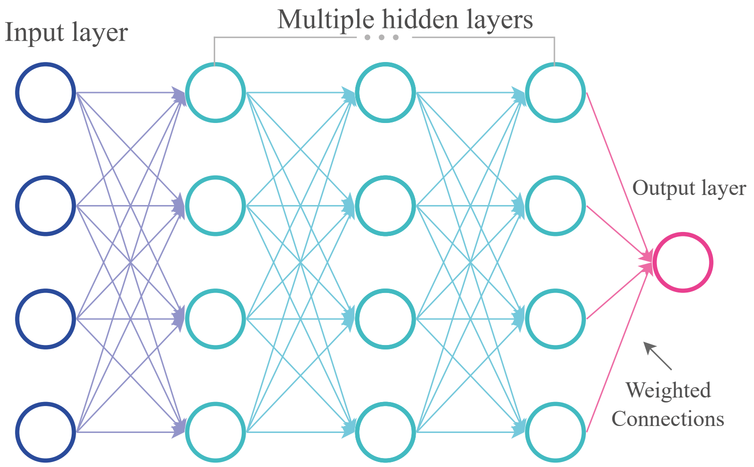


Figure 7.3. A dense neural network for edge classification.

After hyperparameter tuning, I found the following hyperparameters for our model (see Section 5.4 for explanation of these hyperparameters):

- Hidden layers: 5
- Layer architecture: [128, 128, 1024, 1024, 128, 1]
- Activation functions: [hidden: relu(), output: sigmoid()]
- Batch size: 128

In order to train the network, following additional hyperparameters are chosen:

- Binary Cross-Entropy (BCE) loss
- AdamW Optimizer:
  - $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$,
  - amsgrad=True, weight_decay=0.01
- Learning rate: $\alpha = 0.001$
- Activation Normalization: Layer Norm
- Data split: $[90\%, 5\%, 5\%]$

The network is trained for 50 epochs as the generalization gap between the training and the validation errors do not change significantly. However, one can train longer to investigate whether the network's performance changes drastically. Each training epoch represents a full training cycle, *i.e.* when the network passes through the training and validation dataset once. The network output for both true and false edges is shown in Figure 7.4.

I use Area Under the ROC (AUC) for model evaluation as defined in Section 5.4.6. Figure 7.5 shows the ROC curve for the current model. The AUC equals 0.9875 (left panel), indicating reasonably high model performance. The

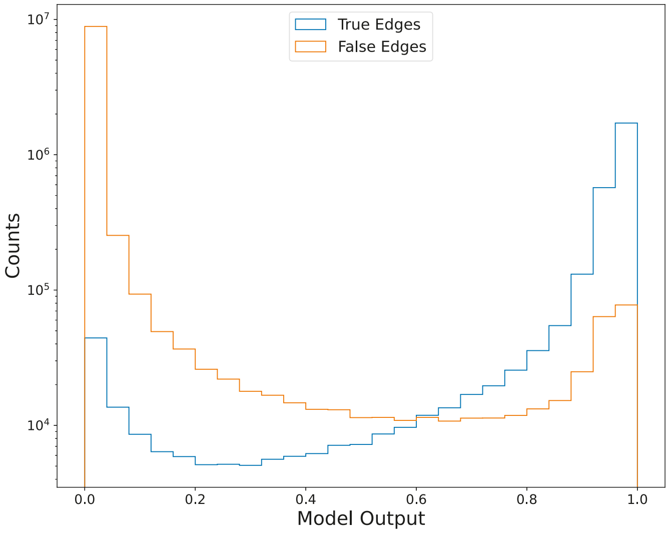


Figure 7.4. Model outputs on test dataset for true and fake edges.

efficiency and purity for various cuts on the classification score are shown in the right panel of the figure.

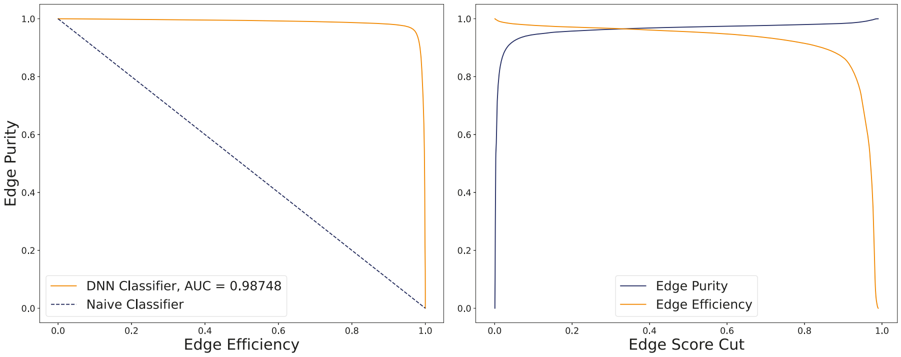


Figure 7.5. Model evaluation: (*left*) ROC with AUC ~ 0.9879, (*right*) edge efficiency (ε_E) and edge purity (p_E) as a function of edge score cut.

It is clear that by increasing the cut on the edge classification score, the edge purity increases. However, at the same time, the edge efficiency of the classifier gradually decreases. Hence, there is a trade-off between choosing a particular edge efficiency and purity value. In this scenario, one needs to choose an optimal value of the cut. For example, the edge score cut of $s = 0.5$ gives $\varepsilon_E = 0.96$ and $p_E = 0.97$.

The signal efficiency (ε_{sig}) is defined as the *true positive rate*, or *recall*, whereas the misidentification rate (ε_{bkg}) is defined as the *false positive rate* from the ROC. The background rejection rate is defined as one over misidentification rat ($1/\varepsilon_{\text{bkg}}$). The background rejection rate *vs* the signal efficiency for various edge scores is shown in Figure 7.6.

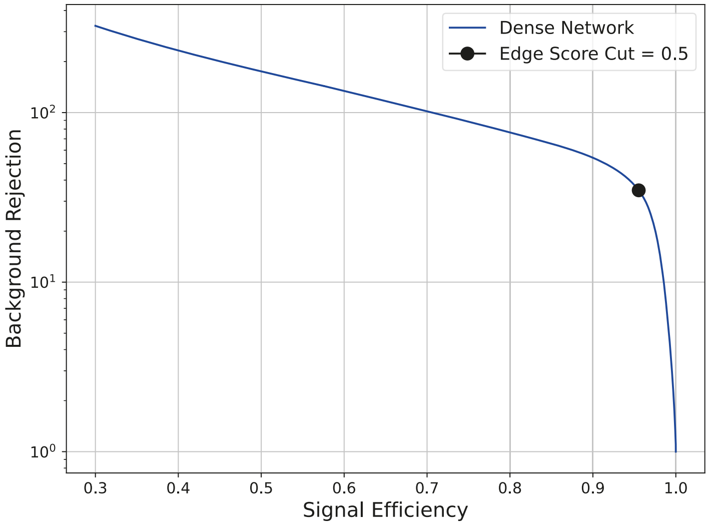


Figure 7.6. Signal to Background rejection rate of DNN for various edge score cuts.

In Figure 7.6, the black dot corresponds to the edge score cut of $s = 0.5$. Table 7.1 shows the signal efficiency, misidentification rate, and background rejection rate for edge score cuts of 0.5 and 0.7.

Table 7.1. *Signal efficiency, misidentification rate and background rejection rate for an edge score cut of 0.5 and 0.7.*

Edge score	Signal efficiency	Misidentification rate	Background rejection rate
0.5	0.955	0.029	34.8
0.7	0.935	0.023	43.4

Increasing the cut value will decrease signal efficiency and misidentification rate while increasing the background rejection rate.

7.1.4 Track Formation

For track formation, the DBSCAN method is used to find tracks. To choose an optimal value of DBSCAN distance parameter, ε_{db}, the tracking efficiencies, track purity and clone rate are calculated for different values of ε_{db}. As mentioned in Section 6.2.4, smaller values of ε_{db} corresponds to a higher cut on the edge score. Figure 7.7 shows tracking efficiencies, purity and clone rate as a function of ε_{db}.

Among the three values, marked with vertical lines, the value of ε_{db} is chosen to be 0.2 as an optimal value for the DBSCAN. $\varepsilon_{\text{db}} = 0.25$ is also a better choice but with slightly less tracking purity. In total, $2 \cdot 10^3$ events are

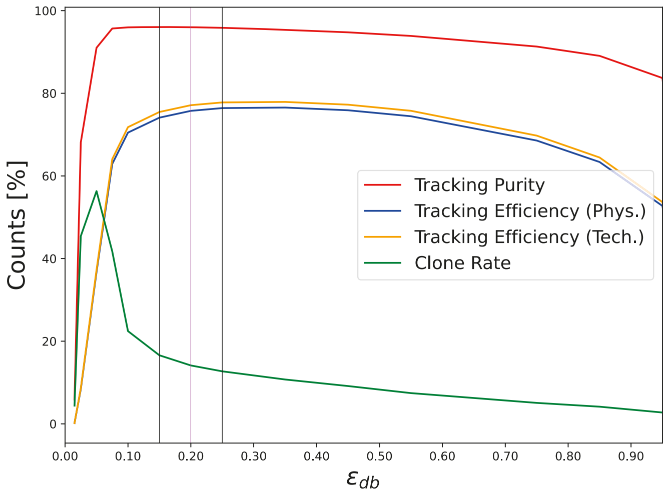


Figure 7.7. The tracking efficiencies, track purity and clone rate as function of ε_{db}.

used for track formation that took approximately 6 minutes and 31.4 seconds (real time) using single worker whereas it took 32.1 second (real time) using 8 worker to process all events on a Intel Core i7 CPU with 16 GB of RAM.

7.2 Track Evaluation

The track evaluation is performed according to the method described in Section 4.2. The evaluation criteria is taken as follows:

- Minimum True Hits (N_t): ≥ 7 STT hits for Reconstructable Particles
- Minimum Reco. Hits (N_r): ≥ 5 STT hits for Reconstructed Tracks
- Matching Fractions (MF): $> 50\%$

The tracking efficiencies, ghost rate and clone rate are given in Table 7.2:

Table 7.2. *Tracking efficiencies, ghost rate (GR), clone rate (CR) for $N_t \geq 7$, $N_r \geq 5$ and MF $> 50\%$.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	76.3 ± 0.272	77.2 ± 0.278	3.64 ± 0.329	17.2 ± 0.107

For a more fine-grained understanding of the performance, the efficiencies are calculated as a function of the transverse momentum (p_t), the lab polar angle of the track (θ), and the azimuthal angle of the track (ϕ). These variables are defined in Equation 7.1:

$$p_t = \sqrt{p_x^2 + p_y^2}, \quad \theta = \tan^{-1}(p_t, p_z), \quad \phi = \tan^{-1}(p_y, p_x) \qquad (7.1)$$

Figure 7.8 shows results as as function of p_t using the aforementioned evaluation criteria. On the left panel, the number of selected, reconstructable, and matched particles are shown; on the right panel, tracking efficiencies are shown for various p_t. A large fraction of particles is generated with $p_t < 0.5$ GeV/c. This is the region where one expects crossing and spiraling tracks due to low momentum. Hence the tracking efficiencies are too small in this region. Recall that, to fully traverse the STT, a particle needs a minimum $p_t > 0.25$ GeV/c (see Eq. 3.5). However, for particles with $p_t > 0.5$ GeV/c, the tracking efficiencies are large and almost flat for the rest of the p_t region.

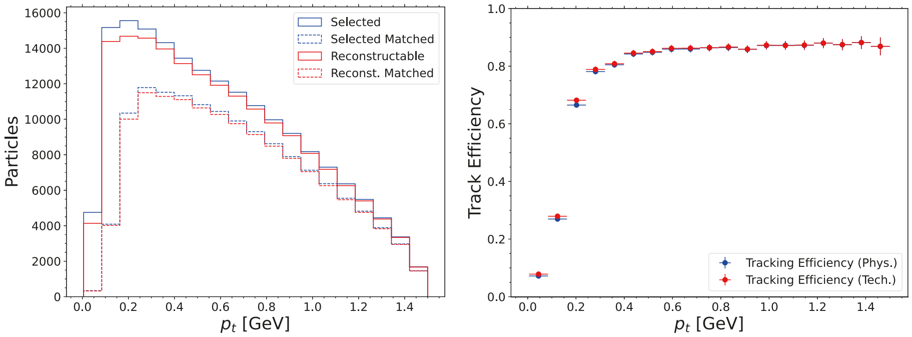


Figure 7.8. Number of selected, reconstructable, and matched particles (*left*), and tracking efficiencies (*right*) as function of p_t.

The tracking efficiencies as a function of θ are shown in Figure 7.9. The left panel shows the number of selected, reconstructable, and matched particles. On the right, tracking efficiencies are plotted as function of θ within the STT acceptance. There is no significant angular dependence on tracking efficiencies over the whole range. The slight but smooth reduction in efficiencies at the edges of STT is due to particles generated close to the boundaries of STT acceptance. As a result, a fraction of particles leaves the STT in the longitudinal direction leaving fewer hits in the transversal direction.

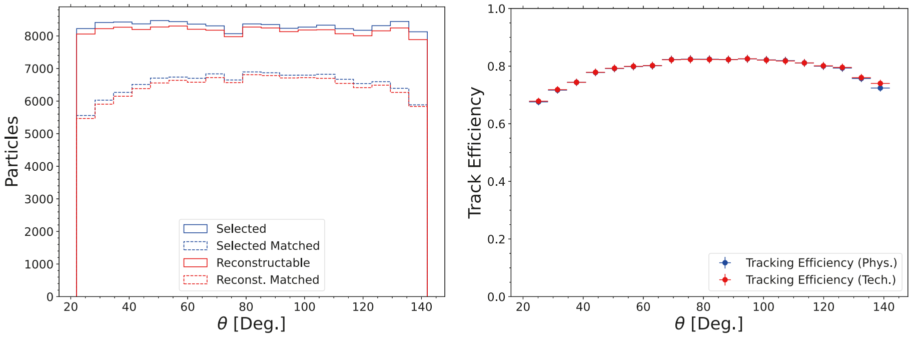


Figure 7.9. Number of selected, reconstructable, and matched particles (*left*), and tracking efficiencies (*right*) as function of θ.

I also investigated dependence of the tracking efficiencies on the azimuthal angle, ϕ, corresponding to different sectors of the STT. The number of selected, reconstructable, and matched particles (left) as well as the tracking efficiencies (right) are shown in Figure 7.10. The efficiencies are almost consistent throughout the whole ϕ range spanning from $-180°$ to $180°$ with small dips at $\pm 90°$. These correspond to the vertical gap between the two halves of STT, where cluster-jet target and target-beam dump are located. It is reassuring that our classifier is not biased towards a certain $\phi-$region and can reconstruct particles with equal accuracy in the whole detector.

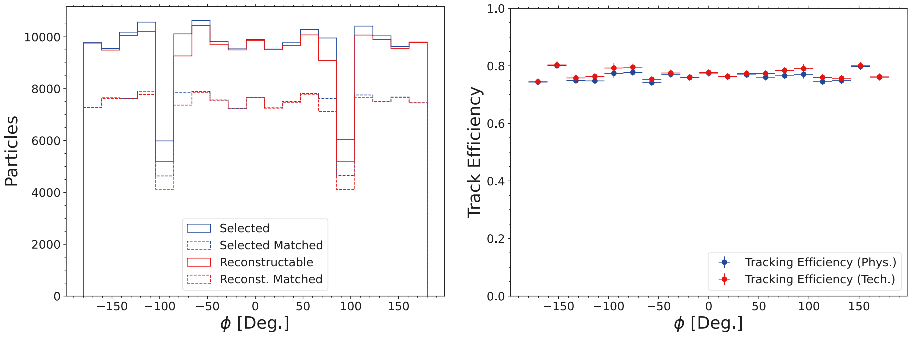
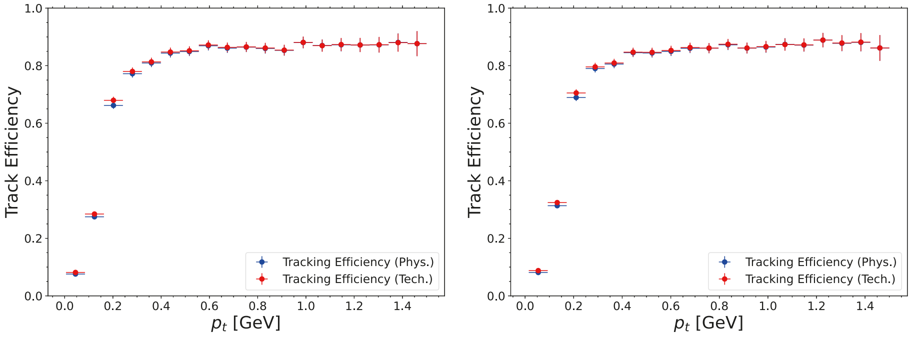


Figure 7.10. Number of generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of ϕ.

In order to see whether our classifier is biased towards a certain charge, the tracking efficiencies as function of p_t are calculated for μ^+ and μ^- separately. The tracking efficiencies as function of p_t for both cases are shown in Figure 7.11.



(a) Positive Charge

(b) Negatitive Charge

Figure 7.11. Tracking efficiencies for positive (*right*) and negative (*left*) muons as function of p_t.

It appears that our classifier is not biased towards a particular charge. To show this effect, the difference of overall efficiency ($\varepsilon_{phys.}$) from both positive and negative particles is plotted as function of p_t as shown in Figure 7.12. It is clear that the difference of efficiencies for both positive and negative charge is close to zero within the error bars, hence it is safe to conclude that there no bias towards a particular charge.

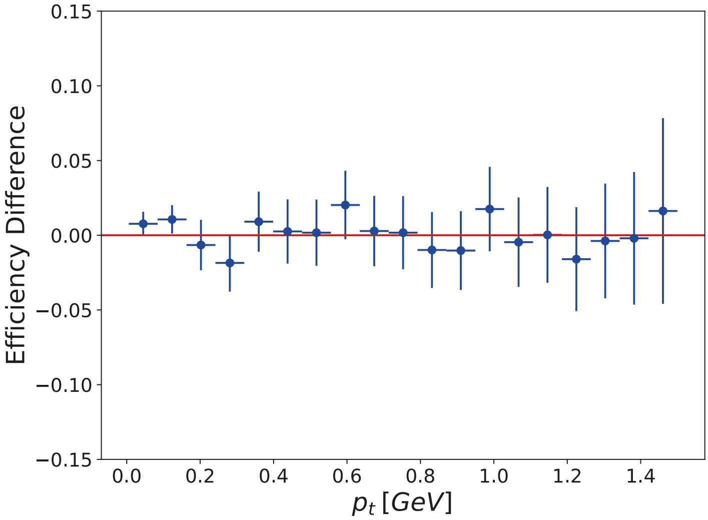


Figure 7.12. The difference of overall efficiencies of positive and negative charge as function of p_t.

7.3 Summary of Results

The tracking efficiencies, ghost rates, and clone rates are calculated for various MF greater than 50%, 75% and 95%, and two different values of N_r taken as 5 and 6 for reconstructed tracks. The results are summarized in Table 7.3.

Table 7.3. *Tracking efficiencies, ghost rate (GR), clone rate (CR) for loose, moderate and strict matching criteria with at least 5 as well as 6 hits for reconstructed tracks.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	76.3 ± 0.272	77.2 ± 0.278	3.64 ± 0.329	17.2 ± 0.107
7	5	75	58.2 ± 0.225	58.6 ± 0.230	12.0 ± 0.307	27.4 ± 0.141
7	5	95	53.5 ± 0.213	53.8 ± 0.216	14.8 ± 0.300	29.7 ± 0.148
7	6	> 50	75.5 ± 0.270	76.8 ± 0.278	3.78 ± 0.337	13.9 ± 0.098
7	6	75	57.7 ± 0.224	58.6 ± 0.230	12.6 ± 0.314	24.5 ± 0.135
7	6	95	53.0 ± 0.211	53.8 ± 0.216	15.2 ± 0.307	27.1 ± 0.144

In the case of $N_r = 5$, increasing the matching criteria from 50% to 75% the $\varepsilon_{phys.}$ result in a $\sim 24\%$ drop in its values whereas the ghost rate is increased by a factor of 3.3. The clone rate increases by a factor of 1.6. For $N_r = 6$, these variables drop within the same scale as for $N_r = 5$ *i.e.* $\varepsilon_{phys.}$ drops by 24%, the ghost rate increases by a factor of 3.3 and the clone rate increases by a factor of 1.8.

By using a standard deep learning model such as DNN for an edge classification task, we achieve tracking efficiencies of up to 80% specific to the network model and training parameters I have chosen. The low efficiencies can be attributed to various factors. A large class imbalance in the data is one major cause, as the network might need to learn better to distinguish between two classes. The class imbalance is not handled for the sake of comparison with Graph Neural Networks. A second major factor might be the low p_t tracks that may curl to intersect other trajectories. This factor is a major drawback in the DBSCAN and CCL methods used to form tracks. These methods are designed to find connected components in a predicted graph that breaks intersecting trajectories into several sub-tracks resulting in a large clone rate.

To mitigate large clone rates, one might need a different algorithm to account for shared hits from intersecting tracks. For example, a *pathfinding* or a *walkthrough* algorithm can be used that can account for shared hits. In addition, the walkthrough algorithm can also be integrated with DBSCAN or CCL. The idea is to run the walkthrough algorithm on each component produced by either DBSCAN or CCL algorithms. If successful, one may recover a small fraction of tracks treated as clones. Development and integration of this method have yet to reach the production level. Hence it has not been tested.

8. Application of Geometric Deep Learning in the Straw Tube Tracker (STT)

In the previous chapter, track reconstruction is performed for the Euclidean representation of the STT. Many real-world systems produce data that falls into the non-Euclidean domain, such as sets, graphs, and manifolds. The STT detector is a non-Euclidean geometry, where straw tubes are arranged into six sector in the form of a hexagon. This chapter explores geometric deep learning models for track reconstruction in the STT [164].

8.1 Geometric Deep Learning Pipeline

The *Geometric Deep Learning (GDL)* pipeline is slightly different from the deep learning pipeline as shown in Figure 6.1. The *data generation* and *track formation* stages are similar to the stages in Chapter 7. However, the stages of the *edge construction* and *edge classification* are somewhat different. In GDL, the events are treated as bidirectional graphs, and specialized graph neural networks are used for edge classification. The stages of the pipeline are discussed in the following sections.

8.1.1 Data Generation

The dataset used in geometrical deep learning is the same as in Section 7.1.1 *i.e.* five $\mu^+\mu^-$ pairs per event are generated in the momentum range of 100 MeV/c to 1.5 GeV/c. In total, 10^5 events are generated. The track reconstruction is performed in the $r\phi-$projection of STT, and skewed layers have been removed from the data. Choosing the same conditions allows for comparing both methodologies used for track reconstruction.

8.1.2 Edge Construction

The raw input data, represented as a point cloud, is converted into *bidirectional* graphs. The edges are directed from one node to another in forward and backward directions. The edges of the same particle are labeled as true edges, while those formed by nodes from different particles are marked as false edges. For this purpose, I used a heuristic method similar to the one used in Section 7.1.2 that captures all true edges while keeping the number of fake edges low. In this method, edges are formed in successive steps starting from the innermost layer toward the outermost layer in the radial direction. The edge formation is limited to adjacent or neighboring layers. In addition, edge construction is contained to adjacent sectors. If a particle produces two hits on the same layer, then both hits are included in the graph. If a hit is missing in a layer, then the graph is constructed using the next adjacent layer.

Figure 8.1 shows a constructed graph of an event. To the left in the figure, the true graph, *i.e.* the ground truth of an event, is shown. To the right, one can see an input graph with all possible combinations of edges. Here, only the first three layers are visualized.

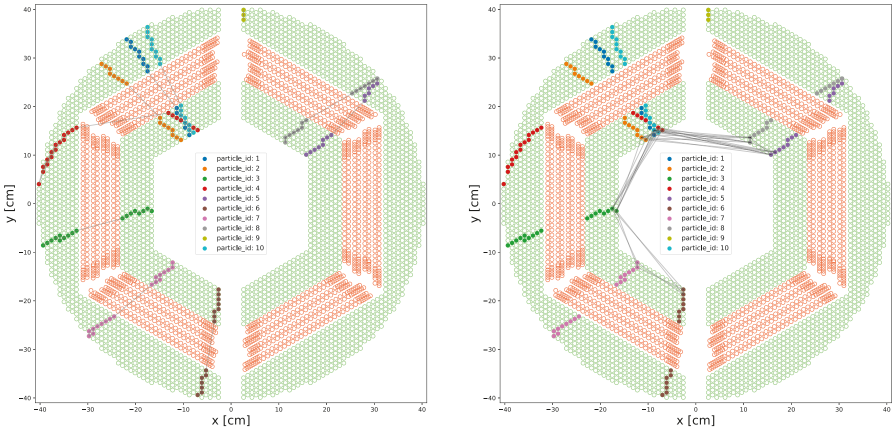


Figure 8.1. Graph representation of an event: *(left)* True Graph, *(right)* Input Graph.

The graphs are represented in the form of an adjacency list, $I \in \mathbb{N}^{2 \times n_{edges}}$, in coordinate (COO) format where n_{edges} is the total number of edges per event. The node features are stacked in a matrix $X = [x_i] \in \mathbb{R}^{n_{nodes} \times 3}$ and their labels in column vector $y \in \mathbb{R}^{n_{edges}}$. Our hit graph now looks like $\mathscr{G}_{COO} := (X, I)$. This format is chosen to efficiently process graphs that have less time complexity and take less computer memory compared to the adjacency matrix format.

8.1.3 Edge Classification

Three different kinds of graph neural networks under message-passing framework [114] are tested for edge classification: *(i)* Graph Convolutional Network (GCN) [113], *(ii)* Graph Attention Network (GAT) [115], *(iii)* Interaction Graph Neural Network (IGNN) [116]. These models further consist of residual connections [173] to speedup training. The network architecture, *i.e.* nodes, and layers of these networks are the same as used in Ref. [165] that are tuned for a track reconstruction task on the TrackML dataset [152]. In this thesis, these models are only tuned for various message-passing aggregation functions such as `sum()`, `mean()`, `max()`, `sum_max()`, etc.

The networks are trained for 20 epochs using 10000 events with data split of 80%, 10%, and 10% of total events for training, validation, and testing. By inspecting the validation loss, it is found that both GAT and IGNN are very close in performance compared to the GCN. IGNN is chosen to perform edge classification. Figure 8.2 shows validation loss for all three networks.

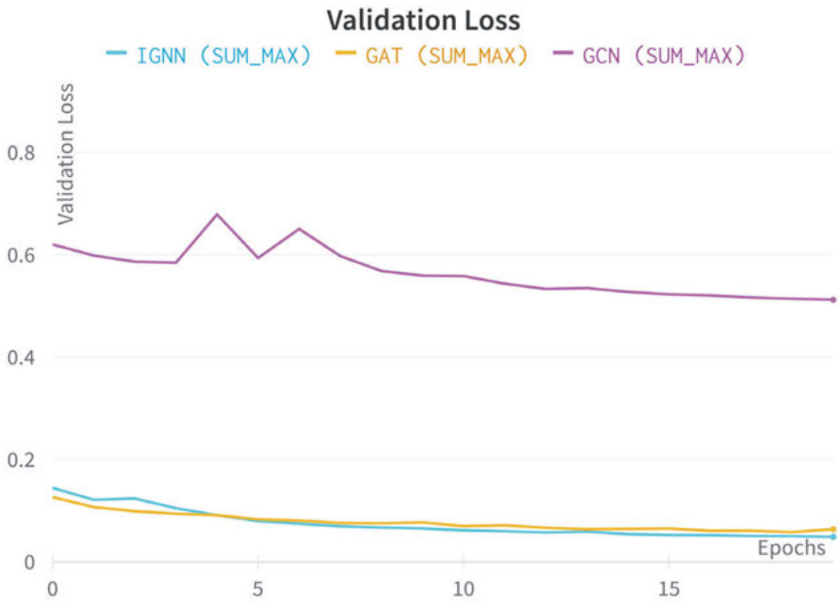


Figure 8.2. Validation loss for IGNN, GAT and GCN with SUM_MAX aggregation.

The IGNN consists of three modules: *(i)* encoder module, *(ii)* graph module, and *(iii)* output or decoder module. The encoder module consists of an edge network and a node network, and its task is to encode input node features to a vector of hidden features and to create edge features from neighboring nodes. In the graph module, aggregated neighboring edge features are passed to the node network, and the neighboring node features are passed to the edge network. This is a message-passing step where information is exchanged between nodes and edges. This step is repeated eight times. The final output is then passed to the output module that performs binary classification using the binary-cross-entropy loss function. As a result, each edge is assigned an edge score. A schematic diagram of IGNN used in this work is shown in Figure 8.3.

For network training on full dataset, *i.e.* 10^5 events, the following hyperparameters are used (see Section 5.4 for an explanation of hyperparameters):

Hyperparameters for GNNs

- Message-passing: sum_max()
- Binary Cross-Entropy (BCE) loss

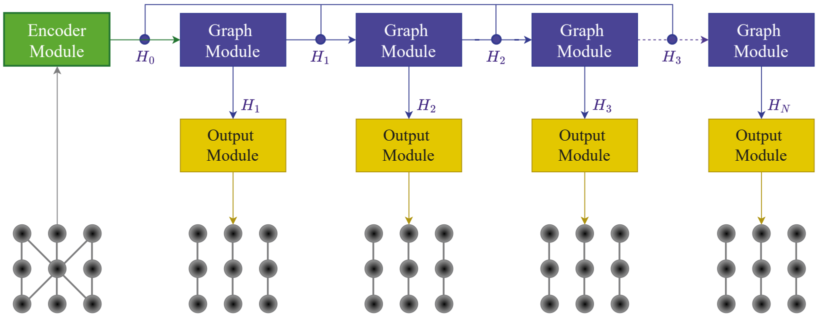


Figure 8.3. Schematic of Interaction Graph Neural Network (Interaction GNN).

- AdamW Optimizer:
  - $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$,
  - amsgrad=True, weight_decay=0.01
- Learning rate: $\alpha = 0.001$
- Activation Normalization: Layer Norm
- Batch size: 1
- Data split: $[90\%, 5\%, 5\%]$

Figure 8.4 shows the output of the Interaction Network on the test dataset:

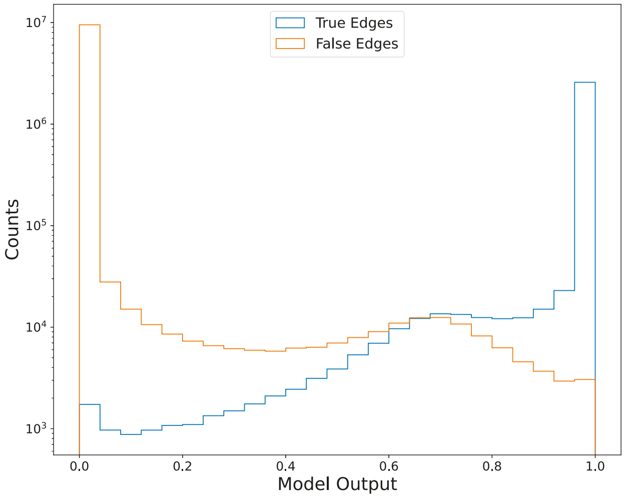


Figure 8.4. Model outputs on test dataset for true and fake edges.

The probability of false edges peaked around 0, on the other hand, the probability of predicted true edges peaked around 1. The fraction of false edges is much higher than the true edges, which is due to the imbalance in our dataset.

The performance of the network, *i.e.* its ability to distinguish true and false edges, is evaluated using the AUC metric. The ROC curve is drawn using edge efficiency and edge purity (see Section 5.4.6). The edge efficiency (ε_E) is a fraction of selected true edges, and edge purity (p_E) is the fraction of true edges among the selected ones. From the ROC curve, we obtain the AUC value to be 0.9997. The higher AUC values show higher network performance. The Figure 8.5(a) shows the ROC curve built from the edge efficiency and edge purity, and Figure 8.5(b) shows edge efficiency and purity for various values of edge score cuts.

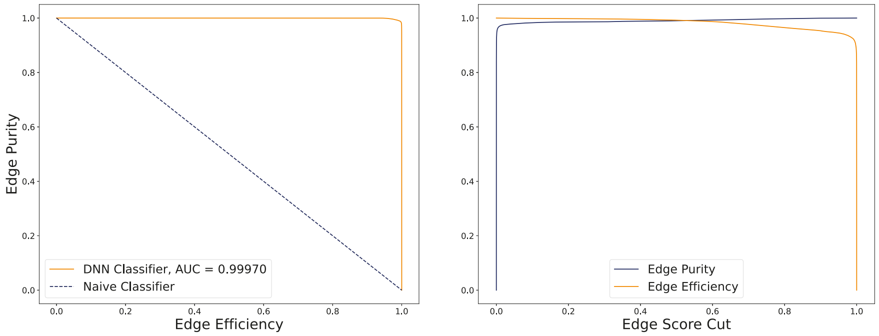


Figure 8.5. Model evaluation: ε_E and p_E as a function of edge score cut *(left)*, ROC with AUC $= 0.9997$ *(right)*.

The edge score cut of $s = 0.5$ gives $\varepsilon_E = 0.992$ and $p_E = 0.990$. The signal to background rejection as function of edge score cuts is shows in Figure 8.6.

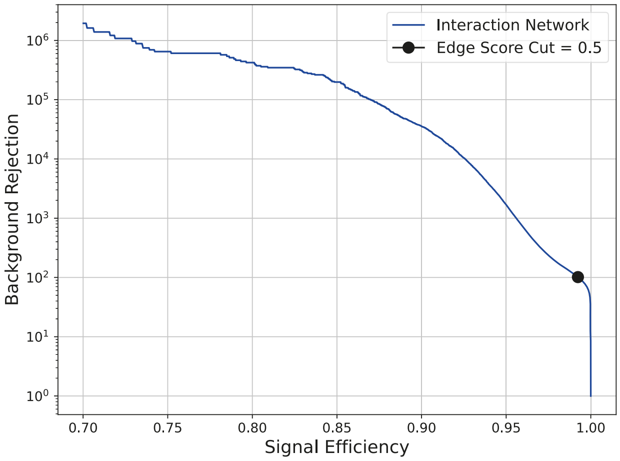


Figure 8.6. Signal to Background rejection of IGNN for various edge score cuts.

The background rejection rate is defined as one over misidentification rate (ε_{bkg}), where ε_{bkg} is the fraction of fake edges that pass the edge score cut, whereas the signal efficiency (ε_{sig}) is defined as the number of true edges passing the edge score cut, divided by the total number of true edges. Table 8.1

shows the signal efficiency, misidentification rate, and background rejection rate for score cuts of 0.5 and 0.7.

Table 8.1. *Signal efficiency, misidentification rate and background rejection rate for an edge score cut of 0.5 and 0.7.*

Edge score	Signal efficiency	Misidentification rate	Background rejection rate
0.5	0.992	0.0099	101.4
0.7	0.977	0.0049	213.5

Increasing the edge score cut decreases the signal efficiency but results into a high background rejection rate.

8.1.4 Track Formation

The DBSCAN method is used to form tracks from the edge scores. To choose an optimal value of $\varepsilon_{\mathrm{db}}$ for DBSCAN (see Section 7.1.4 for details), I scanned tracking efficiencies, tracking purity and clone rate as function of $\varepsilon_{\mathrm{db}}$ as shown in Figure 8.7.

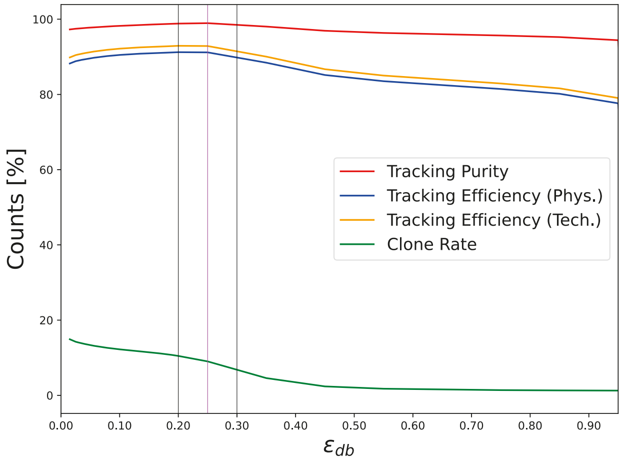


Figure 8.7. The efficiency and purity curves for various $\varepsilon_{\mathrm{db}}$ values.

Three values of $\varepsilon_{\mathrm{db}}$ are marked with vertical lines at 0.20 (black), 0.25 (magenta), and 0.30 (black). The smaller $\varepsilon_{\mathrm{db}}$ values indicate a high cut on the edge score. We chose the middle value for $\varepsilon_{\mathrm{db}} = 0.25$ based on high tracking efficiencies, track purity, and relatively low clone rates. In the case of bidirectional graphs, the DBSCAN method gives the *weakly-connected-components* of the graphs that are taken as track candidates. For $2 \cdot 10^3$ events, track formation took approximately 6 minutes and 31.4 seconds (real-time) with a single worker, whereas it took 32.1 seconds (real-time) with eight workers to process all events on an Intel Core i7 CPU with 16 GB of RAM.

8.2 Track Evaluation

The track evaluation is performed according to the method described in Section 4.2. The evaluation criteria are taken as mentioned in Section 7.2 for the Euclidean case. For a matching fraction > 50%, $N_t \geq 7$ STT hits for reconstructable particles and $N_r \geq 5$ STT hits for reconstructed tracks, the results are shown in Table 8.2:

Table 8.2. *Tracking efficiencies, ghost rate (GR), clone rate (CR) for $N_t \geq 7$, $N_r \geq 5$ and MF $> 50\%$.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	91.0 ± 0.309	92.6 ± 0.318	1.25 ± 0.322	11.5 ± 0.082

We observe an almost 20% rise in the technical and physical tracking efficiencies compared to the Euclidean case. Furthermore, we see that the ghost rate decreases by 66% and the clone rate by 33%, see Table 7.2. Hence, we observe a significant improvement in the model's performance trained on complete graphs for the same evaluation criteria. Furthermore, a track evaluation is performed to investigate the dependence of tracking efficiencies on the transverse momentum (p_t), lab polar angle (θ), and the azimuthal angle (ϕ), see Equation 7.1 for these variables.

The tracking efficiencies as a function of p_t are shown in Figure 8.8. The number of selected, reconstructable, and matched particles for various values of p_t are shown on the left panel. On the right, corresponding tracking efficiencies are shown as a function of p_t.

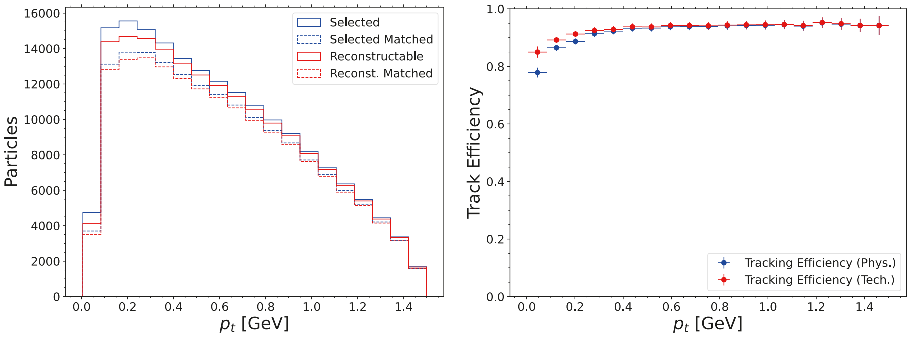


Figure 8.8. Number of selected, selected and matched, reconstructable, and reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) *vs* p_t.

The tracking efficiencies have significantly improved for various p_t values compared to the efficiencies shown in Figure 7.8 for the Euclidean case. Most importantly, the gain in efficiencies at $p_t < 0.2$ GeV/c is nearly as high as 80%, which we recall was the upper limit in the Euclidean case. This is the region where one expects spiraling and intersecting trajectories. Recall that, to fully traverse STT, a particle needs a minimum $p_t > 0.25$ GeV/c (see Eq.

3.5). For high p_t values, the tracking efficiencies gradually rise as the particle trajectories are relatively straight; hence the efficiencies are relatively high.

It is also interesting to look into how the tracking efficiencies depend on the different values of θ as shown in Figure 8.9. Compared to the results shown in Figure 7.9, not only the tracking efficiencies are higher for whole $\theta-$range, but also there is a significant gain in efficiencies near the boundaries of the STT acceptance.

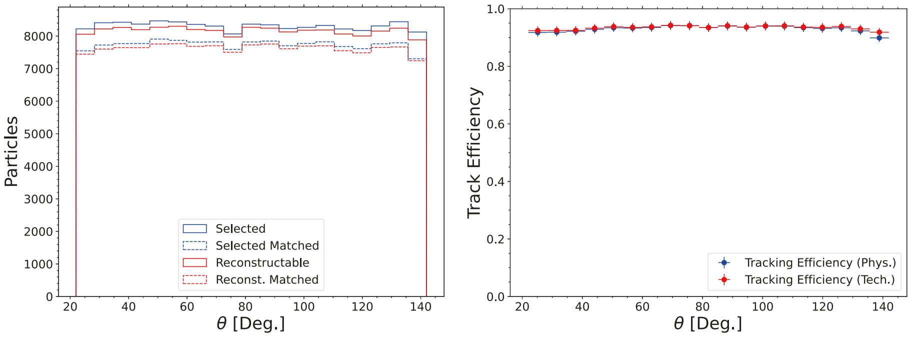


Figure 8.9. Number of generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of θ.

The number of selected, reconstructable, and matched particles as a function of θ is shown on the left panel. One can see that the number of reconstructed particles is homogeneously distributed over the whole θ range that corresponds to the STT acceptance. Small dips at the edges are the boundaries of STT acceptance, hence a small drop in tracking efficiencies as shown on the right panel of Figure 8.9.

The dependence of the efficiency on the azimuthal angle ϕ is shown in Figure 8.10.

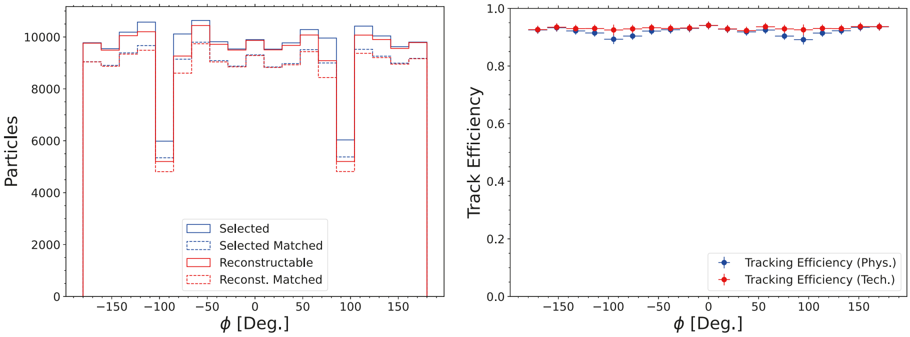
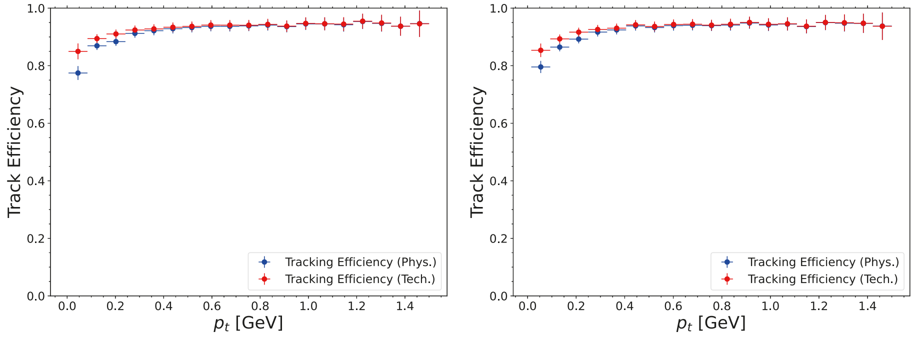


Figure 8.10. Number of generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of ϕ.

Apart from two dips in curves at ±90°, the particles are reconstructed equally well at different regions of the STT. The dips correspond to a vertical gap reserved for the cluster-jet target and target-beam dump; hence a fraction of particles are lost. The loss in efficiency, therefore, should not be attributed to the tracking algorithm but to the hit acceptance. This is also illustrated by the difference in the technical and the physical efficiency in the right panel of Figure 8.10.

Lastly, I looked into tracking efficiencies for μ^+ and μ^- separately as a function of p_t, shown in Figure 8.11.



(a) Positive Charge

(b) Negative Charge

Figure 8.11. Tracking efficiencies for positive (*right*) and negative (*left*) muons as function of p_t.

The efficiency curves for both categories of muons are similar; hence one can conclude that the efficiencies are independent of the charge of reconstructed particles. One can see more clearly in Figure 8.12 where the difference in overall tracking efficiencies, $\varepsilon_{phys.}$, for positive and negative charges is shown.

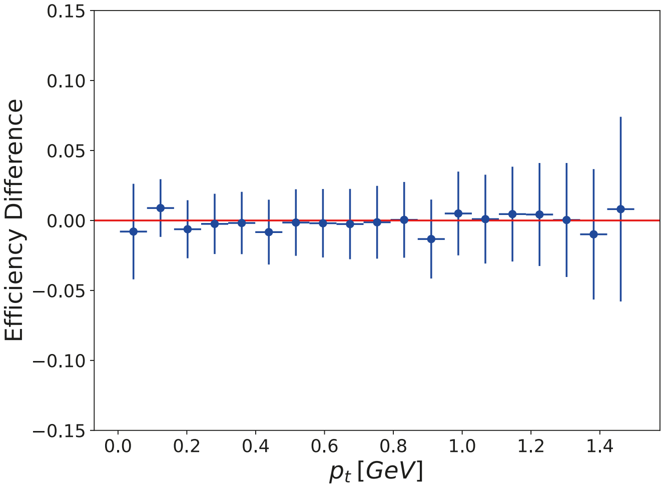


Figure 8.12. Efficiency difference for positive and negative charge as a function of p_t.

8.3 Summary of Results

Similar to the results shown in Table 7.3, the tracking efficiencies, ghost rates, and clone rates are calculated for various MF greater than 50% (loose), 75% (moderate) and 95% (strict), and two different values of N_r that are taken as 5 and 6 for reconstructed tracks. The values of N_t is kept as ≥ 7 STT hits for reconstructable particles. The results are summarized in Table 8.3.

Table 8.3. *Tracking efficiency, ghost rate (GR), clone rate (CR) for loose, moderate and strict matching criteria with at least 5 and 6 hits for* ***reconstructed*** *tracks.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	92.0 ± 0.312	93.0 ± 0.319	1.34 ± 0.315	14.1 ± 0.090
7	5	75	81.7 ± 0.286	82.4 ± 0.292	3.56 ± 0.310	21.3 ± 0.115
7	5	95	74.8 ± 0.268	75.4 ± 0.274	5.78 ± 0.304	25.5 ± 0.127
7	6	> 50	91.0 ± 0.309	92.6 ± 0.318	1.25 ± 0.322	11.5 ± 0.082
7	6	75	81.0 ± 0.284	82.4 ± 0.292	3.23 ± 0.317	19.1 ± 0.110
7	6	95	74.1 ± 0.267	75.4 ± 0.274	5.28 ± 0.312	23.6 ± 0.124

There is a significant improvement in all track evaluation parameters compared to Euclidean case, see Section 7.3 for a comparison. The gradual drop in efficiencies for different matching criteria is rather small. The rise in ghost rate is much smaller and even for the strict criteria it is slightly above 5%, which is a significant improvement. However, the clone rate is still higher than 10% for any case shown in Table 8.3 which is understood as the inefficiency of the DBSCAN method at low p_t that gives rise to intersecting tracks.

In Figure 8.13, an example of a true (left) and reconstructed (right) event is shown. The tracks from crossing or overlapping tracks are broken by the DBSCAN method.

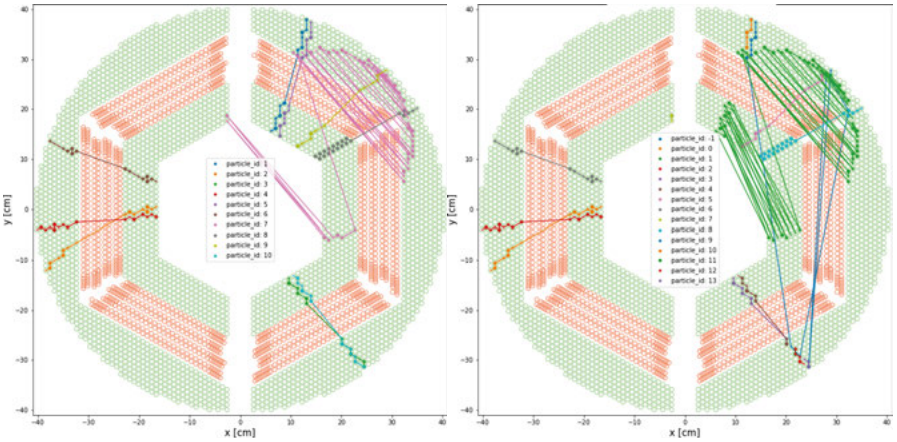


Figure 8.13. An example of a true and a reconstructed event.

The drop in efficiency and the high clone rate is the result of crossing or overlapping tracks in the events. One can verify this assumption by choosing a clean sample where none of the tracks cross or overlap. For events that contain well-separated tracks, the overall tracking efficiency rises to 97.6% with a ghost rate of 0.220% and a clone rate of 7.92%. The clone rate is reduced to almost half of the original value in this case. The remaining loss in efficiency is caused by occasional tracks with extremely low pt. Here, our graph construction heuristic failed to properly build ground truth of a graph, see Figure 8.13 (right). In such cases, the edge classification shows a strange behavior: some true edges are marked as false and some of false ones are marked as true. Hence, a better heuristic method designed for curling particles will increase tracking efficiencies. To tackle high clone rate, one needs a track-building method that can handle shared hits resulting from intersecting particles, see a discussion on a *walkthrough* algorithm and its integration with the DBSCAN or CCL in Section 7.3.

9. Hyperon Reconstruction in the STT

The next step is to investigate how the GNN algorithm performs with strongly interacting hadrons, such as pions and protons. Furthermore, we want to know whether our algorithm can reconstruct tracks that originate a measurable distance from the interaction point. Therefore, in this chapter I have investigated the $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ reaction. The final states are produced at a secondary decay vertex due to the weak decay of the Λ and $\bar{\Lambda}$. Hence, this is a suitable benchmark for track reconstruction at the PANDA experiment. In the following, this reaction is discussed briefly, before reconstructing the final state particles using geometric deep learning.

9.1 The $\bar{p}p \to \bar{\Lambda}\Lambda$ Reaction

The Λ hyperons have a long lifetime ($\sim 10^{-10}$ s), thus they fly a measurable distance in the detector before decaying. The Λ hyperons themselves are neutral and therefore, they leave no signals in the detector. The reaction is shown in Figure 9.1.

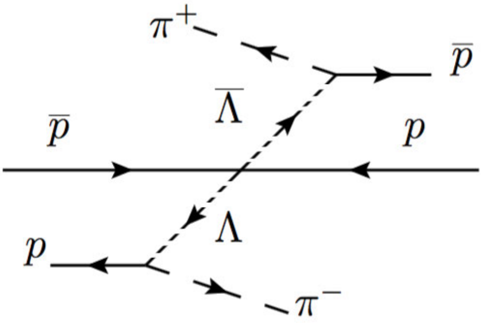


Figure 9.1. The topology of $\bar{p}p \to \bar{\Lambda}\Lambda$ reaction in the Center-of-Mass (CM) frame of the reaction. Image is credited to Ref. [10].

I have chosen to study this reaction at a beam momentum of 1.64 GeV/c, where both previous measurements by the PS185 experiment and other PANDA simulation studies, have been performed. The production model is based on a parametrization of data from PS185 at the Low Energy Antiproton Ring (LEAR). They found the production cross section at $p_{beam} = 1.64$ GeV/c to be $\sigma(p\bar{p} \to \Lambda\bar{\Lambda}) = 64 \pm 0.4 \pm 1.6 \mu$b [14]. For beam momenta $p_{\bar{p}} < 1.667$ GeV/c, it was found that the differential cross-section can be parameterized by following Legendre polynomial [175].

$$P(x) = a_0 + a_1 x + a_2(0.5(3x^2 - 1)) + a_3(0.5(5x^3 - 3x)) \tag{9.1}$$

where $x = cos\theta$ and θ is the production angle of $\bar{\Lambda}$ in the Center-of-Mass (CM) frame of the reaction. The coefficients, a, are given as:

$$a_0 = 1 \tag{9.2}$$

$$a_1 = -5.053 + 5.701 p_{\bar{p}} - 1.131 p_{\bar{p}}^2 \tag{9.3}$$

$$a_2 = -7.078 + 7.484 p_{\bar{p}} - 1.490 p_{\bar{p}}^2 \tag{9.4}$$

$$a_3 = -7.439 + 7.298 p_{\bar{p}} - 1.399 p_{\bar{p}}^2 \tag{9.5}$$

This means that the scattering angular distribution of the produced antihyperon is strongly peaked in the forward direction of the $\bar{p}p$ CM system, as shown in Figure 9.2. As a result, the Λ peaks in the backward direction in the CM system which means that in the lab system, it is close to at rest.

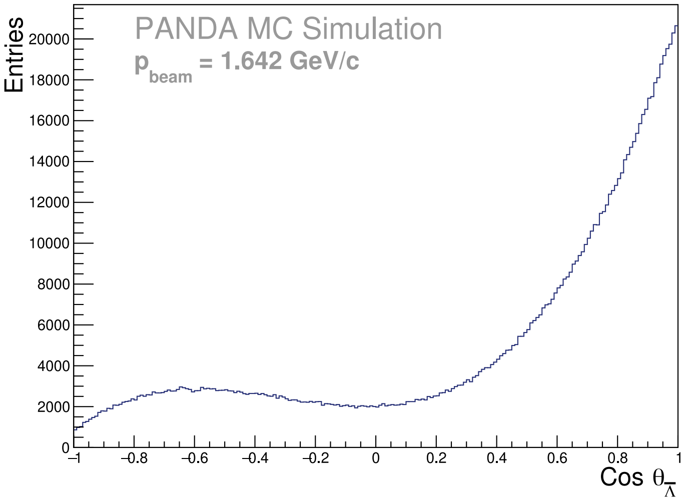


Figure 9.2. The angular distribution of $cos(\theta)$ in the Center-of-Mass (CM) frame of the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$ reaction.

9.1.1 Reaction Kinematics

Let's first look into the reaction in terms of the kinematics of the final state particles, to learn what kind of detector signals we expect. In total, 10^6 events are simulated for visualizing the reaction kinematics. The longitudinal (p_l) versus transverse (p_t) momentum distribution of all decay particles are shown in Figure 9.3.

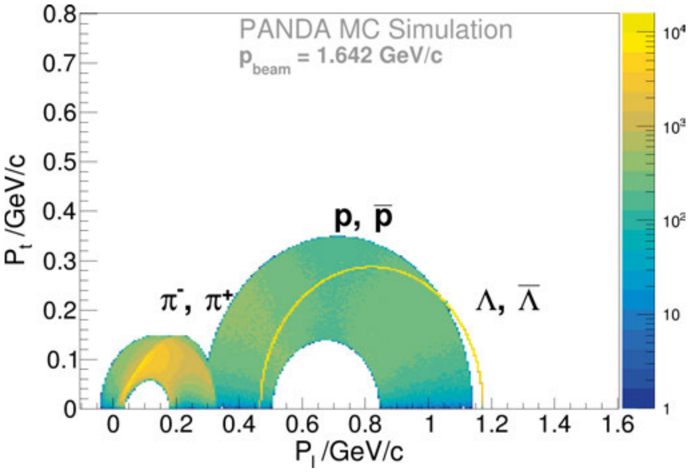


Figure 9.3. The MC p_l *versus* p_t distributions of all decay products.

The distributions represent the undistorted MC truth data and do not include any smearing effects or detector inefficiencies. One can also learn a lot by studying the magnitude of the momentum ($|p|$) versus the polar angle (θ) of emitted particles in the lab frame. The $|p|$ *versus* θ distributions for all generated final state particles are shown in Figure 9.4.

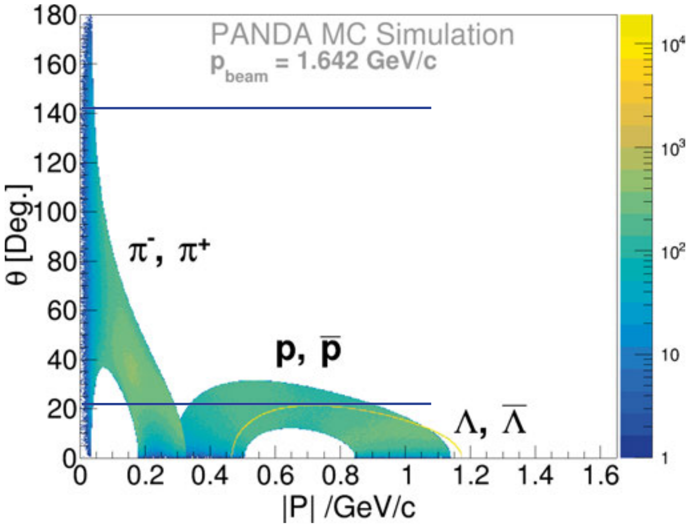


Figure 9.4. The MC $|p|$ *versus* θ distribution of the Λ, $\bar{\Lambda}$ and their decay products. The blue lines indicate the STT acceptance.

Here, we consider the ideal case, *i.e.* the angles and momenta are the Monte Carlo truth, hence not distorted by the detector resolution and acceptance. One should note that the relation between the particle lab angle and the detector coverage is not one-to-one since the tracks may not originate in the interaction point. Hence, a track angle of a particle from a displaced vertex is not the same as the angle of a detector module. The figure shows that the Λ and $\bar{\Lambda}$ hyperons are preferentially emitted in the forward direction, though their decay products go in many different directions. The protons $(\bar{p}, p)$ take the larger share of the momentum, while only a small fraction goes to the pions (π^+, π^-).

The decay vertex distribution of Λ and $\bar{\Lambda}$ hyperons is shown in Figure 9.5. We see that the final state particles often will be produced several centimeters away from the IP. However, a tiny fraction of Λ hyperons is produced at or close to the IP.

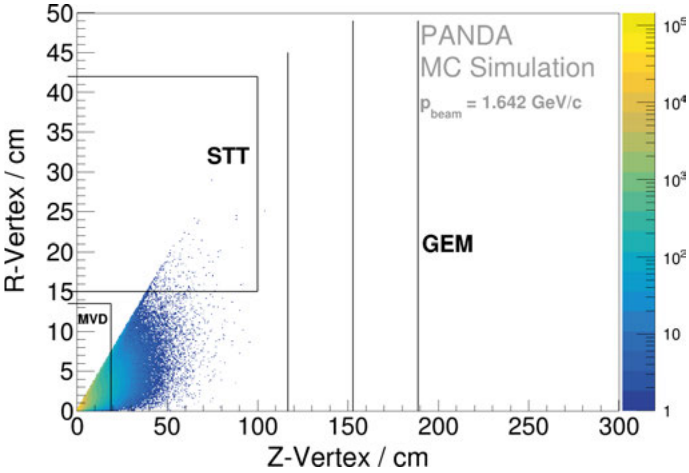


Figure 9.5. The MC decay vertex distribution of Λ and $\bar{\Lambda}$.

9.2 Geometric Deep Learning Pipeline

In this section, we will briefly describe the stages of the GDL pipeline (see Chapter 8) to reconstruct the final state particles $\bar{p}, \pi^+, p$ and π^- from the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$ reaction.

9.2.1 Data Preparation

The data presented in this study consist of 10^5 $\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$ events, generated by PandaRoot. In addition, the total charge from these particles also changes on an event-by-event basis as antiproton in many events escape the STT. The distribution of the number of hits per event is shown in Figure 9.6.

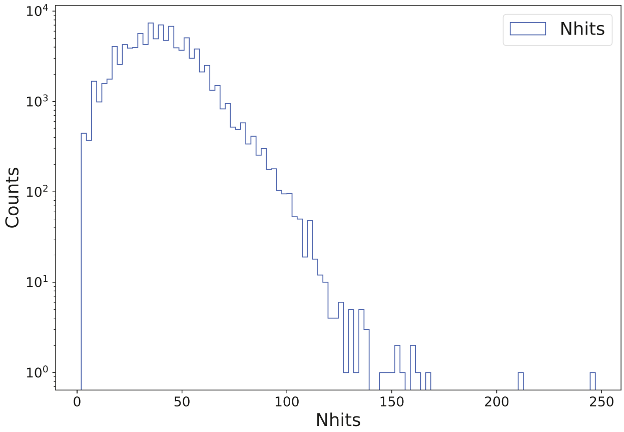


Figure 9.6. The number of hits per event in in STT.

From the machine learning point of view, data produced by the STT for this reaction is relatively sparse. The distribution shows that detector hits in a large fraction of events are around 50 hits. A tiny fraction of events has more than 150 hits.

9.2.2 Edge Construction

The input graphs are built from the edges using the same method as described in Section 8.1.2 with a slight modification: due to the sparsity of data, the condition of edge construction in the adjacent sectors has been relieved. As a result, edge construction is performed in the whole $r\phi-$plane. This choice is made for two reasons: *(i)* to decrease the data sparsity, *(ii)* to have both true and false edges in the data. The latter is particularly important as a significant fraction of events is found with ≤ 3 tracks that are sometimes not in the adjacent sectors. As a result, the input graphs contain only the true edges. The number of edges created in this reaction is shown in Figure 9.7.

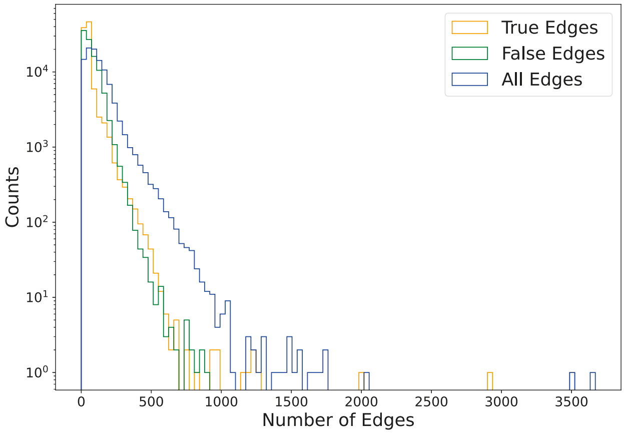


Figure 9.7. The histogram shows true (orange), false (green) and total (blue) edges constructed from the dataset.

In certain events, the number of edges go beyond 1000, these are the events where a pion is found to be curling inside the STT (see Figure 9.13).

9.2.3 Edge Classification

The edge classification is performed using the IGNN on the same lines as in Section 8.1.3. The model output is shown in Figure 9.8.

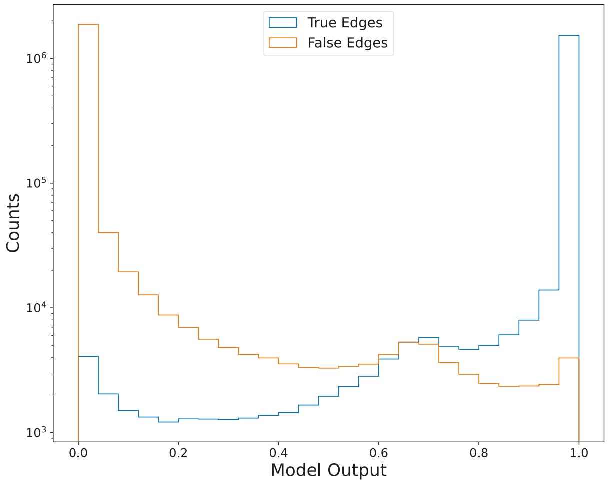


Figure 9.8. Model outputs on test dataset for true and false edges.

The number of true edges peaks around the edge score of 1, whereas the fake edges peak around the edge score of 0. A small bump is discerned in the model output around the edge score of 0.7 (recall Figure 8.4 with the same behavior).

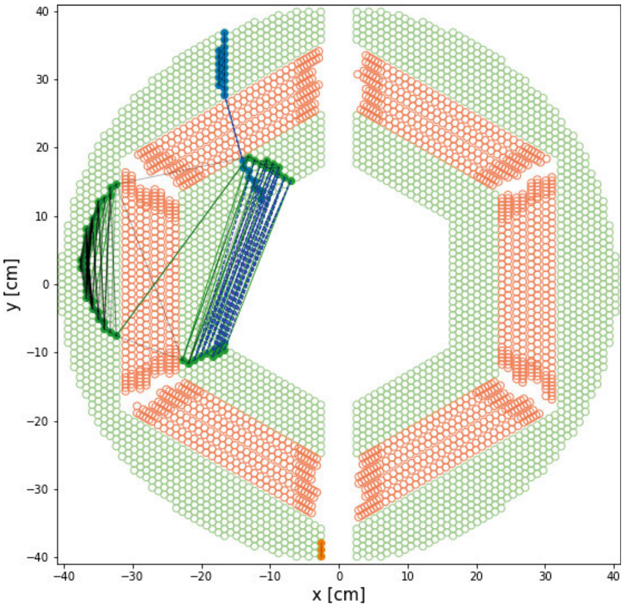


Figure 9.9. Reconstructed event with model output between 0.6 and 0.8.

Figure 9.9 shows filtered edges between edge scores of 0.6 and 0.8 with false negative (blue), false positive (red), and true positive (black) edges, where green is the true graph. By zooming in on this region, one can see that our model has wrongly labeled certain false edges as true and vice versa. This might be due to the bad construction of the ground truth graph for curling particles.

Further, the model is evaluated using the Area Under the ROC (AUC). The ROC is drawn using the edge purity and edge efficiency (see Section 5.4.6 for definitions). In addition, both edge purity and efficiency for various edge score cuts are also measured. The curves are shown in Figure 9.10.

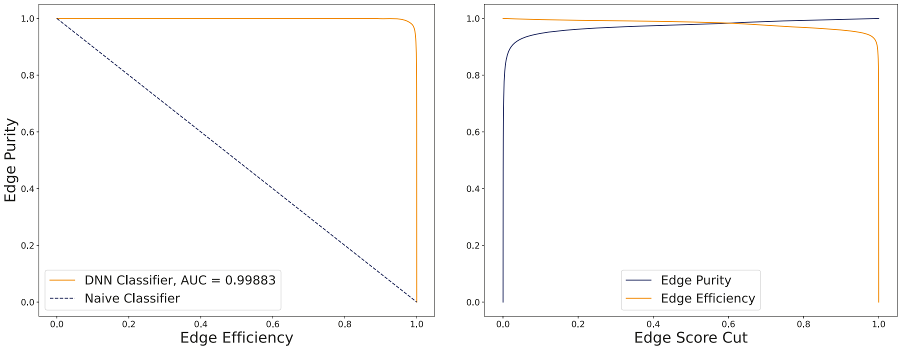


Figure 9.10. Model evaluation: ROC with AUC > 0.9988 *(right)*, edge efficiency and edge purity as a function of edge score cut *(left)*.

The high AUC value indicates high model performance during the training process. Increasing the score cut, the edge purity increases and edge efficiency decreases, and vice versa. The background rejection rate ($1/\varepsilon_{\mathrm{bkg}}$) *versus* the signal efficiency ($\varepsilon_{\mathrm{sig}}$) for various edge scores is shown in Figure 9.11.

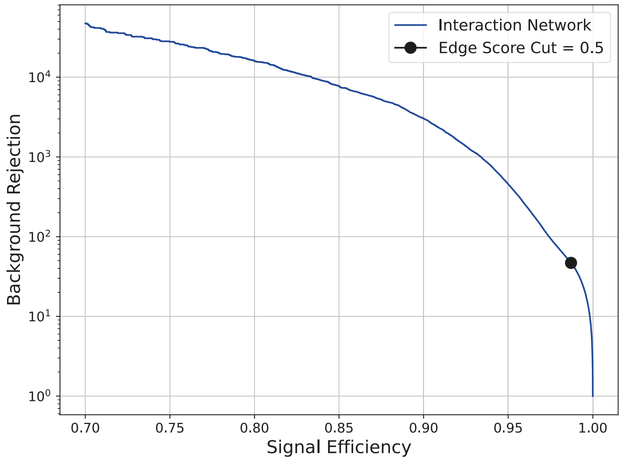


Figure 9.11. Signal efficiency *versus* background rejection rate of IGNN.

The signal efficiency ($\varepsilon_{\mathrm{sig}}$), misidentification rate ($\varepsilon_{\mathrm{bkg}}$) background rejection rate ($1/\varepsilon_{\mathrm{bkg}}$) for edge score cuts of 0.5 and 0.7 are shown in Table 9.1.

Table 9.1. *Signal efficiency, misidentification, and background rejection rates for an edge score cut of 0.5 and 0.7.*

Edge score	Signal efficiency	Misidentification rate	Background rejection rate
0.5	0.987	0.021	46.9
0.7	0.976	0.011	90.1

9.2.4 Track Formation

The DBSCAN method is used as described in Section 6.2.4 for track formation. I scanned the $\varepsilon_{\mathrm{db}}$ parameter for the hyperon data used in this chapter to find its optimal value. For this purpose, I have calculated tracking efficiencies, tracking purity, and clone rate for different values of $\varepsilon_{\mathrm{db}}$, where lower $\varepsilon_{\mathrm{db}}$ values indicate higher edge score cut. For the track evaluation, we require matching fractions greater than 50%. Figure 9.12 shows $\varepsilon_{\mathrm{db}}$ indicated by lines at 0.10 (black), 0.15 (magenta), and 0.20 (black), respectively.

Among the three values of $\varepsilon_{\mathrm{db}}$ indicated by vertical lines at 0.1 (black), 0.15 (magenta), 0.2 (black), the value of 0.15 is chosen for the DBSCAN. This value yields overall tracking efficiency of $87.1 \pm 0.536\%$ with a clone rate of $3.79 \pm 0.087\%$ and a ghost rate of $0.44 \pm 0.621\%$. However, two other choices of ε_{db} produced similar results. In total, $2 \cdot 10^3$ events are used in track formation that took approximately 32.1 seconds (real-time) with eight workers to process all events on an Intel Core i7 CPU with 16 GB of RAM.

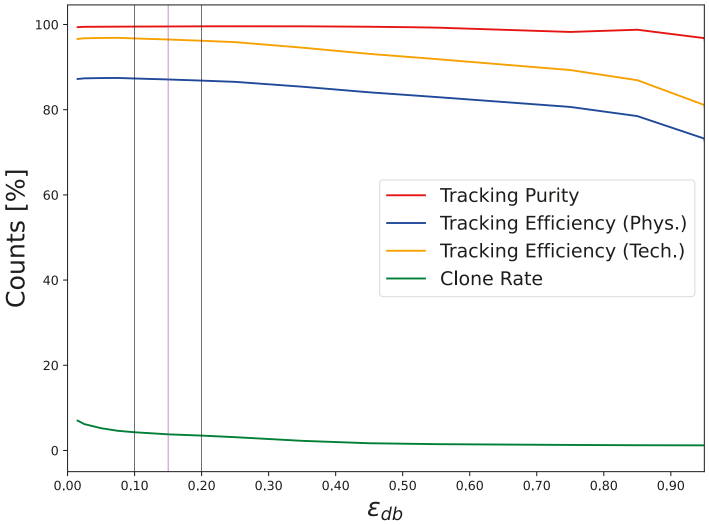


Figure 9.12. The efficiency and purity curves for various ε_{db} values. The smaller ε_{db} values indicate high cut on edge score.

9.3 Track Evaluation

The track evaluation is performed according to the method described in Section 4.2. The evaluation criteria are taken as mentioned in Section 7.2. For a matching fraction of > 50%, $N_t \geq 7$ STT hits for reconstructable particles and $N_r \geq 5$ STT hits for reconstructed tracks, the results are shown in Table 9.2.

Table 9.2. *Tracking efficiencies, ghost rate (GR), clone rate (CR) for $N_t \geq 7$, $N_r \geq 5$ and MF $> 50\%$.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	89.6 ± 0.548	97.1 ± 0.620	0.46 ± 0.609	4.88 ± 0.098

The low clone-rate is expected with this kind of low-multiplicity reaction, which is in contrast to the 5 muon pair case studied in Chapter 7 and Chapter 8. However, there is an occasional occurrence of low-momentum pions curling inside the STT, hence the drop in tracking efficiency. Figure 9.13 shows two example events where a low p_t particle curls inside the STT.

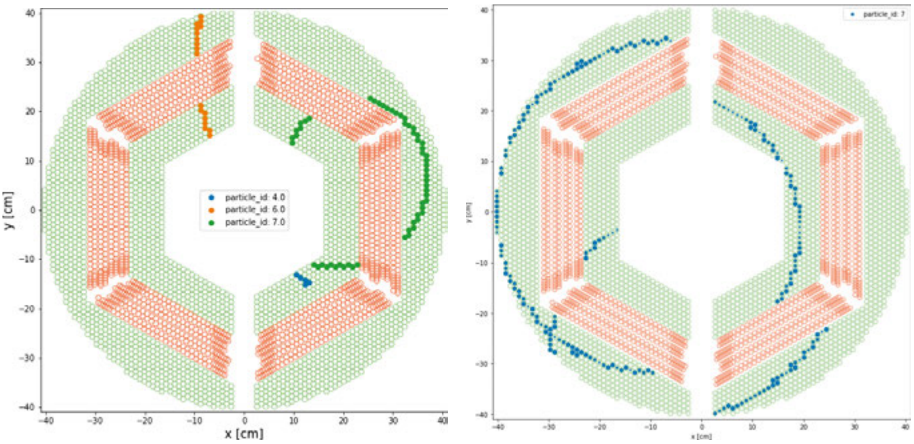


Figure 9.13. Low p_t particles curling inside the STT.

The tracking efficiencies are measured as a function of the transverse momentum (p_t), the polar lab angle (θ), the azimuthal angle (ϕ) and the radial distance (d_0) between the interaction point and the decay vertex:

$$d_0 = \sqrt{v_x^2 + v_y^2} \tag{9.6}$$

where v_x and v_y are vertex positions in XY-plane.

In Figure 9.14, the left panel shows the number of selected, reconstructable, and matched particles as a function of p_t, and the right panel shows the tracking efficiencies as a function of p_t.

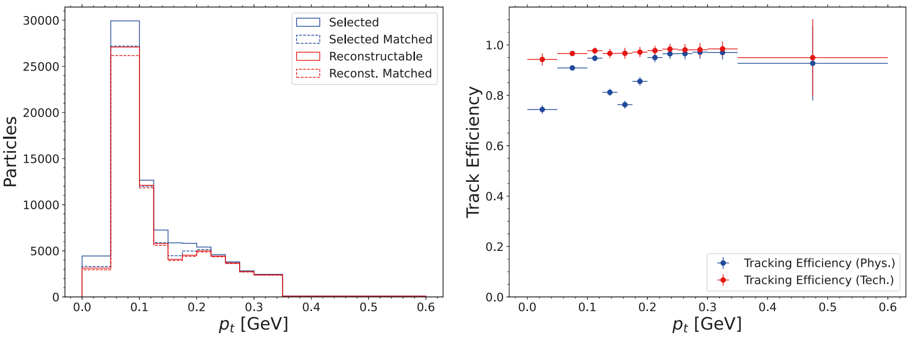


Figure 9.14. Number of selected, reconstructable, and matched particles (*left*), and tracking efficiencies (*right*) as function of p_t.

The large error bars at large p_t arise due to the small number of tracks generated in this region. This is also illustrated in the left panel, which shows the number of selected, reconstructable, and matched particles. The dip in $\varepsilon_{phys.}$ may be attributed to the fact that we have two different particle species, *i.e.* pions and protons, with very different kinematic properties. As a result, the left panel of Figure 9.14 can be understood to contain two distributions, one peaking at low p_t and one peaking at large p_t. The dip is the region between these distributions, and here, the hit acceptance is lower due to the design of the detector. Note that in Figure 9.14, the bin size varies between 0 and 0.6 with a larger size towards high p_t.

The tracking efficiencies as function of θ are shown in Figure 9.15.

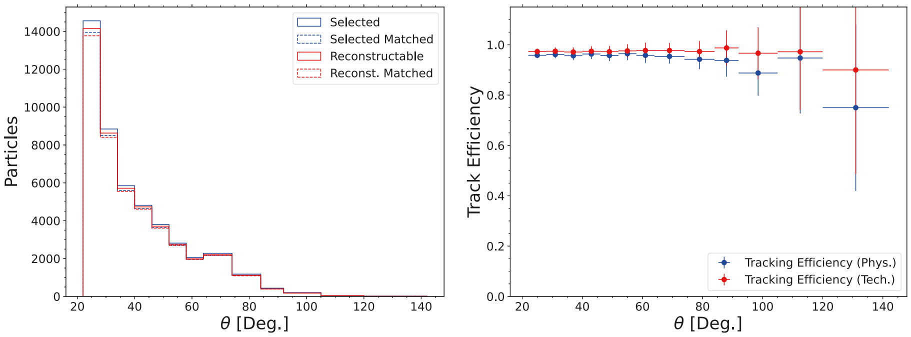


Figure 9.15. Number of generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of θ.

The number of selected, reconstructable, and matched particles are shown in the left panel. Tracking efficiencies as a function of θ are displayed on the right. Both efficiencies are similar up to 80°; however, there are few generated tracks at large polar angles, in line with Figure 9.4.

The tracking efficiencies as a function of ϕ are shown in Figure 9.16. One can conclude that tracking efficiencies are not dependent on the ϕ angle; hence our machine learning method is not biased towards a particular slice of ϕ. As discussed in Chapter 7 and Chapter 8, the small dip at ±90° indicates the gap between two halves of the STT.

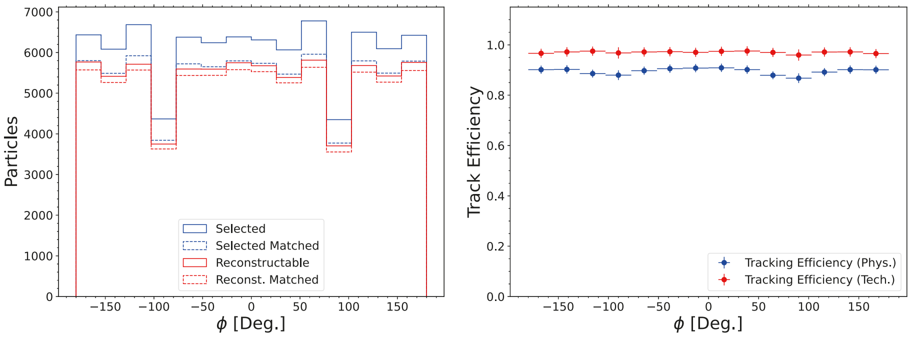


Figure 9.16. Number of generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of ϕ.

The final state particles from the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^{+}p\pi^{-}$ reaction are generated away from the IP due to Λ hyperons flying a measurable distance from the IP before decaying. Therefore, I have investigated how the single-track reconstruction efficiency varies with the distance from the IP. The radial distance parameter d_0 defined in Equation 9.6 quantifies the position of secondary decay vertices of the final state particle from the IP in $r\phi-$plane of STT. The tracking efficiencies as function of d_0 are shown in Figure 9.17:

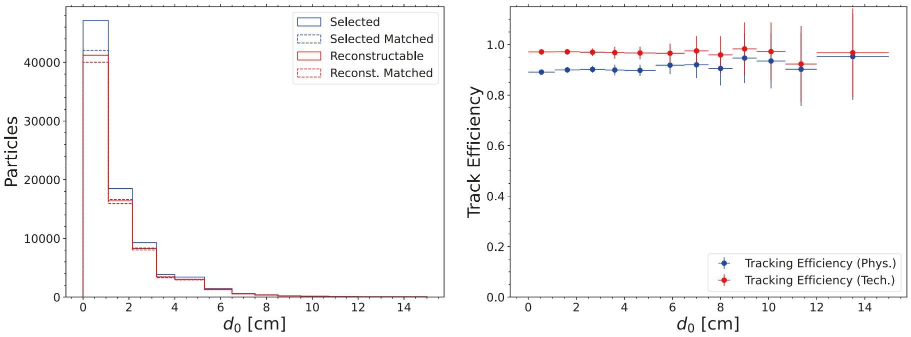
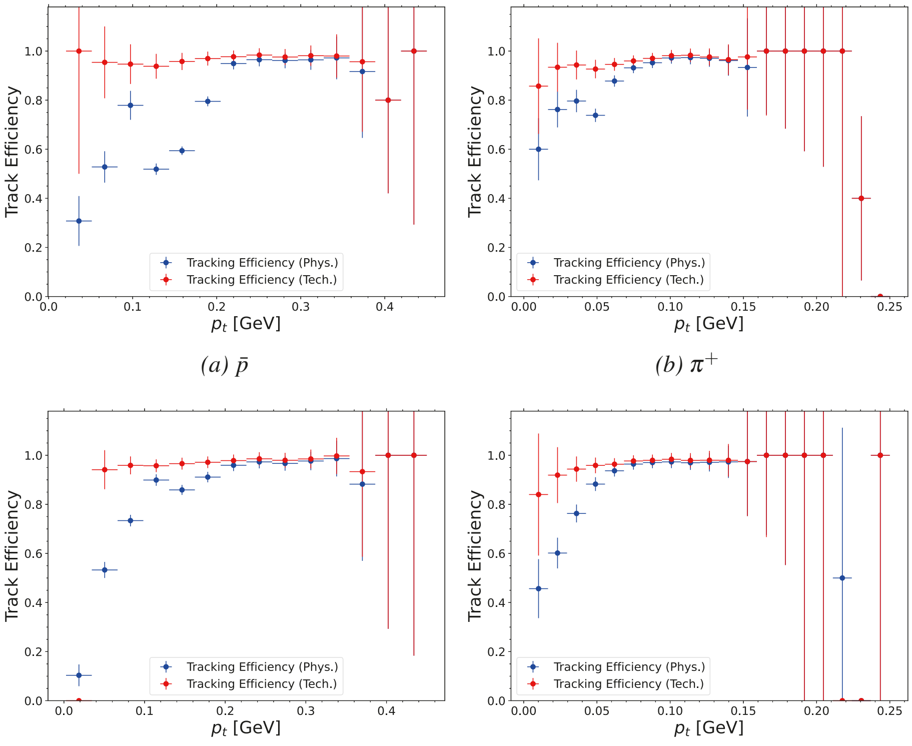


Figure 9.17. Number of selected/generated, reconstructable and matched tracks (*left*) and tracking efficiencies (*right*) as function of the distance to the IP, denoted d_0.

From the left panel showing the number of tracks, it is clear that a significant fraction of particles is generated close to the interaction point. However, a reasonable fraction of final state particles is produced away from the IP. Our method has reconstructed these particles reasonably well up to at least 10 cm away from IP; beyond that, one cannot decisively determine the performance in terms of efficiency, however, the flat dependence of the efficiency on the distance d_0 is promising. Hyperons produced in different reactions, for example, from Ξ or Ω decays, can be used for this purpose.

It is also relevant to study the reconstruction of pions and protons separately. The tracking efficiencies are calculated as a function of p_t and d_0. The results are shown in Figure 9.18. A large fraction of protons are emitted at relatively large p_t *i.e.* between 0.15 to 0.4 GeV/c. On the other hand, most of the pions are emitted below 0.15 GeV/c. It is reassuring and also quite remarkable that our method can reconstruct pions of such low momentum, given that low p_t tracks have been a stumbling block for other track reconstruction algorithms. This is an important result as it will allow to study other hadron physics reactions, from example, $\bar{p}p \rightarrow \bar{\Xi}^+\Xi^-, \bar{\Omega}^+\Omega^-$ as shown in Figure 4.1.



(a) $\bar{p}$

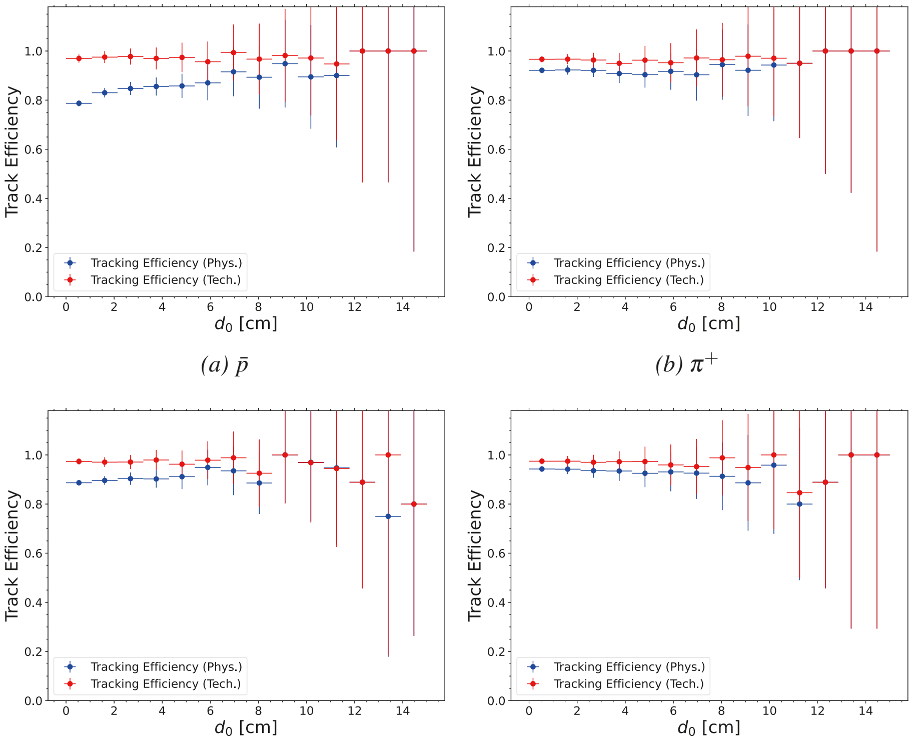
(b) π^+

(c) p

(d) π^-

Figure 9.18. Tracking efficiencies as a function of p_t for protons and pions.

Another important aspect of hadron physics reactions is the secondary decay vertices due to long lived particles especially hyperons. It is also interesting to see how tracking efficiencies vary for each particle with the distance between primary and secondary vertex. To investigate this, tracking efficiencies are shown as a function of d_0 for each proton and pion separately, see Figure 9.19.



(a) $\bar{p}$

(b) π^+

(c) p

(d) π^-

Figure 9.19. Tracking efficiencies as a function of d_0 for protons and pions.

By looking into single track efficiencies (red curves) for all particles, one can conclude that the efficiency is rather flat, and one cannot really say that the efficiency seems to drop at large d_0 due to less number of particles produced at higher d_0 values. As shown in Ref. [18], most Λ hyperons from Ξ^- and Ω^- decays (see Figure 4.1 for reaction topologies) also decay at $d_0 < 15$ cm. Hence, this algorithm seems promising for the hyperon channels of interest.

9.4 Summary of Results

Similar to the results shown in Table 8.3, the tracking efficiencies, ghost rates, and clone rates are calculated for various MF greater than 50% (loose), 75% (moderate) and 95% (strict), and two different values of N_r that are taken as 5 and 6 for reconstructed tracks. The values of N_t is kept as ≥ 7 STT hits for reconstructable particles. The results are summarized in Table 9.3.

Table 9.3. *Tracking efficiency, ghost rate (GR), clone rate (CR) for loose, moderate and strict matching criteria with at least 5 and 6 hits for reconstructed tracks.*

N_t	N_r	MF [%]	$\varepsilon_{phys.}$ [%]	$\varepsilon_{tech.}$ [%]	GR [%]	CR [%]
7	5	> 50	89.6 ± 0.548	97.1 ± 0.620	0.46 ± 0.609	4.88 ± 0.098
7	5	75	84.3 ± 0.524	91.1 ± 0.591	2.05 ± 0.601	8.97 ± 0.135
7	5	95	79.4 ± 0.502	85.7 ± 0.565	3.45 ± 0.595	12.7 ± 0.163
7	6	> 50	87.1 ± 0.536	96.5 ± 0.617	0.44 ± 0.621	3.79 ± 0.087
7	6	75	82.2 ± 0.514	91.1 ± 0.591	1.87 ± 0.614	7.71 ± 0.127
7	6	95	77.5 ± 0.493	85.7 ± 0.565	3.26 ± 0.608	11.5 ± 0.158

Using the minimum hits for reconstructed tracks (N_r) as 6 STT hits, the track efficiency for reconstructable tracks, $\varepsilon_{tech.}$, is slightly higher than $N_r \geq 5$ STT hits for various MFs. The ghost and clone rates also decrease when $N_r \geq 6$ is used instead of $N_r \geq 5$. However, in either case, the efficiencies drop when the value of the matching fraction is increased, resulting in a large ghost rate and clone rate. Overall, the tracking efficiencies are acceptable for the various criteria we investigated [29] for 50% and 75% matching fractions.

The track reconstruction was performed in the $r\phi-$plane of the STT; aforementioned results can be further improved by including the $z-$axis and incorporating the MVD and GEM detectors. In this manner, not only will the hit statistic per event improve, but in addition, more final state particles will be included in the sample. Moreover, handling the curly tracks with a robust ground truth algorithm will improve the above results. Both of these tasks are under development. To record the forward going $\bar{\Lambda}$ hyperons, the forward tracking stations in the Forward Spectrometer will play an essential part in hyperon reconstruction. The particle reconstruction with the forward tracking stations using geometric deep learning has been explored in Ref. [19].

Part III:
Hyperon Simulations with Realistic Target Profiles

10. Goals and Strategies

To study the effects of residual gas from the cluster-jet target on the possibility to distinguish a reaction of interest from the background, the reaction $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ at a beam momentum of 1.64 GeV/c has been used as a benchmark channel. The reason for this choice is that this reaction has been rigorously studied at the PS185 experiment at this beam momentum and hence the features of this reaction are well-known. This study is performed for both point-like and extended target profiles, and for both the reaction of interest and the most likely background channel. To compare the different target scenarios, a figure-of-merit (FoM) is defined that is given in Section 10.1.

10.1 Figure of Merit

In this study, we define the figure-of-merit (FoM) as the expected ratio of events from the signal channel ($p\bar{p} \to \Lambda\bar{\Lambda} \to \bar{p}\pi^+ p\pi^-$) and the non-resonant background ($p\bar{p} \to \bar{p}\pi^+ p\pi^-$) channel weighted by cross sections and decay branching fractions:

$$\mathrm{FoM} = \frac{\varepsilon(p\bar{p} \to \Lambda\bar{\Lambda})}{\varepsilon(p\bar{p} \to \bar{p}\pi^+ p\pi^-)} \cdot \frac{\sigma(p\bar{p} \to \Lambda\bar{\Lambda}) \cdot BR(\Lambda \to p\pi^-)^2}{\sigma(p\bar{p} \to \bar{p}\pi^+ p\pi^-)} \tag{10.1}$$

Here, ε denotes the global reconstruction efficiency after applying a set of event selection criteria. The cross sections and branching fractions are given in Table 10.1. We will come back to these numbers in in Section 11.1.1 and Section 11.1.2.

Table 10.1. *Cross sections and branching fractions for signal as well as non-resonant background as described in Section 11.1.1 and Section 11.1.2.*

$\sigma(\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-)$	$BR(\Lambda \to p\pi^-)$	$\sigma(\bar{p}p \to \bar{p}\pi^+ p\pi^-)$
$64.1 \pm 0.4 \pm 1.6\,\mu b$	$63.9 \pm 0.5\%$	$15.4 \pm 5.2\,\mu b$

This figure-of-merit tells us how well we will be able to suppress the background from a reaction with the same final state particles. The $\bar{p}, \pi^+, p$ and π^- from the non-resonant channel originate in the IP, whereas the corresponding particles from Λ and $\bar{\Lambda}$ decays originate from the decay vertices which are typically at least a few centimeters away from the IP. Furthermore, the $p\pi^-$ ($\bar{p}\pi^+$) from a Λ ($\bar{\Lambda}$) decay have different kinematic properties (*e.g.* invariant mass) than those produced in a non-resonant reaction.

10.2 Strategy

The strategy adopted in this work is to calculate the FoM for an ideal, *i.e.* point-like, target, denoted as IdealIP case (Chapter 11). Next, using the same analysis procedure as in the case of IdealIP, the FoMs of the extended target profile scenarios are calculated (Chapter 12). I have considered two extended target profiles, denoted NormalIP and NormalIP+Cryo (see Section 3.5). Finally, the FoM from IdealIP, NormalIP and NormalIP+Cryo cases are compared (Section 12.3).

11. Hyperon Reconstruction using a Point-like Target

The $\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+p\pi^-$ reaction been rigorously studied within the PANDA simulation framework [9, 10, 17, 18, 174, 175] but until now only with an ideal, *i.e.* a point-like, target. In the analysis presented here, I use the event selection criteria defined in Ref. [10], but with an updated version of the PandaRoot software. By using the same criteria, we can compare this analysis with the analysis to the one in Ref. [10]. This provides an important benchmark of the updated software. However, the main purpose of simulating the point-like case is to have a reference, or a "best case scenario", to compare with the simulations where a more realistic target profile is used.

11.1 Data Generation

For the analysis, the signal channel as well as the non-resonant background channels are simulated at a beam momentum of 1.642 GeV/c. To quantitatively compare the two samples, *e.g.* when calculating the FoM, one needs to apply relative weights to each sample based on their cross-sections. The data generation and weight calculation are discussed below.

11.1.1 The Signal

The signal channel of interest is the $\bar{p}p \to \bar{\Lambda}\Lambda$ reaction, where the $\bar{\Lambda}\Lambda$ states are produced through the strong interaction according to the model where the $\bar{\Lambda}$ is produced with a forward peaking distribution in the CM system, see Section 9.1 and the figure therein. The Λ hyperons further decay through the weak interaction, where the most prominent decays $\Lambda \to p\pi^-$ and $\bar{\Lambda} \to \bar{p}\pi^+$ have a branching fraction of $BR(\Lambda \to p\pi) = 63.9 \pm 0.5\%$ [5]. We assume the same BR for $\bar{\Lambda} \to \bar{p}\pi^+$.

The weak decays $\Lambda \to p\pi^-$ and $\bar{\Lambda} \to \bar{p}\pi^+$ are simulated using a flat phase space distribution *i.e.* ignoring the polarization and spin correlations. This is considered sufficiently realistic for this comparative study.

11.1.2 Non-resonant Background

The most relevant background considered for this study is the non-resonant $\bar{p}p \to \bar{p}\pi^+p\pi^-$ channel, where final state particles are the same as in the signal sample. The main difference is that in the case of the non-resonant background, all final state particles originate in the IP. Also the background sample is simulated using a flat phase space distribution for all final state particles with no intermediate Λ or $\bar{\Lambda}$.

There is no established world-average value for the production cross-section of the non-resonant channel. However, several collaborations have reported different, somewhat conflicting values. The differences could be due to model dependencies arising from the multitude of possible intermediate states ($\Delta\Delta$, $\bar{p}p\rho$, etc). Therefore, the weighted average cross section is calculated to be $\sigma(\bar{p}p \to \bar{p}\pi^+ p\pi^-) = 15.4 \pm 5.2\,\mu$b as described in Ref. [10].

11.1.3 Sample Sizes and Weights

I have generated 10^6 events of each channel, and propagated them through the PandaRoot simulation chain explained in Subsection 3.6.1. The large simulated samples ensures small statistical uncertainties. To compare the yield of the non-resonant background channel with the signal reaction, we define a weight to account for the difference and the reduction due to the branching fraction of the Λ and $\bar{\Lambda}$ decay:

$$w_{\text{bkg}} = \frac{N_{\text{signal}}}{N_{\text{bkg}}} \frac{\sigma(\bar{p}p \to \bar{p}\pi^+ p\pi^-)}{\sigma(p\bar{p} \to \Lambda\bar{\Lambda}) \cdot BR(\Lambda \to p\pi)^2} \tag{11.1}$$

where N_{signal} and N_{bkg} are the sizes of signal and background samples. By applying these weights, the obtained yields can be compared using the Figure of Merit defined in Equation 10.1, hence the yields will give an idea of the expected signal-to-background ratios. The values of cross-sections and branching ratios are given in Table 10.1. In total, one million events are generated for each channel. The relative weighting factors for non-resonant background summarized in Table 11.1.

Table 11.1. *The sample size, cross-section and weight factors of signal and non-resonant background channels.*

Channel	$\bar{p}p \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$	$\bar{p}p \to \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
Cross section [μb]	64	15.4
Weighting factor	1	0.589

11.2 Analysis Procedure

The analysis is subdivided into two stages: the *pre-selection* and the *final selection*. The two-staged event selection is motivated by the simplicity in the data handling and the computing time for large data samples. In principle, both stages can be combined into one single event selection stage.

In the pre-selection, a coarse set of criteria is imposed to reconstruct the full event topology and to reject poorly reconstructed events. At this stage, an ideal particle identification (PID) match is performed for all charged tracks *i.e.* each track is correctly assigned a particle type. At present, the PID algorithms of PandaRoot are not fully tested, especially not for hyperons. Furthermore, in this study it is desirable to minimize all uncertainties that can be attributed to anything else than the interaction volume. In the final selection, emphasis is given to the capability of the event selection algorithm to suppress non-hyperon background. After the final selection, one can calculate the final reconstruction efficiency. These steps are explained in the following sections.

11.2.1 Pre-selection

In pre-selection, exclusive $p\bar{p} \to \bar{\Lambda}\Lambda$ reaction where $\bar{\Lambda} \to \bar{p}\pi^+$ and $\Lambda \to p\pi^-$ will be reconstructed. The pre-selection criteria is applied according to Ref. [10] that were optimized using the general hadronic background simulation using the Dual Parton Model (DPM) [176]. This criteria will first be shortly summarized and then discussed in more detail later.

1. Events with at least all final states of reaction are considered *i.e.* one p, one π^-, one $\bar{p}$ and one π^+ are considered. In addition, a mass selection on reconstructed $p\pi^-$ and $\bar{p}\pi^+$ is applied.
2. A vertex fit, requiring that the p and π^- ($\bar{p}$ and π^+) come from the same point in space, is applied. This fit has one one over-constraint (1C). The combinations with a fit probability of less than 1% are rejected, while those with a larger probability are labeled as Λ or $\bar{\Lambda}$ candidates. For multiple Λ or $\bar{\Lambda}$ candidates per event, the candidate with the smallest χ^2 is kept for further analysis.
3. The Λ and $\bar{\Lambda}$ candidates are combined to reconstruct the initial kinematics of the $\bar{p}p-$system. A successful kinematic fit, based on conservation of the initial $\bar{p}p$ four momentum, is required. Since the fit has four over-constraints, it is denoted a 4C fit.

After applying the aforementioned pre-selection criteria, the complete reaction topology can be reconstructed. At this point, each event contains exactly one $p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+p\pi^-$ candidate. In the following, we will discuss the selection criteria in more detail.

11.2.1.1 Pre-selection Mass Cut

At the start, events are selected if the final state contains at least one of each of the expected p, π^-, $\bar{p}$ and π^+, otherwise the event is rejected. Furthermore, all possible combinations of $\bar{p}\pi^+$ and $p\pi^-$ are taken into account to reconstruct

Λ's. At this point, the following coarse mass selection is applied:

$$|m(\bar{\Lambda}) - m(\bar{p}\pi^+)| < 0.3\ \text{GeV/c}^2 \tag{11.2}$$

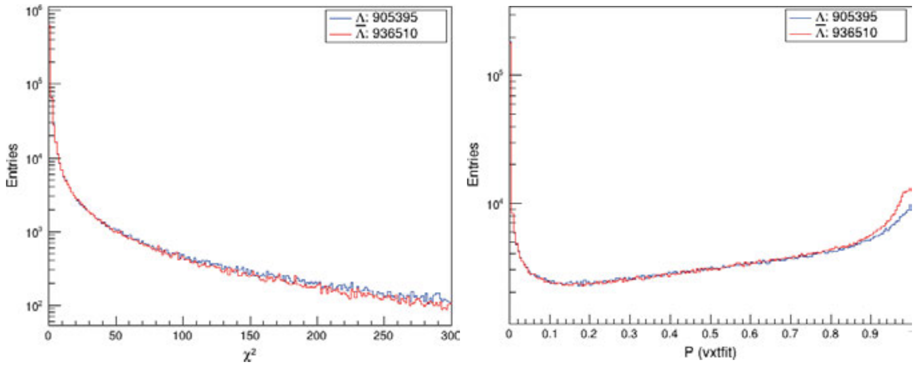
$$|m(\Lambda) - m(p\pi^-)| < 0.3\ \text{GeV/c}^2 \tag{11.3}$$

This mass selection cut on the invariant mass of $\bar{p}\pi^+$ and $p\pi^-$ removes poorly reconstructed events.

11.2.1.2 Vertex Fit and Selection of Best Candidate

The distinct feature of the $p\bar{p} \to \bar{\Lambda}\Lambda$ reaction is the presence of two displaced vertices arising from the weak decays: $\bar{\Lambda} \to \bar{p}\pi^+$ and $\Lambda \to p\pi^-$. Since the Λ (and $\bar{\Lambda}$) is neutral, it leaves no signal in the detector but has to be reconstructed from its daughter particles whose tracks typically originate in a point separated from the IP.

Two independent vertex fits are performed on all possible combinations of $p\pi^-$ and $\bar{p}\pi^+$. The location of the Λ decay vertex is unknown since the Λ is neutral. Hence, the vertex position is the unmeasured parameters in the fit. In a solenoid magnetic field, two constraints are associated with each track originating from a common vertex [177, 178] giving in total four constraints per vertex. The number of over-constraints is then given by the total number of constraints (here 2*2), minus the number of unknowns (3). Hence, we have one over-constraint and therefore we refer to the vertex fit as a 1C fit



(a) *(b)*

Figure 11.1. The vertex fit χ^2 and probability distribution for Λ (black) and $\bar{\Lambda}$ (green) for signal sample.

In addition, the fit probability is required to be greater than 0.01, *i.e.* P > 0.01, to exclude poorly reconstructed candidates. As $P \to 1$, the increase in the number of events is due to the covariance matrix from the Kalman Filter. After applying the vertex fits, more than one $\bar{\Lambda}\Lambda$ pair might exist in one event. To select the best pair of the event, the χ^2 of the vertex fits are compared, and the candidate with the smallest χ^2 value is chosen for further analysis. In the end, at most one $\bar{\Lambda}\Lambda$ pair exist per event. The fit probabilities are shown in Figure 11.1 for the signal sample.

11.2.1.3 Four Constraint (4C) Fit

Once the vertex fits are performed, each event consists of at most one $\bar{\Lambda}\Lambda$ pair from which we can construct the initial $\bar{p}p-$ system that is kept for further analysis. A four-constraint (4C) fit is applied which restricts the four-momentum of the $\bar{\Lambda}\Lambda$ pair to the initial $\bar{p}p-$ system.

As a result of this fit, the four-momentum vectors of hyperons are re-calculated with reduced but correlated uncertainties. The new values of the four-momentum vectors will be used in final selection. The 4C fit probability for both signal and non-resonant background is shown in Figure 11.2.

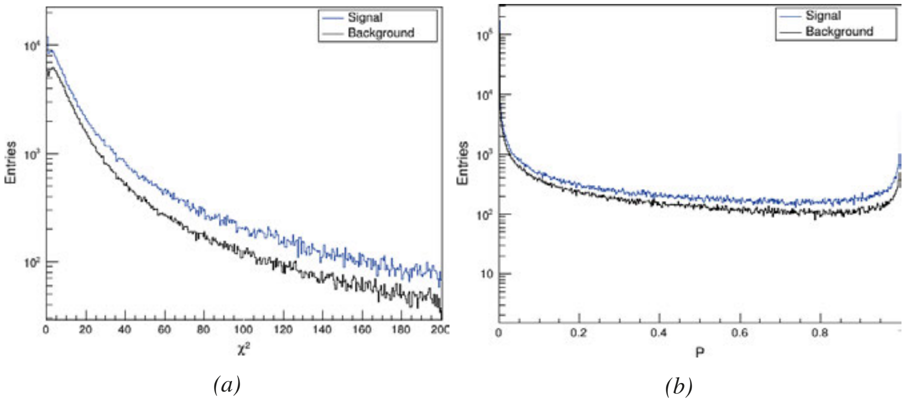


Figure 11.2. The 4C fit χ^2 and probability for signal sample (blue), the non-resonant background (red). Histograms are scaled according to Table 11.1.

11.2.1.4 Results from the Pre-selection

Once the pre-selection criteria are applied, the reconstruction efficiency for the $\bar{p}p \to \bar{\Lambda}\Lambda$ reaction can be calculated. The results are summarized in Table 11.2.

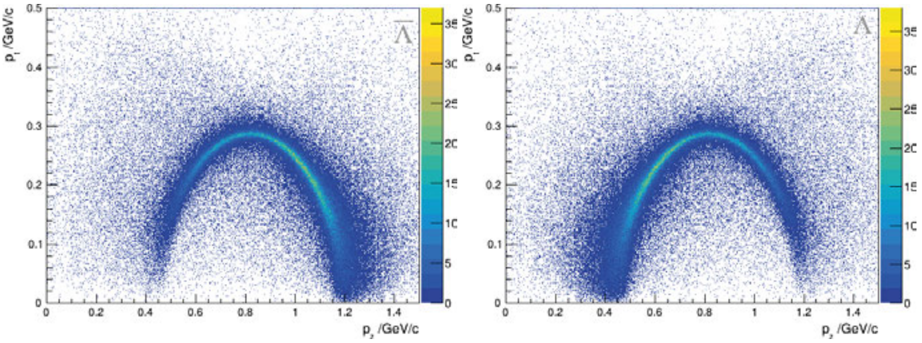
We see that there is an excess of reconstructed particles with respect to the generated ones, in particular for $\bar{p}$ and p in both samples. There are two known reasons for this. First, a tiny fraction of the surplus is due to double-counting of particles by the IdealTrackFinder. For example, when a particle hits the middle of a silicon strip sensor in the MVD (*i.e.* close to the boundary of two sensor cells), the hit can be included in two different tracks. The issue of double-counting is specific to the IdealTrackFinder and its effect is almost negligible. The second, and more important, reason is the production of secondary particles. In PandaRoot, this is done in the particle transport stage by GEANT and it mimics the production of secondaries that we expect in the real experiment. Hence, these particles are of a kind that we need to handle in

Table 11.2. *The reconstruction efficiency after the pre-selection for signal and non-resonant background samples. The final states are obtained after the PID selection. The Λs are obtained after the best vertex fit χ^2 selection while the $\bar{p}p$ state after the successful 4C fit.*

Channel	$p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$	$p\bar{p} \to \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
$\bar{p}$	1043690	1043463
p	1196514	1101627
π^+	1004109	935421
π^-	834721	917451
Λ	905395	950820
$\bar{\Lambda}$	936510	889489
$\bar{p}p \to \bar{\Lambda}\Lambda$	332079	379841
Efficiency %	33.20 ± 0.07	37.98 ± 0.07

real data. Secondary particles are included during the track building process, and contribute significantly to the particle yield. However, at later event selection stages, when applying *e.g.* mass cuts, vertex fits and kinematic fits, these secondaries will be filtered out.

The longitudinal versus the transverse momentum distributions of the $\bar{\Lambda}$ and Λ are shown in Figure 11.3 for signal sample. There is substantial amount of correctly reconstructed Λ hyperons and $\bar{\Lambda}$ anti-hyperons along the parabolas shown in panels *(a)* and *(b)* of Figure 11.3.

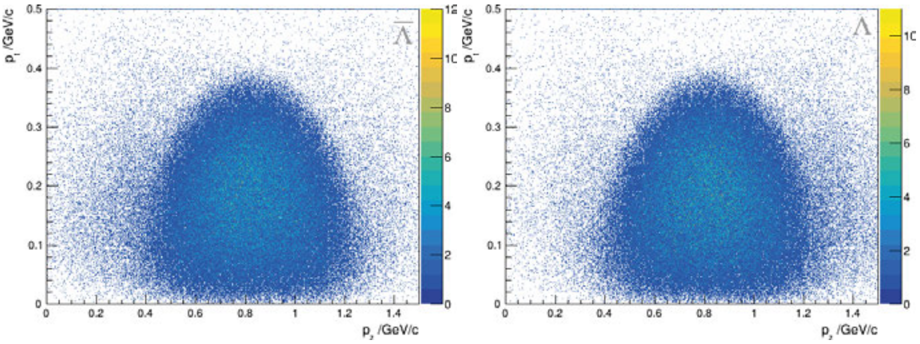


(a)

(b)

Figure 11.3. The longitudinal (p_z) versus transverse (p_t) four-momentum distributions of $\bar{\Lambda}$ (left) and Λ (right) candidates after pre-selection for signal.

The signal sample contain background events distributed evenly in the momentum space. In panels *(c)* and *(d)* of Figure 11.4, the non-resonant background appears as a blob in the momentum space.



(a) *(b)*

Figure 11.4. The longitudinal (p_z) versus transverse (p_t) four-momentum distributions of $\bar{\Lambda}$ (left) and Λ (right) candidates after pre-selection for non-resonant background.

11.2.2 Final Selection

After the pre-selection, the sample is refined by applying additional quality criteria. The purpose of the final selection is not only reduce the background contribution, but also to exclude poorly reconstructed events as well. The following final selection is used:

1. Selection on χ^2 distribution of 4C fit
   - $\chi^2 < 100$
2. Selection on invariant mass of $\bar{p}\pi^+$ and $p\pi^-$
   - $|m_{\text{fit}}(\bar{p}\pi^+) - m_{\text{PDG}}(\bar{\Lambda})| < 5 \cdot \sigma_{m_{\text{fit}}}(\bar{p}\pi^+)$ GeV/c^2
   - $|m_{\text{fit}}(p\pi^-) - m_{\text{PDG}}(\Lambda)| < 5 \cdot \sigma_{m_{\text{fit}}}(p\pi^-)$ GeV/c^2
3. Selection on the total $z-$distance from IP
   - $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm

The aforementioned final criteria are discussed in more detail in the following.

11.2.2.1 Selection on χ^2 Distribution of 4C Fit

As described in Section 11.2.1.3, the 4C fit is applied to ensure four-momentum conservation. In addition, we require χ^2 to be smaller than 100. The motivation is to remove events where the particle momenta are poorly reconstructed. Though originating from the reaction of interest, the data quality is insufficient for e.g. spin studies. More details are given in Ref. [10]. This cut removes approximately 22% of the events from the sample whereas $\approx 19\%$ of events are removed from the non-resonant background sample. It should be noted that this cut is more effective in reducing the number of events with a different number of particles in the final state. The χ^2 distribution together with the cut value is shown in Figure 11.5.

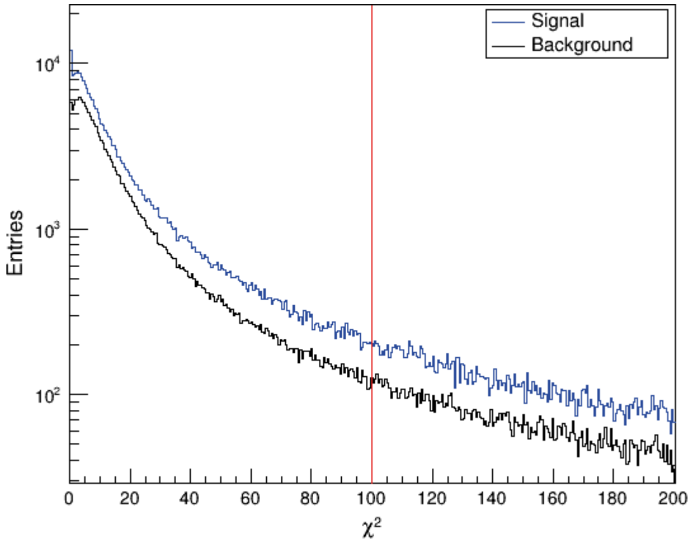
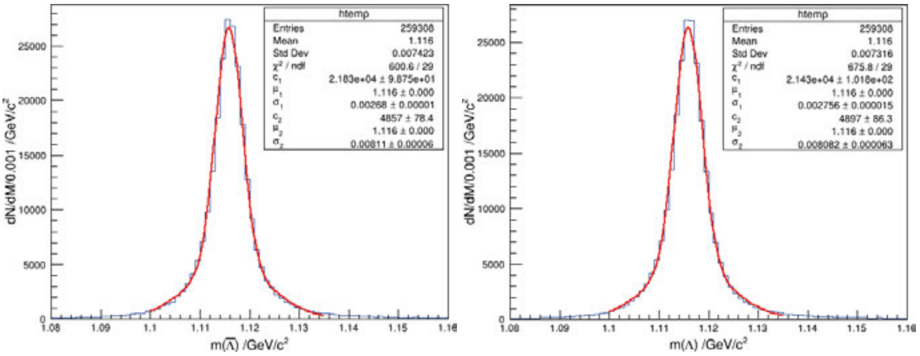


Figure 11.5. χ^2 distribution of signal (red) and non-resonant background (blue). The cut value is marked with a line. Histogram are scaled according to Table 11.1.

11.2.2.2 Selection on Invariant Mass of $\bar{p}\pi^+$ and $p\pi^-$

After the fit quality criterion, a cut on the invariant mass of $\bar{p}\pi+$ and $p\pi^-$ is applied. It is assumed that the shape of the peak corresponding to a $\bar{\Lambda}$ or a Λ is double-Gaussian, *i.e.* sum of two Gaussian distributions. The widths quantify the resolution of the detector and the reconstruction. The double Gaussian fits for invariant mass distributions are shown in Figure 11.6 for signal sample.



(a) *(b)*

Figure 11.6. The invariant mass resolution is obtained by double Gaussian fit on the invariant mass distributions (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ for the signal sample.

The parameters of double Gaussian function used for this fit are represented by c_1, μ_1, σ_1 for the first Gaussian (G1) and c_2, μ_2, σ_2 for the second Gaussian (G2). For example, in Figure 11.6(a), the G1 function fit yield $2.164 \cdot 10^4$, whereas G2 only yields 4846. Therefore, G1 has been used to define the mass resolution of $\bar{\Lambda}$. The double Gaussian fit to the invariant mass distributions yields the mean values and widths for the Gaussian with the larger contribution to the fit as shown in Table 11.3.

Table 11.3. *Invariant mass of reconstructed $\bar{\Lambda}$ and Λ.*

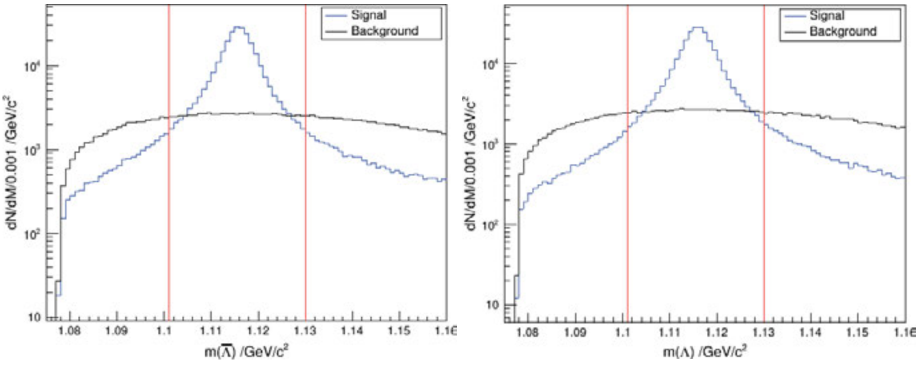
Particle	$\mu_{m_{\text{fit}}}$ $[GeV/c^2]$	$\sigma_{m_{\text{fit}}}$ $[GeV/c^2]$
$\bar{p}\pi^+$	1.11590 ± 0.00001	0.00268 ± 0.00001
$p\pi^-$	1.11595 ± 0.00001	0.00276 ± 0.00002

The mass values are consistent with the PDG value of the Λ: $m_{\text{PDG}}(\bar{\Lambda}) = 1115.683 \pm 0.006$ MeV/c^2. The mass resolution is used to define invariant mass cut and is defined as the five standard deviation around the PDG value of the Λ mass:

$$|m_{\text{fit}}(\bar{p}\pi^+) - m_{\text{PDG}}(\bar{\Lambda})| < 5 \cdot 2.680 \cdot 10^{-3} \text{ GeV/c}^2 \tag{11.4}$$

$$|m_{\text{fit}}(p\pi^-) - m_{\text{PDG}}(\Lambda)| < 5 \cdot 2.756 \cdot 10^{-3} \text{ GeV/c}^2 \tag{11.5}$$

The invariant mass distributions of both Λ and $\bar{\Lambda}$ for all samples is shown in Figure 11.7.



(a) *(b)*

Figure 11.7. The invariant mass distributions of (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ of signal (blue) and non-resonant background (black) with the cut value (red lines). The histograms are scaled according Table 11.1.

The mass window is marked by the vertical lines on this figure. It should be noted that the distribution of non-resonant background is nearly flat within this mass window while the signal has the shape of a well-pronounced peak. However, the background is larger than the signal in some regions. This is because the mass window is optimized by generic hadronic background from the DPM generator rather than non-resonant background channel [10].

11.2.2.3 Selection on the Total $z-$distance from IP

An important aspect of ground-state hyperons is their relatively long life time, meaning they travel a measurable distance in the detector before decaying. This results in a distinct topology with decay vertices well separated from the point of production (IP). A selection criterion can be imposed on the displaced vertex position. In a fixed target experiment, such as PANDA, the particles are boosted along the beam, or $z-$axis, which makes this displacement particularly prominent along $z-$axis (see Figure 9.5). Therefore, we require the sum of the displacement for Λ and $\bar{\Lambda}$ decays to be larger than 2 cm, *i.e.* $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm [10].

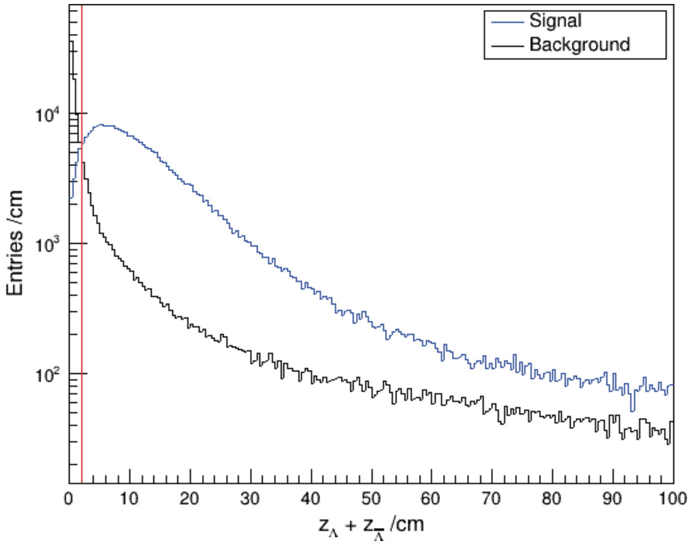


Figure 11.8. The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution of signal (blue) and non-resonant background (black) samples with cut value (red). The histograms are scaled according to the Table 11.1.

The Figure 11.8 shows the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution for both signal and non-resonant background samples. The shape of signal sample is similar to the exponential attenuation which is expected to be the case for such a distribution. The background sample, however, peaks around $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) = 0$. A cut value of $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm (shown as vertical line) removes a large fraction of background events from the signal.

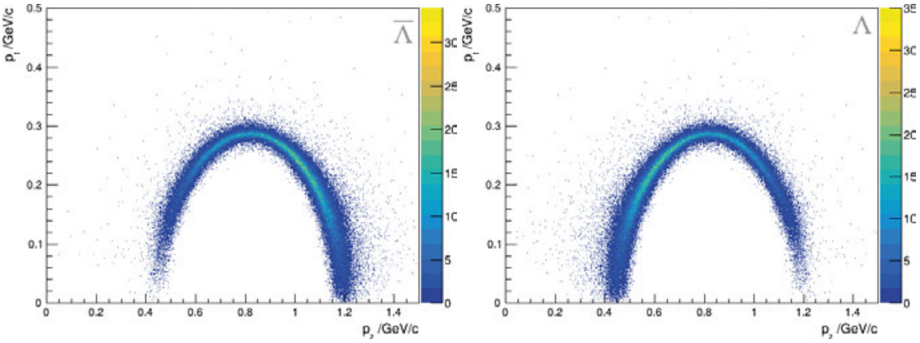
11.2.2.4 Results from the Final Selection

After the final selection is performed, the final efficiency is calculated. The results are summarized in Table 11.4 for the signal as well as the non-resonant background samples. It is important to note that the selection procedure on signal and non-resonant background samples is performed independently of each other.

Table 11.4. *The reconstruction efficiency after the final selection for signal and non-resonant background. The weighted number of events is calculated according to Table 11.1.*

Channel	$p\bar{p} \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$	$p\bar{p} \rightarrow \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
Pre-selection	332079	379841
$\chi^2 < 100$	259308	307342
Mass cut	222344	28787
$z_{\bar{\Lambda}} + z_{\Lambda} > 2$ cm	200772	3443
Efficiency %	20.08 ± 0.05	0.34 ± 0.01

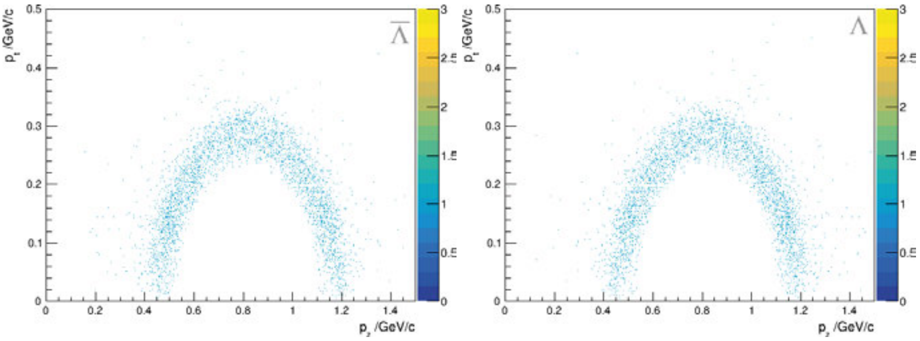
In Figure 11.9, the longitudinal (p_z) versus transverse (p_t) momentum distributions of Λ and $\bar{\Lambda}$ are shown. In the signal sample in Figure 11.9 (a) and (b), the final selection has removed a large number of background and poorly reconstructed signal events, compared to the corresponding figure from the pre-selection, i.e. Figure 11.3. The remaining events lies along the parabolic curves as show in Figure 11.9.



(a) *(b)*

Figure 11.9. The p_z versus p_t four-momentum distributions of $\bar{\Lambda}$ (left) and Λ (right) candidates after final selection for signal sample.

The momentum distribution of the non-resonant background sample is shown in Figure 11.10 (a) and (b) for both the $p\pi^-$ ($\bar{p}\pi^+$) falsely identified as Λ ($\bar{\Lambda}$). The signal events are what remain after applying the final selection. It shows that only a tiny fraction of 0.32% events remain. The efficiency in our case is slightly better than in the simulations performed in Ref. [27], where an older version of the PandaRoot software was used. The events (signal) are aligned along the parabolic curves in Figure 11.10 (c) and (d) as expected.



(a) *(b)*

Figure 11.10. The longitudinal (p_z) versus transverse (p_t) four-momentum distributions of $\bar{\Lambda}$ (left) and Λ (right) candidates after final selection for non-resonant background.

11.3 The IdealIP Case

In the IdealIP case, perfect vacuum conditions exists inside the beam pipe. Thus the internal target is considered to be a *point-like* target. Until now, all PANDA simulation studies except in Ref. [179] have been carried out using the IdealIP case.

We can now calculate the FoM using the information in Table 11.4, the weights in Table 11.1 and Equation 10.1. This yields FoM as

$$FoM = 99 \tag{11.6}$$

This constitutes a best-case scenario that serves as a reference when studying the more realistic cases, *i.e.* the NormalIP and NormalIP+Cryo.

12. Hyperon Reconstruction using Extended Target Profiles

In this chapter, the results of benchmark studies for extended target cases will be presented as described in Section 3.5. The *IdealIP* case, discussed in Chapter 11, is used as a reference. Hence, we compare three different cases; one ideal and two realistic ones.

12.1 The NormalIP Case

The vacuum simulations are performed by the cluster-jet target group at the University of Münster [1], the resulting density profile of both target and residual gas is shown in Figure 12.1.

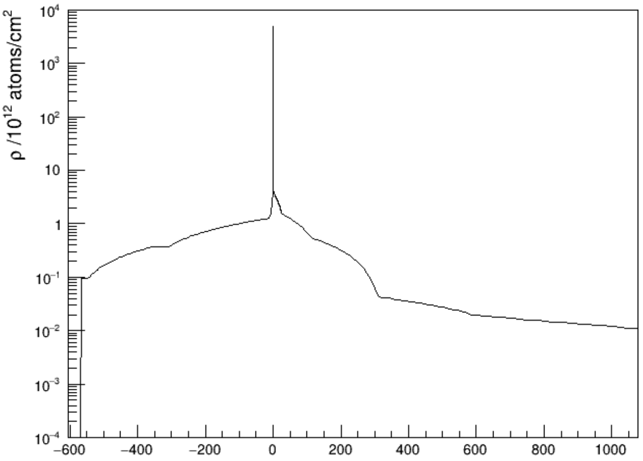
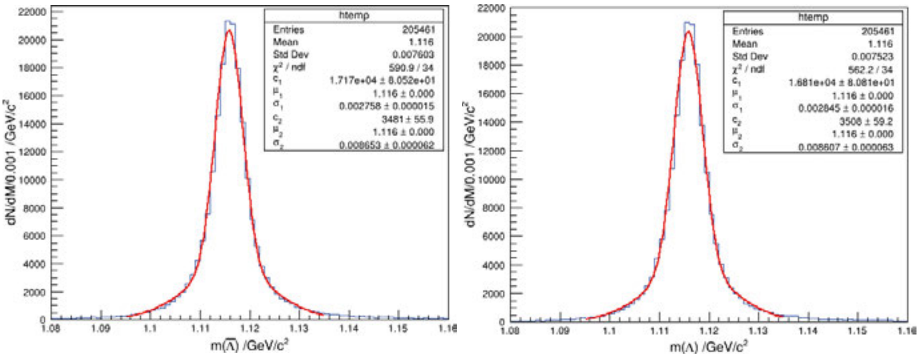


Figure 12.1. The density distribution of cluster-jet target as well as the residual gas along the beam pipe, the NormalIP case. The histogram has been divided by the bin width (due to variable bin width of data generated by the Münster Group [1]).

The density distribution defines an extended effective target region, *i.e.* the density is non-zero outside the design interaction point, available for the incoming beam. In this case, no measures are taken to remedy the residual gas, thus it represents one of the worst-case among the extended target cases. Due to the non-zero target density outside the design IP, it is expected that the requirement of displaced vertex positions will have a different impact on data that originate from interactions between the beam and the gas, compared to the IdealIP case (*c.f.* Chapter 11). This is because in case of an IdealIP, the location of the production vertex can be considered known with *infinite* precision while for a realistic target volume, one needs to reconstruct it by tracking the neutral Λ and $\bar{\Lambda}$ back to a common vertex. Alternatively, one can, as we have done here, still assume that the interaction takes place in the ideal point, although this is only approximately true: The proton-pion invariant mass is also sensitive to the point of origin. In the following, the invariant mass and decay vertex selection from the final selection will be discussed.

12.1.1 Invariant Mass Selection

First, a double Gaussian fit is applied to the invariant mass distributions of both Λ and $\bar{\Lambda}$ particles as shown in Figure 12.2.



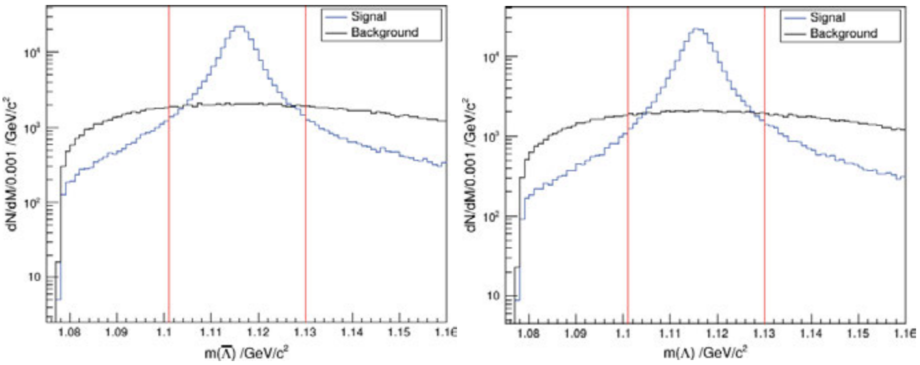
(a) *(b)*

Figure 12.2. The invariant mass resolution is obtained by double Gaussian fit on the invariant mass distributions (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ for the signal sample only, the NormalIP case.

Second, the mass resolution is extracted from the Gaussian fitting the largest contribution to the fit. The fits yields the mean values as $\mu_{m_{\text{fit}}}(\bar{p}\pi^+) = 1.116$ GeV/c^2, $\mu_{m_{fit}}(p\pi^-) = 1.116$ GeV/c^2 and the width as $\sigma_{m_{\text{fit}}}(\bar{p}\pi^+) = 2.758 \cdot 10^{-3}$ GeV/c^2 and $\sigma_{m_{\text{fit}}}(p\pi^-) = 2.845 \cdot 10^{-3}$ GeV/c^2 for the Gaussian with larger contribution to the fit. We note that the widths are approximately 3% larger than in the IdealIP case, which is a reflection of the distorted resolution. However, the difference is very small which is reassuring. The obtained mass values are consistent with the PDG value of the Λ: $m_{\text{PDG}}(\bar{\Lambda}) = 1115.683 \pm 0.006$ MeV/c^2. Third, a mass window is defined as five standard deviation around the PDG value of the Λ mass:

$$|m_{\text{fit}}(\bar{p}\pi^+) - m_{\text{PDG}}(\Lambda)| < 5 \cdot 2.758 \cdot 10^{-3} \text{ GeV/c}^2$$
$$|m_{\text{fit}}(p\pi^-) - m_{\text{PDG}}(\Lambda)| < 5 \cdot 2.845 \cdot 10^{-3} \text{ GeV/c}^2$$

The Figure 12.3 shows the invariant mass distribution of both Λ and $\bar{\Lambda}$ particles for signal (blue) as well as the non-resonant background (black) samples. The mass window defined above is marked with red vertical lines.



(a) *(b)*

Figure 12.3. The invariant mass distributions of (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ of signal (blue) and background (black) with cut value (red), scaled according Table 11.1.

12.1.2 The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm Criterion

The final criterion is is the requirement of displaced decay vertices along the $z-$axis. We require $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm to be fulfilled. The resulting distribution is shown in Figure 12.4 with the cut indicated as a red line.

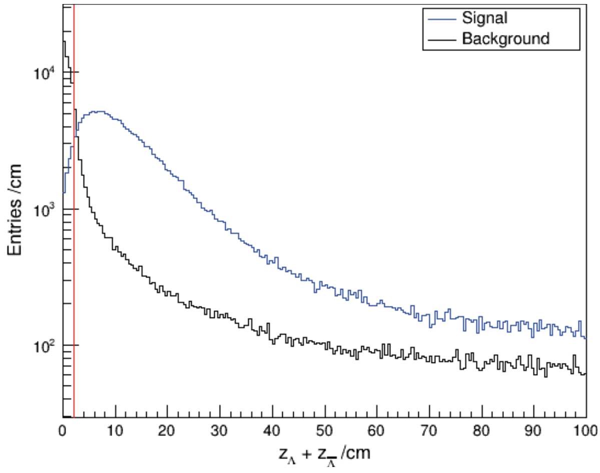


Figure 12.4. $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution of signal (blue) and non-resonant background (black) samples with the cut value (red), scaled according to Table 11.1.

The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution is expected to differ with respect to the IdealIP case due to the collisions occurring away from the nominal interaction point. We see from Figure 12.4 that indeed, the cut value needs to be adjusted in order to have the desired effect given the new realistic conditions. The cut value needs to be adjusted, which I will come back to in Subsection 12.1.4.

12.1.3 Results from the Final Selection

The final reconstruction efficiency is given in Table 12.1 for the signal as well as the non-resonant background samples.

Table 12.1. *The final reconstruction efficiency for signal as well as the non-resonant background samples, the NormalIP case.*

Channel	$p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$	$p\bar{p} \to \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
Pre-selection	260194	292007
$\chi^2 < 100$	205461	238505
Mass cut	175180	23722.0
$z_{\bar{\Lambda}} + z_{\Lambda} > 2$ cm	155239	5929
Efficiency %	15.52 ± 0.04	0.59 ± 0.01

The final efficiency of signal sample is 15.5% which is smaller than the IdealIP case which is 20.1%. However, the contribution of non-resonant background is slightly higher than the IdealIP case. One can see the impact of selection on $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution (highlighted in blue in the table), which is expected to influence the efficiencies due to decay vertex distribution along the z-axis.

The FoM for the NormalIP case according to Equation 10.1 is given below:

$$FoM = 44 \tag{12.1}$$

The value of FoM is 44 in this case which is almost half the value of FoM of IdealIP case. This significant reduction in FoM is mainly due to higher contribution of non-resonant background. However, one should keep in mind that a signal-to-background ratio of 44 means that the background contamination from non-resonant production is only 2.2%. As discussed earlier, optimizing the selection on $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution will further improve the FoM.

12.1.4 Adjusting the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ Criterion

The decay vertex cut can be slightly adjusted to the point of intersection between the signal and non-resonant background distributions. This point is located at $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2.5$ cm on distribution as shown in Figure 12.5. One can see that increasing the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ cut to a larger value inevitably leads to a reduced signal efficiency. However, it has the desired outcome that it reduces the background contribution even more. The Figure 12.6 shows FoM

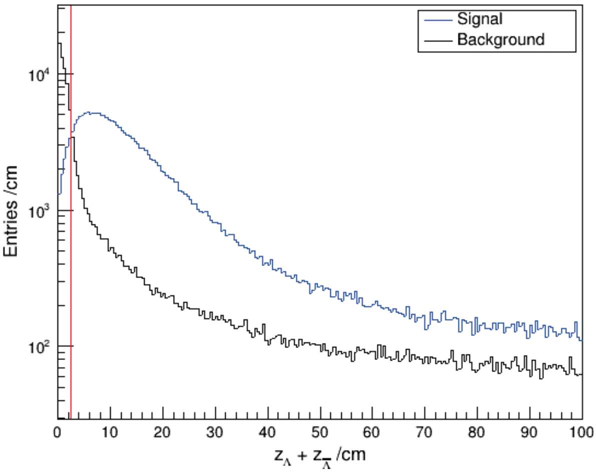
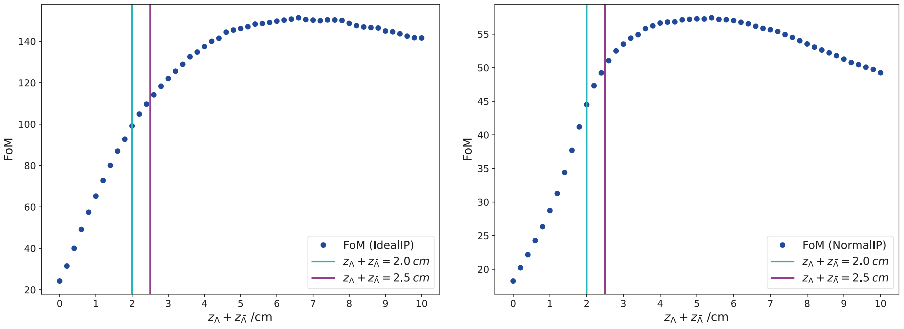


Figure 12.5. The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution of signal (blue) and non-resonant background (black) samples. The histograms are scaled according to Table 11.1. The cut is shown with a red line on the figure.

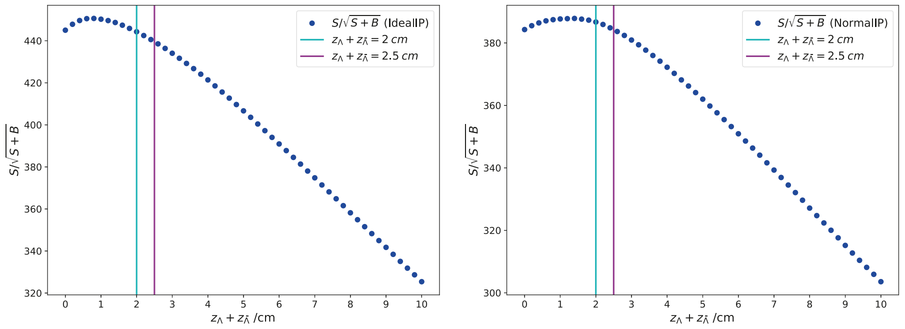
versus $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ for IdealIP and NormalIP cases. The cut values are marked vertical lines at 2 cm (cyan) and 2.5 cm (magenta).



(a) *(b)*

Figure 12.6. The FoM *v.s.* $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ for IdealIP and NormalIP cases. The old and new cut values are marked at 2 cm (cyan) and 2.5 cm (magenta), respectively.

Here, one should note that it is customary to optimize the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ cut as in Ref. [10] using the significance $S/\sqrt{S+B}$ as the Figure-of-Merit, and with B calculated from the DPM generator. Figure 12.7 shows the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ versus the signal-to-background ($S/\sqrt{S+B}$) for the IdealIP and the NormalIP cases and using the non-resonant $p\bar{p} \to \bar{p}\pi^+ p\pi^-$ as the background. If the cuts were optimized by the generic hadronic background, *e.g.* by using the DPM generator, then one might see the cuts on the optimum of the



(a)

(b)

Figure 12.7. The $S/\sqrt{S+B}$ *v.s.* $z_{fit}(\bar{\Lambda})+z_{fit}(\Lambda)$ for IdealIP and NormalIP cases. The old and new cut values are marked at 2 cm (cyan) and 2.5 cm (magenta), respectively.

$S/\sqrt{S+B}$ curves. However, this is not true as only the non-resonant background channel is used.

With the new cut value at 2.5 cm, the final efficiency is given in Table 12.2 for both signal and non-resonant background channels.

Table 12.2. *The final efficiency for signal as well as the non-resonant background samples for* $z_{fit}(\bar{\Lambda})+z_{fit}(\Lambda) > 2.5$ *cm.*

Channel	$p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^{+}p\pi^{-}$	$p\bar{p} \to \bar{p}\pi^{+}p\pi^{-}$
Generated	10^6	10^6
Pre-selection	260194	292007
$\chi^2 < 100$	205461	238505
Mass cut	175180	23722.0
$z_{\bar{\Lambda}} + z_{\Lambda} > 2.5$ cm	152670	5173
Efficiency %	15.27 ± 0.04	0.52 ± 0.01

One can see that the higher values of FoM corresponds to the lower efficiencies of signal and non-resonant background samples. There is trade-off between FoM and the efficiencies. The FoM with the new cut value is given below:

$$FoM = 50 \tag{12.2}$$

The FoM has risen to 50 after adjusting the cut which is approximately 14% increase in value of FoM compared to before optimization. This corresponds to a signal-to-background ratio of 2% for the non-resonant background.

12.2 The NormalIP+Cryo Case

This is the second extended target case with mere difference that a cryogenic pump, or cryopump, is employed to remove the residual gas from the beam pipe thus improving the efficiency of the benchmark study as discussed in previous sections. The density distribution of both target as well as residual gas for the NormalIP+Cryo case is shown in Figure 12.8.

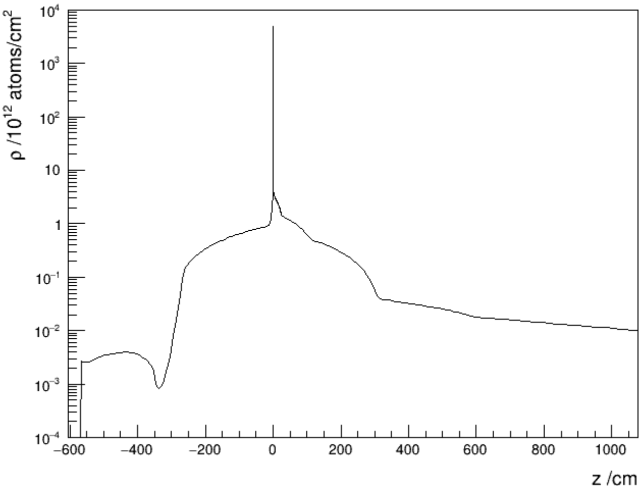


Figure 12.8. The target and residual gas density distribution along the beam axis. The histogram has been divided by the bin width (due variable bin width of data generated by the Münster Group [1]).

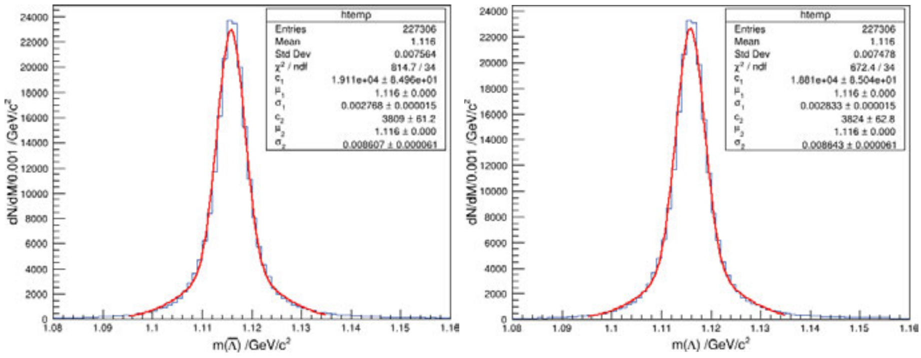
The cryopump is employed at 300 cm upstream of the beam pipe which visible by dip the density distribution at $z = -300$ cm. Due to gas extraction from the beam pipe, the density distribution on the left of the origin is substantially reduced. However, the distribution doesn't change much on the right side. The analysis for this extend target profile is given in the following sections.

12.2.1 Invariant Mass Selection

The double Gaussian fits to the invariant mass distributions of Λ and $\bar{\Lambda}$ particles. The fits yields the mean of $\mu_{m_{\text{fit}}}(\bar{p}\pi^+) = 1.116$ GeV/c^2, $\mu_{m_{\text{fit}}}(p\pi^-) = 1.116$ GeV/c^2 and the standard deviation of $\sigma_{m_{\text{fit}}}(\bar{p}\pi^+) = 2.768 \cdot 10^{-3}$ GeV/c^2 and $\sigma_{m_{\text{fit}}}(p\pi^-) = 2.833 \cdot 10^{-3}$ for the Gaussian with larger contribution to the fit. The double Gaussian fits are shown in Figure 12.9.

The mass window is defined as five standard deviation around $m_{\text{PDG}}(\Lambda)$ which, similarly to NormalIP, is about 3% larger compared to the IdealIP case:

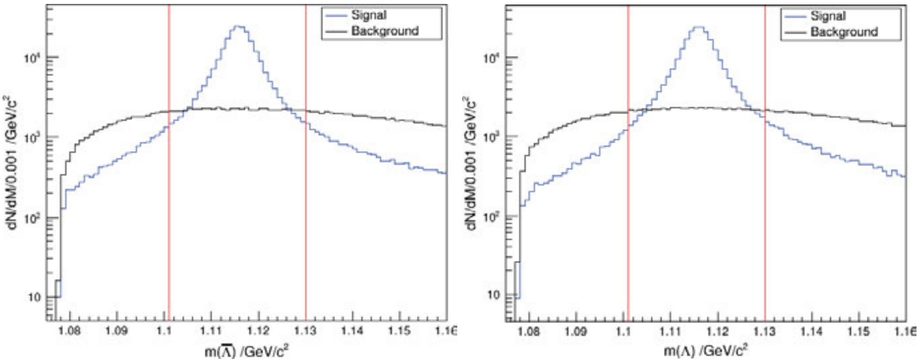
$$|m_{\text{fit}}(\bar{p}\pi^+) - m_{\text{PDG}}(\Lambda)| < 5 \cdot 2.768 \times 10^{-3} \text{ GeV/c}^2$$
$$|m_{\text{fit}}(p\pi^-) - m_{\text{PDG}}(\Lambda)| < 5 \cdot 2.833 \times 10^{-3} \text{ GeV/c}^2$$



(a) *(b)*

Figure 12.9. The invariant mass resolution is obtained by double Gaussian fit on the invariant mass distributions (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ for the signal sample.

The invariant mass distributions of Λ and $\bar{\Lambda}$ along with the mass windows are shown in Figure 12.10.



(a) *(b)*

Figure 12.10. The invariant mass distributions of (a) $m_{fit}(\bar{p}\pi^+)$ and (b) $m_{fit}(p\pi^-)$ of signal (blue) and non-resonant background (black). The histograms are scaled according Table 11.1 whereas the red lines represent the mass window.

The mass resolutions obtained in the NormalIP+Cryo case are not significantly different from the NormalIP or the IdealIP cases. Hence the corresponding mass windows also do not differ much from the these cases.

12.2.2 The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2$ cm Criterion

The decay vertex cut for the NormalIP+Cryo is shown in Figure 12.11 with the cut value is marked with the red line. The signal and non-resonant background samples are scaled according to Table 11.1. One can see that the background events are peaked around $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) = 0$ while the signal events around

$z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) \approx 9$ cm. As in the NormalIP case, where the cryopump is not used, the decay vertex cut can also be slightly shifted to the right of cut value, which we will come back to in Section 12.2.4.

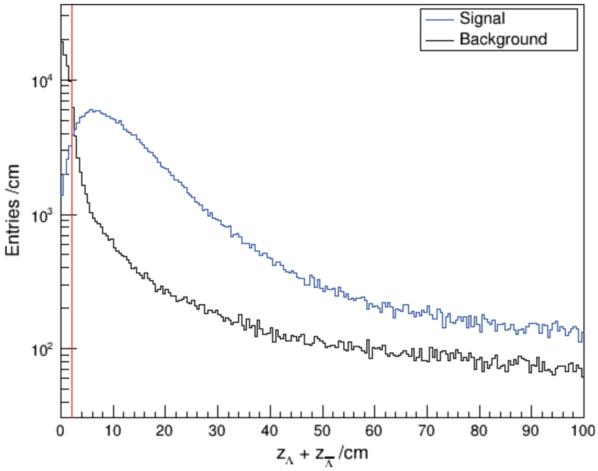


Figure 12.11. The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution of signal (blue) and non-resonant background (black) samples with cut value shown with a vertical red line. The histograms have been scaled according to Table 11.1.

Apparently, the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution is similar to the NormalIP case, as the cryopump has little effect on the residual gas away from the interaction point as shown in the extended target profiles for both cases in Figures 12.1 and 12.8.

12.2.3 Results from the Final Selection

After applying the selection criteria, step by step, the $p\bar{p} \rightarrow \bar{\Lambda}\Lambda$ reaction is reconstructed from the final state particles for both signal and non-resonant background samples. The final reconstruction efficiency is given in Table 12.3.

Table 12.3. *The final efficiency for signal as well as the non-resonant background samples.*

Channel	$p\bar{p} \rightarrow \bar{\Lambda}\Lambda \rightarrow \bar{p}\pi^+ p\pi^-$	$p\bar{p} \rightarrow \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
Pre-selection	287424	326858
$\chi^2 < 100$	227306	266872
Mass cut	193991	26611
$z_{\bar{\Lambda}} + z_\Lambda > 2$ cm	173597	6503
Efficiency %	17.36 ± 0.05	0.65 ± 0.01

The FoM for this case, using Equation 10.1, is given below:

$$FoM = 45 \tag{12.3}$$

We note that it is almost half of the FoM in the IdealIP case, while the difference is negligible when compared to NormalIP case.

12.2.4 Adjusting the $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ Criterion

As the FoM of NormalIP and NormalIP+Cryo are not very different from one another, one can conclude that $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ cut can be shifted from 2 cm to 2.5 cm. The new cut value is shown in Figure 12.12 at the intersection of signal and background distributions.

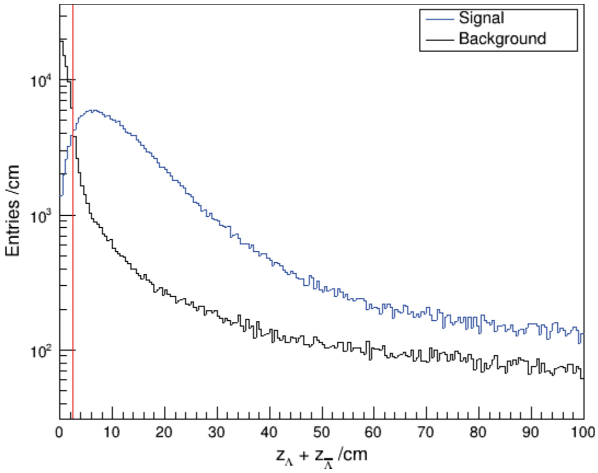


Figure 12.12. The $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda)$ distribution of signal (blue) and non-resonant background (black) samples. The histograms are scaled according to Table 11.1. The cut is shown with a red line on the figure.

The final efficiency after the new decay vertex cut defined as $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) >$ 2.5 cm is shown in Table 12.4.

Table 12.4. *The final efficiency for signal as well as the non-resonant background samples for $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2.5$ cm.*

Channel	$p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$	$p\bar{p} \to \bar{p}\pi^+ p\pi^-$
Generated	10^6	10^6
Pre-selection	287424	326858
$\chi^2 < 100$	227306	266872
Mass cut	193991	26611
$z_{\bar{\Lambda}} + z_{\Lambda} > 2.5$ cm	170604	5581
Efficiency %	17.06 ± 0.04	0.56 ± 0.01

As expected, the efficiencies are a little smaller than the values before adjusting the decay vertex cut. The new FoM is given below:

$$FoM = 52 \tag{12.4}$$

which implies a 16% increase in the FoM, this means that the signal-to-background ratio will be around 1.9%. One can see that the higher FoM means the lower efficiencies as compared to earlier case. However, the difference in FoM for the NormalIP and NormalIP+Cryo is still very small.

12.3 Summary of Results

In this section, the results will be summarized from the all cases, including the IdealIP. Results from both with and without adjusting the decay vertex cut are presented in Table 12.5. The summary of results with optimized decay vertex cut is given in Table 12.6.

Table 12.5. *Summary of final efficiencies and the FoM for various target types for signal as well as non-resonant background samples. The values corresponds to the* $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2.0$ *cm criterion.*

Target	ε_{sig} [%]	ε_{bkg} [%]	*FoM*
IdealIP	20.08 ± 0.05	0.34 ± 0.01	99
NormalIP	15.52 ± 0.04	0.59 ± 0.01	44
NormalIP+Cryo	17.36 ± 0.05	0.65 ± 0.01	45

Table 12.6. *Summary of final efficiencies and the FoM for various target types for signal as well as non-resonant background samples. The values corresponds to the* $z_{fit}(\bar{\Lambda}) + z_{fit}(\Lambda) > 2.5$ *cm criterion.*

Target	ε_{sig} [%]	ε_{bkg} [%]	*FoM*
NormalIP	15.27 ± 0.004	0.52 ± 0.01	50
NormalIP+Cryo	17.06 ± 0.004	0.56 ± 0.01	52

One can now make a comparison between the ideal and and the two realistic effective target profile scenarios. First, the FoM drops by almost 50% in the NormalIP configuration when compared to the IdealIP case. It is significant to the extent that one can not ignore the impact of residual gas while performing simulation studies which are by default IdealIP case. The lower FoM indicates that the background suppression is significantly impaired, but not to the extent that it would make studying this channel unfeasible. On the contrary: a signal-to-background ratio of 44-52 indicates a very clean sample, in the sense that the signal-to-background rate will be around 2%. However, the event selection algorithm can be further optimized in order to suppress background in the best possible way. Furthermore, the FoM is not very different within the extended target profiles *i.e.* it is merely a 4% rise in the FoM of NormalIP+Cryo compared to the NormalIP case. Overall, the background suppression is still very good in both cases. The cryopump does not have significant impact downstream of the beam-pipe. However, it is beneficial in other ways *e.g.* it reduces the residual gas around IP which means that there will be fewer beam losses due to interactions with the gas. This will improve the chances of keeping the luminosities large.

Part IV: Conclusions, Outlook, Summary and Acknowledgments

13. Conclusions

13.1 Part II: Realistic Track Reconstruction

By applying deep learning and geometric deep learning models, I have developed a complete track reconstruction method for the STT detector. The STT is considered one of the most crucial detectors in PANDA; see a discussion in the beginning of Part II. The task of reconstructing particle trajectory is formulated as an edge classification problem where one can leverage the ability of neural networks. The strategy is to first reconstruct muons trajectories in the STT by employing two different machine learning domains for edge classification. These domain are labeled as Euclidean where a DNN is used, and non-Euclidean where a GNN is used. For sake of convenience, I used graph-based terms for both, as one can think of Euclidean domain as a special case of the non-Euclidean one. So each STT event is treated as a graph with nodes and edges, where nodes are the hit positions and edges are the links between them. Hence particle trajectories are broken down into a list of true and false edges. A predicted event is a weighted graph where each edge has score. Next, clustering algorithm such as the DBSCAN algorithm filters out edges by applying a certain threshold on the edge score, providing us with track candidates.

My ultimate goal is to reconstruct particles with two distinct features: *(i)* particles with extremely low p_t and *(ii)* particles that originate away from the IP. Handling both of these issues is essential to study many hadron physics studies, in particular hyperons. I took a gradual approach to tackle both. First, I reconstructed muon trajectories with two different deep learning models. In the first case, a DNN is used for edge classification and then reconstructed tracks. The tracking efficiencies ($\varepsilon_{phys.}$, $\varepsilon_{tech.}$) are found to be smaller than 80%, which is not good enough for successful hadron studies. Therefore, I employed an interaction GNN (IGNN) for edge classification. Using a GNN has an advantage over a DNN, since it can process a full graph all at once, whereas DNN performs edge classification on each edge in isolation. Thanks to this feature of the IGNN, the tracking efficiencies increase by more than 10% units for every evaluation criteria (see Section 7.2 and Section 8.2). The results demonstrate that the IGNN has an advantage over the DNN in every respect. Therefore, it is chosen for further analysis.

Finally, final state particles from the $p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ are reconstructed using geometric deep learning. The same strategy as for the IGNN reconstruction with muons, is employed to reconstruct final state hadrons from this reaction. The tracking efficiencies are found to be $\geq 90\%$ for the evaluation criteria used in Section 9.3. Look at the low p_t particles; our method works as low as $p_t \sim 0.05$ GeV/c for pions and $p_t \sim 0.1$ GeV/c protons. I also investigated the ability of our method to reconstruct tracks away from the IP. I found that my method works well for the full range of radial vertex distances d_0 that I investigated, *i.e* at least up to 14 cm. This is an important result since we expect other hyperons such as Ξ^- and Ω^- decaying through intermediate Λ hyperons, will have Λ decay vertices mostly less than 15 cm in the radial direction from the IP [18]. Hence our approach using IGNN has proved promising for hadron physics studies. Slight modifications in our approach can produce even better results, as discussed under outlook (Chapter 14). Especially interesting is to see whether this can reduce the clone rates.

13.2 Part III: Hyperon Simulations with Realistic Target Profiles

The strategy adopted in this part of the thesis is to first thoroughly study the $p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$ reaction (*i.e.* Chapter 11). The approach was the same as in Ref. [10], which enabled a comparison between an old and a new version of the PandaRoot software. With the new version, the global reconstruction efficiency has significantly improved from $16.10 \pm 0.04\%$ to $20.1 \pm 0.05\%$. However, the efficiency for the non-resonant background has increased from $0.048 \pm 0.002\%$ to $0.344 \pm 0.006\%$. This matter is under investigation that what new features in PandaRoot have caused this change?

Until now, all physics analyses within PANDA have been studied under ideal target conditions, *i.e.* assuming the beam-target interaction to be confined to one point in space. In this work, we have also investigated effects of gas dissipation from the target into the beam pipe. The gas act as an effective target that is substantially extended in space, hence resulting in a large beam-target interaction volume. By studying the non-resonant $p\bar{p} \to \bar{p}\pi^+ p\pi^-$ in parallel to the $p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$, we learn how efficiently we can suppress background with ideal and realistic target conditions. This calls for a fine-tuning of the selection criteria, in particular the vertex position cut that has been adjusted in order to improve the signal-to-background ratio in this work.

It was found that the selection procedure, adopted for exclusive reconstruction of $p\bar{p} \to \bar{\Lambda}\Lambda \to \bar{p}\pi^+ p\pi^-$, was highly effective in suppressing the non-resonant background. This is also the case when we perform the simulations with a realistic target profile: although the signal-to-background ratio is reduced by about a factor of two, the background is still only around 2%. By tuning the selection criteria, this can be further improved.

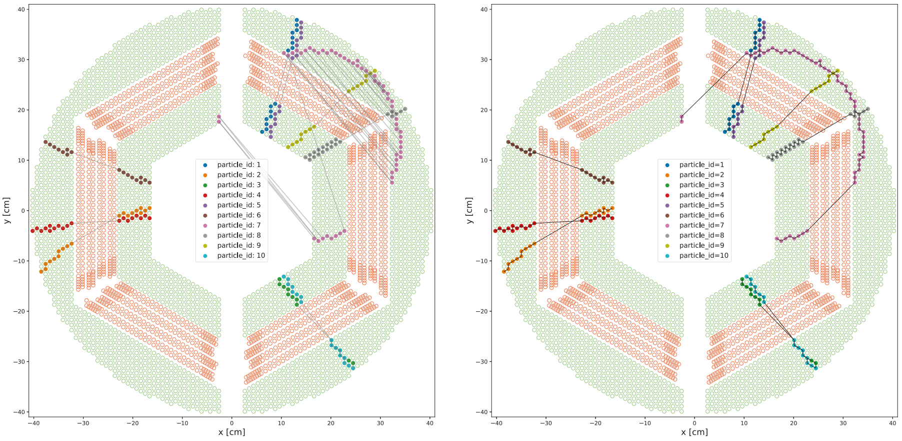
Two extended target scenarios were studied: one with the default pumping system, and one with an additional cryopump. The results do not differ much between these two scenarios; the main asset of using the cryopump is instead to preserve good beam conditions. In the near future, it is expected that vacuum conditions in the beam-pipe near the target will further improve, since extensive developments are ongoing in the cluster-jet target group at the University of Münster. It is expected that this will further improve the capability to suppress the background of the hyperon production channels.

14. Outlook

14.1 Part II: Realistic Track Reconstruction

After demonstrating the ability of machine learning in reconstructing final state particles that have secondary decay vertices in the STT, I will now discuss the next steps towards a more complete tracking framework for PANDA. The performance can be improved in several ways. First, adding more detectors such as the MVD and the GEM implies more data for the GNN, which should improve the performance. This is part of ongoing work on a heterogeneous setup where not only the data from detectors are heterogeneous, but also the deep learning pipeline uses heterogeneous neural networks. The work with this pipeline is not yet completed, hence results from this work will instead be presented in the future.

A second way to improve the performance is to implement a better ground truth heuristic designed specifically for low p_t tracks. Due to ambiguous ground truth for such tracks, it is suspected that true edges are labeled as fake ones. Hence a better ground truth will improve the tracking efficiencies. I have developed a heuristic method that can solve this problem. For example, Figure 14.1 shows the difference between the current and new methods to generate ground truth, especially for curly tracks.



(a) Current Ground Truth *(b)* Future Ground Truth

Figure 14.1. Current (left) and future (right) heuristic method to build ground truth for input graphs. The figures shows muon data from Chapter 8.

Our current method works poorly for low p_t tracks especially those that curl back towards the IP. To account for this problem, one needs to follow along the particle trajectory while generating ground truth information. This can be achieved in several ways. For example, the timing of hits can be used. However, in the STT, the time resolution is poor and as a result, one can not exploit this option. However, in many other, faster detectors such as the MVD, this could be viable. One can also use radial distance of the hits from the IP, but that fails for curly tracks. The third option is to use *order-of-occurrence* of hits in a particle trajectory. We have found that, for each particle, such an order exists in our data. However, one needs to thoroughly investigate if this order is indeed true for all events. This method is used to create a better ground truth for low p_t tracks, see the left panel in Figure 14.1.

Developing a better track formation algorithm is the fourth way to improve our performance. In this thesis, tracks are built using connected-components (CC) algorithm[1]. By design, the CC algorithm breaks intersecting tracks into multiple sub-tracks. One drawback of these algorithms is the high clone rate caused by low p_t intersecting tracks. To handle this situation, one might opt for a different method for this purpose. For example, a path-finding algorithm named Walkthrough is developed by the Exa.TrkX collaboration [155] as an alternative to connected components. At present, also this method has difficulties with intersecting tracks. The Walkthrough can however easily be adapted to account for handling intersecting tracks. Due to the combinatorial way of track building, this algorithm is slow. One way to speed up the track formation is to create a hybrid solution where the Walkthrough algorithm runs successively after the CC. First, run CC on the data with a low cut on the edge score allowing intersecting tracks to form single components. After that, one can run the Walkthrough on each component the CC generates. The successive operation of both algorithms is expected to increase our tracking efficiencies. This work is currently under development and may be tested in the future to see its effectiveness.

14.2 Part III: Hyperon Simulations with Realistic Target Profiles

In PANDA, the analysis with an extended target is the first physics study performed with the realistic target profiles *i.e.* including the residual gas. The target scenarios are under constant development, especially the NormalIP+Cryo. The study performed in this report will provide useful insight concerning the impact of different vacuum scenarios on the physics analysis. Furthermore, this study can serve as a template for other physics channels to understand

[1] CC $\sim$ DBSCAN and CCL

how the realistic conditions might affect the results of the corresponding simulation studies done in other groups.

In the analysis presented here, the interaction point was assumed to be at $(0,0,0)$ also in the case of realistic target profiles. This is of course only approximately true. An alternative approach is to study whether the primary interaction vertex can be reconstructed by the point of closest approach of the Λ and the $\bar{\Lambda}$ candidates. A better-known interaction point should yield a more powerful vertex displacement cut.

15. Summary in Swedish

PANDA är ett framtida experiment som håller på att byggas i Darmstadt i Tyskland. Experimentet har designats för detaljerade studier av hadroner - sammansatta partiklar som binds samman av stark växelverkan. De vanligaste och mest välkända hadronerna är protoner och neutroner, som består av lätta upp- och nedkvarkar. Anta att en partikel består av två lätta kvarkar och en tung och instabil särkvark. En sådan partikel kallas hyperon. Den lättaste av dessa är Λ-hyperonen – och de tyngre Σ, Ξ och Ω^-. Λ-hyperonen kan betraktas som tyngre syskon till protonen och neutronen eftersom de liknar varandra, men har något olika massa.

Denna avhandling är ett steg på vägen mot att uppnå realistiska simuleringar respektive dataanalys med PANDA. Under utvecklingsfasen av PANDA utförs olika studier, t.ex. hur hyperoner produceras och detekteras, under idealiska förhållanden. En fördel med dessa är att man kan fokusera på just det man vill undersöka utan att oroa sig för eventuella olägenheter som kan uppstå på grund av bristande perfektion i mjukvara eller apparatur. I praktiken måste man dock utföra dessa studier med data som producerats under de förhållanden som faktiskt gäller i det verkliga experimentet. Ett exempel på detta är processen att rekonstruera partiklarnas spår genom detektorn. Därför måste de experimentella förhållanden som man ”härmar” i simuleringarna ligga så nära det verkliga experimentet som det bara är möjligt.

I den här avhandlingen har jag fokuserat på två huvudsakliga ämnen. För det första har jag utvecklat ett förfarande baserat på maskininlärning som rekonstruerar partikelspår. För det andra har jag undersökt vilka effekter realistiska vakuumförhållanden har på fysiksimuleringarna. Låt oss dyka ner i båda dessa utmaningar, var för sig, och diskutera dem i detalj.

Processen att rekonstruera partikelbanor är ett av de kritiska stegen i kedjan för dataanalys från ett experiment av denna typ. När en kollision äger rum i en detektor flödar partiklarna ut från kollisionspunkten mellan strålen och strålmålet. Partiklarna passerar genom olika detektordelar där de växelverkar och därmed avger signaler i form av pulser - så kallade partikelträffar, eller bara träffar. Det finns två olika kategorier av partikelspår som kan uppstå från en kollision. För det första finns spår som börjar exakt där kollisionen inträffade, så kallade primära spår. För det andra finns spår som bildas när en primär, men instabil, partikel sönderfaller till dotterpartiklar, som i sin tur lämnar signaler i detektorn. Vi kallar den punkt där ett spår börjar för vertex. När man rekonstruerar spår grupperar man partikelträffar så att man kan följa partikelns bana genom detektorn. Från partikelbanan kan man sedan bestämma de kinematiska egenskaperna hos en laddad partikel, såsom dess position och rörelsemängd. Sekundära spår är en utmaning eftersom ursprungsvertex inte har något bestämt läge, utan varierar mellan olika reaktioner. Fysikstudier vid PANDA sker vid ganska låga energier jämfört med till exempel LHC vid CERN. Detta resulterar i lågenergispår från strålens och strålmålets kollisionspunkt, men också från sönderfallspunkten, och dessa spår är komplicerade att rekonstruera.

Det finns många algoritmer för att rekonstruera spåren. Det är dock inte alla som uppfyller de särskilda behoven för ett hadronfysikexperiment som PANDA. Vissa algoritmer är till exempel bra på att rekonstruera primära spår, men inte sekundära, medan det för andra kan vara tvärtom. Vissa presterar bättre när det finns relativt få spår, men presterar dåligt när man behandlar ett stort antal spår. De flesta har svårigheter att rekonstruera spiralformade spår. De särskilda egenskaperna hos PANDA-experimentet kräver således att man utforskar nya tekniker som kan uppfylla de speciella behov som uppstår till följd av experimentets utmanande egenskaper. Jag har utforskat en lösning inom området maskininlärning.

Den lösning jag har utvärderat närmare är en mjukvaruimplementation av den så kallade Straw Tube Tracker (STT)-detektorn, eller strådetektorn, och de träffar som bildas inuti den. För att kombinera dessa träffar till spår har jag använt mig av maskininlärningstekniker. Här betraktas spåren som grafer med noder och kanter. Träffpositionerna utgör noder, medan kanter binder samman träffpositioner. En länk anses vara sann om två noder bildar ett spår och falsk om de inte gör det. Kanterna skapas med hjälp av ett kombinatoriskt tillvägagångssätt som omfattar alla möjliga tolkningar. Nu är huvuduppgiften att avgöra vilka kanter som är sanna och vilka som är falska. För detta ändamål kan vi antingen använda några komplicerade matematiska operationer eller maskininlärning.

I min avhandling använder jag neurala nätverk för att förutsäga vilka kantkombinationer som är sanna respektive falska. Eftersom det är omöjligt att exakt avgöra detta, kommer det neurala nätverket att tilldela varje kant en viss sannolikhet. Där slutar maskininlärningsalgoritmen sitt arbete och nästa steg tar vid. Jag har använt två olika nätverk. Det första kan behandla alla noder och kanter samtidigt och kallas Graph Neural Network (GNN). Det andra, som endast behandlar med kanterna var för sig, kallas Dense Neural Network (DNN). När kanterna förutsägs med en viss sannolikhet kan man sedan använda en klustringsalgoritm för att gruppera identifierade kanter till spår, som i sin tur motsvarar partiklar.

För att se hur väl de båda modellerna kan fungera har jag jämfört spårningseffektiviteten för de båda nätverken. Spårningseffektiviteten är förhållandet mellan antalet rekonstruerade spår och det totala antalet genererade spår. I simuleringar med elektromagnetiskt växelverkande myonpartiklar fann jag att GNN ger en betydligt bättre spårningseffektivitet jämfört med DNN. Detta beror på att GNN bearbetar en hel graf på en gång och därför kan lära sig förhållandet mellan noderna bättre än DNN, som bara ser ett segment av helheten i taget. Motiverad av GNN:s höga effektivitet ville jag också undersöka hur GNN fungerar för starkt växelverkande hadroner, i synnerhet hyperoner. Jag använde därför samma analysförfarande för hyperoner av typen Λ som produceras i proton-antiproton-annihilationer och som sönderfaller till pioner och protoner. Eftersom Λ-hyperonerna är neutrala är det bara deras sönderfallsprodukter, det vill säga protonerna och pionerna, som syns i detektorn. I detta fall uppstår två utmaningar: *(i)* spåren, särskilt pionernas banor, blir starkt krökta i PANDA-detektorns magnetfält, *(ii)* sönderfallsprodukternas ursprungsvertex befinner sig flera centimeter eller till och med decimeter från kollisionspunkten. Dessa båda problem har tidigare varit stötestenar för många algoritmer. I mitt fall fann jag dock att genom att rekonstruera protoner och pioner med hjälp av GNN så är spårningseffektiviteten nästan lika hög som i fallet med myonerna. Viktigast av allt är GNNs förmåga att rekonstruera lågenergipartiklar. Därför är mitt förfarande en lovande väg framåt för både myoner och hadroner. Metoden kan anses vara en lämplig kandidat för att rekonstruera många hadroniska reaktioner där andra går bet.

I den sista delen av min avhandling undersökte jag vilka effekter restgas har på hyperonrekonstruktionen. Simuleringsstudier utförs främst genom att man antar att strålmålet är *punktformigt*. Detta innebär att kollisionspunkten mellan stråle och strålmål kan anses vara oändligt väl lokaliserad, alltså idealisk. I detta fall är kinematiken hos partiklarna som bildats relativt enkel att rekonstruera med god noggrannhet. I det verkliga experimentet kommer vi dock att ha restgas som bildats när stråle och mål växelverkar och som läcker ut i strålröret. Detta måste beaktas i fysiksimuleringar. Genom att simulera effekterna av restgasen inuti strålröret får vi en växelverkansvolym mellan stråle och strålmål i stället för en växelverkanspunkt. Detta förvränger de rekonstruerade partikelegenskaperna.

För att undersöka effekterna av restgasen har jag simulerat Λ hyperonproduktion med ett scenario för idealiskt vakuum (d.v.s. en växelverkanspunkt) och två realistiska scenarier (d.v.s. med växelverkansvolymer). Varför kan man inte få bort den kvarvarande gasen? Ett av scenarierna innehåller faktiskt ett sådant försök: en kryopump som ska avlägsna restgasen. Kryopumpen är dock inte perfekt och därför kommer vi fortfarande att ha kvar en del restgas i strålröret. Sammantaget finns det alltså tre scenarier: idealiskt mål (IdealIP), en utökad målvolym (NormalIP) och en minskad målvolym tack vare kryopumpen (NormalIP+Cryo). Jag har jämfört resultaten från alla tre scenarierna med ett enhetligt analysförfarande.

För att angripa detta problem analyserar jag återigen Λ-hyperoner som produceras i annihilationer mellan antiprotonstrålen och vätestrålmålet. Den här gången tar jag också hänsyn till att hyperonerna sönderfaller till protoner och pioner, som jag redan har nämnt. Vår reaktion sker i två steg: det inledande tillståndet, där proton och antiproton annihilerar varandra och bildar hyperoner, och det slutliga tillståndet, där protoner och pioner bildas i hyperonsönderfallen. Nu behöver vi bara identifiera och mäta protonerna och pionerna och sedan rekonstruera vår reaktions initialtillstånd. Vi upprepar vårt förfarande för alla tre ovan nämnda scenarier och definierar ett meritvärde (FoM) som kvantifierar hur restgasen påverkar vår förmåga att identifiera den intressanta reaktionen samtidigt som vi avlägsnar bakgrunden. FoM för IdealIP, NormalIP och NormalIP+Cryo beräknas till 99, 44 respektive 45. FoM för de realistiska fallen sjunker alltså med en faktor på hälften jämfört med idealfallet. Vi noterar också att den totala rekonstruktionseffektiviteten för hyperoner sjunker från 20,1% till 15,5% respektive 17,34%. Även om restgasen har en inverkan på effektiviteten är det inte en drastisk effekt. Därför är det fortfarande möjligt att studera hyperoner i PANDA, även under realistiska vakuumförhållanden.

16. Acknowledgments

It's the early hours of a Wednesday morning in March as I write these acknowledgments, bleary-eyed and still buzzing from the adrenaline of the final stretch. The past few weeks have been a blur of late nights, endless revisions, and countless cups of coffee. But I wouldn't have made it to this point without the unwavering support of my family, friends, and colleagues, who have cheered me every step of the way.

First, I would like to express my appreciation to my main supervisor Karin Schönning, whose unwavering support, and mentorship have been a cornerstone of my success. I still remember how she went out of the way to inform me about my selection as a Ph.D., and patiently answered every email from the Migrationverket for nearly a year before I arrived in Uppsala. I remember all the amazing opportunities she tried to hook me up with—hats off to Jacek, who answered many inquiries from the Migrationverket along with Karin. I want to thank Michael Papenbrock and Carl Nettleblad, my co-supervisors, and Stefan Leupold, whose expertise and guidance have been valuable. Your encouraging words and detailed feedback have been essential to me.

A special thank you to all my fellow Ph.D. students at Uppsala university: Walter, Elisabetta, Jenny, Gabriela, Jana, Malin, Ghandharva, and An Di. I probably need to remember many names. Also, to those from the PANDA collaboration, especially Waleed, with whom I enjoyed my machine learning discussion at every PANDA collaboration meeting. Ralf and Tobias, whose technical assistance is beyond valuable; getting through my work with them was more straightforward. I would also like to thank my colleagues from the HEP.TrkX and Exa.TrkX groups from Lawrence Berkeley Laboratory, Xiangyang Ju, and Daniel Murnane, with whom I have countless discussions regarding geometric deep learning. With your help, it was easier to complete this work.

Thank you to my parents, Shamim and Akram, for your endless support. You have always stood behind me. I also thank my younger sisters Ambreen, Noreen, and Ammara and brothers Zeeshan, Mohsan, and Zahid, who has been a constant source of love and support throughout this journey. Their unwavering belief in me and continuous encouragement have kept me motivated and focused.

Finally, special thanks to my 2-year-old munchkin, Ibrahim, who has been a constant source of joy and inspiration, even on the most challenging days. His contagious laughter, boundless energy, and innocent curiosity remind me to stay present and have kept me going through long hours of research and writing. This thesis is dedicated to him, with the hope that one day he will look back on this moment with pride and grows up to pursue his passions with the same level of determination and tenacity that I have learned from this experience! I will remind you how you plucked my plants, took my pencils, and took over my computing desk.

Thanks to my wife, Rabia, for constantly listening to my rant and talking things out, for cracking jokes when things became too serious, and for the efforts she has made for me to pursue a Ph.D.

Appendices

A. Hyperon Analysis

A.1 Efficiency and Uncertainties

The global reconstruction efficiency is defined as the following:

$$\varepsilon = \frac{n_{\text{rec}}}{n_{\text{gen}}}$$

where n_{rec} are the reconstructed events and n_{gen} are total generated events. It is assumed that the number of signal events are Poisson distributed, so uncertainties on number of total generated events and reconstructed events are given as:

$$\sigma(n_{\text{rec}}) = \sqrt{n_{\text{rec}}}$$
$$\sigma(n_{\text{gen}}) = \sqrt{n_{\text{gen}}}$$

The uncertainty of efficiency is given by the following error propagation equation:

$$\sigma_\varepsilon = \sqrt{\left(\frac{\partial \varepsilon}{\partial n_{\text{rec}}}\right)^2 \sigma(n_{\text{rec}})^2 + \left(\frac{\partial \varepsilon}{\partial n_{\text{gen}}}\right)^2 \sigma(n_{\text{gen}})^2}$$
$$\sigma_\varepsilon = \sqrt{\frac{n_{\text{rec}}}{n_{\text{gen}}^2} + \frac{n_{\text{rec}}^2}{n_{\text{gen}}^3}}$$

A.2 Software Versions

For performing hyperon analysis, the following software versions are used:

- FairSoft: `nov20p1`
- FairRoot: `v18.6.3`
- PandaRoot: `v12.0.3`

B. Deep Learning

The superscript "(i)" will denote the i^{th} training example while superscript "$[l]$" will denote the l^{th} layer.

B.1 Notations

Neural network parameters are divided into three categories: *(i) Network Architecture*, *(ii) Network IO*, and *(iii) Network State*. The first two categories are collectively called *hyperparameters i.e.* there is no standard way to chose them. However, to make a sensible choice, one needs to perform *hyperparameter tuning*. The last set of parameters defines the state of a network, whose values are learned and improved over time during the network training. The indices and dimensions of these parameters are listed as follows:

Parameters – Network Architecture

- n_x: input size
- n_y: output size (or number of classes)
- $n_h^{[l]}$: number of hidden units of the l^{th} layer, $n_h^{[0]} \equiv n_x$ and $n_h^{[L]} \equiv n_y$
- L: number of layers in the network

Parameters – Network IO

- m: number of examples in the dataset
- $X \in \mathbb{R}^{n_x \times m}$ is the input matrix, $x^{(i)} \in \mathbb{R}^{n_x}$ is the i^{th} input example represented as a column vector
- $Y \in \mathbb{R}^{n_y \times m}$ is the label matrix, $y^{(i)} \in \mathbb{R}^{n_y}$ is the output label for the i^{th} example
- $a^{[l]} \in \mathbb{R}^{n_n^{[l]}}$ is output vector of all nodes in l^{th} layer, $a_j^{[l]} = g^{[l]}(\sum_k w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]})$ is the output of each node in a layer where $g^{[l]}$ is the activation function and $W^{[l]}, b^{[l]}$ are state parameters attached to each node
- $\hat{y} \in \mathbb{R}^{n_y}$ is the predicted output vector. It can also be denoted a $a^{[L]}$ where L is the number of layers in the network.

The state parameters are the weight (W) and biases (b) attached to each connection of a network node. There dimensions are given as follows:

Parameters – Network State

- $W^{[l]} \in \mathbb{R}^{n^{[\text{next}]} \times n^{[\text{previous}]}}$ is the weight matrix, where $n^{[\text{next}]}$ and $n^{[\text{previous}]}$ are number of nodes in the next and previous layers, respectively
- $b^{[l]} \in \mathbb{R}^{n^{[\text{next}]}}$ is the bias vector in the l^{th} layer

State parameters, W and b, are collectively refered to as $\theta \equiv W, b$.

B.2 Data Vectorization

Every example in a dataset is represented in the form vectors as input vector $x := (x_1, x_2, ..., x_{n_x})^T$, output vector of hidden nodes in layer l as $a^{[l]} := (a_1^{[l]}, a_2^{[l]}, ..., a_{n_h^{[l]}}^{[l]})^T$ and output vector $y := (y_1, y_2, ..., y_{n_y})^T$.

For simplicity, the training index (i) is stripped off from both weight and bias vectors: so $W^{[l](i)} \sim W^{[l]}$ and $b^{[l](i)} \sim b^{[l]}$.

$$W_{jk}^{[l]} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{j1} & \cdots & \cdots & w_{jk} \end{bmatrix}, \qquad b_j^{[l]} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_y} \end{bmatrix}$$

where the dimensions of $W^{[l]}$ are denoted by $j = n_h^{[l]}$ and $k = n_h^{[l-1]}$.

Each example (i) in a dataset can be represented as vectors $x^{(i)}, a^{[l](i)}$ and $y^{(i)}$:

$$x^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_{n_x}^{(i)} \end{bmatrix}, \quad a^{[l](i)} = \begin{bmatrix} a_1^{[l](i)} \\ a_2^{[l](i)} \\ \vdots \\ a_{n_h}^{[l](i)} \end{bmatrix}, \quad y^{(i)} = \begin{bmatrix} y_1^{(i)} \\ y_2^{(i)} \\ \vdots \\ y_{n_y}^{(i)} \end{bmatrix}$$

The full dataset can be represented as tabular form by matrices $X, A^{[l]}$ and Y:

$$X = \begin{bmatrix} | & | & | \\ x^{(1)} & \cdots & x^{(m)} \\ | & | & | \end{bmatrix}, \quad A^{[l]} = \begin{bmatrix} | & | & | \\ a^{[l](1)} & \cdots & a^{[l](m)} \\ | & | & | \end{bmatrix}, \quad Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

The *forward propagation* equation in *matrix form* are given as follows:

$$\begin{aligned} Z^{[l]} &= W^{[l]} A^{[l-1]} + b^{[l]} \\ A^{[l]} &= g^{[l]}(Z^{[l]}) \\ \hat{Y} &= A^{[L]} \end{aligned} \tag{B.1}$$

where input $X \equiv A^{[l=0]}$ for null layer and $\hat{Y} = A^{[l=L]}$ output of final layer, and it is an estimate of the target value Y.

B.3 Speeding up Network Training

To speed up network training, several techniques are available: (i) weight initialization, (ii) input normalization, and (iii) activation normalization. These techniques are explained briefly in the following.

B.3.1 Parameter Initialization

It is crucial to consider the initial or starting values of network parameters $w_{jk}^{[l]}$ and $b_j^{[l]}$ which are often randomly or zero initialized. In many cases, such initialization may hinder the convergence of neural network. Initialing these parameters with reasonable values will help a neural network converge faster to find the global minimum.

There are many suggestions found in the literature to initialize these parameters which depends upon the type of activation function used. If the activation function is Tanh or Sigmoid then Xavier initialization is recommended [124].

$$W_{Xavier}^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{1}{n^{[l-1]}}\right) \tag{B.2}$$

$$b_{Xavier}^{[l]} = 0 \tag{B.3}$$

All the weights in a layer l are initialize randomly from normal distribution with $\mu = 0$, and $\sigma^2 = \frac{1}{n[l-1]}$ where $n[l-1]$ are number of neurons in layer $l-1$. Another variant of Xavier initialization is where the variance is the harmonic mean of $1/n^{[l-1]}$ and $1/n^{[l-1]}$.

$$W_{Xavier}^{[l]} \sim \mathcal{N}\left(\mu = 0, \sigma^2 = \frac{2}{n^{[l-1]} + n^{[l]}}\right) \tag{B.4}$$

$$b_{Xavier}^{[l]} = 0 \tag{B.5}$$

If ReLU is used as activation function, for example, a common initialization is He initialization [125] in which the weights are initialized by multiplying variance of the Xavier initialization by a factor of 2.

$$W_{He}^{[l]} \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{2}{n^{[l-1]}}) \tag{B.6}$$

$$b_{He}^{[l]} = 0 \tag{B.7}$$

The discussion on justification on why the Xavier and He initialization is better than the random initialization can be found in Ref. [180].

B.3.2 Input Normalization

It is a common practice to preprocess data before training a network. Preprocessing includes normalization and standardization of features. Which means all variables transform to a common scale. This important if the features are different from each other. In normalization, features are often rescaled into a range $[0,1]$ or $[-1,1]$ (zero-centered). In case of standardization, the features are rescaled to a mean of 0 and a standard deviation of 1.

$$x_{norm.} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{B.8}$$

$$x_{stand.} = \frac{x - \mu}{\sigma} \sim \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{B.9}$$

where μ is the mean, σ is the standard deviation of the input variable x. The ε is used to avoid dividing by zero. Normalizing the input using standard normal distribution is often called *standardization* of data. One should be careful in scaling features, as both methods mentioned above of rescaling features are prone to outliers.

B.3.3 Activation Normalization

In activation normalization, either Batch Normalization (BatchNorm) [181] or Layer Normalization (LayerNorm) [182] are used.

In BatchNorm, first, the activations of a batch of inputs are normalized for each feature map by calculating the mean and standard deviation of the activations. Second, the activations are normalized using these statistics. The BatchNorm helps to reduce the effect of internal covariate shift, that occur due to the change in the distribution of inputs to a layer during training. In LayerNorm, however, inputs are normalized by adjusting and scaling the activations of all inputs for each feature map. It first calculates the mean and standard deviation of the activations across all inputs in a feature map, then normalizes the activations based on these statistics.

References

[1] A. Khoukaz, *Residual Gas from PANDA Targets*, Talk at PANDA Collaboration Meeting (2021)

[2] PANDA Collaboration, *Physics Performance Report for PANDA: Strong Interaction Studies with Antiprotons*, PANDA Physics Book (2009), arXiv:0903.3905 [hep-ex]

[3] M. Srednicki, *Quantum Field Theory*, Cambridge University Press (2007)

[4] MissMJ and Cush, *Standard Model of Elementary Particles*, Wikimedia Images: SMPP, Accessed: 2022-10-1

[5] P.A. Zyla et al., *Particle Data Group*, Prog. Theor. Exp. Phys. **2020**, 083C01 (2020) and (2021) update

[6] M. Gell-Mann, *The Eightfold Way: a Theory of Strong Interaction Symmetry* (1961), doi: 10.2172/4008239

[7] Y. Ne'eman, *Derivation of Strong Interactions From a Gauge Invariance*, Nuclear Physics, **26**-2, 222–229 (1961), doi: 10.1016/0029-5582(61)90134-1

[8] E2m and Stannered, *Meson Nonet and Baryon Octet*, Wikimedia images: Nonet and Octet, Accessed: 2022-10-1

[9] E. Thomé, *Multi Strange and Charmed Antihyperon-Hyperon Physics for PANDA*, PhD-Thesis, Uppsala Universitet, Sweden (2012)

[10] W. Ikegami Andersson, *Exploring the Merits and Challenges of Hyperon Physics: with PANDA at FAIR*, PhD thesis, Uppsala Universitet, Uppsala (2020)

[11] G. Barucca et al., *PANDA Phase One*, Eur. Phys. J. A, **57**-184 (2021), doi: 10.1140/epja/s10050-021-00475-y

[12] L. Fabbietti and L. Tolos, *Strangeness in Nuclei and Neutron Stars*, Prog. Part. Nucl. Phys. **112**, 103770 (2020), doi: 10.1016/j.ppnp.2020.103770

[13] L. Canetti et al. *Matter and Antimatter in the Universe*, New J. Phys. **14**, 095012 (2012)

[14] T. Johansson, *Antihyperon-hyperon production in antiproton-proton collisions*, Proceedings of 8th Int. Conf. on Low Energy Antiproton Physics 95 (2003)

[15] E. Klempt et al. *Antinucleon - Nucleon Interaction at Low Energy: Scattering and Protonium*, Physics Reports **368** 119-316 (2002).

[16] P. D. Barnes et al. *Observables in High Statistics Measurements of the Reaction $\bar{p}p \to \bar{\Lambda}\Lambda$*, Phys. Rev. C, **54**, 1877–1886 (1996)

[17] S. Grape, *Studies of PWO Crystals and Simulations of the $\bar{p}p \to \bar{\Lambda}\Lambda$, $\bar{\Lambda}\Xi^0$ Reactions for the PANDA Experiment*, PhD Thesis, Uppsala University, Uppsala (2009)

[18] J. Regina, *Time for Hyperons : Development of Software Tools for Reconstructing Hyperons at PANDA and HADES*, Doctoral Thesis, Uppsala University, Uppsala (2021)

[19] W. Esmail, *Deep Learning For Track Finding and the Reconstruction of Excited Hyperons n Proton Induced Reactions*, Doctoral Thesis, Ruhr-Universität Bochum, Bochum (2021)

[20] H. Gutbrod, *FAIR Baseline Technical Report - Volume 1 Executive Summary*, GSI, Darmstadt, (2006)

[21] H. Gutbrod, *FAIR Baseline Technical Report - Volume 2 Accelerator and Scientific Infrastructure*, GSI, Darmstadt, (2006)

[22] Heuer et al., *First Science and Staging Review of the FAIR Project* (2022)

[23] PANDA Collaboration, *Strong Interaction Studies with Antiprotons*, Physics Performance Report for PANDA, (2009), arXiv:0903.3905 [hep-ex]

[24] C.F. Perdrisat and V. Punjabi and M. Vanderhaeghen, *Nucleon Electromagnetic Form Factors*, Prog.Part.Nucl.Phys., **59**, 694–764 (2007), doi: 10.1016/j.ppnp.2007.05.001

[25] B. Singh et al., *Feasibility studies of time-like proton electromagnetic form factors at PANDA at FAIR*, Eur. Phys. J. A **52**, 325 (2016), doi: 10.1140/epja/i2016-16325-5

[26] PANDA Collaboration, *Feasibility studies for the measurement of time-like proton electromagnetic form factors from $\bar{p}p \to \mu^+\mu^-$ at PANDA at FAIR*, arXiv:2006.16363 [hep-ex]

[27] G. Barucca et al. *The Potential of Λ and Ξ^- Studies with PANDA at FAIR*, Eur. Phys. J. A **57**, 154 (2021). doi: 10.1140/epja/s10050-021-00386-y

[28] M. Ablikim et al. (BESIII Collaboration), *Complete Measurement of the Λ Electromagnetic Form Factors*, Phys. Rev. Lett., 123, 122003 (2019), doi: 10.1103/PhysRevLett.123.122003

[29] G. Barucca et al., *Study of excited Ξ baryons with the PANDA detector*, Eur. Phys. J. A **57**, 149 (2021). doi: 10.1140/epja/s10050-021-00444-5

[30] N. Brambilla et al., *The XYZ States: Experimental and Theoretical Status and Perspectives*, Physics Reports, **873**, 1–154 (2020), doi: 10.1016/j.physrep.2020.05.001

[31] S. K. Choi et al. (Belle Collaboration), *Observation of a Narrow Charmoniumlike State in Exclusive $B^\pm \to K^\pm\pi^+\pi^- J/\psi$ Decays*, Phys. Rev. Lett, **91**, 262001 (2003), doi: 10.1103/PhysRevLett.91.262001

[32] G. Barucca et al. (PANDA Collaboration), *Precision Resonance Energy Scans with the PANDA Experiment at FAIR*, Eur. Phys. J. A **55**, 42 (2019) doi: 10.1140/epja/i2019-12718-2

[33] B. Sing et al. (PANDA Collaboration), *Study of Doubly Strange Systems using Stored Antiprotons*, Nuclear Physics A, **954**, 323–340 (2016), doi: 10.1016/j.nuclphysa.2016.05.014

[34] PANDA Collaboration, *Technical Design Report for the PANDA Solenoid and Dipole Spectrometer Magnets* (2009), arXiv:0907.0169 [physics.ins-det]

[35] PANDA Collaboration, *Technical Design Report for the: PANDA Micro Vertex Detector* (2012), arXiv:1207.6581 [physics.ins-det]

[36] F. Sauli, *GEM: A New Concept for Electron Amplification in Gas Detectors*, Nuclear Instruments and Methods in Physics Research Section A, Volume **386**, 531–534 (1997), doi: 10.1016/S0168-9002(96)01172-2

[37] Deutsches Elektronen-Synchrotron (DESY), *Gas Electron Multipliers (GEMs)*,https://flc.desy.de/tpc/basics/gem/index_eng.html, [Online; Accessed

March-15-2023]

[38] B. Voss, R. Karabowicz, *GEM Tracker*, https://panda.gsi.de/article/gem-tracker, [Online; Accessed 19-March-2023]

[39] The PANDA Collaboration, *Technical Design Report for the PANDA (AntiProton Annihilations at Darmstadt) Straw Tube Tracker*, Eur. Phys. J. A, **49**, 25 (2013)

[40] D. Steinschaden, *Optimization Studies and Performance Simulations for the Time-of-Flight System of PANDA*, PhD Thesis, Technische Universität Wien, Fakultät für Physik, Vienna (2018)

[41] PANDA Collaboration, *Technical Design Report for the PANDA Barrel Time-of-Flight* (2018)

[42] PANDA Collaboration, *Technical Design Report for the PANDA Barrel DIRC Detector* (2019), arXiv:1710.00684 [physics.ins-det]

[43] PANDA Collaboration, *Technical Design Report for the PANDA Endcap Disc DIRC* (2019), arXiv:1912.12638 [physics.ins-det]

[44] PANDA Collaboration, *Technical Design Report for the PANDA Muon System* (2021)

[45] PANDA Collaboration, *Technical Design Report for the PANDA Electromagnetic Calorimeter (EMC)* (2008), arXiv:0810.1216 [physics.ins-det]

[46] B. Hommels, *The LHCb Outer Tracker*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, **462**, 278–284 (2001), doi: 10.1016/S0168-9002(01)00125-5

[47] PANDA Collaboration, *Technical Design Report for the PANDA Forward Tracker* (2012)

[48] PANDA Collaboration, *Technical Design Report for the PANDA Forward Time of Flight* (2018)

[49] PANDA Collaboration, *Technical Design Report for the PANDA Forward Spectrometer Calorimeter* (2017), arXiv:1704.02713 [physics.ins-det]

[50] PANDA Collaboration, *Technical Design Report for the PANDA Luminosity Detector* (2015)

[51] PANDA Collaboration, *Technical Design Report for the PANDA Internal Targets* (2012)

[52] A. Akram, *Towards a Realistic Hyperon Reconstruction with PANDA at FAIR*, Licentiate Thesis, Uppsala University, Uppsala (2021)

[53] M. Al-Turany et al. *The FairRoot Framework*, J. Phys. Conf. Ser. **396**, 022001 (2012)

[54] S. Spataro et al. *The PandaRoot Framework for Simulation, Reconstruction and Analysis*, J. Phys. Conf. Ser. **331**, 032031 (2011)

[55] R. Brun and F. Rademakers, *ROOT: An Object Oriented Data Analysis Framework*, Nucl. Instrum. Meth. A **389**, 81–86 (1997)

[56] A. Ryd et al. *EvtGen: A Monte Carlo Generator for B-Physics*, EVTGEN-V00-11-07 (2005)

[57] H. Petersen and M. Bleicher and S. A. Bass and H. Stocker, *UrQMD v2.3: Changes and Comparisons* (2008), arXiv:0805.0567 [hep-ph]

[58] T. Sjöstrand et al. *An Introduction to PYTHIA 8.2*, Comput. Phys. Commun., **191**, 159–177 (2015)

[59] H. Pi, *An Event Generator for Interactions Between Hadrons and Nuclei: FRITIOF version 7.0*, Comput. Phys. Commun., **71**, 173–192 (1992)

[60] S. Agostinelli et al. *GEANT4–A simulation Toolkit*, Nucl. Instrum. Meth. A **506**, 250–303 (2003)

[61] R. Frühwirth, *Application of Kalman Filtering to Track and Vertex Fitting*, Nucl. Instrum. Meth. A **262**, 444–450 (1987)

[62] S. Spataro, *Event Reconstruction in the PandaRoot Framework*, J. Phys. Conf. Ser. **396**, 022048 (2012)

[63] J. Therhaag, *TMVA Toolkit for Multivariate Data Analysis in Root*, 35th International Conference on High Energy Physics, (2010), doi: 10.22323/1.120.0510

[64] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Advances in Neural Information Processing Systems 32, 8024–8035 (2019)

[65] M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Online Paper (2015)

[66] M. Hansroul and H. Jeremie, and D. Savard, *Fast Circle Fit With the Conformal Mapping Method*, CERN-DD/88/20 (1988)

[67] P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Conf. Proc. C, **590914**, 554–558 (1959)

[68] P.V.C. Hough, *Method and Means for Recognizing Complex Patterns*, Patent, United States (1962)

[69] Alexopoulos et al., *Implementation of the Legendre Transform for Track Segment Reconstruction in Drift Tube Chambers*, Nucl. Instrum. Meth. A, **592**, 456–462 (2008)

[70] A. Glazov et al., *Filtering Tracks in Discrete Detectors Using a Cellular Automaton*, **329**, 262–268 (1993)

[71] I. Kisel et al., *Cellular Automaton and Elastic Net for Event Reconstruction in the Nemo-2 Experiment*, Nucl. Instrum. Methods Phys. Res. A, **387**, 433–442 (1997)

[72] J. Hertz et al., *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Company (1991)

[73] A. Strandlie, and R. Frühwirth, *Track and Vertex Reconstruction: From Classical to Adaptive Methods*, Rev. Mod. Phys., **82**, 1419–1458 (2010)

[74] Regler et al., *Filter Methods in Track and Vertex Reconstruction*, International Journal of Modern Physics C, **07**, 521–542 (1996)

[75] R. Mankel, *Pattern Recognition and Event Reconstruction in Particle Physics Experiments*, Reports on Progress in Physics, **67**, 553–622 (2004)

[76] F. James, *Fitting Tracks in Wire Chambers Using the Chebyshev Norm Instead of Least Squares*, Nucl. Instrum. Meth., **211**, 145–152 (1983)

[77] N. Chernov et al., *Track and Vertex Reconstruction in Discrete Detectors Using Chebyshev Metrics*, Comput. Phys. Commun., **74**, 217–227 (1993)

[78] S. Amrouche et al., *The Tracking Machine Learning Challenge: Accuracy Phase* (2021), arXiv:1904.06778 [hep-ex]

[79] ATLAS Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, Tech. Rep. CERN-LHCC-2017-021, ATLAS-TDR-030, CERN, Geneva (2017), doi: 10.17181/CERN.FOZZ.ZP3Q

[80] R. Karabowicz, *Global Track Finder for PANDA experiment*, GSI SCIENTIFIC REPORT (2010)

[81] S. Wolfram, *Statistical Mechanics of Cellular Automata*, Rev. Mod. Phys. **55**, 601-644 (1983), doi: 10.1103/RevModPhys.55.601

[82] A. Strandlie, J. Wroldsen, R. Frühwirth and B. Lillekjendlie, *Particle Tracks Fitted on the Riemann Sphere*, Comput. Phys. Commun. **131** 95-108 (2000).

[83] J. Schumann, *Entwicklung eines schnellen Algorithmus zur Suche von Teilchenspuren im "Straw Tube Tracker" des PANDA-Detectors*, Bachelor Thesis, Fachhochschule Aachen, Campus Jülich, Jülich (2013)

[84] J. Schumann, *Beschleunigung eins Spurfindealgorithmus für den Straw Tube Tracker des PANDA-Detektors durch Parallelisierung mit CUDA C*, Master Thesis, Fachhochschule Aachen, Campus Jülich, Jülich (2015)

[85] J. Regina, *Time-Based Reconstruction of Hyperons at PANDA at FAIR*, Licentiate Thesis, Uppsala University, Uppsala (2021)

[86] W. Ikegami Andersson et al. *A Generalized Approach to Longitudinal Momentum Determination in Cylindrical Straw Tube Detectors*, Comput Softw Big Sci **5**, 18 (2021), doi: 10.1007/s41781-021-00064-0

[87] R. O. Duda and P. E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Commun. ACM **15** (1972), p. 11.

[88] Lehrach et al., *Beam Performance and Luminosity Limitations in the High-energy Storage Ring (HESR)*, Nucl. Instrum. Meth. A, **561**, 289–296 (2006)

[89] A. Samuel, *Some Studies in Machine Learning Using the Game of Checkers*, in IBM Journal of Research and Development, **3**-3, 210–229 (1959), doi: 10.1147/rd.33.0210.

[90] P. Cunningham et al., *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, Springer Berlin, Heidelberg (2008). doi: 10.1007/978-3-540-75171-7

[91] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996)

[92] Lindholm et al., *Supervised Machine Learning*, Cambridge University Press (2020)

[93] Sutton et al., *Reinforcement Learning: An Introduction*, MIT Press (2018)

[94] W. S. McCulloch, W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics, **5**, 115-133 (1943)

[95] W. Pitts, WS S. McCulloch, *How we know universals the perception of auditory and visual forms*, The bulletin of mathematical biophysics, **9**, 127-147 (1947)

[96] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press (2016)

[97] D. Rumelhart, G. Hinton, R. Williams, *Learning representations by back-propagating errors*, Nature, **323**, 533–536 (1986), doi: 10.1038/323533a0

[98] S. Hochreiter, J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, **9**(8), 1735–1780 (1997), doi: 10.1162/neco.1997.9.8.1735

[99] K. Cho et al., *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (2014), arXiv:1409.1259 [cs.CL]

[100] Y. LeCun, *Generalization and Network Design Strategies*, Technical Report CRG-TR-89-4, University of Toronto (1989)

[101] A. Vaswani et al., *Attention Is All You Need*, 31st Conference on Neural Information Processing Systems (2017), arXiv:1706.03762 [cs.CL]

[102] M. Bronstein et al., *Geometric Deep Learning: Going beyond Euclidean data*, IEEE Signal Processing Magazine, **34**, 18–42 (2017), doi: 10.1109/MSP.2017.2693418.

[103] M. Bronstein et al., *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges* (2021) arXiv:2104.13478 [cs.LG]

[104] T. Cohen et al., *Spherical CNNs*, in Proceedings of the 6th International Conference on Learning Representations (2018), arXiv:1801.10130 [cs.LG]

[105] M. Zaheer et al., *Deep Sets*, 31st Conference on Neural Information Processing Systems (2017), arXiv:1703.06114 [cs.LG]

[106] J. Lee et al., *Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks*, Thirty-sixth International Conference on Machine Learning (2019), arXiv:1810.00825 [cs.LG]

[107] J. Masci et al., *Geodesic Convolutional Neural Networks on Riemannian Manifolds* (2015) arXiv:1501.06297 [cs.CV]

[108] W.L. Hamilton, *Graph Representation Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, **14**-3, 1–159 (2020)

[109] A. Sperduti and A. Starita, *Supervised Neural Networks for the Classification of Structures*, IEEE Transactions on Neural Networks, **8**, 714–735 (1997), doi: 10.1109/72.572108.

[110] M. Gori, G. Monfardini, F. Scarselli, *A New Model for Learning in Graph Domains*, IEEE International Joint Conference on Neural Networks, **2**, 729–734 (2005), doi: 10.1109/IJCNN.2005.1555942.

[111] A. Micheli, "*Neural Network for Graphs: A Contextual Constructive Approach*, IEEE Transactions on Neural Networks, **20** no. 3, 498–511 (2009), doi: 10.1109/TNN.2008.2010350.

[112] F. Scarselli et al., *The Graph Neural Network Model*, IEEE Transactions on Neural Networks, **20**, 61–80 (2009), doi: 10.1109/TNN.2008.2005605.

[113] T. N. Kipf, M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, Fifth International Conference on Learning Representations (2017), arXiv:1609.02907 [cs.LG]

[114] J. Gilmer et al., *Neural Message Passing for Quantum Chemistry*, in Proceedings of the 34 th International Conference on Machine Learning (2017), arXiv:1704.01212 [cs.LG]

[115] P. Velickovic et al., *Graph Attention Networks*, Sixth International Conference on Learning Representations (2018), arXiv:1710.10903 [stat.ML]

[116] Battaglia et al., *Interaction Networks for Learning about Objects, Relations and Physics* (2016), arXiv:1612.00222 [cs.AI]

[117] Battaglia et al., *Relational Inductive Biases, Deep Learning, and Graph Networks* (2018), arXiv:1806.01261 [cs.LG]

[118] Zhou et al., *Graph Neural Networks: A Review of Methods and Applications* (2018), arXiv:1812.08434 [cs.LG]

[119] Zhang et al., *Deep Learning on Graphs: A Survey* (2018), arXiv:1812.04202 [cs.LG]

[120] Lee et al., *Attention Models in Graphs: A Survey* (2018), arXiv:1807.07984

[121] Zhang et al., *Graph Convolutional Networks: A Comprehensive Review*, Computational Social Networks, **6**-11 (2019), doi: 10.1186/s40649-019-0069-y

[122] Wu et al., *A Comprehensive Survey on Graph Neural Networks* (2019), arXiv:1901.00596 [cs.LG]

[123] S.K. Kumar, *On Weight Initialization in Deep Neural Networks* (2017), arXiv:1704.08863 [cs.LG]

[124] Xavier Glorot and Yoshua Bengio, *Understanding the Difficulty of Training Deep Feedforward Neural Networks*, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, **9**, 249–256 (2010)

[125] K. He et al., *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* (2015), arXiv:1502.01852 [cs.CV]

[126] Ghosh et al., *Comparative Analysis of Genetic Algorithm, Simulated Annealing and Cutting Angle Method for Artificial Neural Networks*, Machine Learning and Data Mining in Pattern Recognition (Book), doi: 10.1007/11510888_7 (2005)

[127] F. Glover and M. Laguna, *Tabu Search*, Handbook of Combinatorial Optimization, Springer, Boston, MA (1998), doi: 10.1007/978-1-4613-0303-9_33

[128] Dauphin et al., *Identifying and Attacking the Saddle Point Problem in High-dimensional Non-convex Optimization*, arXiv:1406.2572 [cs.LG] (2014)

[129] Y. Nesterov, *A Method for Unconstrained Convex Minimization Problem With the Rate of Convergence O(1/k2)*. Doklady ANSSSR (translated as Soviet.Math.Docl.), **269**, 543–547 (1983)

[130] Duchi et al., *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*, Journal of Machine Learning Research, **12**, 2121–2159 (2011)

[131] Matthew D. Zeiler, *ADADELTA: An Adaptive Learning Rate Method* (2012), arXiv:1212.5701 [cs.LG]

[132] T. Tieleman and G. Hinton, *Lecture 6.5-RmsProp: Divide the Gradient by a Running Average of Its Recent Magnitude*. COURSERA: Neural Networks for Machine Learning, 4, 26–31, (2012)

[133] D. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, International Conference on Learning Representations (2015), arXiv:1412.6980 [cs.LG]

[134] T. Dozat, *Incorporating Nesterov Momentum into Adam*, International Conference on Learning Representations (2016)

[135] Reddi et al., *On the Convergence of Adam and Beyond*, International Conference on Learning Representations (2018), arXiv:1904.09237 [cs.LG]

[136] I. Loshchilov, F. Hutter, *Decoupled Weight Decay Regularization*, International Conference on Learning Representations (2019), arXiv:1711.05101 [cs.LG]

[137] Sebastian Ruder, *An Overview of Gradient Descent Optimization Algorithms*, arXiv:1609.04747 [cs.LG] (2016)

[138] K. Albertsson et al., *Machine Learning in High Energy Physics Community White Paper*, arXiv:1807.02876 [physics.comp-ph] (2018)

[139] HEP ML Community, *A Living Review of Machine Learning for Particle Physics*, https://iml-wg.github.io/HEPML-LivingReview/

[140] Kisel et al., *Applications of Neural Networks in Experimental Physics*, Phys. Part. Nucl., **24**, 657–676 (1993)

[141] Bruce H. Denby, *Neural Networks in High Energy Physics: A Ten Year Perspective*, Comput. Phys. Commun., **119**, 219–231 (1999)

[142] John J. Hopfield, *Neural Networks and Physical Systems With Emergent Collective Computational Abilities*, Proc. Nat. Acad. Sci., **79**, 2554–2558 (1982)

[143] Bruce H. Denby, *Neural Networks and Cellular Automata in Experimental High Energy Physics*, Comput. Phys. Commun., **49**, 429–448 (1988)

[144] C. Peterson, *Track Finding With Neural Networks*, Nucl. Instrum. Methods Phys. Res. A, **279**, 537–545 (1989)

[145] C. Peterson and J. R. Anderson, *A Mean Field Theory Learning Algorithm for Neural Networks*, Complex Systems, 1, 995–1019 (1987)

[146] G. Stimpfl-Abele and L. Garrido, *Fast Track Finding With Neural Networks*, Comput. Phys. Commun., **64**, 46–56 (1991)

[147] S. Baginyan et al., *Tracking by a Modified Rotor Model of Neural Network*, Comput. Phys. Commun., **79**, 165–178 (1994)

[148] A. Badalà et al., *Neural Tracking in ALICE*, Nucl. Instrum. Methods Phys. Res. A, **502**, 503–506 (2003)

[149] A. Badalà et al., *Combined Tracking in the ALICE Detector*, Nucl. Instrum. Methods Phys. Res. A, **534**, 211-216 (2004)

[150] M. Ohlsson et al., *Track Finding with Deformable Templates: The Elastic Arms Approach*, Comput. Phys. Commun., **71**, 77–98 (1992)

[151] M. Ohlsson, *Extensions and Explorations of the Elastic Arms Algorithm*, Comput. Phys. Commun., **77**, 19–32 (1993)

[152] TrackML Challenge, *Particle Tracking Challenge on Kaggle*, Web: TrackML Challenge

[153] Project HEP.TrkX, *HEP Advanced Tracking Algorithms With Cross-cutting Applications* (2016), Web: Project HEP.TrkX

[154] S. Farrell et al., *The HEP.TrkX Project: Deep Neural Networks for HL-LHC Online and Offline Tracking*, EPJ Web of Conference, in Proceedings of Connecting The Dots/Intelligent Tracker (CTD/WIT 2017), **150**, 00003 (2017), doi: 10.1051/epjconf/201715000003

[155] Project Exa.TrkX, *HEP Advanced Tracking Algorithms at the Exascale* (2019). Web: Project Exa.TrkX

[156] S. Farrell et al., *Novel Deep Learning Methods for Track Reconstruction*, Connecting the Dots Workshop (2018), arXiv:1810.06111 [hep-ex]

[157] N. Choma et al., *Track Seeding and Labelling with Embedded-space Graph Neural Networks*, Connecting the Dots Workshop (2020), arXiv:2007.00149

[physics.ins-det]

[158] Ju et al., *Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors*, Machine Learning and the Physical Sciences (NeurIPS) (2019), arXiv:2003.11603 [physics.ins-det]

[159] Biscarat et al., *Towards a Realistic Track Reconstruction Algorithm Based on Graph Neural Networks for the HL-LHC* (2021), arXiv:2103.00916 [physics.ins-det]

[160] J. Pata et al., *MLPF: Efficient Machine-learned Particle-flow Reconstruction Using Graph Neural Networks*, Eur. Phys. J. C **81**, 381 (2021), arXiv:2101.08578 [physics.data-an]

[161] Hewes et al., *Graph Neural Network for Object Reconstruction in Liquid Argon Time Projection Chambers* (2021), arXiv:2103.06233 [hep-ex]

[162] W. Guan et al., *Quantum Machine Learning in High Energy Physics* (2020), arXiv:2005.08582 [quant-ph]

[163] Tuysuz et al., *Performance of Particle Tracking Using a Quantum Graph Neural Network* (2020), arXiv:2012.01379 [quant-ph]

[164] A. Akram and X. Ju, *Track Reconstruction using Geometric Deep Learning in the Straw Tube Tracker (STT) at the PANDA Experiment*, Connecting the Dots Workshop (2022), arXiv:2208.12178 [hep-ex]

[165] Xiangyang et al., *Performance of a Geometric Deep Learning Pipeline for HL-LHC Particle Tracking*, Eur. Phys. J. C, **81**, 876/10 (2021), doi: 10.1140/epjc/s10052-021-09675-8

[166] Donald E. Knuth, *Components and Traversal*, https://cs.stanford.edu/ knuth/fasc12a+.pdf (2022)

[167] He et al., *The Connected-Component Labeling Problem: a Review of State-of-the-art Algorithms*, Pattern Recognition, **70**, 25–43 (2017), doi: 10.1016/j.patcog.2017.04.018

[168] A. Hennequin et al., *SparseCCL: Connected Components Labeling and Analysis for sparse images*, Conference on Design and Architectures for Signal and Image Processing (DASIP), Montreal, QC, Canada, 2019, pp. 65-70, doi: 10.1109/DASIP48288.2019.9049184

[169] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, *Exploring network structure, dynamics, and function using NetworkX*, in Proceedings of the 7th Python in Science Conference (SciPy2008), 11–15, (2008)

[170] M. Fey, J.E. Lenssen, *Fast Graph Representation Learning with PyTorch Geometric*, ICLR Workshop on Representation Learning on Graphs and Manifolds (2019), arXiv:1903.02428 [cs.LG]

[171] A. Alicke et al., *Track Finding for the PANDA Experiment*, Connecting the Dots Workshop [Conference presentation] (2022)

[172] R. Liaw et al., *Tune: A Research Platform for Distributed Model Selection and Training* (2018), arXiv:1807.05118 [cs.LG]

[173] K. He et al., *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (2016), arXiv:1512.03385 [cs.CV]

[174] G. Perez Andrade, *Production of the $\bar{\Sigma}^0$ hyperon in the PANDA experiment at FAIR*, Master Thesis, Uppsala University, Uppsala (2019)

[175] Catarina E. Sahlberg, *On Antihyperon-Hyperon Production in Antiproton-Proton Collisions with the PANDA Experiment*, Master Thesis, Uppsala University, Uppsala (2007)

[176] A. Capella et al., *Dual Parton Model*, Physics Reports, **236**, 225–329 (1994). doi: 10.1016/0370-1573(94)90064-7

[177] P. Avery, *Vertexing and Kinematic Fitting, Part I: Basic Theory*, Lecture given at SLAC (1998)

[178] P. Avery, *Vertexing and Kinematic Fitting, Part III: Vertex Fitting*, Lecture given at SLAC (1998)

[179] M. Gustafsson, *Monte Carlo simulations of D-mesons with extended targets in the PANDA detector*, Master Thesis, Uppsala University, Uppsala (2016)

[180] DeepLearning.AI, *Initializing neural networks*, `https://www.deeplearning.ai/ai-notes/initialization`, [Online; Accessed 30-Dec-2022]

[181] Sergey Ioffe, Christian Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167 [cs.LG] (2015)

[182] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, *Layer Normalization*, arXiv:1607.06450 [stat.ML] (2016)

Glossary

Barrel DIRC Barrel Detection of Internally Reflected Cherenkov 36
BToF Barrel Time-of-Flight 34, 35

COSY COoler SYnchrtron 25
CR Collector Ring 25, 26

DIRC Detection of Internally Reflected Cherenkov 35
Disc DIRC Disc Detection of Internally Reflected Cherenkov 36

EMC Electromagnetic Calorimeter 36
ESR Experimental Storage Ring 24

FAIR Facility for Anti-proton and Ion Research 13, 24
FRICH Forward Ring Imaging CHerenkov 39
FSC Forward Spectrometer Electromagnetic Calorimeter 38, 40
FToF Forward Time-of-Flight 39
FTS Forward Tracking Stations 38

GSI GSI Helmholtz Center for Heavy Ion Research 24

HESR High Energy Storage Ring 13, 24–26

LMD Luminosity Detector 38

MRS Muon Range System 36, 40
MS Muon System 36

NESR New Experimental Storage Ring 25
nominal interaction point In particle physics, an interaction point (IP) is the place where particles collide. One differentiates between the nominal IP, which is the design position of the IP, and the real or physics IP, which is the position where the particles actually collide. The real IP is the primary "vertex" of the particle collision 31

p-LINAC Proton LINear ACcelerator 24, 25
PANDA antiProton ANihilation at DArmstadt 13, 24

QCD Quantum Chromodynamics 27

RESR Recuperated Experimental Storage Ring 25, 26

RICH Ring Imaging CHerenkov 38

SIS18 Heavy-ion Synchrotron 18 24

ToF Time-of-Flight 38

UNILAC Universal Linear Accelerator 24

Acta Universitatis Upsaliensis

Uppsala Dissertations from the Faculty of Science

Editor: The Dean of the Faculty of Science

1–11: 1970–1975

12. *Lars Thofelt*: Studies on leaf temperature recorded by direct measurement and by thermography. 1975.
13. *Monica Henricsson*: Nutritional studies on Chara globularis Thuill., Chara zeylanica Willd., and Chara haitensis Turpin. 1976.
14. *Göran Kloow*: Studies on Regenerated Cellulose by the Fluorescence Depolarization Technique. 1976.
15. *Carl-Magnus Backman*: A High Pressure Study of the Photolytic Decomposition of Azoethane and Propionyl Peroxide. 1976.
16. *Lennart Källströmer*: The significance of biotin and certain monosaccharides for the growth of Aspergillus niger on rhamnose medium at elevated temperature. 1977.
17. *Staffan Renlund*: Identification of Oxytocin and Vasopressin in the Bovine Adenohypophysis. 1978.
18. *Bengt Finnström*: Effects of pH, Ionic Strength and Light Intensity on the Flash Photolysis of L-tryptophan. 1978.
19. *Thomas C. Amu*: Diffusion in Dilute Solutions: An Experimental Study with Special Reference to the Effect of Size and Shape of Solute and Solvent Molecules. 1978.
20. *Lars Tegnér*: A Flash Photolysis Study of the Thermal Cis-Trans Isomerization of Some Aromatic Schiff Bases in Solution. 1979.
21. *Stig Tormod*: A High-Speed Stopped Flow Laser Light Scattering Apparatus and its Application in a Study of Conformational Changes in Bovine Serum Albumin. 1985.
22. *Björn Varnestig*: Coulomb Excitation of Rotational Nuclei. 1987.
23. *Frans Lettenström*: A study of nuclear effects in deep inelastic muon scattering. 1988.
24. *Göran Ericsson*: Production of Heavy Hypernuclei in Antiproton Annihilation. Study of their decay in the fission channel. 1988.
25. *Fang Peng*: The Geopotential: Modelling Techniques and Physical Implications with Case Studies in the South and East China Sea and Fennoscandia. 1989.
26. *Md. Anowar Hossain*: Seismic Refraction Studies in the Baltic Shield along the Fennolora Profile. 1989.
27. *Lars Erik Svensson*: Coulomb Excitation of Vibrational Nuclei. 1989.
28. *Bengt Carlsson*: Digital differentiating filters and model based fault detection. 1989.
29. *Alexander Edgar Kavka*: Coulomb Excitation. Analytical Methods and Experimental Results on even Selenium Nuclei. 1989.
30. *Christopher Juhlin*: Seismic Attenuation, Shear Wave Anisotropy and Some Aspects of Fracturing in the Crystalline Rock of the Siljan Ring Area, Central Sweden. 1990.

31. *Torbjörn Wigren*: Recursive Identification Based on the Nonlinear Wiener Model. 1990.
32. *Kjell Janson*: Experimental investigations of the proton and deuteron structure functions. 1991.
33. *Suzanne W. Harris*: Positive Muons in Crystalline and Amorphous Solids. 1991.
34. *Jan Blomgren*: Experimental Studies of Giant Resonances in Medium-Weight Spherical Nuclei. 1991.
35. *Jonas Lindgren*: Waveform Inversion of Seismic Reflection Data through Local Optimisation Methods. 1992.
36. *Liqi Fang*: Dynamic Light Scattering from Polymer Gels and Semidilute Solutions. 1992.
37. *Raymond Munier*: Segmentation, Fragmentation and Jostling of the Baltic Shield with Time. 1993.

Prior to January 1994, the series was called *Uppsala Dissertations from the Faculty of Science*.

Acta Universitatis Upsaliensis

Uppsala Dissertations from the Faculty of Science and Technology
Editor: The Dean of the Faculty of Science

1–14: 1994–1997. 15–21: 1998–1999. 22–35: 2000–2001. 36–51: 2002–2003.

52. *Erik Larsson*: Identification of Stochastic Continuous-time Systems. Algorithms, Irregular Sampling and Cramér-Rao Bounds. 2004.
53. *Per Åhgren*: On System Identification and Acoustic Echo Cancellation. 2004.
54. *Felix Wehrmann*: On Modelling Nonlinear Variation in Discrete Appearances of Objects. 2004.
55. *Peter S. Hammerstein*: Stochastic Resonance and Noise-Assisted Signal Transfer. On Coupling-Effects of Stochastic Resonators and Spectral Optimization of Fluctuations in Random Network Switches. 2004.
56. *Esteban Damián Avendaño Soto*: Electrochromism in Nickel-based Oxides. Coloration Mechanisms and Optimization of Sputter-deposited Thin Films. 2004.
57. *Jenny Öhman Persson*: The Obvious & The Essential. Interpreting Software Development & Organizational Change. 2004.
58. *Chariklia Rouki*: Experimental Studies of the Synthesis and the Survival Probability of Transactinides. 2004.
59. *Emad Abd-Elrady*: Nonlinear Approaches to Periodic Signal Modeling. 2005.
60. *Marcus Nilsson*: Regular Model Checking. 2005.
61. *Pritha Mahata*: Model Checking Parameterized Timed Systems. 2005.
62. *Anders Berglund*: Learning computer systems in a distributed project course: The what, why, how and where. 2005.
63. *Barbara Piechocinska*: Physics from Wholeness. Dynamical Totality as a Conceptual Foundation for Physical Theories. 2005.
64. *Pär Samuelsson*: Control of Nitrogen Removal in Activated Sludge Processes. 2005.

65. *Mats Ekman*: Modeling and Control of Bilinear Systems. Application to the Activated Sludge Process. 2005.
66. *Milena Ivanova*: Scalable Scientific Stream Query Processing. 2005.
67. *Zoran Radovic´*: Software Techniques for Distributed Shared Memory. 2005.
68. *Richard Abrahamsson*: Estimation Problems in Array Signal Processing, System Identification, and Radar Imagery. 2006.
69. *Fredrik Robelius*: Giant Oil Fields – The Highway to Oil. Giant Oil Fields and their Importance for Future Oil Production. 2007.
70. *Anna Davour*: Search for low mass WIMPs with the AMANDA neutrino telescope. 2007.
71. *Magnus Ågren*: Set Constraints for Local Search. 2007.
72. *Ahmed Rezine*: Parameterized Systems: Generalizing and Simplifying Automatic Verification. 2008.
73. *Linda Brus*: Nonlinear Identification and Control with Solar Energy Applications. 2008.
74. *Peter Nauclér*: Estimation and Control of Resonant Systems with Stochastic Disturbances. 2008.
75. *Johan Petrini*: Querying RDF Schema Views of Relational Databases. 2008.
76. *Noomene Ben Henda*: Infinite-state Stochastic and Parameterized Systems. 2008.
77. *Samson Keleta*: Double Pion Production in $dd \rightarrow \alpha\pi\pi$ Reaction. 2008.
78. *Mei Hong*: Analysis of Some Methods for Identifying Dynamic Errors-invariables Systems. 2008.
79. *Robin Strand*: Distance Functions and Image Processing on Point-Lattices With Focus on the 3D Face-and Body-centered Cubic Grids. 2008.
80. *Ruslan Fomkin*: Optimization and Execution of Complex Scientific Queries. 2009.
81. *John Airey*: Science, Language and Literacy. Case Studies of Learning in Swedish University Physics. 2009.
82. *Arvid Pohl*: Search for Subrelativistic Particles with the AMANDA Neutrino Telescope. 2009.
83. *Anna Danielsson*: Doing Physics – Doing Gender. An Exploration of Physics Students' Identity Constitution in the Context of Laboratory Work. 2009.
84. *Karin Schönning*: Meson Production in pd Collisions. 2009.
85. *Henrik Petrén*: η Meson Production in Proton-Proton Collisions at Excess Energies of 40 and 72 MeV. 2009.
86. *Jan Henry Nyström*: Analysing Fault Tolerance for ERLANG Applications. 2009.
87. *John Håkansson*: Design and Verification of Component Based Real-Time Systems. 2009.
88. *Sophie Grape*: Studies of PWO Crystals and Simulations of the $\bar{p}p \rightarrow \bar{\Lambda}\Lambda, \bar{\Lambda}\Sigma^0$ Reactions for the PANDA Experiment. 2009.
90. *Agnes Rensfelt*. Viscoelastic Materials. Identification and Experiment Design. 2010.
91. *Erik Gudmundson*. Signal Processing for Spectroscopic Applications. 2010.
92. *Björn Halvarsson*. Interaction Analysis in Multivariable Control Systems. Applications to Bioreactors for Nitrogen Removal. 2010.
93. *Jesper Bengtson*. Formalising process calculi. 2010.
94. *Magnus Johansson*. Psi-calculi: a Framework for Mobile Process Calculi. Cook your own correct process calculus – just add data and logic. 2010.
95. *Karin Rathsman*. Modeling of Electron Cooling. Theory, Data and Applications. 2010.

96. *Liselott Dominicus van den Bussche*. Getting the Picture of University Physics. 2010.
97. *Olle Engdegård*. A Search for Dark Matter in the Sun with AMANDA and IceCube. 2011.
98. *Matthias Hudl.* Magnetic materials with tunable thermal, electrical, and dynamic properties. An experimental study of magnetocaloric, multiferroic, and spin-glass materials. 2012.
99. *Marcio Costa.* First-principles Studies of Local Structure Effects in Magnetic Materials. 2012.
100. *Patrik Adlarson*. Studies of the Decay $\eta \rightarrow \pi^+\pi^-\pi^0$ with WASA-at-COSY. 2012.
101. *Erik Thomé.* Multi-Strange and Charmed Antihyperon-Hyperon Physics for PANDA. 2012.
102. *Anette Löfström.* Implementing a Vision. Studying Leaders' Strategic Use of an Intranet while Exploring Ethnography within HCI. 2014.
103. *Martin Stigge.* Real-Time Workload Models: Expressiveness vs. Analysis Efficiency. 2014.
104. *Linda Åmand.* Ammonium Feedback Control in Wastewater Treatment Plants. 2014.
105. *Mikael Laaksoharju.* Designing for Autonomy. 2014.
106. *Soma Tayamon.* Nonlinear System Identification and Control Applied to Selective Catalytic Reduction Systems. 2014.
107. *Adrian Bahne.* Multichannel Audio Signal Processing. Room Correction and Sound Perception. 2014.
108. *Mojtaba Soltanalian.* Signal Design for Active Sensing and Communications. 2014.
109. *Håkan Selg.* Researching the Use of the Internet — A Beginner's Guide. 2014.
110. *Andrzej Pyszniak.* Development and Applications of Tracking of Pellet Streams. 2014.
111. *Olov Rosén.* Parallel Stochastic Estimation on Multicore Platforms. 2015.
112. *Yajun Wei.* Ferromagnetic Resonance as a Probe of Magnetization Dynamics. A Study of FeCo Thin Films and Trilayers. 2015.
113. *Marcus Björk.* Contributions to Signal Processing for MRI. 2015.
114. *Alexander Madsen.* Hunting the Charged Higgs Boson with Lepton Signatures in the ATLAS Experiment. 2015.
115. *Daniel Jansson.* Identification Techniques for Mathematical Modeling of the Human Smooth Pursuit System. 2015.
116. *Henric Taavola.* Dark Matter in the Galactic Halo. A Search Using Neutrino Induced Cascades in the DeepCore Extension of IceCube. 2015.
117. *Rickard Ström.* Exploring the Universe Using Neutrinos. A Search for Point Sources in the Southern Hemisphere Using the IceCube Neutrino Observatory. 2015.
118. *Li Caldeira Balkeståhl.* Measurement of the Dalitz Plot Distribution for $\eta \rightarrow \pi^+\pi^-\pi^0$ with KLOE. 2015.
119. *Johannes Nygren.* Input-Output Stability Analysis of Networked Control Systems. 2016.
120. *Joseph Scott.* Other Things Besides Number. Abstraction, Constraint Propagation, and String Variable Types. 2016.
121. *Andrej Andrejev.* Semantic Web Queries over Scientific Data. 2016.

122. *Johan Blom*. Model-Based Protocol Testing in an Erlang Environment. 2016.
123. *Liang Dai*. Identification using Convexification and Recursion. 2016.
124. *Adriaan Larmuseau*. Protecting Functional Programs From Low-Level Attackers. 2016.
125. *Lena Heijkenskjöld*. Hadronic Decays of the ω Meson. 2016.
126. *Delphine Misao Lebrun*. Photonic crystals and photocatalysis. Study of titania inverse opals. 2016.
127. *Per Mattsson*. Modeling and identification of nonlinear and impulsive systems. 2016.
128. *Lars Melander*. Integrating Visual Data Flow Programming with Data Stream Management. 2016.
129. *Kristofer Severinsson*. Samarbete = Samverkan? En fallstudie av AIMday vid Uppsala universitet. 2016.
130. *Nina Fowler*. Walking the Plank of the Entrepreneurial University. The little spinout that could? 2017.
131. *Kaj Jansson*. Measurements of Neutron-induced Nuclear Reactions for More Precise Standard Cross Sections and Correlated Fission Properties. 2017.
132. *Petter Bertilsson Forsberg*. Collaboration in practice. A multiple case study on collaboration between small enterprises and university researchers. 2018.
133. *Andreas Löscher*. Targeted Property-Based Testing with Applications in Sensor Networks. 2018.
134. *Simon Widmark*. Causal MMSE Filters for Personal Audio. A Polynomial Matrix Approach. 2018.
135. *Damian Pszczel*. Search for a new light boson in meson decays. 2018.
136. *Joachim Pettersson*. From Strange to Charm. Meson production in electron-positron collisions. 2018.
137. *Elisabeth Unger*. The Extremes of Neutrino Astronomy. From Fermi Bubbles with IceCube to Ice Studies with ARIANNA. 2019.
138. *Monica Norberg*. Engagerat ledarskap för att skapa förutsättningar för allas delaktighet. Utgångspunkter i kvalitetsarbetet. 2019.
139. *Peter Backeman*. Quantifiers and Theories. A Lazy Aproach. 2019.
140. *Walter Ikegami Andersson*. Exploring the Merits and Challenges of Hyperon Physics. with PANDA at FAIR. 2020.
141. *Petar Bokan*. Pair production of Higgs bosons in the final state with bottom quarks and τ leptons in the ATLAS experiment. Search results using LHC Run 2 data and prospect studies at the HL-LHC. 2020.
142. *Carl Kronlid*. Engineered temporary networks. Effects of control and temporality on inter-organizational interaction. 2020.
143. *Alexander Burgman*. Bright Needles in a Haystack. A Search for Magnetic Monopoles Using the IceCube Neutrino Observatory. 2020.
144. *Eleni Myrto Asimakopoulou*. Search for charged Higgs bosons with tau-lepton signatures at the ATLAS experiment of the Large Hadron Collider and development of novel semiconductor particle detectors. 2021.
145. *Oscar Samuelsson*. Sensor Fault Detection and Process Monitoring in Water Resource Recovery Facilities. 2021.
146. *Helena Fornstedt*. Innovation Resistance. Moving Beyond Dominant Framings. 2021.

147. *Jenny Regina.* Time for Hyperons. Development of Software Tools for Reconstructing Hyperons at PANDA and HADES. 2022.
148. *Viktor Thorén.* Hadron Physics in a Polarized World.Exploring Electromagnetic Interactions with Spin Observables. 2022.
149. *Thomas Mathisen.* Exotic Decays of Vector-Like Quarks and Development of a Test Procedure for the ITk Strip Module at the ATLAS Detector for the HL-LHC. 2022.
150. *Adeel Akram*. Towards Realistic Hyperon Reconstruction in PANDA. From Tracking with Machine Learning to Interactions with Residual Gas. 2023.