Optimisation of the SHiP muon shield and event reconstruction in the SND@LHC emulsions using Machine Learning

Filips Fedotovs

A thesis presented to the University College of London
for the degree of Doctor of Philosophy

Department of Physics and Astronomy
University College of London
United Kingdom
October 15, 2025

Supervisor: Prof. Mario Campanelli

Co-Supervisor: Dr. Timothy Scanlon

Declaration

I, Filips Fedotovs confirm that the work presented in my thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This work studies the use of machine learning techniques in high-energy physics to increase the discovery potential of two experiments. Five different studies were carried out within two CERN experiments: SHiP and SND@LHC, the former being an approved proposal, and the latter having taken data since the start of LHC Run 3. These experiments, run by a largely overlapping collaboration, are designed to measure or discover Feebly Interacting Particles (FIPs), predicted by Hidden Valley theories, or neutrinos. They require very strong signal/background separation, making them ideal testing grounds for Machine Learning techniques.

One of the key issues of the SHiP proposal is the muon shield, which is expected to deflect muons produced by the interactions of the SPS beam on a beam dump. In order for the experiment to reach its sensitivity to FIPs, no muon can reach the sensitive part of the detector, leading to a very complex magnetic field structure. Evolutionary algorithms were employed to derive an optimal magnetic shield design, also accounting for practical factors. The SND@LHC detector, located 480 metres away from the ATLAS collision point, behind a large amount of concrete and rock, measures neutrinos and searches for FIPs interacting in an emulsion tracker. Neural networks were applied to analyse high-resolution emulsion data in the presence of noise and misalignments. Multiple aspects of the reconstruction and analysis have been covered, including tracking, vertexing, particle classification, and momentum estimation.

Under specific conditions, machine learning can greatly improve the performance of standard algorithms, even if their practical implementation is far from straightforward...

Impact Statement

This thesis applies machine learning (ML) techniques to enhance the discovery potential of two CERN experiments: SHiP and SND@LHC, which search for Feebly Interacting Particles (FIPs) and neutrinos. These particles are essential to exploring physics beyond the Standard Model. The research addresses critical challenges in these experiments, such as background suppression, complex detector design, and the reconstruction of faint particle signals using high-resolution emulsion data.

The thesis contributes novel approaches to experimental high-energy physics, including the use of evolutionary algorithms for optimising magnetic shielding and neural networks for precise particle tracking, vertexing, classification, and momentum estimation, using a modular and scalable architecture suitable for large datasets. These tools not only improve the performance of SHiP and SND@LHC but are transferable to other experiments across particle and astroparticle physics, and even medical physics.

Educationally, this research exemplifies interdisciplinary training by combining physics, computer science, and data analysis. It offers valuable material for graduate-level courses in machine learning for science and can help modernise experimental physics curricula.

ML techniques developed in this work are relevant for any domain requiring the identification of rare patterns within large, noisy datasets. Applications include medical imaging, radiation monitoring, industrial sensing, and defence. The approach to optimising magnetic field configurations under real-world constraints can inspire similar solutions in engineering design and environmental shielding.

Moreover, the publics growing interest in AI and fundamental science creates a strong foundation for outreach. This work supports communication initiatives that highlight how artificial intelligence is revolutionising scientific discovery. It has the potential to inspire young researchers and inform science and technology policy.

Results are already being implemented within CERN collaborations and will be disseminated through peer-reviewed publications, conference presentations, open-source repositories, and participation in international ML/HEP workshops. The tools and methods are being shared with the wider scientific community to promote reuse and further development.

Dedication

This thesis is dedicated to the loving memory of my grandparents, Dzidra, Zigfrid, Nina, and Eugene. Their unwavering support, endless wisdom, and boundless love have been my guiding light throughout this journey. Although they are no longer with us, their spirit continues to inspire and motivate me every day. I am forever grateful for their belief in me and the values they instilled in me. This work is a tribute to their legacy and a testament to the profound impact they had on my life.

I miss you dearly and strive to make you proud with each step I take.

Acknowledgements

Completing this PhD thesis has been a challenging but incredibly rewarding journey, and it would not have been possible without the support, guidance and encouragement of many individuals.

First, I would like to express my deepest gratitude to my supervisor, Mario Campanelli, for providing me with this research opportunity, for his invaluable guidance, unwavering support and insightful feedback. His expertise and mentorship have been instrumental in shaping my academic growth and success. *Grazie infinite.*

I am immensely grateful to my committee members, Prof. Tim Scanlon and Prof. David Waters, for their constructive critiques and suggestions, which have significantly enhanced the quality of this work. I am also grateful to my examiners, Prof. Gavin Hesketh and Prof. Mitesh Patel, for their careful review of my thesis and for the valuable feedback provided during the viva.

The study was carried out on two different and complex experiments, which required considerable assistance and expertise from the relevant teams. I particularly acknowledge Oliver Lantwin, Sergei Shirobokov, and Evgeny Kurbatov for imparting valuable insights and support during my research work at SHiP. At SND@LHC, I extend my gratitude to Antonia Di Crescenzo for the opportunity to conduct my research for this experiment and to Antonio Iuliano for assisting me in navigating the new and previously unfamiliar field of emulsion detectors. Additionally, I thank Thomas Ruf for helping with the software aspects of both experiments.

I would like to acknowledge the financial, infrastructural, and academic support from UCL, without which this research would not have been possible.

To my parents, Aija and Pavel, and my brother, Alan, thank you for your patience, encouragement, and your wonderful sense of humour. Your belief in me has continually motivated me, and I am forever grateful for your support.

I extend my deepest gratitude to my beloved partner, Ania Marchetti, for your boundless love, understanding, and steadfast faith in me. Your support has been my anchor through the most challenging times, and your presence has made this journey even more significant.

Finally, I want to acknowledge all those who are not named here but have provided invaluable support and help during the past six years, both in happy and difficult times throughout my doctorate.

Contents

Chapter 1

Introduction

The detection of the Higgs boson in 2012[1, 2] solidified the Standard Model (SM) of particle physics as one of the most well-established theories, accurately describing most of the known particles and their interactions. However, substantial evidence, among others from astrophysical observations and neutrino oscillation experiments, suggests that the Standard Model is incomplete, with the primary issues outlined below.

Description of gravity: the Standard Model does not include the gravitational force, and attempts to formulate a renormalizable quantum theory of gravity so far have failed. Rather, it is described by general relativity, another successful but conceptually different framework. In general, these theories minimally overlap: the SM addresses the microscopic realm, while general relativity focuses on cosmological structures. However, conflicts arise in areas where both theories intersect. One notable example is the *cosmological constant problem*[3], where the energy density estimates of the Standard Model and the ΛCDM[1] model vary significantly.

Baryon asymmetry of the universe: it is believed that at the moment of the Big Bang, matter and antimatter were produced in equal quantities, but current observations indicate that matter significantly dominates over antimatter. The successful explanation of this phenomenon requires a charge-parity (CP) violation that has been observed in numerous hadronic interactions[4, 5, 6], but its magnitude is way too small[7] to explain BAU.

Dark matter: according to Newton's laws of gravity and given that their visible matter is concentrated in a central core, it is expected that the rotational velocity of the stars in galaxies decreases with their distance to the galactic centre, with a behaviour similar to the velocity of planets in the solar system. However, astronomical studies of distant galaxies have shown[8] a different picture: the galaxy rotation curves are flatter than expected for their observed mass distributions. Similar behaviour was observed[9] in much larger cosmological structures such as galactic clusters. The most natural explanation for this puzzle implies that most of the actual mass of the galaxies is made up of the substance called *dark matter* (DM), which does not participate in electromagnetic interactions and is invisible to us. It

---

[1] A concordance cosmological model that naturally incorporates general relativity as a gravitational theory.

is widely accepted that dark matter makes up approximately 87% of all matter in the universe[10], yet its exact nature remains unknown as none of the particles in the Standard Model could be good DM candidates.

Neutrino masses: in the original formulation of the Standard Model, it was assumed that neutrinos are massless, as there was no significant experimental evidence to prove otherwise. Eventually, neutrino oscillation experiments[11, 12, 13] have shown that neutrinos do possess masses. Their exact values are still unknown, but they are estimated to be on the sub-eV scale[14], well below the energy resolution of modern physics detectors. A neutrino can be a typical Dirac particle and obtain a mass within the framework of the SM using the Higgs mechanism. This implies the values of Yukawa coupling constants on the order of $\mathcal{O}(10^{-12})$, which are significantly lower than for their charged lepton counterparts. Not all physicists are comfortable with this idea, and other possibilities, such as the neutrino being a Majorana particle with a mass-generating seesaw mechanism are considered. In any case, right-handed neutrinos (or right-handed components) must exist and mix with ordinary neutrinos.

All of these problems suggest that although the Standard Model is a remarkable accomplishment that has significantly improved our understanding of the universe, its shortcomings stimulate physicists to look for a more comprehensive theory.

Starting in the 1970s, SuperSymmetry theory, commonly known as SUSY, emerged as the foremost candidate for this new theory, predicting the presence of additional heavy particles. The search for new particles, predicted by SUSY or other theories, has motivated physicists to explore the *energy frontier* of particle physics using collider-based experiments such as LEP and LHC, with the latter achieving an unprecedented value of centre-of-mass energy of 13.6 TeV[15]. Despite this extraordinary feat, no evidence of massive non-SM particles has been observed so far. This fact alone is not sufficient to rule out new physics theories, as masses of new particles could be so large as to be out of the reach of current colliders.

This leaves physicists with two possible paths: pushing the energy frontier further by building colliders with higher centre-of-mass energies, or investigating new physics searching for theories predicting light particles with small interaction rates. The first task is extremely difficult as existing colliders are already at the peak of current technology, and further advancements would entail significant financial and logistical hurdles. Although there are plans for projects like the Future Circular Collider[16], these are several decades from being operational. This fact has led more physicists to explore other options, such as X-ray astronomy[17], precise tests of the predictions of the SM, and exploration of other Beyond the Standard Model (BSM) theories.

Several recent BSM models incorporate the Hidden Sector (HS), which contains light but weakly interacting particles. These particles would not have been discovered already: not because they are too heavy, but because their production or detection rate is too small. This encourages researchers to investigate the *intensity frontier*, which can be probed by high interaction rates combined with low background environments. Colliders face challenges in achieving the necessary intensities because of constraints on the bunch-crossing interaction rate. In addition to that, crossing points of hadron colliders such as Fermilab[18] or LHC are subject to a particularly challenging background due to the complex physics of the proton-proton collision events, resulting in reduced sensitivity to the rare interaction processes.

The ideal experimental conditions to explore this regime are found in beam-dump fixed-target experiments, where a dense material is exposed to an intense beam of incoming particles, leading to a large number of interaction events inside the target. All particles with relatively large interactions with matter are absorbed, and only neutrinos and Dark Matter particles would reach the detector. Despite the much lower centre-of-mass energy than that of collider experiments, it compensates for it with a higher interaction rate. Furthermore, beam-dump experiments actively reuse existing infrastructure, such as collider injectors and tunnels, making these experiments relatively inexpensive. A somewhat intermediate approach between standard collider experiments and beam dumps is to have detectors located far from a collider's interaction point, shielded by large amounts of rock and concrete. Also in that case, the vast majority of particles produced in the collision are absorbed, leaving only neutral, weakly interacting particles reaching the detector.

The high-intensity, fixed-target experiments also suffer from a background that can obscure the detection of potential signals. For highly sensitive searches, a zero background environment is desired: a stringent requirement that necessitates considerable technological advancements in experiment design, shielding, and advanced data analysis techniques. Consequently, this necessitates more efficient data extraction, modelling, and interpretation from experiments than existing techniques can provide. This requirement does not concern only beam dump experiments; even future collider facilities such as HL-LHC[19] and FCC will face the same problems with increasing luminosity and associated pile-up.

One potential solution to this issue lies within the realm of artificial intelligence (AI), particularly its subset, machine learning (ML), which has seen significant advances in recent decades and has become increasingly successful in practical applications such as image correction[20], speech[21], and facial recognition[22].

To address this gap and explore the potential benefits of ML in practice, a study has been carried out on two experiments focused on exploring BSM physics at the low-background/high intensity frontier: SHiP[23] and SND@LHC[24]. SHiP, a beam target experiment, is designed to investigate the hidden sector, and Section 3 of this thesis will be devoted to employing evolutionary algorithms to optimise the SHiP magnetic shield against muon background. SND@LHC[24], an experiment located 480 metres from LHC interaction point 1, aims to investigate neutrinos originating from the LHC and to search for low-mass dark matter particles. Chapter 4 outlines efforts to improve the quality of the reconstruction of emulsion data at SND@LHC through the application of artificial neural networks. Both studies deal with complex physical and machine learning concepts, the theoretical basis of which is provided in Chapter 2. Lastly, Section 5 summarises the findings and describes a strategy for future research.

While promising, the performance of machine learning methods depends on the realism of the training data. Since many analyses rely on Monte Carlo simulations, any inaccuracies in modelling physics or detector responses can affect generalisability. However, as simulation frameworks continue to improve, the corresponding machine learning models, such as those in this work, are also expected to become more robust and accurate.

Chapter 2

Theoretical background

This chapter aims to establish a crucial theoretical foundation for the research carried out in this thesis and is divided into six sections. The first section offers a brief overview of the Standard Model, the theoretical framework for the forthcoming discussions on background effects and signal signatures specific to the SHiP and SND@LHC experiments. For a more detailed analysis of the Standard Model and particle detection techniques, readers can consult [25, 26].

The second part will be dedicated to a discussion of the *feebly interacting particles* (FIP) predicted by a variety of BSM models. The SHiP experiment aims to carry out model-independent studies of hidden particles, focussing primarily on the heavy neutral lepton or HNL. In contrast, SND@LHC is better suited for dark photon searches. Therefore, these topics will be emphasised in the discussion.

The third section will primarily examine the characteristics of both SM and BSM particles, their interactions with detection instruments, and offer a brief overview of past experimental searches in this area.

The fourth and fifth sections describe the experiments SHiP and SND@LHC, respectively, covering their physical objectives, detector components, and the relevant software frameworks used.

The subject of machine learning will be introduced in the last part since ML has been used to address challenges in both SHiP and SND@LHC experiments. This will include an in-depth description of evolutionary algorithms that will be used in Chapter 3 devoted to SHiP and the subject of deep learning that will be used in Section 4 dedicated to the SND@LHC experiment.

2.1 The Standard Model

Since its development in the mid-20th century, the Standard Model of particle physics offers the most successful description of known elementary particles and all their interactions except gravity. The theory is believed[27][28] to be mathematically self-consistent up to a Planck scale, and until now, no experiment has provided any results that would directly contradict its predictions. The Standard Model itself is based on the mathematical framework of quantum field theory (QFT). In this framework, particles are not permanent entities but excitations of underlying quantum fields, where energy can be converted into matter and vice versa. This makes processes such as particle creation and their mutual destruction possible. A classic example is electron-positron annihilation, where the lepton-antilepton pair most commonly produces two photons. In QFT, this is described by the interaction between the electron, positron, and electromagnetic fields.

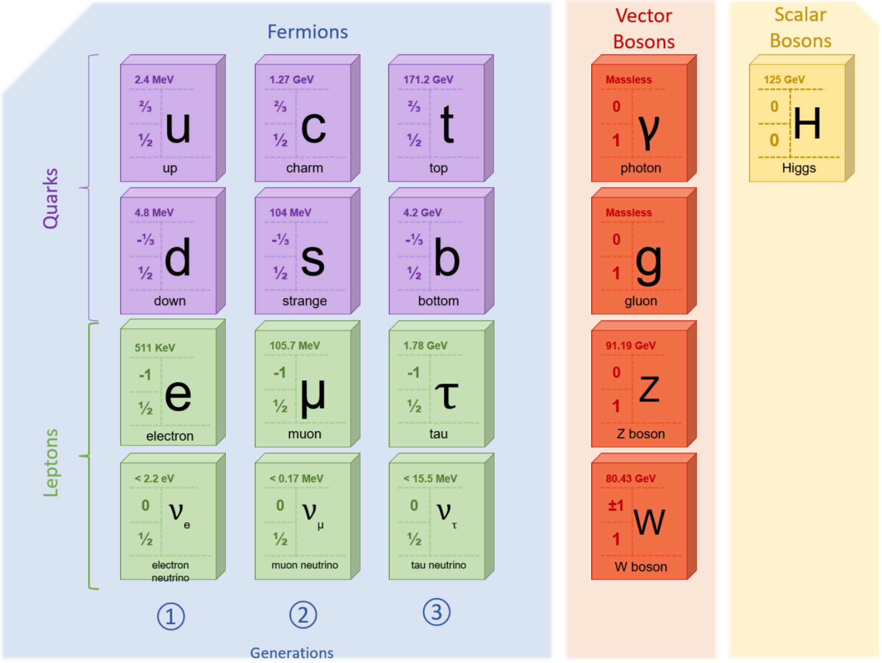


Figure 2.1: The particle content of the Standard Model, generated using data from [29].

The Standard Model of particle physics consists of two types of particles: fermions and bosons, and three forces: strong, electromagnetic, and weak, with the latter two unified into the framework of electroweak interaction. The interaction among these particles and forces prior to spontaneous symmetry breaking can be described using the Lagrangian formalism of the Standard Model:

$$\mathcal{L}_{SM} = \underbrace{\sum_{f}^{N_f} i\bar{\psi}_f\gamma^\mu D_\mu\psi_f}_{\text{Fermion terms}} - \underbrace{\sum_{v}^{N_v}\frac{1}{2}TrF^{(v)}_{\mu\nu}F^{(v)}_{\mu\nu}}_{\text{Vector boson terms}} + \underbrace{|D_\mu\phi|^2 - V(\phi)}_{\text{Scalar (Higgs) boson term}} + \underbrace{\sum_{f}^{N_f}[-\frac{g_f}{\sqrt{2}}\nu(\bar{f}_L f_R + \bar{f}_R f_L)]}_{\text{Fermion mass terms}} + h.c., \tag{2.1}$$

where each term is described in more detail in the following subsections.

2.1.1 Fermions

Particles with half-odd integer spin, known as fermions, obey Fermi-Dirac statistics and are subject to the Pauli exclusion principle. The Dirac equation describes their free motion, and can be expressed in the Lagrangian formalism as follows:

$$\mathcal{L}_{\mathcal{D}} = i\bar{\psi}\gamma^{\mu}\partial_{\mu}\psi - m\bar{\psi}\psi, \tag{2.2}$$

where ψ is the Dirac spinor field of the fermion, m is its mass, $\partial_{\mu} = \frac{\partial}{\partial x^{\mu}}$ is the partial derivative with respect to one direction of space-time and γ^{μ} are the Dirac gamma matrices.

Fermions in the Standard Model have twelve distinct varieties that can be divided into two equally numbered groups of quarks and leptons. These two categories can be further subdivided into three generations, each featuring particles with comparable properties but varying masses. Most of the fermions observed in the universe belong to the first generation, as the higher generations are heavier, unstable, and ultimately decay into first-generation particles. The reason behind nature's selection of this specific arrangement remains unknown.

Leptons are fermions that do not feel the strong nuclear interaction. They can be divided into two categories: charged leptons and neutral neutrinos. The charged group consists of electrons that are part of atoms, as well as their heavier and less stable counterparts, muons and taus. The neutral part of the leptonic sector is made up of three particles called neutrinos. Neutrinos are produced through weak interactions with charged leptons and are thus named after them: electron, muon, and tau neutrinos. These neutrino flavour eigenstates are not the basic neutrino mass eigenstates: as seen in neutrino oscillation experiments, each neutrino flavour is a linear combination of the three mass eigenstates ν_1, ν_2 and ν_3, mixing through the following equation:

$$\begin{pmatrix}\nu_e\\ \nu_\mu\\ \nu_\tau\end{pmatrix} = \begin{pmatrix}U_{e1} & U_{e2} & U_{e3}\\ U_{\mu1} & U_{\mu2} & U_{\mu3}\\ U_{\tau1} & U_{\tau2} & U_{\tau3}\end{pmatrix}\begin{pmatrix}\nu_1\\ \nu_2\\ \nu_3\end{pmatrix}, \tag{2.3}$$

where the terms U_{li} represent the elements of the PMNS matrix[30].

Quarks are typically heavier than leptons, and are involved in all four fundamental forces, including the strong interactions. According to the strong interaction theory, quarks obey colour confinement, and in fact, they have never been observed in isolation and always form composite structures called hadrons. The most commonly known hadrons are mesons, which consist of quark and anti-quark pairs, and baryons, which are made of three quarks[1].

Most of the observable baryon matter, such as protons and neutrons, is composed of two types of quark: an up quark and a down quark. These particles are also components of pions, which are the most commonly observed mesons in experiments. Heavier quarks, like charm or beauty, create metastable mesons, for example, J/ψ or Υ, which can only be generated in multi-GeV energy events.

Similarly to leptons, the quarks' flavour eigenstates in the weak interactions do not match exactly their mass eigenstates, with the relationship between these states

---

[1] Exotic multiquark states, such as *tetraquarks* (four quarks) and *pentaquarks* (five quarks), have also been observed.

described by the CKM matrix[31].

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}. \tag{2.4}$$

2.1.2 Bosons

Bosons are particles with integer spin values and obey Bose-Einstein statistics. Unlike fermions, bosons are not restricted by the Pauli exclusion principle and can share the same quantum state. The Standard Model recognises two types of bosons: vector bosons with spin one and scalar bosons with spin zero.

According to the SM, the **vector bosons** are mediators of the three fundamental forces: electromagnetic, weak, and strong. The motion of these particles is described by the Maxwell-Proca equation, expressed in the Lagrangian formalism as follows:

$$\mathcal{L}_{kin} = -\frac{1}{2} Tr F_{\mu\nu} F_{\mu\nu}, \tag{2.5}$$

where $F_{\mu\nu}$ is a field tensor for a particular vector boson F.

The electromagnetic force is mediated by the photon, a massless and uncharged particle. The weak force has three propagators: the W^-, W^+, and Z bosons. The W bosons are charged under the electromagnetic force and have a mass of 80.4 GeV[2]. The Z boson is a neutral particle that has a slightly higher mass of 91.2 GeV. The strong force is propagated by the electrically neutral gluon, which acts as a "glue" keeping the quarks together. Gluons themselves are massless, but they are charged under the strong force and can self-interact.

The only observed elementary **scalar** particle in the Standard Model is the Higgs boson. It has the highest mass of any fundamental boson, at 125 GeV, is subject to the weak force, and it interacts with all massive particles in the Standard Model. The discovery of the Higgs boson was pivotal in confirming the Higgs mechanism, the mass-generating schema for most SM particles. Since the Higgs boson is a fluctuation of the scalar field, its motion is described by the Klein-Gordon equation, which can be expressed as:

$$\mathcal{L}_{\text{kin}} = |D_\mu \phi|^2 - V(\phi), \tag{2.6}$$

where ϕ is the Higgs doublet, $V(\phi)$ is the scalar potential governing its self-interactions, and D_μ is the gauge-covariant derivative defined as $D_\mu = \partial_\mu - ig_W \frac{\tau^a}{2} W^a_\mu - ig'_W \frac{Y}{2} B_\mu$. Here, ∂_μ is the ordinary spacetime derivative, W^a_μ and B_μ are the $SU(2)_L$ and $U(1)_Y$ gauge fields, τ^a are the Pauli matrices, g_W and g'_W are the corresponding gauge couplings, and Y is the hypercharge of the Higgs doublet.

2.1.3 Interactions

According to Noether's theorem, each symmetry corresponds to a conservation law. This is seen in the Standard Model with the three sets of rotations: $U(1)_Y$, $SU(2)_L$

---

[2] To avoid clutter, natural units in which c = ℏ= 1 will be used throughout the thesis.

and $SU(3)$ local phase transformations, which give rise to the conserved currents of the electromagnetic, weak and strong interactions, respectively.

Electroweak interactions can be divided into two categories: neutral current (NC) interactions, which involve electrically neutral vector bosons, and charged current (CC) interactions, which are mediated by either positively or negatively charged W bosons.

Following spontaneous symmetry breaking, electromagnetic and weak forces emerge as a mixture of the hypercharge B field and the weak isospin fields $W^{(1)}$, $W^{(2)}$, and $W^{(3)}$. Together, they constitute the gauge group $SU(2)_L \times U(1)_Y$, where the generators for weak hypercharge and weak isospin are commonly denoted as Y and T, respectively. This gauge group unifies electromagnetic and weak interactions.

The interaction characteristics of the B field with a charged fermion emerge from the requirement that the fermion's Lagrangian remains invariant when its wavefunction phase undergoes a local phase transformation. This condition is satisfied if the additional calibration or the so-called gauge field is introduced and its interaction with the Lagrangian can be expressed as:

$$\mathcal{L}_B = -i\frac{g_W'}{2} Y \bar{\psi}\gamma^\mu B_\mu \psi. \tag{2.7}$$

Here, the quantity g_W' represents the coupling strength of the field B and the hypercharge Y.

The interaction properties with the field W arise from the requirement for the Lagrangian to be invariant under rotation in the weak isospin space, with the interaction Lagrangian of the form:

$$\mathcal{L}_{W_j} = -i\frac{g_W}{2} \bar{\psi}\gamma^\mu T_j W_\mu^j \psi, \tag{2.8}$$

where the quantity g_W represents the coupling strength of the W field and T_j is a particular component of the weak isospin. The physical photon and Z-boson fields arise from the following linear combinations:

$$A_\mu = +B_\mu \cos\theta_W + W_\mu^{(3)} \sin\theta_W, \tag{2.9}$$

$$Z_\mu = -B_\mu \sin\theta_W + W_\mu^{(3)} \cos\theta_W, \tag{2.10}$$

where θ_W is the weak mixing angle. The observable electric charge e is represented by the following combination:

$$e = g_W' \cos\theta_W = g_W \sin\theta_W. \tag{2.11}$$

In the Standard Model, all fundamental particles except neutrinos, photons, Z bosons, gluons, and Higgs possess an electric charge and thus take part in electromagnetic interactions. The fact that the gauge vector field is massless and uncharged makes the electromagnetic force unlimited in range. The coupling g_Z of the Z boson to the fermions has the following form:

$$g_Z = g_W' \cos\theta_W = g_W \sin\theta_W. \tag{2.12}$$

The coupling of the Z boson to fermions is more complicated, as its strength depends on the chiral state of the fermion in question and varies between different types of fermions. Interactions of Z bosons can include electrically neutral fermions, such as neutrinos, as long as they possess the appropriate chirality. For particles, this would be left-handed, and for antiparticles, it would be right-handed.

The two physical W bosons, W^+ and W^-, are involved in charged current interactions, which can be represented by a combination of the two fields.

$$W_\mu^\pm = \frac{1}{\sqrt{2}}(W_\mu^{(1)} \mp W_\mu^{(2)}). \tag{2.13}$$

In the weak interaction picture, all left-handed fermions (and right-handed antifermions) can be split into weak isospin doublets, which differ by one unit of electric charge:

$$\begin{pmatrix} \nu_e \\ e^- \end{pmatrix}_L, \begin{pmatrix} \nu_\mu \\ \mu^- \end{pmatrix}_L, \begin{pmatrix} \nu_\tau \\ \tau^- \end{pmatrix}_L, \begin{pmatrix} u \\ d' \end{pmatrix}_L, \begin{pmatrix} c \\ s' \end{pmatrix}_L, \begin{pmatrix} t \\ b' \end{pmatrix}_L.$$

Fermions with right-handed chirality, which are part of the weak isospin singlets, do not interact with W bosons and thus are not involved in weak interactions. For this reason, the interactions involving the W and Z bosons do not conserve parity. It should be noted that the coupling constant g_W is significantly stronger than that of the electromagnetic force. However, the substantial masses of the W and Z bosons limit the range of this interaction.

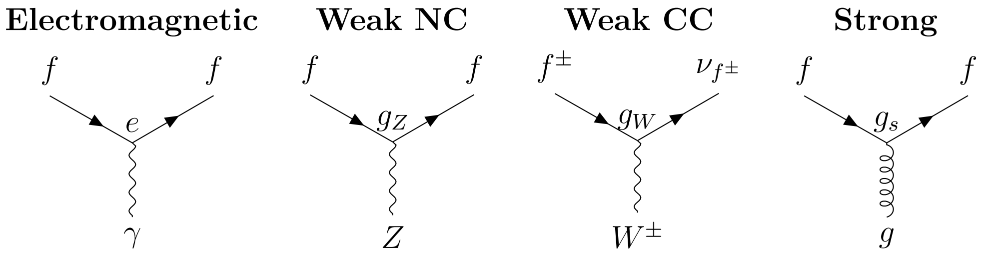


Table 2.1: Examples of Feynman diagrams for vertices involving electromagnetic, weak neutral-current, weak charged-current, and strong interactions.

Another important concept is the interaction of the Higgs field with vector bosons and fermions through the weak force. It is theorised that, during the high-temperature conditions prevalent in the early universe, the ground state of the Higgs potential was symmetric, with a unique minimum at $\phi = 0$. As the universe underwent cooling, the form of the Higgs potential changed, causing the $\phi = 0$ configuration to no longer be a state with the lowest energy, thereby becoming unstable. This instability forces the Higgs field to evolve towards and settle into one of the degenerate minima of the new potential, a phenomenon known as *spontaneous symmetry breaking*. As a result of this process, the Higgs field attains a non-zero expectation value ν and interacts with other fields, such as the W and Z bosons, causing these particles to obtain mass. The complete Lagrangian is quite complex and includes many terms; below is the component that accounts for fermion masses:

$$\mathcal{L}_m = -\frac{g_f}{\sqrt{2}}\nu(\bar{f}_L f_R + \bar{f}_R f_L), \tag{2.14}$$

where g_f is the Yukawa coupling of a particular fermion f. Since the coupling of the fermion to the Higgs field is proportional to its measured mass, it can be calculated by using the relation:

$$g_f = \sqrt{2}\frac{m_f}{\nu}. \tag{2.15}$$

Strong interactions arise from the requirement that the Lagrangian fermion is invariant under rotation in color[3] space. The corresponding gauge-vector boson is a gluon. The strong force in SM is described by quantum chromodynamics theory or shortly QCD, and its interaction Lagrangian can be expressed as:

$$\mathcal{L}_{QCD} = igA^{\mu}_{a}\bar{\psi}\gamma^{\mu}T^{a}\psi - gf^{abc}(\partial_{\mu}A^{a}_{\nu})A^{b\mu}A^{c\nu} - g^2 f^{eab}f^{ecd}A^{a}_{\mu}A^{b}_{\nu}A^{c\mu}A^{d\nu}. \tag{2.16}$$

In a similar way to the electroweak interaction, the strong interaction involves a charge-like quantity, known as g, which is only present for quarks and gluons in the Standard Model and is thus involved in QCD processes. The strength of this coupling is the greatest among all forces, reaching a value close to $\mathcal{O}(1)$ at low energies, which is why it is called "strong".

The gauge group of quantum chromodynamics is $SU(3)$, a non-abelian group, resulting in the appearance of new self-interaction terms. Coupled with the significant interaction strength, this confines the strong force to subatomic distances.

2.2 Feebly interacting particles

A variety of BSM theories involve expanding the Standard Model's Lagrangian with extra terms to represent a Hidden Sector (HS) of particles and interactions completely separate to those in the Standard Model. The two sectors can anyway, couple through *portal* interactions:

$$\mathcal{L} = \mathcal{L}_{SM} + \mathcal{L}_{portal} + \mathcal{L}_{HS}. \tag{2.17}$$

From an experimental physics perspective, the most intriguing term is $\mathcal{L}_{portal}$, which involves a feeble interaction with visible Standard Model particles and can be tested in experiments. There are numerous proposed portals; however, in this thesis, only renormalisable fermion, vector, and scalar sectors will be discussed on the basis of the available literature [32, 33, 34, 35, 36, 37].

2.2.1 Fermion portal

Neutrinos are much lighter than other fermions: although their masses can be explained through an interaction with the Higgs boson, like the other fermions (Dirac

---

[3] The term *color* in this context refers to the three states of the fermion color vector, which is similar to the spin but with three elements. It should not be confused with the term *colour* used in everyday life.

mechanism), the large disparity in the Yukawa couplings between neutrinos and the other fermions appears to be unnatural. This raises questions about the possibility of an alternative mass-generating mechanism.

Quite a popular alternative framework to explain the smallness of neutrino masses is the seesaw mechanism, which involves the addition of a Majorana mass term to the neutrino Lagrangian term.

$$\mathcal{L}_\nu = \mathcal{L}_{Dirac} + \mathcal{L}_{Majorana} = -m_D(\overline{\nu_R}\nu_L + \overline{\nu_L}\nu_R) - \frac{1}{2}m_M(\overline{\nu_R^c}\nu_R + \overline{\nu_R}\nu_R^c). \tag{2.18}$$

This expression can be rearranged into the following form:

$$\mathcal{L}_\nu = -\frac{1}{2}\begin{pmatrix}\overline{\nu_L} & \overline{\nu_R^c}\end{pmatrix}\begin{pmatrix}0 & m_D \\ m_D & m_M\end{pmatrix}\begin{pmatrix}\nu_L^c \\ \nu_R\end{pmatrix} + h.c.. \tag{2.19}$$

This additional term includes only weak isospin singlets and therefore it can be safely added without breaking the gauge invariance of the SM lagrangian.

The physical states of the neutrino are represented by the eigenvalues of the mass matrix, and assuming that $m_D << m_M$ can be approximated as:

$$m_\nu \approx \frac{m_D^2}{m_M} \quad \text{and} \quad m_N \approx m_M. \tag{2.20}$$

The left term represents the known SM light neutrino with the Dirac mass term similar to other fermions in the SM, but the physical mass is suppressed by the Majorana term. This hierarchy $m_D << m_M$ is considered natural since m_M is a gauge-singlet Majorana mass term and can take a much larger value without destabilising the theory.

The physical states of light SM and heavy neutrino can be written as the linear combination of light- and heavy-mass eigenstates in the following form:

$$\nu \approx \cos\theta(\nu_L + \nu_L^c) - \sin\theta(\nu_R + \nu_R^c) \quad and \quad N \approx \cos\theta(\nu_R + \nu_R^c) + \sin\theta(\nu_L + \nu_L^c), \tag{2.21}$$

where, N denotes the physical state of the heavy neutrino, obtained as a linear combination of the right-handed neutrino ν_R and the left-handed neutrino ν_L (and their charge-conjugates) with mixing angle θ. This equation paints the following picture:

- For each SM neutrino mass eigenstate ν_i, there is a heavier counterpart N_i commonly called a *sterile* neutrino or HNL.
- Each SM neutrino mass eigenstate ν_i consists almost entirely of left-chiral states and therefore can participate in weak interactions.
- On the other hand, each heavy neutrino N_i is predominantly in a right-chiral state $N_i \simeq \nu_{RI}$, and as a result, its interaction via the weak force is significantly suppressed by the $\sin\theta$ factor.

The term θ is the mixing between two mass eigenstates and can be approximated by $\theta \approx \arctan\frac{m_D}{m_M}$.

In addition, if neutrinos are confirmed to be Majorana particles, meaning that they are their own antiparticles, the lepton number would no longer be conserved in the

Standard Model. As a result of CP violation during the creation and oscillation of right-handed neutrinos, the lepton asymmetry is apportioned between left-handed leptons and right-handed neutrinos. This causes the asymmetry in the left-hand sector to be partially transformed into baryon asymmetry through rapid sphaleron transitions occurring in the hotter, earlier phases of the universe[38]. This process satisfies two of the conditions required by Sakharov to explain the Barion Asymmetry of the Universe (BAU), namely baryon number and CP violation. All of the above provides a substantial foundation for the Neutrino Minimal Standard Model (νMSM)[39], which extends the Standard Model by incorporating a set of heavy neutrinos for each generation. The νMSM is an elegant theory that necessitates only minor adjustments to the existing Standard Model framework and proposes a potential dark matter candidate N_1, assuming its mass lies within the keV range. This theory explains the anomalously small mass of neutrinos and offers a solution to the BAU problem. Nonetheless, νMSM has several limitations:

- It relies on the basis of the unnaturalness of the Yukawa couplings of Dirac neutrinos. Naturalness can be a useful tool in particle physics[40], but it is not necessarily a fundamental law of nature. To date, there is no substantial experimental evidence for the seesaw mechanism in the neutrino sector.
- Even if the seesaw mechanism exists in nature, there is no guarantee that it will be possible to confirm experimentally: contrary to the Yukawa coupling, the Majorana mass term m_M is not constrained by the Higgs field expectation value and can take an arbitrary number. If it is sufficiently large, it might be inaccessible for near-future particle physics experiments.
- Even if HNLs are observed, they do not automatically explain dark matter if they fail to satisfy the mass requirements. For sterile neutrinos as dark matter candidates, the allowed mass window is typically estimated to lie in the range of approximately 1 keV[41] to 50 keV[42].
- There is no evidence that neutrinos are Majorana particles; the confirmation of this hypothesis would be the observation of neutrinoless double beta decay ($0\nu\beta\beta$)[43].

Nevertheless, this theory is appealing from an experimental point of view, as some of the parameter space can be probed with existing technology.

2.2.2 Vector portal

This minimal extension of the Standard Model consists of adding to its Lagrangian two additional terms corresponding to the dark sector:

$$\mathcal{L}_{\mathcal{DS}} = \mathcal{L}_{\mathcal{A}'} + \mathcal{L}_{\chi}, \tag{2.22}$$

which can be expanded as:

$$\mathcal{L}_{\mathcal{DS}} = -\frac{1}{4}F'_{\mu\nu}F'^{\mu\nu} + \frac{m^2_{A'}}{2}A'^{\mu}A'_{\mu} - \frac{1}{2}\epsilon F'_{\mu\nu}F^{\mu\nu} + g_D A'^{\mu}\begin{cases}\bar{\psi}_{\chi}\gamma_{\mu}\psi_{\chi}\\ i[(\partial_{\mu}\phi^{\dagger}_{\chi})\phi_{\chi} - \phi^{\dagger}\partial_{\mu}\phi_{\chi}]\end{cases}. \tag{2.23}$$

The first term $-\frac{1}{4}F'_{\mu\nu}F'^{\mu\nu}$, describes a kinetic term of the new Abelian gauge field A'_μ associated with a dark $U(1)_D$ symmetry. It is analogous to the photon kinetic term in the Standard Model, which is why A'_μ is commonly referred to as a *dark photon*. However, unlike the SM photon, the dark photon can acquire a mass through the explicit mass term $\frac{m^2_{A'}}{2}A'^{\mu}A'_\mu$, which modifies its gauge properties and allows for new phenomenology. The actual portal that connects the dark sector to the SM is represented by the kinetic mixing term $-\frac{1}{2}\epsilon F'_{\mu\nu}F^{\mu\nu}$ where ϵ is the coupling strength. The last term of the equation can describe a Lagrangian of either scalar or fermion dark matter candidates χ.

The primary advantage of the dark photon theory is that it allows for a variety of direct and indirect detection methods but this model is not as comprehensive as the νMSM theory, as it is unlikely to provide an explanation for the Standard Model discrepancies that are not related to dark matter, such as the source of neutrino masses and the baryon asymmetry of the universe. Moreover, it is less constrained than νMSM and contains more parameters. For example, the dark photon $\mathcal{A}'$ can be either a massive or massless boson, and the dark particle χ may not be present at all. If it is, it could be either a fermion or a scalar.

2.2.3 Scalar portal

There are many theoretical realisations of the scalar portal; here, the simplest case will be presented with the following hidden scalar Lagrangian:

$$\mathcal{L}_{\mathrm{s}} = \frac{1}{2}\partial_\mu s \partial^\mu s + \frac{\lambda_s}{4}s^4 + \frac{\lambda_{hs}}{2}\Phi^\dagger \Phi s^2. \tag{2.24}$$

The first part of the Lagrangian is a kinetic term for a scalar field s, and the second describes the self-interaction of s with the coupling constant λ_s. The mass of the particle s is acquired through the last term with the coupling parameter λ_{hs} and can be expressed as:

$$m_s = \sqrt{\frac{\lambda_{hs}}{2}}v, \tag{2.25}$$

where v is the Higgs vacuum expectation value.

Phenomenologically, the model predicts that s interacts with nucleons via Higgs exchange, leading to spin-independent direct detection signals. Collider searches also look for missing energy signatures, such as invisible Higgs decays and monojet events at the LHC, while indirect detection efforts aim to observe gamma rays or cosmic rays from scalar dark matter annihilation. If the scalar mass lies within the weak scale, it behaves like a typical WIMP, whereas in the sub-GeV or ultralight regime, it could have different cosmological implications.

The model, while simple and renormalizable, faces fine-tuning issues, particularly in explaining why the Higgs-portal coupling remains small enough to evade direct detection limits while still producing the observed relic abundance. Despite its challenges, the Higgs portal scalar dark matter remains one of the most straightforward and testable frameworks for new physics beyond the Standard Model.

2.3 Experimental methods

2.3.1 Detection of the Standard Model particles

From the detection point of view, all particles in the Standard Model can be split into three groups: stable or quasi-stable particles that interact directly with the detector systems through electromagnetic or strong force, unstable particles that decay in the detector before being detected (and are indirectly observed through their decay products), and stable but weakly interacting neutrinos.

Electromagnetic interactions

Charged fermions can be easily identified due to the three primary ways they interact with the detector: ionisation, bremsstrahlung, and deflection by an external magnetic field.

Charged particles passing through a medium can cause ionisation by transferring their kinetic energy to the electrons in the medium. Energy loss can be accurately modelled using the Bethe-Bloch equation:

$$\frac{1}{\rho}\frac{dE}{dx} \approx -\frac{4\pi\hbar^2 c^2\alpha^2}{m_e v^2 m_u}\frac{Z}{A}\left\{\ln\left(\frac{2\beta^2\gamma^2 m_e c^2}{I_e}\right) - \beta^2\right\}. \tag{2.26}$$

Most of the terms in this equation, such as Z, $\hbar$ or c, are either physical constants or parameters that describe the chemical properties of the medium. Their values remain constant for different materials, so the ionisation mainly depends on the material density ρ and the mass and momentum of the incident particle, parameterised by the variables β and γ.

Bremsstrahlung is another important energy-loss mechanism that begins to dominate once charged particles reach high energies: as they pass near the nuclei of the medium, they decelerate and emit photons. In dense materials, successive bremsstrahlung and pair-production interactions of electrons lead to the development of *electromagnetic showers*, which are fully contained and measured in electromagnetic calorimeters. This results in energy loss that can be described by:

$$\frac{dE}{dx} \approx -4\alpha n Z^2 r_e^2 \ln\left(\frac{287}{\sqrt{Z}}\right) E\left(\frac{m_e}{m}\right)^2, \tag{2.27}$$

where all terms except E and $\left(\frac{m_e}{m}\right)^2$ relate to the chemical composition and density of the medium. The E term denotes the energy of the incident particle, and the additional mass-dependent factor suppresses bremsstrahlung for heavier particles: since the probability of radiation scales as $1/m^2$, muons and hadrons lose far less energy by bremsstrahlung compared to electrons. This strong mass dependence makes it straightforward to distinguish between electrons, which produce characteristic showers, and muons, which traverse the calorimeters with only minimal radiative losses and thus behave as minimum-ionising particles. Photons also cause electromagnetic showers in the calorimeters, but, being electrically neutral, they do not leave tracks in the tracking detectors.

If the particle is moving in a uniform magnetic field, and the density of the medium is small enough to allow it to travel without being absorbed, the charged particle trajectory will change according to the Lorentz law and form a helix. For a single charged particle, the parameters of the helix are related to the particle's momentum and to the value of the magnetic field according to the following equation:

$$\mathbf{p} = \frac{0.3BR}{\cos\lambda}, \tag{2.28}$$

where the λ is a pitch angle, B is the strength of the magnetic field in units of tesla, R is the radius of curvature in metres and $\mathbf{p}$ is the momentum of the particle, in GeV. The trajectory curvature of the particles can be measured in the tracking detectors. Combining this information with the energy measurement enables the reconstruction of the particle's full kinematic profile.

Strong interactions

While electromagnetic interactions govern the behaviour of charged leptons and photons in the detector, hadrons interact primarily via the strong force. When high-energy hadrons such as protons, neutrons, or pions traverse detector material, they undergo inelastic collisions with nuclei in the medium. These interactions initiate a cascade of secondary hadrons, which in turn interact and produce further particles. The resulting cascade is referred to as a *hadronic shower*.

Hadronic showers typically consist of a mixture of charged and neutral pions, kaons, protons, and neutrons, as well as electromagnetic sub-showers from the decay channel $\pi^0 \to \gamma\gamma$. Because of this mixed composition, hadronic showers are generally broader and less predictable than electromagnetic showers.

To contain and measure these cascades, detectors employ *hadronic calorimeters* (HCALs), usually placed outside the electromagnetic calorimeters. The HCALs measure the total energy deposited by the shower, allowing reconstruction of the incident hadrons energy.

In collider environments, collimated groups of hadrons originating from a single quark or gluon appear as *jets*. Jet reconstruction and identification play a central role in many physics analyses. However, due to the presence of electromagnetic sub-showers and detector resolution effects, jets can occasionally be misidentified as isolated electron or photon signals. This represents an important source of background in searches involving leptonic final states.

Prompt decays

The high-mass particles, including the Higgs boson, the massive vector bosons, and the tau lepton, do not have sufficient lifetimes to interact with detector components directly but disintegrate into a set of lighter SM particles, creating a decay vertex. By reconstructing the daughter particles and measuring their kinematics, it is possible to establish the mother's particle type and its invariant mass.

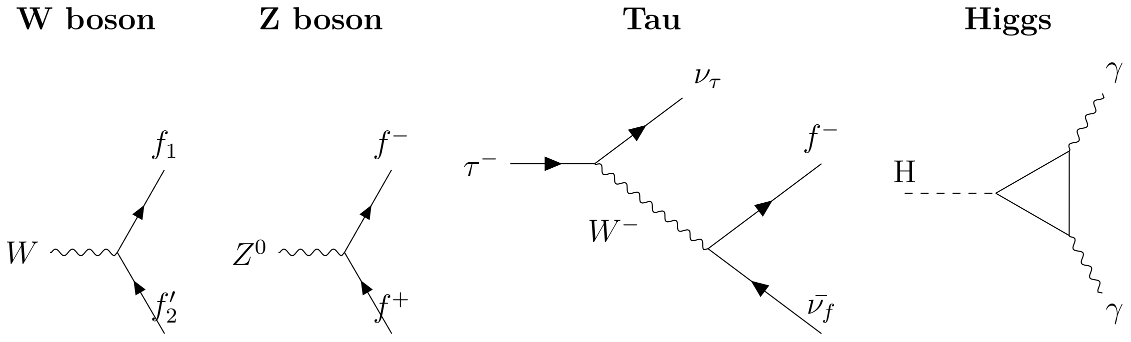


Table 2.2: Examples of the Feynman diagrams depicting the prompt decay of massive SM particles.

Neutrinos

One of the most interesting yet challenging groups of particles to detect is the neutrinos, as they only interact with matter via a weak force. In collider detectors such as ATLAS, the presence of neutrinos is indirectly reconstructed from the measured total momentum imbalance of the other measured particles. This technique is useful for identifying events containing neutrinos, but it is not ideal for neutrino studies: only the transverse missing energy of the event is recorded, and neutrinos escape the detector without interacting with it, so their properties cannot be really studied.

An alternative method to detect neutrinos is to construct a specialised detector that can reconstruct NC or CC interactions caused by neutrinos, examples of which are depicted in the diagrams below.

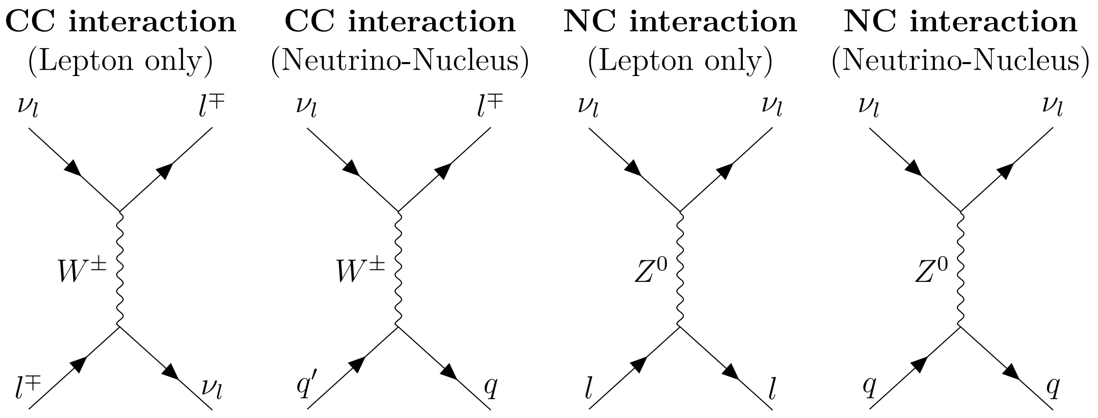


Table 2.3: Examples of the Feynman diagrams depicting the neutrino interactions on the elementary particle level.

To be able to detect even a small number of neutrinos, the target in neutrino detectors is usually extremely large in volume or made from high-density materials like lead or tungsten.

When a neutrino interacts within the detector, it can be observed through several methods. Detectors such as Super-Kamiokande[44], which use a liquid tank, identify neutrinos by detecting and amplifying light from Cherenkov radiation. This radiation is produced when the particles from the neutrino interaction exceed the speed of light in the tank's medium. The amplification process is facilitated by photomultiplier tubes installed on the walls of the tank.

The approach that will be used in both the SHiP and SND@LHC experiments, on the other hand, involves studying neutrinos by directly detecting the products of the neutrino interaction with the detector target, either by using scintillators or nuclear emulsion detectors. The latter employs photographic films with micrometric resolution that record charged particle tracks as sequences of silver grains. After exposure, the emulsions are chemically developed, and modern automated scanning systems reconstruct the three-dimensional trajectories, enabling precise measurements of interaction vertices and short-lived particle decays. The signal signature of the neutrino interaction event depends on:

- **The propagator:** in CC interactions involving W bosons, a new lepton is always produced, whereas, in NC interactions, only momentum transfer occurs between the incident neutrino and the target particle via the Z boson.
- **Flavour configuration:** in CC interactions, the neutrino flavour determines the type of new lepton formed and the flavour transition of the target particle. These probabilities can be computed using the PMNS and CKM matrices for nuclear interactions. For instance, when a muon neutrino interacts with a target down quark through a charged interaction, it typically emits a muon and changes the down quark's flavour to an up quark, thereby conserving charge, lepton number, and baryon number.

In addition to the interaction physics involved in neutrino events, the kinematics of the process is also critical, particularly for nuclear interactions. If the energies involved are low, then the incident neutrino merely passes momentum to the target quark system. These are generally the cases of quasi-elastic scattering (QES) when CC scattering occurs, or elastic NC scattering (ES), with the latter practically impossible to detect because of the absence of any detectable interaction products.

However, if the energy of the incident neutrino is sufficiently high, it can result in a more violent event: since the quarks shown in the Feynman diagrams in Table 2.3 are generally confined within either a neutron or a proton, their displacement leads to the hadronisation process with multiple hadrons emerging as a result of this interaction. These types of events are characterised as deep inelastic scattering or DIS processes that can occur for both the CC and NC interaction modes and can be easily distinguished by the presence or absence of the leading lepton, allowing the measurement of the CC/NC ratio.

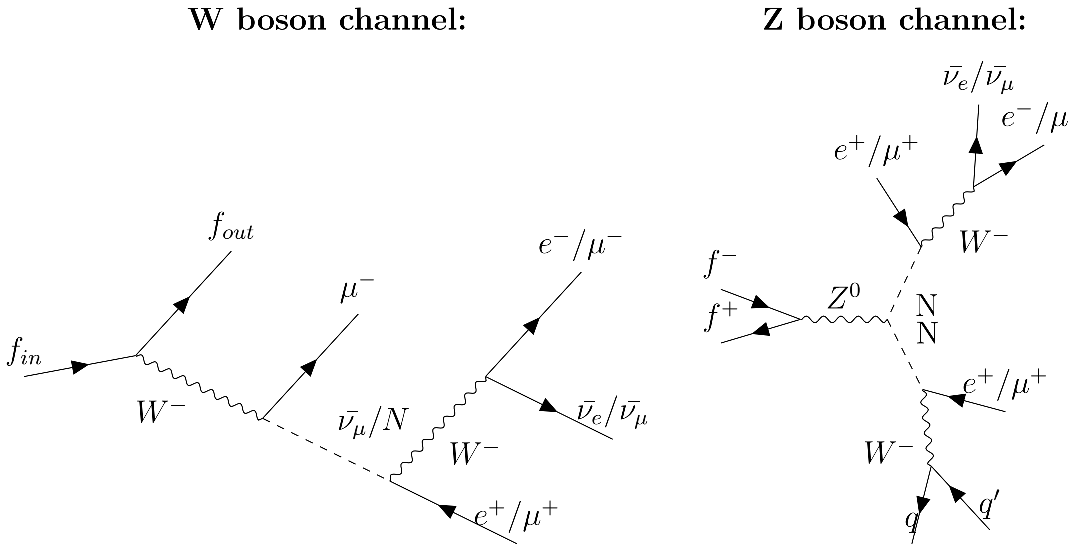
2.3.2 Detection of the feebly interacting particles

Experimental searches of HNL

HNL, like neutrinos, can be produced in decays of real or virtual W and Z bosons, so HNL searches are usually performed in collider mode in the case of real vector bosons or fixed targets for virtual ones.

The second important factor in the search for HNLs is detector hardware and software, which should satisfy the following requirements:

- **Efficient decay particle identification:** the are two main channels of HNL



W boson channel: **Z boson channel:**

Table 2.4: Examples of the hypothetical HNL chain via Z and W boson channel.

decay: leptonic through the $N \rightarrow \nu_l l^\pm l^\mp$ chain and the hadronic channel $N \rightarrow l^\pm qq'$. Identifying the type and charge of particles is essential in order to accurately reconstruct the decay topology. The branching ratio of the hadronic channel $N \rightarrow l^\pm qq'$ can reach $\approx 70\%$ of the total number of HNL decays, and therefore, the ability to reconstruct and identify mesons is highly desirable.

- **Sufficient detector size:** the HNL, as a weakly interacting particle, can travel a significant macroscopic distance before undergoing decay and, therefore, will escape detection if the size of the fiducial volume is insufficient. The tracking and vertexing software should be able to identify displaced tracks and vertices, which is a significant indication of HNL decay. Additionally, it should be able to detect prompt decays as well.
- **Efficient triggering system:** additional requirements include triggering events by missing energy or leptons with large momentum.

Many background processes can mimic the HNL signal features. These include misidentified metastable mesons, cosmic muons, and random track pairs that are mistakenly reconstructed as vertices.

In recent decades, numerous searches have been carried out using various experimental facilities, but to date, no conclusive detection of HNLs has been reported. However, these efforts have greatly restricted the possible phase-space allowed for these particles. Typically, this can be represented as a two-dimensional scatterplot where the particle mass is plotted on one axis and the interaction strength on the other. This plot can be broadly separated into four distinct quadrants.

Upper-left quadrant: these are low-mass (below 3 GeV) particles with relatively strong coupling to existing SM particles. This is the easiest sector to probe, as it does not require high energies nor large luminosities. Unsurprisingly, this is the first sector that has been extensively probed by various experiments such as PS191[45], CHARM[46] and NuTev[47].

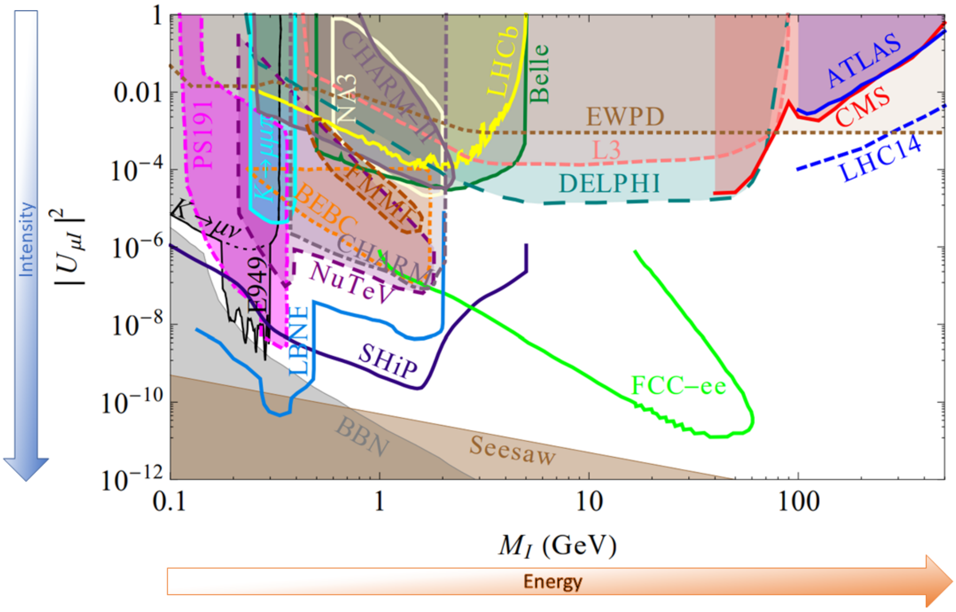


Figure 2.2: The parameter space of the hypothetical Heavy Neutral Lepton (HNL) and the shaded areas that were excluded by theoretical limits or experiments in the past. Note that only HNL coupling with a muon neutrino is explored in this plot. Taken from [23].

Upper-right quadrant: these are also strongly interacting particles, but with a higher mass of up to 500 GeV. Their production requires higher centre-of-mass energies that can be obtained in collider experiments, such as the Large Electron Positron Collider (LEP) or LHC, where the centre-of-mass energy can reach 13 TeV. Numerous experiments such as L3[48], Delphi[49] and ATLAS[50] have probed this sector, but apart from a few statistical flukes such as the *750 GeV diphoton excess at LHC*[51], no evidence of the new physics has been discovered.

Lower-left quadrant: this parameter space covers particles with low coupling and masses below 10 GeV. This area can be effectively constrained by various cosmological and physical models that set the limit on the value of the lowest coupling strength. The rest of the space can be probed by fixed-target experiments that provide sufficiently high statistics and a low-background environment. The SHiP experiment[23] aims to probe an unexplored domain of weakly interacting particles with masses up to $\mathcal{O}(10^1)$ GeV.

Lower-right quadrant: this parameter space covers weakly interacting high-mass particles. This is the most challenging domain of parameter space, which is beyond the reach of modern experimental equipment. Fixed-target experiments are limited by the mass of particles, such as b quarks, which could decay into HNLs. Collider experiments lack intensity because of the limits on the frequency of the beam crossing that generates the interaction. In addition to insufficient luminosities, the analysis of the interaction environment in hadron colliders is challenging because of the high QCD background produced by the proton-proton collisions. Electron-positron colliders, such as LEP, generate cleaner interaction events; however, their attainable centre-of-mass energies are much lower than those of their hadron counterparts, due to the high synchrotron energy losses. Another significant limitation of current collider experiments, such as ATLAS[52] and CMS[53] is their small detection volumes designed primarily to reconstruct prompt decay events. This enables most hypothetical weakly interacting particles in this area of the parameter space to escape the apparatus undetected.

Experimental searches of the dark photon

Numerous techniques can be employed to detect dark photons:

- **Resonance searches:** the process $A' \to l^+l^-$ is investigated by measuring the four-momentum of the produced leptons and looking for bumps in the invariant mass spectrum. Provided good mass resolution and substantial noise suppression, the dark boson can be identified by a narrow peak over the continuous background spectrum.
- **Missing energy/mass:** this method investigates the process $A' \to \chi\chi$ that can occur if the dark photon $\mathcal{A}'$ couples with dark sector particles and at least one of them is lighter than $\frac{m_{\mathcal{A}'}}{2}$. The signal signature consists of the reconstructed missing energy that has been carried away by the escaping dark-matter particles. This method requires a complete reconstruction of the initial and final states of the SM particles, including their energy and momentum.
- **Detection of scattering processes:** this technique requires the presence of particles χ in the dark sector and investigates it by examining the scattering event $\chi e^- \to \chi e^-$ and reconstructing the electromagnetic shower that was created by the deflected electron.

While in the HNL case, the search parameter space is based on two variables: the HNL mass and its coupling, in the dark-photon case, the picture is more complicated: In addition to the dark photon mass $m_{\mathcal{A}'}$ and its coupling ϵ, the mass and coupling of the hypothetical hidden particle sector χ also have to be considered making the search parameter space 4-dimensional.

Experimental searches for scalar dark matter

The expected experimental signature depends on the particular realisation of the theory; some of the examples are mentioned below:

- **Invisible Higgs boson decay:** the existence of hidden particles is inferred from the measurement of the Higgs branching ratio $\Gamma(h \to ss)$. This method has substantial discovery potential due to its model independence, but the observation of the Higgs invisible decays on its own will not constitute strong proof of dark matter detection.
- **Dark matter annihilation:** in this method, a signature $\langle s \rangle s \to \gamma\gamma$ or $\langle s \rangle s \to f\bar{f}$ is used to infer the interaction of the scalar particle s with the dark matter particles in the universe $\langle s \rangle$.
- **Detection of scattering processes:** this method is similar to the scattering technique which was described for the dark photon, but in this particular case, a DIS is considered, where the quark in the nucleus gets displaced by the dark scalar particle, producing a hadronic shower. The interaction is mediated by the Higgs boson h, which couples to the s and the SM quark. Given the need

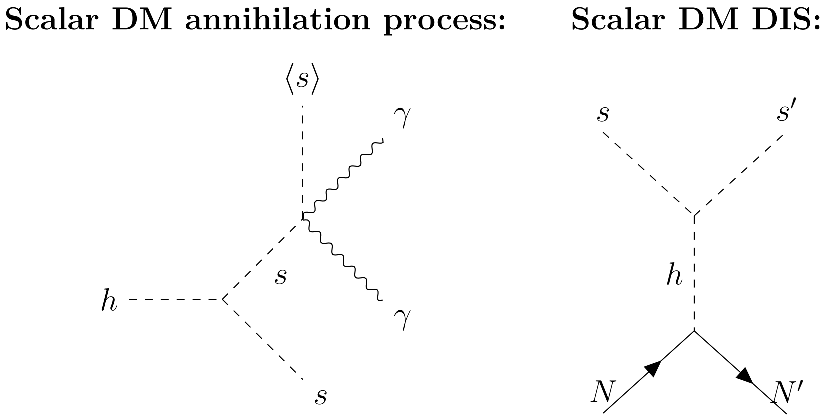


Table 2.5: Feynman diagrams depicting hypothetical scalar dark matter signal signatures.

for several assumptions, it is difficult to accurately predict the result of the process.

2.4 The SHiP experiment

2.4.1 Overview

The Search for Hidden Particles experiment, or SHiP, is expected to be exposed to a secondary proton beam from the CERN Super Proton Synchrotron facility (SPS) hitting a fixed target with a rate of up to 4×10^{19} protons a year[54]. The experiment is designed to maximise beauty and charm-meson production, the decay products of which could decay into hidden sector particles such as heavy neutral leptons.

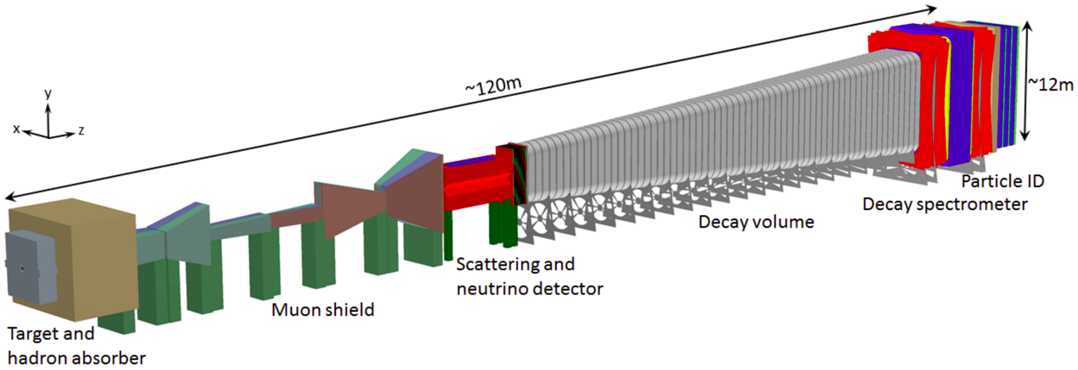


Figure 2.3: Overview of the SHiP experiment. Taken from [54].

Although similar fixed target experiments have been performed in the past, their exploration domain was limited to masses below 450 MeV in PS191[55] or to HNLs with couplings to active neutrinos $|U_{\mu_i}|^2$,$|U_{e_i}|^2$ above 10^{-7} (CHARM[56]). SHiP aims to accumulate 2×10^{20} protons on the target in 5 years of operation while keeping the background levels at approximately 0.1 events, achieving the required sensitivity to probe weakly interacting particles with couplings $|U|^2$ up to 10^{-10} (for HNL) and masses reaching 5 GeV[57].

The SHiP detector is proposed to be installed in the Experimental Cavern North 3 (ECN3): an experimental underground cavern at CERN, located at the end of the P42 beam line[58]. The overall layout of SHiP is presented in Figure 2.3 and can be divided into two main parts. The first section acts as a "frontline" that absorbs protons and reduces the accompanying SM background. It is made of the following components: a molybdenum-tungsten target, a hadron absorber, and the active muon shield or AMS that sweeps muons originating from the target out of detector acceptance. The second part consists of two detecting apparatuses: The scattering and neutrino detector, or SND, which focuses on the low mass dark matter sector; and the Hidden Sector Decay Spectrometer or HSDS for short. The HSDS is a system of the following components: a decay volume with a pressure of 10^{-6} bar where hidden particles are expected to decay; the straw tracker, which reconstructs the decay vertex kinematics; and the Particle ID, which consists of electromagnetic calorimeters and muon detectors. The Particle ID facilitates the recognition of particles that originate from decays and helps to differentiate between various hidden sector models[54].

Since SHiP is not expected to collect data until 2032[59][60], the SHiP data analysis at the time of writing (2025) uses simulated data within the FairShip simulation environment, which is based on the FairRoot software framework[61]. Primary proton collisions are generated using Pythia8[62], the particle interactions with surrounding material are simulated by GEANT4[63] and the neutrino interactions are modelled with GENIE[64].

2.4.2 Target complex

The SHiP target complex plays a vital role in the experiment, aiming to enhance the production of heavy mesons while reducing unwanted neutrino and muon backgrounds. Besides fulfilling physics requirements, the target must also manage energy dilution, provide effective cooling, and offer radiation shielding due to the high beam power, which can reach up to 350 kW and peak at 2.56 MW per spill. This makes the SHiP production target one of the facility's most demanding challenges.

The desired performance is achieved with a longitudinally segmented hybrid target composed of blocks with four interaction lengths (58 cm) of a titanium-zirconium-doped molybdenum alloy in the core of the shower, followed by six interaction lengths (58 cm) of pure tungsten. These blocks are interleaved with sixteen 5-mm slits for water cooling. To adhere to material limits on thermomechanical stress, the thickness of each slab, along with the positioning of each cooling slit, has been optimised to ensure uniform energy distribution and adequate energy extraction. The total dimensions of the target are 1.2 m long with transverse dimensions of 30 $\times$ 30 cm^2 to maximise shower containment.

The target pile is assembled within a compartment featuring dual walls. The inner chamber facilitates significant water circulation between the target slabs, maintaining a pressurised cooling system between 15 and 20 bars to elevate the boiling point. An anticipated flow rate is 180 m^3 per hour. The outer vessel offers a primary helium circulation around the inner container, creating an inert environment to diminish corrosion while also forming a protective layer that can detect any potential water leaks from the internal vessel. Connections and hooks are incorporated into the design to allow for full remote operation using the target area's crane.

2.4.3 Scattering neutrino detector

The SHiP neutrino target is designed to detect and reconstruct neutrino interactions using the Emulsion Cloud Chamber (ECC) technique, previously employed in experiments such as OPERA and DONUT.

The ECC structure alternates nuclear emulsion films with dense lead plates, ensuring precise charged-particle tracking while enhancing neutrino interaction probability. The target is modular and consists of individual units called *bricks*, each containing 57 emulsion films interleaved with 56 lead plates. These bricks provide micrometric spatial resolution, allowing the reconstruction of neutrino interaction vertices and the identification of secondary decay products. The ECC approach is particularly effective for detecting ν_τ interactions by identifying the short-lived tau lepton and its subsequent decay into electrons, muons, or hadrons.

Additionally, the target includes a Compact Emulsion Spectrometer (CES), which improves charge identification by tracking the deflection of charged particles in a magnetic field generated by the Goliath magnet and is crucial for distinguishing neutrinos from antineutrinos.

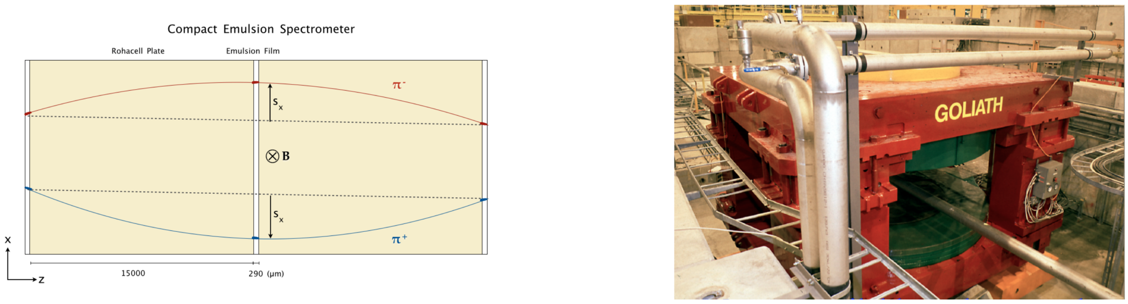


Figure 2.4: **Left:** Schematic representation of the sagitta measurement using the CES. **Right:** Photograph of the Goliath magnet. Taken from [65].

The modular target structure is arranged in multiple walls, interleaved with planes of electronic detectors that provide precise timing information and identify the specific brick where the interaction occurred. The ECC data is collected by a high-speed emulsion scanning system, which utilises advanced automated microscopes. This system is designed to handle the large volume of data expected from SHiP, allowing efficient extraction of neutrino interaction events.

2.4.4 Active muon shield

One of the biggest challenges facing the SHiP experiment is to achieve a low background environment. It is estimated that a second-long SPS spill contains, on average 4×10^{13} protons that will give rise to a cascade of hadrons such as π, K, ω and charm mesons at the target. The hadrons themselves can be blocked by a 5-meter-thick iron absorber; however, their decay products pose a more serious problem. In the case of a one-second spill, up to 10^{11} muons can be produced. Muons are particles that can easily penetrate materials such as iron and travel long distances with minimal energy loss. However, since they are charged, they can be diverted from the detector by using a magnet if the integral of the magnetic field over the muon path $\int B\,dl$ is sufficient.

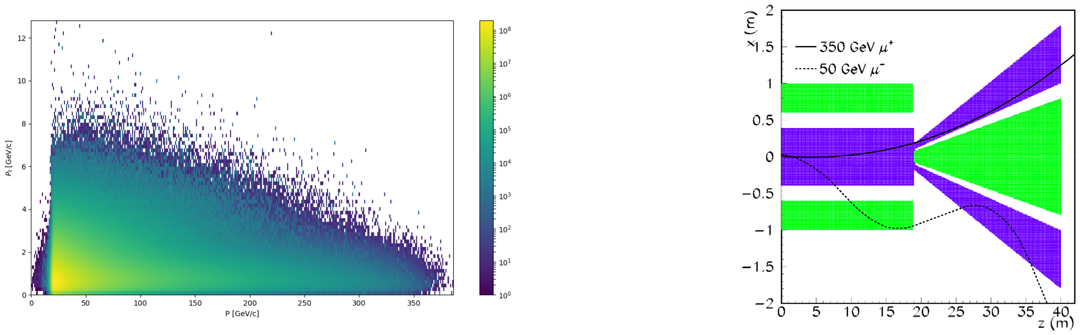


Figure 2.5: **Left:** Simulated distribution of transverse momentum versus momentum of muons weighted to one proton spill. **Right:** The illustrated principle of muon shield operation. Different magnetic polarities are indicated with blue/green colour. Solid and dotted lines represent muon trajectories of 350 GeV and 50 GeV, respectively. Taken from [66].

Unfortunately, the same magnet can have a different effect on leptons with lower momentum, causing them to bend back in the direction of the detector as they pass through the "return" field. To counteract this, a second magnet with opposite polarity can be added, thus deflecting the lower-momentum muon once more. This idea is not new: the DONuT experiment has used a two-magnet system to remove high-momentum muons from detector acceptance[67]. The SHiP experiment is expected to accumulate a much larger number of protons on the target than the DONuT (around 3.6×10^{17}), requiring more stringent requirements on the muon rate in SND and HS. This implies a more complex shield shape to deflect muons with a wide range of kinematics. Because of the complex physics involving muon propagation through the magnet shield, there is no analytical solution available that could predict how the shield will behave under certain experimental conditions.

2.4.5 Hidden Sector Decay Spectrometer

The Hidden Sector Decay Spectrometer is designed to search for FIPs by using two main components: a veto-protected, depressurised background-free environment for their decay and the particle detection and identification complex. Its layout follows the established schema in experimental particle physics: a charged particle tracker for trajectory reconstruction, followed by an electromagnetic calorimeter (ECAL) and hadronic calorimeter (HCAL) for energy measurements, and concluding with a muon detector to detect and identify the energetic muons. A detailed description of each detector component and its role in the HSDS follows below.

Upstream veto tagger

The Upstream Veto Tagger (UVT) is located between the tau neutrino detector and the vacuum vessel with the main goal of significantly reducing contamination from unwanted particles while preserving high efficiency for signal detection.

One of the primary roles of the Upstream Veto Tagger (UVT) is to reduce the backgrounds from neutral kaons produced upstream of the vessel down to negligible levels. Additionally, the UVT provides muon vetoing, preventing high-energy muons from entering the vacuum vessel through the front window and causing unwanted interactions. These backgrounds mainly arise from neutrino and muon interactions upstream, particularly in the passive materials of the tau neutrino detector. While the tau neutrino detector's muon system can detect some of these interactions, it is not optimised for vetoing, necessitating the dedicated UVT system. It also provides an additional layer of veto redundancy, ensuring the robustness of the hidden sector physics searches conducted in SHiP.

The UVT is constructed with plastic scintillator bars designed to span a 4×12 m^2 area. There are 120 scintillator bars organised in one horizontal layer. Each bar is 4 metres in length, 12 cm in width, and 1 cm in thickness, featuring wavelength-shifting (WLS) fibres for enhanced light collection. These WLS fibres extend along the entire length of each bar and are connected at both ends to silicon photomultipliers (SiPMs), which ensure optimal photon detection and clear signal conversion.

The electronics are housed upstream of the detector, ensuring easy accessibility for maintenance while minimising interference with the detectors active volume. The support structure is designed to minimise passive material, preventing the secondary scattering of muons back into the decay volume. Additionally, the detector's design ensures no material extends laterally beyond $|x| = 2$ m, preventing interference with the muon flux.

Due to the substantial muon flux in SHiP, it is crucial to employ a high-precision timing system to distinguish actual physics events from background noise. As such, UVT readout electronics are considered to facilitate rapid processing and exact timing measurements, achieving high detection efficiency (99.9%) and quick response times (≈ 1 ns resolution) with minimal dead time ($\approx 0.01\%$). With the capability to support a signal rate of up to 100 kHz, the UVT can manage high event rates without compromising performance, which is vital for sustaining low noise levels while ensuring a high rate of background rejection.

HSDS vacuum vessel

The HSDS vacuum vessel in the SHiP experiment plays a crucial role in ensuring a low-background environment for new physics searches. It is designed to be depressurised to 10^{-6} bar, minimising neutrino and muon interactions with residual air. The vessel integrates a liquid scintillator-based background tagger, providing nearly full solid-angle coverage to detect and reject background events.

The vacuum vessel features an elliptical cross-section (10 m vertical, 5 m horizontal) and spans 62 m in length, comprising a 50 m decay volume followed by a 12 m magnetic spectrometer. The elliptical shape helps deflect muon flux, focused by the upstream muon shield in the horizontal plane, away from the beam axis, improving background suppression. To withstand external atmospheric pressure (1 bar), the vessel has a ≈ 3 cm thick structural wall reinforced with transverse (1 m pitch) and longitudinal (1.5 m pitch) ribs, creating segmented cells for liquid scintillator containment. A finite element analysis has been conducted to optimise wall thickness and structural stability.

The vessels end lids consist of 8 mm thick aluminium membranes, welded to the

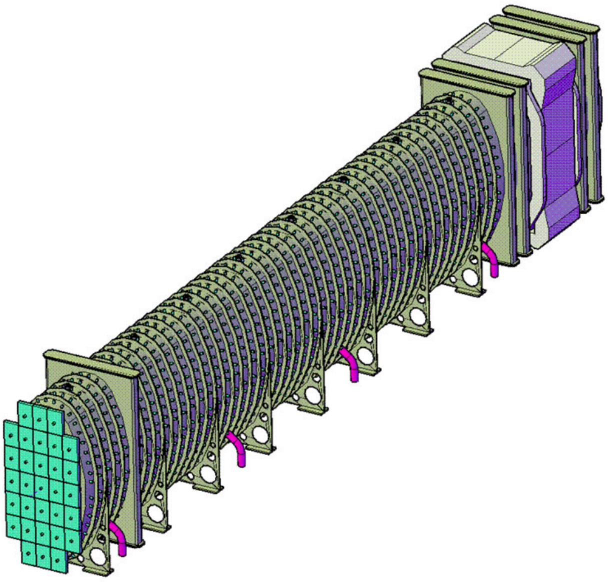


Figure 2.6: A layout of the baseline vacuum vessel. Taken from [65].

strong inner shell. Additionally, a 300 mm liquid scintillator layer in front of the entrance window enhances background tagging capabilities. Five large prismatic openings allow the insertion of tracking detectors without obstructing the decay volume. The first tracking station, located 5 m from the entrance, serves as a veto for residual background events, while four more stations downstream contribute to the magnetic spectrometer.

To attain the necessary low-pressure environment within the vessel, a vacuum system is equipped with oil-free screw pumps, Roots blowers, and turbomolecular pumps that utilise magnetic levitation. This setup ensures a 10^{-6} bar pressure level, equivalent to 0.1 Pa. The pumping sequence involves an initial evacuation (1-10 Pa) via screw pumps, followed by high-vacuum evacuation using turbomolecular pumps.

Surround Background Tagger

The Surround Background Tagger (SBT) is a critical component of the SHiP experiment, designed to detect and veto background particles entering the decay volume. It surrounds the vacuum vessel with a liquid scintillator-based detector, providing near-complete coverage to distinguish genuine signal events from background interactions. The tagger enhances the background rejection capability, allowing precise identification of muons and neutral particles. The SBT utilises liquid scintillators as the detection medium, benefiting from their high efficiency, low energy threshold, and fast response time (≈ 5 ns decay time). The scintillators consist of a solvent (LAB, PXE, or PC) and a fluor (PPO or PMP) to optimise light yield, transparency, and emission spectra. The LAB+PPO mixture is preferred due to its high light yield ($\approx 10{,}000$ photons/MeV), low auto-absorption, and industrial availability.

The SBT is embedded within the double-walled vacuum vessel structure, with 30 cm thick scintillator layers divided into 863 segments. Each section is instrumented with two large-area photodetectors, either traditional 8-inch PMTs or Wavelength-Shifting Optical Modules (WOM), which use wavelength-shifting materials to enhance photon collection. The WOM technology, originally proposed for IceCube, offers a cost-effective and modular alternative to large-area PMTs.

The SBT infrastructure includes storage, purification, and transport systems to maintain oxygen-free conditions, preventing scintillator degradation. The on-site system features nitrogen purging, stainless steel tubing, and hydrostatic pressure balancing to ensure uniform filling and operation.

The spectrometer tracker

The spectrometer tracker in the SHiP experiment is designed to reconstruct charged particle tracks from hidden particle decays with high efficiency while effectively rejecting background events. It plays a crucial role in momentum measurement, vertex identification, and event selection, ensuring that only true signal events are considered. The spectrometer rejects background by leveraging:

- Vertex reconstruction: ensures tracks originate from hidden particle decays.
- Invariant mass and flight direction constraints: suppresses random track combinations.
- Timing measurements from tracking detectors: assists in eliminating the combinatorial background.

The spectrometer consists of a large aperture dipole magnet and four tracking stations, arranged symmetrically around the magnet to provide optimal coverage. The fifth veto station is placed just downstream of the vacuum vessel entrance lid to reject upstream-generated background tracks.

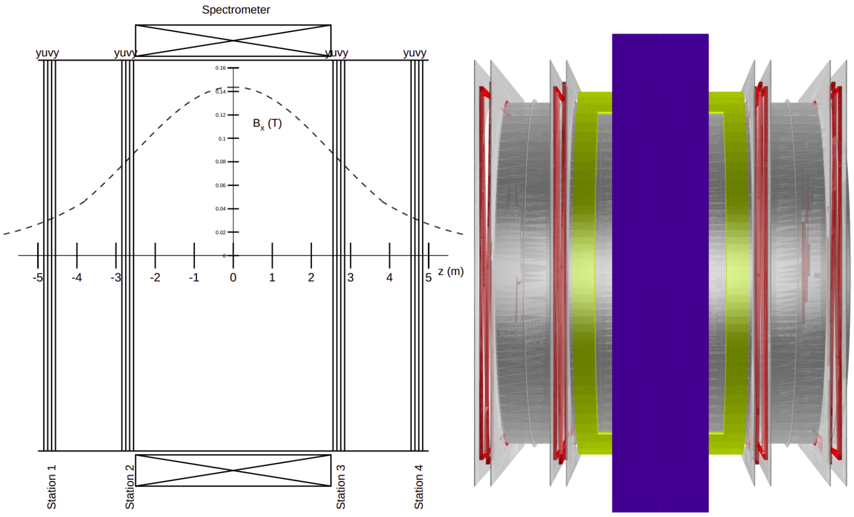


Figure 2.7: Spectrometer layout. **Left:** Position of the tracking stations (each with four views) and dipole magnet overlaid with magnetic field component B_x as a function of z. **Right:** 3D view of the spectrometer as implemented in the FairShip simulation. Taken from [65].

The magnet has a horizontal gap of 5 m, a height of 10 m, and a length of 5 m, generating a magnetic field of up to 0.14 T at its peak. The field integral between the second and third tracking stations is approximately 0.65 Tm, enabling precise momentum measurements.

The tracking stations are separated by 2 m, with a five-metre gap between stations 2 and 3, ensuring a sufficient lever arm for momentum measurement. Each tracking station consists of four measurement views (Y-U-V-Y). Y views measure the vertical coordinate, and U and V stereo views are rotated at an angle $\pm\theta_{stereo}$, allowing for transverse x-coordinate measurements.

The system utilises straw tube tracking technology, a reliable method for precise tracking in low-pressure settings, effectively providing high-resolution measurements while minimising material interactions. Each straw tube tracker consists of very thin polyethene terephthalate (PET) straws to reduce multiple scattering, each with a length of 5 metres and an outer diameter of 9.83 mm, featuring a 30 μm gold-plated tungsten wire centred for charge collection. An essential design feature is ensuring gas-tightness, as the system operates in a vacuum to reduce neutrino interactions.

The readout electronics for the straw tube tracker are designed for high-speed data processing with minimal power consumption. The system includes:

- Front-End (FE) boards for signal amplification, shaping, and discrimination.
- CITIROC ASICs or FPGA-based digitisers for hit timing and signal digitisation.
- High-speed Ethernet-based DAQ system for efficient data transmission.

On average, each straw provides a signal gain of several 10^4, with a drift time resolution anticipated to be around 3-4 ns. This allows for precise tracking, with simulations conducted using GEANT4 and FairShip, indicating a momentum resolution of less than 1% for momenta below 50 GeV.

Calorimetry

The SHiP calorimeter system is a high-precision, cost-effective solution for particle identification, energy measurement, and background suppression. It is made of an electromagnetic calorimeter (ECAL) and a hadronic calorimeter (HCAL). These calorimeters ensure precise electron, photon, and pion detection and complement the tracking system for accurate event reconstruction.

The ECAL, positioned right after the timing detector, is optimised for electron/photon energy measurements in the 0.370 GeV range, achieving an energy resolution better than 10%. It is essential for correctly identifying electrons and pions in order to separate the anticipated signal decay $HNL \to e^{\pm}\pi^{\mp}$ from the main background channels such as $K^0_S \to \pi^+\pi^-$ and $K^0_L \to \pi^+\pi^-\pi^0$. The ECAL employs a so-called "shashlik" technology, consisting of lead-scintillator sampling layers with wavelength-shifting (WLS) fibres and fast photodetectors. The elliptically shaped ECAL matches the tracking systems acceptance, ensuring optimal efficiency.

The HCAL, placed immediately after the ECAL, is designed to separate pions from muons, particularly in the low-momentum region ($p < 5$ GeV), where standard muon detectors may have limited efficiency. It consists of sampling modules with alternating layers of iron absorbers and polystyrene scintillators, arranged in two longitudinal sections for improved pion/muon separation. The first section (H1) contains 18 sampling layers, while the second section (H2) has 48 layers, maximising pion suppression while maintaining high efficiency for muons. The total calorimeter length is about 3.3 metres, corresponding to approximately 6.2 hadronic interaction lengths (λ_I).

Both calorimeters share a common readout system, featuring fast ADCs, FPGAs, and gigabit Ethernet links for real-time data processing. The front-end electronics are optimised for precise timing (ns-level resolution) and energy extraction. To ensure long-term stability and accuracy, an in-situ calibration system is employed using cosmic muons and decaying in-flight electrons, allowing continuous adjustments to maintain detector performance.

Muon detector

As implied by its name, the purpose of the SHiP muon detector is to identify muons from signal decays while efficiently rejecting background events, particularly those involving pion misidentification. It works in coordination with the electromagnetic (ECAL) and hadronic (HCAL) calorimeters to enhance muon-hadron separation. The system is designed to detect muons with high efficiency ($\approx 99\%$) and minimise pion misidentification below 0.1% over a wide $\approx$ 3-100 GeV momentum range. The latter is a key challenge, mitigated through a combination of calorimeter energy deposition analysis and precise hit pattern recognition in the muon system. The system is also capable of operating in veto mode, rejecting hadrons by ensuring they leave no hits in the muon layers.

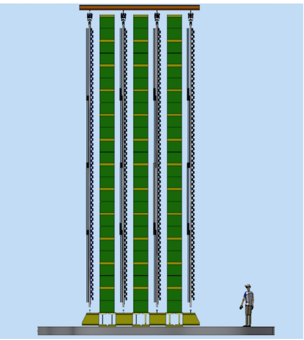


Figure 2.8: Muon detector layout: green thick layers are the passive iron filters and grey thin layers are the active modules. Taken from [65].

The detector consists of four active tracking stations interleaved with three iron filters, ensuring that only high-energy muons penetrate all layers while hadrons are absorbed. Each station spans an area of 6×12 m^2, providing large-area coverage. The calorimeter system contributes 6.7 nuclear interaction lengths (λ_I), while the muon filters add another 10.2 λ_I for further particle differentiation.

The active layers use extruded plastic scintillator strips with wavelength-shifting (WLS) fibres and SiPM readout, a robust and cost-effective technology employed in neutrino experiments like MINOS, T2K, and Belle II. The detector includes 3,840 strips (510 cm wide, 3 m long, 12 cm thick) arranged in horizontal and vertical orientations, ensuring precise tracking. Each scintillator bar is read out at both ends, enhancing light collection and timing accuracy. The total system comprises 7,680 SiPMs, ensuring high granularity and redundancy.

The data acquisition (DAQ) system processes approximately 3.5 MB/s of zero-suppressed data. The readout electronics consist of 7,680 channels, grouped into 240480 boards, with signals transmitted through 20 Ethernet links. The system employs SiPMs from manufacturers like Hamamatsu, KETEK, and SensL, with ongoing R&D to optimise bias voltage control, temperature compensation, and calibration using pulsed LEDs.

Trigger and the data acquisition

The SHiP DAQ system is designed to handle the triggerless readout of all sub-detector data, enabling real-time event processing. Unlike traditional trigger-based systems, SHiP implements a fully software-based trigger on an Event Filter Farm (EFF), where all raw detector data is sent via a high-speed Ethernet network and processed in real-time. The system architecture is inspired by the LHCb trigger upgrade but benefits from lower data rates and the ability to use commercial networking hardware instead of radiation-hardened custom components.

The Timing and Fast Control (TFC) system generates the clock and synchronisation signals, ensuring consistent data acquisition across all sub-detectors. Each front-end (FE) module digitises signals and sends them in time-stamped frames (100 ns slices), which are grouped into Multi Event Packages (MEPs) before being transmitted to the EFF for event reconstruction. The EFF performs real-time track reconstruction and event selection, reducing the data volume before storing events in the CERN central archive. The DAQ system is designed for 5 Gbps bandwidth, handling a 100 kHz event rate with only three high-performance servers.

The Experiment Control System (ECS) oversees detector configuration, monitoring, and run control, integrating slow control (hardware monitoring) and fast control (DAQ operations). A separate monitoring farm (MF) provides real-time feedback on data quality. The SHiP trigger software runs on the EFF, using advanced C++ reconstruction algorithms to identify physics events while minimising background. The system is designed to store ≈ 1 PB of data per year, with an emphasis on efficient processing and minimal dead time, making it a highly scalable and cost-effective DAQ solution for long-term operation.

2.5 SND@LHC experiment

2.5.1 Introduction

SND@LHC is a compact and standalone experiment at CERN designed to study neutrinos produced at LHC. The experiment is stationed in the unused T118 tunnel located 480 metres downstream of the ATLAS interaction point (IP1). This location enables the reduction of the SM background from IP1 and to probe the high-energy neutrinos in the pseudo-rapidity ranges that are not accessible to existing experiments such as ATLAS, CMS and LHCb.

The first official expression of interest was submitted by the SHiP collaboration to LHCC in February 2020[68], followed by the letter of intent[69] in August of the same year. The technical proposal[70] was submitted in February 2021, which was quickly followed by the installation of the detector on the T118 during the Long

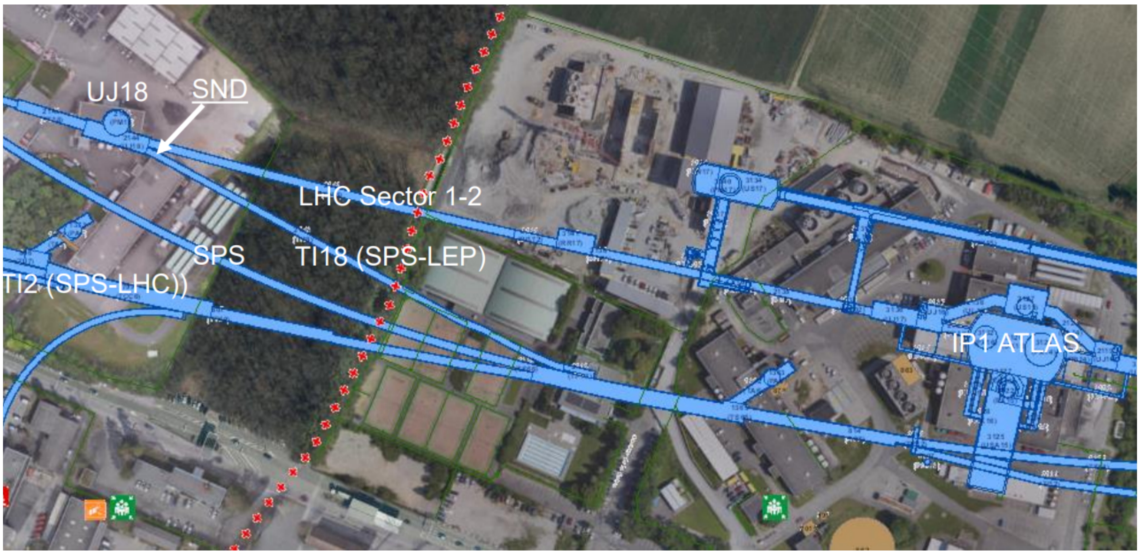


Figure 2.9: Overview of the SND@LHC experiment. Taken from [23].

Shutdown 2. It started accumulating data in April 2022, and the first observation of eight neutrino CC interaction candidate events was reported in May 2023[71]. LHC Run 3, which is scheduled to take place between 2022 and 2026, is expected to produce a total of 300 inverse femtobarns of data and detect up to ≈ 400 neutrinos.

In addition to the SND@LHC, there is another experiment with similar goals and design called FASERν[72]. The primary distinction between these projects is the position of the detectors in relation to the IP1 point: FASERν is located along the beam collision axis in tunnel T112, allowing it to explore the pseudo-rapidity range $\eta > 9$, which is mainly composed of neutrinos from pion and kaon decays. On the other hand, SND@LHC is located off-axis with a pseudo-rapidity range of $7.2 < \eta < 8.6$, making it more sensitive to neutrinos produced in heavy-flavour decays, such as charm. In addition to the location aspect, the FASERν target has a mass of 1.2 tons, which is higher than that of SND@LHC (810 kg), resulting in a significantly higher neutrino yield[73].

The two experiments, by taking measurements in different η regions, allow a more accurate assessment of machine backgrounds and systematic errors in the calculation of neutrino fluxes. This is possible because both experiments use the same simulation tools, which makes it possible to directly compare the results obtained in neutrino studies.

2.5.2 Detector layout

The SND@LHC is a hybrid device that combines two distinct technologies: nuclear emulsions and electronic sensors. The nuclear emulsion part plays the role of the high-resolution vertex detector, which can reconstruct charged particles with micrometric resolution. The electronic detectors complement the emulsion chamber by providing additional information about neutrino interaction vertices, such as the hadronic and electronic shower energy, timestamp of the events, and whether a muon has been emitted in the process. This configuration enables the experiment to identify all three neutrino flavours and detect the presence of FIPs.

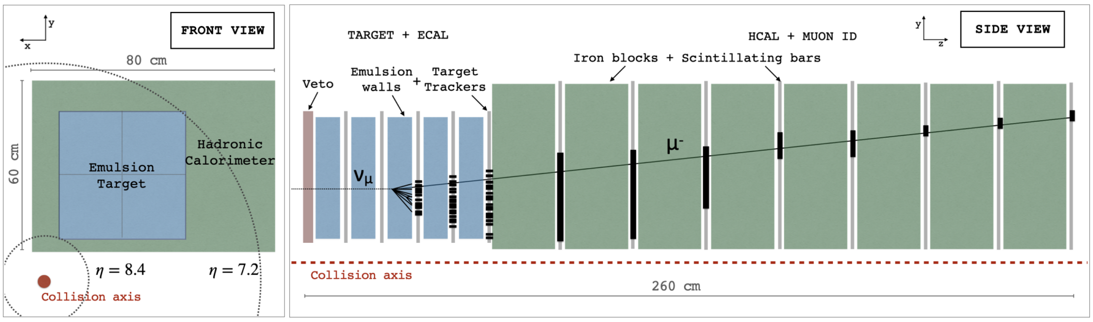


Figure 2.10: The frontal (left) and side (right) projection of the SND@LHC detector schematic layout. Taken from [71].

The concept is not new and is heavily based on the designs of OPERA[74] and of the SND detector at SHiP[23], from which some of the hardware elements, such as SciFi target trackers and tungsten plates, have been reused. This strategy has helped reduce the overall detector cost and keep the installation and commissioning of the apparatus on schedule.

The modular configuration of the SND@LHC detector allows its emulsion and electronic systems to operate independently, which has enabled the SND@LHC collaboration to detect muon neutrinos by using electronic detectors alone. Functionally, the SND@LHC detector can be split into three parts: the veto system, the target, and the hadronic calorimeter combined with a muon detection unit.

Veto system

The Veto system consists of two parallel planes with a gap of 4.3 cm. Each plane is made of seven $1 \times 6 \times 42$ cm^3 stacked EJ-200 scintillating bars that are read at both ends by silicon photomultipliers or SiPMs.

As the name suggests, the veto system detects and tags incoming charged particles originating outside of the detector so that their tracks can be rejected at the data acquisition stage. This is done to suppress background from the muons originating from IP1 so that they do not contaminate the experimental detection channels involving muon track signatures.

Although the rock surrounding the T118 tunnel filters most of the external background, muons, due to their high permeability, are highly abundant in detector acceptance with expected fluxes of up to $\approx 2 \times 10^4$ cm^{-2} fb. Therefore, to achieve a zero background environment in the SND@LHC detector volume, the overall efficiency of the veto system should ideally be equal to 100%. In practice, there is always a level of small inefficiency: in 2022, its value was estimated to be around 4.5×10^{-4}[71], sufficient for the physics goals of the experiment.

Target

The target section consists of five 390×390 mm^2 walls with an overall mass of 830 kg. Each wall is made of four Emulsion Cloud Chambers (ECC) followed by a scintillating fibre (SciFi).

Each ECC unit is made of 60 nuclear emulsion films interleaved with 59 one-millimetre thick tungsten plates. The tungsten plates act as the target material, where the incoming neutrinos participate in the CC and NC interactions with the constituent nuclei.

SciFi is an important part of the detector as it complements the emulsion system with additional time and spatial information. It also acts as the sampling hadron and electromagnetic calorimeter, enabling energy measurements of deposited showers. Each SciFi station is made up of two 40 × 40 cm^2 planes with alternating x and y views. Each view consists of six densely packed, staggered layers of Kuraray SCSF-78MJ scintillating fibre glued together. As the particle crosses the x- and y- views, it produces light which is read out by a pair of SiPM arrays.

Combining the information from these two subsystems allows reconstruction of the path of the charged particles with a resolution of the order of $\approx 150\ \mu\text{m}$.

Hadronic calorimeter and a muon detection system

The main purpose of this detector part is to identify muons and measure the energy of the hadronic jets. It can be divided into two parts: the first five stations or *upstream* (US) are used as a particle timing detector, and the last three stations or *downstream* (DS) provide muon tracking information.

Each US station is made of three stacked iron blocks FT822 ($80 \times 20 \times 20\ \text{cm}^3$) that act as passive material. Each set of blocks is accompanied by a scintillating detector plane. Constructionally, they are similar to the veto system: it is a set of ten stacked horizontal $82.5 \times 6 \times 1\ \text{cm}^3$ scintillator bars, each read out at both ends by eight Hamamatsu S14160-6050HS SiPMs.

The DS part of the detector also uses the same iron block configuration as the US but has two layers of thinner scintillating bars: one with horizontal alignment and one with vertical alignment. This configuration is capable of providing tracking information, although with a coarse resolution of ≈ 1 cm. Because the frontal cross-section of the DS has a rectangular shape, the dimensions of the scintillating bars are different. The horizontal planes consist of 60 bars each measuring $81 \times 1 \times 1\ \text{cm}^3$ each read out by a single Hamamatsu S14160-6050HS SiPM on both sides, and the vertical planes consist of 60 bars of $60 \times 6 \times 1\ \text{cm}^3$ with each bar read out by a single SiPM located at the top.

2.5.3 Software

The software components of the SND@LHC experiment can be divided into three different categories: detector control systems, data acquisition or DAQ, and offline analysis.

Detector control system

The detector control system, or DCS, monitors the environmental parameters of the detector such as the temperature and humidity of the emulsion detector and the current flow of electronic components.

Although the DCS does not produce any experimental data, it plays a crucial role in maintaining the optimal environmental conditions of the SND@LHC detector that affect the volume and quality of the taken data. Certain components, such as emulsion films, must be kept at temperatures below 15 degrees Celsius to avoid deterioration effects. This is achieved through the control of complementary "life-support" subsystems such as the chiller and evaporator, and safety interlocks that disable the SciFI electrical system in case of overheating.

Data acquisition

The DAQ is responsible for collecting signals from the electronic sensors of the detector and converting them into a digital form that can be recorded and stored as structured data. This is quite a complex process that involves many physical components, which are listed below:

- 36 DAQ boards that are based on the Mercury SA1 module from Enclustra. They read and digitise the signals from SiPMs and send the result to a server.
- The Trigger Timing Control (TTC) crate is responsible for connecting with the Beam Synchronous Timing (BST) system. It obtains data from the BST, such as the bunch crossing and revolution frequency of the LHC, the GPS absolute time, and the beam parameters. This information is used to synchronise the DAQ boards and add meta-information to the recorded physics data regarding the conditions of the LHC run.
- The dedicated DAQ server[4] receives all recorded hits from each DAQ board and groups them into data clusters called *events*. The online software noise filter is used to assess events based on certain topological criteria, such as the hit distribution across the detector plane. Events that pass the selection are recorded in ROOT[75] format on a local disk and eventually transferred to EOS[5], from which they can be used for further data analysis. The number of events per second that get recorded varies and depends on the LHC instantaneous luminosity: in 2022 it has been reported[71] to be 5.4 kHz.

It should be noted that DAQ only deals with information from electronic systems and does not interact with the emulsion detector. The ECC accumulates the data passively by being exposed to charged particles and the formation of silver halide in the emulsion plates. This necessitates a different offline digitisation process, which will be discussed in the following section.

Offline Analysis

Due to the complex nature of the experimental setup of SND@LHC, numerous software packages are required to perform various tasks such as simulation, reconstruction, and high-level analysis of observed events.

Many of these have been developed and enhanced in recent years, forming robust software frameworks that can be used for a variety of experiments. These can be classified into three categories: simulation frameworks, event reconstruction software, and data analysis tools.

---

[4] Located at building 2175 at CERN.

[5] EOS is a disk-based, low-latency storage service used in CERN for data storage.

Simulation frameworks are essential in experimental particle physics, as they enable the modelling of a variety of experimental elements, such as the expected Standard Model and instrumental backgrounds, as well as the corresponding detector response. Most particle physics detectors initially exist as virtual constructs which can undergo years of rigorous performance studies and optimisation before their physical construction. The simulation ecosystem used in the SND@LHC experiment consists of:

- The DPMJET[76] event generator is used to simulate the LHC process $p+p \to \sum_i^N X_i$ where N number of decay products X_i have been generated.
- The FLUKA simulation framework emulates the passage and energy loss of the decay products from IP to the T118 tunnel, since the geometry and chemical composition of the penetrated medium are known a priori. The main result of the simulation is the expected rate of incoming particles and their spectra in the T118 tunnel.
- Using the results of the previous step, the Genie[64] simulates neutrino interactions with the detector material. In addition to Genie, particle guns using Pythia[62] are used to generate specific background-producing events.
- The Geant4 toolkit is used to simulate the movement of particles in the SND@LHC and their interaction with the detector components.

For convenience and consideration of the specific details of the SND@LHC, the last two steps are handled by the *sndsw* toolkit[77], which includes the relevant software libraries and the virtual geometry of the detector.

The final output of the simulation steps is produced and saved in a ROOT format similar to that during DAQ: the only difference is the additional truth information generated by the simulation. No additional preparation is required for the data from the electronic detectors, regardless of whether they are from DAQ or simulation, since the events have already been reconstructed and are ready for analysis.

As with most of the experiments at CERN, the bulk of the data analysis and visualisation can be performed using the ROOT data analysis framework. The SND@LHC collaboration uses its derivative: the sndsw toolkit, which is based on the FairRoot software framework[61][6].

2.5.4 Emulsion vertex detector

Nuclear emulsion

The SND@LHC vertex detector is composed of a nuclear emulsion, a technology that has been around for a long time. This section will provide a brief review based on [78, 79, 80] of the emulsion technology, its principles, evolution, shortcomings, and advantages that provide a valuable context for subsequent research.

The composition of the nuclear emulsion is similar to that of the film used in commercial photography and consists of silver halide crystals immersed in gelatin. The

---

[6] FairRoot itself is a derivative of the ROOT Data Analysis Framework programme developed at CERN.

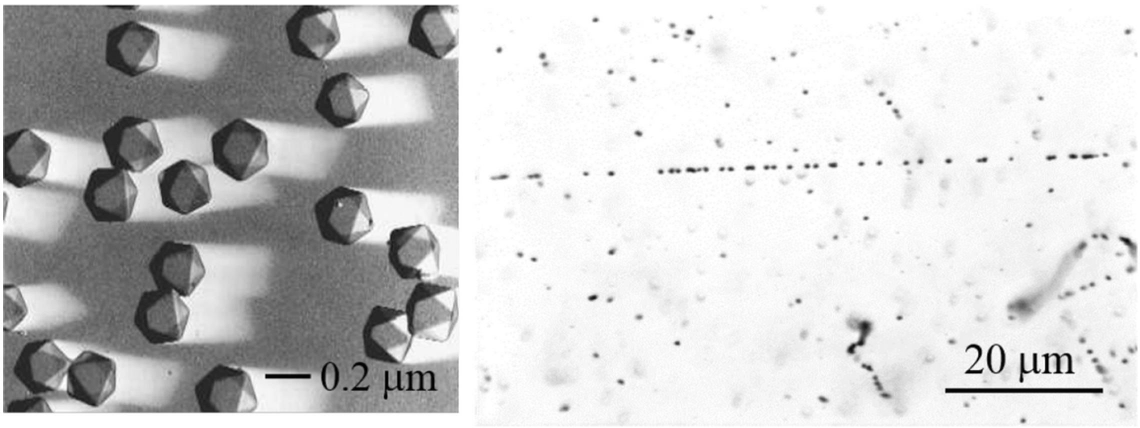


Figure 2.11: Left: Silver bromide crystals as seen with an electronic microscope. Right: An example of a latent image of the track left by the ionising particle. Taken from [79].

exact type of silver halide varies from film to film depending on the requirements. Typically, nuclear emulsions in particle physics are made of silver bromide (AgBr) with a small fraction of iodine. The AgBr crystal functions as a semiconductor, featuring a band gap of approximately 2.7 eV. When a charged particle travels through the crystal, it excites electrons from the valence band to the conduction band. Through the artificial chemical process known as *sensitisation*, the AgBr crystal acquires a positively charged sensitisation centre that serves as a shallow 21-25 meV trap, inhibiting electrons from returning to the valence band. Once the electron is captured, the sensitisation centre acquires a negative charge, which then attracts migrating positively charged interstitial silver ions. These ions interact with the trapped electron through the chemical ionic process $Ag^+ + e^- \rightarrow Ag$. After the sensitisation centre donates the electron, it regains its positive charge and is ready to trap another electron.

This process can be repeated multiple times until an aggregate of multiple silver atoms is formed on the crystal. A cluster of four or more silver atoms is considered *developable*, and the sensitisation centre at this stage forms a so-called *latent image centre*. These chemical changes are too subtle to be seen with the naked eye; however, amplification of the signal can be achieved by a development procedure. It involves immersing the emulsion in a solution consisting of a reduction chemical that interacts with the latent image sites through the reaction $\mathrm{Red} + n\mathrm{Ag}^+ \Leftrightarrow n\mathrm{Ag} + \mathrm{Ox} + m\mathrm{H}^+$, where Red and Ox are the developing agent and the oxidised developing agent, respectively; n is the number of ions and m is the number of hydrogen ions produced. This is a discriminatory reaction that mainly acts on the latent image sites, leaving the silver halide crystals that were unaffected by the passing charged particles intact. These are removed by the fixing procedure, leaving the metallic silver deposits alone.

Due to the high power $\approx 10^8$ of development amplification, the silver grains are large enough to be observed with the optical microscope. Each silver halide crystal has a limit of detection efficiency of $\approx 17\%$, well below 100%, which requires a higher ratio of silver to gelatin and finer silver halide crystals compared to commercial photographic films.

Despite the need for chemistry expertise to create and develop emulsions, it was a simple enough process to be employed as far back as the 19th century to discover radioactivity [81]. The ease of use of the photographic technique made it a desirable choice for particle physics experiments conducted in the mid-twentieth century studying cosmic rays at higher altitudes (mountain, rocket, and air balloon experiments), which resulted in the discovery of particles such as π and K mesons.

As particle physics entered the era of particle accelerator experiments in the second half of the century, emulsion detection technology started experiencing a gradual decline in usage in favour of electronic detectors. This was for numerous reasons:

- Absence of timing data: unlike the majority of electronic detectors, emulsions cannot assign timestamps to each track. Instead, they integrate all tracks during an exposure into one dataset, complicating the disentanglement of physics events during data analysis.
- Effect of the emulsion on the charged particle path: the emulsion is a dense material that significantly influences the path of the charged particles due to the effects of Coulomb repulsion. This particularly affects the trajectory of the low-momentum particles, making them highly nonlinear and difficult to reconstruct.
- The handling of the emulsion requires a level of care that includes humidity and the temperature regime. Incorrect handling can cause various unwanted effects, such as *fog*, which is caused by false activation and crystallisation of silver halides, leading to unwanted noise in the developed film.
- Initially, the emulsion detector devices were made of a single film plate, and to reconstruct the whole event, the emulsion had to be sufficiently thick. In the 1940s-1950s numerous attempts were made to increase the thickness of emulsion from a typical 50-100 μm to 500-1000 μm with some exceptional attempts to create 2000 μm thick films. This was still insufficient to contain the entire track of the particles, and the increase in thickness led to difficulties in scanning at the processing stage.
- Scanning the emulsion was a time-consuming and laborious task. This was bearable in experiments with low intensity that involved the observation of rare cosmic ray events, but in studies with higher luminosity from accelerators, this has become a major impediment.

However, the work on emulsion technology did not stop entirely, and a significant amount of effort has been spent to overcome at least some of these challenges. The first major breakthrough[82] that occurred in the 1950s was a solution to the film thickness problem: the Emulsion Cloud Chamber detector or ECC for short. Instead of increasing the thickness of the emulsion, a set of emulsion plates is interleaved with passive material layers, usually made of plastic or metal plates. By orienting the box so that the emulsion films are perpendicular to the incoming particles, it is possible to reconstruct their complete path by piecing together their imprints in consecutive emulsion films. In combination with the high crystal granularity of the constituent emulsion films, this technique has allowed experimenters to achieve an exceptionally high tracking resolution down to 1 μm.

The second significant milestone was the development of the tomographic read-out technique by the Nagoya group in Japan in 1974[83]. The key idea of the procedure is to sample multiple images of the same emulsion layer using an optical microscope, with each shot having a different focal plane depth. These snapshots can then be superimposed to produce a mini 3-D image of grain formations in the layer where grain clusters are subject to the track fit. The ones that pass the detection threshold are reconstructed as tracks with their attributes, such as fit quality and slope angle.

The main principles behind this technique are still used today; however, many aspects of the digital technology boom, such as CCDs and faster electronics, have greatly improved the speed and quality of scanning systems. For example, one of the earlier track selector systems, TS(1983), could scan around 0.03 cm^2 of emulsion per hour, while in 2015 the HTS system achieved a scanning speed of 4700 cm^2/h, which represents a factor of ×150k improvement over the first generations[79].

Although nuclear emulsion has never become a mainstream particle physics detector technology, it is still widely used in physics experiments that require a particularly high spatial event resolution, which is beyond the capability of electronic detectors. Neutrino experiments are one of the main benefactors of this technology, as ECC-based detectors are well suited to reconstruct short decay topologies involving short-lived particles such as charm mesons and tau leptons. Due to these qualities, nuclear emulsions were instrumental in the first observation and study of tau neutrinos in experiments such as DONuT and OPERA and, for the same reason, play an important role in the SND@LHC project.

Detector layout

The emulsion detector in the SND@LHC consists of five slightly separated walls, each with a thickness of approximately 78 mm. Originally[69], each wall was envisioned to host a single ECC unit with a surface area of 400 × 400 mm^2 to maximise the acceptance of incoming neutrinos. Eventually, this idea was discarded to accelerate detector construction and avoid potential problems: nuclear emulsion films of this large size have never been produced and would require additional R&D and validation efforts. Instead, the collaboration has opted to use four smaller ECC cells with an area of 192 × 192 mm^2 each, making a total of 20 ECC units per detector. This setup leads to a decrease in the detection area between adjacent cells, but it is more feasible in terms of implementation.

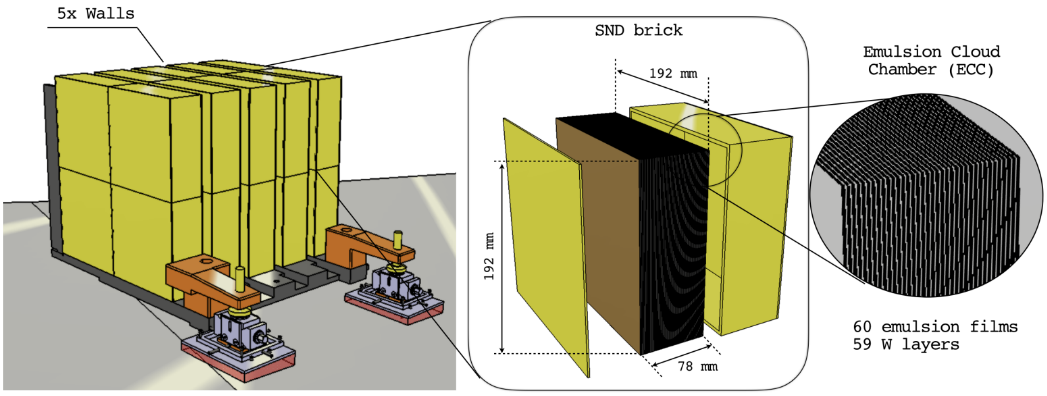


Figure 2.12: The layout of the SND@LHC emulsion detector. Taken from [70].

Each ECC brick is a highly evolved version of the ECC detector described previously and is made of sixty 310 μm thick emulsion films interleaved with 59 one-millimetre thick metal layers made of tungsten. The tungsten plates act as an individual high-density target, maximising the neutrino/FIP interaction rate per unit of volume, whereas the emulsion films record the passage of the charged decay products. The emulsion film itself is a composite structure made of two 70 μm thick emulsion layers separated by the plastic base.

This setup makes ECC a precise tracking device with sub-micrometric spatial and milliradian angular resolution. In addition to the tracking capabilities, the ECC as a whole acts as a fine sampling electromagnetic calorimeter well adapted to measurements of electromagnetic showers.

To protect the emulsion films from light exposure, humidity, and other environmental factors, the ECC is assembled inside the aluminium box, which is then tightly sealed before the brick is installed in the SND@LHC detector.

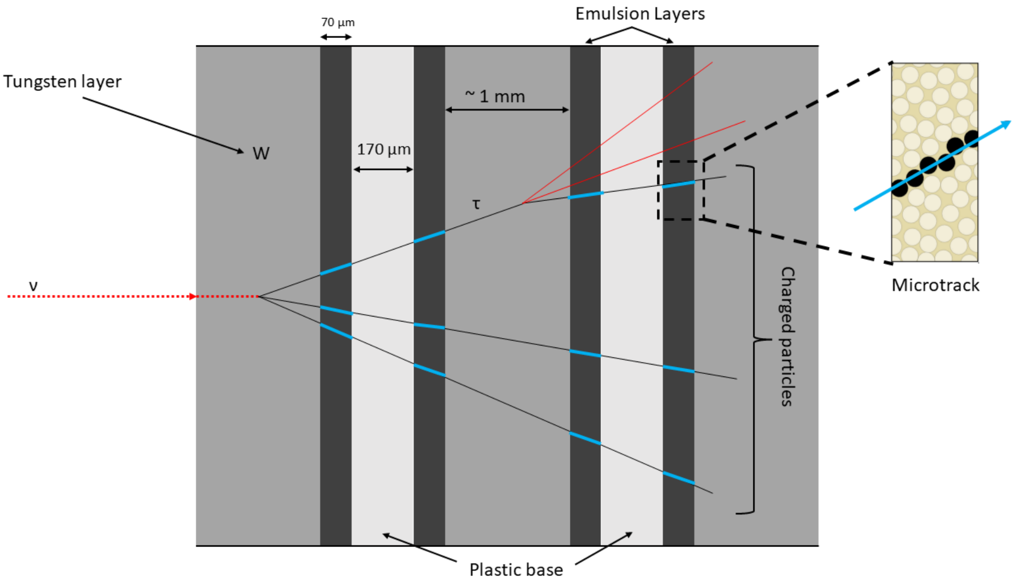


Figure 2.13: An illustration of the emulsion brick element. Thin red and grey lines represent the trajectory of the tracks and thick blue lines indicate the microtracks that end up in the raw data.

Data acquisition process

As mentioned above, the acquisition of data from the emulsion detector follows a different protocol from that of electronic detectors and is a complex procedure made up of multiple steps.

Extraction: the ECC units are ready for extraction once they are exposed to a sufficient flux of incoming particles. The SND@LHC technical proposal postulates a replacement every 25 fb^{-1}; this must occur during the scheduled LHC technical stop when the radiation levels in the T118 tunnel are safe for maintenance personnel. After extraction, the emulsion films are sent to the CERN dark room facility for chemical treatment.

Chemical development: the chemical development of the SND@LHC emulsion films is carried out in the designated dark room at CERN that has all the required facilities, such as development tanks, climate control, and chemical disposal. Film development itself consists of five steps:

1. Development: attempt to convert the silver clusters into visible metallic silver grains. For this purpose, a Fujifilm developer solution that was originally customised for the OPERA experiment is used.
2. Stop the development with acetic acid once it is deemed sufficient.
3. Fix: remove all remaining residual silver halide crystals with the custom Fujifilm fixer[7].
4. Wash: remove all silver thiosulfate complexes from the emulsion with water.
5. Thickening: restore the emulsion film to its original thickness with the use of glycerine and the Fujifilm Driwell solution.

Once the films have developed, they are ready to be transported to a scanning laboratory for microscopy.

Emulsion readout: there are four stations dedicated to scanning the SND@LHC emulsion plates: Bologna[84], Lebedev Institute[85], Napoli[86] and CERN. Each of these stations employs a microscope equipped with a computer-controlled monitored stage, which can move along both the x- and y-axes.

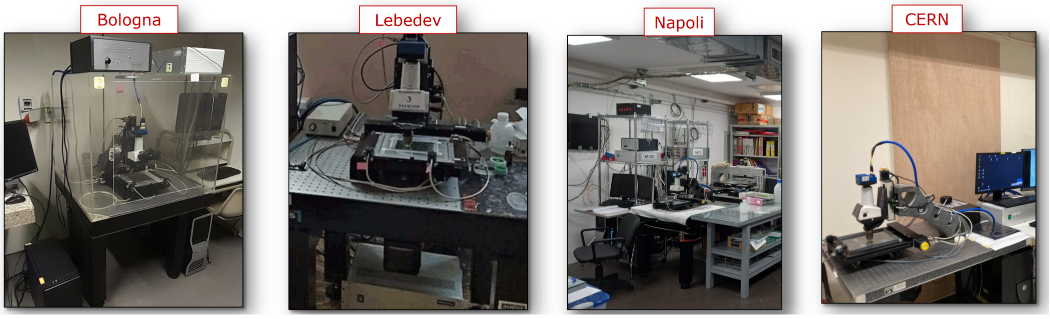


Figure 2.14: Scanning stations distributed across four locations.

In addition to the horizontal movement, the microscope can adjust the focal plane of the objective lens through the thickness of the whole emulsion, which enables the creation of a sequence of tomographic images of each field of view at equally spaced depth levels.

To make the optical path homogeneous in the film, the lens is immersed in oil with the same refraction index as the emulsion. These sequences are processed and analysed by the vision processing board in order to recognise aligned clusters of dark silver crystals produced by charged particles along their trajectories, which are reconstructed as micro-tracks.

The final step of the emulsion readout is to find suitable pairs of microtracks from two different emulsion layers within a single plate that could be associated with a path of the same particle. These pairs are generally referred to as *hits* or *base tracks*. In contrast to single point-like pixels in electronic detectors such as ATLAS, the hit

---

[7] Fujifilm UR-F1

in the context of ECC detectors is a more complex object. It is represented by the vector in the 5-dimensional space that can be defined by five variables: x, y, z, tx, and ty. The first three elements are the spatial coordinates of the point at which the base track crosses the middle of the plastic base. The last two variables are used to parameterise the change of the x and y coordinates of the base track line with respect to the z-axis.

Along with the geometrical information, an additional set of metadata is recorded for each base track, such as the total number of micro-track clusters, pixel intensities, and the chi-square goodness of base-track line fit. The total scanning speed depends on the laboratory but is expected to reach 180 cm^2 per hour[70].

Emulsion film alignment: once all plates of a given ECC brick have been scanned, the output data must be assembled to build a three-dimensional data model of the hits. Because each emulsion plate should have the same thickness and size, its base track sets are arranged in the same order as during the data-taking process in the target.

Unfortunately, this is not a straightforward task. The precision of the stacking films in the brick assembly is limited to approximately $50 \div 100 \mu\text{m}$[87], and the films can be distorted from their original spatial configuration due to external factors such as humidity and temperature during the data-taking stage. The tracking precision is not sufficient due to the low quality of the raw assembled hit data. This is a common problem for ECC detectors: unlike electronic detectors, where the relative positions of the detector layers are fixed and well-known, each emulsion unit will have its own unique set of distortion transformations.

To compensate for these effects, an alignment procedure is performed on the ECC data, the main objective of which is to transform the hit coordinates so that their relative coordinate configuration is matching the one maintained during the data taking as closely as possible. The procedure consists of an iterative pattern-matching procedure computing the parameters of the transformations:

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_f \\ y_f \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{2.29}$$

where x_f and y_f are single film track coordinates and x_b and y_b are the corresponding aligned ones.

A least squares fit is applied after maximising the number of matching pairs within predefined positions and the slope tolerances measured in particular areas of the emulsion film[8]. The main objective of the procedure is to calculate the parameters of the shift and rotational transformations of the hit coordinates at which the average least squares fit is optimal. By applying the above procedure to each pair of consecutive films, the relative displacements along the x-, y-, and z-axes can be reduced to a few μm, which is sufficient for the tracking procedure.

---

[8] Usually these areas are the corners of the emulsion plate.

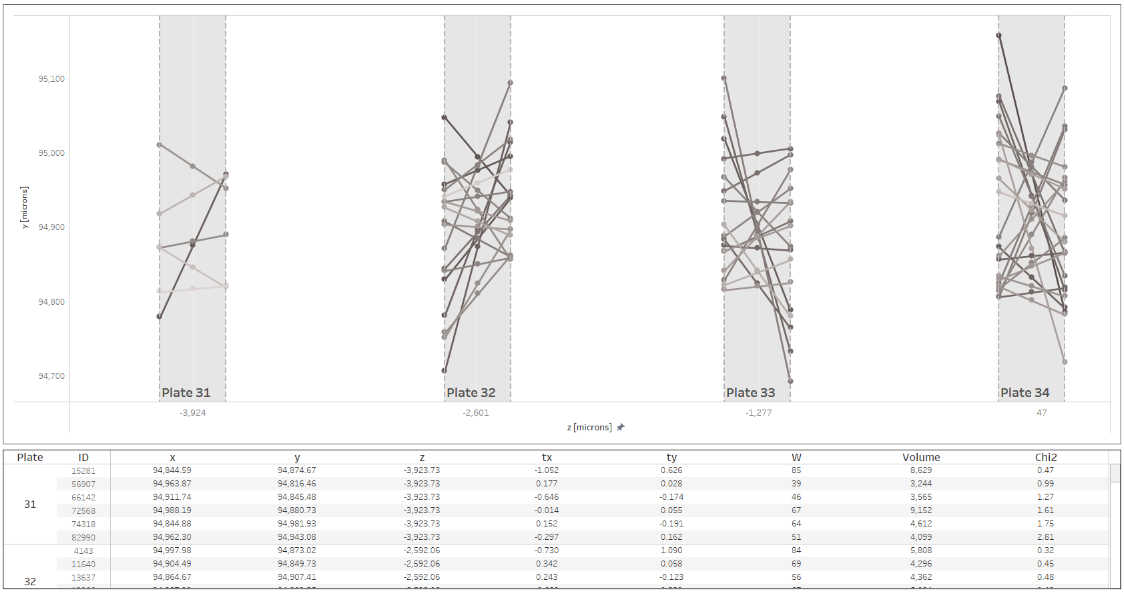


Plate	ID	x	y	z	tx	ty	W	Volume	Chi2
31	15281	94,844.59	94,874.67	-3,923.73	-1.052	0.626	85	8,629	0.47
	56907	94,963.87	94,816.46	-3,923.73	0.177	0.028	39	3,244	0.99
	66142	94,911.74	94,845.48	-3,923.73	-0.646	-0.174	46	3,565	1.27
	72568	94,988.19	94,880.73	-3,923.73	-0.014	0.055	67	9,152	1.61
	74318	94,844.88	94,981.93	-3,923.73	0.152	-0.191	64	4,612	1.75
	82990	94,962.30	94,943.08	-3,923.73	-0.297	0.162	51	4,099	2.81
32	4143	94,997.98	94,873.02	-2,592.06	-0.730	1.090	84	5,808	0.32
	11640	94,904.49	94,849.73	-2,592.06	0.342	0.058	69	4,296	0.45
	13637	94,864.67	94,907.41	-2,592.06	0.243	-0.123	56	4,362	0.48

Figure 2.15: Visualisation of a reduced sample of Run 0 data generated by the DAQ process. Grey areas indicate plate locations, while lines show base tracks with spatial and angular information. The dataset has been downsampled for illustration purposes and does not represent typical occupancy.

Standard data reconstruction process

The size of the data generated by the DAQ process is enormous: even the emulsion detector used in Run 0 that has the lowest integrated luminosity of ≈ 0.46 fb^{-1} has a typical hit density reaching $\mathcal{O}(10^5)$ cm^{-3}. The overwhelming majority of these hits are caused by the background events, with less than 1% belonging to the signal.

The task of the data reconstruction process is to create a data representation that would assist in the efficient discrimination of background and signal events. This process can be divided into three distinctive phases:

1. **Tracking:** identify and assemble the suitable series of hits that could be created by the same particle into volume tracks[9].
2. **Vertexing:** identify the group of two or more tracks that seem to originate from the same point in space and that can be associated with particle decay or interaction. These groups of tracks are clustered into higher-order structures called *vertices*.
3. **Event building:** particles originating from one vertex can undergo subsequent decay or interaction, creating a secondary vertex in the detector data. The task of the event-building exercise is to group suitable vertex candidates into events that provide a holistic view of the particular physical process.

The standard tracking for the SND@LHC experiment is performed using a software framework called FEDRA[88] originally developed for the OPERA experiment. It consists of multiple shared libraries of C++ classes and routines that perform various steps of data reconstruction, including tracking and vertexing. Tracking is a

---

[9] The term *volume* is used to distinguish higher-order tracks that are made of multiple base tracks in emulsion.

relatively complex procedure that involves processing large amounts of data and can be split into four steps.

1. **Data selection:** only base tracks with more than 20 silver grain clusters per microtrack pair and a chi-square value from the fit of two matching microtracks that is less than 2.4 are utilised for tracking. At the time of writing this thesis, optimisation is ongoing to make a more thorough selection based on other base track parameters, namely tx and ty.
2. **Seeding:** the seeding step selects all possible pairs of adjacent base tracks that meet the spatial and angular acceptance criteria. These seeds serve as a trigger for the next step.
3. **Track propagation:** the seeds of the previous step are used as a starting point for the track propagation procedure. This is the iterative algorithm, which uses an adapted version of the Kalman filter[89] and at each propagation step performs three operations: *filtering*, *prediction*, and *update*. Filtering involves building an estimated representation of the particle trajectory parameters at each plate based on previous measurements. These include the coordinates and angle components at which the track is expected to traverse in a given plate. The uncertainties associated with the variables are also estimated and encoded in the covariant matrix. Since the SND@LHC emulsion vertex detector is free of any externally induced magnetic fields, the track's interception parameters at the next plate are predicted by using an equation of a straight line in a three-dimensional space. If a hit is found on the next plate that is close enough to the prediction, it will be measured and added to the track. The magnitude of the alteration to the current state is determined by the discrepancy between the measured and estimated parameters of the hit, as well as the covariance matrix. The larger the difference, the less the adjustment. If no hit is found at the next plate, then the prediction is extrapolated to the next plate: the detection efficiency of the emulsion plate is less than 100% and therefore one or two gaps[10] are tolerated before suspending the propagation procedure. Once propagation finishes, a *smoothing procedure* is applied: using the state calculated at the last step, it retrospectively recalculates the estimated track parameters and their residuals at previous points to calculate the overall quality of the track fit.
4. **Final output:** once all seeds have been utilised, each group of hits is assigned a numerical tracking identifier, as well as its fit value. If a hit has not been included in any track, it will not have any tracking information; however, it is still kept in the data for potential re-tracking procedures or examinations of the reconstruction effectiveness.

In addition to tracking, FEDRA performs a vertexing procedure[11] using volume tracks generated from the previous step. The main purpose of the vertexing procedure is to identify a set of tracks that represent the decay products of a common mother particle. This group of tracks is identified by analysing their trajectories: if all tracks within a group with sufficient certainty seem to be emerging from a small

---

[10] Commonly called 'holes' in the experimental particle physics.

[11] FEDRA uses libVt++ libraries originally written for Hera-B.

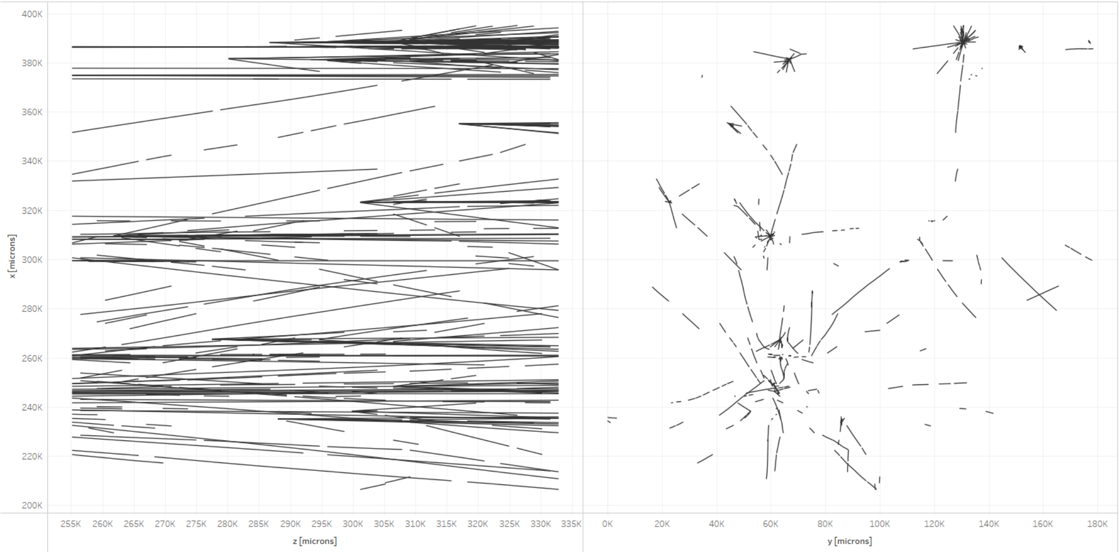


Figure 2.16: A visualisation of a reduced sample of Monte Carlo-generated data which has undergone FEDRA tracking. Two images show the same data from different points of view. The dataset has been downsampled for illustration purposes and does not represent typical occupancy.

point in space in a fiducial volume of the detector, then it can be classified as a complex object called *vertex*. It is important to make a clear distinction between a term *vertex* defined in Feynman diagrams and the observed vertex in the context of experimental physics. The latter one is a much broader term that also includes a group of particles coming from a much more complex physics processes than just two-body decays and can result in a vertex with track multiplicities exceeding two. One of the typical examples of such a process is a neutrino-nucleus DIS process, which, in addition to the muon, can also yield a jet of hadrons. This algorithm employs techniques akin to those used in tracking; however, the amount of data that needs to be handled is much less.

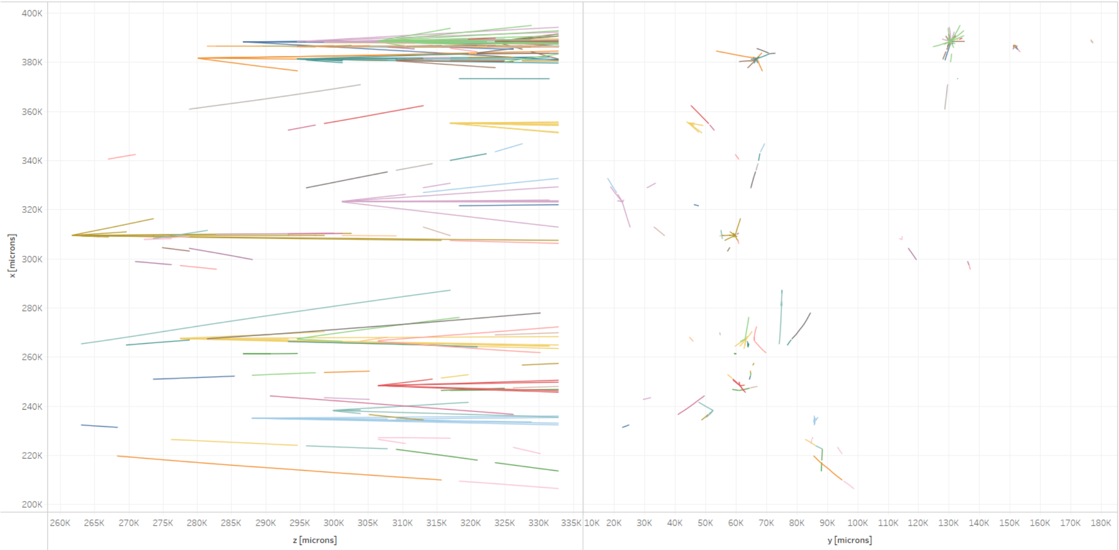


Figure 2.17: A visualisation of a small sample of the Monte-Carlo data which has undergone the FEDRA vertexing procedure, where each vertex is distinguished by its colour. Two images represent the same data but from different points of view.

1. **Data selection:** only reconstructed tracks which consist of at least four hits are used, and all tracks that start at the first plate are eliminated to reduce the background.
2. **Seeding:** the seeding step selects all possible pairs of volume tracks that are close to each other. For this purpose, an Euclidean distance between tracks' starting points is calculated: the seed is formed if the distance is less than some preset value (around 4000 microns).
3. **Seed fit:** in the absence of magnetic fields, each track in the seed can be modelled as a straight line. For each pair of non-parallel lines in a three-dimensional space, there is a point at which the distance between them is minimal and can be interpreted as the vertex origin. Cuts in the minimum approach distance and some additional topological considerations are used to reject seeds that are likely to be formed by two unrelated tracks.
4. **Seed merging:** all seeds from the previous step that have a common track are merged to form vertices with greater multiplicity. The process of merging uses a Kalman filter technique in a similar manner as for tracking, but instead of hit coordinates, the vertex origin position and its uncertainties are used.
5. **Final output:** once all suitable seeds have been combined, each group of tracks taking part in the merging process is given a numerical vertex identifier, which is accompanied by additional metadata on the vertex characteristics, such as the origin and the quality of the fit. In contrast to detectors such as ATLAS, which have been designed to accurately measure the momentum and type of particles, the tracks in the SND@LHC emulsion vertex detector lack this information, making it more challenging to calculate the vertex parameters, including its invariant mass.

The final outcome of the reconstruction is kept in a ROOT file, which is ready for the final analysis.

2.5.5 Physics goals

While the research programme of the SND@LHC is less ambitious than that of SHiP, the neutrino sector still offers an excellent opportunity to perform tests of the Standard Model and probe for new physics.

One of the measurements where SND@LHC can have an impact due to the limitations of current experimental results is the neutrino-nucleon cross-section at medium-high energies. Existing measurements have been made either with energies below 350 GeV or extremely high energetic regimes of 10 TeV-1 PeV[90], leaving the region between 350 GeV and 10 TeV unexplored. SND@LHC is designed to probe this unexplored energy range and directly measure $pp \to \nu_X$ cross-section in this intermediate regime[91].

The LHC produces all three types of neutrinos in the necessary energy range. The forward neutrinos are particularly energetic because of the Lorentz boost along the beam axis. Therefore, having a detector with a high pseudorapidity acceptance ($\eta > 5$) is a desirable feature that is not present in machines such as ATLAS or

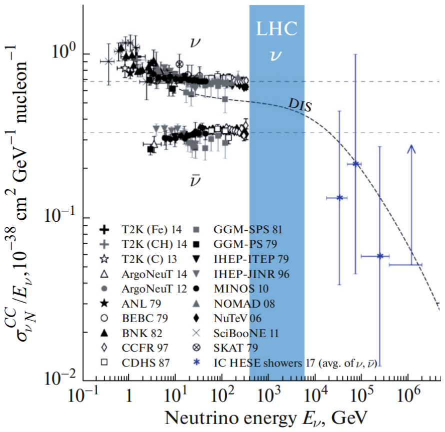


Figure 2.18: Available measurements of the neutrino cross-section. The blue area indicates the energy range targeted by SND@LHC. Adapted from [92, 91].

.

CMS. Additionally, the LHC-generated neutrinos have a large flux, which leads to a high number of signal events even in small neutrino detectors such as SND@LHC.

There are four main channels of neutrino production in LHC: charm, beauty, pion, and kaon-hadron decays. In [93], the leptonic decays of the W and Z bosons were considered as additional sources of neutrinos, but their rapidity distribution ($4 < |\eta| < 5$) falls outside of the acceptance of SND@LHC, making this contribution negligible. Each production channel is expected to provide a different contribution to the respective neutrino flavour energy spectrum. As a consequence, identifying the flavour and energy of the detected neutrinos provides insight into heavy-flavour production.

The main physics measurements, Standard Model tests and searches available at the SND@LHC experiment are described below.

Charmed-hadron production in pp collisions

In the SND@LHC pseudorapidity range, ν_e and $\bar{\nu}_e$ neutrinos come mainly from charmed hadron decays: by measuring and unfolding the electron-neutrino energy spectrum, the charmed-hadron production process at LHC can be directly probed. The main challenge of this procedure is to subtract the decays of the kaon and beauty hadrons that have an estimated contribution of about 10% and 3%, respectively. Although various data analysis techniques can be performed to mitigate this problem, it still limits the precision of the measurement.

Lepton flavour universality test in ν interactions

The majority of tau neutrinos ν_τ and $\bar{\nu}_\tau$ are produced by the process $D_s \to \tau + \nu_\tau$ and the subsequent decay $\tau \to W + \nu_\tau$, with only a small fraction (8%) coming from beauty hadron decays. Unlike tau, the electron neutrino comes from decays of all charmed mesons such as D^0, D, D_s, and Λ_c. Therefore the ν_e/ν_τ or R_{13} can be expressed as:

$$R_{13} = \frac{N_{\nu_e+\bar{\nu}_e}}{N_{\nu_\tau+\bar{\nu}_\tau}} = \frac{\sum_i f_{c_i} Br(c_i \to \nu_e X)}{f_{D_s} Br(D_s \to \tau\nu_\tau)}. \tag{2.30}$$

The precision of this method is limited to 30% due to the low number of tau neutrinos. In addition to the ν_e/ν_τ, a ν_e/ν_μ or R_{12} ratio is also considered where muon neutrinos are more abundant:

$$R_{12} = \frac{N_{\nu_e+\bar{\nu}_e}}{N_{\nu_\mu+\bar{\nu}_\mu}} = \frac{1}{1+\omega_{\pi/k}}. \tag{2.31}$$

Despite the complications that arise because ν_μ has many production mechanisms, including pions and kaons, which are represented by $\omega_{\pi/k}$, the lepton flavour universality (LFU) test involving electron and muon neutrinos has a lower uncertainty of 15%.

By comparison, lepton universality tests performed in Z boson decays at LEP[94] and W boson decays at the LHC[95, 96] achieve combined statistical and systematic uncertainties at the $\approx$ 1-2% level. While the SND@LHC measurement is comparatively coarse, its goal is not to compete but to complement these results by probing LFU in a novel environment: the LHC forward region with different production mechanisms, systematic uncertainties and higher TeV energy regime. Any significant deviation from Standard Model predictions observed in this environment could provide particularly robust evidence for new physics, while agreement offers an additional independent cross-check.

A precedent illustrating the value of complementary measurements is the lepton flavour universality test conducted by LHCb[97], which compared the branching ratios of $B^+ \to K^+e^+e^-$ and $B^+ \to K^+\mu^+\mu^-$. The measured R_{12} was 0.846, and with initial uncertainties of approximately 4-5%, this produced a 3.1σ tension with the Standard Model. Subsequent measurements by LHCb, with a combined uncertainty of about 11-12%[98], reduced this discrepancy and resolved the anomaly. Notably, this uncertainty is not far from the projected SND@LHC precision, highlighting that SND@LHC, in principle, can provide meaningful complementary tests of LFU in a completely different experimental environment.

Measurement of the NC/CC ratio

Neutral-current neutrino and charged-current neutrino interactions are easily distinguishable in the SND@LHC detector through the final-state products: unlike NC, the CC interactions produce a charged lepton that can be detected. By measuring each individual production rate, the ratio P can be calculated:

$$P = \frac{\sum_i \sigma_{NC}^{\nu_i} + \sigma_{NC}^{\bar{\nu}_i}}{\sum_i \sigma_{CC}^{\nu_i} + \sigma_{CC}^{\bar{\nu}_i}}. \tag{2.32}$$

This can be expressed in terms of the Weinberg angle θ_W as:

$$P = \frac{1}{2}\{1 - 2\sin^2\theta_W + \frac{20}{9}\sin^4\theta_W - \lambda(1-2\sin^2\theta_W)\sin^2\theta_W\}. \tag{2.33}$$

The primary source of uncertainty in this measurement is the asymmetry between neutrinos and antineutrinos, whose exact value is unknown. It is estimated to be below 20% across the entire energy spectrum, and the overall systematic and statistical uncertainty is predicted to be around 10% and 5%, respectively.

While this precision is about 15 times worse than that achieved by CHARM in the 1980s, where $\sin^2\theta_W$ was determined with $\sim$1% accuracy [99], the goal of SND@LHC is not to provide a competitive determination of the Weinberg angle. Instead, the measurement serves as an important internal consistency check of detector performance and analysis methods in a new energy and phase-space regime.

FIP scattering

Apart from testing the Standard Model, SND@LHC is also capable of probing the hidden sector by searching for feebly interacting particles. Depending on a particular BSM scenario, direct searches for particles with masses of up to a few GeV can be performed by detecting their scattering off either atomic electrons or nucleons.

The most commonly hypothesised Hidden Sector particles that could recoil off electrons in the SND@LHC detector are the products of the dark photon decay process $\mathcal{A}' \to \chi\chi'$. In the considered mass range of $m_{\mathcal{A}'} \leq 1$ GeV there are two main dark photon generation mechanisms:

- Meson decays: radiative decays of light mesons, such as:

$$\pi^0, \eta, \eta' \to \gamma\mathcal{A}' \tag{2.34}$$

$$\omega \to \pi^0\mathcal{A}' \tag{2.35}$$

- Bremsstrahlung of protons: dark photons radiation from the primary protons at the LHC collision point.

The main signature of the DM candidate χ scattering in the detector is the reconstructed shower induced by the recoiled electron. This search channel is not background-free, and multiple Standard Model processes exhibit a similar topology. These include electron-neutrino elastic scattering $\nu_x e^- \to \nu_x e^-$ and deep inelastic and quasi-elastic charge current scattering with unreconstructed tracks at the neutrino vertex. However, in the studies performed for the technical proposal[70] their overall contribution after the analysis cuts was found to be negligible.

The projected exclusion limits have a significant overlap with the MiniBooNE [100] and NA64[101] experiments, as seen in the figure below, and therefore the discovery potential of this method is somewhat limited.

However, the SND@LHC direct detection method will complement the limits of NA64, which are based solely on missing-energy detection techniques. The theoretical model considered for the **nucleon** case involves a scalar particle χ coupled to the SM via the leptophobic portal that can be described by the following Lagrangian:

$$\mathcal{L}_{leptophobic} = -g_B V^\mu J^B_\mu + g_B V^\mu(\partial_\mu\chi^\dagger\chi + \chi^\dagger\partial_\mu\chi). \tag{2.36}$$

Where J^B_μ is the baryon current that can be expressed as $J^B_\mu = \frac{1}{3}\sum_q \bar{q}\gamma_\mu q$. The presence of χ particles in the SND@LHC detector volume can be detected by their elastic or inelastic scattering off the protons, which manifests itself as a visible experimental signature of the charged track emerging from the interaction location. The χ particle originates from the decay of the vector mediator V, which can have multiple production channels such as:

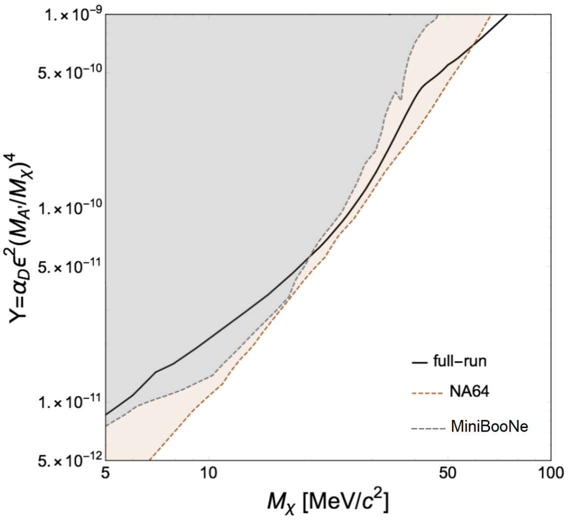


Figure 2.19: SND@LHC 90% C.L. exclusion limits in the background-free scenario for a Light Dark Matter particle χ originating from the prompt decay of $\mathcal{A}'$. Based on the assumption $m_{\mathcal{A}'} = 3\ m_\chi$ and $\alpha_D = 0.1$. The filled regions are excluded by MiniBooNE[100] (grey) and NA64[101] (orange).

- Unflavoured meson decays:
$$\pi^0, \eta \to \gamma V \tag{2.37}$$
- Bremsstrahlung of protons:
$$p + p \to V + X \tag{2.38}$$
- Drell-Yan process:
$$q + \bar{q} \to V + X \tag{2.39}$$

Each of these channels has a different contribution depending on the expected mass of the mediator, but the bremsstrahlung of protons is considered a leading mechanism. The background yield consists mainly of NC DIS or Resonant (NC RES) scattering of electron and muon neutrinos off protons and is expected to be reduced to a negligible level at the data analysis stage.

The Figure 2.20 below shows the sensitivity of SND@LHC to the leptophobic portal at the end of the expected data-taking period. A significant area of the accessible parameter space is currently unexplored by previous experiments, presenting a good opportunity to search for new physics.

2.6 Machine learning

2.6.1 Introduction

Machine learning is a division of the artificial intelligence field that is mainly concerned with a class of regression and classification problems for which building an explicit algorithm is either cost-prohibitive or its data modelling accuracy is insufficient to provide an adequate solution. ML provides an alternative approach

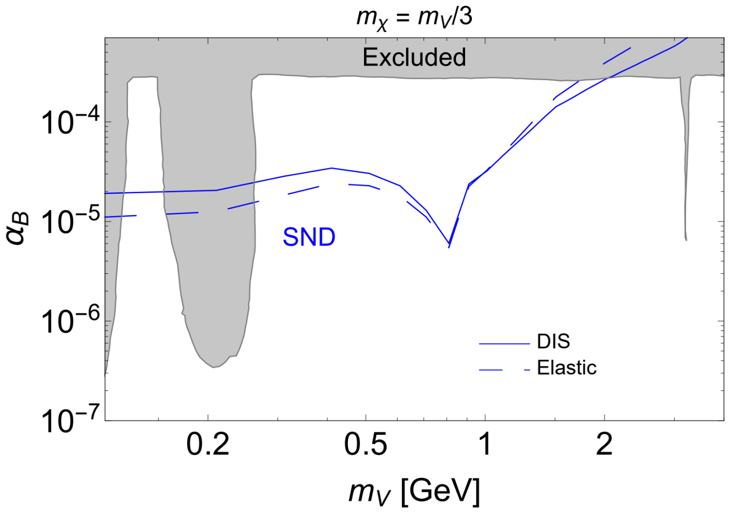


Figure 2.20: Sensitivity of the SND@LHC experiment to the leptophobic portal for both elastic (dashed blue line) and deep-inelastic scattering (solid blue line). The grey region was already excluded by previous experiments.

by which the computing device builds its own solution algorithm by analysing and interpreting the input data.

From a historical point of view, this method has been known for a long time, and the first practical attempts can be traced to the works of Marvin Lee Minsky[102] and Arthur Samuel[103] in the 1950s. Although those works were highly innovative for the period, the area of machine learning had limited practical use for a long time due to the lack of computing power and accessible digital data. In the past two decades, the landscape has been drastically altered by the affordability and speed of computers, as well as the emergence of cloud-based data services. The result of this is a great deal of digital material, including semi-structured and unstructured data, that is hard to process and analyse using regular analytical techniques.

One of the typical examples is the task of recognising and interpreting scanned handwritten text into machine-readable output. The simplest data object that could represent each handwritten character is the monochromatic $n \times m$ array, where each element can be represented by a Boolean variable. Building an explicit recognition algorithm would require accounting for every variation in handwritten text data that inevitably occurs due to various factors such as individual writing style, image distortion, and ink quality. Reducing the image resolution could partly alleviate this challenge, but at the cost of losing useful information, and hence reducing the accuracy. For more complex data sets, such as RGB photographic images and audio recordings, the implementation of human-prescribed algorithms becomes impractical due to the enormous amount of development time.

These challenges combined with advances in computing technology have stimulated rapid development of the machine learning field, elements of which at the time of writing have already penetrated many aspects of human life such as healthcare[104], customer service[105][12] and transport[106].

---

[12] The author participated in an industrial project to develop an AI chatbot for ASOS, a British online retailer of fast-fashion and cosmetics.

2.6.2 Structure

To address the wide spectrum of specific problems mentioned above, machine learning has evolved into a broad discipline containing many approaches. Although the exact structure of ML is subject to debate, it can be roughly divided into two main categories: *unsupervised* and *supervised* learning.

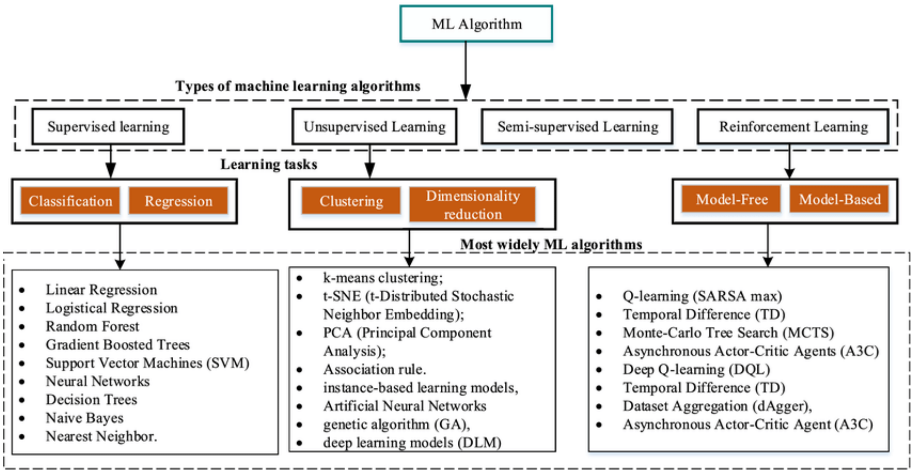


Figure 2.21: Examples of algorithms used in Machine Learning. Taken from [107].

The main objective of **unsupervised learning** is to extract hidden variables and patterns of interest from unlabelled data. One of the typical examples is clustering, which involves grouping data points $\boldsymbol{x_i}$ based on their characteristics. Principal component analysis (PCA) is another commonly used tool that aims to reduce the dimensionality of the input $\boldsymbol{x_i}$ by building a more compact representation of its feature set. In this research, a particular data mining technique called the evolutionary algorithm (EA) will be used, which involves finding an optimal value $\boldsymbol{x_{opt}}$ across the large domain of possible solutions.

Supervised learning involves training machine algorithms with labelled data where the term *labelled* indicates that for a given $\boldsymbol{x_i}$ data point, an associated output variable $\boldsymbol{y_i}$ is already defined. A typical example of such a data set is an MNIST dataset[108] that contains 70k handwritten character images and associated labels. The main objective of supervised learning is to provide a function that correctly maps an input $\boldsymbol{x_i}$ to an output $\boldsymbol{y_i}$ based on previous learning experience (in the MNIST case, associate a handwritten character to the letter or number it represents). The output $\boldsymbol{y_i}$ can be a continuous range of values for regression models or a discrete set if the task is of a classification nature. A type of ML called *deep learning* will be used in the SND@LHC experiment, which involves data processing using artificial neural networks or ANNs. Inspired by human biology, ANNs can have large degrees of freedom that enable them to model complex data representations and extract implicit features that may be inaccessible to other methods, such as logistic regression or decision trees.

2.6.3 Limitations of machine learning

Despite its successes, machine learning suffers from many limitations[13] that can potentially lead to failure. Although certain limitations depend on the particular type of ML used, to date three main sets of problems have been identified: data quality, the complex nature of the subject, and the interpretability of solutions.

Data quality

To provide the correct solution to the problem, machine learning supervised algorithms require exposure to training datasets that assist in building the input-output mapping function. These datasets are required to be representative of the problem in general: failure to achieve this condition can lead to the situation where the ML algorithm performs adequately on the training set but fails to do the same on the previously "unseen" data. Unsupervised algorithms also suffer from data-related issues but with their own specifics. For example, optimisation algorithms are sensitive to the definition of the objective function, which is used to quantify how adequate the solution is to solve the problem. If the definition is poorly chosen, the algorithm may find the solution that satisfies the criteria defined by the objective function but fails to solve the actual problem.

Complexity

Most of the ML tools require specialist knowledge and, in the majority of cases, are not trivial to set up and run. As an example, utilising artificial neural networks requires a lot of preparation from the hardware and software perspective. This includes multiple tasks such as: installing the relevant libraries, preparing data, choosing the right ANN architecture, training it, and eventually utilising its output. While there are plenty of available resources on the topic, the exact methodology using artificial neural networks is still considered a "dark art". For example, ANNs have many parameters, such as the number of layers, dropout, and learning rate, the exact values of which depend on the problem in question and, therefore, have to be worked out through trial and error. This is one of the reasons that ML is avoided for problems that can be solved with simpler methods.

Interpretation

From the user's perspective, the ML algorithm is a "black-box" function, the inner workings of which are hard to conceptualise. This makes it difficult to trace the source of any possible issues that arise in ML. For example, it is not uncommon for neural networks to learn from the wrong features present in the training dataset and fail to recognise them in the validation if those features are not present there. As a hypothetical example, the image recognition model can only be presented with images of white cats and black dogs. Instead of focusing on the morphological features such as shape and size, the model may focus on the colour itself and fail on the dataset involving black cats and white dogs. Another common issue observed in ML is an occasional "lack of common sense" effect that leads to solutions that

---

[13] The ethical issues of using ML will not be considered here as they are irrelevant in the context of the work being carried out.

are abnormal and unphysical. The concealed nature of ML algorithms makes it extremely difficult to trace and resolve the root causes of these issues.

2.6.4 Evolutionary algorithms

Evolutionary algorithm, or EA, is an optimisation method inspired by Darwin's theory of evolution. In this approach, the problem is solved by treating solutions as individuals that evolve into a better set of solutions. The exact mode of EA operation depends on its architecture; most canonical versions have the following steps in their optimisation logic:

1. **Initialisation:** multiple solutions with random[14] characteristics are generated in the beginning. They act as a start-up population for the evolution process.
2. **Evaluation:** each individual is tested on the problem in question, after which it receives its performance score called *fitness*. Generally, it is a scalar number, the value of which indicates how well the given individual solves the problem.
3. **Selection:** several individuals are sampled based on the fitness calculated in the previous step.
4. **Genetic operations:**
    - Crossover: the best individuals undergo recombination with each other by partially exchanging their genetically encoded characteristics. This is a conservative operator that generates the offspring with a mixture of its parent characteristics.
    - Mutation: the best individuals may undergo a small change in their genetic code that results in new genetic material in the population.
5. **Re-injection:** the new solutions created in Step 4 are released back into the population, and the process restarts from Step 2.
6. **Termination:** steps 2-5 can be repeated many times, but eventually the algorithm stops once a programmed terminating condition has been met.

Evolutionary algorithms were used in various areas such as cancer research[110], vehicle design[111] and traffic signal light control optimisation [112]. In [113], a study was conducted that used EA strategies to improve the design of a satellite dish beam, making it more durable and resistant to vibrations. Optimisation has produced a highly asymmetric structure, which was $20,000\%$ better than any traditional shape. It was discovered that the asymmetry itself was the main reason for the enhanced stability: The unequal lengths of the rungs caused the vibrational waves to meet in different phases, resulting in destructive interference.

This case reveals an interesting feature of EA: a lack of bias toward any symmetry or design consideration, which can provide a new route in solving the problem.

---

[14] There are variations of evolutionary algorithms called seeding EAs where the initial individual generation is driven heuristically. This approach has been described in [109].

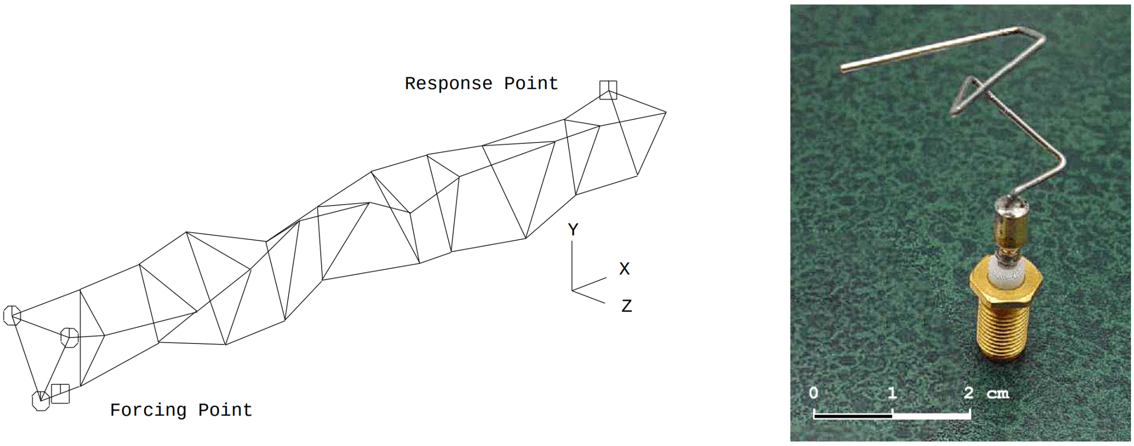


Figure 2.22: **Left:** A satellite boom dish derived after EA optimization in [113] **Right:** An actual evolved antenna used in ST5 spacecraft. Taken from [114].

2.6.5 Artificial neural networks

Introduction

Artificial neural networks ANNs represent a family of machine learning models whose organisational structure has been inspired by biological neural networks that participate in the activity of the animal brain. The basic unit of the ANNs is the artificial neuron, which loosely replicates the operation of the biological neuron.

The purpose of the artificial neuron is to transform signals from an arbitrary number m of inputs x_i to produce an output y according to the following mathematical formula:

$$y = \phi(\sum_{i}^{m} \omega_i x_i), \tag{2.40}$$

where the term ω_i is the weighting coefficient of each input x_i and ϕ is a non-linear transfer or activation function that takes the argument in the bracket and converts it into a real number, which generally takes a value between 0 and 1.

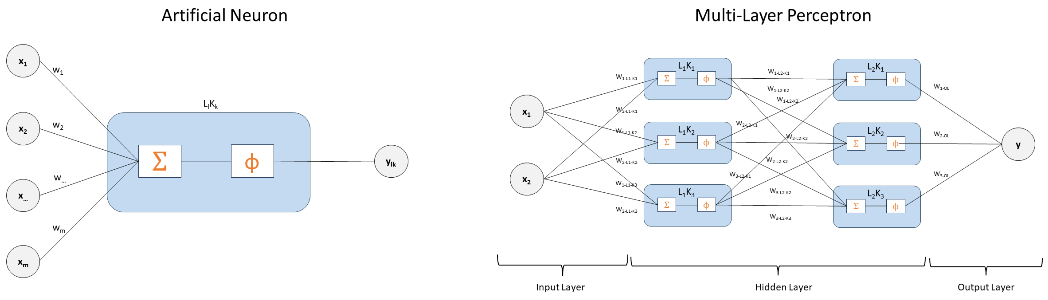


Figure 2.23: **Left:** Schematic representation of the artificial neuron. **Right:** Example of the two-layer perceptron.

Each artificial neural unit can be combined into groups called layers, and each layer can be mutually interconnected into a structure called a *multilayer perceptron* (MLP). Because of their scalability, MLPs can possess many degrees of freedom, limited only by the available computer resources. This makes them very powerful in modelling and extracting hidden patterns from complex datasets. This process is accomplished by sequentially passing, multiplying, and adding a continuously altered input signal from one layer to the next until it is converted into an internal representation that can be used for prediction. This is commonly referred to as a *forward pass* or *propagation*.

In principle, the multilayer perceptron can work with any type of data, but there are some situations where it is more practical to use more specialised types of ANN architectures. Although the full description of the types of ANN is outside the scope of this thesis, two types of neural networks used in this research will be discussed in more detail: convolutional neural networks (CNNs)[15] and geometrical neural networks (GNNs).

Convolutional neural networks

One of the examples in which a classical MLP struggles is image data processing: depending on the particular conditions under which the photograph was taken, the objects in the image are subject to various transformations such as rotations, shifts, and scaling. For example, if the classifier's task is to differentiate between cats and dogs[16], two pictures of two different cats must be classified as the same label. However, their shapes and morphology can differ considerably depending on the cat's breed, position, and the transformations mentioned above, which can result in MLP wrongly identifying one of the pictures as a dog. To make MLP more accurate, it will require a large number of layers and artificial neurons, which can be prohibitive from a computational point of view.

To address these challenges, convolutional neural networks are used instead. In addition to MLP layers, the typical CNN incorporates an additional set called convolutional layers that are specifically designed to recognise relationships between spatially distributed data points such as pixels.

Like most ML concepts, the idea of convolutional operations was inspired by biology and comes from the pioneering work of D.H. Hubel and T.N. Wiesel[115]. In 1959, they were studying cells in the striate cortex of lightly anesthetised cats. The main outcome of the experiment was the discovery of the so-called *excitatory* and *inhibitory* regions of the cat's retina, which were formed in a particulate fashion, making it receptive to specific observed visual patterns.

This work and the following works of D.H. Hubel and T.N. Wiesel have inspired the creation of the first computer neural network named *Neocognitron*[116] the main objective of which was to simulate the visual pattern recognition system of humans.

From a conceptual point of view, this neural network already represents two key features of modern CNNs: the convolutional and the downsampling layers, which specialise in extracting visual pattern attributes from the data. Unfortunately, in the 1980s, this network was too ahead of its time: the convolutional operation requires simple but numerous simultaneous matrix operations to yield sufficient performance.

This limited the practical use of CNNs until the widespread use of graphical unit

---

[15] Also commonly referred to as ConvNets.

[16] For some reason, "cats and dogs" classification tasks are commonly used in the CNN-related literature as an example and often form the first tutorial task for beginners.

processors[117] (GPUs) in the 2000s, which were originally created for the video game industry but were equally capable of solving "embarrassingly parallel"[17] problems in deep learning. In 2012, AlexNet ANNs made up of eight convolutional layers achieved astonishing top-1 and top-5 error rates of 37.5% and 17.0% in the classification of a data set consisting of 1.2 million images [118].

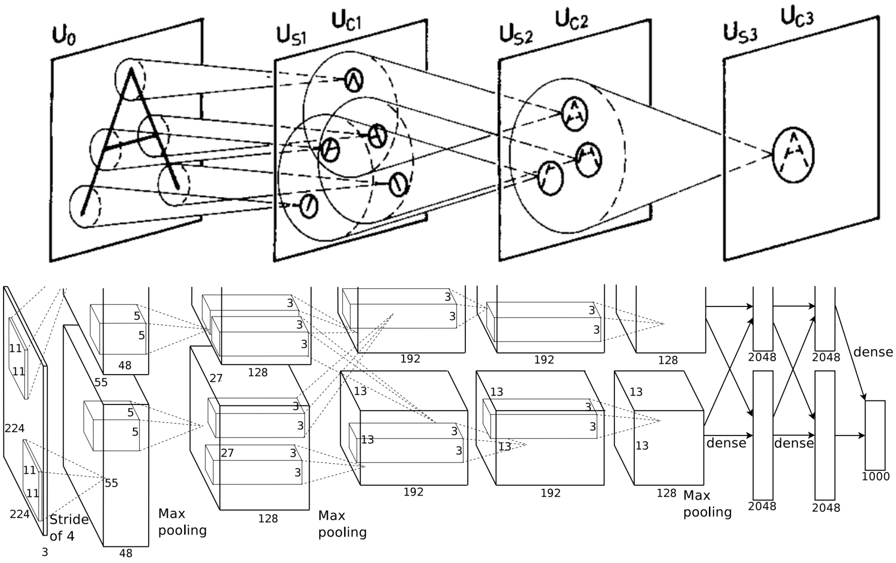


Figure 2.24: **Top:** Schematic representation of the Neocognitron proposed in [116] **Bottom:** Architecture used in more modern CNN called "AlexNet"[118].

Certainly, AlexNet was not the first multilayer CNN to achieve good performance in image recognition tasks. LeNet-5 [119] has shown an even lower error rate of below 1% in letter recognition tasks. Neither was it the first CNN to use graphical processors: already in 2006, it was demonstrated that GPUs can speed up CNN training by more than four times [120]. Yet, the AlexNet was the first CNN that combined all the previous innovations, such as multiple layers, the use of GPUs, ReLu activation function, backpropagation training, and dropout normalisation techniques, to achieve superior performance on the complex task of classifying colour photo images.

Although it has eventually been superseded by more advanced ANNs such as the one used in [121], the architecture of AlexNet provides the basis for the majority of the mainstream convolutional neural networks used today. The layout of the AlexNet-like CNN that will be used in this investigation is made up of the following components.

1. **Convolutional layer**: a set of matrices called filters or kernels, where each element initially takes a random real value. As a result, each kernel has a unique pattern that can sample specific features of the image. The process of sampling is called *convolving* (hence the name of the layer), which is achieved by sliding the filter across the image and continuously performing the dot product between the kernel and the corresponding area of the image. Following the convolution, a feature map containing dot product results is created for each kernel, accompanied by the individual activation function, which serves as the input for the next layer.

---

[17] Problems that can be easily separated into tasks that can be solved independently.

2. **Pooling layer** performs non-linear downsampling of the input map to reduce the size of the input feature map and identify the large-scale features. It reduces the number of CNN parameters and hence the amount of computation. This is done by splitting the feature map into rectangular areas of a particular size. For each region, the maximum measured value[18] is extracted and recorded in the new compact map along with its own activation unit. Although this step introduces some degree of information loss, this is not critical for pattern recognition tasks, since the presence and relative position of the image object features are more important rather than their relative distance.

3. **Flattening layer**: more convolutional and pooling layers can be incorporated into the CNN, where at each additional layer, the propagated information represents more abstract and complex features of the image data. After several iterations of convolution and pooling, the final feature map undergoes the process of *flattening*, which involves mapping the multidimensional feature map to a one-dimensional vector.

4. **Fully connected layer** where the output of the previous step is used for further analysis of the data using a classic MLP network, which has been described above.

5. **The output layer** consists of a group of artificial neurons, the number of which represents the size of the output result vector.

Each layer described above can be parametrised using many variables, such as the number and geometry of kernels, pooling size, and the choice of the activation function, to name just a few. This enables the user to fine-tune the performance and speed of the CNN depending on the task requirements and available computational resources.

Geometrical neural networks

The main purpose of geometric learning is to extract information from complex relational data sets that can be expressed as graphs. The main motivation for the subject came in the mid-20th century from the field of chemistry, which was concerned with studying the properties of molecular compounds. The need to analyse their complex geometry led to the development of the first GNN-like algorithm by A. Leihman and B. Weisfeiler[122].

However, by the 20th century, it was realised that graph learning could be applied to a wide spectrum of data concepts such as social networks, semantics, and even image analysis: arrays representing image pixels can also be expressed as graphs. Therefore, the convolutional neural network described in the previous section is a particular example of geometrical learning, but due to the rigid structure of the constituent convolutional and pooling operators, the CNNs can work on the Euclidean domains only.

To overcome this limitation, a more advanced neural network was required; however, surprisingly, a lot of time has passed since the work of A. Leihman and B. Weisfeiler to the first implementation of the modern geometrical neural network.

---

[18] Other aggregation functions such as taking the average are also commonly used.

The first report describing GNN was published in 2008[123], which makes it a relatively new concept compared to other neural network architectures.

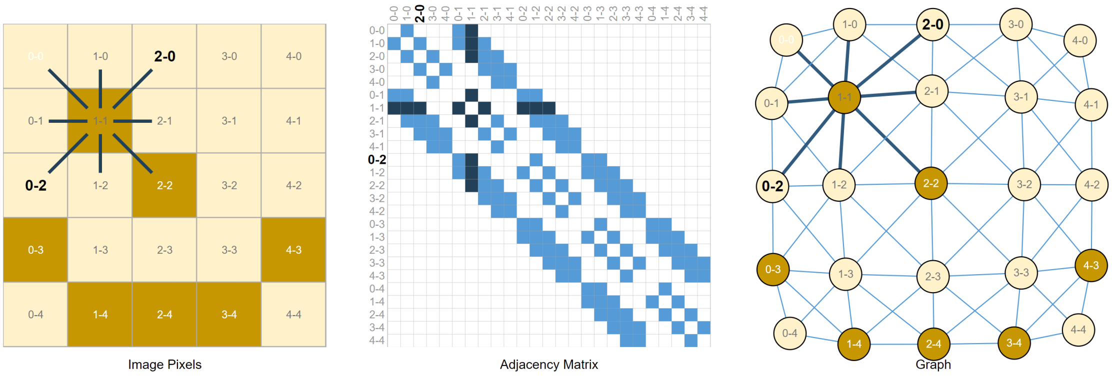


Figure 2.25: Schematic representation of the image pixels on the left, corresponding graph representation on the right and the adjacency matrix in the middle that encodes the relationship between pixels or nodes. Taken from [124].

There is a lot of ongoing research in the field of geometrical neural networks, with a wide variety of types available, so it is difficult to cover the topic in its entirety within the confines of this thesis. Since GNNs are merely a tool and not the subject of this investigation, only a brief overview based on the available literature[125] and [126] will be used, with illustrations used from [124]. There are two key aspects that make GNNs a powerful tool: graph data representation and a corresponding neural network machinery that can exploit them.

The data representation commonly used in geometrical learning is a graph structure consisting of two essential elements:

1. **Nodes**: the sets of data points that make up the graph and are usually the main subject of the analysis. Each node can possess a vector of variables called attributes that characterise the properties of the node. As an example, the molecular structure of the substance can be expressed as a graph, where the nodes are individual atoms and each possesses individual attributes such as atomic number or weight.

2. **Edges** which represent the connections between nodes and their own vector of attributes that describe the property of the edge. In the molecular example above, the edge can represent the valence bonds between atoms, and its attributes could describe the associated bond properties, such as the number of electrons and the bond strength. Edges participate in information exchange between nodes, where the data can flow in both directions for undirected edges and in one direction for directed edges.

By combining two components, it is possible to create graphs of varying complexity: fully and partially connected graphs with nodes and edges that may or may not have attributes, the only limitation being the available computing power. This makes graphs very versatile for describing practically any kind of data set, including the vectors and arrays used in the MLP and CNN architectures, respectively.

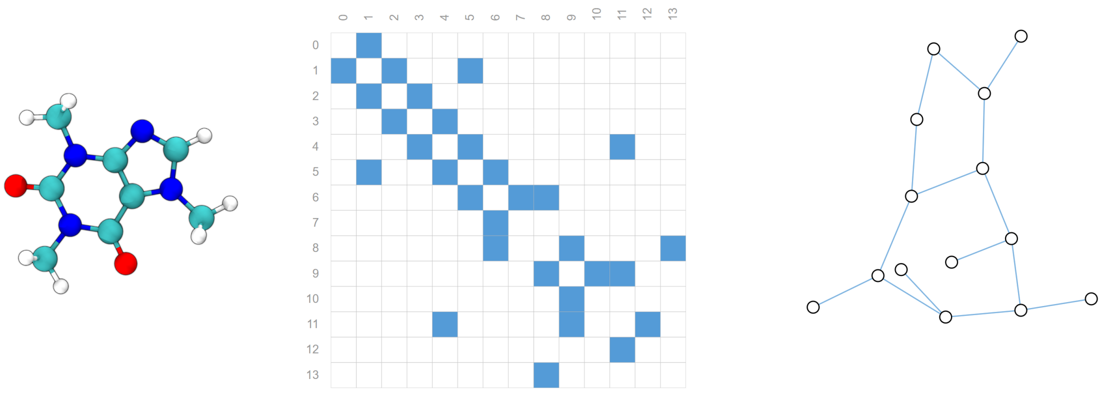


Figure 2.26: 3-d model of the caffeine molecule on the left, corresponding graph representation on the right and the adjacency matrix in the middle that encodes the relationship between atoms or nodes. Taken from [124].

When it comes to data processing, there are three classes of problems that geometrical neural networks are concerned with:

1. **Node-level analysis**: the main objective of this task is to predict the properties of the node based on available data such as its attributes, properties of the adjacent edges and other nodes. For a very simple GNN, this can be done simply by assigning to each node its own individual MLP neural network, which would remap the inputs to the other node. This results in the second layer representing the graph, which structurally is identical to the initial graph, but with nodes having different and more abstract attributes generated by the MLP. This operation can be repeated numerous times, creating a multilayer architecture in which each subsequent layer stores higher-level features. For the hypothetical chemical case, a node-level analysis task could involve the classification of atoms in a molecule on the basis of their properties.

2. **Edge-level analysis**: conceptually, the edge-level analysis is the same as for nodes, where the latter are replaced by edges. This is used in classification and regression tasks where the connection between nodes is the subject of study. For the hypothetical chemical case, an edge-level analysis task could involve the classification of the bond strength between atoms and their type.

3. **Graph-level analysis**: this is an instance in which the entire graph and its arrangement are being evaluated for a classification or regression task. Generally, this involves concatenation of all the information from the nodes and edges into one master node. In essence, this is the GNN analogue of the flattening operation used in CNNs. For the hypothetical chemical case, a node-level analysis task could involve the classification of the molecular substance or its properties as a whole.

In order to leverage the connectivity information encoded in graphs, modern GNNs use two additional techniques: *pooling* and *message passing*. The pooling operation can be used in two ways:

1. **Graph-structure preserved pooling**: this type of pooling is used in situ-

ations where nodes do not have any attributes, and edge properties are used instead. The operation involves aggregating all attribute vectors of the connecting edges and mapping to the node via the equation:

$$\boldsymbol{x}_i = \bigoplus_{j \in N_i} \boldsymbol{e}_{ij}, \tag{2.41}$$

where $\boldsymbol{x}_i$ is a new vector of elements of node i, N_i is a number of adjacent edges, $\boldsymbol{e}_{ij}$ is a feature vector of the edge ij and $\bigoplus$ is an arbitrary aggregation operator of choice (sum, mean, max, etc.). The graph structure is invariant under this pooling transformation as the number of nodes remains constant.

2. **Graph-structure reduction pooling**: the pooling operator merges the neighbouring nodes into one node using the following equation.

$$\boldsymbol{x}_k = \bigoplus_{i \in N_k} \boldsymbol{x}_i, \tag{2.42}$$

where $\boldsymbol{x}_k$ is a vector of elements of the new node k, N_i is a number of participating nodes, $\boldsymbol{x}_i$ is a feature vector of node i. This type of pooling is similar to the downsampling operation used in CNNs, resulting in a smaller graph with a lower number of nodes. However, the GNN version of the pooling has additional complications associated with the possible existence of edges, which need to be rearranged to match the new node configuration. A more in-depth review can be found in [127].

The second powerful technique used in geometrical neural networks is *message parsing*, which involves exchanging information between the nodes and can be described by the equation:

$$\boldsymbol{x}_i = \bigoplus_{j \in N_i} \boldsymbol{x}_j, \tag{2.43}$$

where $\boldsymbol{x}_i$ is a new vector of elements of the node i, N_i is a number of connected nodes and $\boldsymbol{x}_j$ is a feature vector of the neighbour node j. The single application of the message parsing results in every node accumulating information from its nearest neighbours, leaving the structure of the graph unchanged. This process can be repeated multiple times, allowing the nodes to gain knowledge from their more distant nodes by utilising the data that their neighbouring nodes have collected in the preceding step.

Assuming a sufficient number of iterations, each node can eventually incorporate all the data given enough computing power. It is important to note that the operations described above can equally apply to edges with some small algorithm adjustments.

Training artificial neural networks

Regardless of the architecture chosen and its complexity, the model itself does not have a prediction capability. To make meaningful predictions, any supervised NN model requires some degree of data training. One of the most common learning methods used at the time of writing is the *backpropagation* technique, the main objective of which is to find the optimal parameters of the neural network so that the predictions of the model are as close as possible to the target values.

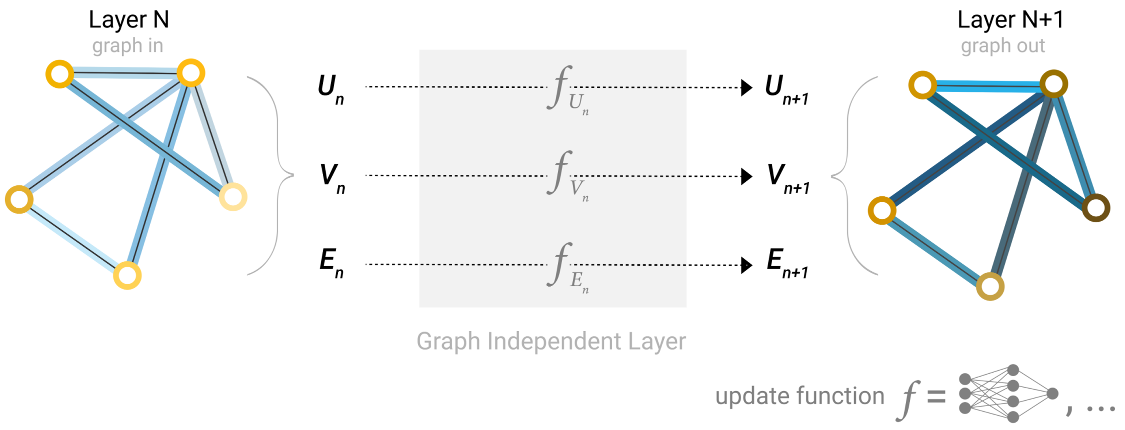


Figure 2.27: Schematic representation of the node-, edge- and graph-level analysis of the graph data by using the simplest realisation of the GNN. Taken from [124].

The backpropagation algorithm can be split into four parts: *forward pass*, *backward pass*, *summation of the gradients* and *weight update*. For simplicity, the example of the binary classification task will be used below, and the bias parameters will be incorporated into the definitions of weights. During the **forward pass** each input $\boldsymbol{x}_i$ is fed into the input layer of the neural network and propagated through each layer and each neuron node until the output node values are calculated. During the propagation of each node, the activation value and the output of the connected nodes of the preceding layer are recorded for the latter step. Their predictions $\boldsymbol{p}_i \in [0, 1]$ are compared with the target values of $\boldsymbol{y}_i \in \{0, 1\}$: the difference in these values is quantified by the loss function. The loss function that was used in this thesis is called cross-entropy loss (CEL)[19] and it takes the following form:

$$L_{CEL}(X, \theta) = -\sum_i y_i \ln(p_i). \tag{2.44}$$

Arguments X and $\boldsymbol{\theta}$ are the input pairs $(\boldsymbol{x}_i, \boldsymbol{p}_i)$ and the parameters of the neural network, respectively. For a completely untrained model, these predictions will be random, and hence, the loss function is expected to be high. To lower it, the next step is to propagate it backwards all the way to the input to work out the changes that need to be applied to the parameters $\boldsymbol{\theta}$: backward pass.

The main objective of the **backward pass** is to calculate the rate of change of the loss function $L_{CEL}(X, \boldsymbol{\theta})$ with respect to the parameters of the neural network $\boldsymbol{\theta}$ or $\frac{\partial L_{CEL}(X,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ that eventually is used for the weight update. This is not as trivial as it sounds: the loss function above is defined for a whole network; however, individual losses per node, except the input ones, must be calculated. This hurdle is overcome by using the sum rule, which states that the sum of the function derivatives is equal to the derivative of the function sum. Therefore, the backward step begins by calculating the derivative of each node n using the chain rule according to the

---

[19] The *mean squared error* (MSE) loss function is also commonly used, but it was abandoned in favour of CEL due to the latter having steeper gradient which has a higher penalty on the predictions that are far from the target.

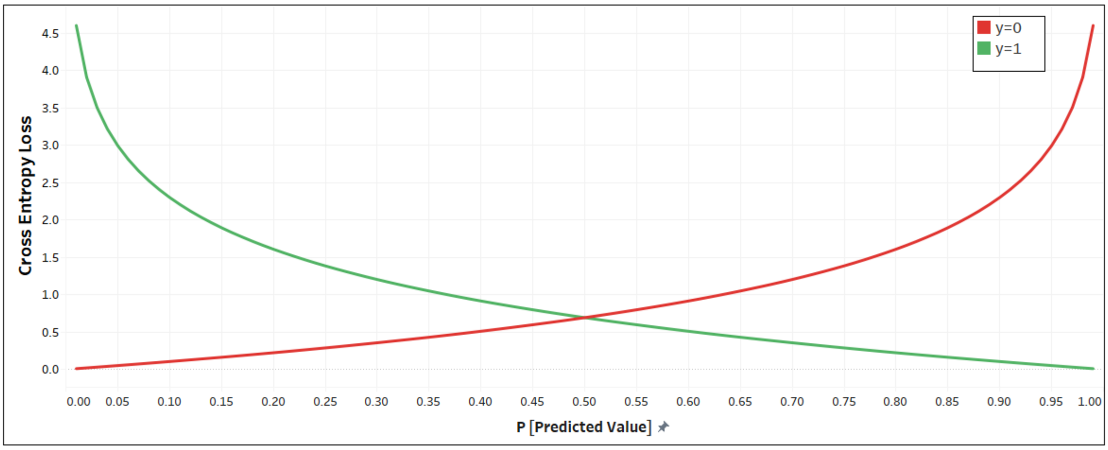


Figure 2.28: The plotted Cross Entropy Loss for the binary classification example function of the predicted value for the target labels y=0 (red) and y=1 (green).

formula:

$$\frac{\partial L_{CEL}}{\partial w^l_{en}} = \frac{\partial L_{CEL}}{\partial a^l_n}\frac{\partial a^l_n}{\partial w^l_{en}}, \tag{2.45}$$

where a^l_n is the activation of node n in layer l which can be expressed as:

$$a^l_n = \sum_{n=0}^{r_{l-1}} w^l_{en} o^{l-1}_n, \tag{2.46}$$

and w^l_{en} is the weight of the connection e to the node. The first term on the right-hand side of equation 2.45 indicates the change of L_{CEL} with respect to activation and is usually referred to as an error and is denoted by

$$\delta^l_n = \frac{\partial L_{CEL}}{\partial a^l_n}, \tag{2.47}$$

and by using the previous definition of a^l_n 2.46 the equation 2.45 can be rewritten as:

$$\frac{\partial L_{CEL}}{\partial w^l_{en}} = \delta^l_n o^{l-1}_e, \tag{2.48}$$

where the left term is an error term for node n and the right side reflects the output from the node at the previous layer $l-1$ that is connected through e. The terms o^{l-1}_e are already known since these were recorded during the forward pass step. The δ^l_n are trickier and their exact form depends on whether they are located in the final or hidden layer. For the final output node in the last layer $l = F$, the term 2.47 can be worked out using the definition of loss 2.44 and 2.47 which after some laborious algebra can be expressed as:

$$\delta^F_n = a^F_n - y_n. \tag{2.49}$$

The hidden layers can be derived using calculus, and the result is a bit more complicated but still relatively easy to understand:

$$\delta^l_n = \psi(a^l_n)\sum_{e=1}^{r^{l+1}} w^{l+1}_{en}\partial^{l+1}_e. \tag{2.50}$$

The equation above reveals that the calculations of the partial derivatives flow from the last layer to the first hidden layer, thus giving the process its name. Once the δ_n^l functions have been calculated, they can be plugged into expression 2.49 to calculate the partial derivative of the individual loss per node n: $\frac{\partial L_i}{\partial w_{en}^l}$.

The **summation of the gradients** is the consolidation step that takes all the calculated partial derivatives from the previous part and combines them into the total loss derivative using the sum rule:

$$\frac{\partial L_{CEL}(X, \boldsymbol{\theta})}{\partial w_{en}^l} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial L_i}{\partial w_{en}^l}. \tag{2.51}$$

This expression is used to **update the weights** using the *stochastic gradient descent* (SGD) technique according to the formula:

$$\Delta w_{en}^l = -\alpha \frac{\partial L_{CEL}(X, \boldsymbol{\theta})}{\partial w_{en}^l}, \tag{2.52}$$

where the parameter α is a step size commonly called the *learning rate* in the field of machine learning.

The operations described above, especially the last, have an associated computational cost, which for a complex neural network can become a significant problem. To address this issue, the mini-batch gradient descent technique is employed, in which the training data set is divided into smaller segments of equal size called *batches*, each comprising several data points. In this method, the SGD is executed once per batch after computing the loss gradients for the individual data points.

After processing all batches, it is said that the NN model has undergone training for a single *epoch*, a term in machine learning that signifies the neural network model has been through one complete cycle of the data set.

Once the epoch has been finished, the model loss is evaluated on both the training dataset and the previously "unseen" validation dataset to prevent bias and overtraining. Whether the model needs further training or should be abandoned depends on the improvement versus the previous epoch in both the training and validation data sets. Multiple outcomes can be observed; the most common are listed below:

- Model performance on both validation and training datasets is improving. In these cases, training is continued for another iteration(s).
- The model is still improving on the training set, but a plateau is observed in the validation domain: this is the sign of overfitting, which is caused by the neural network modelling the noise rather than the desired data features. In this instance, model training has to be abandoned as its performance can significantly degrade on unseen data.
- Oscillating training or validation loss: this is a sign that possibly the learning rate is too large to converge to the optimal point. Before continuing further training, its value is reduced to allow SGD to reach the local optimum point.

Overfitting is a fairly common problem in NN training that prompted the development of various regularisation techniques. One of the most commonly used is *dropout*[128], which attempts to regularise the neural network by disabling some nodes during the training process. Although it seems to improve the performance of the neural network model, the exact reason why it works is not fully understood[129], which just shows how unexplored the field of deep learning really is.

2.7 Conclusion

This chapter has outlined the theoretical framework underpinning the research. The Standard Model (SM) was presented as the prevailing theory of particle physics, describing fundamental particles and their interactions with remarkable accuracy. Nonetheless, it leaves key questions unanswered, including the origin of neutrino masses, the nature of dark matter, and the baryon asymmetry of the Universe: motivating the exploration of extensions collectively known as Beyond the Standard Model (BSM) physics.

Among these, special emphasis was placed on feebly interacting particles (FIPs). Heavy Neutral Leptons (HNLs), predicted in frameworks such as the seesaw mechanism and the νMSM, provide a minimal SM extension capable of addressing neutrino masses, baryogenesis, and possibly dark matter. Complementary approaches, including the vector and scalar portals, offer well-motivated connections to hidden sectors. Despite existing experimental constraints, vast regions of parameter space remain accessible to dedicated facilities such as SHiP and SND@LHC.

The interaction of both SM and BSM particles with detectors was reviewed, highlighting how signatures from electromagnetic, strong, and weak processes can be used to distinguish new physics from background. Neutrinos play a central role in both experiments, and lessons from past searches inform the design of these new facilities.

SHiP and SND@LHC were introduced as complementary experiments: SHiPs high-intensity fixed-target setup targets long-lived hidden sector particles, while SND@LHC exploits the unique forward neutrino flux at the LHC. Both face challenges of low signal rates and significant backgrounds, requiring advanced detection and analysis techniques.

Finally, the motivation for applying modern machine-learning methods, including evolutionary algorithms and deep learning, was established, given their potential for optimising reconstruction, classification, and signal discrimination in high-energy physics.

In summary, this chapter provides the theoretical and methodological basis for the analyses presented in Chapters 3 and 4, dedicated to SHiP and SND@LHC, respectively.

Chapter 3

Optimisation of the active muon shield at SHiP experiment

3.1 Motivation

The purpose of this chapter is to address the engineering challenge of deriving the optimal shape of AMS raised in Section 2.4. One way to overcome this problem is to treat it as a black-box function: simulate a passage of Monte Carlo-generated muons through the AMS and count the number of particles reaching the detectors. Those operations are computationally expensive: evaluation of just one shield geometry by using all available simulated muons requires an order of 10^3 CPU hours. Considering that the current AMS proposal has 42 degrees of freedom, the search domain of possible solutions is practically infinite. Then it becomes a non-trivial optimisation problem necessitating the use of dedicated search strategies.

Bayesian optimisation (BO) has previously been used as an approach to optimise the geometry of the SHiP muon shield[130]. The fundamental concept involves constructing a probabilistic model of the objective function and using this model to identify the most promising points for evaluation in the actual objective function. Although it was successful in finding a satisfactory optimal shield configuration, some drawbacks of the technique were identified:

- BO performance scales poorly with the number of evaluations: after approximately 1.4×10^3 iterations, the cost of computing new predictions has exceeded the simulation time itself.
- BO algorithms progress linearly, making it difficult (though not impossible[131]) to parallelise the process between multiple computing nodes. This is because the algorithm cannot make the next prediction until the current point has been evaluated.
- Bayesian algorithm struggles with high-dimensional search spaces[132]: this can become a significant problem for re-parametrised AMS configurations with higher dimensionality.

To address these challenges, an optimisation strategy based on a different set of principles was adopted, namely an evolutionary algorithm or EA. There are several properties of EAs that make them attractive for muon shield optimisation:

- The performance of the EA algorithm is independent of the number of iterations, and the CPU cost of the genetic operations is negligible compared to the actual time of evaluating the population.
- In contrast to BOs, the EAs are easier to parallelise on the computing cluster. The size of the population can be increased without significantly slowing down the algorithm, provided there are sufficient computing nodes available.
- Evolutionary algorithms have been shown to be effective in tackling high-dimensional problems[133].

Considering that magnet shape optimisation in SHiP is mainly an engineering design problem, it is worth exploring the method that has been successful in similar applications previously noted in the theory chapter.

3.2 Methodology

Successful EA implementation requires three key ingredients:

1. Genetic representation: the complex geometry of the magnet sections needs to be encoded into a vector of parameters that form the distinctive genetic code of the AMS configuration.
2. Fitness function: the merit of configuration performance. A poorly chosen function can lead the optimiser to produce a solution that has an optimal score but does not solve the original problem adequately.
3. EA architecture: it is desirable to choose an algorithm that is customised to a given problem and that exploits its search space efficiently.

All three aspects have been tackled in the following subsections.

3.2.1 Genetic representation

The current design of AMS uses eight sections of the electromagnet prisms, each expected to be made of non-grain-oriented (NGO) or grain-oriented (GO) electrical steel. The genetic encoding structure is illustrated in Figure 3.1 below. The first two segments of the magnet are fixed and constitute parts of the target and hadron absorber. Thus, only six segments of the magnet undergo optimisation, each defined by seven parameters, representing specific lengths of the magnet prism components. Consequently, the genetic representation of the AMS, referred to as *DNA* for simplification, can be further divided into twelve *chromosomes*. The chromosome L_i denotes the lengths of each magnet prism i, while the chromosome WH_i specifies the respective cross-sectional geometry.

In theory, these parameters can take any positive real value and should be kept as relaxed as possible to give more opportunities for the optimiser to explore. In practice, there are engineering considerations that restrict the domain of geometric vector space. The existing manufacturing proposals for AMS postulate the use of the *step-shield* option, an approximation of the prism shape with rectangular-shaped magnets, each having a length of 50 cm. This restricts the minimum length of the magnet section to 100 cm (for a magnet to have a cross-sectional area difference along its length). The maximum length of each magnet is about six metres, due to space limitations in the proposed experiment installation area and manufacturing constraints. A minimum 2 cm air gap between the opposite limbs with opposite polarity is required to achieve the desired magnetic field profile[134].

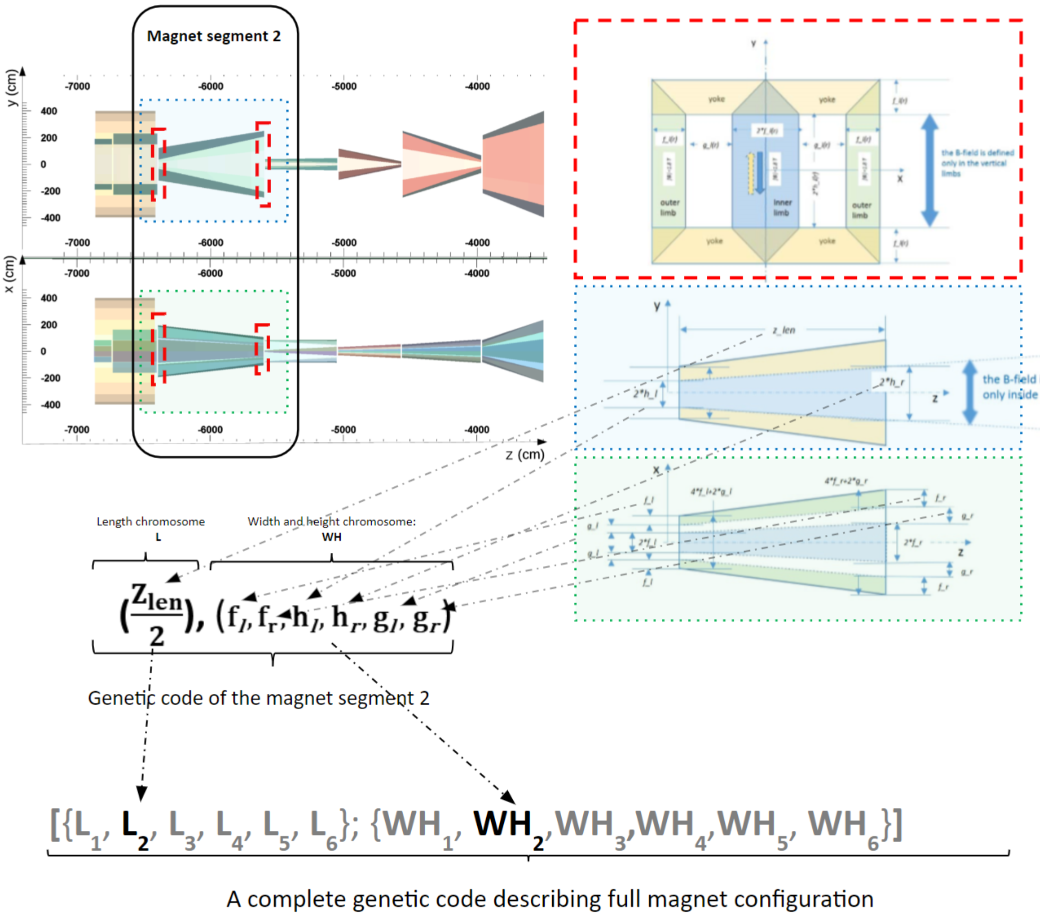


Figure 3.1: A schematic description of the muon shield parametrisation and genetic encoding. Figures are taken from [135]. Please note that the current proposed magnetic field strength has been dropped from 1.8 to 1.6 T.

Reasonable limits were set for other genes on the WH chromosomes to eliminate AMS configurations with unrealistic sizes from the search area. The mutation operator in EA must be adjusted to take these boundaries into account: for instance, if the size of the magnet is already at the minimum of 100 cm, the mutation can either increase this size or remain unchanged.

3.2.2 Fitness function

The ideal muon shield should have the following properties:

- A low mass so the amount of material used and the associated manufacturing cost is reduced.

- It should be short because the long lengths of AMS negatively impact the sensitivity of the experiment to the expected long-lived particle decays.
- The ability to keep the number of muons entering sensitive detector areas to a minimum.

The first two metrics are relatively easy to work out: lengths and weights can be calculated using geometry parameterisation and known values for the densities of the NGO or GO steel material.

Ideally, to calculate a background rate, a full simulation of muon propagation through the virtual geometry of the SHiP experiment would be necessary. The kinematics of muons should be consistent with the anticipated spectra in actual experimental settings, and the sample size should be equivalent to a proton spill of at least one second. The effectiveness of AMS can be assessed by combining particle-weighted deposits in various sensitive regions of the SHiP detection systems, such as SND and HS.

In practice, this is computationally prohibitive, and a more simplified approach is used. The entire simulated muon sample is reduced from $\simeq 1.7 \times 10^7$ to $\simeq 4.9 \times 10^5$. The resampling procedure strongly emphasised the "dangerous" muons, leptons with kinematic properties that have been found to go more frequently through the AMS in previous optimisation studies. The full virtual geometry of the experiment has been reduced to a simplified setup consisting of a target, a hadron absorber, an AMS, and a so-called sensitive plane. This area, measuring 5×10 m, is located 5 metres upstream of the decay volume and coincides with the first tracking station T1: a veto device that tags and identifies incoming background particles.

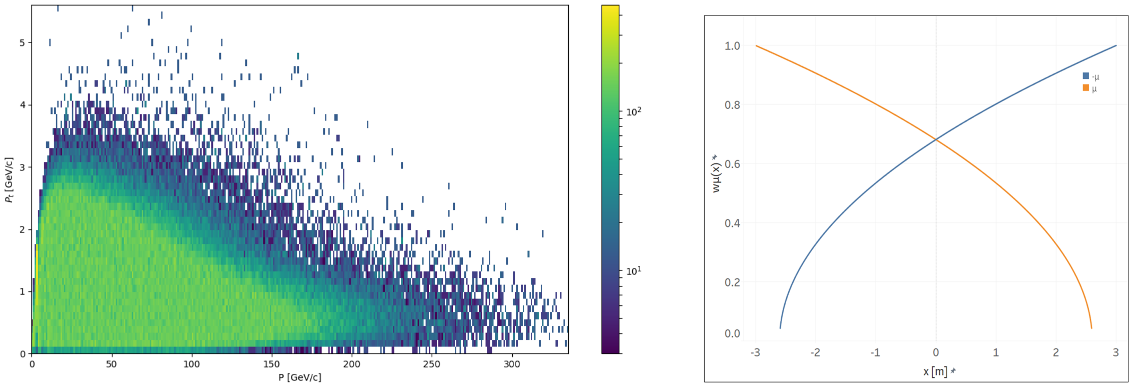


Figure 3.2: **Left:** Transverse momentum versus momentum distribution of 4.86×10^5 resampled muons used for fitness function evaluation. **Right:** The weight function for muon (blue) and antimuon (orange) hits in the sensitive plane area.

The background rate is determined by summing the weights of all simulated hits left by muons with a momentum of at least 1 GeV on the sensitive area. For a muon $\mu^\pm$, the weight function is defined as

$$W_\mu(x_\mu) = \sqrt{\frac{5.6 \mp (x_\mu \pm 3)}{5.6}}, \tag{3.1}$$

where x_μ is the horizontal hit coordinate in metres, and $W_\mu(x_\mu) = 0$ for $|y|$ or changing with x outside the regions as shown in Fig. 3.2.

This function is designed to penalise muons that deviate into areas where they should not occur if the muon shield is functioning properly. For a negatively charged muon, hits occurring at $x \leq 2.6$ m from the plane centre are assigned a weight of zero, since such muons are deflected towards the left side of the plane as expected. The interval $2.6 < x \leq +2.6$ m represents the extended boundary of the sensitive plane with an additional buffer of 0.1 m on each side to penalise the muons in the vicinity of the sensitive area. In this interval, the weight increases continuously as the muon hit position drifts towards the dangerous right-hand side of the plane. To further discourage muons from entering this region, the penalty extends beyond the sensitive boundary: for $2.6 < x \leq +3.0$ m, the weight rises to its maximum value of 1.0, after which it drops back to zero. The situation is analogous for positively charged antimuons, but mirrored with respect to the plane centre.

In order to make the EA results comparable to the previous studies, the following fitness function has been used:

$$f(M, W_\mu(x_\mu)) = \begin{cases} 10^8 & \text{if } M > 3kt \\ (1 + \exp(10 \times \frac{M - M_0}{M_0})) \times (1 + \sum_\mu W_\mu(x_\mu)) & \text{if } M \leq 3kt \end{cases}, \tag{3.2}$$

where M_0 is the mass of the baseline configuration derived in [66] and is ≈ 1.9 kt.

3.2.3 EA architecture

The primary difficulty in implementing an EA lies in the numerous parameters that need to be configured before optimisation. These parameters can be broadly divided into two categories:

1. **EA instances:** global architectural choices such as the size of the initial population, elitism, and selection mode. These choices must be set for the entire optimisation run.
2. **EA parameters:** the parameters of the particular EA instance, such as mutation probabilities and selection size, which can be modified during the run.

From the literature research, it seems that there are three main approaches to configuring EA for an optimisation task:

1. **Take an arbitrary configuration:** very often in papers describing the use of evolutionary algorithms, the parameters are stated without any justification (one example is here [136]). Some publications [137] [138] [139] have been dedicated to studying the problem in more detail and determining the optimal values for some optimisation problems. These results have questionable value for this research, as they were derived by using test functions such as the "K-armed bandits" or the "Knapsack problem". They do not prescribe the best values for EA optimisation of magnetic shields in physics experiments.
2. **Genetic encoding of EA parameters:** an interesting approach discussed in [140], where EA parameters such as mutation rate or crossover mode are added to the genetic code of each individual as an additional chromosome and become the subject of evolution itself. This approach is limited to parameters: it is not clear how to genetically encode EA instances such as initial population size or elitism.

3. **Error and trial**: experiment with different values of the EA parameters on a small subset of a problem to find the most suitable EA. This approach requires additional research work that may not be justified for "one-off" optimisation tasks.

This report has adapted the more advanced version of the last method, *Meta-Optimisation*[141]. The full details can be found in Appendix A. Although this approach requires substantial investment in its optimisation, it is justified for this case, as the optimisation of AMS is a continuous iterative process. The output of the EA hyper-tuning analysis has yielded the following schema:

1. **Initialisation:** Two island pools are equally populated with a total of 400 generated individuals with random characteristics.
2. **Evaluation:** For each individual on both islands, the script sends AMS simulation jobs to the HTCondor batch system: a CERN distributed computing service. On average, it takes about 2-3 hours for each job to complete.
3. **Selection:** Once all individuals have been evaluated, the top 50% from each island is selected for breeding and mutation.
4. **Crossover:** Each individual can undergo a single-point crossing procedure: two individuals are aligned together, the random point on both of their DNA is chosen, and the content on the right of the point is swapped, resulting in two offspring AMS configurations that have a mixture of the parents' DNA. This happens with 20% probability.
5. **Mutation:** Each offspring experiences a mutation process in which its DNA elements are necessarily altered. The new value adheres to a Gaussian probability density function where the original DNA element is the mean, and 2.5 cm is the standard deviation. Nevertheless, the resultant DNA element changes are always rounded to the nearest integer.
6. **Re-injection:** New solutions created in step 4 are released back into the population, and the process restarts from step 2. All evaluated fitness values are cached, which avoids rerunning computationally expensive simulations for the same AMS configuration again.
7. **Migration:** Two islands are not completely isolated: there is a 10% chance in each generation that 5% of the best individuals will travel to another island.
8. **Termination:** The AMS optimisation process spans several days. After each generation, the algorithms outcomes are monitored for progress. If the best fitness value obtained falls below the baseline established through Bayesian optimisation (approximately 29[130]), the optimisation is terminated, and the top-performing individual is retained for detailed analysis.

The model has been implemented in Python. The DEAP library [142] was used as a starting point, but some library components had to be customised to include the Gaussian mutation mechanism and obey the genetic boundary conditions.

3.3 Results

3.3.1 Optimisation run 1

The first run tested the performance of the algorithm using the previously defined fitness function and the muon sample. This was done to compare the EA performance with other alternative AMS optimisers that were used or are currently being used: Bayesian optimisation and *Local Generative Surrogates* (L-GSO)[143].

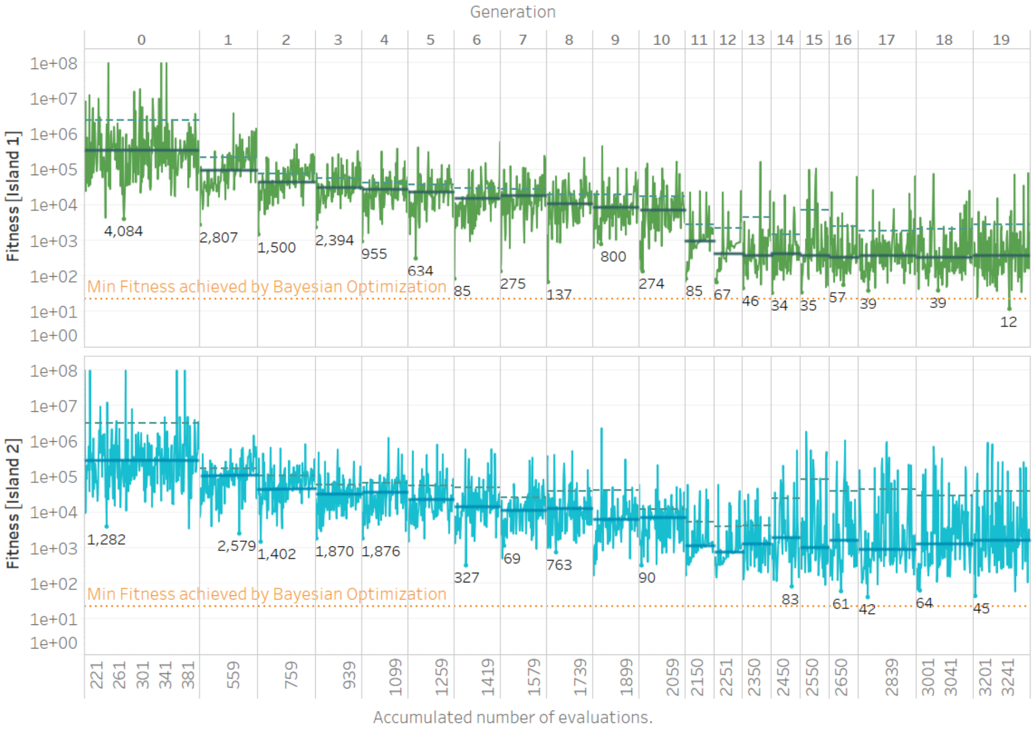


Figure 3.3: Evolution of Island 1 (top) and Island 2 (bottom) over generations. The dashed green and blue lines represent the average fitness, while the solid lines denote the median fitness to mitigate outlier bias. The yellow dashed line represents the fitness achieved by the previous method using Bayesian optimisation. Between generations 11 and 16, the selection size was manually decreased to impose stronger selective pressure.

The optimisation process took approximately two weeks and was terminated when a solution (EA1) exhibiting a fitness value of 12, which is below the Bayesian optimisation baseline, was identified and selected for further study. The visualisation of geometry has shown a very different configuration from the BO baseline (BO1). Quantitatively, there are substantial differences as well: the EA1 has a length of 38.08 m versus 35.44 m of BO1 and a mass of 1.92 kilotons versus the BO1 value of 1.27 kt. Therefore, EA1 is longer and heavier than BO1, meaning its low fitness function is mainly driven by muon deflection performance.

Another simulation was conducted using the same basic geometric arrangement, but with the most comprehensive existing sample of Monte Carlo-generated muons, for three different configurations: EA1, BS1, and L-GSO1. The results of the analysis demonstrated that the EA1 configuration exhibited significantly superior filter performance compared to the BO1 candidate and L-GSO1. EA1 allowed for an order of magnitude fewer background hits in the sensitive plane of the First Tracking

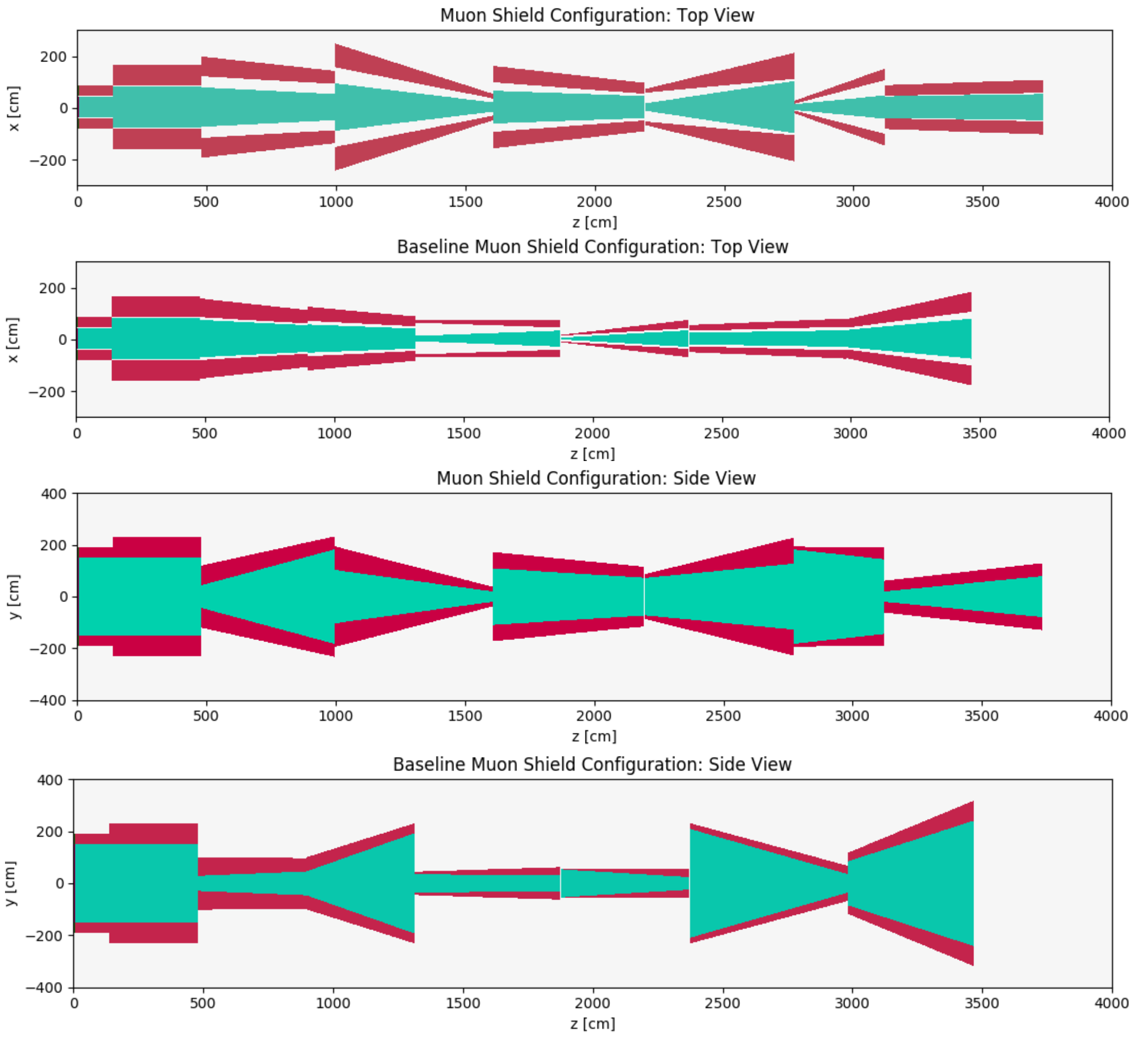


Figure 3.4: The EA candidate and the baseline Bayesian optimised design visual comparison. In the top view, the inner limb of the magnet is represented in bright turquoise colour, and the outer limb is represented in ruby red. In the side view, turquoise is used for both the inner and outer limbs, while the magnet yoke is depicted in ruby.

Station than BO1, and two orders of magnitude fewer hits than L-GSO1[1].

3.3.2 Optimisation run 2

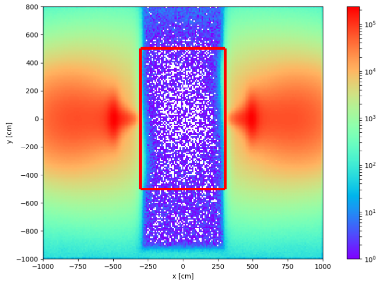
The second optimisation run has been initiated with the following objectives:

- To test whether the given EA instance consistently delivers optimal results.
- Assess the robustness of the EA instance to slight changes in the fitness function: the geometry shape has been rendered using a *step approximation* where the magnet shield shape is made of 50 cm long rectangular blocks. Although the definition and parameterisation of the fitness function have remained the same, this option changes the AMS response to the incoming muon stream.
- Modify the EA output to include weight and length information so that the geometrical aspects of evolution can be studied.

---

[1] L-GSO1 is derived from the BO1 geometry but is shorter and lighter.

L-GSO



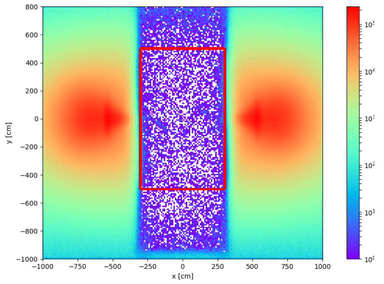
μ Hit loss:

$5.6 \pm 0.03 \times 10^4$

Length: **34.79 m**

Mass: **1.15 kt**

Bayesian



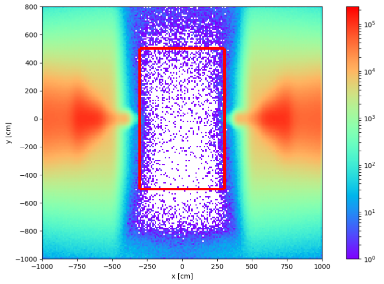
μ Hit loss:

$7.2 \pm 0.09 \times 10^3$

Length: **35.4 m**

Mass: **1.27 kt**

EA Run 1



μ Hit loss:

$3.2 \pm 0.05 \times 10^2$

Length: **38.08 m**

Mass: **1.92 kt**

Run 2 has taken approximately the same amount of time as Run 1 and has yielded comparable results. However, this time the evolution profiles of Island 1 and Island 2 are more similar: this could indicate a bigger role of the migration operator. Another interesting feature is the effect of manually adjusting the selection size for some generations: compression leads to a bigger spread of fitness values within a generation. Overall, the periodical change in the selection pressure does seem to have a beneficial effect on the optimisation progress: after each change, the minimum values of fitness started rapidly decreasing.

Although both Run 1 and Run 2 have converged to similar fitness values, the solutions are conceptually different from each other. The Run 2 solution with the lowest fitness value, EA2, has a different geometry compared to EA1: It is lighter, less massive, and has an unusual morphology. This shows that evolutionary algorithms do indeed lack any design bias by following the principle "If it works, it works".

Further examination of the geometry evolution has revealed the following picture, which can be interpreted as follows. At first, the initial population is randomly scattered across a weight-length phase space. This distribution is not uniform, but it follows a degree of positive linear correlation between the mass and length of the AMS. After a few generations, the evolution enters an exploration phase in which various areas of the search space are probed. The area is limited to length ranges of 25-35 metres and to masses between 1 kt and 2.2 kt. The fact that Run 1 has yielded a solution that lies well outside of the exploration domain in Run 2 indicates that the algorithm has a high degree of stochasticity, which is driven by the population initialisation process. It also indicates that the search space is vast and potentially contains more optimal solutions. After 17 generations, a strong family of similar individuals starts dominating a whole population, which eventually converges into a single clump with an average mass of 1.7 kt and a length of 33 metres.

Diversity collapse is a well-known EA problem, and while it does not cause signif-

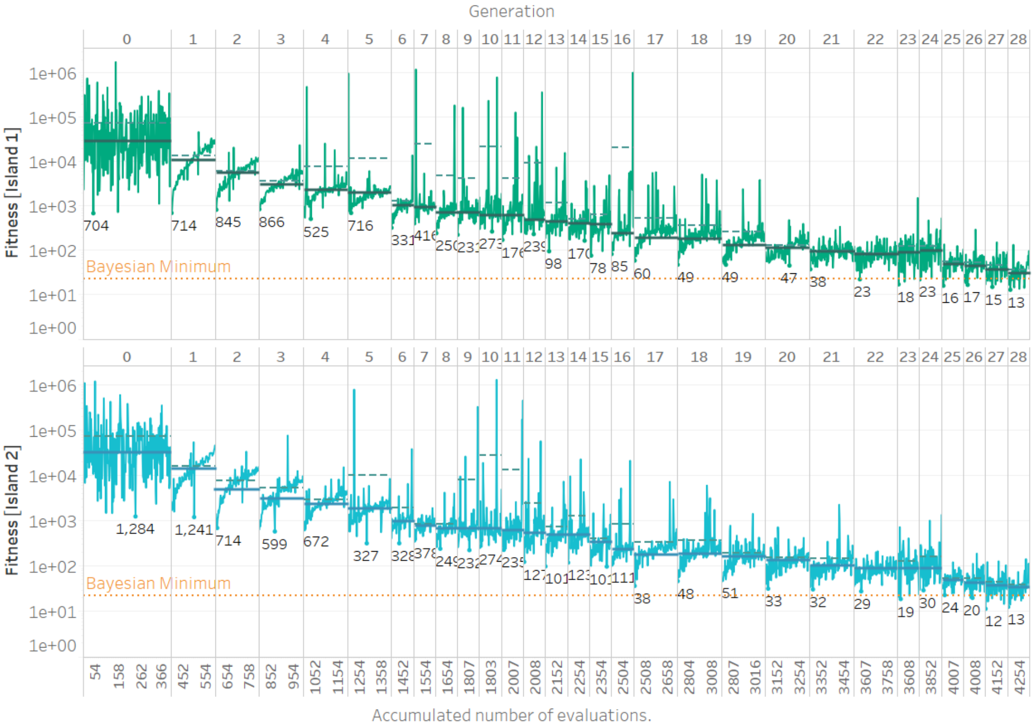


Figure 3.5: The Run 2 evolution of Island 1 (top) and Island 2 (bottom) over generations. The dashed green and blue lines represent the average fitness, while the solid lines denote the median fitness to mitigate outlier bias. The yellow dashed line represents the fitness achieved by the previous method using Bayesian optimisation.

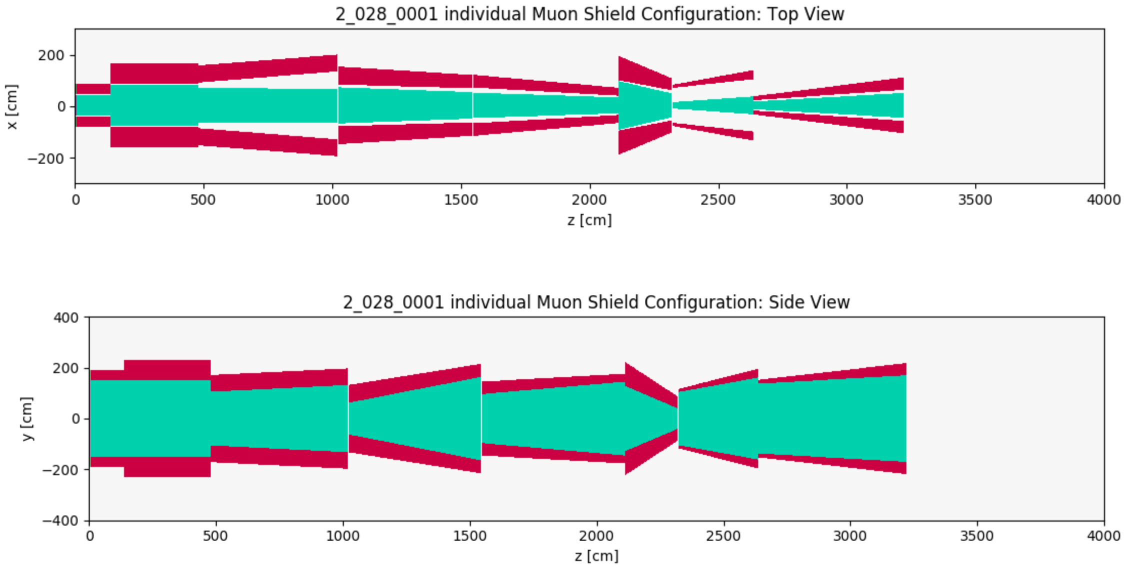


Figure 3.6: The visualised geometry of the EA2 candidate. In the top view, the inner limb of the magnet is represented by turquoise colour, and the outer limb is represented by ruby red. In the side view, turquoise is used for the inner and outer limbs, while the magnet yoke is depicted in ruby.

icant issues for AMS optimisation at this stage, it can become a problem in future extensions of the EA model [144]. It is also important to remember that it is a highly simplified picture, where the 42-dimensional space is reduced to two dimensions.

Figure 3.7: The evolution process of Run 2 on the mass-length phase space read from top left slides clockwise. The colour indicates the island to which an individual belongs. **Top Left:** The distribution of lengths (in metres) and masses (in kilotons) in the initial population. **Top right:** In the fifth generation, multiple small congregated spots of evolving individuals are observed. **Bottom right:** After 17 generations, a shift towards larger lengths is observed along with a decrease in diversity. **Bottom left:** The last generation is condensed into a single cluster.

Overall, the second run results met most of the initial expectations and confirmed the validity of the chosen EA configuration. At that time, several potential model improvements were identified, such as increasing the total population size beyond 400 individuals to ensure better coverage of the weight-length phase space and reduce run-to-run variability, and programming the EA model to monitor evolutionary progress and adjust the selection size automatically, thereby limiting the need for manual intervention. However, these improvements were not implemented, as the research was concluded shortly thereafter.

3.3.3 Optimisation run 3

So far, the optimisation has been concerned with using the static data set consisting of 4.86×10^5 muons. This simulation is unable to accurately replicate the kinematic spectrum of muons that would be observed in a real experiment, because of its limited size. This can lead to a situation where the performance of AMS is optimal only for a specific kinematic distribution of muons but significantly degrades under real-life conditions.

As an alternative to the CPU costly increase in the number of muons, a new approach has been tested, which in principle could yield muon shields that are robust to changes in the muon spectrum. The main idea of the approach is to make the AMS and the muon spectrum evolve together and compete with each other, so only robust AMS geometries are generated as a result. The algorithm used is more complex than for Run 1 and Run 2 and has an extra step, which is concerned with the evolution of the muon spectrum, as shown in the figure below:

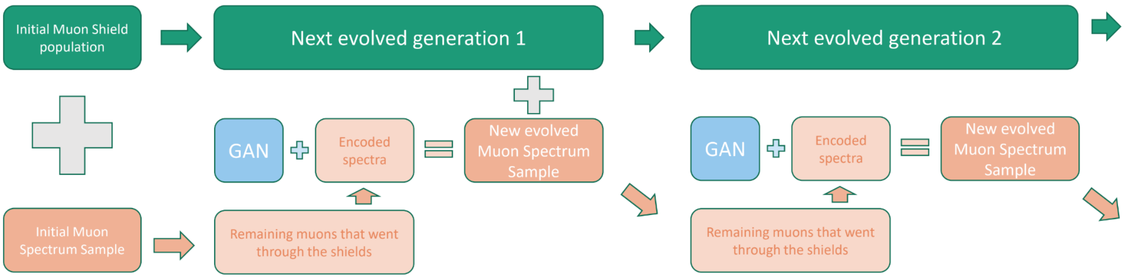


Figure 3.8: A schematic diagram representing the Run 3 optimisation workflow.

The breakdown of the optimisation cycle was as follows:

1. **Initialisation:** two island pools were equally populated with a total of 600 individuals with random characteristics similarly to previous optimisation runs, but with a 50% increase in the number of initial population.
2. **AMS evolution cycle:** this included standard steps involved in the evolution cycle, such as evaluation, selection, cross-over, and mutation. For the initial population, a nominal resampled muon data set was used to calculate the loss function.
3. **Muon sample adjustment:** this is the additional step that involved collecting all muons that had evaded the AMS of the current generation and ended up in the acceptance area. Since the total number of muons collected could be significantly less than the required amount of $\approx 5 \times 10^5$, an additional tool has been used to fast-simulate the muon samples. It has been specifically developed for the SHiP experiment[145] and is supported by *generative adversarial networks* (GAN). GAN is a machine learning method that involves two rival neural networks: the *generator*, which seeks to create an image, and the *discriminator*, which aims to determine whether the image is authentic or counterfeit. It is commonly used to create or improve images and videos, but at SHiP, GANs have been used to expand small muon samples into larger datasets. Since GAN-based muon simulation is much faster than traditional Monte Carlo generators, it has become the preferred simulator for accelerated muon spectra evolution. After integrating the remaining muons into a GAN model, a new muon sample is created for each iteration of EA, matching the original sample in the number of simulated muons, but with modified spectra. This freshly generated muon sample is subsequently utilised to calculate the loss function for the upcoming AMS generation.
4. **Termination:** after a few cycles of subsequent AMS and muon sample evolution, the run has been terminated manually.

Although Run 1 and Run 2 have been characterised by rapid improvements in the average fitness of the AMS population over consecutive generations, Run 3 has shown the opposite picture. For both islands, the average fitness function has been continuously increasing with each generation, with the last generations performing worse than the first one.

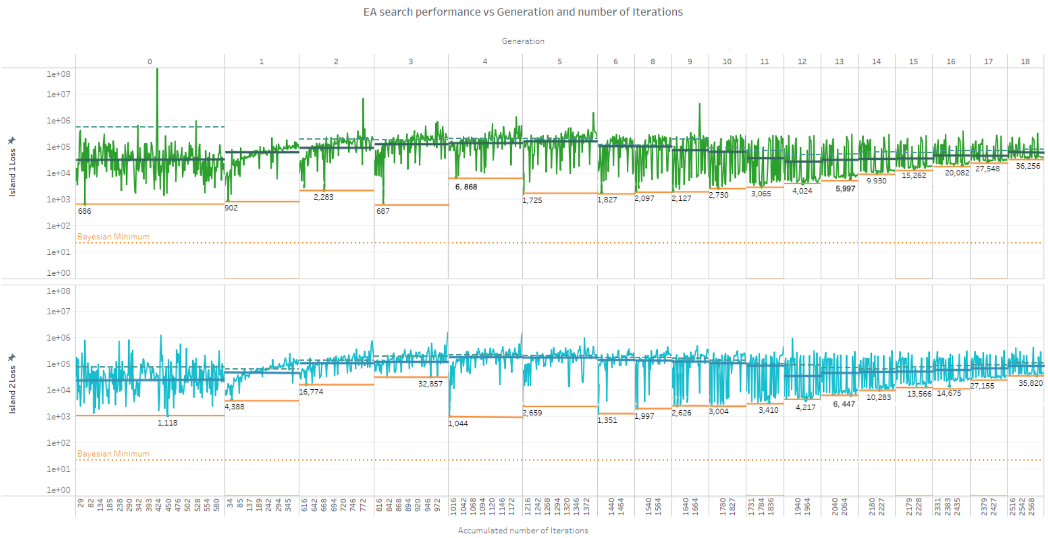


Figure 3.9: The Run 3 evolution of Island 1 (top) and Island 2 (bottom) over generations. The dashed green and blue lines represent the average fitness, while the solid lines denote the median fitness to mitigate outlier bias. The yellow dashed line represents the fitness achieved by the previous method using Bayesian optimisation. Please note that data for Generation 7 is missing due to a technical problem that occurred during the evolution run.

In order to understand the exact cause of this degrading behaviour, a selected series of samples of evolving muon spectra has been plotted below and interpreted as follows:

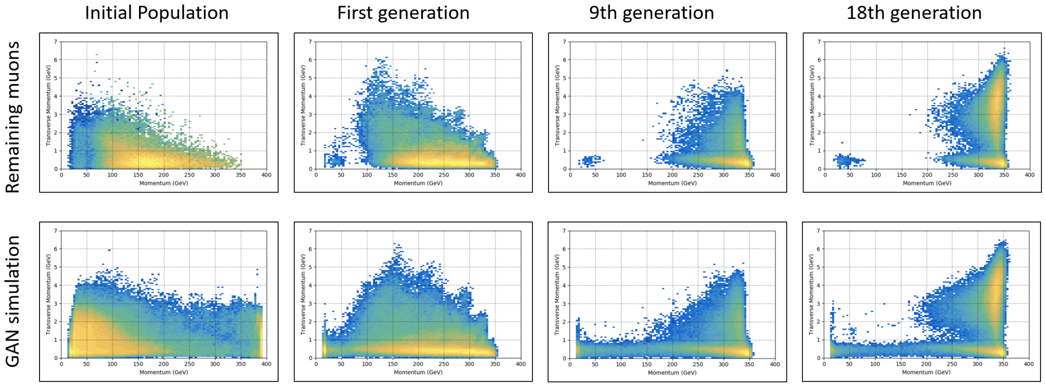


Figure 3.10: A few selected steps of the muon spectra evolution observed in Run 3.

- The initial population of the muon shields consists of randomly generated designs, and therefore, the overall kinematic profile of the collected muon follows the one in the baseline sample.
- Since the consecutive generation has undergone selection, it starts to deflect a significant amount of muons, causing a gap in the 50-100 GeV range. Two different regions of the phase space of the muon momentum can be identified:

muons with a total momentum of less than 50 GeV and those with high energy, which feature both an overall momentum greater than 100 GeV and a transverse momentum ranging from 1 to 7 GeV.

- The gap increases with each generation, where the later generations of AMS learn to screen muons within the 50-250 GeV range effectively. The muon spectrum adjusts rapidly, causing the absolute momentum of newly generated muons to increase.

Intuitively, this behaviour can be understood as AMS evolution having a slow response to the drastic changes in the muon spectrum. Consequently, evolved populations in consecutive generations fail to adapt to the increasing flux of highly energetic muons, which is, in fact, the expected outcome: the optimisation could not converge in principle, as the probability of identifying a universal shield design capable of deflecting any possible muon spectrum is extremely low. Instead, this run served as a pilot study to push the evolution-GAN framework to its limits and to identify potential operational problems. One such issue concerns the quality of the GAN-based muon simulation, as the tails of the input sample spectrum appear to be unrealistically exaggerated in the generated output.

3.4 Conclusion

3.4.1 Summary

In summary, a set of three optimisation procedures for the active muon shields has been carried out, each of which had its own objectives.

The first pilot run after $\approx 3k$ iterations has yielded an AMS geometry with a superior shielding performance compared to other optimisation techniques, although at the cost of having a heavier geometry. However, those optima perform well only on unrealistic problems with simplified geometry, an idealised magnetic field, and a small muon sample size.

In order to address the problem of the simplified geometry, the second run has been performed by using the magnet step approximation. The EA algorithm has again converged to the local optimum point with a fitness value similar to Run 1, but after a greater number of $\approx 4k$ iterations.

In Run 3, the principle of co-evolving different muon shield geometries together with muon kinematic distributions was tested as a way to address the challenge of the limited muon sample size identified earlier. Although the co-evolution procedure was, as expected, unable to converge, with successive generations of the muon shield failing to adapt to the concurrently evolved spectra of dangerous muons, the process itself ran smoothly overall and showed that, aside from minor technical adjustments, the framework can be reliably executed.

Although evolution strategies clearly have demonstrated some advantages of the evolutionary algorithm optimisation speed and the quality of converged solutions, their true capabilities have never been tested in real-life conditions. Only the evolved AMS design from the first run has been tested on the full muon sample using a simplified SHiP setup, and it does not include other background contributions such as muon rescattering and electron bremsstrahlung radiation.

The fitness function has a major limitation in that it does not take into consideration the muon background in other detector systems, such as SND. The solution to this problem necessitates not only the development but also the agreement of relevant parties, such as detector groups, on a metric that is suitable for all. Unfortunately, the European Strategy for Particle Physics report in 2020 did not prioritise the SHiP experiment, which led to a temporary suspension of its implementation in the proposed form. In the meantime, efforts were redirected towards the SND@LHC experiment, which, while more limited in scope, offered a simpler setup and the opportunity to collect data during LHC Run 3. In 2024, however, SHiP was finally approved at a different location at the LHC, which allowed AMS activities to resume; however, due to the placement in a different tunnel and specific site restrictions, the muon shield design had to be substantially revised to meet the new operational requirements.

3.4.2 Future research

The direction of further research critically depends on the constraints of the SHiP detector components in the new location. The application of evolutionary algorithms is justified if the future active magnetic shield geometry has a high number of degrees of freedom, as the model studied in this thesis does. If this is the case, the following key aspects of muon shield optimisation could be the subject of further research.

AMS reparametrisation

If the geometry of the muon shield uses a step approximation, then the prismatic shape of the magnets could be modelled as a set of rectangular blocks. The muon shield can be reparameterised in terms of numerous 50 cm long individual magnets, each having its own geometrical characteristics that can be varied independently. This will provide more degrees of freedom to the AMS configurations and can result in an additional number of optimal solutions that were previously unattainable.

Accurate fitness function

Previously, it was mentioned that the fitness function has many drawbacks that prevent it from accurately representing many of the effects expected in the actual operation of the AMS. The challenge is to find the right balance between its accuracy and its computational time: the definition of the loss function will be useless no matter how precise it is if the time needed to calculate it is too long.

Competitive evolution

If this method is to be explored further, the next logical step would be introducing small controlled spectrum variations based on the response of the previous muon shield generation. For example, only 10-20% of the muon spectrum could be altered per generation, focusing on the momentum phase space where the shield is less effective. This approach would challenge the next generation to adapt to changing conditions while avoiding large, catastrophic shifts in the overall charged particle spectrum, potentially allowing the AMS population to gradually improve both its shielding capability and resilience to variations in the incoming muon spectra.

Hybrid approach

The typical optimisation can be divided into two distinct phases: the exploration mode, in which the solution space is sampled for potential solutions, and the tuning mode, in which the selected solutions are incrementally improved.

The current approach of using the same EA architecture for both phases is not ideal, as the two phases have distinct differences that require a more individual approach. For example, in the exploration phase, it is preferable to have low selection pressure, large step sizes for genetic operators, and different selection strategies such as the tournament. This could prevent premature convergence of the entire population to a single group of genetically similar solutions. The tuning phase, on the other hand, has opposite requirements, such as small steps in the genetic operators to prevent the potential solutions from oscillating around the optimal point, high selection pressure, and higher contributions from the crossover operators.

A more agile approach would be to develop two distinct EA architectures, each tailored to its specific task. Partially, this has already been realised in the Run 1 and Run 2 optimisation cycles, where a higher selection pressure has been applied to the later AMS generations.

An interesting idea discussed during one of the SHiP collaboration meetings was to combine evolutionary algorithms with other optimisation approaches, such as L-GSO[143]. In this proposal, the EA is solely responsible for exploring the search space for potential solutions, while the L-GSO uses these solutions as input and investigates the local optima in their vicinity.

Chapter 4

Application of machine learning to tracking and vertexing in the SND@LHC experiment

4.1 Motivation

The main element of the FEDRA data reconstruction process is a Kalman filter[89], which was discussed in the theory chapter. This algorithm is highly effective in dealing with noisy data, as it does not require the storage of all previous measurements, since the most recent internal state accumulates all the past history at each step. These particular qualities made it popular in applications that involve navigation and control of vehicles such as cruise missiles, spacecraft, and marine navigation, where memory and computing power can be limited.

In the context of particle physics, the Kalman filter was first implemented in 1989 at the DELPHI detector at LEP[1] [146] and since then the KF technique has remained the main backbone of online data event reconstruction on major high-energy physics experiments such as ATLAS or CMS. Despite this strong track record, the Kalman filter is known to be limited by two key assumptions: the linearity of the particle propagation model and the Gaussian distribution of the noise processes, which is not necessarily accurate for all particle detectors. The ECC brick used at SND@LHC is a good example of such a system: it is a complex environment made of dense tungsten plates and emulsion films that are well-known to deflect and distort the path of the passing charged particles, especially in the low-momentum regime of 0-20 GeV.

In combination with the high density of the LHC recorded data and uncertainties introduced by the alignment procedure, the LHC detector data present a significant challenge for linear estimator algorithms. Even when conducting research on low-hit density, idealised Monte Carlo simulated sets with a perfect alignment configuration, four key issues have been identified with the standard reconstruction output. These will be discussed in further detail below.

---

[1] LEP was a Large Electron Positron collider at CERN that has been eventually decommissioned in 2000 and replaced with LHC.

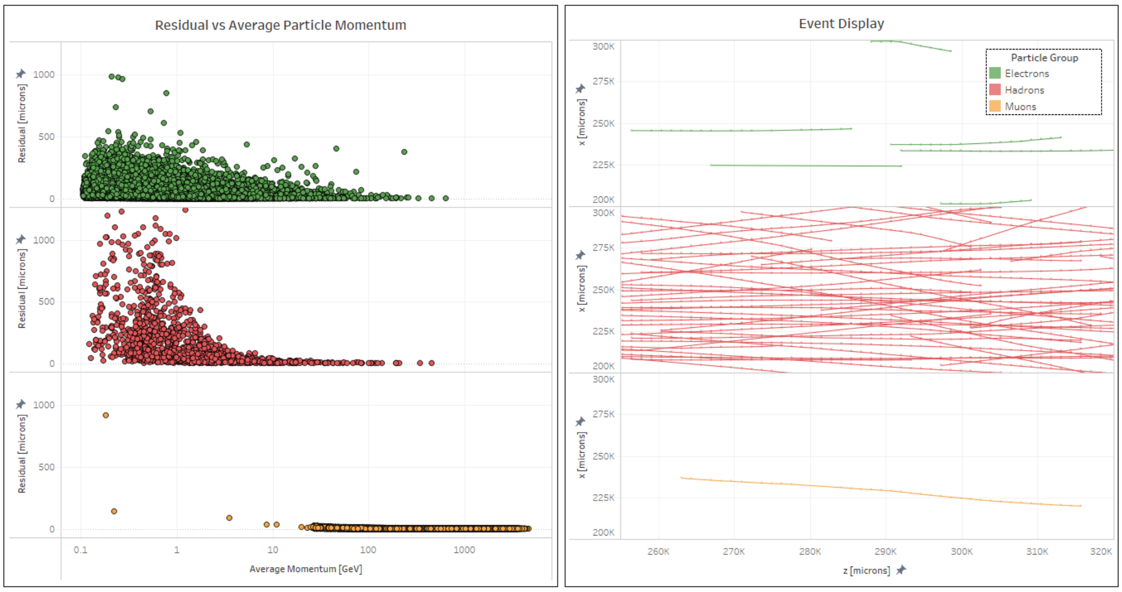


Figure 4.1: A study of the MC simulated interactions in one of the SND@LHC ECC units (B31) where the data is split by main particle groups: electrons, hadrons, and muons. **Left:** Residual-momentum scatter plot where each coloured circle represents a unique particle. The residual represents the average deviation of the particle's trajectory from the fitted straight line, while the momentum is calculated by measuring the momentum across the particle's path and taking its average value. **Right:** An event display projected on an xz-plane with visualised particle tracks.

4.1.1 The integrity of the reconstructed tracks

Ideally, each charged particle that passes through the emulsion detector should be identified and represented as a single track. In reality, the trajectory of a particle can be composed of multiple tracks, each representing a segment of the particle's journey.

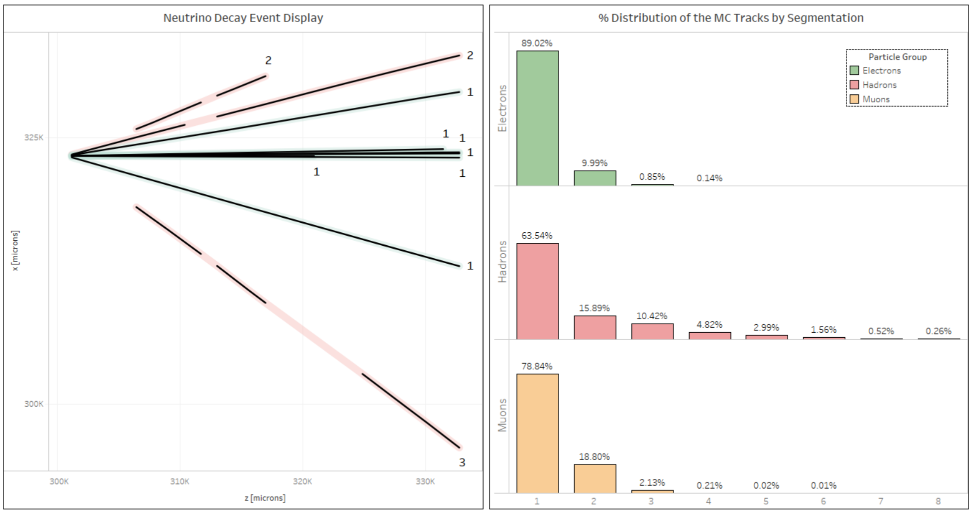


Figure 4.2: **Left:** An event display projected on an xz-plane with visualised simulated muon neutrino decay with a production of daughter particles. The transparent backgrounds represent the true path of the particle, while the black lines represent the tracks reconstructed by FEDRA. The numbers at the ends quantify the segmentation of the particular particle MC track. **Right:** percentage distribution of MC tracks segmentation for the three main particle groups.

Quantitatively, this effect can be characterised by a *segmentation* metric, defined as the number of reconstructed segments per truth particle. For a single particle, the segmentation value can range from one up to the maximum number of segments that can be reconstructed within a given ECC unit. The phenomenon of track segmentation in emulsion is inherently complex and arises from a variety of factors that contribute to the issue:

1. **Holes:** Missing hits in ECC on one or more consecutive plates can cause the FEDRA tracking algorithm to stop extending the track if the maximum tolerable number of gaps has been encountered. It will still attempt to reconstruct the remaining segment, but the latter will end up in the output as a separate track. In some cases, these gaps are caused by the limited detection efficiency of the plate, which, for the data set in question, is modelled at about 85%. In theory, the chances of encountering two or more consecutive gaps are small ($\approx 2.25\%$ and less), but even a single hole causes a lag in the Kalman filter's knowledge of the particle state, which can result in rejecting the following hit. In addition to the actual holes, there can be "perceived" holes at plates where the hits are present, but their measured base-track variables, such as slopes or positions, are outside of FEDRA acceptance.
2. **Track length:** At the SND@LHC experiment, the reconstructed tracks in the emulsion must have a minimum of four[2] constituent hits to be accepted for data analysis. If an incident particle leaves fewer than eight hits in the emulsion, the FEDRA reconstructed track will have a maximum segmentation of one[3]: If at least four hits are identified, the remaining segment will not meet the minimum hit criteria. As the length of the trajectory of the passing particle increases, so does its maximum theoretical value of segmentation, which can be approximated by the formula $Seg_{max} = \frac{N_{TrHits}}{TrHits_{min}}$ where a numerator is the total number of hits that the particle deposits in ECC and the denominator is the minimum accepted number of hits per reconstructed track.
3. **Particle trajectory:** The more predictable the trajectory of a particle, the easier it is for the tracking algorithm to model its behaviour and produce sufficiently accurate predictions for the next hit. Not all particles act in the same manner in the ECC; if their mass or momentum is too low, the alterations to their trajectory, due to the interaction with the detector medium, like Coulomb scattering, can be too great for the Kalman filter to adapt.
4. **Electromagnetic activity:** High-energy charged particles often produce secondary ionisation electrons (delta-rays) or initiate bremsstrahlung and electromagnetic showers inside the emulsion. These secondaries leave additional base tracks close to the primary trajectory. FEDRA may occasionally incorporate such hits into the evolving track fit, leading to a sudden change in the fitted parameters. Once the fit diverges, the algorithm terminates the current segment and attempts to reconstruct the remaining trajectory as a separate track. This results in artificial segmentation even when the true underlying trajectory is continuous.

---

[2] A criterion has been agreed by the SND@LHC collaboration in order to exclude tracks coming from the EM showers.

[3] In some cases, this can be exceeded if the reconstructed track contains hits from other particles.

There are three main groups of charged particles that are reconstructed in the emulsion detector: hadrons, electrons, and muons. Table 4.1 below illustrates the effect of segmentation on different groups of particles and their sources.

Table 4.1: High-level statistical description of the data used for this research, split by particle group and its origin, the full details of which can be found in Appendix C. The average segmentation is only calculated for tracks that have been at least partially reconstructed by FEDRA.

Particle source	**Particle group**	**% of all tracks**	**Average momentum (GeV)**	**Average residual (microns)**	**Average number of constituent hits**	**Average segmentation**
EM showers and delta-rays	Electrons	71.82 %	0.86	43.2	5.00	1.35
External muons from IP1	Muons	27.01 %	1,150.08	0.7	50.76	1.28
Muon neutrino interaction	Muons	0.02 %	271.69	1.9	50.00	1.04
	Hadrons	0.06 %	35.19	105.9	32.00	1.33
$\mu/e/\gamma$ **nuclear interaction**	Hadrons	0.27 %	6.05	284.3	19.00	1.67
Hadronic interactions	Hadrons	0.82 %	3.86	255.6	17.00	1.80

Muons are predominantly produced in high-energy collisions at the LHCs interaction points, and as a result, they typically carry significant momentum, often averaging in the TeV range. Even in this regime, muons lose relatively little energy while traversing the ECC, producing straight, nearly parallel tracks that extend across the full detector. However, these through-going muons also induce electromagnetic activity and parasitic delta-ray tracks, which, when combined with their considerable track length and the presence of holes, drive the muon segmentation upwards, thereby inflating the number of reconstructed muon tracks observed in the data.

From the analysis point of view, external muons have no physics discovery value apart from being used in the alignment procedure and are treated as a background. This background can be eliminated by removing any tracks that start at the first plate: only interactions that occur within the ECC unit are considered; this solution, in principle, should be removed up to $\approx 85\%$ of the entire muon background[4], the value of which is limited by the emulsion detection efficiency.

Unfortunately, for FEDRA reconstructed tracks, the percentage of them that can be removed is at a much lower value of around $\approx 67\%$ due to the segmentation: If the track is split into two or more segments, only the first fragment will be removed, leaving the remaining segments in the data. In addition to overloading the vertexing algorithm with unnecessary tracks to process, these leftover segments are a potential source of random crossings that can create fake or spoil existing signal vertices. Therefore, it is desirable to reduce muon segmentation as much as possible.

Apart from increasing muon segmentation, the muon-induced electromagnetic showers also generate substantial cascades that are deposited in the emulsions. By track count, the electrons produced in these showers are expected to contribute a

---

[4] Inevitably, it can also remove tracks caused by the signal interactions in the preceding ECC units (if applicable), but this should not cause any concern since only interactions coming from the ECC detector of interest are considered.

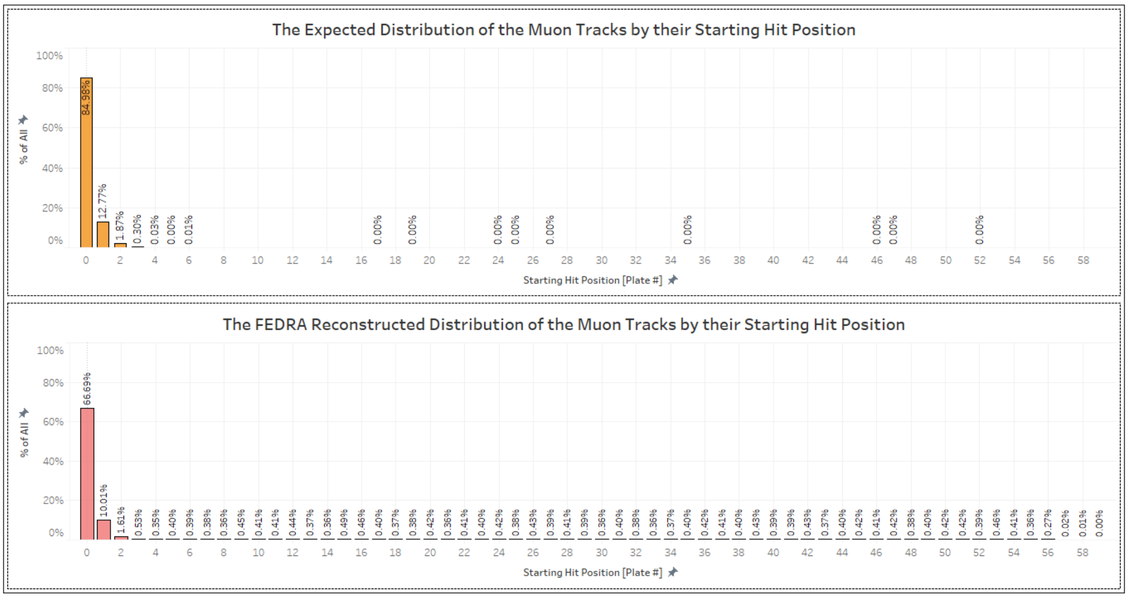


Figure 4.3: The percentage distribution of tracks by the plate according to Monte-Carlo simulation truth (Top) and the FEDRA reconstructed (Bottom) track origins.

background comparable to that of external muons in the ECC detector. To mitigate their impact, reconstructed tracks are required to contain at least four hits. Typical shower electrons have low momentum, often below 5 GeV, and small radiation lengths, which result in short trajectories. Their tracks are highly non-linear, with average residual values nearly forty times greater than those of incoming muons. Nevertheless, because their average hit content is small, their segmentation values remain relatively modest and comparable to those observed for external muons.

It is important to note that the Monte Carlo-generated data samples available at the time of writing this thesis consisted only of muon neutrino signals. The SND@LHC physics programme also includes the detection of electron neutrinos. This raises the issue of track segmentation, which is a difficult task due to the low mass of the electron (around 511 keV). This makes it more susceptible to the electric fields of the atoms in the emulsion and tungsten layers, which can cause its trajectory to deviate, making it difficult to accurately reconstruct the electron's path.

The remaining hadron particle group consists mainly of pions and protons with a small number of other hadrons, such as kaons and strange baryons. Hadrons are expected to produce less than 2% of all tracks in the emulsion detector data, but they are essential for the analysis, as the majority of signal signatures involve hadronic particles produced in the DIS processes and their secondary decays. Their low momentum in combination with relatively large lengths makes them susceptible to the majority of the factors listed above, which results in the highest average segmentation of all particle groups. The signal event reconstruction is greatly affected by this, as it can cause a considerable distortion of the actual parameters of the reconstructed vertex, including its multiplicity.

4.1.2 Track reconstruction quality

Segmentation is not the only criterion that the tracking algorithm should satisfy. In this investigation, three other metrics are used to assess the quality of the reconstruction.

1. **Track reconstruction efficiency:** this parameter quantifies the number of suitable MC simulated tracks that were at least partially reconstructed by the tracking algorithm. Only tracks that contain four or more hits generated by the simulation are considered in the calculation.

2. **Track hit reconstruction efficiency:** this criterion quantifies how complete the FEDRA track is compared to its MC-generated counterpart. For example, if, according to the Monte Carlo simulation, a given track should have 20 hits, but the reconstructed track has only 15 hits, then the *track hit reconstruction efficiency* or THRE will be 75%. This metric has a degree of correlation with segmentation: the higher the segmentation, the lower the THRE since each constituent reconstructed track segment will have a lower number of matched hits. However, the reverse is not necessarily true: the low THRE can be observed for tracks without segmentation, where the remaining hits were ignored during the tracking procedure.

3. **Track hit reconstruction purity:** in the example mentioned above, the FEDRA counterpart had only 15 hits that corresponded to the MC simulated track; however, the length of the FEDRA reconstructed track can be greater, for example, 25 hits. This usually happens when hits belonging to two or more different particles have been merged in a single track. The *track hit reconstruction purity* or THRP is calculated simply by dividing the number of matched hits by the number of total hits that make up a reconstructed track. In this example it is simply calculated as: $\text{THRP} = \frac{15}{25} = 60\%$

The evaluation of the FEDRA tracks performance on the Monte-Carlo simulated data has yielded the following picture:

Table 4.2: FEDRA track reconstruction performance split by particle group and its origin taken from the calibration domain. Average values are calculated for tracks that have been at least partially reconstructed by FEDRA.

Particle source	**Particle group**	**Number of simulated tracks**	**Overall track reconstruction efficiency (%)**	**Track hit reconstruction efficiency (%)**	**Track hit reconstruction purity (%)**
EM showers and delta-rays	Electrons	114,555	39.7 ± 0.2	42.1 ± 0.2	62.0 ± 0.2
External muons from IP1	Muons	43,000	98.2 ± 0.05	80.2 ± 0.1	98.2 ± 0.0
Muon neutrino interaction	Muons	25	$96.0^{+4.0}_{-20.0}$	95.2 ± 4.0	99.8 ± 0.1
	Hadrons	91	$91.2^{+8.8}_{-10.5}$	68.7 ± 3.8	97.1 ± 0.9
$\mu/e/\gamma$ **nuclear interaction**	Hadrons	433	65.4 ± 4.5	45.7 ± 1.8	99.2 ± 0.4
Hadronic decays and inelastic scattering	Hadrons	1,306	61.4 ± 2.6	37.9 ± 0.9	99.6 ± 0.1

The first significant insight from the table is that FEDRA has difficulty tracking electrons, with a tracking efficiency of less than 40%, and the quality of the reconstructed tracks is very poor. As previously mentioned, in the context of muon-neutrino interaction physics, electrons are of secondary importance; however, the ability to reconstruct electromagnetic showers and their constituent electrons will become important in searches for dark photons and in the detection of electron neutrinos.

The muons, on the other hand, demonstrate the best overall performance in terms of track utilisation and the quality of the reconstructed tracks. Nevertheless, some nuances can be observed. The first is a slight discrepancy of approximately 2% between the track reconstruction efficiency of muons originating from the interaction point (IP1) and those emerging from ν_μ interactions. The latter exhibit a somewhat lower TRE; however, due to the limited statistics for neutrino-induced muons (only a few dozen tracks), the associated uncertainties are large, making it difficult to draw a definitive conclusion. More striking is the behaviour of the track hit reconstruction efficiency. Here, external IP1 muons exhibit a significantly lower efficiency of $\approx 80\%$, which is about 15% below that of muons from neutrino interactions.

The reason for this somewhat counterintuitive behaviour is the same as for segmentation: TeV-scale muons generate a higher density of secondary delta-rays and, to a lesser extent, bremsstrahlung photons along their path. FEDRA may inadvertently include some of these secondary hits in the reconstruction, causing the main track to diverge and segment, which in turn reduces the effective hit utilisation. In contrast, muons from neutrino interactions typically have lower momenta (200-300 GeV), which are still sufficient to maintain nearly linear trajectories while producing far fewer delta-rays and electromagnetic secondaries. Consequently, FEDRA can follow these tracks more continuously, resulting in a higher THRE.

The picture for hadrons is more straightforward, but less favourable than for muons. Being more massive, hadrons in this dataset generally carry relatively low momentum. For instance, hadrons produced in muon neutrino interactions have average momenta of approximately 35 GeV, whereas those originating from secondary nuclear or hadronic interactions carry only a few GeV on average. Their low momentum makes hadrons particularly susceptible to Coulomb scattering within the emulsion medium, which renders their tracks non-linear and challenging for FEDRA to reconstruct. The resulting increase in residuals (ranging from ≈ 100 μm to over 250 μm), combined with their low momentum, reflects the comparatively poorer track quality observed for hadrons.

4.1.3 Identification of the particle type

Unlike the event data from complex particle detectors such as CMS or ATLAS, the reconstructed tracks in ECC lack a lot of information, such as momentum or particle type.

The lack of a magnetic field in the emulsion detector at SND@LHC is responsible for the inability to bend the path of charged particles. Partially, this can be complemented with data from other SND@LHC detector systems; however, this would require matching events from the electronic detectors with accumulated events in the emulsion. In general, electronic detector systems are expected to point to the approximate location of the event of interest in the ECC unit rather than annotate each track in the emulsion with the relevant information.

Another solution to this problem lies in the utilisation of the high resolution of the emulsion to infer the particle properties from their track morphology. It is expected that the particles with high mass or high momentum will leave highly linear tracks, while the path of the low-momentum particle will exhibit more complex behaviour, as they will be more prone to Coulomb repulsion effects. Therefore, it is possible to obtain approximate values of the particle energy and its type, but currently, FEDRA does not offer any capabilities in this area.

4.1.4 Vertex reconstruction quality

The FEDRA vertexing capabilities have been examined using similar metrics for tracking quality with some slight modifications in the definitions, such as comparing tracks instead of hits between FEDRA and Monte Carlo-matched vertices:

1. **Vertex reconstruction efficiency:** This parameter quantifies the number of suitable simulated vertices that were at least partially reconstructed by the vertexing algorithm. Only vertices that contain two or more tracks, according to the Monte Carlo simulation, are considered in the calculation.
2. **Vertex track reconstruction efficiency:** This criterion quantifies the effectiveness of the FEDRA in reproducing the true vertex multiplicity of its MC simulated counterpart. For example, if according to the Monte Carlo simulation, a given vertex should have 6 tracks, but the FEDRA counterpart shares only 3 tracks, then the *vertex track reconstruction efficiency* or VTRE will be 50%.
3. **Vertex track reconstruction purity:** In the example mentioned above, the FEDRA counterpart had only three tracks that were linked to the Monte Carlo simulated track; however, the FEDRA reconstructed vertex can have a greater multiplicity, for example, nine. This is a random crossing effect that occurs when two or more unrelated tracks are merged as a vertex entity. The *vertex track reconstruction purity* or VTRP is calculated simply by dividing the number of matched tracks by the multiplicity of the reconstructed vertex. In this example, it is simply calculated as: $\text{VTRP} = \frac{3}{9} = 33.33\%$

The evaluation of the FEDRA vertexing performance on the Monte Carlo simulated data has yielded the following picture presented in Table 4.3 below:

Table 4.3: FEDRA vertex reconstruction performance split by the origin group taken from the control domain. Average values are calculated for vertices that have been at least partially reconstructed by FEDRA.

Particle source	**# MC simulated vertices**	**# Matched FEDRA vertices**	**Vertex reconstruction efficiency (%)**	**Vertex track reconstruction efficiency (%)**	**Vertex track reconstruction purity (%)**
EM showers	1072	61 ± 7.8	5.7 ± 0.7	93.9 ± 2.3	91.8 ± 2.2
Hadronic decays	146	82 ± 9.1	56.2 ± 6.2	77.8 ± 2.4	93.8 ± 0.5
ν_μ **interactions**	8	$7^{+1.0}_{-2.7}$	$87.5^{+12.5}_{-33.1}$	72.1 ± 8.0	81.6 ± 6.6

It follows a similar pattern to the tracking picture: the FEDRA vertexing algorithm struggles with vertices from electromagnetic showers and hadronic interactions. In the electromagnetic category, the vertex reconstruction efficiency (VRE) is very low ($\approx 6\%$), reflecting the difficulty of identifying true vertices made of low-momentum tracks. The FEDRA vertexing algorithm assumes that tracks are approximately straight when performing the vertex fit. Electrons and positrons from EM showers are typically low-momentum and therefore undergo significant multiple scattering in the emulsion, producing non-linear trajectories. This violates the straight-track assumption, causing the algorithm to struggle in performing vertex fit and resulting in a very low probability of successful vertex reconstruction.

For those rare matched vertices, the vertex track reconstruction efficiency (VTRE $\approx 94\%$) and purity (VTRP $\approx 92\%$), which quantify the quality of the matches, appear unusually high. This can be explained by a ceiling effect: the overwhelming majority of EM vertices correspond to photon conversions into an e^+e^- pair, producing two-track vertices. Since the minimum matching criterion requires two tracks, any correctly matched seed automatically yields a very high VTRE. The small deviation from 100% arises from a handful of EM vertices with simulated multiplicities greater than two, which are in fact artefacts of limitations in Geants interaction modelling: occasionally, photon conversions or secondary interactions are merged or misclassified, producing vertices with more than the expected two tracks.

By contrast, hadronic decays achieve a higher overall VRE ($\approx 56\%$) but lower VTRE and VTRP, reflecting the more complex vertex topology and higher chance of random track crossings. Finally, muon-neutrino interactions show the most balanced performance, with the highest VRE ($\approx 88\%$), albeit on a very small sample of vertices $\mathcal{O}(10)$, making the signal yield particularly sensitive to this metric.

It is important to note that only simulated vertices with tracks reconstructed by FEDRA are taken into account in this analysis: improved tracking could potentially lead to a greater number of reconstructed tracks, which could result in more vertices.

Previous attempts have been made to address the issues mentioned above, including the use of nonlinear Kalman filters[147], adapting conventional vertexing algorithms to combine fragmented tracks, and techniques to estimate track momentum by examining its deviation caused by Coulomb scattering[148]. In line with the subject of this thesis, a different approach will be taken here, which involves the use of perhaps one of the most well-known machine learning applications: deep learning with artificial neural networks.

4.2 Literature review: ANN in particle physics

There is a relatively large amount of literature on the use of ANNs in physics experiments. However, their applicability to the research topic of this thesis is varied and can be split into four categories.

1. Use of ANNs for event reconstruction in emulsion detectors: these would have the biggest overlap with the work being carried out here, and hence they have the highest value from the point of applicability.
2. Use of ANNs for event reconstruction in electronic detectors: they still have

value in terms of applicability, but their methodologies need to be adapted to the emulsion data environment, accounting for any specifics associated with emulsion data analysis.

3. Use of ANNs for event analysis in emulsion detectors: this involves cases where the data from the standard event reconstruction is analysed with the use of artificial neural networks. They do not have direct applicability to the research carried out here, but they help direct it since they provide the requirements for the quality of the reconstructed data.

4. Use of ANNs for event analysis in electronic detectors: this is the least interesting domain in the literature, since it describes methods that solve different classes of problems in different experimental environments. The implementation of deep learning methods is very sensitive to the data being used, and changing them can render the methods and associated ANN models useless in a different experimental setting. Understanding the general challenges of using artificial neural networks and how they are solved is the only practical benefit that can be gained by analysing this type of literature, and therefore it will not be considered here.

The analysis of the available materials on the topic has been categorised according to the classification above and is presented below.

4.2.1 Use of ANNs for event reconstruction in emulsion detectors

The use of ECC technology is a niche application in the field of experimental high-energy physics, and therefore, the number of experiments using emulsion detectors is relatively small. Additionally, the time constraint is a significant factor, since the application of ANNs before the 21st century was limited by the relatively small computing power available at that time.

The following relatively modern experiments with emulsion detectors can be compared to SND@LHC: CHORUS[149], DONuT[150] and OPERA[74]. Perhaps unsurprisingly, the literature on the use of artificial neural networks for these experiments is practically non-existent. The closest attempts to solve at least some of the problems mentioned in the motivation Section 4.1 have been reported for the DONuT experiment, where the track momentum in the emulsion was deduced by measuring the multiple Coulomb scattering[151]. This was an analytical approach without the use of machine learning and was therefore only mentioned here for completeness.

4.2.2 Use of ANNs for event reconstruction in electronic detectors

The proposals to use neural networks for tracking can be traced back to the 1990s when A. Eide et al.[152] explored the idea of using simple feedforward neural networks for general electronic detectors at the time. Since then, there has been no noteworthy progress in the use of neural networks for reconstruction, as standard algorithms were able to adequately handle the event rates available back then.

The LHC was launched in 2008, offering the opportunity to search for new physics. However, the high luminosity of the LHC has pushed the traditional Kalman filter-based tracking algorithms to the limit, due to the increasing pile-up.

To address current and anticipated challenges in future high-luminosity experiments[19] a pilot project called HEP.TrkX[153] was established in 2016. The first outcome of the project was a preliminary report[154] investigating the possibility of using CNN's and RNN's for tracking charged particles in LHC detectors such as ATLAS and CMS. A follow-up report[155] was published next year with the following conclusions:

- While demonstrating promising results, CNNs and any image-based methods will struggle in scaling up to realistic high-luminosity data because of their high dimensionality and sparsity.
- RNNs still provide an adequate alternative to the existing KF-based tracking algorithms that show similar performance.
- The hit data in the particle detector can be modelled as a graph, and therefore GNNs represents another strong alternative to the sequential tracking used in the standard tracking algorithm and the proposed RNN's alternative.

Based on these findings, two ways to use ANNs for tracking purposes have been identified: RNNs and GNNs.

The main advantage of the RNN route is that it can be easily incorporated into existing standard tracking algorithms simply by replacing the Kalman filter as a hit predictor. The subject of RNNs is a relatively mature and understood field, with the first implementation in 1986[156]. Unfortunately, they suffer from some serious issues, such as the *vanishing gradient problem*[157], which makes them unable to effectively learn long-sequence data. Although some attempts have been made to rectify it, such as the modifications of *long short term memory* (LSTM)[158], there is barely any hard evidence in the literature that RNNs work well on large tracking data. Subsequent studies have been conducted on this subject[159], but there is no noticeable progress in the development of RNN trackers.

The GNN approach is conceptually different from the proposed RNN-assisted tracking. Instead of performing sequential tracking, the detector hit data are treated as a graph of nodes where each node is assigned hit attributes, such as spatial coordinates. The GNN tracking mechanism is based on predicting weights for all physically possible edges and leveraging this information to build track representations. The compactness and versatility of the graph data representation enables it to model any kind of detectors, including complex multi-layered systems such as ATLAS. Therefore, it is not surprising that the subject of graph-based tracking is an active area of research that regularly produces reports such as [160][161][162][163] praising the performance of GNNs.

Certainly, there have been issues identified[164] which include standard ANN problems such as the requirement of generous computer resources and the concealed nature of deep learning. An additional set of challenges specific to GNNs has also been identified, such as converting physics problems into graphs, immaturity of the available software libraries, and lack of integration with the experimental pipeline.

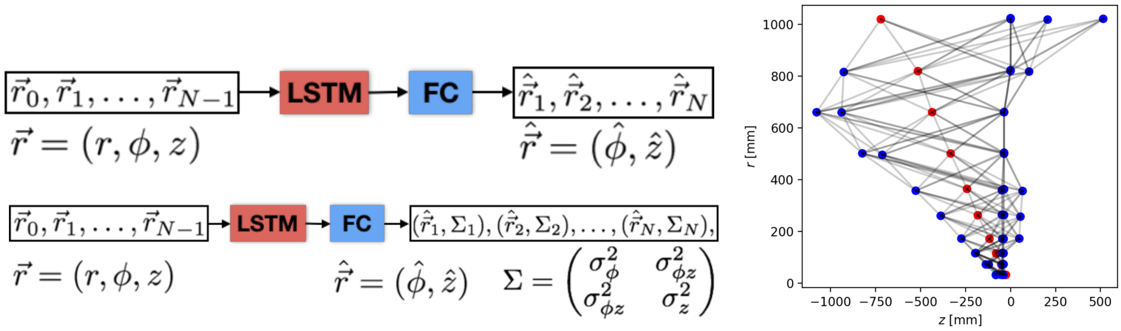


Figure 4.4: The schematic representation of RNN-based tracking (left) and the GNN alternative (right). Taken from [154].

Whether those logistical problems are the only blockers or there are more fundamental issues is unclear: so far, no solid evidence of practical use of the GNN-tracking algorithm for the actual non-simulated data analysis has been found in the literature.

There are reports such as [165] and [166] that describe the potential use of GNNs for physics problems, but the content of the reports presents GNN as a suggestion rather than an actual working tool that solves the real problem. This can partially be explained by the time lag between data collection and publication, which for LHC experiments can take many years. For example, the following ATLAS paper[167] was published only recently in 2023, but the data was collected between 2015 and 2018, when the GNN tracking algorithms were only at the conception level.

Thus far, this section has been largely devoted to the utilisation of ANN for track reconstruction, rather than other issues that are expected to arise from the SND@LHC emulsion data, such as track identification and vertexing. There are two main reasons for this bias:

- Particle tracking is a critical component of event reconstruction, yet it is the most difficult part and thus serves as a hindrance to subsequent processes. Therefore, it is unsurprising that a great deal of effort is dedicated to improving the tracking capabilities, as these alone are likely to benefit the consecutive vertexing step.
- Some of the problems mentioned in the motivation Section 4.1 are more relevant for ECC detectors than their electronic counterparts. For example, the segmentation of tracks correlates with the maximum reconstructable number of hits: the more hits that make up the track, the more likely the fragmentation. The ATLAS or CMS pixel detectors consist of four layers, making track segmentation practically impossible. Neither particle identification is a pressing issue: many electronic devices use magnetic fields and the composite detector layer architecture to deduce the momentum of the particle and its type.

There is evidence of some activity in the literature on the use of ANNs in vertex reconstruction, but it lacks large pilot research projects such as HEP.TrkX or Exa.TrkX[168] and is mainly driven by small collaboration groups in an attempt to solve vertexing problems for their specific experiment. It should be noted that, despite the differences between the objective, the experimental design and the data used there and the present study, there are still some intriguing and beneficial concepts that can be applied to the research carried out here.

The successful use of neural networks was reported for the PandaX-III experiment designed to find the neutrino-less double decay of Xenon-136. The authors have used an adaptation of the VGG16 classification model[169], which itself is a derivative of the AlexNet architecture, to reconstruct the event vertex and its position. CNN reconstruction was claimed to have improved the energy resolution of the experiment from 10.1% to 4.0% FWHM, and that the method had been verified on the experimental data from the PandaX-III prototype[170].

Another $0\nu\beta\beta$ experiment called Next has claimed to use convolutional neural networks for event classification and vertex rejection. The adopted CNN architecture was conceptually different: it used a specific implementation of CNN with only two convolutional layers, each employing tensor kernels and pooling operators, enabling it to work on the 3D data. The use of CNNs has been claimed to reduce the background to $\approx 10\%$ while retaining the signal efficiency at about 65%[171].

CNNs are not the only type of neural network that was considered for vertex reconstruction tasks: other types were used to develop the vertexing algorithm[172] for the proposed International Linear Collider (ILC). In terms of complexity, the proposed ILC detectors are comparable to those of the LHC and the physics events are expected to produce a wide range of signatures that include primary and secondary vertices. To cope with the challenging task of reconstructing and identifying them, a complex track data pipeline was created that consists of two parts: seed finding and vertex building. The seed finding process is performed by a simple forward neural network, which, in addition to the track seed classification, calculates the location of the vertex origin. The vertex-building part of the algorithm takes the seeds from the previous step and increases their multiplicity by adding the tracks one by one. Conceptually, this is a similar strategy used in the tracking proposals mentioned above: replicate the Kalman filter fitting used in standard vertexing al-

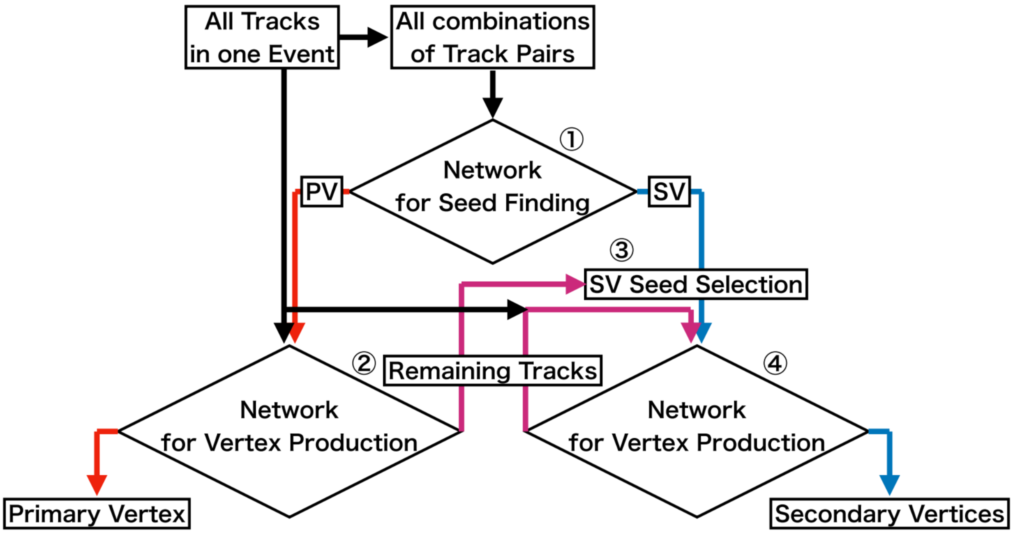


Figure 4.5: Schematic diagram of the ICL proposed vertex finder. Taken from [172].

gorithms with the RNN analogue. The authors claim that their vertex algorithm shows better reconstruction efficiency than its non-ML counterpart LCFIPlus[173], albeit with worse purity. Since the ICL at the time of writing was still in conception, the performance studies were done on the Monte Carlo simulation and were never tested on the actual data and should be treated as preliminary.

A brief but intriguing paper was published from the *Connecting the Dots and Workshop on Intelligent Trackers* event that discussed the use of feedforward neural networks for vertex reconstruction in the general cylindrical ATLAS-like detector[174]. Although the reported neural network architecture was found to have a limited modelling capacity at high-track multiplicity, the approach itself is interesting: it skips the track reconstruction part and builds the vertices from the raw hits. In principle, assuming sufficient implementation, this approach could work for the SND@LHC detector, where the tracking takes most of the offline reconstruction time but is mainly concerned with the background processes, while the signal signatures are represented by the vertex.

In conclusion, neural networks have the potential to handle certain aspects of the physics event reconstruction for the SND@LHC experiment. However, three key aspects make the ECC detectors different from their electronic counterparts:

1. **Data quality:** the DAQ processing of emulsion films is very manual compared to the electronic detectors that collect and store data in an automatic regime. The development of the emulsion films is fully manual, and the scanning is semi-automatic at best, making the collected event information vulnerable to data quality issues. The use of different microscopes can also contribute to the problem if the scanning settings or methodology differ drastically. Different plates from the same ECC unit can be scanned by various laboratories, resulting in the detector having a non-homogeneous quality pattern throughout its length. In the DONuT experiment, it was claimed[67] that up to 50% of the predicted vertices have not been reconstructed in the emulsion due to poor data quality alone.

2. **Alignment:** the deformation of the emulsion films during the data collection and scanning procedure distorts the reconstructed data. The exact pattern and degree of distortion of each ECC unit are individual, making it difficult to replicate in a Monte Carlo simulation. Electronic detectors suffer less from this problem because their constituent parts are more rigid and remain stable over time.

3. **Hit density:** the typical electronic detectors, such as ATLAS, while exposed to high data rates, can separate them into events using the time stamp. The process is not 100% efficient, and pile-up phenomena occur when more than one physical interaction is present in the same event. After the expected increase in the luminosity of the LHC, the average event with 200 interactions is projected to have approximately $\mathcal{O}(10^5)$ hits. Although it might seem large, this is a meagre amount compared to the typical number of hits in the ECC data, where the typical density is reaching 2.5×10^6 cm^{-3} in the Run 1 data only. Due to the data-accumulating nature of the emulsion detector, all hits are treated as one large event, which is a significant computational challenge, especially for tracking tasks.

4.2.3 Use of ANNs for event analysis in emulsion detectors

As with event reconstruction, there is not much literature on the use of ANNs for data analysis in emulsion detectors. A paper[175] describes a successful application of ANN in the OPERA experiment, where the first observation of a ν_τ CC interaction with charm production has been observed with a significance of 3.5σ. It was a rather simple feedforward neural network that would analyse the kinematic variables of the event and produce the classification.

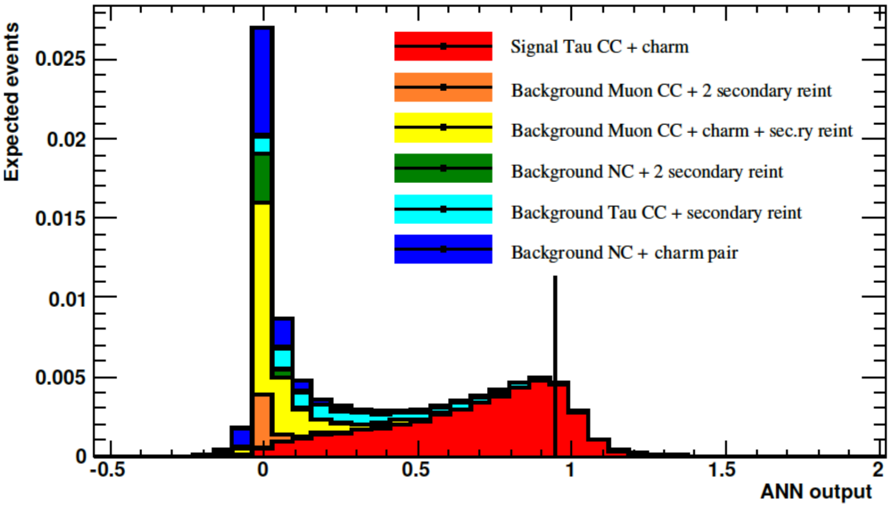


Figure 4.6: The output of the artificial neural network is displayed in the graph, with each process contributing in a different hue. The vertical black line indicates the ANN output for the event being studied in this analysis. Taken from [175].

If the assumption of using deep learning is relaxed, then there is slightly higher activity in terms of the application of simpler ML tools. *Boosted decision trees* are used in SND@LHC for background vertex rejection analysis. A more complex task of reconstructing electromagnetic showers has been attempted for the SHIP experiment[176] with the use of various ML tools such as *support vector machines* (SVM), *XGBoost classifiers* (XGB), and *conditionally random fields* (CRF). Performance was found to be on par with that of the standard algorithm, but the comparison was not adequate, as some of the information used by the baseline algorithm was not utilised.

4.3 General methodology for SND track and vertex reconstruction

In this section, a comprehensive methodology will be presented that applies to all research lines performed in this chapter. This part will cover important topics such as the data and tools utilised and the evaluation criteria that apply to all aspects of emulsion reconstruction and analysis. However, more specific information on a particular reconstruction study will be outlined in the respective sections.

4.3.1 Data

Two key data sources have been used for this study: Monte Carlo simulation and a sample of the emulsion data collected during Run 1. Data sets have been prepared for studies in tabular format as a comma-separated text file of values or CSV. The main advantages of this file format are its accessibility to most third-party analysis tools, including Excel, and its simplicity, which enables one to inspect them easily without using any advanced technical skills. However, there are certain disadvantages associated with them, including their limited capacity to model complex data and the absence of metadata. Consequently, they are appropriate only for the initial demonstration phase and should be substituted in the future with more appropriate formats like ROOT or JSON. In this study, CSVs were predominantly used as input, while a Python object representation was utilised for high-level output.

The simulated and actual data sets are divided into domains called *bricks*, which correspond to the ECC cells in the emulsion detector. Each brick is identified by a two-digit number, where the first digit represents the wall sequence, while the second digit indicates the particular unit out of four from which the data have been collected. For example, the "Brick 23" domain contains data that have been simulated or collected in the second wall of the emulsion detector in ECC Unit 3.

Within each brick, the smallest unit of information is a base track or hit that is parameterised by a minimum of five variables: x, y, z, tx, and ty. The first three variables describe the spatial location of the hit in the local Cartesian coordinate system of units of ECC within each wall, with the z-axis aligned along the direction of the IP1 beam collision axis and the y-axis pointing upward from the detector floor, as illustrated in Figure 2.10.

Typically, for each brick, the position of the plate farthest along the z-axis serves as the point of origin, with all preceding plates having negative values of z. In the frontal view projection, the x-axis and y-axis are positioned at the farthest bottom-right corner of the brick wall. From a purely emulsion analysis perspective, the alignment of the coordinate system can be arbitrary for each ECC unit, as long as it remains consistent within that cell. The definition of reference points may vary slightly depending on whether a simulation or actual data is being considered.

The tx and ty parameters quantify the slope of the hit along the x and y-axis, respectively, and are expressed as the dimensionless ratio.

In addition to the five variables described above, hits can carry additional information content that depends on the type of data, which will be described in more detail in the following:

Monte Carlo simulated data

For this data set, a simulation of two types of events has been performed throughout the entire SND@LHC detector.

- 100 signals muon-neutrino interaction events, including secondary decays.
- 83,984 incoming muon background events, including muon-induced processes such as delta rays and bremsstrahlung.

It is important to emphasise that the number of simulated signal events does not necessarily equate to the number of detectable signal events: some can leave a barely noticeable footprint in the reconstructed data, such as a few hits that cannot be reliably identified. Hence, even with a perfect reconstruction of these events, the expected signal yield is anticipated to be lower than the initial simulated value.

The Monte Carlo sets play a crucial role in this research and has three objectives: provide training data for neural network models, allow the determination and validation of the reconstruction parameters, and the final independent test of the quality of the reconstruction. In line with these objectives, the simulated data were split into three domains, as shown in the figure below.

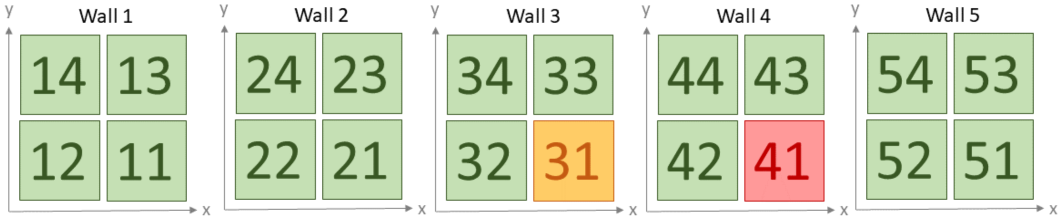


Figure 4.7: Schematic representation of the Monte-Carlo generated data split into three sets of ECC units. **Green:** Bricks used for the training of neural networks (*ML domain*). **Orange:** ECC cells used for the determination and tuning of the global reconstruction parameters (*Calibration domain*). **Red:** The ECC units used for the final testing (*Control domain*).

The choice of bricks has been mainly dictated by the following considerations:

- The bricks that have identifiers ending with ’1’ are situated closer to the collision axis. As a result, they are anticipated to have the highest number of signal and background events. To replicate real-life conditions as closely as possible, the final testing should be conducted on the brick located in the lower-right quadrant.
- The number and density of background events increase with the wall number as a result of the additional products of interactions in the preceding ECC units. Therefore, it is desirable to choose walls toward the end of the emulsion detector to obtain a more challenging testing environment.
- The ML domain also requires some bricks with a high density of hits; otherwise, the neural network will perform poorly in the calibration and control bricks.

In addition to spatial information, the data simulated using the Monte Carlo method also include truth information, allowing the determination of the type of original event and its topology. It provides labels for machine learning data points and plays an essential role in a wide range of reconstruction assessments. The simulation data model, shown schematically in Figure 4.8, groups each hit into tracks, tracks into vertices, and vertices into events.

It is important to note that not all events contain vertices. For example, a muon originating from IP1 is modelled as an "external" particle, although it should theoretically originate from some decay outside of the SND@LHC detector.

Some caveats have to be taken into account when dealing with Monte Carlo metadata, such as:

- The event identifiers have been generated globally for the whole emulsion detector, while the typical emulsion analysis is done at the brick level. For example, the vertex originating at the first wall will produce the daughter particles in the proceeding wall with the same Mother ID, which can lead to the vertex overcounting. To mitigate this problem, an additional data manipulation procedure has been performed, which involved reassigning a special value of Mother ID to all particles in a given brick that originate in the preceding ECC unit. This procedure prevents an underestimation of the signal efficiency of a given brick.
- The Mother and Track IDs restart for every single event, and therefore, are not unique when considered in the context of a single simulation or even a single brick. Therefore, some trivial data reprocessing has been performed, which involved concatenating Brick and Event IDs to each track and vertex entity to make it unique.

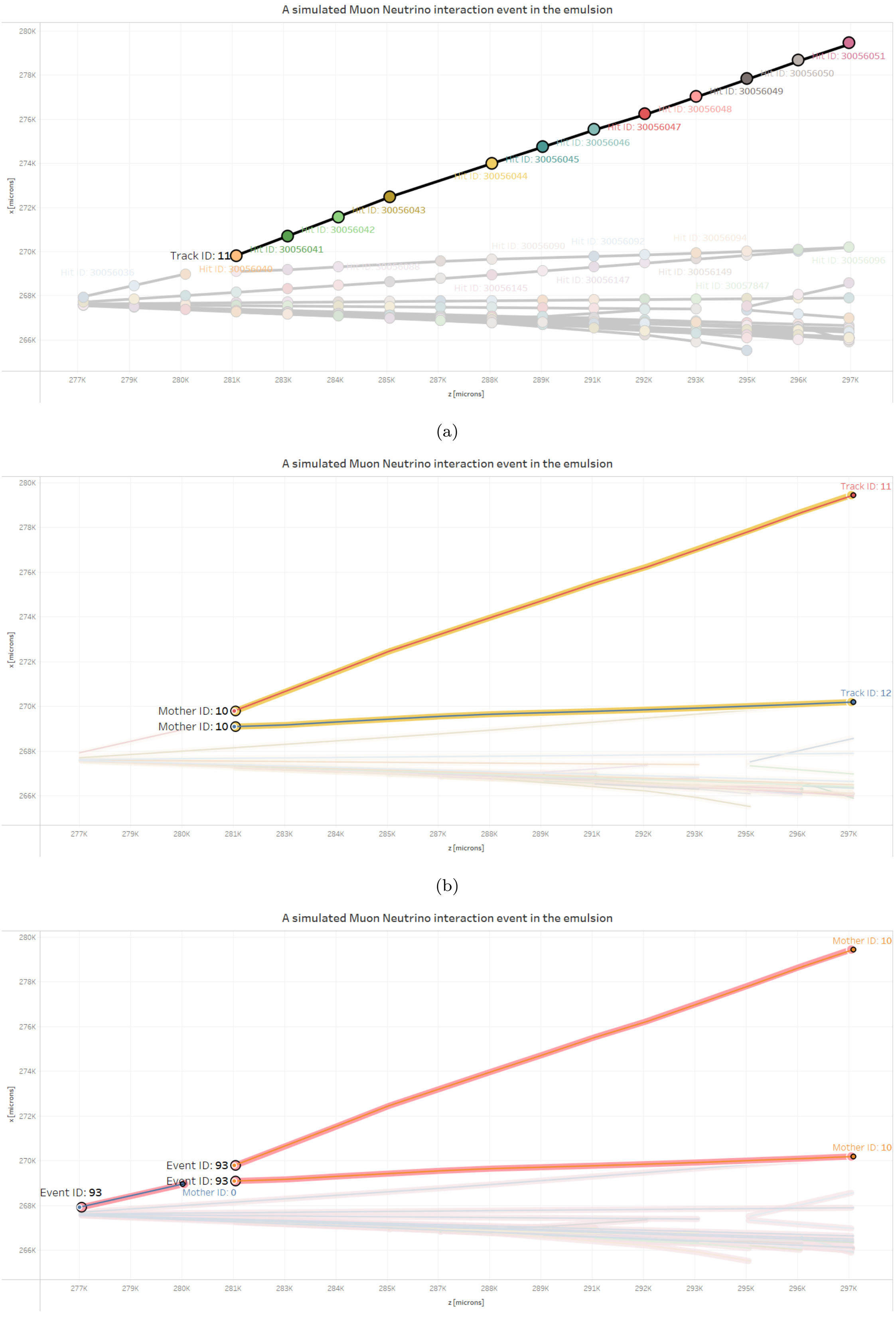
Apart from the topological information, Monte Carlo simulation contains other useful metadata that enables identification of the physical process in the event and its kinematical properties, such as:

- **PDG code:** it is a number assigned on a track level to identify the particle that created the track. The numbering logic follows the Particle Data Group convention for Monte Carlo particle simulations[177].
- **Mother PDG code:** it is almost the same as above, but instead of identifying the flavour of the particle responsible for the hit, it identifies the type of particle from which it came as a result of decay. All particles originating outside of the detector, such as IP1 muons, have this identifier set to zero.
- **P:** it is the magnitude of the three-momentum vector in units of gigaelectron volts (GeV) of the truth particle that leaves the track in the detector. Since the particle is expected to lose energy as it traverses the film plates, the momentum is defined at the hit level.

Other information, such as the production time and the type of interaction process, is available but is barely used, and therefore is not mentioned here.

The simulated data considered here have been processed using the FEDRA tracking and vertexing procedure. In addition to the Monte Carlo-generated truth metadata, it also adds standard tracking and vertexing information that serves both as a benchmark and as a material for the reconstruction enhancement procedures used here. The data structure is very similar to that used for the simulation. Hit IDs form FEDRA-reconstructed tracks that, in turn, can serve as building blocks for higher-order structures such as FEDRA-reconstructed vertices. The concept of an event is poorly defined within the FEDRA framework and is limited to identifying a given vertex as primary or secondary.

Although the Monte Carlo data used are an important tool without which this research would be impossible, there are some serious limitations of the simulation that potentially can diminish the effectiveness of the studies performed here:



(a)

(b)

(c)

Figure 4.8: Data structure of a simulated signal event: (a) Each hit has its unique identifier and can act as a building block for a track with the same *Track ID* (in this example, it is 11). (b) Every track, in turn, can form higher-level structures, such as vertices, which are denoted by *Mother ID*. (c) Tracks and vertices form the highest order structure in the simulation called *event* and denoted by *Event ID*.

- **Low density:** even the simulated Brick 51, which has the highest average density of hits (approximately 1.21×10^3 cm^{-3}), does not come close to matching the conditions observed in real data. For example, just a 1 cm × 1 cm cross-sectional area of Run 1 Brick 42 data has an average density of around 1.6×10^6 hits per cubic centimetre, which is more than three orders of magnitude higher than in the simulation. A much higher density of the actual data implies a higher volume of random crossing effects than that observed during neural network training and the reconstruction parameter calibration.
- **Low event diversity:** only muon neutrino and muon background events have been simulated, leaving out other important background contributions. The current simulation also ignores tau and electron neutrino interactions, which considerably limit the scope of this research and should be revisited for future studies to address all requirements of the SND@LHC programme.
- **Simplistic modelling:** the degradation of data quality due to plate misalignment effects is not taken into account in the Monte Carlo simulation. The emulsion detector is modelled as a rigid structure that preserves the unrealistic perfect alignment configuration, which is unattainable in real-life conditions. This presents another obstacle for the neural network models that have been trained on the "perfect" data only.

Run 1 data sample

To assess the reconstruction techniques developed in this study in real-life conditions, a small sample of the Run 1 data has been selected as illustrated in the diagram below:

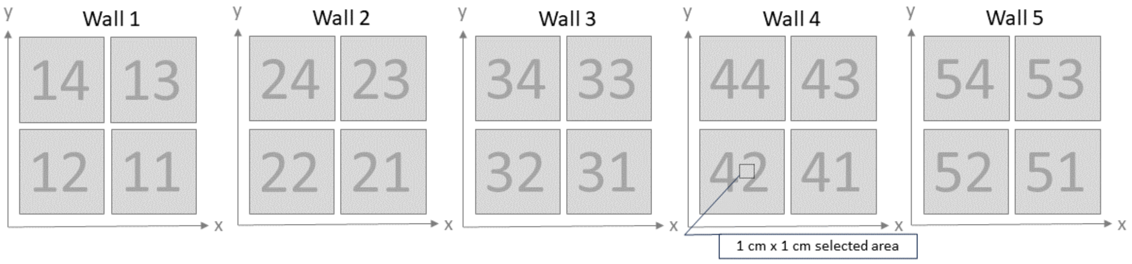


Figure 4.9: Schematic representation of the location of the Run 1 data sample.

On the granular level, the Run 1 data have a similar structure to that used in the simulation; however, there are some important differences:

- **Lack of truth information:** real data lack information present in the Monte-Carlo simulation, such as the event topology and the particle type.
- **Scan metadata:** it contains microtrack metadata collected during the emulsion scanning procedure, such as micro-track fit parameters and the number of silver clusters that are not produced in the Monte Carlo simulation.
- **High event diversity:** the data provided in this section contains all the events that can be recorded in the emulsion while being exposed to the Run 1 cycle. This includes various sources of background that were observed in

the T118 tunnel, such as thermal neutrons generated by the LHC beam, and hadrons produced by the interaction of IP1 muons with the rock upstream of the SND@LHC detector.

- **Instrumental imperfections:** the data contain two artefacts of the emulsion handling procedure: distortion of the detector geometry compared to the "perfect" MC simulated layout and fog, which results in additional thermally induced noisy hits in the data.

To demonstrate the difference in event densities between simulated and recorded data, a summary table has been compiled and presented below.

Table 4.4: High-level statistical description of the data used for this research, split by domains and brick number.

Data domain	**Brick ID**	**Total number of hits**	**Average hit density [cm^{-3}]**	**Maximum hit density [cm^{-3}]**	**Total number of FEDRA tracks**	**Average FEDRA track density [cm^{-2}]**	**Maximum FEDRA track density [cm^{-2}]**
MC ML	11	2.95×10^6	$(1.02 \pm 2.51) \times 10^3$	1.38×10^4	6.77×10^4	$(2.73 \pm 2.63) \times 10^2$	1.25×10^3
	12	8.59×10^5	$(2.98 \pm 21.8) \times 10^2$	1.07×10^4	1.90×10^4	$(2.28 \pm 2.44) \times 10^2$	9.05×10^2
	13	7.71×10^5	$(2.68 \pm 21.7) \times 10^2$	1.34×10^4	1.83×10^4	$(2.28 \pm 2.55) \times 10^2$	1.06×10^3
	14	1.16×10^6	$(4.05 \pm 23.3) \times 10^2$	1.25×10^4	2.73×10^4	$(2.43 \pm 2.58) \times 10^2$	9.75×10^2
	21	3.36×10^6	$(1.17 \pm 2.31) \times 10^3$	1.46×10^4	7.76×10^4	$(2.18 \pm 2.60) \times 10^2$	1.21×10^3
	22	9.77×10^5	$(3.40 \pm 19.2) \times 10^2$	1.06×10^4	2.14×10^4	$(1.62 \pm 2.31) \times 10^2$	9.21×10^2
	23	8.84×10^5	$(3.07 \pm 18.2) \times 10^2$	1.55×10^4	2.07×10^4	$(1.41 \pm 2.34) \times 10^2$	1.06×10^3
	24	1.32×10^6	$(4.60 \pm 19.7) \times 10^2$	1.16×10^4	3.12×10^4	$(1.51 \pm 2.36) \times 10^2$	9.39×10^2
	32	9.97×10^5	$(3.47 \pm 16.4) \times 10^2$	1.18×10^4	2.20×10^4	$(1.21 \pm 2.05) \times 10^2$	9.40×10^2
	33	8.71×10^5	$(3.03 \pm 14.7) \times 10^2$	1.21×10^4	2.08×10^4	$(9.60 \pm 19.6) \times 10$	1.06×10^3
	34	1.34×10^6	$(4.65 \pm 17.3) \times 10^2$	1.08×10^4	3.17×10^4	$(1.19 \pm 2.11) \times 10^2$	8.59×10^2
	42	1.07×10^6	$(3.71 \pm 14.7) \times 10^2$	1.32×10^4	2.41×10^4	$(8.70 \pm 18.6) \times 10$	9.15×10^2
	43	9.08×10^5	$(3.16 \pm 13.2) \times 10^2$	9.43×10^3	2.19×10^4	$(7.20 \pm 16.8) \times 10$	9.59×10^2
	44	1.36×10^6	$(4.73 \pm 16.0) \times 10^2$	1.50×10^4	3.22×10^4	$(9.20 \pm 18.6) \times 10$	1.01×10^3
	51	3.47×10^6	$(1.21 \pm 1.93) \times 10^3$	1.28×10^4	8.10×10^4	$(1.26 \pm 2.08) \times 10^2$	1.12×10^3
	52	1.05×10^6	$(3.67 \pm 13.2) \times 10^2$	1.06×10^4	2.32×10^4	$(6.60 \pm 15.1) \times 10$	7.54×10^2
	53	9.12×10^5	$(3.17 \pm 12.3) \times 10^2$	1.05×10^4	2.20×10^4	$(5.80 \pm 14.9) \times 10$	9.48×10^2
	54	1.42×10^6	$(4.94 \pm 16.8) \times 10^2$	2.00×10^4	3.41×10^4	$(8.50 \pm 17.5) \times 10$	1.13×10^3
MC calibration	31	3.36×10^6	$(1.17 \pm 2.10) \times 10^3$	1.52×10^4	7.76×10^4	$(1.67 \pm 2.42) \times 10^2$	1.28×10^3
MC control	41	3.34×10^6	$(1.16 \pm 1.98) \times 10^3$	1.72×10^4	7.72×10^4	$(1.43 \pm 2.25) \times 10^2$	1.30×10^3
Run 1 data sample	42	2.87×10^7	$(1.63 \pm 0.67) \times 10^6$	2.52×10^6	5.48×10^5	$(1.82 \pm 2.67) \times 10^4$	1.50×10^5

The following quantities are presented in the table:

- **Toal number of hits**: this is the total number of base tracks present in a data sample, regardless of whether they have been used for reconstruction or not. For the non-simulated data, this number also includes hits that would be rejected by the tracking algorithm due to the cuts on the microtrack parameters.

- **Average hit density**: This is derived simply by dividing *total number of hits* by the fiducial volume in cm^3 that contains the base tracks. For Monte Carlo-generated sets, this volume is calculated using the nominal dimensions of the ECC unit of 19.2 cm ×19.2 cm × 7.8 cm. The Run 1 data sample originally had been defined as having dimensions of 1 cm ×1 cm × 7.8 cm; however, the effective volume has been changed to 1.51 cm ×1.55 cm × 7.5 cm. This is due to the alignment procedure, which involves shifting each plate with respect to the other in an attempt to reverse the ECC deformation effects. The error was calculated by splitting the data sample into cm^3 volumes, measuring the hit density in each and calculating the variance. It is important to note that the density pattern of the data used does not follow the normal Gaussian distribution, especially for the simulated data. This is due to the specific ways the Monte Carlo data were produced: instead of uniformly populating the bricks with muon background particles, only the specific areas around the neutrino interaction have been saturated with the passing muon tracks, leaving other areas of the detector sparsely populated.
- **Maximum hit density**: this is calculated by splitting the data sample into one cm^3 volume, measuring the hit density in each and taking the highest value.
- **Total number of FEDRA tracks**: this is the total number of tracks that have been identified by the FEDRA standard reconstruction procedure. Only tracks made up of a minimum of four hits are considered.
- **Average FEDRA track density**: this can be interpreted as the average number of tracks incident on an ECC plate per unit area. It is derived by splitting the area of each plate into 1 cm × 1 cm fragments, measuring the number of hits that were used in the tracking procedure and taking the mean and variance.
- **Maximum FEDRA track density**: this can be interpreted as the maximum number of tracks incident on an ECC plate per unit area. It is derived by splitting the area of each plate into 1 cm × 1 cm fragments, measuring the number of hits that were used in the tracking procedure and taking the highest value.

The list of data parameters above has been shortened to give a general idea of the data volumes involved in training, calibration, and final testing. There are other important quantities, such as the number of vertices and signal events, which were omitted here for conciseness but will be covered in more detail in the relevant sections in due course.

4.3.2 Reconstruction strategy

In the majority of the reconstruction workflows, whether the standard reconstruction or the neural network-driven process is considered, the basic unit of the data flow is called the *seed*. Depending on the specific task, the seed is simply the combination of two objects in the emulsion data that make up more complex higher-order structures. For example, in the case of tracking, that would be a pair of constituent hits that make up a volume track or at least one of its segments. For the vertexing operation, the seed would consist of a pair of tracks that can potentially become part of the vertex.

A typical reconstruction that will be performed within this report involves the following steps: seed creation and refinement, neural network analysis, and final output consolidation, which will be described in more detail below.

Seed creation and refining

The data preparation serves the following purposes:

1. Clean and convert the input data to a suitable format.
2. Restructure the data into possible seeds.
3. Reduce data flow by eliminating background seeds while preserving the relevant signal combinations by using the standard analytical techniques as much as possible, so that the neural network is not overloaded with high rates of data flow. Typically, this is achieved by numerous filtering steps, the order of which is dictated by computational efficiency considerations: the simplest filters are applied first to reduce data flow as much as possible for complex steps that require more memory and CPU resources.
4. Develop a suitable data point representation for neural network analysis, which varies depending on the type of neural network being used. For CNNs, it is necessary to use three-dimensional numpy arrays, while GNNs require a graph object of the seed.
5. If the data are used for neural network training, the additional two steps are used: assigning labels to seeds using simulated truth information and seed resampling. The latter is needed to ensure that each batch used in the ANN training has an even distribution of seeds with different labels, resulting in a smooth learning process.

Neural network analysis

Four important aspects of neural network handling need to be discussed in more detail: data preparation, choice of architecture, training and use during reconstruction.

The training data set is generated by merging all data within the ML domain, denoted by green colour in Figure 4.7 and then generally split into the *training* and *validation* samples with a split of 90:10%. In some complicated cases where the acceptance of neural networks is optimised, a more rigorous schema is followed, which involves splitting training data into three sets: training, validation, and *test* samples with a ratio of 85:10:5%.

One of the most difficult parts of dealing with neural networks is choosing their architecture. Ideally, the complexity of the neural network should be just sufficient to effectively model the subject data, but not more than that. Otherwise, an overcomplicated NN model will consume more computational resources than necessary and will be prone to overfitting. Striking the right balance is, in essence, a very similar type of problem that was solved in Chapter 3 and requires optimisation of many NN parameters such as the number of layers, size of kernels, the value of the dropout and many more, depending on the type of the neural network. This is particularly true for CNNs, which have a large, high-dimensional parameter space and for which grid search and other optimisation strategies have been employed. However, during the late phases of this research, CNNs were mostly abandoned in favour of GNNs, which are easier to optimise and for which the principle of parsimony was applied as follows:

1. Pick a decently sized training sample.
2. Start with the simplest NN configuration.
3. Train the model on the training sample until the performance on the validation set begins to plateau.
4. Record the loss and accuracy achieved.
5. Slightly increase the model complexity and use it for the next iteration.
6. Continue repeating steps 3-5 until no further improvement is observed.
7. Choose the model that achieved the overall performance plateau.

Once the optimal configuration of the neural network is chosen and the training data are ready, neural network learning can begin using the backpropagation process described in Section 2.6.5. It can be described briefly as an iteration of the following steps:

1. Split the training data into smaller batches of equal size. Depending on the type of neural network, batch sizes of 4 (for CNNs) to 16 (for GNNs) were used.
2. Perform a backpropagation on each batch until all of them have been processed in the given training sample. This denotes an epoch.
3. Run both validation and training samples through the neural network and record loss and accuracy.
4. Decrease the learning rate slightly according to the predetermined schedule.
5. Continue repeating steps 2-4 until no further improvement is observed.
6. Stop training once the plateau for both loss and accuracy has been reached.

After the training process is completed, the neural network can be applied to assist in reconstructing the Monte Carlo data in both the calibration and control domains. Additionally, the network can also be used for Run 1 data. Compared to the computationally intensive backpropagation process used in training, the actual task of reconstruction is much simpler. For each input data point, only a single feedforward pass is performed, which yields a prediction that can be used for classification or regression tasks.

It is important to ensure that the pre-selection steps for the reconstruction input data match or exceed those used for neural network training. In this way, the gap in the neural network performance between running on the ML domain and "unseen" data is minimised as much as possible.

Final output consolidation

The assimilation process of the neural network predictions varies depending on the type of reconstruction used. In general, they follow some adjustable algorithm that remaps or groups the original data into a more complex representation, which can be used for analysis or further reconstruction steps. The initial pre-reconstruction state of the input data is always preserved for later optimisation and testing.

4.3.3 Tools

The tools used for this research are very similar to those used in Chapter 2 and can be divided into three main categories: storage and computing infrastructure, a reconstruction software framework, and third-party tools that are used for the analysis and visualisation of input and output data.

Storage and computing infrastructure

The primary environment within which reconstruction processes are carried out is called LXPLUS[178], which stands for **L**inu**X** **P**ublic **L**ogin **U**ser **S**ervice. It consists of a cluster of public machines provided by CERN, where each machine runs either CERN CentOS 7 Linux (CC7) or Enterprise Linux 9 (EL9) operating system and has access to numerous compilers, such as C++ and Python and their respective library sets. All CERN users can interact with each machine via the Secure Shell Protocol (SSH), which removes the need to run programs on a local computer.

However, the LXPLUS itself is not designed to run jobs with high CPU consumption: instead, it offers an HTCondor batch submission system[179]. The purpose of this system is to ingest and distribute user-submitted jobs across the large batch farm consisting of around 100k cores. Its queueing system introduces a degree of latency when it comes to job execution, but it allows for the submission of a theoretically unlimited[5] number of tasks. This is ideal for reconstruction jobs that can be parallelised and split into smaller chunks, and is highly applicable to the problems attempted in this thesis.

In addition to computing services, LXPLUS provides access to storage facilities such as Andrew File System[180] (AFS) and EOS[181][6]. From the read-write speed and storage perspective, EOS is superior to AFS as it has a considerably higher storage quota of 1 terabyte per user versus the modest 100 GB offered by the AFS service. Therefore, EOS is the first choice when it comes to storing large volumes of physical data.

However, EOS is a low-cost disk storage solution, which, compared to AFS, is less robust when it comes to disk failures, and it does not offer a backup facility. More importantly, although the HTCondor service can handle input and output data through EOS, it still necessitates AFS storage for submission and log files.

---

[5] In practice, there are service limits due to the HTCondor fair-share policies.

[6] The meaning of the "EOS" acronym is surprisingly unclear. According to the official website, it stands for **EOS** **O**pen **S**torage, which creates a recursive situation similar to the infamous chicken-and-egg dilemma. There is a possibility to define it as **E**xabyte **O**pen **S**torage, which would be a more reasonable definition. However, no concrete evidence is provided in the official documentation to support this claim.

Therefore, while CERN has indicated[182] intentions to eliminate AFS, it is still widely used along with EOS.

Reconstruction software framework

To address the numerous challenging problems attempted in this research, a software framework called ANNDEA[183] or **A**rtificial **N**eural **N**etwork-**D**riven **E**mulsion **A**nalysis has been developed. ANNDEA is a Python-written framework, the main purpose of which is to assist the user in all aspects of reconstruction, such as:

- Data ingestion and preparation.
- Creation and training of neural networks.
- HTCondor submission management.
- Utilising the output of neural networks.

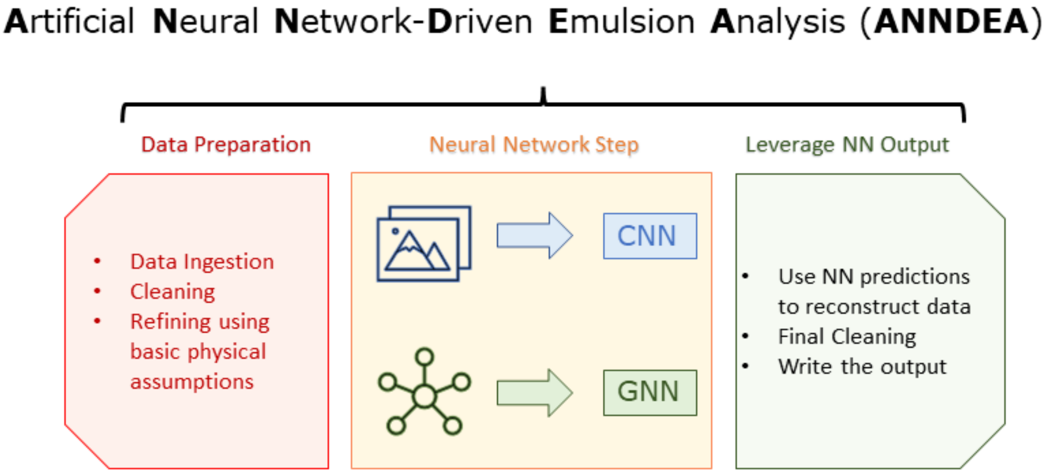


Figure 4.10: Schematic representation of the ANNDEA framework.

At the time of writing this thesis, ANNDEA consisted of five main modules that perform tracking, ECC-data alignment, track desegmentation, track classification and regression analysis, and vertex reconstruction. Thanks to the active use of *TensorFlow* and *PyTorch* libraries, ANNDEA naturally incorporates neural networks into its existing reconstruction pipelines, and its open architecture enables it to modify existing or add new reconstruction modules easily. Provided that it runs on the terminal multiplexer such as *tmux* with appropriate authentication tokens, it can perform tasks in the fully automatic mode without the need for user intervention.

Despite ANNDEA capabilities, the full description of which is beyond the scope of the thesis[7], there are some significant limitations of ANNDEA, such as:

1. ANNDEA is still in a "proof-of-concept" stage and therefore some of the aspects of its algorithm, such as data preparation and interaction with the HTCondor service, are not optimised to their full potential. For example, ANNDEA

---

[7] A comprehensive documentation can be found on a dedicated wiki page available at [184].

actively uses *Pandas* data frames for data preparation, which are memory-consuming and can become a significant bottleneck of the reconstruction speed and performance.

2. Currently, ANNDEA is not yet integrated into the native SND@LHC *sndsw* framework, which presents an additional data exchange barrier to the collaboration as a whole.
3. Unlike powerful frameworks such as ROOT, ANNDEA itself has very limited visualisation capabilities, necessitating third-party software packages.

Other tools

In addition to ANNDEA, a set of third-party tools has been used that includes:

Tableau Desktop: Tableau Desktop has been used to produce the bulk of graphs and event visualisations for this report. Conceptually, Tableau offers a shallower learning curve compared to MATLAB or ROOT, enabling the user to produce complex visualisations with minimal effort. One limitation of Tableau software is its focus on the commercial data analyst market, which means it may not have all the functionalities needed for scientific purposes, such as heatmaps. However, it remains a robust tool and is available to students through a complimentary one-year student license.

Matplotlib: to fill some gaps in the Tableau visualisation capabilities, the matplotlib Python library has been used to produce some of the plots used in this thesis.

Tableau Prep: while in principle the Panda's library can perform any required data cleansing and manipulation, it requires a significant level of laborious coding that is only justifiable for the routine data pipelines, but not for the ad hoc or one-off analysis tasks. Tableau Prep provides an intuitive graphic user interface that enables the end user to perform complex data cleansing and transformation with little to no coding. Certainly, it lacks the sophisticated features present in more advanced visual data preparation platforms such as *Alteryx Designer*, but it is a more affordable option since it comes as a complementary package to the Tableau Desktop.

4.3.4 Measuring performance

Defining and assessing reconstruction performance is important for several reasons:

- It enables the evaluation of the efficiency of different reconstruction processes.
- Helps identify areas for improvement and optimisation.
- Offers a uniform measurement that allows for the comparison of ANNDEA and standard reconstruction techniques, with the latter serving as a reference point.

Overall, it offers an impartial quantitative evaluation of the performance of this study.

Three distinct aspects of the reconstruction pipeline are evaluated using various sets of metrics: the performance of neural network training, the reconstruction data cut flow, and the quality of the final output.

Neural network training monitoring

For classifier neural networks, two key metrics are used to assess learning performance: *cross entropy loss* defined earlier in Equation 2.44 and *accuracy*, which is simply a number of data points $\mathbf{x}_i$ for which the ANN predictions $\mathbf{p}_i$ match the labels $\mathbf{y}_i$, divided by the total number of evaluations. The range of values for this parameter is 0 to 1. Typically, values between 0.5 and 1.0 are observed. A value of 0.5 suggests that neural network predictions are essentially random, while values closer to 1.0 indicate better learning performance. It is uncommon to find values significantly below 0.5, and if they do occur, it may be an indication of incorrect labelling in the validation dataset.

Although accuracy is intuitively easier to understand, compared to the loss metric, it suffers from a degree of information degeneracy when it comes to measuring the performance of the classification of neural networks. This can be demonstrated by a hypothetical example in which two neural networks, A and B, perform a binary classification of a data point $\mathbf{x}$. In this example, neural network A assigns a probability $\mathbf{p}_A = [0.0, 1.0]$ and neural network B assigns a probability $\mathbf{p}_B = [0.49, 0.51]$. If the label $\mathbf{y}$ is $[0.0, 1.0]$, both neural networks will have a 100% accuracy for that specific data point, although prediction A is closer to the value of the label $\mathbf{y}$ compared to model B. Because CEL can capture this information, it is a more reliable measure to evaluate the progress of neural network training compared to accuracy.

The accuracy metric has another notable drawback in that it can be misleading when solely relied upon to gauge the neural network's performance in testing scenarios. This can be demonstrated by a hypothetical example in which the binary classification neural networks A and B train in different training sets with 50%:50% label split and both reach an accuracy of 95%. If neural network A is evaluated using data that have the same equal class distribution, while model B is evaluated using data that are heavily skewed towards one label, neural network A will probably produce a similar performance to the one exhibited during the training. The same cannot be stated for model B, where the acknowledged 5% error rate can lead to a significant number of misclassified data points that belong to the prevailing label.

Another complication arises from the fact that the accuracy score measures the overall rate of correct predictions for the given dataset, which is an average of the prediction rate for each data class. However, a neural network can perform well in one data class but struggle in another. Depending on the magnitude of this asymmetry and the relative distribution of the test data labels, the numerical network evaluation performance will produce different results.

For neural networks that perform regression tasks, only one metric is sufficient to monitor NN training performance, called *mean squared error loss* or MSEL, which is defined through the equation:

$$L_{MSEL}(\theta) = \frac{1}{N}\sum_{i=0}^{N}(p_i - y_i)^2. \tag{4.1}$$

An attractive feature of the MSEL is that, in the context of momentum or energy prediction, it has a straightforward interpretation of a measurement resolution.

Reconstruction data cutflow

This part concerns the evaluation of seed refining performance in various filtering steps, including fiducial and neural network cuts. Each step in the cutflow results in the following types of seeds described in the table below:

Table 4.5: Confusion matrix for seed binary classification with desired elements (Green) and undesired elements (Red).

	Genuine seed	**Fake seed**
Passed the filter	True Positive (**TP**)	False Positive (**FP**)
Rejected by the filter	True Negative (**TN**)	False Negative (**FN**)

In the ideal scenario, the reconstruction cut flow should not yield any off-diagonal entries, which are the consequence of the rejection of genuine seeds or acceptance of fake random pairs. Generally, two metrics are used to quantify these effects: *recall* and *precision*.

Recall measures the percentage of genuine seeds that successfully pass through the filter in relation to their initial quantity. It is computed using the following formula:

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}. \tag{4.2}$$

Precision is a metric that quantifies the proportion of authentic seeds among all seeds that pass through the filter. It is computed using the following formula:

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}. \tag{4.3}$$

The primary aim of efficient reconstruction is to maximise both quantities at the end of the flow. However, in practice, it is difficult to achieve this objective because improving one metric frequently results in a decline of another. Typically, when the cut becomes more stringent, there is an increase in the number of authentic seeds that fail it, and vice versa: relaxing the criteria results in a higher number of false positives. The primary purpose of integrating deep learning into the reconstruction process is to mitigate this trade-off by utilising all the accessible seed information. Consequently, it enables a more precise selection process that traditional techniques cannot achieve, as they depend on a restricted set of variables for candidate classification.

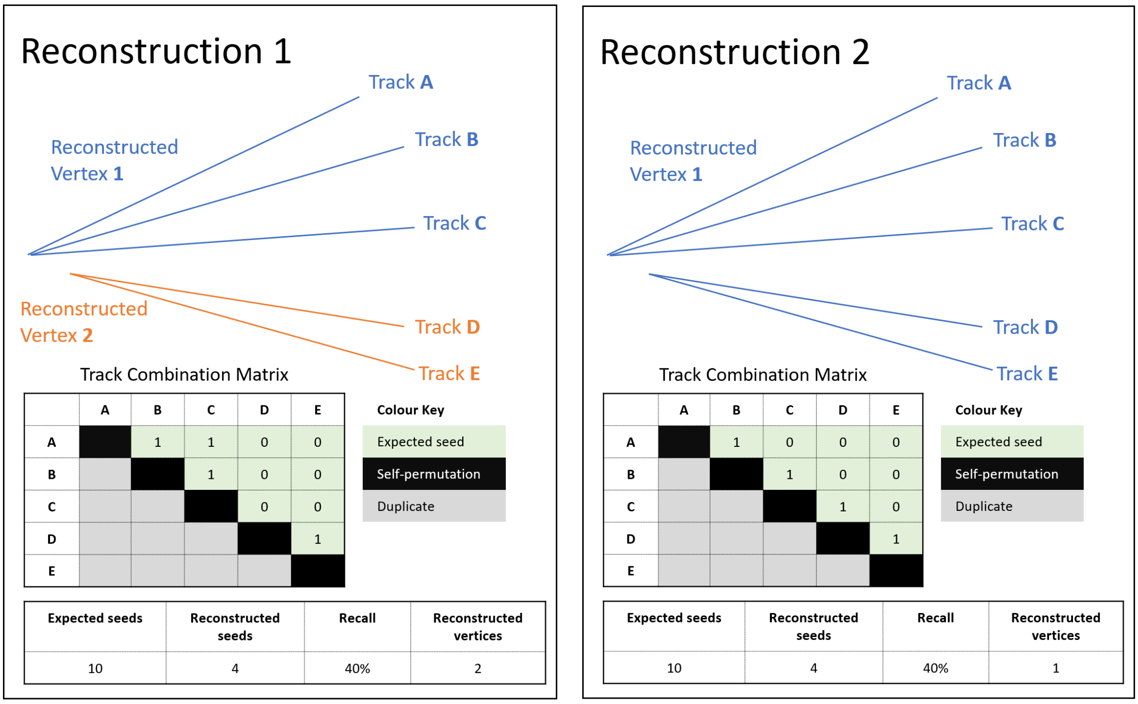
Nonetheless, using neural networks alone for large volumes of unrefined seeds is computationally expensive. Therefore, it is more practical to combine these two methods. By employing simple and fast algorithms at the beginning of the flow, obvious fakes can be rejected, while the neural network handles fewer but more challenging inputs. Within this schema, the preselection steps should have the highest computationally possible efficiency to prevent an irreversible loss of signal before the ANN step and the overall loss of efficiency as a result.

The step involving neural networks has more room for deciding the optimal point between precision and recall, which in principle can be optimised by using a combined metric such as F-score, which is defined as:

$$F_1 = 2 \times \frac{Recall \times Precision}{Recall + Precision}. \tag{4.4}$$

However, to demonstrate the capabilities of machine learning-driven methods compared to the conventional techniques using the Kalman filter, a more simple approach will be taken. It consists of tuning the reconstruction workflow so that its recall approximately matches that of FEDRA and comparing the precision between the two reconstruction methods.

Although the result of the comparison can help guide the direction of the development of the reconstruction algorithm, it cannot be used as conclusive evidence that one technique performs better than another. This is because recall and precision encode limited information and do not reflect the exact reconstructed seed configuration. It is perfectly possible for two methods to have the same recall and precision values, but the reconstructed state of the final physics event can be different, such as in the illustrated example below.



Reconstruction 1

	A	B	C	D	E
A	■	1	1	0	0
B	▨	■	1	0	0
C	▨	▨	■	0	0
D	▨	▨	▨	■	1
E	▨	▨	▨	▨	■

Expected seeds	Reconstructed seeds	Recall	Reconstructed vertices
10	4	40%	2

Reconstruction 2

	A	B	C	D	E
A	■	1	0	0	0
B	▨	■	1	0	0
C	▨	▨	■	1	0
D	▨	▨	▨	■	1
E	▨	▨	▨	▨	■

Expected seeds	Reconstructed seeds	Recall	Reconstructed vertices
10	4	40%	1

Figure 4.11: A schematic illustration of a hypothetical situation where a reconstruction of a single vertex is performed by two different models. Both models achieve the same recall, but their results are different. **Left:** Model 1 has identified all seeds that group tracks A, B and C and a seed connecting tracks D and E. Since there is no link between the groups, they are reconstructed as two different vertices. **Right:** Model 2 has identified all seeds AB, BC, CD, and DE. Also, some seeds, such as AE or AC, have been rejected by the model, filters the configuration of seeds enables them to group into one vertex.

Quality of the final output

While the previous two evaluations provide important information on the performance of the reconstruction workflow, the quality of the output provides a final assessment of the overall algorithm and its value for physics analysis. Some of the metrics that are used for this task have already been mentioned in Section 4.1 and will be covered in more detail here:

TRE: this parameter measures the number of MC simulated tracks that were recognised to some extent by the tracking algorithm. To calculate it, these steps are followed:

1. Extract all MC-generated tracks from the reconstructed dataset and eliminate any tracks with fewer than four hits. By default, this study will apply the *short track cut*, which represents the minimum number of hits per segment, to all volume tracks. Henceforth, it will not be reiterated for the sake of brevity.
2. Calculate the total number of tracks from the previous step.
3. Extract all the tracks reconstructed by the tracking algorithm.
4. Take the MC simulated tracks from Step 1 and match them with the reconstructed tracks from Step 3. The concept of *track matching* refers to the manipulation of data in which the subject track, reconstructed or generated by one algorithm, is mapped to the corresponding track of another algorithm. The link is established by counting the number of shared hits with a minimum accepted number of one[8]. When the MC-generated track is mapped to multiple track segments reconstructed by the counterpart algorithm, the segment with the highest hit overlap is chosen.
5. Count the number of matched MC simulated tracks from the previous step and divide by the result from Step 2.

THRE: this criterion quantifies how complete the reconstructed track is compared to its MC simulation counterpart. Its calculation steps are as follows:

1. Extract the suitable MC simulated tracks from the reconstructed dataset.
2. Extract all tracks reconstructed by the tracking algorithm.
3. Take the reconstructed tracks from the previous step and match each of them with the MC-generated tracks.
4. For all matched simulated tracks, count the original number of their constituent hits. If two or more reconstructed tracks share the same simulated track, the number of hits for the latter is counted multiple times.
5. Count the total number of shared hits used in the matching procedure and divide by the result from the previous step.

---

[8] This is a much more relaxed criterion than the one applied to the definition of the simulated and reconstructed tracks. It has been chosen deliberately to make these metrics more robust and consistent for future cases involving the reconstruction of short tracks, making up electromagnetic showers as an example. Occasionally, there will be cases where the reconstructed track is matched to the MC-simulated counterpart through accidentally using one of its hits; however, these effects are captured better in other metrics.

THRP: This criterion quantifies how many wrong hits have been used by the tracking procedure to reconstruct a track. Its calculation steps are as follows:

1. Extract all the reconstructed tracks from the dataset and count the total number of constituent hits.
2. Match the reconstructed segments with the MC simulated tracks and count the number of shared hits.
3. Divide the result from the previous step by the result of Step 1.

Segmentation: is a metric that describes the number of segments reconstructed per truth particle. This can be calculated using the following steps:

1. Extract all the tracks generated by the Monte-Carlo simulation from the reconstructed dataset.
2. Extract all tracks reconstructed by the tracking algorithm.
3. Match the tracks obtained in Step 1 with the tracks from Step 2, keeping a record of all matched segments, even if there are multiple segments per MC simulated track.
4. Calculate the total number of simulated tracks from the previous step and then divide it by the total number of matched reconstructed segments.

VRE: this parameter quantifies the number of suitable MC-generated vertices that were at least partially reconstructed by the vertexing algorithm. Its calculation steps are as follows:

1. Extract all the MC simulated vertices and eliminate any vertices with fewer than two constituent tracks. By default, this *vertex multiplicity cut* will apply to all vertices.
2. Extract all vertices reconstructed by the vertexing algorithm.
3. Each track that belongs to the reconstructed vertex is matched with the corresponding MC-generated track. This step is required to perform vertex matching in the next step.
4. Take the simulated vertices from Step 1 and match them with the reconstructed vertices from Step 3. Similarly to track matching, *vertex matching* refers to the manipulation of data in which the subject vertex, reconstructed or generated by one algorithm, is mapped to the corresponding vertex of another algorithm. The link is established by counting the number of shared tracks with a minimum accepted number of two. If the MC simulated vertex is mapped to multiple distinct vertices reconstructed by the counterpart algorithm, the vertex with the highest track overlap is chosen.
5. Divide the total number of matched vertices by the result of Step 3.

VTRE: this criterion quantifies the effectiveness of FEDRA in reproducing the true vertex multiplicity of its simulated counterpart. Its calculation steps are as follows:

1. Extract all the simulated vertices.

2. Extract all the vertices generated by the vertexing algorithm and match their constituent tracks with their Monte-Carlo simulated counterparts.
3. Match the reconstructed vertices from the previous step with the MC simulated vertices.
4. Count the total number of tracks that overlap for the matched vertices and divide by the original number of tracks that make up the matched MC-generated vertex. If two or more reconstructed vertices share the same MC-simulated vertex, the number of tracks for the latter is counted multiple times.

VTRP: this criterion quantifies the rate of fake tracks that make up the reconstructed vertices. Its calculation steps are as follows:

1. Extract all Monte-Carlo-generated vertices.
2. Extract all vertices reconstructed by the vertexing algorithm and, for each track that belongs to the reconstructed vertex, match the simulated track.
3. Count the total number of tracks that belong to the reconstructed vertices from the previous step.
4. Match the vertices from the previous step with the simulated counterparts.
5. Count the total track overlap for the matched vertices and divide by the original number of tracks that make up the matched MC-generated vertex. If two or more reconstructed vertices share the same simulated vertex, the number of tracks for the latter is counted multiple times.

So far, the metrics described in this section require truth-level information from the simulation for calculation, which makes them unusable for the actual data, which lack any truth information. To address this challenging limitation, a set of additional methodologies is also applied.

- **Objective metrics:** some characteristics of the reconstruction data that are impartial to the availability of Monte-Carlo simulation metadata will be used, such as *hit utilisation*, which is simply the ratio of hits used for the tracking procedure versus the total number of hits present in the data.
- **Consistency checks:** this involves validating the picture of the reconstructed data using independently verified physical expectations. For example, the number of reconstructed long tracks in the emulsion can be compared with the observed muon flux, which has been independently measured using electronic detectors.
- **Visual inspection:** this technique involves manually reviewing the reconstructed event of interest to spot any anomalies or errors in the reconstruction process. Although it is a laborious and highly subjective method that was widely used for emulsion analysis in the mid-20th century, in combination with modern visualisation software, it remains a powerful tool in the hands of a skilled particle physicist.

4.4 Reducing track segmentation

4.4.1 Purpose

In this part, an issue related to track segmentation discussed in Section 4.1 will be addressed. The solution entails merging reconstructed track segments that the FEDRA did not identify as a single track, and consists of the following steps:

1. Clean and prepare the data that has been previously tracked by FEDRA.
2. Form all possible track combinations and refine them using basic geometrical considerations.
3. Pass the remaining seeds through the neural network filters.
4. Combine all seeds remaining from the previous steps into new tracks.

If the segment is integrated into the newly formed track, the initial tracking identifiers will be replaced with new ones, enabling them to be perceived as a cohesive entity in the data. The investigation conducted in this study will focus on the FEDRA reconstructed tracks; however, this method can be applied to the output of any tracking algorithm.

4.4.2 Data preparation

The primary objective of data preparation is to create a file containing only the necessary details to reduce the storage requirements for data reconstruction. This involves taking the initial input file and stripping any data apart from the following variables: x, y, z, tx, ty, *Brick ID* and the *Track ID*, which contains a unique identifier for the reconstructed track within a data set. Only hits used for track reconstruction and those that are part of tracks of sufficient length are considered. In case data are needed to train the neural network, an additional variable *MC Track ID* is brought in, which holds an identifier for the corresponding MC-generated track.

4.4.3 Seed refining

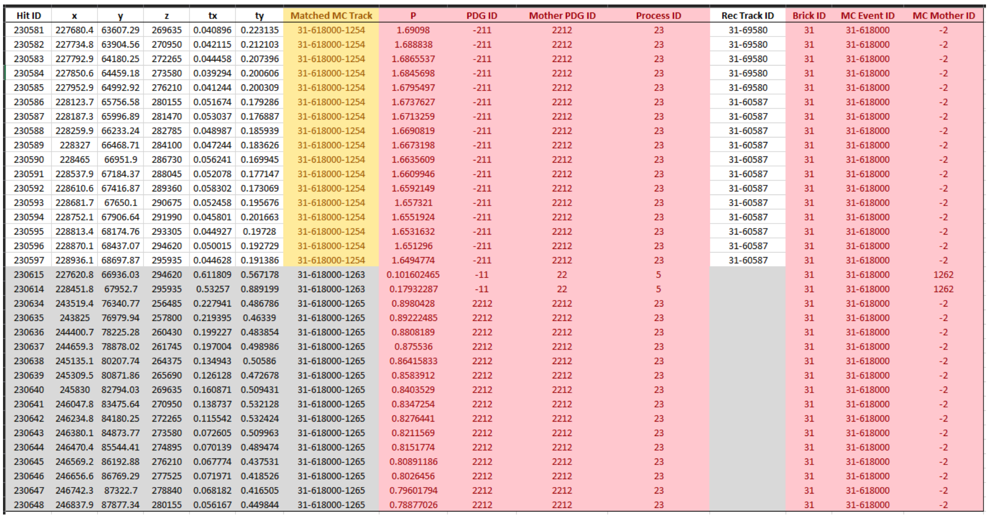
The objective of seed refinement is to identify the correct combination of the track segments that ultimately allows them to be merged into complete volume tracks. This operation can be divided into the following steps[9]: *seed initialisation*, *preliminary selection* and *fiducial cuts*, which will be explained in more detail in the following.

Seed initialisation

As previously stated, a track seed refers to a pair of track segments originating from the same parent particle track. The initial number of combinations increases

---

[9] The process described here has been expressed in the simplistic mathematical form. ANNDEA's actual workflow is more complex: it yields the same results but is optimised for computational performance.



Hit ID	x	y	z	tx	ty	Matched MC Track	P	PDG ID	Mother PDG ID	Process ID	Rec Track ID	Brick ID	MC Event ID	MC Mother ID
230581	227680.4	63607.29	269635	0.040896	0.223135	31-618000-1254	1.69098	-211	2212	23	31-69580	31	31-618000	-2
230582	227734.8	63904.56	270950	0.042115	0.212103	31-618000-1254	1.688838	-211	2212	23	31-69580	31	31-618000	-2
230583	227792.9	64180.25	272265	0.044458	0.207396	31-618000-1254	1.6865537	-211	2212	23	31-69580	31	31-618000	-2
230584	227850.6	64459.18	273580	0.039294	0.200606	31-618000-1254	1.6845698	-211	2212	23	31-69580	31	31-618000	-2
230585	227952.9	64992.92	276210	0.041244	0.200309	31-618000-1254	1.6795497	-211	2212	23	31-69580	31	31-618000	-2
230586	228123.7	65756.58	280155	0.051674	0.179286	31-618000-1254	1.6737627	-211	2212	23	31-60587	31	31-618000	-2
230587	228187.3	65996.89	281470	0.053037	0.176887	31-618000-1254	1.6713259	-211	2212	23	31-60587	31	31-618000	-2
230588	228259.9	66233.24	282785	0.048987	0.185939	31-618000-1254	1.6690819	-211	2212	23	31-60587	31	31-618000	-2
230589	228327	66468.71	284100	0.047244	0.183626	31-618000-1254	1.6673198	-211	2212	23	31-60587	31	31-618000	-2
230590	228465	66951.9	286730	0.056241	0.169945	31-618000-1254	1.6635609	-211	2212	23	31-60587	31	31-618000	-2
230591	228537.9	67184.37	288045	0.052078	0.177147	31-618000-1254	1.6609946	-211	2212	23	31-60587	31	31-618000	-2
230592	228610.6	67416.87	289360	0.058302	0.173069	31-618000-1254	1.6592149	-211	2212	23	31-60587	31	31-618000	-2
230593	228681.7	67650.1	290675	0.052458	0.195676	31-618000-1254	1.657321	-211	2212	23	31-60587	31	31-618000	-2
230594	228752.1	67906.64	291990	0.045801	0.201663	31-618000-1254	1.6551924	-211	2212	23	31-60587	31	31-618000	-2
230595	228813.4	68174.76	293305	0.044927	0.19728	31-618000-1254	1.6531632	-211	2212	23	31-60587	31	31-618000	-2
230596	228870.1	68437.07	294620	0.050015	0.192729	31-618000-1254	1.651296	-211	2212	23	31-60587	31	31-618000	-2
230597	228936.1	68697.87	295935	0.044628	0.191386	31-618000-1254	1.6494774	-211	2212	23	31-60587	31	31-618000	-2
230615	227620.8	66936.03	294620	0.611809	0.567178	31-618000-1263	0.101602465	-11	22	5		31	31-618000	1262
230614	228451.8	67952.7	295935	0.53257	0.889199	31-618000-1263	0.17932287	-11	22	5		31	31-618000	1262
230634	243519.4	76340.77	256485	0.227941	0.486786	31-618000-1265	0.8980428	2212	2212	23		31	31-618000	-2
230635	243825	76979.94	257800	0.219395	0.46339	31-618000-1265	0.89222485	2212	2212	23		31	31-618000	-2
230636	244400.7	78225.28	260430	0.199227	0.483854	31-618000-1265	0.8808189	2212	2212	23		31	31-618000	-2
230637	244659.3	78878.02	261745	0.197004	0.498986	31-618000-1265	0.875536	2212	2212	23		31	31-618000	-2
230638	245135.1	80207.74	264375	0.134943	0.50586	31-618000-1265	0.86415833	2212	2212	23		31	31-618000	-2
230639	245309.5	80871.86	265690	0.126128	0.472678	31-618000-1265	0.8583912	2212	2212	23		31	31-618000	-2
230640	245830	82794.03	269635	0.160871	0.509431	31-618000-1265	0.8403529	2212	2212	23		31	31-618000	-2
230641	246047.8	83475.64	270950	0.138737	0.532128	31-618000-1265	0.8347254	2212	2212	23		31	31-618000	-2
230642	246234.8	84180.25	272265	0.115542	0.532424	31-618000-1265	0.8276441	2212	2212	23		31	31-618000	-2
230643	246380.1	84873.77	273580	0.072605	0.509963	31-618000-1265	0.8211569	2212	2212	23		31	31-618000	-2
230644	246470.4	85544.41	274895	0.070139	0.489474	31-618000-1265	0.8151774	2212	2212	23		31	31-618000	-2
230645	246569.2	86192.88	276210	0.067774	0.437531	31-618000-1265	0.80891186	2212	2212	23		31	31-618000	-2
230646	246656.6	86769.29	277525	0.071971	0.418526	31-618000-1265	0.8026456	2212	2212	23		31	31-618000	-2
230647	246742.3	87322.7	278840	0.068182	0.416505	31-618000-1265	0.79601794	2212	2212	23		31	31-618000	-2
230648	246837.9	87877.34	280155	0.056167	0.449844	31-618000-1265	0.78877026	2212	2212	23		31	31-618000	-2

Figure 4.12: A small example of the input data in tabular form that has been subjected to the FEDRA tracking process. The red colour denotes unused columns, and the grey colour denotes rows with hits not used by tracking that are discarded from the data. Yellow is an optional column included only if the data form the training set for neural network training.

according to equation $N_{sd} = N_{tr}^2$, which can be large for a standard ECC unit. Hence, initially, only a basic seed representation in the shape of a vector is employed:

$$\mathbf{sd}_{12} = \begin{bmatrix} seg_1 & x_1 & y_1 & z_1 & seg_2 & x_2 & y_2 & z_2 & z_{sd} & label \end{bmatrix} \tag{4.5}$$

where seg_1 is a track identifier of the first segment and seg_2 is a track identifier of the second.

The elements x_1, y_1, z_1 represent the coordinates of the last hit along the z-axis for segment seg_1, and the elements x_2, y_2, z_2 represent the coordinates of the first hit along the z-axis for segment seg_2. z_{sd} is the minimum value of the z-coordinate observed for any hit within the seed.

The *label* element is not mandatory and is only included in the data sets intended for neural network training and is ignored in the seed refinement phase. It is a Boolean variable that can take a value of 1 or 0, indicating whether the constituent segments share the same MC simulated track or not.

Preliminary selection

This simple but powerful description of the seed vector described above allows the execution of the subsequent seed preselection tasks:

1. **Select unique seeds:** only seeds composed of two *different* track segments are considered. Furthermore, the ordering of the segments is irrelevant: the seed [AB] is equivalent to [BA], so only one representative is retained. Together, these two conditions reduce the number of seeds from N_{tr}^2 to $N_{sd} = \frac{N_{tr}^2 - N_{tr}}{2}$.
2. **SLG cut:** *Segment Longitudinal Gap*, or SLG, refers to the distance along the z-axis between the end hit of the preceding segment and the start hit of the following segment. It can be easily calculated using the formula $SLG = z_2 - z_1$.

In most cases of track segmentation, a positive distance is anticipated. However, it is not uncommon to encounter scenarios in the data set in which two segment tracks from the same particle overlap along the z-axis, leading to negative SLG values. This is caused by flaws in the reconstruction process when wrong hits are used for the given track. Due to the complete morphological differences between these cases, the desegmentation process has been divided into two separate phases: initially running the desegmentation process using a neural network trained specifically for these types of tracks, followed by utilising the output for a subsequent round of the desegmentation process where SLG is greater than or equal to zero with the relevant neural network model. For Phase 1, the SLG can take any value within the range $z_{sd} \leq SLG < 0$, while for the second phase, it can take values $0 \leq SLG \leq 7000$ microns. The numerical value represents the maximum computationally possible limit achieved within this research for the MC simulated domains. The effect of this cut on the seed reduction cannot be estimated analytically and depends strongly on the data set in question.

3. **STG cut:** *Segment Transverse Gap* or STG is the maximum distance along the xy plane between the end of the preceding segment and the next start of the segment, calculated as $r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Its value is dynamic and depends on the desegmentation phase: for $SLG \geq 0$, it is set according to the formula: $STG_{max} = STG_{min} + (SLG \times 0.96)$. This is done to account for possible deviations of the particle path with increasing z. For the case of the overlapping track where $SLG < 0$, this cut simply becomes $STG_{max} = STG_{min}$. The assumption here is that the tracks that overlap should be close enough together. The value of STG_{min} has been set at 160 microns as it was found to be the most optimal value to achieve the best SNR balance.

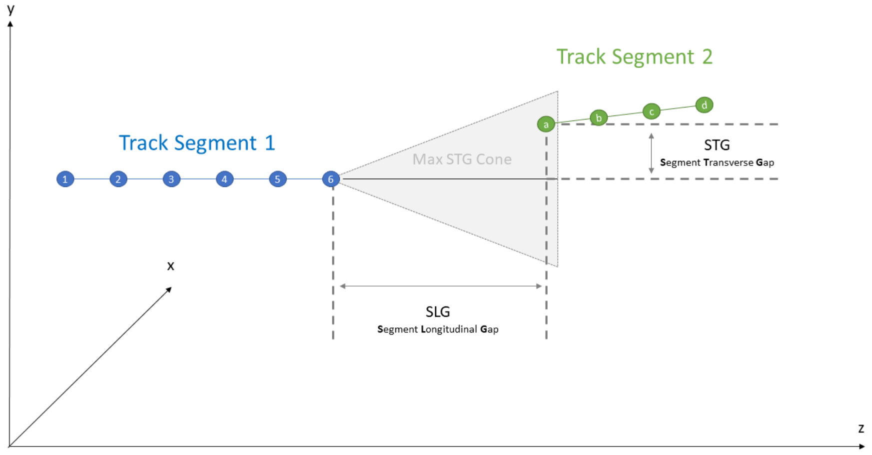


Figure 4.13: A schematic representation of the SLG and STG cuts used for the seed preselection. In this example, the track segments are well separated and do not overlap along the z-axis.

Fiducial cuts

Although the initial step greatly assists in preselecting and refining possible seeds for neural network fitting, the background-to-signal ratio remains too high to depend solely on neural networks. To address this, a more sophisticated approach is adopted, which involves examining more intricate details about the seed. The basic vector representation of the seed, as specified in the expression 4.8, is inadequate for this task. Therefore, a more complex seed object is introduced, which encompasses a comprehensive representation of the seed, such as its hit coordinates and calculated attributes. This representation allows for the performance of more complex calculations of seed parameters such as the *Distance of the Closest Approach*, abbreviated as DOCA. Originally used for vertexing processes, it is equally useful for track merging and consists of the following steps:

1. Fit straight lines for each of the track segments in the seed.
2. Find the parameters of the shortest possible vector that connects two lines from Step 1.
3. The length of the vector in Step 2 is DOCA.

In a perfect scenario, for an authentic two-track seed, the DOCA should ideally be zero because track segments from the same particle are expected to originate from a common spatial point. However, in reality, there is a certain level of dispersion resulting from the calculation of DOCA using the line equation from Step 1, which assumes a linear path of particles. As discussed in Section 2.5.4, the emulsion medium, along with imperfections in scanning and reconstruction, can significantly alter the trajectory of the reconstructed particles, particularly for particles with low momentum. Nevertheless, DOCA[10] is one of the most important parameters that help reduce the number of fake seeds, and for this desegmentation process, the chosen value leading to the optimal SNR is 100 microns.

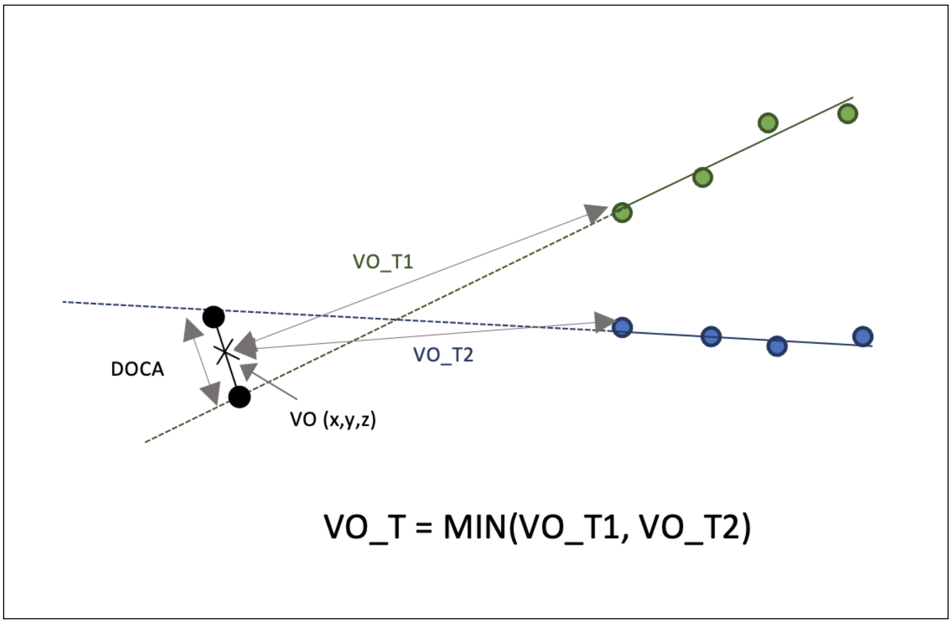


Figure 4.14: Schematic representation of how DOCA is calculated. The coloured dots represent the track hits, the coloured solid lines visualise the track trajectory, and the dotted lines represent the extrapolated lines. In this example, a vertex seed is present rather than a track segment seed; however, the DOCA definition is the same for both types.

---

[10] For track segment merging, only DOCA is used. Other parameters, such as the relative angle, have been attempted, but have eventually been discarded as ineffective.

4.4.4 Neural network training

A set of trained neural networks is used to further refine the seeds produced in the previous step. In this stage, a GNN neural network with *Gaussian mixture model*[185] convolutional operators has been employed throughout all stages, as it has shown better effectiveness compared to CNN and other GNN models. Its optimised setup includes four convolutional layers, each comprising 150 neurons.

The training set used was created by processing data from the ML domain using the data preparation and seed refinement steps described above. Each data point used for training contains a Boolean data label indicating whether both seed tracks belong to the same matched MC simulated particle or not. Two additional processing steps are needed for the training set to make it suitable for the GNN model training: resampling and creating an adequate graph representation.

The goal of resampling is to achieve a balanced distribution of fake and genuine seeds across training batches to facilitate stable training of the neural network. Failure to do so could result in scenarios where the neural network, having been exposed to a large number of uniformly labelled classes, may become optimised for classifying one label effectively while performing suboptimally for another. Despite efforts to refine the distribution of seeds, there remains a significant imbalance with fakes that heavily outnumber authentic seeds, with only approximately 1% genuine seeds observed in the simulation. The resampling process involves identifying the class with the smallest representation and adjusting the sampling of other classes accordingly to ensure a more even distribution.

One drawback of the resampling technique is that when most of the class data points encompass numerous implicit subcategories, downsampling them may cause the neural network to concentrate on a single subcategory during training. Consequently, the network may struggle with the other subcategories once it encounters the imbalanced dataset. An illustrative example can be provided using the *cats and dogs* image dataset, in which there are approximately 100 dog images for each cat picture. When downsampling dog images, the neural network may learn to differentiate between cats and the most prevalent dog breed included in the sample. The presence of uncommon dog breeds in the imbalanced dataset could potentially confuse the neural network, resulting in misclassification errors.

In the context of the research carried out here, the implicit subcategories of fake seeds are mainly driven by the physics of the event, such as the type of interactions, particle flavours involved, and their kinematics. To address this issue, a multi-tier classification method has been implemented where needed. This method entails analysing the dataset using two neural networks in a specific order, with the second neural network being trained on the dataset that has been pre-processed by the first neural network.

Another important step in seed preparation is its data representation, which the neural network can interpret and learn from. In the context of track desegmentation, a graph object has been selected with the following properties:

- Each node on the graph is represented by the hits that make up the track in the seed, with attributes that encode the spatial and angular coordinates.

- Every single node is interconnected with other nodes through non-directional edges.
- Each edge encodes several geometric properties:
  - longitudinal separation: $dz = z_2 - z_1$,
  - transverse separation in the xy plane: $dr = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$,
  - 3D distance between the two nodes: $dl = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$.
- A Boolean flag is also included to indicate whether the edge connects hits within the same or different tracks.

This compact but powerful representation provides much more information on the seed than a typical 3-D pixel image used in CNNs, but, contrary to the latter, occupies significantly less memory as only relevant nodes and their connections are encoded.

(a) Node attributes

Node ID	**Track ID**	x [μm]	y [μm]	z [μm]	tx	ty
A1	A	1	3	10,520	0.00985	-0.01042
A2	A	0	0	11,835	-0.00502	-0.00843
B1	B	2,613	2,143	0	-0.00504	-0.00197
B2	B	2,464	1,974	52,600	-0.00820	-0.00526

(b) Edge attributes

Edge ID	**Node 1**	**Node 2**	dz [μm]	dr [μm]	dl [μm]	Same track
E1	A1	A2	1,315	3.2	1,315.0	True
E2	A1	B1	-10,520	3,376.7	11,048.6	False
E3	A2	B2	40,765	3,157.2	40,887.1	False

Figure 4.15: An example of the graph representation of a seed fragment: **(a)** node attributes with spatial and angular information, **(b)** edge attributes with relative distance parameters and a Boolean flag indicating whether the hits belong to the same track. All distance values are expressed in microns, but before training, they are normalised to the range [0,1] for input to the GNN.

The steps above have been performed for both Phase 1 and Phase 2, which will be described in more detail in the following.

Phase 1

Phase 1 deals with tracks that consist of two or more hits sharing the same plate and hence the same z-coordinate. One of the most common causes of this particular type of segmentation is the reconstruction errors occurring during the tracking procedure of a muon that during propagation induces electron tracks either through bremsstrahlung or *delta ray*[11] processes. If the first hit of the emerging electron track is sufficiently close to the muon propagation path, the Kalman filter may accidentally use it instead for further muon tracking.

Typically, the initial trajectory of the electron tracks is collinear with the muon path, which can be sufficient to divert the tracking procedure from the main muon propagation path to the path of the emerging electron. The latter loses energy rapidly, leading to the discontinuation of the tracking process. The untracked part of the muon trajectory is later reconstructed as a distinct track, leading to the fragmentation of the muon's path of propagation. The plate overlap is caused by the "electron" part of the first segment.

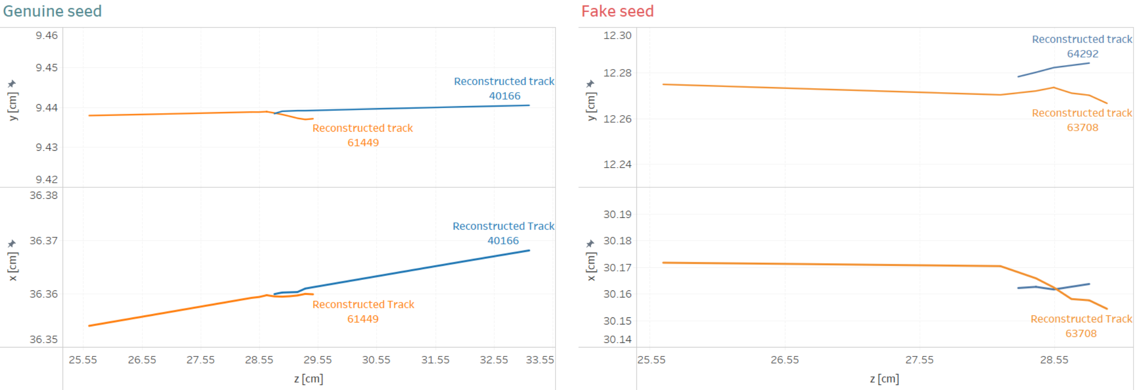


Figure 4.16: An illustration showing genuine (**Left**) and fake (**Right**) seed tracks utilised in Phase 1b training. It is important to highlight that the scales of the horizontal and vertical axes are not the same.

By identifying the simulated mother track ID of the majority of the hits belonging to the reconstructed track, it is possible to establish whether these track segments should be merged and use the ID as a class label. Since Phase 1 has used a two-stage ANN analysis process, two training sets have been generated and used for the training of the neural networks:

- **Set 1a:** A sample of the $\approx$ 36.5k train and an additional $\approx$ 3.6k validation seeds that have been used for the first stage GNN training.
- **Set 1b:** A sample of the $\approx$ 33k train and an additional $\approx$ 3.3k validation seeds that have been pre-filtered by the first stage GNN that has been trained in Set 1a.

Two GNN models have been trained with a starting learning rate of 10^{-4} with the validation loss and accuracy profile plotted in Figure 4.17 below.

The neural networks underwent training using the ANNDEA autopilot functionality, which continues to train them until it detects a plateau determined by the changing slope of the validation loss and accuracy. Subsequently, the best-performing state of the neural networks is saved and can be applied in the Phase 1 desegmentation process.

---

[11] In the framework of SND@LHC, delta rays refer to the electrons that have been displaced by the muons traversing the atoms within the emulsion detector material. Unlike bremsstrahlung, which generates both electrons and positrons, delta rays exclusively yield electrons.

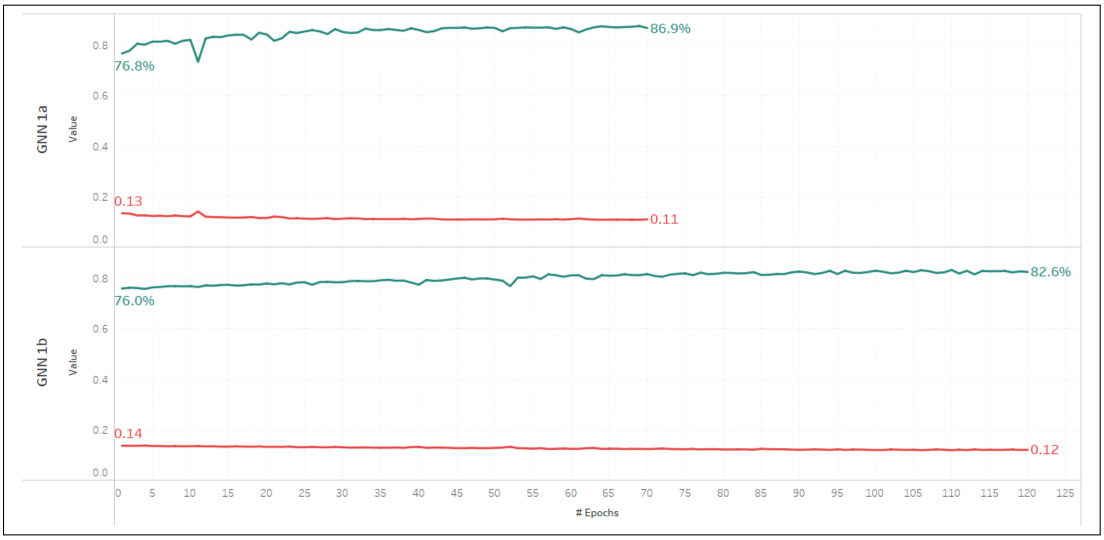


Figure 4.17: A plot of the normalised validation accuracy and CEL loss of the 1a (**Top**) and 1b (**Bottom**) GNN models training. Occasional small spikes in loss with a corresponding drop in accuracy are possible indications of instability of the local optimum neural network states with respect to the incremental weight updates.

Phase 2

Phase 2 concerns tracks with one or no constituent hits that share the same plate and coordinate z. The most common reasons for this type of segmentation have been described in Section 4.1. The neural network training routine is almost identical to Phase 1, with two key differences:

- **Higher data rates:** Given that the track segments can be distanced up to 7000 microns apart, there must be allowance for the dispersion of the radial distance between the final hit of the preceding segment and the initial hit of the following tracks. As a result, the STG cut becomes less effective in ECC units with a higher density of tracks.
- **No staging:** Only one neural network was deemed necessary, eliminating the need to pre-process the seeds by another neural network.

A training set with 100k size and a validation set with 10k seeds from the ML domain that has previously undergone Phase 1 desegmentation has been generated. Two GNN models have been trained with a starting learning rate of 10^{-4} with the validation loss and the accuracy profile plotted in Figure 4.18 below:

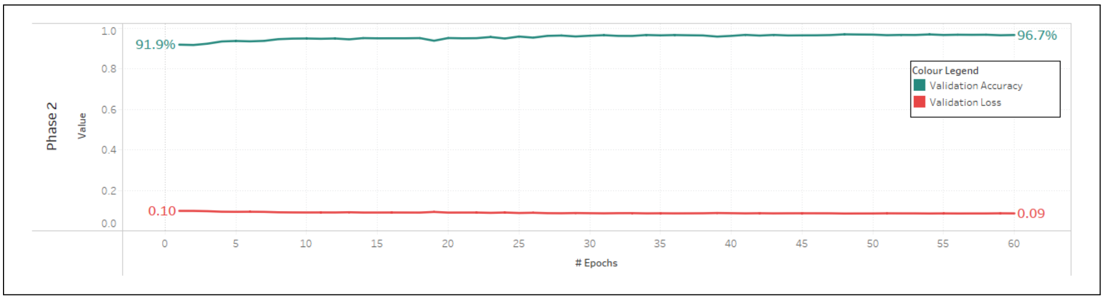


Figure 4.18: A plot of the normalised validation accuracy and CEL loss of the Phase 2 GNN models training.

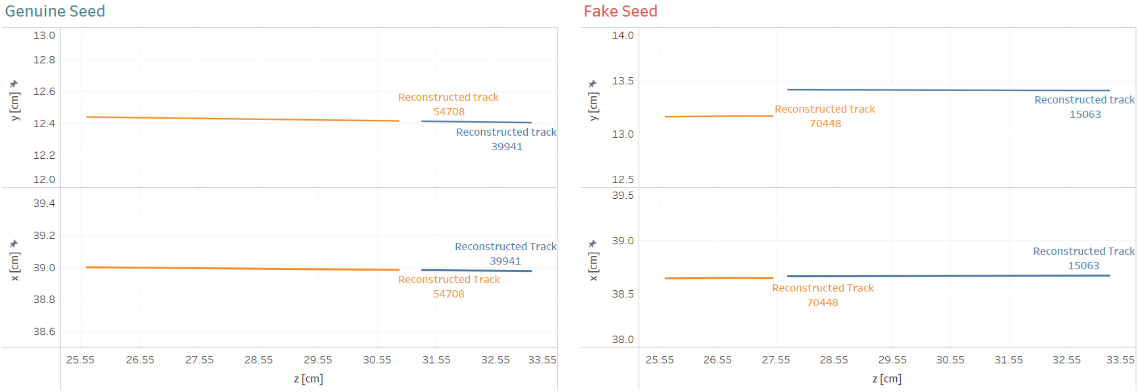


Figure 4.19: A visualisation example of the genuine (**Left**) and fake (**Right**) seed tracks used for Phase 2 training. Note that the scale is not consistent and has been manipulated for demonstration purposes.

In contrast to Phase 1 machine learning training, the Phase 2 model has reached a lower loss and higher accuracy, nearing almost 97% after 60 epochs. This improvement could be attributed to either the larger training sample size or the simpler conceptual nature of the Phase 2 seed morphology, which makes it easier to classify it using the neural network.

4.4.5 Final output

Once the neural networks have been trained, they can be used as additional filter steps that significantly improve the concentration of genuine seeds. In addition to selection, the GNN models also provide another useful output: a classification score from 0.5 to 1.0 that can be loosely interpreted as the quality of the seed fit. The task of the post-neural network step is to leverage this information to provide the most accurate reconstruction of the final events. It consists of two distinct steps: *track building* and *track smoothing* which are described in more detail below.

Track building:

The task of track building is to link all seeds that have a common track segment to a single track. For example, if there are 3 seeds: AB, BC and CD where each letter denotes a particular track segment, they can be merged into a single track that consists of A, B, C and D track segments. A hypothetical seed EF cannot be incorporated as it does not share any common track segments with any seed of the group mentioned above. However, if instead of EF a seed AF is presented, it can be used to complete the final track, provided there is no overlap between F and the tracks B,C, and D along the z-axis. If there is, this conflict is resolved by discarding the seed that attained the lowest prediction value in favour of the seed with a better fit.

Track smoothing:

Each track is expected to follow a unique path, which means that the hits within a single track should not share any plates and should have unique values of z-coordinate. However, after the seed merging procedure, particularly for Phase 1

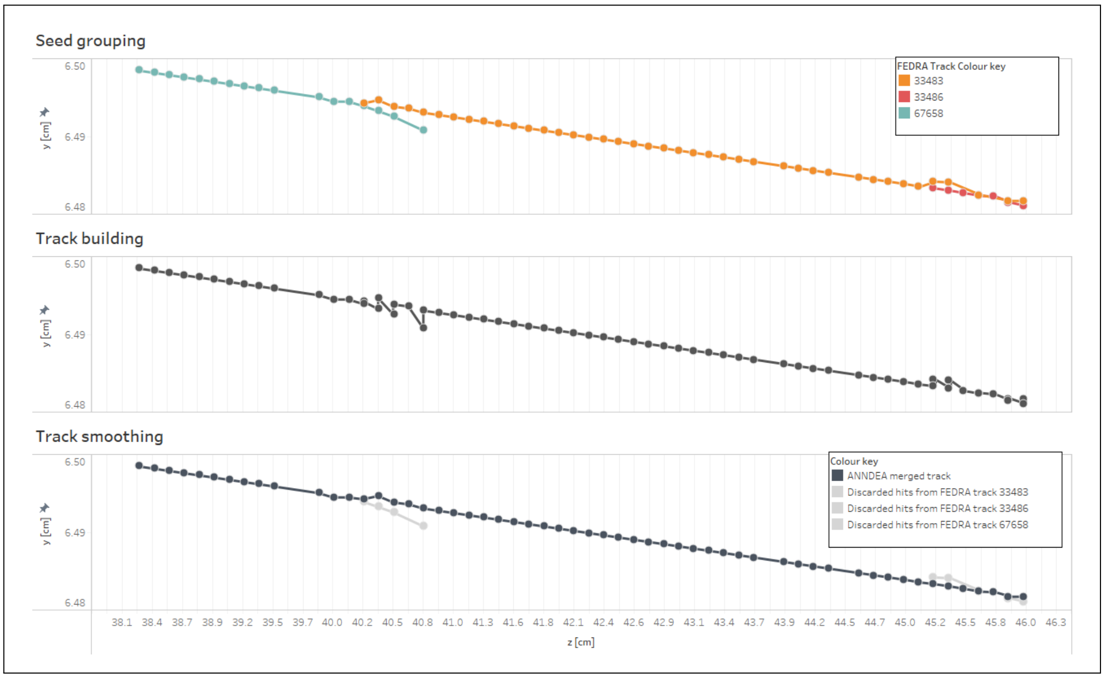


Figure 4.20: A visualised example of the seed grouping (**Top**), track building (**Middle**) and track smoothing (**Bottom**) process.

desegmentation, the final output can yield unphysical solutions where two or more hits can be located on the same plate. The purpose of the track smoothing is to solve this problem by identifying these problematic parts of the merged track and deduplicating by keeping the hits that provide the best track fit. This is done by following the steps described below.

1. Fit a second-degree polynomial $P(z)$ to the overall track. A parabola over a straight line has been chosen to better model the track's nonlinearity, which resulted in its fragmentation in the first place.

2. For each hit h calculate the absolute distance to the polynomial fitted at the particular coordinate z using the formula: $r_h(z) = \sqrt{\Delta x_h^2(z) + \Delta y_h^2(z)}$, where $\Delta x_h(z)$ and $\Delta y_h(z)$ are the components x and y of the displacement vector that connects the hit h and a corresponding point at $P(z)$.

3. If there are two or more hits within a single plate within a track, keep only the hit with the smallest value of $r_h(z)$ and discard the other.

Once the tracks have been built and smoothed, they are used to build a map that transforms the final data representation so that the fragmented tracks appear as a whole.

4.4.6 Performance on the Monte Carlo simulation

To assess the merging performance of the track segment, the Control domain in the simulated data has been subjected to both Phase 1 and Phase 2 of the desegmentation procedure.

Results

Two segmentation stages were conducted using ANNDEA, taking approximately 12 hours to complete. The desegmentation process was monitored using the ANNDEA logging feature, which, at each seed analysis step, computes and registers the distribution of seed labels. These data have presented in Figure 4.21 below:

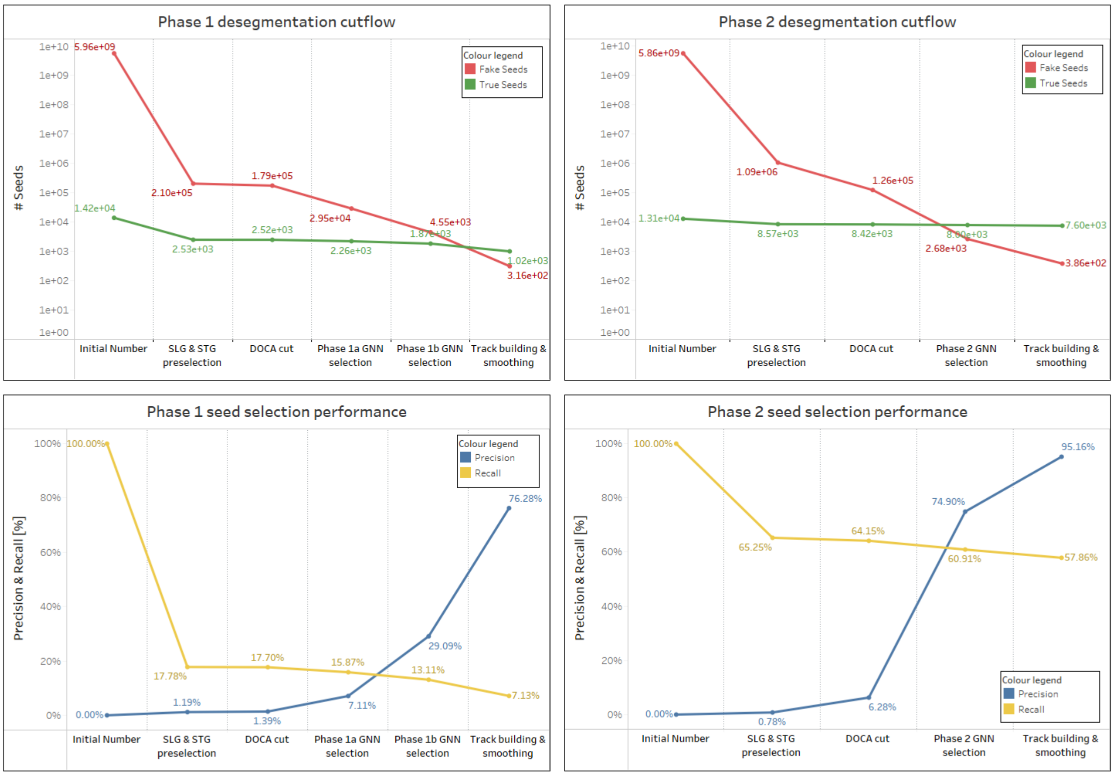


Figure 4.21: A desegmentation seed data flow performance for both Phase 1 in the left part and Phase 2 in the right part of the figure. The genuine and fake seed cutflow is plotted in the top part, while the corresponding precision and recall dynamics are reflected in the bottom graphs.

After obtaining the output of the data reconstruction, the relevant statistical and quality track metrics have been calculated and presented in Table 4.7 below. In order to quantify the significance of the impact, uncertainties were calculated for each metric using methods tailored to the type of quantity, particle source and group. The number of reconstructed tracks was modelled with a Poisson distribution, and the standard error was therefore estimated as the square root of the count. The segmentation was modelled using both binomial and Poisson statistics. The uncertainties of THRE and THRP were estimated using the bootstrap method, which involves repeatedly resampling the data and evaluating the variation of the estimates.

Discussion

The analysis of the results from the simulated data has yielded numerous interesting findings that are worth discussing in more detail.

The first observation is the lack of substantial benefit from the Phase 1 reconstruction. Both Phase 1 and Phase 2 require similar efforts in model training and reconstruction time; however, the first step provides a small amount of improvement, raising the necessity of including it at all. This can be partially explained by looking at the reconstruction cut flow in the left part of Figure 4.21 and noting the following:

- While the initial number of possible seeds for both Phase 1 and 2 is more or less comparable, by the end of the flow, the recall value for the seeds involving overlapping tracks is about eight times lower. The biggest source of this discrepancy is the STG cut that, in Phase 1, removes more than 80% of genuine seeds, limiting its maximum recall to less than 20%.
- Despite using two neural networks in sequence, their final precision is still below 30%, which is significantly lower than the one achieved by a single GNN used in Phase 2. This is partially compensated for by the track building and smoothing process, but at the cost of halving the recall to only ≈ 7 %.

Table 4.6: The control domain statistical data on the ANNDEA desegmentation performance. The impact is measured in percentage or percentage point change. Beneficial changes are coloured green, while deleterious changes are highlighted in red.

Particle source	**Particle group**	**Measured metric**	**FEDRA**	**Phase 1**	**Phase 1 impact**	**Phase 2**	**Phase 2 impact**
EM showers and delta-rays	**Electrons**	Number of tracks	29,968 ± 173	29,692 ± 172	-0.9 ± 0.8%	29,376 ± 171	-2.0 ± 0.8%
		Segmentation	1.351 ± 0.005	1.352 ± 0.005	0.1 ± 0.5%	1.350 ± 0.005	-0.1 ± 0.5%
		THRE (%)	42.1 ± 0.2	42.0 ± 0.2	-0.1 ± 0.3 pp	42.0 ± 0.2	-0.1 ± 0.3 pp
		THRP (%)	62.0 ± 0.2	62.2 ± 0.2	0.2 ± 0.3 pp	62.2 ± 0.2	0.2 ± 0.3 pp
External muons from IP1	**Muons**	Number of tracks	52,664 ± 229	51,668 ± 227	-1.9 ± 0.6%	44,894 ± 212	-14.8 ± 0.5%
		Segmentation	1.276 ± 0.005	1.254 ± 0.005	-1.7 ± 0.5%	1.097 ± 0.005	-14.0 ± 0.5%
		THRE (%)	80.2 ± 0.1	81.6 ± 0.1	1.4 ± 0.1 pp	93.7 ± 0.1	13.5 ± 0.1 pp
		THRP (%)	98.2 ± 0.0	98.3 ± 0.0	0.1 ± 0.0 pp	98.1 ± 0.0	-0.1 ± 0.0 pp
Muon neutrino interactions	**Muons**	Number of tracks	25 ± 5.0	25 ± 5.0	0.0 ± 28.3%	25 ± 5.0	0.0 ± 28.3%
		Segmentation	1.042 ± 0.208	1.042 ± 0.208	0.0 ± 28.2%	1.042	0.0 ± 28.2%
		THRE (%)	95.2 ± 4.0	95.2 ± 4.0	0.0 ± 5.7 pp	95.2 ± 4.0	0.0 ± 5.7 pp
		THRP (%)	99.8 ± 0.1	99.8 ± 0.1	0.0 ± 0.1 pp	99.8 ± 0.1	0.0 ± 0.1 pp
	Hadrons	Number of tracks	108 ± 10	109 ± 10	0.9 ± 13.2%	103 ± 10	-4.6 ± 12.8%
		Segmentation	1.325 ± 0.126	1.337 ± 0.127	0.9 ± 13.3%	1.265 ± 0.123	-4.5 ± 13.0%
		THRE (%)	68.7 ± 3.8	68.5 ± 3.8	-0.2 ± 5.4 pp	73.8 ± 3.5	5.1 ± 5.2 pp
		THRP (%)	97.1 ± 0.9	97.0 ± 1.0	-0.1 ± 1.3 pp	95.7 ±1.2	-1.4 ± 1.5 pp
μ, e, γ nuclear interactions	**Hadrons**	Number of tracks	471 ± 22	471 ± 22	0.0 ± 6.6%	380 ± 19	-19.3 ± 5.5%
		Segmentation	1.668 ± 0.077	1.668 ± 0.077	0.0 ± 6.5%	1.346 ± 0.069	-19.3 ± 5.6%
		THRE (%)	45.7 ± 1.8	45.7 ± 1.8	0.0 ± 2.5 pp	59.3 ± 2.1	13.6 ± 2.8 pp
		THRP (%)	99.2 ± 0.4	99.2 ± 0.4	0.0 ± 0.6 pp	97.7 ± 0.6	-1.5 ± 0.7 pp
Hadron decays and inelastic scattering	**Hadrons**	Number of tracks	1,434 ± 38	1,434 ± 38	0.0 ± 3.7%	1,045 ± 32	-27.1 ± 3.0%
		Segmentation	1.796 ± 0.047	1.796 ± 0.047	0.0 ± 3.7%	1.347 ± 0.041	-25.0 ± 3.0%
		THRE (%)	37.9 ± 0.9	37.9 ± 0.9	0.0 ± 1.3 pp	53.9 ± 1.2	16.0 ± 1.5pp
		THRP (%)	99.6 ± 0.1	99.5 ± 0.2	-0.1 ± 0.2 pp	97.7 ± 0.4	-1.9 ± 0.4 pp

Phase 2, on the contrary, has demonstrated positive impacts, especially on hadrons and incoming muons, by decreasing their segmentation by approximately 20-25% and around 14%, respectively. This leads to an improvement in the efficiency of using the constituent track hits (THRE), albeit with a slight decrease in their purity. This phenomenon occurs because even though there are rigorous efforts to eliminate fake seeds, a small portion still finds its way into the final output. This results in traces consisting of segments left by various particles, ultimately leading to a reduction in the THRP value.

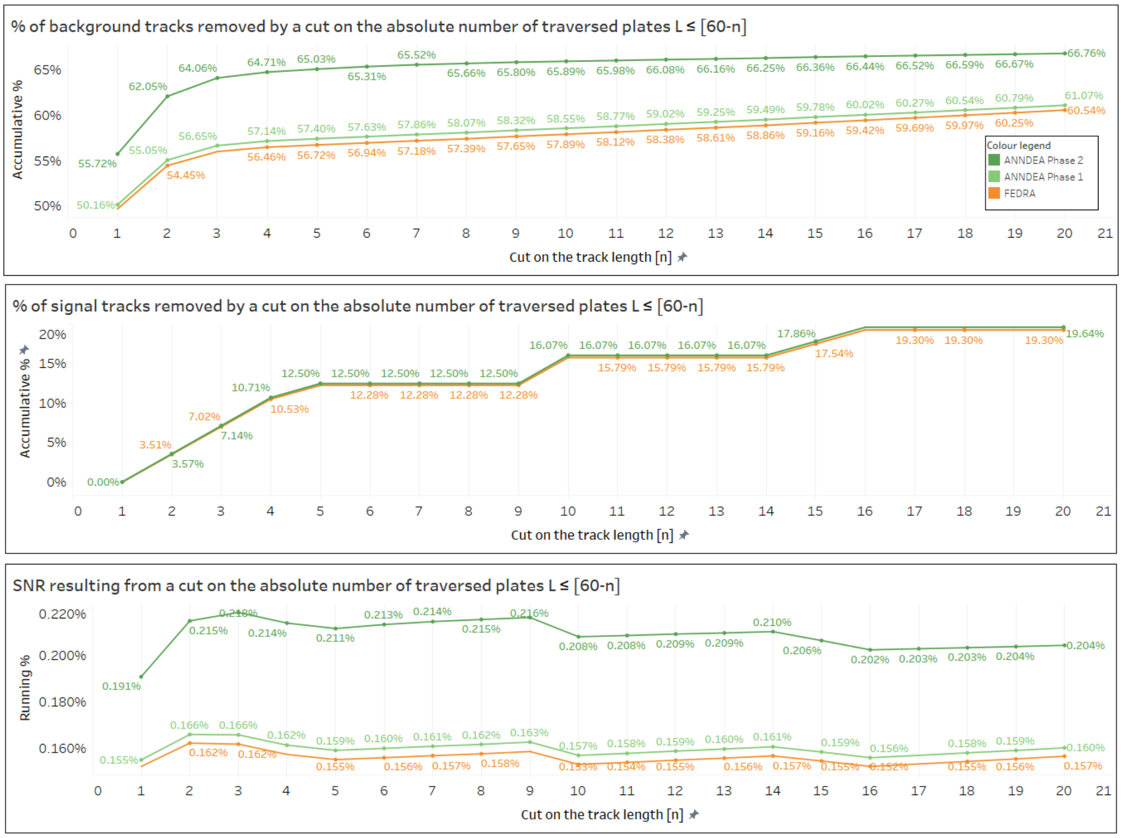
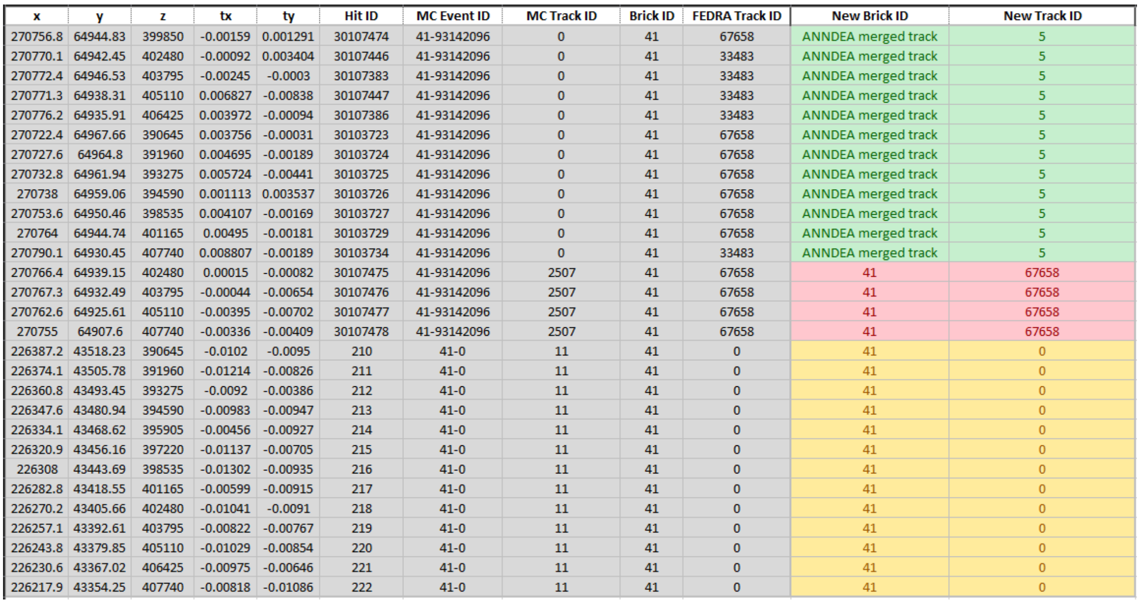


Figure 4.22: A plot on the background elimination impact on the control domain data that is made possible through desegmentation. Additionally, the exclusion of tracks with a hit on the initial emulsion plate is done with the $n = 1$ cut. The term *signal tracks* refers to tracks originating from muon neutrino interactions specifically within the B41 brick.

It is clear from Table 4.7 that desegmentation has a minor impact on signal events related to muon neutrino interactions, where only a small number of hadrons emerging from the muon neutrino interaction are involved in the desegmentation process. This is not surprising considering that the tracks left by particles in the signal events do not suffer from the fragmentation effects and therefore have little room for improvement.

As previously highlighted, the primary goal of the desegmentation procedure is to assist in reducing the absolute number of background track segments left by incoming muons that can potentially interfere with the vertex reconstruction stage. This will be revisited in Section 4.6.5 devoted to vertexing studies. Meanwhile, the benefit that the desegmentation process brings to the relative reduction of background was quantified in a set of graphs presented in Figure 4.22 above.

As it can be seen in this graph, the ANNDEA segmentation procedure helps to



x	y	z	tx	ty	Hit ID	MC Event ID	MC Track ID	Brick ID	FEDRA Track ID	New Brick ID	New Track ID
270756.8	64944.83	399850	-0.00159	0.001291	30107474	41-93142096	0	41	67658	ANNDEA merged track	5
270770.1	64942.45	402480	-0.00092	0.003404	30107446	41-93142096	0	41	33483	ANNDEA merged track	5
270772.4	64946.53	403795	-0.00245	-0.0003	30107383	41-93142096	0	41	33483	ANNDEA merged track	5
270771.3	64938.31	405110	0.006827	-0.00838	30107447	41-93142096	0	41	33483	ANNDEA merged track	5
270776.2	64935.91	406425	0.003972	-0.00094	30107386	41-93142096	0	41	33483	ANNDEA merged track	5
270722.4	64967.66	390645	0.003756	-0.00031	30103723	41-93142096	0	41	67658	ANNDEA merged track	5
270727.6	64964.8	391960	0.004695	-0.00189	30103724	41-93142096	0	41	67658	ANNDEA merged track	5
270732.8	64961.94	393275	0.005724	-0.00441	30103725	41-93142096	0	41	67658	ANNDEA merged track	5
270738	64959.06	394590	0.001113	0.003537	30103726	41-93142096	0	41	67658	ANNDEA merged track	5
270753.6	64950.46	398535	0.004107	-0.00169	30103727	41-93142096	0	41	67658	ANNDEA merged track	5
270764	64944.74	401165	0.00495	-0.00181	30103729	41-93142096	0	41	67658	ANNDEA merged track	5
270790.1	64930.45	407740	0.008807	-0.00189	30103734	41-93142096	0	41	33483	ANNDEA merged track	5
270766.4	64939.15	402480	0.00015	-0.00082	30107475	41-93142096	2507	41	67658	41	67658
270767.3	64932.49	403795	-0.00044	-0.00654	30107476	41-93142096	2507	41	67658	41	67658
270762.6	64925.61	405110	-0.00395	-0.00702	30107477	41-93142096	2507	41	67658	41	67658
270755	64907.6	407740	-0.00336	-0.00409	30107478	41-93142096	2507	41	67658	41	67658
226387.2	43518.23	390645	-0.0102	-0.0095	210	41-0	11	41	0	41	0
226374.1	43505.78	391960	-0.01214	-0.00826	211	41-0	11	41	0	41	0
226360.8	43493.45	393275	-0.0092	-0.00386	212	41-0	11	41	0	41	0
226347.6	43480.94	394590	-0.00983	-0.00947	213	41-0	11	41	0	41	0
226334.1	43468.62	395905	-0.00456	-0.00927	214	41-0	11	41	0	41	0
226320.9	43456.16	397220	-0.01137	-0.00705	215	41-0	11	41	0	41	0
226308	43443.69	398535	-0.01302	-0.00935	216	41-0	11	41	0	41	0
226282.8	43418.55	401165	-0.00599	-0.00915	217	41-0	11	41	0	41	0
226270.2	43405.66	402480	-0.01041	-0.0091	218	41-0	11	41	0	41	0
226257.1	43392.61	403795	-0.00822	-0.00767	219	41-0	11	41	0	41	0
226243.8	43379.85	405110	-0.01029	-0.00854	220	41-0	11	41	0	41	0
226230.6	43367.02	406425	-0.00975	-0.00646	221	41-0	11	41	0	41	0
226217.9	43354.25	407740	-0.00818	-0.01086	222	41-0	11	41	0	41	0

Figure 4.23: An illustrative instance of the tracking data presented in a tabular format following the ANNDEA desegmentation process. The original data is highlighted in grey, the remapped tracking representation is shown in green, the FEDRA tracks that were not involved in the merging are highlighted in yellow, and the hits rejected by the smoothing procedure are indicated in red rows.

remove up to ≈ 8% larger number of background tracks without any significant impact on the signal than if the original FEDRA data were used. This in turn results in a more favourable SNR, especially with the most optimal ratio achieved with $n = 3$ at which tracks with lengths of more than 57 plates are removed.

4.4.7 Performance on Run 1 data

To evaluate the performance of the ANNDEA reconstruction on actual data, a Run 1 data set, as shown in Figure 4.9, was subjected to the desegmentation procedures of both Phase 1 and Phase 2. Unlike the quick and seamless desegmentation process observed in the Monte Carlo simulation, the reconstruction of the Run 1 dataset required a considerably longer period of approximately 10 days.

Several factors have led to this slowness, with the primary issue being a higher track density in the data, leading to a large number of seeds, particularly during the Phase 2 stage. This exceeded the computing and memory capacity of both the local Lxplus and HTCondor machines, requiring further code optimisation. Unfortunately, to expedite the reconstruction process and reduce pressure on the computing infrastructure, the SLG cut criteria had to be reduced from 7000 to 4000 microns.

Results

Since Run 1 data lack any Monte Carlo metadata, a reduced version of the analysis has been performed, which consisted of plotting the track length distribution and comparing it with the corresponding plot for the simulation in Figure 4.24.

Along with the geometrical analysis, a visual inspection of a random sample of the merged tracks has been performed to identify any potential issues plottted on the Figure 4.26 below.

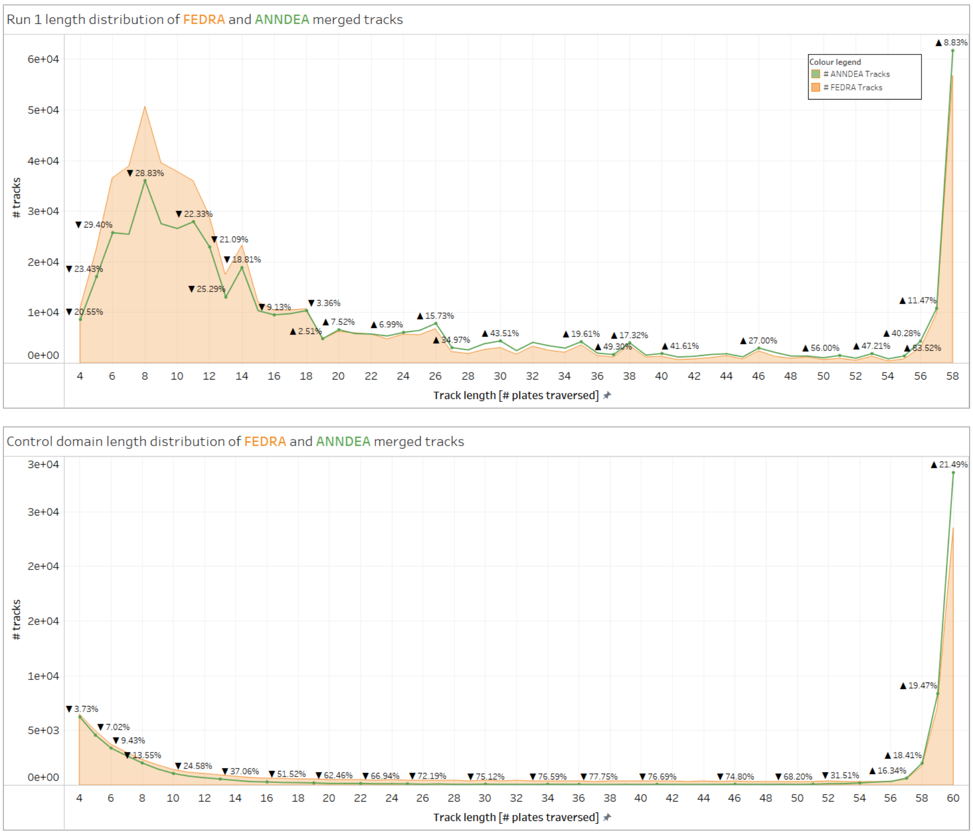


Figure 4.24: Distribution of original tracks from FEDRA (orange filled area) for the Run 1 data sample (**Top**) and the simulation of tracks in the control domain (**Bottom**) with their length quantified in the number of plates crossed. The green line shows in both cases the length distribution after the ANDNDEA merging process. The text annotations show the percentage difference in the number of tracks for each specific length. It is important to mention that the data set used consisted of only 58 scanned emulsion plates.

Discussion

The detailed analysis of the pre- and post-desegmentation track length distribution indicates an overall positive shift in the track distribution towards the longer lengths, as expected. However, contrary to what has been observed in the simulated data, the ANNDEA performance is significantly lower in the actual data: for example, the number of the longest tracks is only reduced by $\approx 8\%$ compared to over $\approx 22\%$ achieved in simulation. Multiple reasons could explain this:

- **More stringent SLG cut** imposed for Phase 2 as part of the computing intensity reduction measures. Unfortunately, it has also reduced the recall at the seed processing stage, resulting in fewer tracks being merged. To estimate the impact of this measure, the control set was rerun with the same SLG settings and compared with the state of the data produced by the nominal ANNDEA configuration in Appendix B. There is a clear ≈ 5 percentage points drop in the desegmentation performance compared to the nominal ANNDEA configuration for the longest tracks, but it is still not sufficient to explain the drop of ≈ 14 percentage points in the defragmentation performance of the Run 1 data.
- **Lower rates of FEDRA track segmentation**, which would result in fewer

opportunities for the desegmentation process. This is very unlikely, as there are no objective reasons to expect that FEDRA will produce more complete tracks in the data set with much higher hit density and that has lower hit reconstruction quality due to the artefacts of the scanning procedure. Another strong argument against this hypothesis is the presence of a significant number of tracks with lengths between 28 and 54 plates that are certainly caused by external muons that have been reconstructed as multiple segments by FEDRA.

- **Disparity between the simulation and Run 1 data**, which results in ANNDEA experiencing a drastically different environment when processing the Run 1 data compared to the simulation. This disparity can be driven by different factors, such as distortion of the reconstructed hit coordinates and higher density, which leads to poorer FEDRA reconstruction performance and consequently to the lower quality of the tracks, which in turn can lead to their false rejection at the desegmentation process. The higher density of tracks can also affect ANNDEA directly by overloading it with fake seeds that were previously not observed during neural network training and that can potentially disrupt ANNDEA from building a complete track segment sequence.

The effectiveness of background removal has also decreased compared to simulation studies, as can be seen in the corresponding Figure 4.25. This outcome is anticipated because the cut is designed to exclude any track that begins at a first plate or tracks that have a length exceeding $58 - n$, where the preferred value for n is generally up to 3 to avoid a substantial decrease in signal yield.

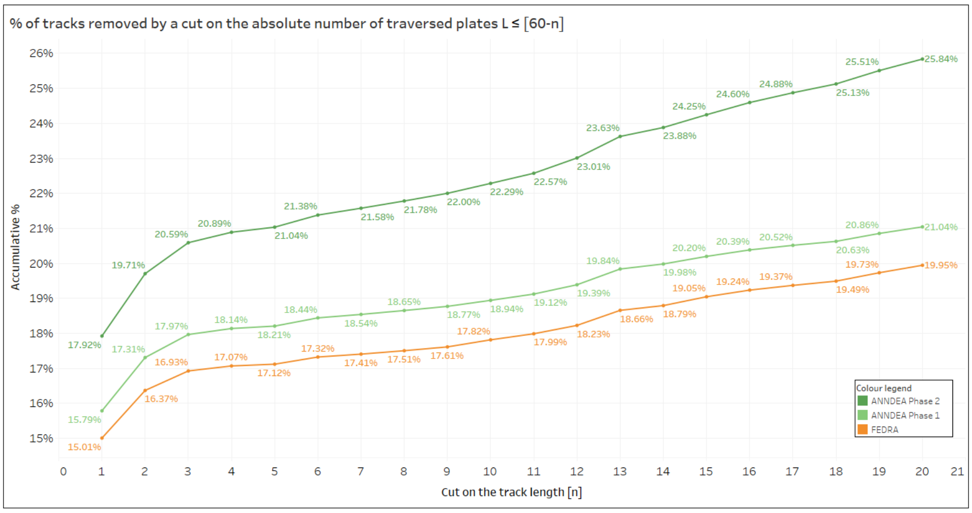


Figure 4.25: The plots of the background removal effect on the Run 1 data can be achieved thanks to desegmentation. The $n = 1$ cut also eliminates any tracks that leave a hit on the first emulsion plate.

Upon visual examination of the combined tracks, it was observed that the quality of the merged tracks generally met the anticipated criteria in terms of the location and direction of the track segments. However, anomalies were detected that deviated from the expected physical behaviour of the event. Specifically, in certain instances, the merged tracks displayed segments with orientations in space that were incongruous to the overall track.

Figure 4.26: A collection of visualisations for Phase 1 (**Top**) and Phase 2 (**Bottom**) tracks projected on both the xz- and yz-planes. Each collection includes three categories that assess the quality of the combined tracks. It is important to note that these categories are subjective and do not determine whether the merged segments are accurate or incorrect.

It is expected that there will be low occurrences of fake seeds and incorrect track merging, since the precision of identifying fake seeds, even when employing neural networks, has never achieved 100% accuracy, not even in Monte Carlo simulations.

However, its impact on the generated data has been minimal, resulting in only a slight reduction (1-2%) in the purity of the hit content of the track. Unfortunately, it is challenging to accurately predict the real impact of these error rates in actual data conditions solely through visual inspection, due to a lack of comprehensive understanding of the correlation between error rates and track density. Therefore, studies using Monte Carlo data with comparable density are necessary.

4.4.8 Conclusion

Summary

To recap, this study has investigated ways to enhance the accuracy of the current standard method for tracking emulsion data in the SND@LHC experiment. The primary aim was to minimise instances where the path of the moving charged particle was reconstructed as several segments, leading to fragmented tracks. The main motivation behind this study was to:

1. Assist in removing the background tracks associated with external muons traversing the entire detector.
2. Reduce data load on vertex reconstruction, typically carried out after the tracking process.
3. Improve the overall quality of the reconstructed data.

The solution that was suggested and investigated involved identifying appropriate segment pairs using a series of geometric characteristics and combining them to form a unified volume track. The use of neural networks played a crucial role in this process, allowing for a more precise classification through their capacity to utilise the granular track data more effectively compared to conventional algorithmic methods that rely on a less detailed set of computed parameters.

The setup was evaluated using a separate sample of the generated data that was not used to train neural networks or tune the parameters of the global desegmentation algorithm. Data processing lasted around 12 hours in total, and analysis of the results has shown a notable decrease in muon background segmentation, estimated to be around 14-15%. A significant enhancement of up to 25% has been observed for hadrons, which are usually the particles most impacted by track segmentation.

The positive impact of segmentation on background elimination was investigated by evaluating the efficiency of track removal for both the initial track data and the desegmented version. The latter made it possible to remove up to $\approx 8\%$ more background tracks without significantly affecting the number of tracks emerging from the signal event.

After conducting simulation experiments, a small sample of scanned Run 1 emulsion data was analysed using a similar approach, although with slightly more stringent criteria for data selection. This adjustment was made because of the large amount of data resulting from a much higher track density in the real data sample compared to the simulation. Despite these modifications, the processing time was approximately 20 times longer than in the simulated data. The outcome was examined by analysing the disparity in track length distribution between pre/post-processed data and then comparing it with the simulation.

Similarly to the simulation, an overall trend has been observed in the distribution of track lengths shifting toward longer segments, although not as significant: there has only been an increase of 8% in the number of reconstructed tracks that traverse the entire ECC cell. The increase achieved is less than the anticipated increase of ≈ 22% seen in the simulation. This difference, partially attributable to the more stringent ad hoc data selection criteria, has led to a reduction in the removal efficiency of long tracks compared to the simulation to less than 4%.

Visual inspection has revealed a small set of rare, but noticeable anomalies where seemingly a set of unrelated track segments has been merged.

Interpretation

Overall, this study has demonstrated the capability of the neural network-assisted algorithm in enhancing the standard tracking data and has shown its value in removing background in simulated and actual scanned emulsion data.

Although the desegmentation process has shown promise, there are certain limitations in the current method that must be resolved before progressing further in this area. These include the insufficient simulation samples used, the suboptimal performance of ANNDEA when handling high data rates, and the potential need to enhance the algorithm by incorporating additional seed processing steps. This could help address possible accuracy challenges that might occur when working with larger data sets such as those in Run 1, as well as the even denser sets collected during Runs 2 and 3.

It is worth mentioning that current studies on the impact of desegmentation have focused exclusively on the tracking aspect, without considering its effects on the vertexing process and final results. The latter will be explored further in Section 4.6, where a more detailed assessment of how segmentation influences the vertexing procedure will be conducted.

Further research

The desegmentation module of ANNDEA is now in a stage where it can be applied in practical emulsion analysis. However, there is potential for additional improvements for future applications in the SND@LHC experiment.

A major constraint affecting this and other studies conducted on SND@LHC is the inadequate simulation of the emulsion data, which fails to closely replicate the conditions in the real ECC unit, leading to a performance gap. This is unsurprising since the algorithms that incorporate deep learning are expected to be particularly sensitive to the changes in their input data environment, which is the price to pay for their ability to model very complex features of the data.

Ideally, it would have been preferable to employ a realistic simulation from the outset; however, in practice, the process of generating Monte Carlo physics data is anything but straightforward. It demands a significant amount of time, expertise, and computational capabilities that are not easily accessible. Consequently, a less complex but more affordable simulated dataset was used to support the development of more sophisticated simulations.

In addition to establishing a strong foundation for the identification of the problem, the segmentation analysis has also influenced the specifications for future Monte Carlo-generated datasets that may apply to various ML-related studies, as outlined below in order of importance:

1. **Higher density:** a minimum average hit density achieved in the simulation should at least match that of Run 1 data, which corresponds to the Figure 4.9 and has $\approx 10^6$ hits cm^{-3}.
2. **Modelling of the ECC geometrical distortion:** the exact modelling of the data alignment imperfections is certainly unfeasible due to the complex nature of the distortion effects of the ECC unit geometry. Alternatively, a *smearing* technique may be employed on the output Monte Carlo data, which slightly adjusts the hit coordinate for each hit based on a Gaussian distribution, thus introducing uncertainty in its value.
3. **Increased variety of physics events:** this is an optional yet very valuable aspect of the simulation, encompassing additional signal and background occurrences, such as interactions involving electron and tau neutrinos, cosmic muons, and the radiation resulting from muon interactions with the tunnel walls. For studies involving dark matter detection, a simulation of the potential DM candidates can also be included. This does not have to be done in one go: since the particles within the ECC rarely interact, each group can be simulated separately and added to the existing simulated data set in phases.

The second limitation of the desegmentation process is primarily technical. The ANNDEA desegmentation algorithm faces computational challenges with the seed processing flow rates seen in the Run 1 data, consequently imposing data flow constraints that hinder its ability to merge tracks effectively. Various strategies can be employed to address this issue, including optimising the code and adjusting the seed preselection parameters. The first strategy is flexible and can be implemented at any point as long as it does not alter the algorithm's results. On the other hand, the second approach is best performed with the Monte Carlo simulation method mentioned earlier, allowing for fine-tuning and testing of the current geometric constraints using a more realistic and densely populated simulated dataset.

Another potential enhancement is motivated by the identification of anomalies in the track merging output while visually analysing the data. One of the fundamental limitations of the segmentation procedure developed in this study is the narrow scope of neural network application. The neural network is trained to differentiate between a pair of tracks coming from the same particle and a pair of randomly associated tracks, and it is sufficient, provided that the number of fakes ending up in the data is kept under control.

Nevertheless, as the density of the track increases, relying solely on the performance of the seed classifier may prove inadequate, potentially allowing a significant number of authentic-looking false seeds to proceed to the track construction phase. This may result in paradoxical scenarios where each seed seems appropriate on its own, but the combined track as a whole appears unphysical. This offers the opportunity to introduce an extra stage in the merging procedure. Before incorporating the seed into the track, a separately trained neural network could evaluate the overall merged track and discard the new seed candidate if it does not align well with the track.

However, this approach is challenging to put into practice, with the most demanding aspect being the generation of appropriate training data sets that include tracks in various merging phases. Moreover, the inclusion of an additional neural network will inevitably result in a decrease in the speed of the desegmentation process and, as such, should be avoided where possible. This is why rerunning the current desegmentation configuration on realistic simulation data should be done before, to avoid the risk of becoming unnecessarily invested in the additional time-consuming algorithm development.

4.5 Track Classification

4.5.1 Purpose

In this section, the problem concerning the measurement of the track momentum and the classification of particle types in the emulsion data, as previously discussed in Section 4.1, will be attempted. The proposed resolution involves inputting the reconstructed tracks into a neural network, which has been trained to identify the absolute momentum value and its type using a prepared Monte Carlo simulated data set. The process consists of the following stages:

1. Prepare the data that has been previously tracked.
2. Remove incoming particles and sufficiently long tracks that are likely to be left by passing muons.
3. Pass the remaining tracks through two different neural networks for particle identification and absolute momentum prediction.
4. Register the prediction value against each suitable track in the final data set.

Viewing this from a more physics-oriented angle, the primary idea of this method is to leverage the fact that distinct types of particles at different energies exhibit unique morphological and trajectory patterns, predominantly influenced by electromagnetic interactions of the charged particle with the nuclei of the ECC constituent atoms. For instance, muons usually possess higher energies and minimal relative energy losses, resulting in long, almost straight tracks. Conversely, electrons, being lighter, are more susceptible to synchrotron radiation losses and Coulomb scattering, ultimately producing short and highly nonlinear tracks.

The volume of data that can be potentially extracted and analysed is complex, involving a minimum of five distinct variables for each hit that constitutes the track. This makes it challenging to interpret using conventional analytical approaches, yet it is straightforward for neural network modelling, which is inherently apt for such problems.

The investigation conducted in this study will focus on the tracks originally reconstructed by FEDRA and that have undergone the ANNDEA desegmentation procedure described in Section 4.4. Nevertheless, this method can be applied to the output of any tracking algorithm, including the ANNDEA tracking module, which will be explored later in Section 4.7.

4.5.2 Data preparation

The main goal of preparing data is to generate a compact data set that only includes relevant information, thus minimising data storage needs during data analysis. This involves taking the initial input file and stripping any data apart from the following variables: x, y, z, tx, ty, *Brick ID* and the *Track ID*, which contains a unique identifier for the reconstructed track within a data set. Only hits used for the reconstruction that are part of sufficiently long tracks are included.

In case data are needed for training the neural network, extra variables *MC Track ID*, *PDG ID*, and P are brought in, which hold an identifier for the corresponding MC-generated track, its particle flavour and the value of the absolute 3-momentum in units of GeV. Since the correspondence between the reconstructed FEDRA track and the MC simulated track is not always exact, the usual track-matching procedure described in Section 4.3.4 is performed.

An additional step is necessary for the P variable as it is set at the individual hit level, reflecting the energy loss experienced by the particle as it travels through the emulsion. However, for regression analysis, a single variable for the whole track is required. In this research, the highest observed track momentum has been chosen, which corresponds to its initial value. The choice was driven by potential use cases that involve reconstructing invariant masses of vertices, which would require this particular definition of P. Nevertheless, future studies could opt for other aggregation methods like *mean* or *min*.

4.5.3 Data selection

Once the data are ready, additional data selection steps are carried out to minimise background, ensuring that it does not interfere with the classification procedure:

1. Eliminating all tracks that have registered a hit on the initial emulsion plate of the specified ECC unit. The purpose of this action is to eliminate all incoming muons and any other external charged particles that are not necessary for the analysis. This procedure also naturally eliminates all tracks that cross the entire length of the ECC unit.
2. Erasing all tracks that extend from $L - 1$ to $L - 2$, where L represents the total number of emulsion layers in a particular ECC cell. The primary aim of this step is to discard any external muons that did not record any hits on the initial or final emulsion layers, as a result of the limited basetrack reconstruction efficiency of these plates.

For an ECC brick made of 60 emulsion layers, this data selection measure would eliminate any tracks that span[12] 58 to 60 plates and any tracks with an initial hit in the first layer.

---

[12] This definition also includes plates with gaps in track or holes.

4.5.4 Neural network training

This specific research focuses on both classification and regression tasks. Despite the similarities in the data set and the neural networks employed, the substantial differences between these procedures justify dividing this section into two respective parts.

Particle type classification

At this point, a GNN neural network featuring Gaussian mixture model convolutional operators is used, having previously shown effective results in the track merging studies as discussed in Section 4.4.4. The optimised configuration of the network consists of four convolutional layers, with each layer containing 200 neurons.

The training set used was created by processing data from the ML domain using the data preparation and selection steps described above. Each data point used for training contains an integer label that can assume three values: 0,1, and 2, where each respectively represents the following particle type:

- **Muons:** This includes all the particles with the absolute value of *PDG ID* equal to 13.
- **Electrons:** This includes all particles with the absolute value of *PDG ID* equal to 11.
- **Other:** This category includes particles that possess any other *PDG ID* values. Given that tau leptons are not part of the simulation used in this study, this group essentially contains only hadrons.

After the data have been categorised, a resampling process is carried out in which all classes are reduced to match the number of data points in the class with the least representation.

Once the resampling is completed, a form of data representation that the neural network is capable of understanding and learning is chosen. For the purposes of this study, a graph object has been selected, possessing the following characteristics:

- Each node on the graph is represented by the hits that make up the track in the seed, with attributes encoding the spatial and angular coordinates.
- Every single node is interconnected with other nodes through non-directional edges.
- Each edge encodes numerous geometric properties, such as the relative distance between nodes.

This representation mirrors that used in Section 4.4.4. However, given that a single track is confined to just one data point, a seed boolean separator becomes unnecessary.

A GNN model has been trained with a starting learning rate of 10^{-4} with the validation loss and the accuracy profile plotted in Figure 4.28 below.

The neural networks were trained using the ANNDEA autopilot functionality, which continues to train them until it detects a plateau determined by the changing

(a) Node attributes

Node ID	x [μm]	y [μm]	z [μm]	tx	ty
N1	8,389	4,927	1,135	-0.00200	-0.00347
N2	8,387	4,926	2,450	0.00749	-0.00267
N3	8,388	4,915	3,765	0.00311	-0.01095

(b) Edge attributes

Edge ID	**Node 1**	**Node 2**	dz [μm]	dr [μm]	dl [μm]
E1	N1	N2	1,315	2.2	1315.0
E2	N1	N3	2,630	12.0	2630.0
E3	N2	N3	1,315	11.0	1315.0

Figure 4.27: Example of graph representation for a track fragment: **(a)** node attributes encoding spatial and angular information, **(b)** edge attributes defined by relative distance parameters. The illustrated track is particularly straight, rendering transverse distance parameters practically negligible. All distance values are expressed in microns, but before training, they are normalised to the range [0,1].

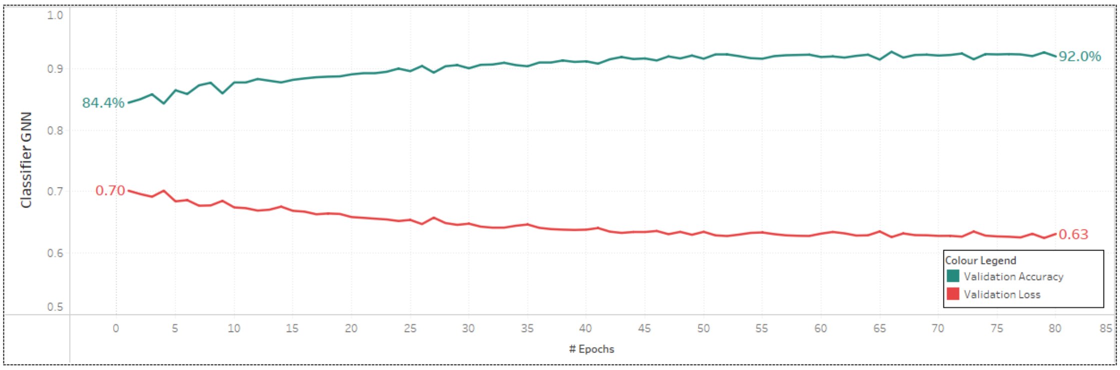


Figure 4.28: A plot of the normalised validation accuracy and CEL loss of GNN model training.

slope of the validation loss and accuracy. Subsequently, the best-performing state of the neural networks is saved and can be applied to the classification task.

Particle momentum measurement

In this specific task, the identical GNN neural network equipped with Gaussian mixture model convolutional operators has been deployed as before, with a slight alteration to its output layer: the number of output channels has been reduced to one, and the *softmax*[13] activation function has been deactivated, allowing the neural network to produce any real number as output.

---

[13] The Softmax activation function is frequently employed in classifier neural networks to standardise their output by transforming a real number vector into a probability distribution of potential outcomes.

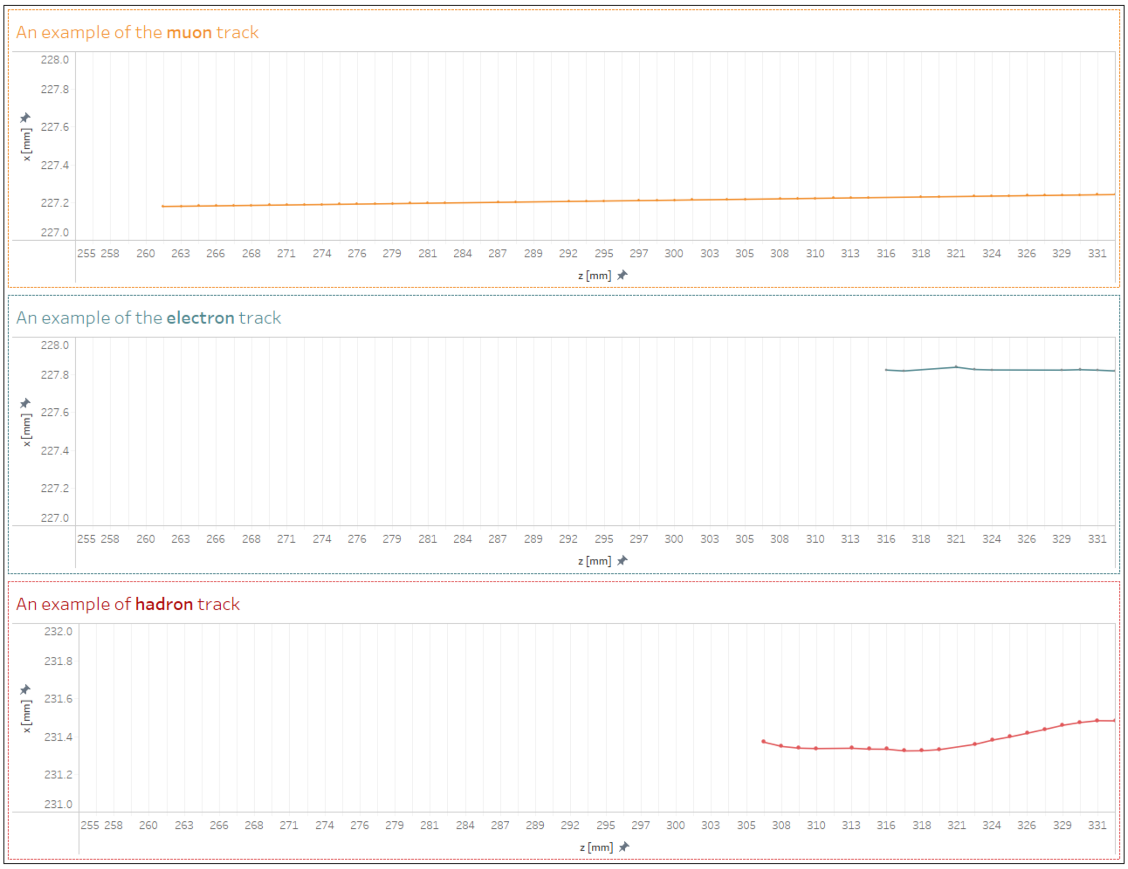


Figure 4.29: An illustration showing a visualised example of the track left by a muon (**Top**), electron (**Middle**) and hadron (**Bottom**). The scale has intentionally been altered for demonstration purposes and is not consistent.

The training set was created using data from the ML domain, applying the same preparation and selection methods that were outlined earlier. The data label values y_i are derived through the subsequent steps:

1. Extract the P_i value that has been computed in the data preparation process for every track i.
2. Limit any P_i values that surpass 100 GeV to this amount.
3. Deduct 50 GeV from the value and divide the result by 50 GeV.

For example, if track i has a maximum momentum of 56 GeV, it will be assigned a label value of $y_i = \frac{56-50}{50} = 0.12$. Conversely, tracks with momenta of 150 GeV and 1 TeV will both receive an identical value of 1.0.

Step 2 is justified by studies of residual versus momentum as shown in Figure 4.1, where the average residual value for the three groups of particles decreases with momentum and plateaus around 100 GeV. This implies that, from the neural network perspective, the tracks with momenta of 150 GeV and 1 TeV will appear almost identical, and it is not sensible to penalise the model for not being able to distinguish these two scenarios. Instead, it is more rational to concentrate on the 0-100 GeV range. Although this approach narrows the range of momentum measurement, from an analytical perspective, it is not an uncommon situation in experimental physics. Even the momentum resolution of the dedicated ATLAS tracking detector deteriorates[186] with increasing particle momentum due to the larger uncertainty of track curvature measurement. Therefore, expecting superior performance from the ECC detector, which was initially not designed to perform this kind of task, would be unreasonable.

The final stage is driven by strictly technical factors: within the realm of data science, this is a conventional method often referred to as *feature scaling*. This is a data preprocessing strategy that is used to normalise the values of features in a dataset, aligning them to a uniform scale. This procedure enhances data utilisation and model precision by reducing the impact of differing scales on machine-learning models.

After the data are labelled, they are preserved in their original form without any resampling because the tag values are on a continuous scale instead of being expressed as discrete categories. The data representation is the same as the one used for classification tasks, with the only distinction being the type of label value.

A GNN model has been trained with a starting learning rate of 10^{-4} with the RMSE and MSE loss plotted in Figure 4.30 below:

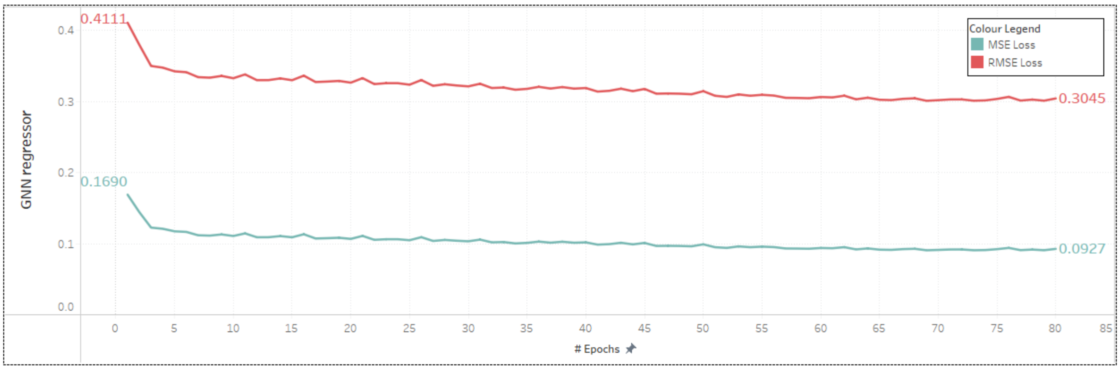


Figure 4.30: A plot of the RMSE and MSE loss of GNN model training.

The neural networks were trained using the ANNDEA autopilot functionality, and the best-performing state of the neural networks was saved and applied to the momentum prediction task.

4.5.5 Final output

After the training of the neural networks is complete, they can be directly applied to examine the tracking data and generate predictions, the form of which is determined by the type of model. The classifier provides a vector of values for $\mathbf{p}_i = \left[\mu, e, h\right]$, where μ, e, and h represent the probability values of the track being muon, electron, and hadron, respectively, and the sum of these values is normalised to one. The regression model outputs a real value $r_i \in [-1, 1]$, which is then converted back to GeV using formula $p_i = (r_i * 50\ \text{GeV}) + 50\ \text{GeV}$.

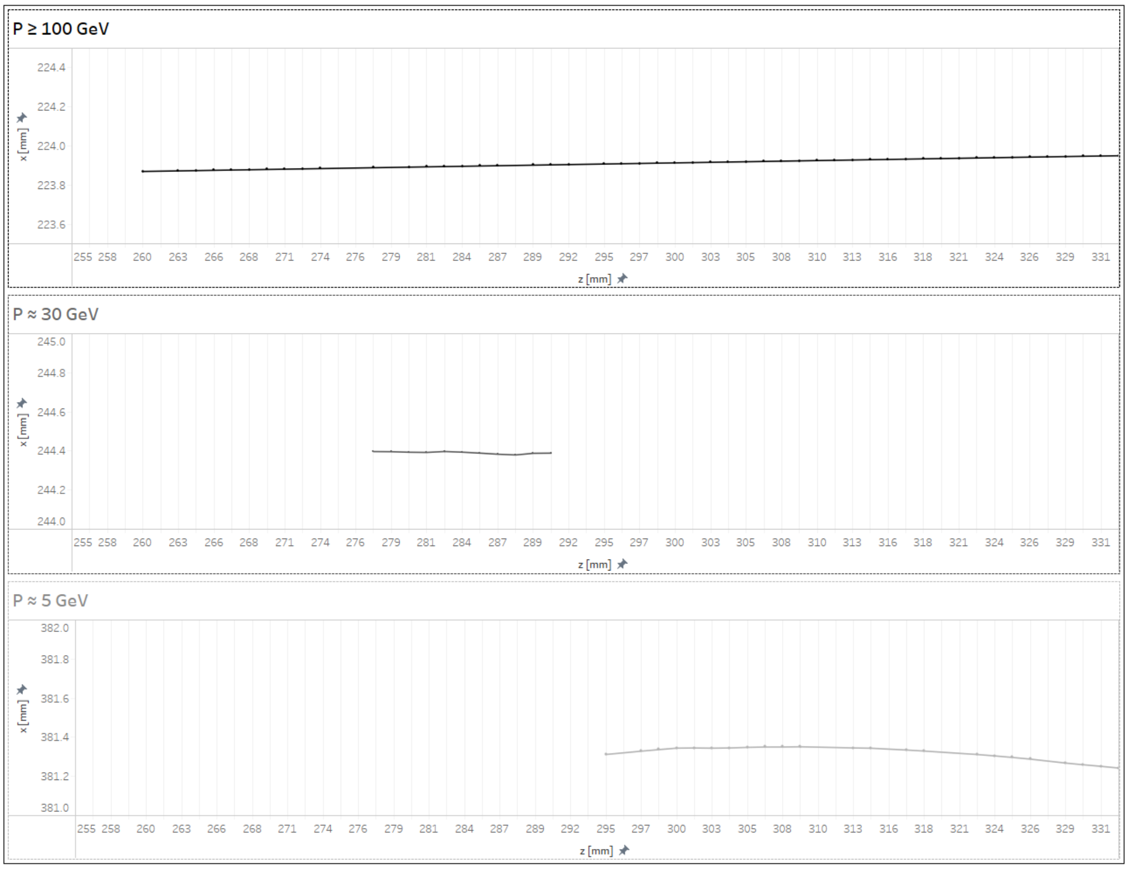


Figure 4.31: An illustration showing a visualised example of the track left by a muon (**Top**), electron (**Middle**) and hadron (**Bottom**). The scale has intentionally been altered for demonstration purposes and is not consistent.

4.5.6 Performance on the Monte Carlo simulation

Classification results

The classification process was carried out using ANNDEA, taking approximately 1-2 hours to complete. The data were collected and the predictions were compared to the matched simulated tracks and their actual truth labels and presented below.

Table 4.7: The classification confusion matrices represent three different views of the same data.

	Truth		
Prediction	Muons	Electrons	Hadrons
Muons	10.4% 2,699 [96%]	3.2% 834 [62%]	0.1% 30 [98%]
Electrons	1.9% 481 [59%]	78% 20,284 [62%]	0.2% 43 [99%]
Hadrons	0.1% 26 [99%]	2.2% 562 [73%]	4.1% 1,061 [99%]

	Prediction		
Truth	Muons	Electrons	Hadrons
Muons	84.2% 2,699 [96%]	15.0% 481 [59%]	0.8% 26 [99%]
Electrons	3.9% 834 [62%]	93.6% 20,284 [62%]	2.6% 562 [73%]
Hadrons	2.7% 30 [98%]	3.8% 43 [99%]	93.6% 1,061 [99%]

	Truth		
Prediction	Muons	Electrons	Hadrons
Muons	75.8% 2,699 [96%]	23.4% 834 [62%]	0.8% 30 [98%]
Electrons	2.3% 481 [59%]	97.5% 20,284 [62%]	0.2% 43 [99%]
Hadrons	1.6% 26 [99%]	34.1% 562 [73%]	64.3% 1,061 [99%]

All of the tables above depict the same data, but each is compiled in a unique manner to address different queries. The left table is a standard confusion matrix that includes the percentage distribution of the tracks across the table, their absolute count, and the mean value of THRP.

The middle table incorporates the track percentage distribution within each row, their absolute count, and the mean value of THRP. It seeks to interpret the results from the simulation perspective: *Given the actual distribution of the particle types in the data, what proportion of them were accurately classified?*.

The table on the right is the transposed form of the middle one, aiming to answer the question: *Given the distribution of the predicted track labels, what proportion of them were correctly classified?*.

Momentum measurement results

The regression analysis took approximately 1 to 2 hours. Data were collected, juxtaposing the predicted values p_i with the matched simulated tracks and their true values P_i, which are visualised below:

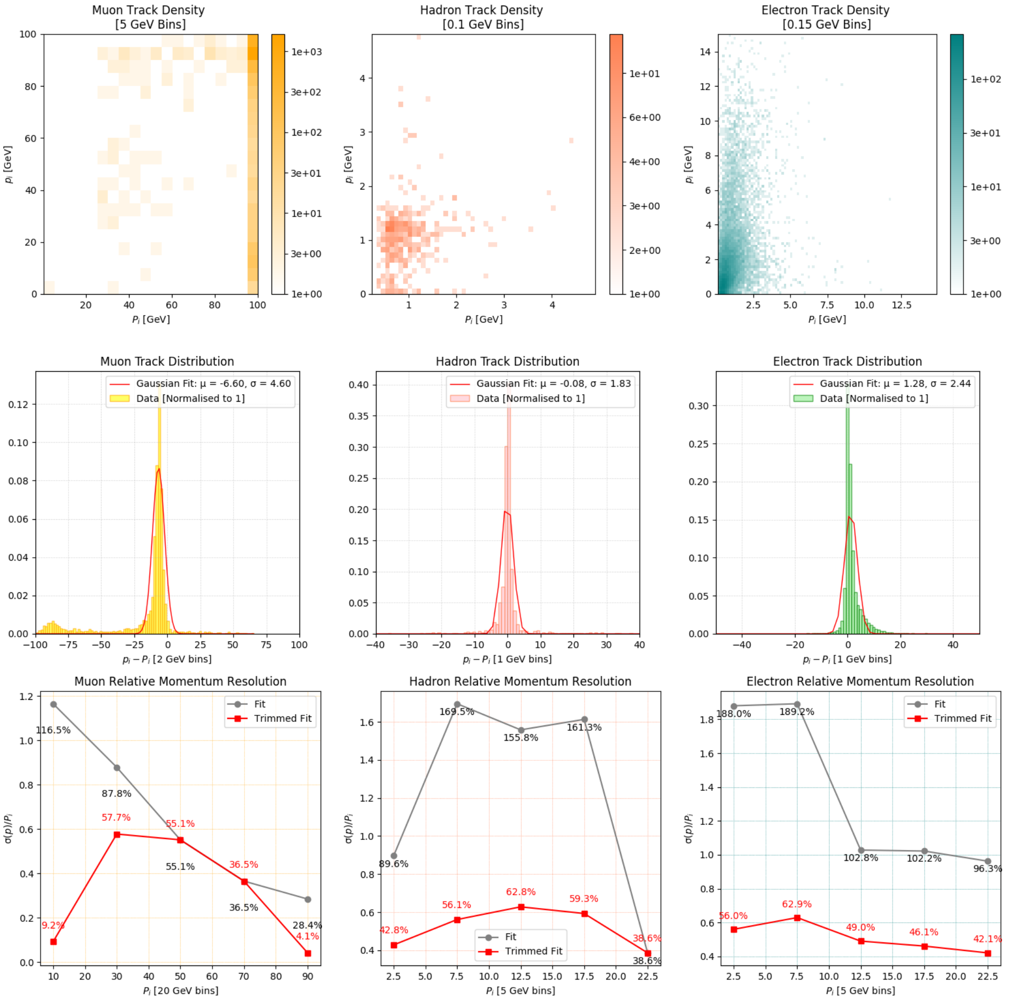


Figure 4.32: A set of three momentum measurement plots split by the matched truth particle type.

The various plot groups aim to assess distinct facets of momentum prediction proficiency: The upper group showcases a heatmap of track density, with the predicted momentum of particles plotted against the vertical axis and the actual momentum on the horizontal axis. This visual representation serves to highlight how the neural network prediction correlates with the simulated particle momentum distribution within the dataset.

The central visualisation set illustrates the distribution of the variances between the forecasted momentum values p_i and their true simulated counterparts P_i. Each distribution is modelled using a Gaussian function, concentrating on fitting the central peak. The σ values represent the absolute momentum resolution, while the μ values help evaluate whether the predictor tends to underestimate or overestimate the true momentum value.

The plots on the bottom illustrate the relative resolution $\frac{\sigma(p)}{P}$, where $\sigma(p)$ denotes the local momentum resolution, obtained by fitting a Gaussian for each true momentum bin P. The purpose of this plot is to show how the measured resolution changes with true momentum P.

Discussion

By looking at and analysing each of the three versions of the confusion matrices, the following observations can be made:

- The overwhelming majority, surpassing 80% of the tracks in the dataset, are attributed to electrons, and the total number of tracks constitutes merely around 35% of the quantity prior to data selection. This suggests a high efficiency in removing background noise through the application of data selection cuts. Although this is consistent with the results from 4.4, it still serves as a reliable verification.
- The total classification precision in the control set has reached 92.4%, surpassing the score reached during the model training phase, a scenario not commonly seen in machine learning, as test data sets usually exhibit reduced performance. The interpretation of this as a positive or negative aspect hinges on the precise cause: on one hand, it could suggest that the training set is sufficiently diverse in terms of track types, allowing the model to effectively generalise predictions across the entire simulated ECC detector. On the other hand, it may indicate possible limitations of the existing simulation set, which might have a restricted variety of events, thereby making it easier to achieve high-precision scores within the identical simulation dataset but not perform well on unrelated data sets.
- ANNDEA correctly identified over 80% of the actual muons and over 90% of the real electrons and hadrons. The most significant errors occur when ANNDEA incorrectly labels true muons as electrons. These errors predominantly affect muons that undergo additional interactions, such as producing delta-rays or small electromagnetic showers. In such cases, the reconstruction algorithm can mistakenly attach electron fragments to the muon track, making it appear electron-like to the classifier.

- Looking at it from a prediction perspective, the following scenario unfolds: approximately 98% of all electrons that ANNDEA identifies align with their MC simulation truth label. For all muons that GNN identifies, there is a roughly 23% probability that the real particle is an electron. The case is slightly worse for particles identified as hadrons, where only about 64% of them are genuinely hadrons, according to the simulated truth.

A common theme can be spotted in these observations: all misclassified hadrons and muons exhibit the greatest overlap with electrons and consistently demonstrate low track-hit purity according to the THRP criteria. This brings to light a pivotal concern of this study: the susceptibility of neural networks to the quality of the underlying data. For both training and validation phases, reconstructed FEDRA tracks, enhanced by the ANNDEA desegmentation process, are used instead of pure MC tracks. This approach is justified because these are the tracks that ANNDEA will encounter in real-world data, although these tracks are not perfect: some comprise fewer hits than the actual particle left behind, while others amalgamate hits from tracks left by different particles. Consequently, the truth particle label assigned during training may only represent the majority of hits in the track, but not necessarily its entirety. For instance, there could be scenarios where the reconstructed tracks incorporate multiple segments from different particle types, thereby ’confusing’ the classifier model and potentially resulting in incorrect predictions.

The outcomes derived from the regression analysis reveal a more controversial scenario: while the reconstructed momentum distribution approximately aligns with the true momentum spectrum, its overall performance remains suboptimal, likely due to the presence of outliers that distort the otherwise symmetric Gaussian shape. The finest resolution is achieved for high-energy muons where $P > 60$ GeV, with relative resolution values ranging from 36% down to 28%. This might be considered poor, even when compared to the relatively modest accuracy benchmarks of energy measurements in electromagnetic calorimeters, and certainly below the precise kinematic measurements achievable with standard tracking devices (typically $\lesssim$ 3-4 % for muons in the same momentum range[186]) However, the shape of the $p_i - P_i$ distribution suggests that significant degradation stems from outliers, likely caused by track segment mixing during the reconstruction process. For high-quality tracks, the resolution improves slightly, reaching 4.1%. However, this value should be interpreted not as a precise momentum measurement, but rather as an indicator of the probability that the particles true momentum exceeds 100 GeV, which the regressor does not directly determine. For hadrons and electrons, the situation is even more challenging: the relative resolution, even for perfectly reconstructed tracks, remains comparable to the particles actual momentum, making precise kinematic measurements unfeasible.

The means μ in the momentum delta distribution $p_i - P_i$ indicate that the regressor has a tendency to overestimate the momentum for electrons and underestimate it for muons, while the hadron group distribution is more symmetric.

4.5.7 Performance on Run 1 data

The performance of ANNDEA was assessed using the Run 1 data sample by implementing both the particle identification and momentum measurement procedures.

These procedures required approximately 3-4 hours, a duration longer than that for the simulated control domain, but not significantly different, as noted for the desegmentation procedure. This can be attributed to two primary factors: the data selection process, which minimises the number of tracks to initially work with, and the absence of seed generation in the analysis phase, implying that the data rate will increase linearly, not quadratically, with increasing density.

Results

The data containing the prediction of the particle type and its momentum measurement are plotted below. Given that the Run 1 data does not contain any Monte Carlo simulation truth and the precise distribution of the particle types in the Run 1 data sample remains unknown, it is not feasible to validate the results with absolute certainty. Nevertheless, by utilising the measurable parameters that are available, a series of plots illustrating the identified particle distribution across the measured momentum-track length phase space has been created, consisting of three plot groups:

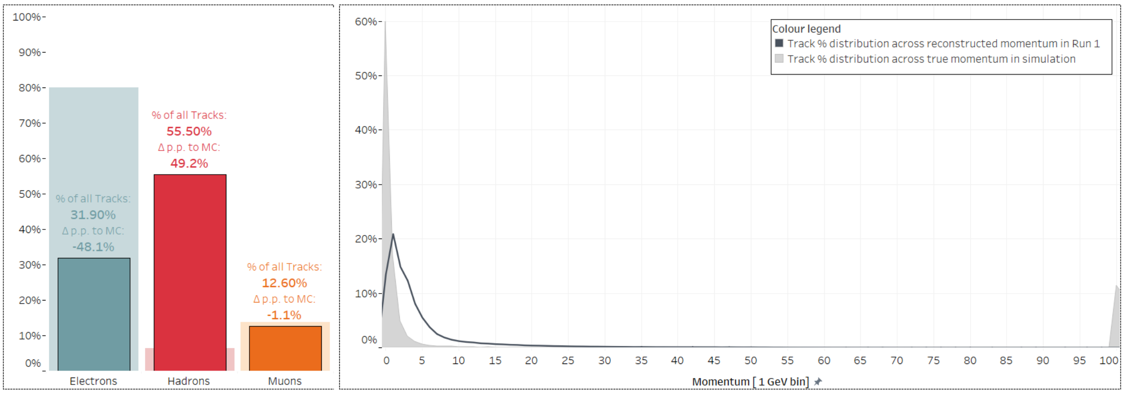
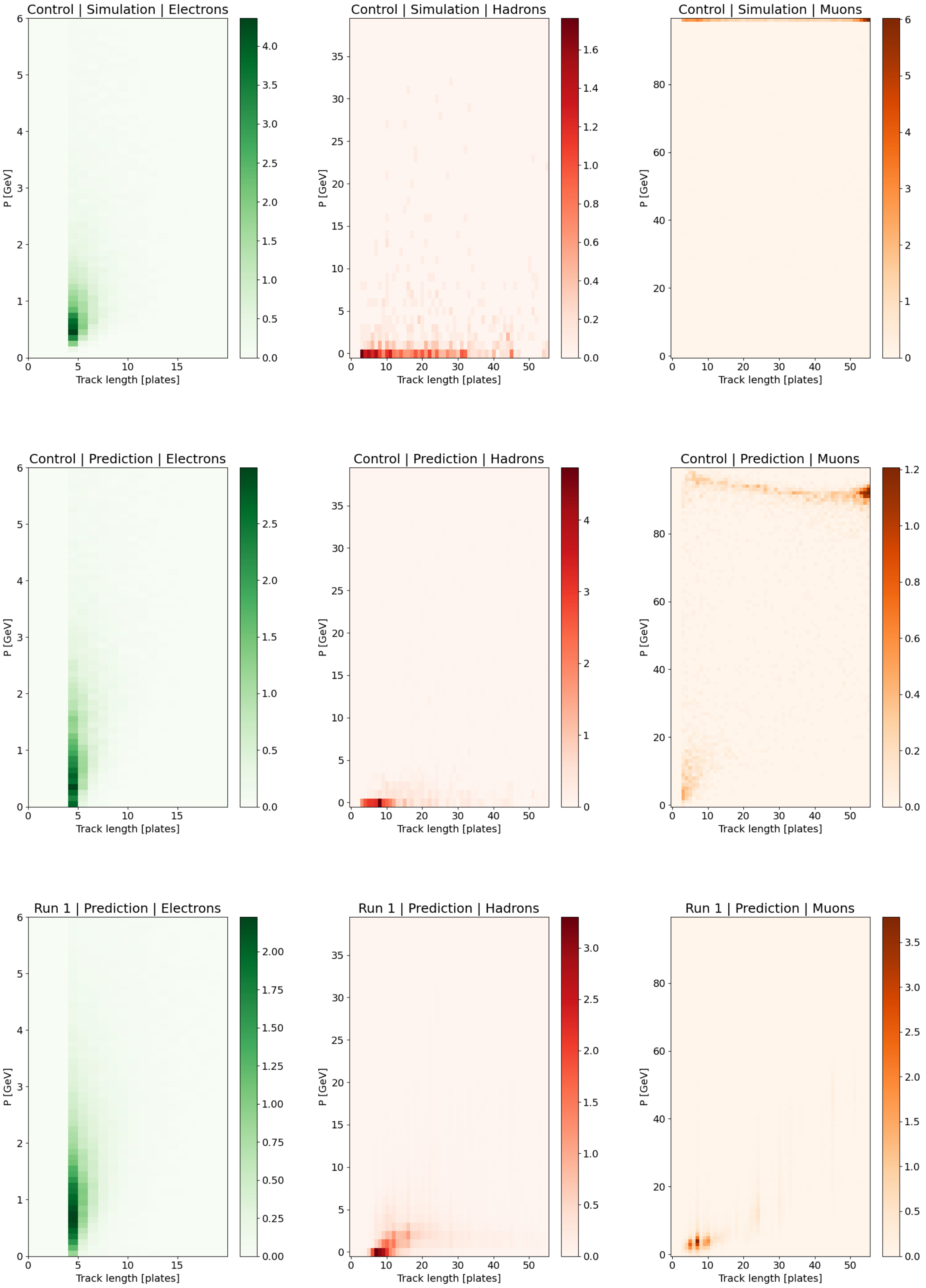


Figure 4.33: **Left:** The simulation's anticipated particle track flavour is represented by a percentage distribution (depicted by lighter coloured bars) and is compared to the predictions (shown by saturated bars) for the Run 1 data sample. **Right:** The expected particle's momentum, as observed in the simulation, is displayed as a percentage distribution (represented by a grey-filled area) and is contrasted with the prediction (indicated by a black line) for the Run 1 data sample.

1. Three diagrams from the control domain, distinguished by the corresponding MC simulated particle type and utilising the genuine simulated momentum on the vertical axis to portray the distribution projected by the simulation.
2. Three diagrams from the control domain, distinguished by the predicted particle type, utilising the measured momentum on the vertical axis to portray the distribution generated by ANNDEA on the simulated data set.
3. Three diagrams from the Run 1 data sample, distinguished by the predicted particle type and utilising the observed momentum on the vertical axis to portray the track distribution generated by ANNDEA on the real data.

These diagrams allow for the verification of the predicted kinematic and particle flavour distribution results from ANNDEA, and their comparison with the simulation predictions.



Control | Simulation | Electrons

Control | Simulation | Hadrons

Control | Simulation | Muons

Control | Prediction | Electrons

Control | Prediction | Hadrons

Control | Prediction | Muons

Run 1 | Prediction | Electrons

Run 1 | Prediction | Hadrons

Run 1 | Prediction | Muons

Figure 4.34: A set of 9 plots, each representing different particle percentage distributions across track length and momentum phase space, depending on the data set, truth/prediction and particle flavour group.

Discussion

Examining the projected particle flavour distribution in Figure 4.33 reveals a significantly higher percentage of hadrons in the Run 1 data than what is seen in the simulation. This could suggest that there are substantially more hadron backgrounds and QCD processes in the real data, or there is an increased frequency of misidentification. The momentum distribution peak at energies under 1 GeV seen in simulation is moved towards 2 GeV and considerably broadened. The rest of the reconstructed spectrum is sparsely populated, including the ≥ 100 GeV range, which is expected to be occupied by IP1 muons. This offers the first distinct sign of a significant issue with the neural network's ability to handle tracks in the real data.

A deeper understanding of ANNDEA performance can be achieved by examining the graphs in Figure 4.34. Here, the extra track length acts as an additional variable, aiding in assessing the correspondence between the anticipated and observed physics. The analysis for each particle is encapsulated below:

- **Electrons** are anticipated to be of short length ($L \lesssim 8$ plates) and possess low absolute momentum values ($P \lesssim 5$ GeV). The ANNDEA simulation manages to more or less mimic its length and momentum distribution; however, the latter is broadened, as it has been noted that the GNN regressor has a slight inclination to overestimate the momentum of electrons. In the actual data, the distribution of electrons appears to be somewhat similar to what is depicted in the two preceding charts, albeit with a minor shift towards a greater track length and a slightly more dispersed momentum distribution.
- **Hadrons**, predominantly, are anticipated to exhibit low absolute momentum values ($P \lesssim 5$ GeV); however, their length distribution is considerably dispersed throughout the range. The simulation performance mirrors the expected scenario to some extent, but there are minor discrepancies in the momentum distribution, where the prediction fails to correctly measure the high-momentum hadron tracks that are anticipated according to the simulation. In the actual data, the predicted hadron track length distribution appears to align with the control set, albeit with a shift towards shorter lengths. However, the momentum distribution seems to be skewed towards larger values.
- **Muons**, predominantly, are predicted to possess high absolute momentum values ($P \geq 100$ GeV) and their length distribution is expected to peak at longer lengths. ANNDEA's performance on the simulation attempts to reproduce this, but there are two distinct sets of discrepancies: a shift in the momentum spectrum towards lower values and a notable presence of tracks in the lower-left quadrant, a region usually populated by electrons. The first discrepancy can be attributed to the tendency of the GNN regressor to underestimate the momentum of muons. The second discrepancy arises from the overlap of electron and muon hits due to errors in the tracking procedure, misleading the GNN classifier. A likely reason for that is the presence of extra noise in the data and the much higher occupancy.

  In the Run 1 data, the classification and regression results for muons stand in stark contrast to the control set: the anticipated and observed distributions of track length and momentum show no resemblance. Instead, muon tracks cluster in the short-track, low-momentum quadrant, which is the opposite of theoretical expectations. This highlights the significant challenges faced by the GNN in both particle-type classification and momentum reconstruction on real data. A likely cause is the emulsion misalignment, which reduces the apparent smoothness of muon tracks compared to their idealised simulated counterparts, leading the network to misclassify them as low-momentum particles undergoing Coulomb scattering.

4.5.8 Conclusion

Summary

In summary, this study has investigated the potential of using deep learning to infer reconstructed track properties from their inherent morphological characteristics. The primary motivation for this work is the lack of specific technological abilities of the SND@LHC emulsion detector to provide details on the type of track particle or its momentum that could potentially be used for event analysis tasks. These tasks include further background identification, such as electromagnetic showers and more efficient signal event selection, including reconstruction of their properties.

The solution that was suggested and investigated involved the use of two trained geometrical neural networks to perform the following operations by using all the track available information, such as constituent hit coordinates and angles:

1. Split tracks into three categories: electrons, muons and hadrons.
2. Provide an estimate of a track's absolute momentum on a scale from 0 to 100 GeV, with the latter value being applicable to all higher-energy particles as well.

The full data processing took approximately four hours, and the subsequent analysis revealed a two-fold outcome. Classification performance reached 92% accuracy, with cases of misidentification primarily associated with poorly reconstructed tracks. In contrast, the regression results for momentum prediction were much more modest. Even in the absence of outliers, the absolute momentum resolution depended strongly on particle type: for electrons and hadrons, the relative resolution was roughly comparable to the particles actual momentum. The regressor performed somewhat better for particles with momentum above 100 GeV; however, since the momentum is effectively capped at 100 GeV, this should not be interpreted as a precise kinematic measurement.

After conducting simulation experiments, a small sample of scanned Run 1 emulsion data was analysed using a similar approach. The outcome was examined by analysing the distribution of tracks across measured quantities such as the reconstructed momentum spectrum, calculated track length, and predicted particle type and comparing it with the picture observed in the simulation. Although for the identified electrons and hadrons, the observation was more or less consistent with expectation, a major discrepancy in the behaviour of identified muons and their MC-generated counterparts has been observed, indicating the need to train the network on a more realistic simulation sample.

Interpretation

The issues addressed in this research are perfect examples of typical problems tackled by deep learning. Unfortunately, they have showcased one of the most common challenges that plague data science projects: data quality and consequently model overfitting.

This becomes particularly clear in the context of the particle type classification task. Here, the neural network demonstrated commendable performance in track identification within a simulated environment but faced difficulties under actual data conditions. This discrepancy in performance substantially limits the applicability of neural networks to the emulsion data analysis: it can lead to misinterpretations of the reconstructed physics in the detector, thus invalidating any potential discoveries.

Therefore, the overall outcome of this experiment is inconclusive: accurate particle identification using a neural network is possible under simulated conditions, but there is no solid proof that it is practical for the actual emulsion data, the data quality of which is significantly lower due to various factors that are not taken into account in simulation.

Momentum measurements present a notably greater challenge, as their precision, even in simulated environments, remains limited for detailed kinematic analysis. Nevertheless, these estimates can still provide useful guidance for broader event characterisation. When applied to real data, performance naturally declines, even for particles such as muons, whose signatures are generally easier to identify due to their high energy. However, the momentum resolution is closely tied to the quality of track reconstruction, and as the latter improves, through better tracking algorithms and detector alignment, a corresponding improvement in momentum resolution is expected. While the current implementation may not yet meet the precision requirements for detailed momentum studies within the SND@LHC experiment, the observed limitations also point to clear directions for refinement. Moreover, this approach may still hold practical value in other experimental setups that lack a dedicated momentum spectrometer, offering a lightweight and adaptable alternative for momentum estimation and becoming a valuable component of the analysis strategy.

Further research

The strategy for further research depends on the type of problem and will be listed separately in the following.

Track classification: Despite the encouraging performance demonstrated on the simulation set, it is crucial to bridge the ANNDEA performance gap between the simulation and the real data to ensure its effectiveness under actual experimental conditions.

Ideally, the first step in the right direction would be to understand what drives this difference so that the causes can be addressed accordingly. This can be achieved by selecting a key performance indicator such as RMSE and converting it to L'_{RMSE} with the following formula:

$$L'_{RMSE}(p_i, y_i) = 1 - L_{RMSE}(p_i, y_i), \tag{4.6}$$

thus, the best classification performance is attained when $L'_{RMSE} = 1$, whereas the poorest performance is reached at $L'_{RMSE} = 0$. This new metric can be decoupled into multiple terms such as the one below:

$$L'_{RMSE} = l_b \; \epsilon_{ml} \; \epsilon_{em} \; \epsilon_{tr}(em, \rho_h). \tag{4.7}$$

The initial term l_b represents the highest possible mean value of L' that can theoretically be reached in the simulated data, assuming perfect tracking accuracy and ideal emulsion alignment.

The second term ϵ_{ml} is the coefficient that encodes the loss degradation as a result of the specific selection of the data representation and the choice of the ML model. If the optimal configuration is selected, it will use the data effectively, and its degradation factor will tend towards 1, maintaining the initial value of l_b. In contrast, any suboptimal selection will lead to $\epsilon_{ml} < 1$, imposing a penalty on the final loss L'_{RMSE}.

The third term ϵ_{em} represents the degradation coefficient due to the noise introduced by the artefacts of the emulsion alignment and as a result of its geometrical distortion effects.

The last term $\epsilon_{tr}(em, p_h)$ describes the penalty factor due to the tracking quality, which is a function of the emulsion alignment and the hit density.

This equation is not to be considered a basic principle of nature; instead, it serves as a useful mathematical approximation. By measuring each term, it allows for the quantification of each effect's contribution and assists in identifying the appropriate solution.

Upon defining the equation, the selection of the simulation data sample becomes necessary. It is recommended to use a simulation that has an average hit density of at least $\approx 10^6$ hits cm^{-3}, which is comparable to the conditions observed in the Run 1 emulsion data. The terms l_b, ϵ_{ml}, and ϵ_{em} in Equation 4.7 are not impacted by the hit density, given the low probability of particles in the emulsion detector interacting with each other. However, its final term $\epsilon_{tr}(em, p_h)$ is affected by it. Thus, for consistency, it is advisable to keep the underlying conditions the same.

Once the simulation set is chosen, the coefficient product $l_b \times \epsilon_{ml}$ can be evaluated by repeating all the steps described in this study, such as model training and classification loss evaluation, using the original MC-generated tracks provided by the simulation truth.

Estimation of the term ϵ_{ml} can be achieved by replicating the preceding step using the same simulation, with an additional step involving smearing its hit coordinates. This must be done to model the degradation in data quality caused by the emulsion alignment effects observed in the actual dataset. The quantity ϵ_{ml} is then determined by dividing the newly assessed L'_1 by the product of the coefficient l_b and ϵ_{ml}, which was obtained in the previous step.

To calculate the last term, another iteration has to be performed on the simulated data sample that has been subject to the tracking procedure using the reconstructed tracks instead. Its value is simply the new evaluated L'_2 divided by the coefficient product $l_b \times \epsilon_{ml} \times \epsilon_{em}$ derived in the previous step.

After the computation of all terms in Equation 4.7, the follow-up action will depend on the final value of L'. The GNN, which has been trained on the tracks that were reconstructed in the dense hit environment, might exhibit adequate performance. Then it can be assessed on real data, employing the same consistency checks that were used earlier in this investigation.

If this is not the case, then the primary factor contributing to the most influential coefficient should be tackled first, provided that it is achievable. For example, if term $\epsilon_{tr}(em, p_h)$ exerts the most substantial effect on classification performance, then improvements should be made to the tracking process where feasible. If this is not possible and the classification performance does not meet the requirements for the physics analysis, then at least it provides sufficient grounds to abandon any further efforts in this area and focus on other problems.

Track momentum measurement: This investigation did reveal a potential for this method, and if this avenue is to be pursued, the identical principles used for the classification problem can be used to pinpoint the precise problem that affects its performance. One of the limitations of the regression method used in this study is the lack of proper research on an optimal regression method due to the limited scope and time. The regression model used has been an adaptation of the existing classification model, and therefore, its real value prediction capabilities might be suboptimal. In theory, this can be achieved by singling out the term ϵ_{ml}. However, in practice, it poses a significant challenge because of the necessity to experiment with many types of neural networks and their architectures.

Expanding the scope of the research: So far, this study has been concerned with tracks only; however, an analysis of more complex objects, such as vertices and electromagnetic showers, can also be explored using similar principles described here. The ANNDEA scalable architecture enables one to incorporate these capabilities with relative ease, and neural network analysis involving more complex structures may prove to be more resilient to jumps in the underlying data quality. An obvious use case would be to identify signal vertices, which is one of the main goals of the SND@LHC experiment.

4.6 Vertex reconstruction

4.6.1 Purpose

This section seeks to investigate the potential of deep learning in vertex reconstruction tasks. The reconstruction pipeline closely resembles the one described in Section 4.4, so the methodology section will emphasise the differences to avoid redundancy.

There are four main steps involved in vertexing:

1. Clean and prepare the emulsion data, which have been previously tracked by FEDRA.
2. Form all possible track combinations and refine them using basic geometrical considerations.
3. Pass the remaining seeds through the neural network filters.

4. Combine all seeds remaining from the previous steps into vertex objects.

The investigation conducted in this study will focus on the FEDRA reconstructed tracks, but in principle, it can be applied to the output of any tracking algorithm.

4.6.2 Data preparation and selection

The data preparation mirrors that of the track desegmentation process, with a minor variation: if data are required to train the neural network, not only is the *MC Track ID* included, but also the *MC Mother ID*, which serves as an identifier for the parent particle the decay of which has produced the track in question.

Data selection, on the other hand, replicates the steps used in Section 4.5. Initially, all tracks that have left a hit at the initial emulsion plate are removed, aiming to eliminate irrelevant incoming muons and other external charged particles. Then tracks that span $L - 3$ plates of the detector are excluded, where L is the total number of emulsion layers of the given ECC unit.

4.6.3 Seed initialisation and preliminary selection

Vertex seed refers to a pair of track segments that may originate from the same decay. The initial number of combinations increases according to equation $N_{sd} = N_{tr}^2$, which can be large for a standard ECC unit. Hence, initially, only a basic seed representation in the shape of a vector is employed:

$$\mathbf{sd}_{ij} = \begin{bmatrix} tr_i & x_i & y_i & z_i & tr_j & x_j & y_j & z_j & label \end{bmatrix} \tag{4.8}$$

where tr_i is a track identifier of the track i and the tr_j is a track identifier of the track j. The elements x, y, and z represent the coordinates of the first hit along the z-axis for the tracks i and j, respectively.

The *label* element is not mandatory and is only included in the data sets intended for neural network training and is ignored in the seed refinement phase. It is a Boolean variable that can take a value of 1 or 0, indicating whether the constituent tracks share the same *MC Mother ID* or not.

The initial selection steps for vertex seeds are akin to those used for desegmentation: self-permutations and duplications are eliminated. However, rather than applying SLG and STG cuts, a simpler selection is made by discarding seeds where the distance parameter R exceeds 4000 microns. The parameter R represents the Euclidean distance between the initial track hits in the seed and is calculated as $R = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$.

4.6.4 Seed fiducial cuts

The fiducial cuts employ a more advanced method compared to the desegmentation equivalent. In addition to the DOCA, which has a vertexing threshold set at 200 microns, an additional pair of cuts is applied based on the reconstructed vertex seed origin (VO). The first cut discards any seed if its VO coordinates fall outside the fiducial volume of the ECC detector.

The second cut takes into account the minimum distance between the origin of the vertex and the closest starting hit of any track in the seed (VO_{T1} or VO_{T2} as depicted in 4.14). If this distance exceeds 4000 microns, the whole seed is rejected. It is important to mention that numerous selection criteria resemble those employed by FEDRA and are primarily motivated by the need to reduce the computational load.

4.6.5 Neural network training

In the vertex reconstruction research, two neural network models were employed to enhance the seeds generated in the previous stage: the GNN neural network, which had been used in two earlier studies, and the CNN, which, for this particular task, has demonstrated promising performance during training. The CNN model architecture had the following configuration, presented in the table below.

Layer	**Activation function**	**Number of neurons/kernels**	**Size**
Convolution 1	ReLU	64	3 ×3 ×3
Pooling 1	N/A	N/A	2 ×2 ×6
Convolution 2	ReLU	64	3 ×3 ×3
Pooling 2	N/A	N/A	1 ×1 ×3
Convolution 3	ReLU	80	5 ×5 ×5
Pooling 3	N/A	N/A	2 ×2 ×6
MLP 1	ReLU	80	N/A
MLP 2	ReLU	64	N/A
MLP 3	ReLU	48	N/A
Output	Softmax	N/A	2 channels

To avoid overtraining, a dropout regularisation technique has been implemented for dense and convolutional layers, whose value was set at 10%.

Before feeding the seeds into the neural network, they are transformed into a three-dimensional monochrome image represented by a $70 \times 20 \times 200$ numpy array. Each pixel in this array has a value of either 1 or 0, indicating whether the corresponding region of space is traversed by the track.

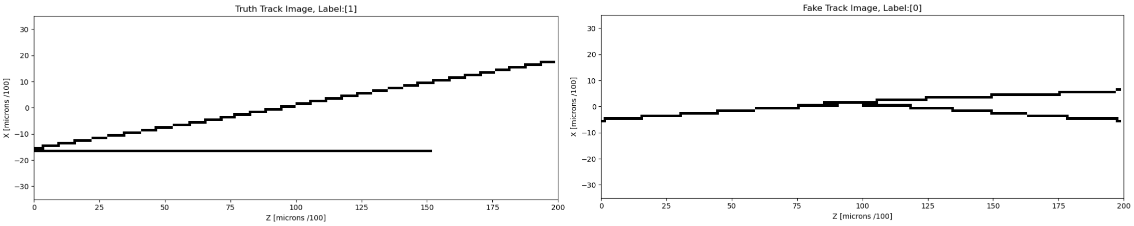


Figure 4.35: An example of genuine (**Left**) and fake (**Right**) seed images used for CNN model training projected on the xz plane .

To minimise the memory usage of three-dimensional numpy arrays while avoiding substantial data loss, the vertex track hit coordinates are subjected to rotations around the z and x-axes and scaling transformations to ensure that the seed features fit within the narrow image dimensions.

By identifying the simulated mother particle ID of the majority of the hits belonging to the reconstructed track, it is possible to establish whether these track segments should form a vertex and use it as a class label. To train both the GNN and CNN models using original and defragmented FEDRA tracks, the following sets of training and validation samples have been produced:

- **GNN set:** A sample of the ≈ 29k train and an additional ≈ 2.9k validation seeds represented as graphs using original FEDRA tracks.
- **GNN* set:** A sample of the ≈ 36.5k train and an additional ≈ 3.6k validation seeds represented as graphs using FEDRA tracks that have undergone the desegmentation process covered in 4.4.
- **CNN set:** A sample of the ≈ 29k train and an additional ≈ 2.9k validation seed images using original FEDRA tracks.
- **CNN* set:** A sample of the ≈ 36.5k train and an additional ≈ 3.6k validation seed images using FEDRA tracks that have undergone the desegmentation process covered in Section 4.4.

Both GNN and CNN models have been trained with a starting learning rate of 10^{-4} with the validation loss and the accuracy profile plotted in Figure 4.36 below:

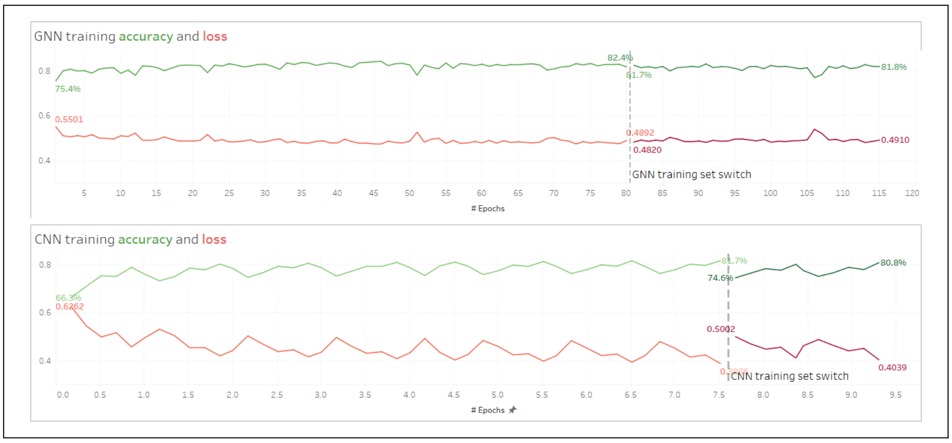


Figure 4.36: A plot of the normalised validation accuracy and CEL loss of the GNN (**Top**) and CNN (**Bottom**) models training on both original FEDRA tracks (paler lines) and their de-segmented versions (More saturated lines).

Since the training process, especially for CNN models, can take a considerable amount of time of up to a day for a single epoch, several time- and hardware resource-saving measures have been taken, such as requesting HTCondor nodes with GPU facilities and using the *data transfer*. The latter is a commonly used technique in deep learning that involves using a neural network model that was previously pre-trained on a similar data set to tune it to a specific domain of the data; however, due to the specific nature of the emulsion data, its true potential has never been fully realised in this study.

The example above is a rare example where this technique is useful: instead of training two instances of the CNN model on the different data sets at the same time, a CNN model that has been pre-trained on the original FEDRA tracks is trained for just another epoch on data set that is made of tracks that have undergone the desegmentation procedure. Strictly speaking, this takes a slightly longer time than doing two instances at the same time; however, it results in fewer GPU computing hours, which generally are scarce on the HTCondor batch farm.

4.6.6 Final output

Once the neural networks have been trained, they can be used as additional filter steps that significantly improve the concentration of genuine seeds. It consists of two distinct steps: seed link analysis and vertex building, which are described in more detail below.

Seed link analysis: This post-neural network step uses classification scores from neural networks to improve SNR and reduce the number of seeds in the final vertex merging step. Detailed algorithm information is in the ANNDEA documentation. At a high level, this step explores links between a subject seed and related seeds based on common tracks. For instance, inspecting seed AB involves looking at related seeds containing track A or B. The presence of seeds like AC and BC with common track C increases confidence that tracks A and B are from the same particle. This confidence is quantified and optimised using calibration data. Link analysis relies on neural network predictions, making its performance dependent on the neural network's effectiveness.

Vertex building: After refining seeds through a combination of analytical methods and NN-based selection techniques, they are combined into vertices of greater multiplicity as long as the merged seeds have at least one track in common. Once the vertices have been assembled, they are utilised to create a table that links the newly formed reconstructed vertex identifier to the tracks in the dataset.

4.6.7 Performance on the Monte Carlo simulation

Results

Two vertexing procedures were conducted using ANNDEA, taking approximately 12 hours to complete. The log data has been presented visually in Figure 4.37 below.

After obtaining the result of the data reconstruction, the relevant statistical and quality track metrics have been calculated and presented in Table 4.8 below.

The results in Table 4.8 are presented separately for different particle sources in order to highlight potential domain-dependent effects on vertex reconstruction performance. In addition to the simulation truth categories already introduced in Table 4.1.4, a dedicated category of fake vertices is reported. These correspond to reconstructed vertices that cannot be matched to any Monte Carlo simulated vertex and arise from random track crossings or reconstruction artefacts. Consequently, they provide a direct measure of the vertexing algorithms susceptibility to instrumental background.

In addition, the number of reconstructed vertices per category is reported, representing the number of vertex identifiers assigned by the respective vertexing algorithm. It can be observed that the number of reconstructed vertices in the non-fake

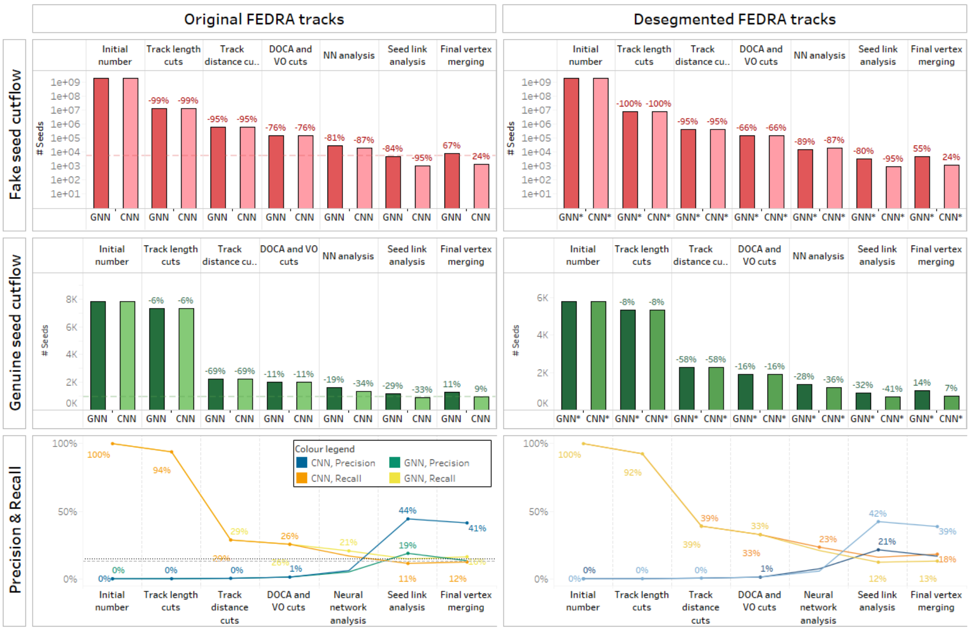


Figure 4.37: A fake (**Top**) and genuine (**Middle**) vertex seed cutflow through various phases of reconstruction, followed by precision and recall charts (**Bottom**).

Table 4.8: Statistical data for the control domain assessing ANNDEA vertexing performance. The impact is represented as a percentage or percentage point change. Positive impacts are shown in green, while negative impacts are shown in red.

Particle source	**Metric**	**FEDRA**	**GNN**	**GNN impact**	**GNN***	**GNN* impact**	**CNN**	**CNN impact**	**CNN***	**CNN* impact**
EM showers	# Rec. Vx.	61±7.8	154±12.4	152.5±38.1%	161±12.7	163.9±39.7%	23±4.8	-62.3±9.2%	32±5.7	-47.5±11.5%
	# Mtch. MC Vx.	61±7.8	171±13.1	180.3±37.0%	174±13.2	185.2±38.8%	22±4.7	-63.9±7.7%	31±5.7	-49.2±9.8%
	VRE (%)	5.7±0.7	16.0±1.1	10.3±1.2pp	16.2±1.2	10.5±1.3pp	2.1±0.4	-3.6±0.7pp	2.9±0.5	-2.8±0.7pp
	VTRE (%)	93.9±2.3	96.3±1.1	2.4±3.1pp	96.1±1.1	2.2±3.2pp	94.1±2.8	0.2±4.6pp	92.9±3.2	-1.0±5.0
	VTRP (%)	91.8±2.2	86.6±1.8	-5.2±3.6pp	88.5±1.7	-3.3±3.6pp	96.0±2.6	4.2±4.2pp	98.5±1.5	6.7±3.3pp
Hadronic decays and μ nuclear interactions	# Rec. Vx.	79±8.9	72±8.5	-8.9±14.9%	58±7.6	-26.6±12.7%	73±8.5	7.6±15.0%	58±7.6	-26.6±12.7%
	# Mtch. MC Vx.	82±9.1	82±9.1	0.0±9.8%	68±8.3	-17.1±10.0%	83±9.1	1.2±8.5%	66±8.1	-19.5±8.9%
	VRE (%)	56.2±6.2	56.2±6.2	0.0±5.5pp	46.6±5.6	-9.6±6.1pp	56.8±6.2	0.6±4.7pp	45.2±5.6	-11.0±5.2pp
	VTRE (%)	77.8±2.4	81.4±2.5	3.6±3.7pp	71.1±3.9	-6.7±5.0pp	80.9±2.4	3.1±3.5pp	66.3±4.5	-11.5±5.2pp
	VTRP (%)	93.8±0.5	92.1±0.7	1.7±0.9pp	91.5±0.9	-2.3±1.1pp	92.2±1.4	-1.6±1.5pp	90.7±1.7	-3.1±1.8pp
ν_μ interactions	# Rec. Vx.	7±2.7	9±3.0	28.6±65.6%	8±2.8	14.3±59.5%	6±2.4	-14.3±47.6%	6±2.4	-14.3±47.6%
	# Mtch. MC Vx.	$7^{+1.0}_{-2.7}$	$8^{+0.0}_{-2.8}$	$14.3^{+71.7}_{-49.4}$%	$8^{+0.0}_{-2.8}$	$14.3^{+71.7}_{-49.4}$%	$6^{+2.0}_{-2.5}$	$-14.3^{+100.3}_{-42.0}$%	$6^{+2.0}_{-2.5}$	$-14.3^{+100.3}_{-42.0}$%
	VRE (%)	$87.5^{+12.5}_{-33.1}$	$100^{+0.0}_{-35.4}$	$12.5^{+33.1}_{-47.9}$pp	$100^{+0.0}_{-35.4}$	$12.5^{+33.1}_{-47.9}$pp	$75.0^{+25.0}_{-30.6}$	$-12.5^{+58.1}_{-43.1}$pp	$75.0^{+25.0}_{-30.6}$	$-12.5^{+58.1}_{-43.1}$pp
	VTRE (%)	72.1±8.0	60.0±7.9	-12.1±9.3pp	61.7±7.5	-10.4±8.3pp	82.9±5.4	10.8±6.0pp	80.0±4.5	7.9±4.5pp
	VTRP (%)	81.6±6.6	76.7±5.1	-4.9±8.1pp	82.9±5.3	1.3±4.3pp	87.9±4.1	6.3±6.1pp	87.5±4.0	5.9±3.4pp
Fake vertices	# Rec. Vx.	1261±36	1053±32	-16.5±3.5%	766±28	-39.3±2.8%	390±20	-69.1±1.8%	295±17	-76.6±1.5%

categories does not always match the number of matched vertices produced by the simulation. This discrepancy arises from imperfections in the vertexing algorithms, which may merge multiple distinct vertices into a single reconstructed vertex or, conversely, split a vertex into several reconstructed vertices (a phenomenon similar to segmentation). As a result, it is entirely possible for two or more reconstructed vertices to correspond to a single MC-generated vertex, and vice versa.

In order to quantify the significance of the impact, uncertainties were calculated for each metric using methods tailored to the type of quantity and particle category. The number of reconstructed vertices was modelled with a Poisson distribution, and the standard error was therefore estimated as the square root of the count. The number of matched vertices and the vertex recognition efficiency VRE were modelled using both binomial and Poisson statistics, since the number of matched vertices depends on the number of reconstructed vertices. The uncertainties of VTRE and VTRP were estimated using the bootstrap method, which involves repeatedly resampling the data and evaluating the variation of the estimates.

Discussion

There are numerous results to analyse; therefore, to maintain clarity in the discussion section, the following structure will be adhered to: first, the performance of the CNN and GNN algorithms, which executed vertexing on the original FEDRA reconstructed tracks, will be compared to the MC simulation and the FEDRA reconstructed picture. The comparison will begin with the seed cutflow statistics, followed by the overall vertex reconstruction results, and then proceed to a more detailed examination of the reconstruction metrics. This will be followed by a prompt discussion of the benefits and advantages that track desegmentation and classification studies performed in Sections 4.4 and 4.5 contribute to the analysis of signal events.

The first notable observation regarding seed processing performance in Figure 4.37 is the remarkably low precision and recall values for CNN and GNN reconstruction pipelines in contrast to those seen in desegmentation studies. The highest precision achieved by CNN is 41%, which is still below the 70-95% range observed in the desegmentation data flow in Figure 4.21. The recall is also low, around 13%. Although the recall for GNN-assisted tracking is slightly higher[14] at 17%, its precision is significantly lower than CNN's, at 13%.

However, compared to the FEDRA output, which has achieved the recall and precision values of 12.5% and 11.8%, respectively, the figures achieved by the neural network already represent a substantial improvement. The generally poor performance can be attributed to the intrinsic complexity of the vertex seed processing, where the number of initial seeds is less constrained than in the cases of track segment seeding and is more difficult to manage.

The second notable observation with respect to seed processing performance is the superior performance of CNNs compared to GNNs, even though both models exhibited similar performance during training, achieving approximately 81-82% accuracy. Surprisingly, this performance disparity arises not from the neural network analysis phase itself, where both models perform similarly, but from the seed link analysis stage.

This suggests that the prediction values generated by the convolutional neural network are more informative than those produced by the GNN. This can be attributed to the lower loss values achieved by CNN during training. However, the reason be-

---

[14] As mentioned in Section 4.3, efforts were made to adjust the algorithm parameters to match the recall with the FEDRA value, thereby enabling a comparison based on precision.

hind this remains unclear: since converting vertex seeds to images inevitably results in greater information loss compared to a similar operation for graphs, GNN models are expected to provide better predictions since they have access to more detailed information about the seed.

Moving from seed to actual vertex reconstruction performance, at a high level, the CNN also displays a superior performance when it comes to the SNR[15] metric. For example, out of the 1407 vertices reconstructed by FEDRA, only 146 or ≈ 10% are genuine, the rest being random crossings. CNN-assisted vertexing generated 491 vertices with a genuine yield of 101 or ≈ 21%. The GNN variant sits somewhere between the mixture of genuine vertices measured at 18%.

The primary factor contributing to these performance discrepancies is the treatment of physics processes such as electron-positron pair production in electromagnetic showers. These vertices account for almost 90% of all simulated vertices, yet they are challenging to reconstruct due to their composition of low-momentum tracks that are highly nonlinear and short. This results in significant uncertainties in the calculation of their parameters, such as DOCA and VO, leading to their rejection during the seed processing phase.

From a physics analysis perspective, they hold no practical value; however, the tracks associated with these processes burden the vertexing algorithms with computations, generating a plethora of fake vertices and consequently increasing instrumental background and skewing the overall vertex reconstruction metrics due to their sheer volume. Paradoxically, it is the poorer performance of CNN in this particular domain that results in the overall better SNR: the limited resolution of seed images is likely to contribute to this by causing the CNN to reject vertex seeds containing tracks coming from these background processes, leading to the overall better performance.

However, once the focus is only given to the secondary hadronic decays and more interesting signal interactions, the differences in performance between FEDRA and neural network-assisted algorithms diminish. In the hadronic sector, all three algorithms achieve comparable vertex reconstruction efficiency, with the reconstructed vertices exhibiting a similar overall quality. In some cases, the performance of individual algorithms slightly exceeds that of FEDRA, while in others it is marginally lower. However, these variations generally fall within the statistical uncertainties and do not indicate any significant advantage or disadvantage. Regarding overall signal reconstruction efficiency, the GNN exhibits the highest value, reconstructing 100% of all signal vertices, whereas the CNN achieves the lowest, with only 75% reconstructed. In terms of signal reconstruction quality, the CNN appears superior, while the GNN shows the lowest performance, particularly for VTRE. However, the extremely limited statistics in this sample result in large uncertainties, rendering all apparent advantages or disadvantages statistically insignificant.

So far, the analysis has been focusing on the vertex performance using original FEDRA tracks: this was to have a fair comparison between the output of the standard vertexing procedure that has used FEDRA tracks only and the neural network-

---

[15] The term *SNR* referenced here pertains to the proportion of any genuine vertices to fake ones regardless of whether they represent a signal event.

assisted algorithms. However, the results of the vertex reconstruction present an excellent opportunity to test the results of the studies in Sections 4.4 and 4.5 in a slightly different context. The first benefit of the desegmentation process can be seen by analysing the columns in the vertexing stats marked with asterisks, where the vertexing quality evidently improves for vertices coming from the secondary hadron decays and neutrino interactions. There is a small decrease in the overall efficiency for hadronic decays, but these are most likely to be caused by track length cuts, the rate of which is sensitive to desegmentation.

An example of the mechanism by which track desegmentation can improve the quality of the reconstructed vertices can be demonstrated by using an example of one of the simulated signal vertices visualised in Figure 4.38 below:

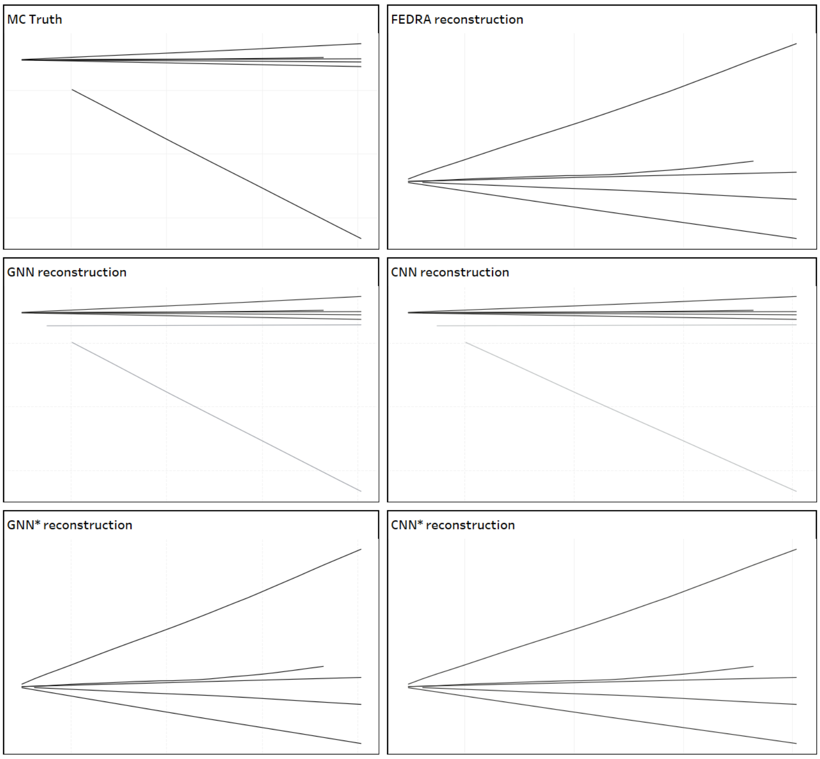


Figure 4.38: An xz-plane projection of a vertex resulting from a simulated neutrino-nuclei interaction in Brick 41 (control domain).

In this example, a muon neutrino participates in the charged current deep-inelastic scattering with one of the nuclei in the tungsten layer of emulsion, resulting in the production of one muon, one proton (the bottom track on the top left image, and two pairs of oppositely charged pions and kaons. FEDRA reconstructs only the part of the vertex containing the muon and mesons, excluding the proton track because of its distance from the interaction origin. The GNN and CNN algorithms do reconstruct the proton track by mistake, associating it with a muon track segment passing nearby into a separate vertex (marked with a faded colour). As a result, each neural network-assisted output contains two vertices: a genuine one, which is the same for all algorithms, and a fake vertex reconstructed by GNN and CNN. However, following the desegmentation procedure, the passing muon track fragment gets merged into a larger track, making it unavailable for vertexing and resulting in fewer fake vertices.

To assess the track classification and its momentum measurement, a more detailed plot of the same event comparing the simulated truth and the holistic reconstructed picture by ANNDEA has been produced in the following.

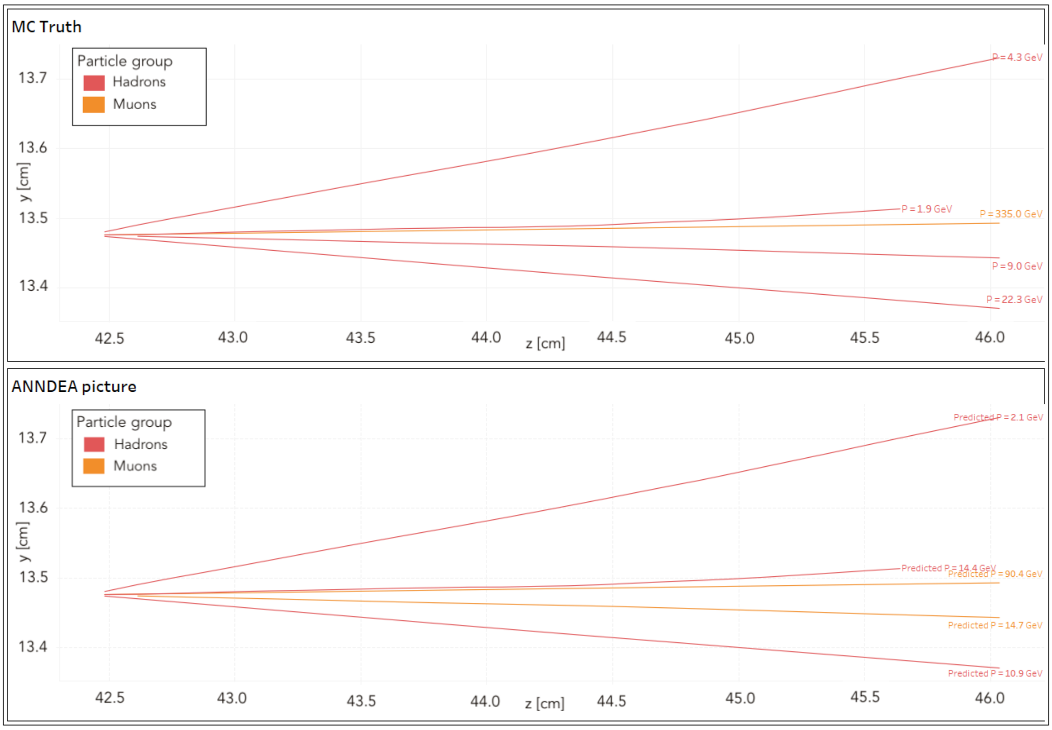


Figure 4.39: A more detailed plot visualising the event mentioned above, focusing on tracks that were reconstructed by the GNN* algorithm. The **Top** part represents the vertex tracks assigned to their particle group and absolute momentum according to the simulation truth. The **Bottom** image is the GNN* vertexing output, along with the track type and momentum prediction carried out earlier by ANNDEA.

It is evident that ANNDEA has accurately identified most of the hadronic tracks, except for one that it mistakenly classified as a muon. This error is likely due to the similar morphology of the tracks. Interestingly, ANNDEA has also assigned a higher momentum value of approximately 90 GeV to the actual muon track, which is close to the 100 GeV threshold set for ANNDEA. This indicates that although both the classification and momentum measurements are not entirely precise, they can complement each other to provide better guidance to the physics event analyst.

For instance, considering the physical principle that a muon-neutrino interaction can yield only one lepton, the track identified as a muon with the highest predicted momentum can be regarded as accurate, while the remaining tracks with lower predicted values of momentum are likely to be mesons.

4.6.8 Performance on Run 1 data

To evaluate the performance of the ANNDEA vertex reconstruction on actual data, a Run 1 data set was performed, as shown in Figure 4.9, using the GNN and CNN models. In order to have a fair comparison with the FEDRA reconstruction performance, the original FEDRA tracks that have not been merged using the desegmentation procedure were used for vertexing.

The reconstruction of the Run 1 dataset took approximately 14 days for both models. Several factors have led to this long reconstruction, with the primary issue being a higher track density in the data, leading to a large number of seeds, the number of which is less constrained than in the desegmentation procedure.

Results

Since Run 1 data lack any Monte Carlo-generated metadata, a reduced version of the analysis has been performed, which consisted of plotting the reconstructed vertex track multiplicity distribution and comparing it with the corresponding plot for the simulation.

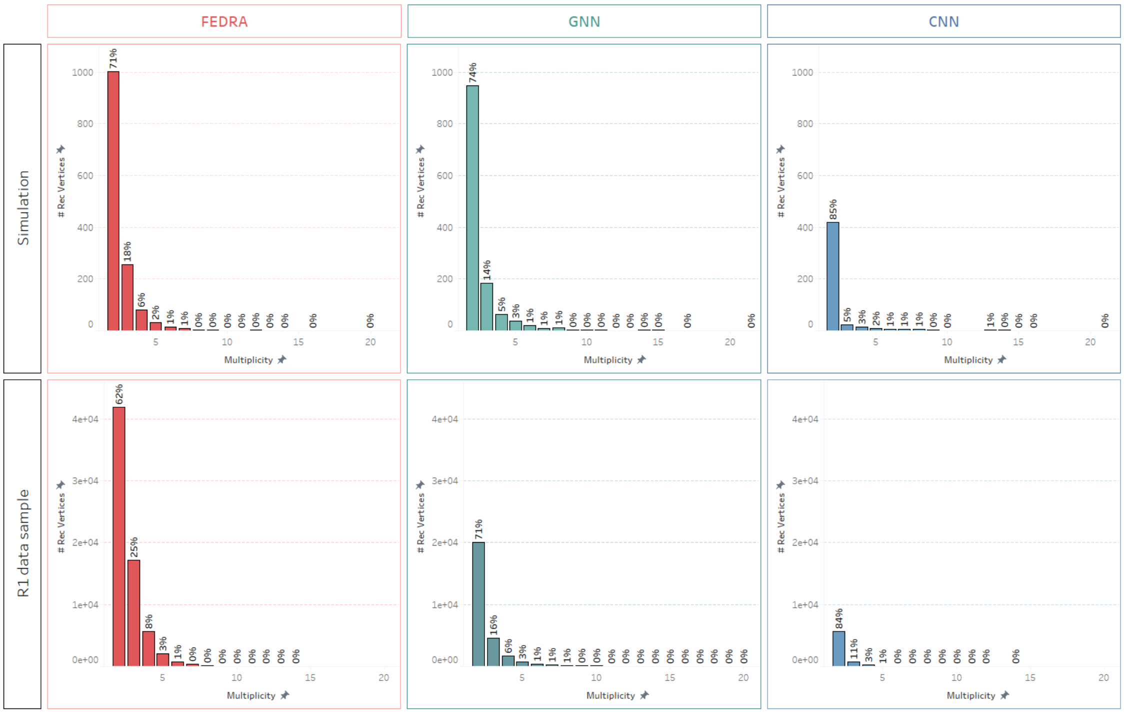


Figure 4.40: Vertex multiplicity distribution for three vertexing algorithms on both simulation and Run 1 data.

Along with the geometrical analysis, a visual inspection of a random sample of the merged tracks has been performed to identify any potential issues below in Figure 4.41.

Discussion

Multiple findings can be inferred by looking at the vertex multiplicity distribution graphs.

All algorithms, irrespective of data type, are heavily skewed toward vertices with low multiplicities, with approximately between 60%-85% of vertices, depending on the algorithm, made of just two tracks. This is perhaps unsurprising since it is

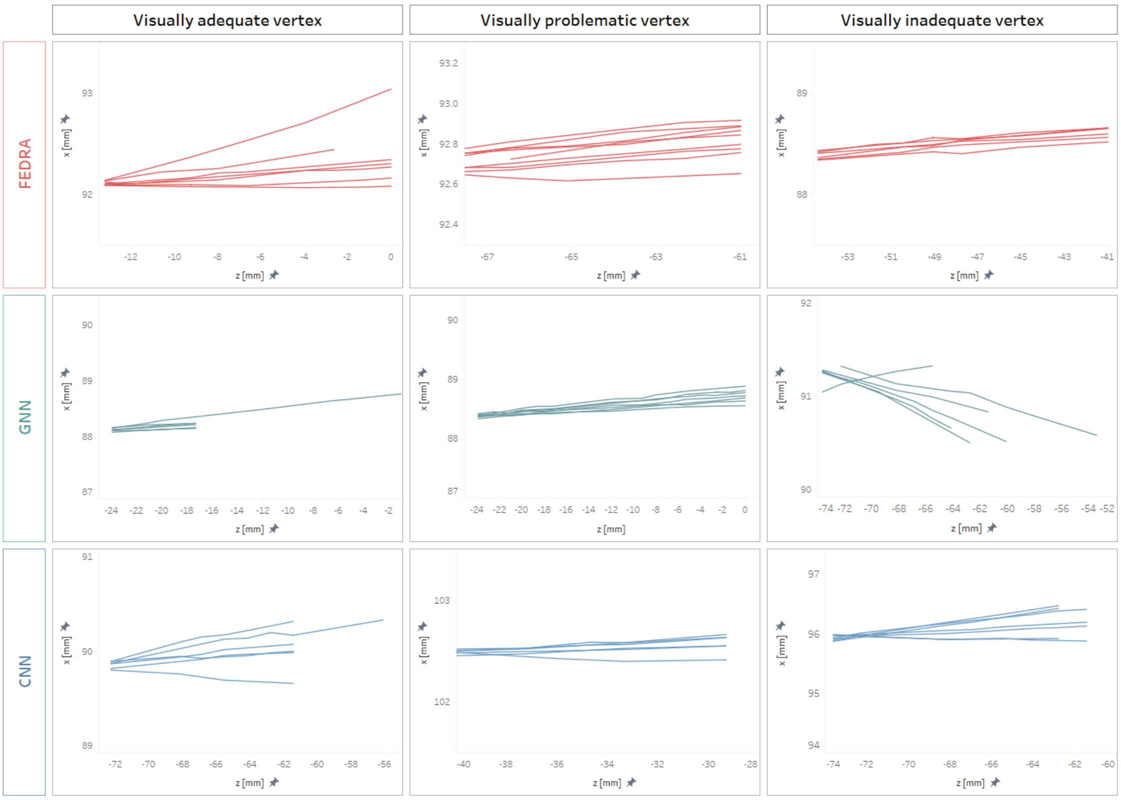


Figure 4.41: A collection of visualizations of vertices reconstructed by FEDRA (**Top**),GNN (**Middle**) and CNN (**Bottom**) algorithms projected on xz plane. Each collection includes three categories that assess the apparent quality of the reconstructed vertex. It is important to note that these categories are subjective and do not determine whether the reconstructed vertices are genuine or random crossings.

expected that the majority of the vertices in ECC are expected to come from electron pair production, which generally yields 2-track vertices. Another plausible reason for the vertex distribution towards low multiplicities is driven by the seed reconstruction efficiency: the more stringent the seed selection, the more likely the resultant merged vertex will comprise fewer tracks. This is apparent for CNN-assisted reconstruction, which, from the simulation studies, has proved to be more conservative than other algorithms and, as a result, has yielded the highest proportion $\approx 85\%$ of two-track vertices.

The second significant observation is the significantly lower number of vertices reconstructed by neural networks compared to FEDRA. In the simulation, the number of GNN-produced vertices is roughly comparable to the number produced by FEDRA, while the number of vertices produced by CNN is less than half compared to FEDRA and GNN. In the actual data, this gap is even larger: FEDRA, GNN, and CNN have produced 91,715, 28,239 and 6,729 vertices, respectively, resulting in a steeper ratio of approximately 100:30:7. Although the exact cause of this reduction is uncertain, analysis of data reconstruction logs suggests that the seed link analysis step played a significant role. This may be attributed to the much higher track density in the Run 1 data sample, which rendered the seed link analysis step less efficient. However, definitive conclusions require studies with Monte Carlo data of similar density.

Before delving into the visual inspection, it is essential to highlight a final, minor yet significant point that was observed but not shown in the graphs: the extensive tails of the GNN multiplicity reaching values over a hundred tracks per vertex, whereas for FEDRA, it is limited to just 21. The ANNDEA vertex reconstruction pipeline inherently does not impose any restrictions on vertex multiplicity, since simulation studies showed that it did not pose any major problems. However, the results of the reconstruction of the actual data suggest that this assumption may need reconsideration.

Numerous findings were inferred from visual inspection of the reconstructed vertices: the first notable observation is a significantly lower quality of the tracks compared to their smoother simulation counterparts. In some instances, their trajectories are so distorted that occasionally, vertexing algorithms get confused and erroneously associate the nearly parallel beam of multiple tracks as a vertex.

A large, almost overwhelming number of vertex anomalies were spotted for GNN-assisted reconstruction, where the final merged vertex obtained a sweeping broom-like appearance with tracks seeming to appear from different directions. For the CNN-assisted algorithm, the proportion of flukes has been significantly lower, and no significant anomalies were detected during random checks on the FEDRA vertices; however, it is important to note that only a small random sample per algorithm was evaluated, leaving some room for a possible bias.

4.6.9 Conclusion

Summary

In summary, this study has explored methods to improve the precision of the existing standard technique for vertexing tracked emulsion data in the SND@LHC experiment. The proposed and examined solution involved adopting a similar approach to the standard vertexing algorithm, with the addition of a neural network component.

The configuration was assessed using CNN and GNN-based reconstruction on an independent set of simulated data that was not used to train neural networks or adjust the parameters of the vertexing algorithm and was contrasted with the results of the FEDRA reconstruction. Although neural network-based methods produced fewer false vertices, the reconstruction performance for key physics events like neutrino-nucleon interactions and hadronic decays was largely comparable across all three methods. The GNN-based algorithm identified more signal vertices than FEDRA but with slightly inferior quality, whereas CNN identified fewer vertices than FEDRA but with superior reconstruction quality. The benefits of desegmentation and track classification have also been assessed, demonstrating a noticeable improvement in fake vertex rejection and analysis of the simulated interaction event.

After the Monte Carlo-based study, a limited sample of scanned Run 1 emulsion data was evaluated using a similar methodology. The results were reviewed by examining the distribution of the multiplicity of the reconstructed vertex and through visual inspection. Compared to simulation, neural network algorithms showed markedly poorer performance than the standard method both in the number and quality of the reconstructed vertices. Anomalies were detected where seemingly unrelated tracks were merged into a single vertex, more frequently in the GNN output and to a lesser degree in the CNN-assisted reconstructions.

Interpretation

While the ANNDEA has achieved some improvements over FEDRA in the simulated set, a clear benefit of using neural networks for vertex reconstruction has not been demonstrated. It seems that all three algorithms are more or less on the same Pareto front: their performance is quite similar, with any minor variations stemming from the selection of adjustable acceptance parameters rather than the inherent strengths of the specific method. Any alteration of these parameters typically results in either decreased efficiency or an increase in the number of false vertices and their lower reconstruction quality.

While GNN-assisted reconstruction shows a marginally improved trade-off capability, the improvement is not substantial enough to warrant further development. The key advantage of Kalman filter-based methods lies in their adaptability to varying data quality conditions. This is evident from the reconstruction outcomes on actual data with a higher track density and misaligned plates, where GNN and CNN both underperformed compared to FEDRA.

It should be noted that the simulation results must be interpreted with caution for various reasons, including the small number of signal events $\lessapprox 10$ and the absence of outcomes from the final event analysis stage. Even if one algorithm reconstructs more signal vertices than another, they may still be discarded in the final analysis due to poor reconstruction quality.

Further research

Future progress in this field is heavily reliant on the quality of simulated data, a point that has been frequently mentioned in prior research and will not be reiterated here. This study has shown that any deep learning algorithm will not function properly in real-world data scenarios without sufficiently dense and realistic Monte Carlo-generated data.

Another problem that can be addressed in the current ANNDEA vertexing module arises from a similar issue seen during desegmentation studies: the limited application of the neural network. As track density increases, relying solely on the performance of the seed classifier may be inadequate, allowing numerous false seeds to proceed to the vertex merging stage. This may result in situations where individual seeds appear valid, but the overall vertex looks unphysical. Unlike the desegmentation study, the present implementation of ANNDEA offers an intermediate solution: the link analysis step, which helps to eliminate many false seeds by leveraging their relational information and neural network predictions. However, it still falls short of the more advanced seed merging and fitting process performed by FEDRA. An adequate ML counterpart to the standard reconstruction process would be to introduce an additional stage in the merging process where a separately trained neural network assesses the overall merged seed group and discards new seed candidates if they do not align well with the resulting vertex.

Another more radical and yet more interesting and promising solution is to abandon the *vertex finding and fitting* paradigm used for both standard reconstruction and current implementation of ANNDEA altogether and opt for a fundamentally different approach using the graph analysis method where the tracks in the emulsion data can be encoded as graph nodes each having its own attributes encoding its coordinates and orientation in space. Within this approach, the possible links between these tracks can be assessed using the GNN model, which performs an edge analysis of the links between these tracks. The upcoming study will investigate this concept within the tracking framework, and if it proves to be successful, it can be similarly used in this context.

4.7 Track reconstruction

4.7.1 Purpose

This final study revisits the standard tracking quality issues raised in Section 4.1 and attempted in Section 4.4, approaching them from a new perspective. Rather than improving the track previously generated by FEDRA, a comprehensive tracking process will be implemented on both the simulated and actual data with the support of the neural networks.

The tracking procedure is arguably the most challenging operation in emulsion data reconstruction due to the large data volume required to process and the associated computational intensity, which requires a carefully designed and optimised pipeline. The literature review in Section 4.2 has provided a promising solution using GNN relation models to assist with the tracking task in ATLAS[160] and has served as a crucial starting point[187] for the tracking algorithm developed here. On a high level, this solution involves building an interconnected graph of hits and classifying it using a previously trained GNN to analyse the connections between these hits and leveraging this information to build tracks.

Nevertheless, several challenges emerge because, unlike the electronic tracking detectors at ATLAS, the emulsion data cannot distinguish hits by event timestamp, requiring all hits gathered in a single exposure cycle to be treated as a single large event. Consequently, this results in a substantially higher density of hits per unit volume than what is typically seen in an ATLAS recorded event, making it unfeasible to construct and analyse an interconnected graph for an entire emulsion unit with current computing capabilities.

In order to circumvent this technological limitation, an adapted version of the tracking process has been developed in ANNDEA that consists of the following steps.

1. Prepare the hit data that has been previously simulated by Monte-Carlo or generated by the emulsion scanning procedure.
2. Split the data into smaller rectangular prisms or *hit clusters*, each occupying a different region of the target ECC unit.
3. Perform GNN-assisted tracking of each set of hits that occupy these clusters individually.

4. Merge all track segments that were reconstructed in each individual cluster in the previous step into complete volume tracks.

The clustered approach enables a better distribution of the data processing load across multiple computing nodes, such as HTCondor, but this parallelisation comes at the cost of making the tracking algorithm quite complex and its execution time-consuming. Although the full description is beyond the scope of this work[16], the most important processing details that will be useful for the interpretation section of the results will be covered in more detail in the following.

4.7.2 Data preparation

Unlike other neural network-based analyses in this study, the data preparation for tracking is quite simple. It requires processing the initial input file to retain only specific variables: x, y, z, tx, ty, and the *Hit Id*, which serves as a unique identifier for each reconstructed hit in the data set. If additional data are necessary for training the neural network, a variable *MC Track ID* is included, providing an identifier for the corresponding MC simulated track, and potentially other optional variables like *PDG ID* if the GNN model needs to focus on particular types of tracks during the training phase.

If only simulation is involved, then no further data selection steps are required; however, for scanned emulsion data, the hit reconstruction quality criteria described in Section 2.5.4 are applied. In addition to the hit selection, the scanned emulsion data undergo an alignment procedure covered in Section 2.5.4: this is an important step to minimise the gap between the simulated perfect conditions on which GNN was trained and the environment encountered in the actual emulsion data.

4.7.3 Splitting dataset into hit clusters

Once the data have been obtained, it is divided into smaller clusters of size $h \times w \times l$ where the longest component l is aligned along the z-axis. This size has been determined from performance and optimisation studies in the calibration domain based on the following factors:

- The trajectories of most particles in the SND@LHC detector are collinear with the z-axis, and therefore the rectangular prism is favoured over the cube.
- The height and width of each cube should not be too small, as it limits the maximum azimuthal angle of the average particle path with respect to the z-axis through the formula $\theta_p \leq \arctan(\frac{h}{2P_{gap}})$ where h is the height or width of the cluster and P_{gap} is the average distance between emulsion plates equal to ≈1.3 mm placing the current limit at ≈67°.
- The number of hit combinations within the cluster, and therefore the memory and computational load during processing, scales cubically with volume.

Hence, the dimensions chosen for this study are 6 mm × 6 mm × 12 mm, representing a compromise between enhanced tracking performance from larger clusters and the

---

[16] A more complete information with technical details can be found on the dedicated wiki page available at [184].

computational benefits of smaller volumes. Consequently, this leads to a minimum number of clusters required for parallel processing, calculated based on the detector's fiducial volume dimensions H, W, and L using the formula:

$$N_{min} = \frac{H}{h} \times \frac{W}{w} \times \frac{L}{l} \tag{4.9}$$

where N_{min} is a round up integer number. For an ECC unit which has dimensions $192 \times 192 \times 78$ mm^3, this means that at least $\approx 6.7 \times 10^3$ jobs have to be processed.

Unfortunately, even this large number is not sufficient, as it represents the reconstruction setup where each cluster independently tracks a distinct region of space. Within this configuration, it is challenging to link segments from the different clusters together, resulting in extremely high segmentation of the output volume tracks. To address this issue, a new set of clusters is initiated, each covering an area that intersects with two of the original clusters in some direction. These additional clusters contain reconstructed tracks that serve as connectors, facilitating the merging of track segments from clusters that do not overlap.

It was found that the best number of extra overlapping hit regions for achieving peak performance, while minimally affecting the HTCondor system, is one in both the x- and y-directions and two in the z-direction. Consequently, the estimated total number of hit clusters that need processing in a standard ECC brick is approximately $\approx 5.2 \times 10^4$. This requirement to handle such a high volume of operations significantly contributes to the prolonged duration of the track reconstruction process outlined here.

4.7.4 Hit cluster tracking

Once the set of hit clusters has been defined, each of them can undergo the local tracking procedure consisting of the following steps: seed initialisation, fiducial cuts, creating graph representation, GNN analysis and segment assembling.

Seed initialisation

In the particular context of hit cluster tracking, the term *seed* refers to a pair of hits i and j potentially originating from the same parent particle track and can mathematically be expressed in the shape of a vector:

$$\mathbf{sd}_{ij} = [h_i \; x_i \; y_i \; z_i \; tx_i \; ty_i \; h_j \; x_j \; y_j \; z_j \; tx_j \; ty_j \; label_{ij}] \tag{4.10}$$

where h_i is a hit i and the h_j is a hit identifier for hit j. The elements x_i, y_i, z_i, tx_i and ty_i represent the spatial coordinates and angle of the basetrack i and the elements x_j, y_j, z_j, tx_j and ty_j represent the same set of parameters for the hit j. The *label* element is not mandatory and is only included in the data sets intended for neural network training. It is a Boolean variable that can take a value of 1 or 0, indicating whether the constituent hits share the same desirable MC-generated track or not.

The word *desirable* implies that in some scenarios, the simulated track is required to represent a particular group of particles. For example, if the GNN model is required to be trained to recognise only seeds coming from the muon tracks, the label still can take a value of 0 even if both hits belong to the same non-muon particle.

Seed preliminary selection

The initial number of seeds is determined by the equation $N_{sd} = N_h^2$, and for a hit cluster located in the dense hit environment, it can still be too large to convert it into a graph object. To reduce the computational load on the neural network, a further set of simple filters, similar to those used in the track segment merging algorithm, is applied:

1. **Eliminate self-permutations**: discard any seeds where two hits are identical, reducing the number of seed candidates from the original N_h^2 to $N_h^2 - N_h$.
2. **Remove duplicates**: the sequence of hits does not matter: The seed $\mathbf{sd}_{ij}$ is equivalent to the seed $\mathbf{sd}_{ji}$, so only the initial one is retained. This process reduces the seed cut flow to $N_{sd} = (N_{tr}^2$ - $N_{tr})/2$.
3. **Enforcing cross-plate connection only:** The connections between nodes that are within the same plate are eliminated under the physical assumption that the particle leaves only one hit per plate.
4. **Relative angle cut**: It is presumed that the particle's path remains fairly consistent, and should there be any deflection from Coulomb scattering, the angular variation remains minimal. Hence, seeds are discarded if the difference between their hit angles along the x or y-axis[17] exceeds a value of 0.2 radians or $\approx 11°$.
5. **Distance cut:** It is assumed that the trajectory of the particles adheres largely to a linear path with limited deviation. The distance from the line is calculated using the formula: $dx = \sqrt{((x_i - x_j) + (tx_j \times (z_i - z_j)))^2}$ and $dy = \sqrt{((y_i - y_j) + (ty_j \times (z_i - z_j)))^2}$. The cut on these values is set at 60 microns following the optimisation studies performed on the calibration domain.

Creating graph representation

In the context of tracking, a graph object has been selected with the following properties:

- Each node on the graph is represented by a hit with attributes encoding the spatial and angular coordinates.
- Every single node is interconnected with other nodes through one-directional edges.
- Each edge encodes numerous geometric properties, such as the distance between hits and their relative angle orientation.
- If the cluster is used for training, a Boolean flag indicating whether the edge connects hits within the same or different tracks is included.

---

[17] Assuming that the physics of interactions is independent of rotation around the trajectory line.

The graph representation scheme has been adopted from the public Github repository[187] but with numerous modifications associated with different coordinate systems used in ATLAS (cylindrical) and SND@LHC (cartesian), and the fact that the latter does not have any magnetic fields.

GNN analysis

The primary goal of the GNN analysis, similar to the approaches in Sections 4.4 and 4.6, is to leverage the graph information to eliminate as many fake seeds as possible while retaining the genuine ones. However, here, rather than classifying the entire graph, it focuses on classifying the links between hits or seeds, a process known as edge analysis. Each seed is rated on a scale from 0 to 1, and if this score falls below a certain model-defined threshold, the corresponding edge is removed. This action decreases the number of seeds and, consequently, the number of edges in the graph.

Track segment assembling

Although the output from the previous step provides a valuable hit relationship structure, it remains too unrefined for substantial analysis because it comprises a collection of seeds rather than complete tracks. The primary objective of this step is to consolidate these seeds into extended track segments. This process is facilitated through a cycling algorithm that constructs tracks in the following steps:

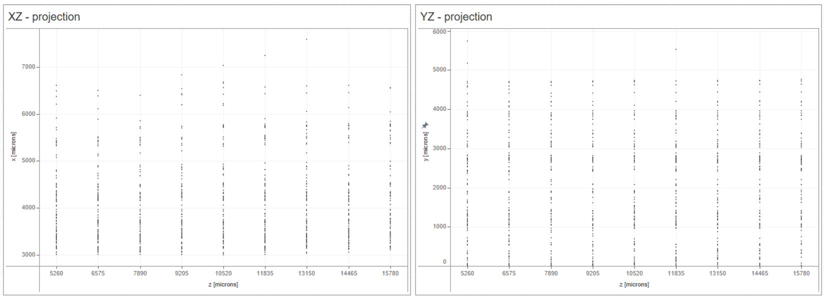
1. Firstly all seeds are merged based on common hits into all possible combinations that can make up a track segment within a cluster.
2. The evaluation of each track segment candidate is based on the quality of fit, which considers various elements, including the initial weighted scoring of constituent seeds by GNN and the number of non-conflicting seeds involved in the merging process.
3. The candidate achieving the highest score is chosen for the definitive cluster-approved track list, and any hit linked to the chosen track candidates is discarded to prevent their involvement in subsequent iterations.
4. Steps 1,2, and 3 are repeated multiple times until there are no suitable seeds to build track segment combinations anymore.
5. All hits that are part of the final track segments get assigned the temporary track segment IDs: all unused hits get discarded.

4.7.5 Final output

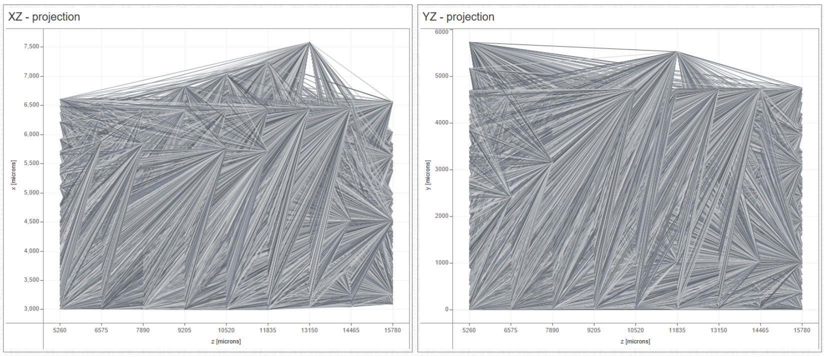
After reconstructing each cluster, the associated track segments, which were identified during the cluster tracking phase, must be connected to form complete volume tracks. This connection must occur across the x-, y-, and z-axes by leveraging overlapping clusters. This facilitates the merging of all track segments, exemplified by a merging procedure in the z-direction:

1. Load the first reconstructed cluster in the z-axis, which contains the reconstructed track with the respective identifiers that were previously assigned during hit cluster reconstruction.

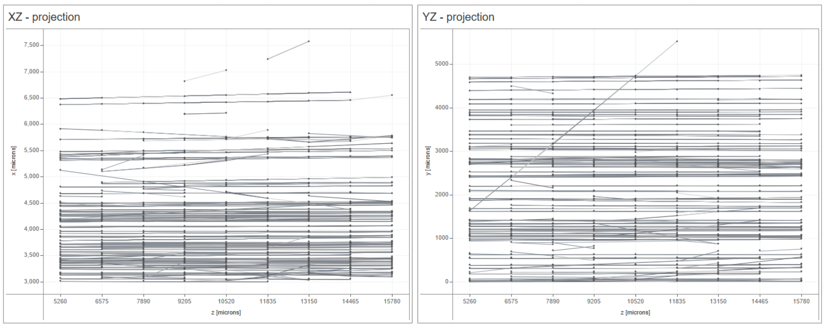
Phase 1: An example of raw cluster hit data



Phase 2: An example of interconnected cluster hit data



Phase 4: An example of interconnected cluster hit data reweighted by GNN model



Phase 5: An example of interconnected cluster hit data that undergone local tracking

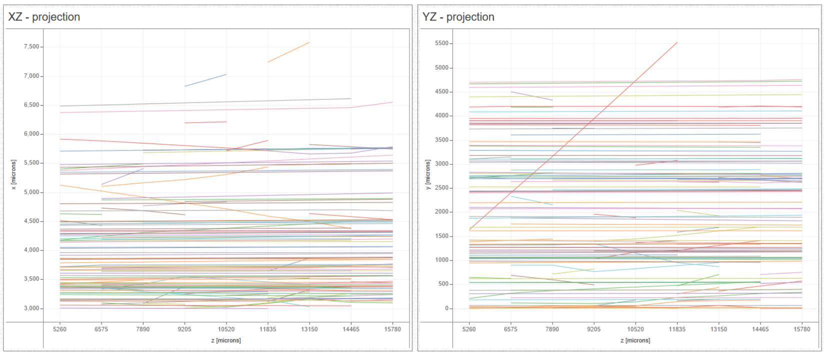


Figure 4.42: A visualised example of a typical hit cluster tracking process. **Phase 1:** Initialised hit cluster containing 850 hits. **Phase 2:** Interconnected hits in the cluster, excluding self-permutations and duplicates. In this example, there are about 320k edges. **Phase 4:** Interconnected hits in the cluster after seed selection and GNN analysis. In this example, there are 2333 edges left. **Phase 5:** Hit cluster tracking result.

2. Begin examining subsequent clusters to determine if any tracks overlap in hits with the master segment. Assign the identifier from the previous track segment, which was initially set during hit cluster processing, to the subsequent

track segment that shows the greatest hit overlap with the master segment.

3. Tracks that do not pass the overlap test are considered distinct tracks that originate in the subsequent brick, thus maintaining their initial identifiers.
4. This procedure continues until the final brick on the z-axis of this specific xy set is encountered, at which point all track segments are combined, where possible.

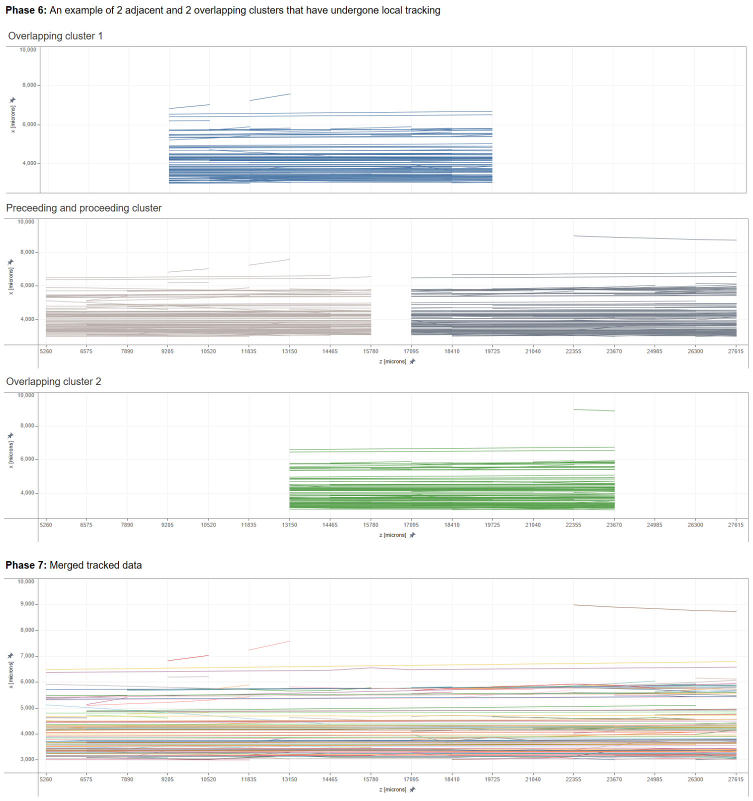


Figure 4.43: A visualised example of a typical hit cluster tracking process. **Phase 6:** A visualised set of clusters overlapping along the z-axis. **Phase 7:** Tracks that have come as a result of merging clusters in Phase 6.

Consequently, most of the segments in this specific cluster set are consolidated into complete tracks, and every cluster along the z-axis for each xy set is combined into one. This integration can only occur when there are intersecting track clusters in which segments share hits from both the previous and subsequent clusters. Therefore, the presence of overlap is crucial to minimise segmentation.

It is anticipated that the majority of incoming tracks absorbed by the target will predominantly have a boost in the z-direction with low values of the transverse momentum, which makes merging only along the z-axis sufficient for most tracks. However, the SND@LHC experiment concentrates on neutrino interactions, which typically involve products with significant transverse momentum. Consequently, it is advisable to reconstruct at least one overlapping cluster on both the x- and y-axes. This involves taking all volume tracks from the previous step and applying the same merging process on the x- and y-axes, resulting in more complete volume tracks, ready for the final output.

A further smoothing process is implemented on each derived track to ensure the absence of irregularities like incorrectly linked hits, causing tracks to display an unnatural "zig-zag" pattern. This procedure is required due to a specific drawback of employing Machine Learning discussed in Chapter 2: the sporadic absence of "common sense" leading to an unrealistic output.

4.7.6 Neural network training

For the tracking procedure, ANNDEA employs a particular physics-motivated instance of the GNN neural network called *Interaction Networks*[188] or IN capable of reasoning about objects and their relations. The key mechanism integrated into the interaction network is the message passing mechanism described in Section 2.6.5, which re-embeds the nodes and edges with the information from their nearest neighbours, thus enabling each node to "learn" about the surrounding network and hence re-establish its particular role in the graph, thus enabling its proper classification. For this study, a particular IN configuration has been used:

- Each node and edge participates in message passing with a number of re-embeddings reaching 5, which determines the knowledge reach of each node and edge about its local neighbourhood. For the dimensions of the hit cluster used in this study, this should be sufficient since each hit cluster can accommodate a track segment with a maximum number of hits ≈8.
- Each re-embedded node and edge is assigned a MLP instance made of 5 hidden layers, each consisting of 80 nodes.
- The GNN learns from this data representation through the back-propagation technique by comparing the input of the embedded representation and the corresponding class label, which determines whether the edge is genuine or not.

Two training data sets were developed by processing data from the ML domain through the data preparation and hit cluster generation steps described above. The main distinction between these datasets lay in the definition of the class label: in the first dataset, the Boolean label was assigned a value of 1 for pairs of hits originating from the same truth particle, whereas in the second dataset, the label was set to 1 for hits generated by non-electrons. This approach was implemented to investigate the feasibility of employing specifically trained GNNs to better reconstruct tracks from certain types of physical events.

Each training sample consisted of three distinct data samples:

- **Training set:** a sample of the ≈ 36.5k data points used for the actual GNN training.
- **Validation set:** a sample of the ≈ 4.2k data points used to determine the optimal value of the acceptance threshold that removes edges based on the prediction value of the GNN, which can run between 0 and 1.
- **Test set:** a sample of the ≈ 2.1k data points used for the final independent test of the trained model and the optimised value of the acceptance of its predictions.

The relative split of the samples has followed a 85:10:5% split: a commonly used ratio in machine learning.

Two GNN models have been trained with a starting learning rate of 10^{-4}, with validation loss and the accuracy profile plotted below.

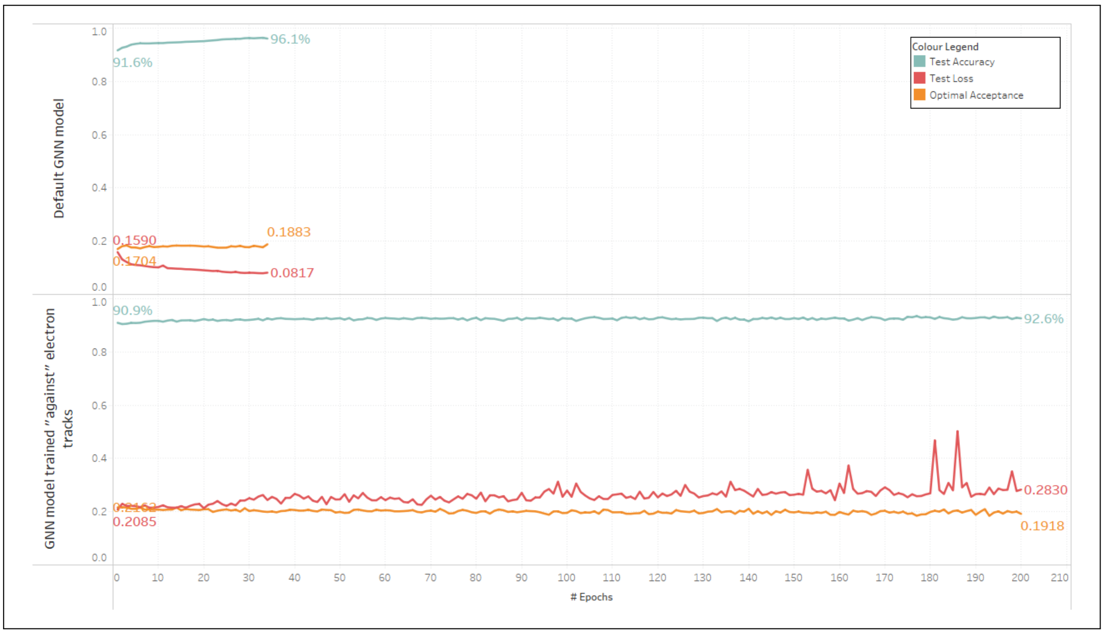


Figure 4.44: A plot of the normalised test accuracy, loss and the optimal value of acceptance of two GNN models training.

Although the standard GNN model rapidly achieved 96% accuracy, the version tailored for hadron and muon track reconstruction has faced difficulties matching this performance, with the final test accuracy just surpassing 92% despite efforts to reduce the learning rate. A likely reason for this issue is the model being penalised when it mistakenly classifies authentic energetic electron hit seeds, which resemble those from muons and hadrons, as the incorrect type due to pre-set opposite labels.

4.7.7 Performance on the Monte Carlo simulation

To assess the tracking performance, the control domain of the simulated data was processed using the default GNN model and its tuned version.

Results

Two tracking procedures were carried out, each lasting about one week. The duration of these experiments was mainly due to the large number of hit clusters submitted to HTCondor and the resulting pressure on the EOS drive system, which had to manage a significant number of data read and write operations. After obtaining the result of the data reconstruction, the relevant statistical and track quality metrics have been calculated and presented in Table 4.9 below:

Table 4.9: The control domain statistical data on the ANNDEA tracking performance. The impact is measured in percentage or percentage point change with respect to FEDRA. Improvements are coloured in shades of green, while worse performances are highlighted in shades of red. The statistical uncertainties were calculated using the same methodology as in Section 4.4.

Particle source	**Particle group**	**Measured metric**	**FEDRA**	**Default GNN**	**Default GNN impact**	**Tuned GNN**	**Tuned GNN impact**
EM showers and delta-rays	**Electrons**	Number of tracks	29,968 ± 173	67,038 ± 259	123.7 ± 1.6%	28,653 ± 169	-4.4 ± 0.8%
		Segmentation	1.351 ± 0.005	1.445 ± 0.004	7.0 ± 0.5%	1.231 ± 0.006	-8.9 ±0.6%
		TRE (%)	39.7 ± 0.2	68.5 ± 0.3	28.8 ± 0.4 pp	30.6 ± 0.2	-9.1 ± 0.3 pp
		THRE (%)	42.1 ± 0.2	50.5 ± 0.1	8.4 ± 0.2 pp	45.0 ± 0.2	3.0 ± 0.2 pp
		THRP (%)	62.0 ± 0.2	66.7 ± 0.1	4.7 ± 0.3 pp	75.3 ± 0.1	13.3 ± 0.3 pp
External muons from IP1	**Muons**	Number of tracks	52,664 ± 229	50,156 ± 224	-4.8 ± 0.6%	48,500 ± 220	-7.9 ± 0.6%
		Segmentation	1.276 ± 0.005	1.427 ± 0.006	11.8 ± 0.7%	1.367 ± 0.006	7.1 ± 0.6%
		TRE (%)	98.2 ± 0.5	$100.0^{+0.0}_{-0.5}$	1.8 ± 0.7 pp	$100.0^{+0.0}_{-0.5}$	1.8 ± 0.7 pp
		THRE (%)	80.2 ± 0.1	84.0 ± 0.1	3.8 ± 0.2 pp	87.6 ± 0.1	7.4 ± 0.2 pp
		THRP (%)	98.2 ± 0.0	96.8 ± 0.0	-1.4 ± 0.0 pp	97.1 ± 0.0	-1.0 ± 0.0 pp
Muon neutrino interactions	**Muons**	Number of tracks	25 ± 5.0	31 ± 5.6	24.0 ± 33.4%	32 ± 5.7	28.0 ± 34.3%
		Segmentation	1.042 ± 0.208	1.240 ± 0.223	19.0 ± 32.0%	1.280 ± 0.226	22.8 ± 32.8%
		TRE (%)	$96.0^{+4.0}_{-20.0}$	$100^{+0.0}_{-20.0}$	$4.0^{+20.0}_{-24.0}$ pp	$100^{+0.0}_{-20.0}$	$4.0^{+20.0}_{-24.0}$ pp
		THRE (%)	95.2 ± 4.0	84.1 ± 5.2	-11.0 ± 6.6 pp	92.6 ± 3.8	-2.6 ± 5.5 pp
		THRP (%)	99.8 ± 0.1	98.7 ± 0.6	-1.1 ± 0.6 pp	99.7 ± 0.1	-0.1 ± 0.2 pp
	Hadrons	Number of tracks	108 ± 10	118 ± 10	9.3 ± 13.7%	111 ± 10	2.8 ± 13.3%
		Segmentation	1.325 ± 0.126	1.371 ± 0.124	3.5 ± 13.6%	1.295 ± 0.121	-2.3 ± 13.1%
		TRE (%)	$91.2^{+8.8}_{-10.5}$	$97.8^{+2.2}_{-10.5}$	$6.6^{+12.7}_{-19.3}$ pp	$96.7^{+3.3}_{-10.5}$	$5.5^{+13.8}_{-19.3}$ pp
		THRE (%)	68.7 ± 3.8	73.3 ± 3.8	4.6 ± 5.4 pp	74.9 ± 3.8	6.2 ± 5.4 pp
		THRP (%)	97.1 ± 0.9	96.2 ± 1.1	-1.0 ± 1.4 pp	97.3 ± 0.9	0.2 ± 1.3 pp
μ, e, γ nuclear interactions	**Hadrons**	Number of tracks	471 ± 22	524 ± 23	11.3 ± 7.1%	526 ± 23	11.7 ± 7.1%
		Segmentation	1.668 ± 0.077	1.450 ± 0.063	-13.1 ± 5.5%	1.462 ± 0.063	-12.4 ± 5.5%
		TRE (%)	65.4 ± 4.5	84.8 ± 4.7	19.4 ± 6.5 pp	85.0 ± 4.8	19.6 ± 6.5 pp
		THRE (%)	45.7 ± 1.8	62.4 ± 1.9	16.7 ± 2.6 pp	59.4 ± 1.8	13.7 ± 2.6 pp
		THRP (%)	99.2 ± 0.4	98.3 ± 0.3	-0.9 ± 0.5 pp	98.9 ± 0.2	-0.3 ± 0.4 pp
Hadron decays and inelastic scattering	**Hadrons**	Number of tracks	1,434 ± 38	1,645 ± 41	14.7 ± 4.2%	1,656 ± 41	15.5 ± 4.2%
		Segmentation	1.796 ± 0.047	1.511 ± 0.037	-15.9 ± 3.0%	1.525 ± 0.037	-15.1 ± 3.1%
		TRE (%)	61.4 ± 2.6	84.8 ± 2.7	23.4 ± 3.7	84.2 ± 2.7	22.8 ± 3.7
		THRE (%)	37.9 ± 0.9	54.2 ± 1.1	16.3 ± 1.4 pp	52.6 ± 1.0	14.7 ± 1.4 pp
		THRP (%)	99.6 ± 0.1	98.6 ± 0.2 pp	-1.0 ± 0.2	98.9 ± 0.2	-0.2 ± 0.7 pp

Discussion

Before delving into the discussion, it is essential to underscore the main distinctions between the tracking conducted by the standard version of GNN, which is trained on all available tracks in the training dataset, and the GNN specifically adjusted for non-electron tracks. Taking into account a scenario with purely simulated data using Table 4.1, electrons are expected to make up the majority of tracks (≈ 72%) by count. Consequently, a significant proportion of truth seeds encountered by a neural network in a standard data set will be from electrons and positrons whose track morphology significantly differs from that of high-energy muons.

The tuned GNN model has been trained on a training set where the identification of hits coming from the same electron-flavoured particle has been penalised by modifying the appropriate class labels. This can be thought of as performing track reconstruction in a "classical regime" which tends to be more optimised for tracking particles on which the standard tracking algorithms like FEDRA have been designed to operate. The default instance of GNN, on the other hand, is more optimised to reconstruct electron tracks produced in the electronic showers and low-energy hadrons: the type of tracks that standard charged particle tracking algorithms traditionally struggle with.

The default GNN model-based tracking procedure, as expected, has demonstrated strong performance in the electron sector, resulting in a significantly higher percentage of reconstructed electron tracks with a reconstruction efficiency almost double that achieved by FEDRA. For muonic tracks, the scenario is twofold: in terms of track reconstruction efficiency, the default GNN-based tracking possesses advantages over traditional methods, particularly in signal events where the reconstruction efficiency reaches 100%. However, this higher efficiency comes at the price of the lower quality of the track reconstruction, particularly for THRE, where the performance is 11% lower compared to standard techniques, affecting the hit completeness of the tracks relative to their MC-generated counterparts. Nevertheless, it should be noted that the quantity of muons originating from signal events is insufficient to draw broad conclusions about ANNDEA's muon reconstruction capabilities. For the muons coming from IP1, the results are more promising, with reconstruction quality slightly superior to that of the conventional algorithm. However, GNN-assisted tracking is certainly prone to splitting tracks during reconstruction, resulting in segmentation being 10-20% higher than observed for FEDRA reconstructed muons. This problem is particularly troubling as it somewhat negates the progress achieved in Section 4.4.

With the tuned GNN model-based tracking procedure, the situation is almost the opposite of the previous case: while the quality of muon reconstruction has slightly improved, it has come at a significant cost to the efficiency of electron track reconstruction, which has decreased to approximately 30%: even lower than the rate attained by FEDRA. It should be noted that although there has been a slight reduction in IP1 muon segmentation due to the tuned GNN tracking procedure, it still surpasses the one achieved by FEDRA by $\approx 7\%$.

Compared to FEDRA, both GNN models demonstrated a superior performance in the hadron sector, marked by an increase in tracking efficiency where the TRE parameter rose from approximately 60-65% to around 85%, and better reconstruction quality, evidenced by a reduction in track segmentation by approximately 12-15% and a THRE metric improvement of roughly 16-17 percentage points.

To illustrate a more qualitative comparison of various reconstruction algorithms, a signal event depicting a muon-neutrino nucleus interaction, where one of the tracks generates a cascade of electromagnetic showers along its trajectory, has been visualised in the figure below:

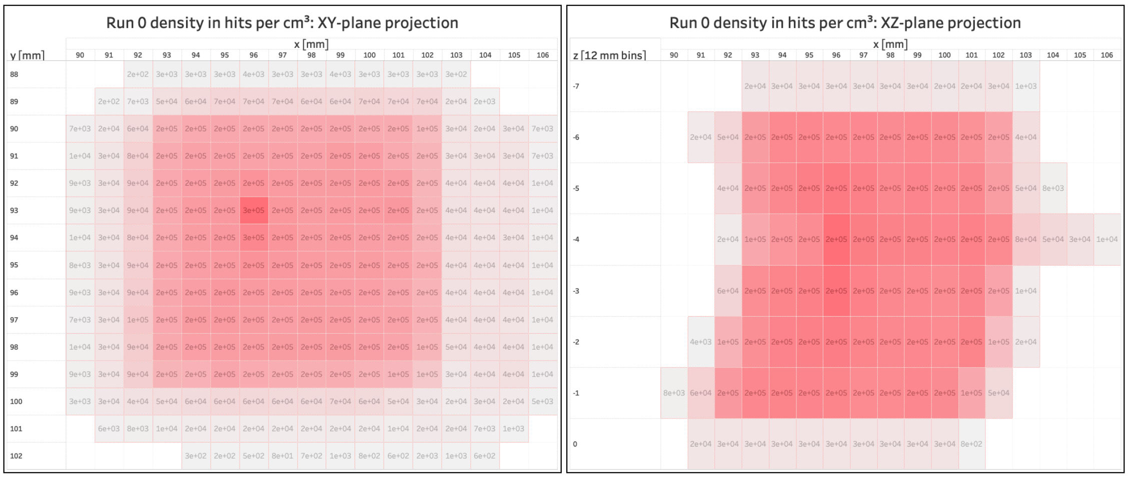
Among the three approaches, the conventional ANNDEA tracking reconstruction surpasses both FEDRA and the GNN-tuned variant in accurately replicating the simulated event truth. In contrast to FEDRA, the electromagnetic shower profile is more complete, and both hadron and muon tracks show fewer segmentation problems compared to the standard reconstruction method. The tuned GNN-aided

Figure 4.45: A visualised example of one of the simulated signal events in the control domain with tracks reconstructed by different algorithms.

algorithm performance falls somewhere in between the FEDRA and the standard GNN reconstruction; although the hadron tracks exhibit slight segmentation issues, the efficiency of electron track reconstruction decreases considerably, mirroring the reduction observed in the FEDRA reconstruction.

4.7.8 Performance on Run 0 data

Before assessing the tracking performance of ANNDEA on the Run 1 data, a similar dataset from Run 0 was utilised for preliminary testing of ANNDEA. This data set was acquired with a simplified configuration of the SND@LHC target, where only a single brick was active, and it was used to calibrate the FEDRA reconstruction parameters. Given that it has accumulated only about ≈ 0.46 fb^{-1}[189] of LHC data, the hit density per unit volume is roughly an order of magnitude lower than in the Run 1 data sample, providing an ideal environment to identify potential early issues in ANNDEA's tracking capabilities on actual data.



Run 0 density in hits per cm³: XY-plane projection

y [mm] \ x [mm]	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106
88			2e+02	3e+03	3e+03	3e+03	4e+03	3e+03	3e+03	4e+03	3e+03	3e+03	3e+03	3e+02			
89		2e+02	7e+03	5e+04	6e+04	7e+04	7e+04	7e+04	6e+04	6e+04	7e+04	7e+04	7e+04	2e+04	2e+03		
90	7e+03	2e+04	6e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+05	3e+04	2e+04	3e+04	7e+03
91	1e+04	3e+04	8e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	3e+04	3e+04	3e+04	7e+03
92	9e+03	3e+04	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	4e+04	1e+04
93	9e+03	3e+04	9e+04	2e+05	2e+05	2e+05	3e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	4e+04	1e+04
94	1e+04	3e+04	8e+04	2e+05	2e+05	2e+05	3e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	3e+04	1e+04
95	8e+03	3e+04	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	4e+04	1e+04
96	9e+03	3e+04	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	4e+04	1e+04
97	7e+03	3e+04	1e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04	4e+04	4e+04	1e+04
98	1e+04	3e+04	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+05	5e+04	4e+04	4e+04	1e+04
99	9e+03	3e+04	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+05	1e+05	3e+04	4e+04	4e+04	1e+04
100	3e+03	3e+04	4e+04	5e+04	6e+04	6e+04	6e+04	6e+04	6e+04	7e+04	6e+04	5e+04	3e+04	2e+04	3e+04	2e+04	5e+03
101		6e+03	8e+03	1e+04	2e+04	2e+04	2e+04	2e+04	2e+04	2e+04	2e+04	1e+04	2e+04	2e+04	7e+03	1e+03	
102					3e+02	2e+02	5e+02	8e+01	7e+02	1e+03	8e+02	6e+02	2e+03	1e+03	6e+02		

Run 0 density in hits per cm³: XZ-plane projection

z [12 mm bins] \ x [mm]	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106
-7				2e+04	3e+04	3e+04	3e+04	3e+04	3e+04	3e+04	2e+04	2e+04	3e+04	1e+03			
-6		2e+04	5e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	4e+04			
-5			4e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	5e+04	8e+03		
-4			2e+04	1e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	8e+04	5e+04	3e+04	1e+04
-3			6e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+04			
-2		4e+03	1e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+05	2e+04			
-1	8e+03	6e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	1e+05	5e+04				
0		2e+04	3e+04	3e+04	3e+04	3e+04	3e+04	3e+04	3e+04	3e+04	3e+04	8e+02					

Figure 4.46: The averaged hit density heat map for the Run 0 data sample is illustrated with the (**Left**) panel showing the frontal projection on the xy-plane and the (**Right**) panel depicting the heatmap projected onto the xz-plane. It is important to note that the density values are expressed as the number of hits per cubic centimetre. However, the dimensions of each heatmap cell are 1 mm x 1 mm x 12 mm, with the latter dimension expanded to accommodate the discrete nature of the hit spatial distribution along the z-axis.

Results

The reconstruction process was performed using a default trained GNN model, and it took about a week. The output was assessed and compared to the FEDRA output using a few criteria: hit utilisation, the track parameter distribution and the visual inspection of the long tracks.

The utilisation of hits has been computed by taking the number of hits consumed during tracking and dividing it by the total hits available in the dataset. FEDRA discards tracks with fewer than four hits to reduce background; similarly, for ANNDEA, tracks under this threshold have their hits marked as unutilised. From the dataset's 2,098,594 available hits, ANNDEA successfully used 540,597 for track reconstruction, resulting in a 25.76% utilisation rate. This value is noticeably higher than FEDRA's, which achieved a 21.57% utilisation on the same data sample.

To evaluate how well ANNDEA can reconstruct tracks with different geometry configurations, the graphs showing track distribution against track length and angle with respect to the z-axis have been plotted for both algorithms and presented below.

A small subset of ANNDEA's longest tracks has been visualised to assess the quality of track reconstruction and identify any anomalies, as illustrated in Figure 4.48 below.

Discussion

The distribution of track lengths in both FEDRA and ANNDEA exhibits a similar pattern that can be divided into three distinct regions: the left region, containing track lengths with four to fourteen hits, the right region, encompassing tracks with 46 to 57 hits, and a middle region that spans the range between. In the left region, short tracks predominated by low-energy electrons are most prevalent, with a significant

Figure 4.47: The distribution of initial tracks derived from ANNDEA and FEDRA versus the number of constituent hits (**Top**) and their azimuthal angle in degrees (**Bottom**). The vertical axis uses a logarithmic scale to represent the number of distinct tracks, and the percentages indicated by coloured values over the lines show the proportion of tracks within the particular bin.

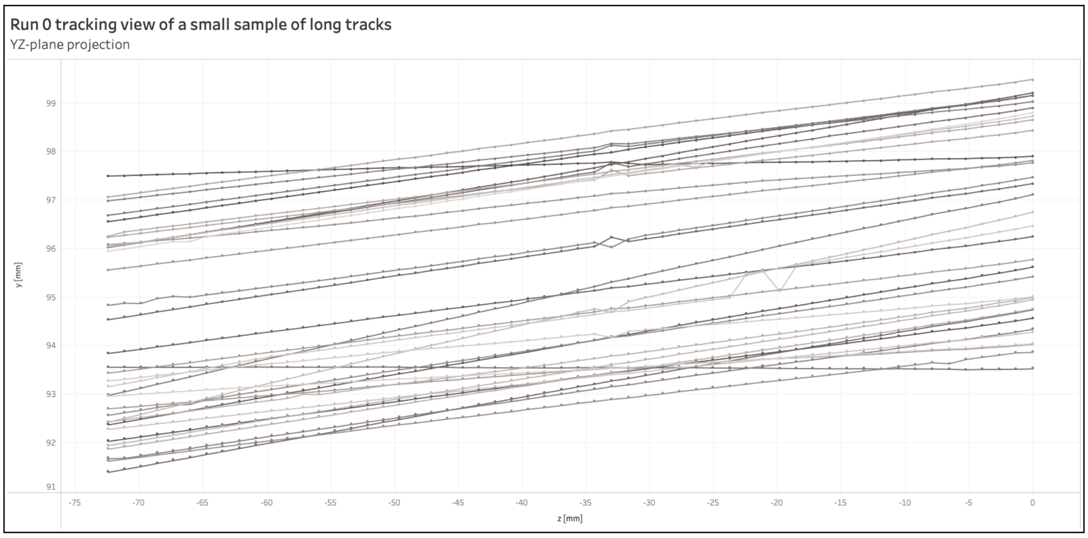


Figure 4.48: A subset of ANNDEA tracks from Run 0 spanning the entire detector.

concentration at 4 hits, representing 16% of all FEDRA and 32% of all ANNDEA tracks, respectively. As the number of hits increases, the number of tracks declines sharply, leading to a more uniform middle region comprising tracks of moderate length. These tracks are likely generated by hadrons, fragments of muon tracks, and cosmic muons penetrating the detector at steep angles. In contrast, the right region predominantly contains IP1 muons that fully traverse the detector, with the width of the peak reflecting the quality of the track reconstruction and the hit detection efficiency.

Although ANNDEA and FEDRA share some similarities in the reconstructed picture, some differences are apparent: ANNDEA's increased utilisation is driven by its ability to reconstruct a significantly higher number of short tracks compared to FEDRA. In terms of reconstructing long muons, ANNDEA has a less prominent peak than FEDRA, which can be attributed to track segmentation issues. This results in a higher proportion of ANNDEA tracks within the 30-44 hits length range, often comprising fragmented muon tracks. This is consistent with simulation results, where a similar pattern is observed: ANNDEA tends to be more skewed towards reconstructing electron and hadron tracks while encountering difficulties with IP1 muons, whereas FEDRA exhibits an opposite bias.

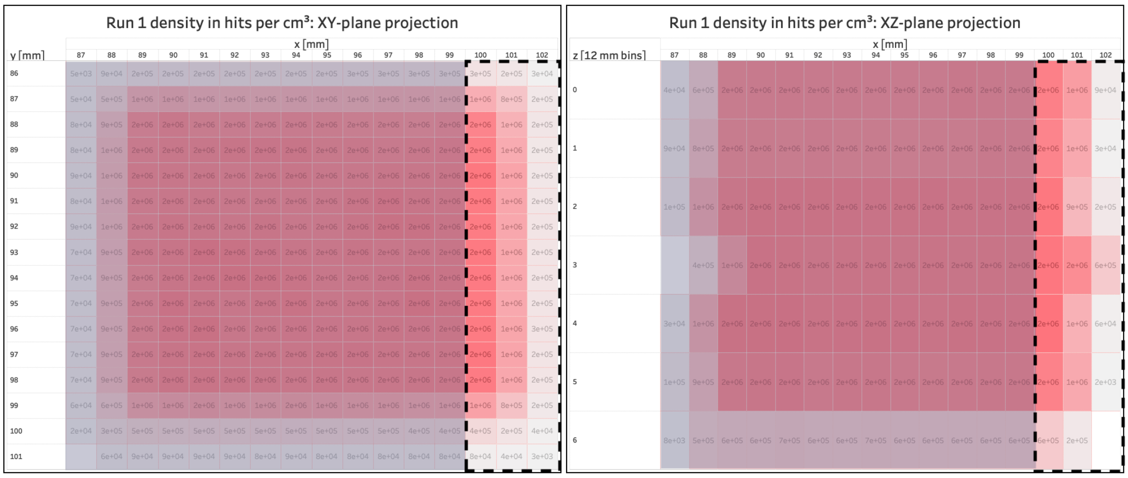
The distribution of track angles can be broadly categorised into three sections, though the boundaries are less distinct in this scenario. The left section contains a peak around $\approx 1^\circ$, associated with muons arriving from the LHC interaction point, which quickly decreases into a relatively flat region covering angles from 10 to 56 degrees. The left side of this section is likely influenced by hadron interactions and the decay products, while the latter portion is primarily due to external backgrounds, such as radiation from surrounding walls and cosmic muons. On the right side of the graph, there is a sharp decline in track numbers as the angle approaches 58°. This decline is not indicative of any physical phenomena but is due to the limitations of scanning and tracking algorithms in reconstructing tracks with steep angles.

When it comes to reconstructing tracks at shallow angles, the FEDRA's and ANNDEA's performance is almost identical. However, ANNDEA demonstrates a clear advantage in handling particles that pass through the detector at steep angles, with a noticeable increase in the number of particles reconstructed in the 10-57 degree range. Although the ability to reconstruct cosmic muon radiation might be of questionable value, this capability proves beneficial for reconstructing interaction products and particle decays with a low Lorentz boost along the z-axis.

The visual inspection indicates that the track morphology is generally healthy, although a series of unusual kinks were detected in the track trajectories between -35 mm and -30 mm. An inquiry with the collaboration team has confirmed that this section experienced significant data quality problems during the scanning of the related plate. Although primarily a cosmetic concern, it is evident that the ANNDEA tracking algorithm is affected by the data quality during scanning, and this should be considered when reconstructing other data sets with similar issues.

4.7.9 Performance on Run 1 data

After conducting the tracking test on the Run 0 dataset, the Run 1 dataset, detailed in 4.3 was addressed. Unfortunately, the tenfold increase in hit density within this sample has posed a substantial computational challenge, rendering all efforts to conduct complete tracking of the sample unsuccessful. Diagnostics swiftly identified the issue: the graph construction process exceeded the maximum wall time permitted by the HTCondor batch system, taking over a week to complete.


Run 1 density in hits per cm³: XY-plane projection

y [mm] \ x [mm]	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
86	5e+03	9e+04	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	2e+05	3e+05	3e+05	3e+05	3e+05	3e+05	2e+05	3e+04
87	5e+04	5e+05	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	8e+05	2e+05
88	8e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
89	8e+04	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
90	9e+04	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
91	8e+04	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
92	9e+04	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
93	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
94	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
95	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
96	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	3e+05
97	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
98	7e+04	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+05
99	6e+04	6e+05	1e+06	1e+06	2e+06	1e+06	1e+06	2e+06	1e+06	1e+06	1e+06	1e+06	1e+06	1e+06	8e+05	2e+05
100	2e+04	3e+05	5e+05	5e+05	5e+05	5e+05	5e+05	5e+05	5e+05	5e+05	5e+05	4e+05	4e+05	4e+05	2e+05	4e+04
101		6e+04	9e+04	9e+04	9e+04	9e+04	8e+04	9e+04	8e+04	8e+04	8e+04	8e+04	8e+04	8e+04	4e+04	3e+03

Run 1 density in hits per cm³: XZ-plane projection

z [12 mm bins] \ x [mm]	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
0	4e+04	6e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	9e+04
1	9e+04	8e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	3e+04
2	1e+05	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	9e+05	2e+05
3		4e+05	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	6e+05
4	3e+04	1e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	6e+04
5	1e+05	9e+05	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	2e+06	1e+06	2e+03
6	8e+03	5e+05	6e+05	6e+05	7e+05	6e+05	6e+05	7e+05	6e+05	6e+05	6e+05	6e+05	6e+05	6e+05	2e+05	

Figure 4.49: The averaged hit density heat map for the Run 1 data sample is illustrated with the (**Left**) panel showing the frontal projection on the xy-plane and the (**Right**) panel depicting the heatmap projected onto the xz-plane. Only the highlighted regions have been processed by ANNDEA, while the shaded areas represent the detector volume that has been omitted during the tracking process.

To mitigate this restriction, a quick workaround was implemented by introducing an extra criterion for the hit seeds to satisfy before engaging in the graph construction process: the z-axis distance cutoff, originally matching the hit cluster's 12,000-micron length, was decreased to 4,000 microns. Although ad-hoc simulation studies indicated a significant decline in track reconstruction quality, this change considerably lowered the graph's interconnectivity and complexity, thereby easing computational demand and reducing graph construction time. Unfortunately, even these drastic measures were insufficient to process the entire sample, resulting in the processing of only a small peripheral subsample measuring 6 mm ×15 mm ×84 mm, as illustrated above.

Results

The calculation for hit utilisation followed a similar method to Run 0. However, this time, a significantly smaller fraction of the data was processed. As a result, FEDRA tracks that did not meet the four-hit criteria had their hits classified as unutilised for an equitable comparison. Out of the entire dataset containing 4,485,287 hits, ANNDEA was able to successfully reconstruct tracks from merely 127,831 hits, yielding a notably lower utilisation rate of 2.85% in contrast to Run 0, despite the fact that nearly twice the number of hits was available. On the other hand, FEDRA showed enhancement, attaining a hit utilisation rate of 26.1%.

To gain a better understanding of why ANNDEA performed poorly, the graphs depicting the track distribution in relation to track length and azimuthal angle for both algorithms have been plotted and visualised in Figure 4.50 below.

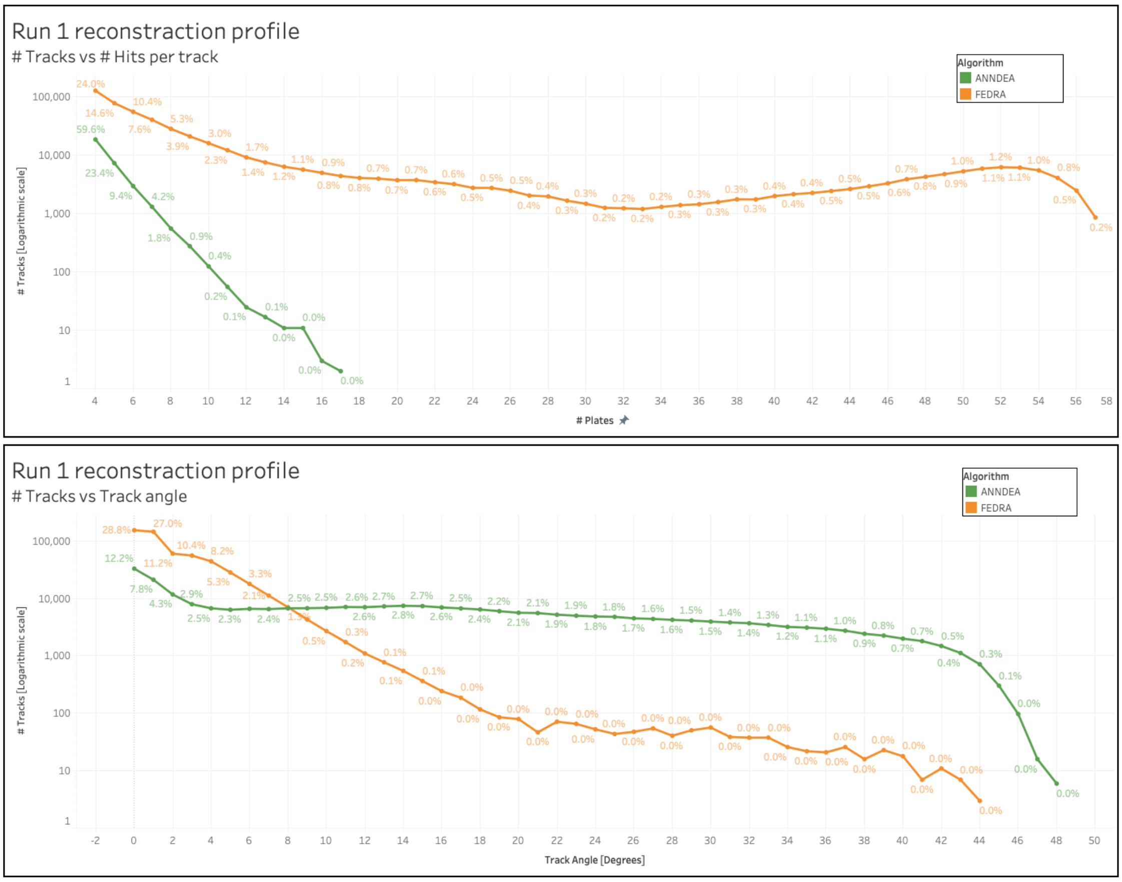


Figure 4.50: The distribution of ANNDEA and FEDRA tracks versus the number of constituent hits (**Top**) and their angle with respect to the z-axis (**Bottom**). The vertical axis uses a logarithmic scale to represent the number of distinct tracks, and the percentages indicated by coloured values over the lines show the proportion of tracks within the particular bin.

Additionally, a small subset of ANNDEA's longest tracks has been visualised to assess the quality of track reconstruction and identify any anomalies, as illustrated below.

Discussion

Although the distribution of FEDRA tracks in terms of track lengths and angles shows a pattern comparable to Run 0, the scenario with ANNDEA is completely different: the distribution reaches a peak at 4 hits per track and rapidly drops to a value of 17, highlighting ANNDEA's inability to reconstruct tracks beyond this length. Analysis of the track angle distribution and visual inspection clearly indicate that the longest tracks reconstructed by ANNDEA are mainly cosmic muons and external radiation crossing the detector at steep angles.

Although ANNDEA struggles significantly with accurately reconstructing muons directly from the LHC, it still demonstrates a substantial advantage over FEDRA in reconstructing tracks at steep angles, with the number of tracks reconstructed at certain angles surpassing FEDRA's by a factor of 100.

A closer examination of the visualised tracks reveals noticeable non-linearity in the reconstructed paths. This can be linked to several factors, including errors in the

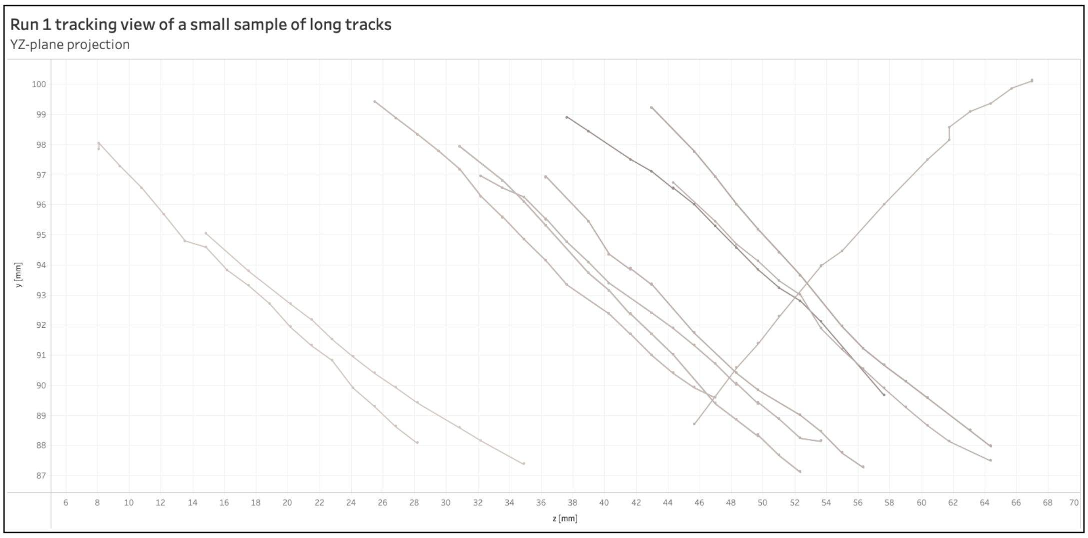


Figure 4.51: A subset of ANNDEAs longest tracks from Run 1 incident at approximately 45 degrees, consistent with cosmic muons to which the ECC brick was briefly exposed during the transportation outside the T118 tunnel.

ANNDEA reconstruction process, flaws in the brick alignment, and potentially the generally lower momentum of the external radiation. However, for cosmic muons, this lower momentum scenario is improbable, given their typically high energies.

4.7.10 Conclusion

Summary

To summarise, this study has aimed at similar goals as outlined in Section 4.4: to improve the precision of the standard method currently used for tracking the emulsion data in the SND@LHC experiment. However, rather than refining the already reconstructed tracks, a more innovative strategy was suggested, which involves tracking the hit data from scratch using geometrical neural networks. The approach included utilising a pre-existing data pipeline identified during the literature review that uses graph edge analysis, originally created for ATLAS, and modifying it to suit the SND@LHC data framework, including adjustments for the specific geometry of the emulsion detector.

To illustrate the flexibility of neural networks when adapted to specific physics processes, two instances of the GNN model were trained: a general model and one that exclusively targets muon and hadron particle tracks. Both setups were assessed using the same control set of simulated data employed in previous studies for comparison purposes. Data processing took approximately one week, and the analysis of results from both models revealed differences and similarities in their reconstructed event pictures. The general GNN model significantly improved the electron reconstruction efficiency from 40% to nearly 70% and enhanced the track quality compared to the standard reconstruction algorithm.

In contrast, the general GNN model exhibited slightly inferior performance compared to FEDRA in terms of muons from signal events, although the number of these events is too small to draw definitive conclusions. The tuned GNN model improved muon reconstruction quality but suffered a substantial reduction in electron reconstruction quality compared to both the general GNN and FEDRA outputs.

Both the general and tuned GNN models demonstrated superior efficiency and quality in hadron particle tracking. A common tendency for both models to fragment muon tracks was also noted.

After conducting simulation experiments, a small sample of scanned Run 0 emulsion data was analysed using a similar approach. The analysis has revealed similar trends to those observed in the simulation. A much higher number of short tracks were reconstructed by ANNDEA compared to the standard tracking procedure, albeit at the cost of higher segmentation of the high-energy muons. Small anomalies in the morphology of the ANNDEA reconstructed tracks have been spotted and attributed to the data quality issues that have arisen during the data acquisition stage.

A data sample from Run 1 was also attempted; however, due to its much higher density, only a portion of the data could be processed using stricter data selection criteria. This led to more simplified and less interconnected graphs for processing by GNNs. The analyses were conducted using a methodology similar to that of Run 0, but the results were quite different. ANNDEA produced significantly fewer tracks and was unsuccessful in reconstructing IP1 muons and long hadrons. Instead, it was more effective in dealing with charged particles traversing the emulsion detector at large angles above 8° with respect to the z-axis.

Interpretation

The studies involving the GNN-enhanced tracking mechanism in the simulation have shown considerable promise of the graph-based approach in reconstructing physics events, especially complex track trajectories like those found in electromagnetic showers, which are typically challenging for conventional tracking algorithms. Although there are specific areas where ANNDEA slightly underperforms compared to FEDRA, such as IP1 muon reconstruction, the performance gap is relatively minor, suggesting that these are likely teething problems that can be addressed with ongoing enhancements. The test on the low-density data sample recorded during Run 0 has demonstrated that, despite challenging data quality, ANNDEA is still capable of demonstrating its strong performance and is less limited than FEDRA in reconstructing particle tracks left by other events that were not taken into account during simulation, such as cosmic muons. However, a few minor yet significant anomalies in the track morphology have also been observed and should not be overlooked.

Unfortunately, the tests on a set with the density that will be anticipated at the actual experimental conditions have highlighted one of the critical issues that arise when analysing graph data: a significant computational resource demand in building the graphs for GNN analysis. The computational requirements for constructing graphs tend to increase cubically with the rise in hit density, and at present, parallelisation is not feasible, thus making graph building a major time-consuming obstacle for the charged particle tracking in emulsion.

Attempts to streamline graph construction by imposing the strictest feasible criteria on hit connections have marginally reduced computational time. Despite these efforts, only partial results have been obtained, exposing ANNDEA's inability to replicate its performance observed in less dense simulations and datasets. Since ANNDEA was not running at its intended algorithm configuration, identifying the precise cause of its subpar performance is a non-trivial exercise.

There are three plausible reasons why it can happen: poor data quality, high density of the data or the result of the cuts, which can be implicitly attributed back to the high-density issue. While the poor data quality certainly had a role in diminishing ANNDEA performance, it is unlikely to be the dominant factor. First of all, Run 0 also had challenging data issues and had higher measured residuals on the long tracks due to the alignment than for the Run 1 data sample, yet the ANNDEA still managed to perform adequate tracking on it.

The second important observation that puts a shadow on the poor data quality theory is the fact that in Run 1, ANNDEA did not have an issue in reconstructing steeply angled tracks with angles exceeding 8° and, in fact, has outperformed FEDRA in this aspect in a similar fashion as in Run 0. This greatly supports the density-related theory: high-angle domains have much less density due to a significantly lower intensity of the related particle sources and since ANNDEA can separate the hits based on their angle, these groups are less challenging to reconstruct. This domain is also less susceptible to the graph restriction measures that have been done purely on the distance along the z-axis component.

Therefore, it can be concluded that high density and the drastic measures that had to be taken to alleviate the computational challenges are to blame; however, which one is the dominant factor is unclear. What can be said with certainty is that the results from the analysis of FEDRA tracks show that the tracking at this density, while challenging, is not fundamentally impossible and that there is a large area for improvement of the ANN-based algorithms.

Further research

In its current form, ANNDEA can be used for tracking; however, it is severely restricted to data sets with the hit densities $\lesssim 10^5$ cm^{-3}, which would make it ideal for low-intensity conditions experienced in the experimental setups such as OPERA. Unfortunately, the typical hit densities observed at the SND@LHC experiment impose significant demands on the computational resources needed for ANNDEA, which are currently beyond the reach of CERN's available infrastructure. Attempts to apply quick fixes by crudely limiting graphs have failed to achieve the desired outcome, necessitating a more thorough revision of the ANNDEA pipeline for sustainable progress. Logs indicate that the bottleneck in the ANNDEA tracking pipeline is the graph creation stage. Once it is completed, the GNN analysis and track-building proceed swiftly, taking merely 10-15 minutes to complete. Thus, to speed up the process, it is beneficial to examine the steps leading to graph creation, particularly hit selection and hit seeding, since these are readily parallelisable.

The hit selection method currently utilised by both FEDRA and ANNDEA involves filtering out basetracks with fewer than 20 silver grain clusters per microtrack pair and a chi-square fit value exceeding 2.4 for two matching microtracks. This strategy aims to discard the background noise: false hits triggered by external influences such as heat, rather than by actual charged particles. While theoretically, these criteria could be altered, determining the precise correlation between these parameters and the likelihood of an incorrect hit activation is quite challenging. The simulations do not go into this level of detail, leaving data-driven methods as the only feasible option. Nonetheless, addressing this uncertainty might not provide a long-term solution, since the primary cause of increased emulsion density is the prolonged exposure of the target to high-intensity muon particles and their secondary effects, including delta-rays and bremsstrahlung, making the fog contribution moderate in comparison.

The seed selection, on the other hand, presents numerous opportunities for further optimisation, although a cautious approach is imperative to prevent depriving the subsequent graph of essential information. In its current setup, ANNDEA employs three primary criteria: the maximum angular difference between two hits, the maximum transverse distance, and the maximum distance along the z-axis. The latter was reduced as a desperate effort to simplify graph complexity, but this simplification adversely affected track reconstruction quality, thus failing to meet the benefit-cost criteria. Analytically, the only sensible course of action would be to adjust the cut on the relative angle between hits as a function of their relative distance rather than applying a constant average value. From a physics perspective, this is logical: the nearer the hits, the smaller the expected angular difference. Nonetheless, implementing this would require simulation studies to determine the optimal function for maximising the SNR, and the impact on reducing extraneous graph complexity is uncertain and might not adequately address the issue.

A more farsighted strategy is to use a neural network to seed the hits. This approach will undoubtedly make the ANNDEA pipeline more complex, but there is a silver lining: the hit seed is a simple object that can be represented by a short vector that encodes relational information, such as spatial distance and relative angle, thus, a simple MLP will be sufficient. Certainly, there is no guarantee that this method will reduce graph complexity to the desired level without considerably compromising the final GNN output. However, since the depth of an MLP can be easily modified, the iterative experiments could at least determine whether such a reduction is theoretically feasible. This technique has been applied in analogous situations[190] involving charged-particle tracking, albeit with non-emulsion detectors.

While tackling the computational challenge is important, there are other issues that need to be resolved, especially the extensive number of separate jobs necessary to process on HTCondor. For a single brick, the number of jobs can surpass 10^4, making their management challenging in practice. Additionally, the task of merging clusters poses issues, as it often results in tracks with multiple hits on the same plate. This creates anomalies that give the tracks an unrealistic "seesaw" pattern.

The ANNDEA is designed to resolve these conflicts in the final stages of the data processing by discarding duplicated hits, but the overall setup is not ideal, and simulation studies have hinted at this being the largest source of the track fragmentation.

A plausible solution to these problems could be an increase in the size of the hit cluster. Ideally, the cluster dimensions should be equivalent to those of the emulsion brick; however, this is impractical due to the prohibitive number of hit seeds that

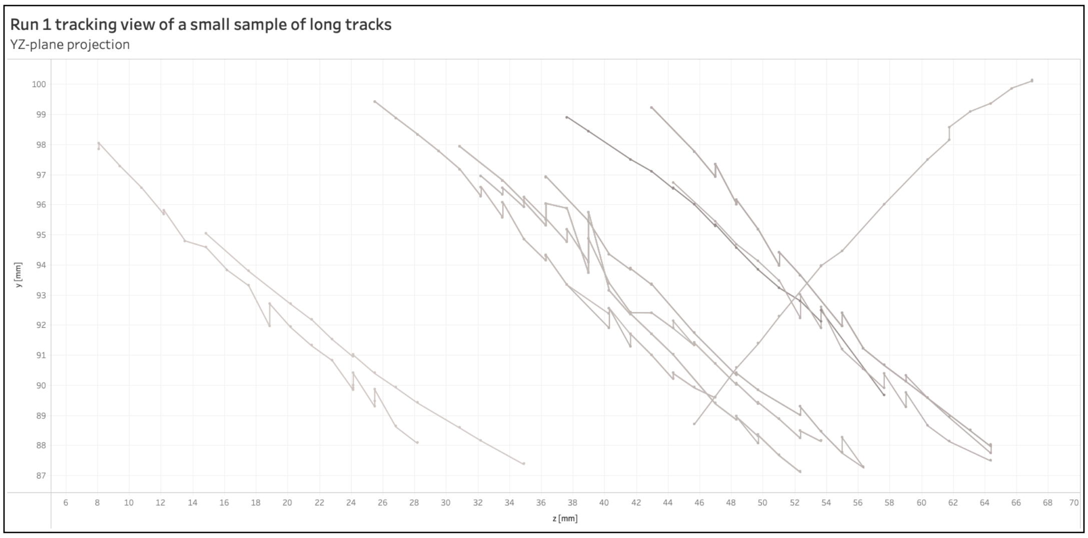


Figure 4.52: A visualised example of reconstructed ANNDEA tracks before applying the corrective measures.

need to be processed. Even FEDRA cannot process an ECC unit containing LHC data in one go, and instead is handled in cross-sectional cells of 1 cm × 1 cm that span the entire length of the ECC unit. For a fair comparison between FEDRA and ANNDEA, the hit cluster size for ANNDEA should be able to process similar dimensions, about 1.5 cm ×1.5 cm × 7.8 cm volume blocks[18]. This adjustment would decrease the number of reconstruction jobs processed on the HTCondor batch farm and minimise instances where clusters need merging, potentially reducing track segmentation.

Unfortunately, enlarging the cluster size would inevitably prolong batch-farm execution times, which, even in the current configuration, exceed the available computational resources. Therefore, the density problem mentioned in the beginning has to be resolved first. However, considering the results that the graph method demonstrated so far on computationally available data, the benefits will likely outweigh this effort.

---

[18] This extra 5 mm is factored in to account for the horizontal and vertical dispersion of the hit coordinates due to the alignment procedure.

Chapter 5

Final Conclusion

This research set out to evaluate the practical benefits of Machine Learning (ML) in high-energy physics, with a particular focus on enhancing discovery potential in the SHiP and SND@LHC experiments at CERN. Through five comprehensive investigations using both simulated and real data, this study has demonstrated that ML holds considerable promise in pushing the boundaries of current data reconstruction and analysis methods.

In the first investigation, evolutionary algorithms proved to be an effective optimisation technique for designing a magnetic shield that minimises background noise in the SHiP detectors. Among all tested configurations, the evolved design yielded the highest background suppression in simulation, showcasing the potential of ML-driven design in experimental setups.

The second study highlighted the power of neural networks in improving the tracking efficiency in the SND@LHC emulsion detector. By intelligently merging previously fragmented track segments, the ML-assisted approach achieved a notable reduction in background tracks: up to 10% more than conventional methods, ultimately enhancing vertex reconstruction accuracy.

In the third investigation, neural networks were successfully applied to leverage the unparalleled resolution of emulsion data, enabling particle classification and momentum estimation from track properties. Although further exploration is warranted, the results affirm the feasibility of using ML for extracting rich, physics-relevant insights from detector data.

The fourth study explored neural networks as an alternative to traditional vertexing algorithms. While no definitive advantage over Kalman filter methods was observed due to limited statistics, the findings suggest that ML-based vertexing is a viable and potentially competitive approach that merits continued investigation.

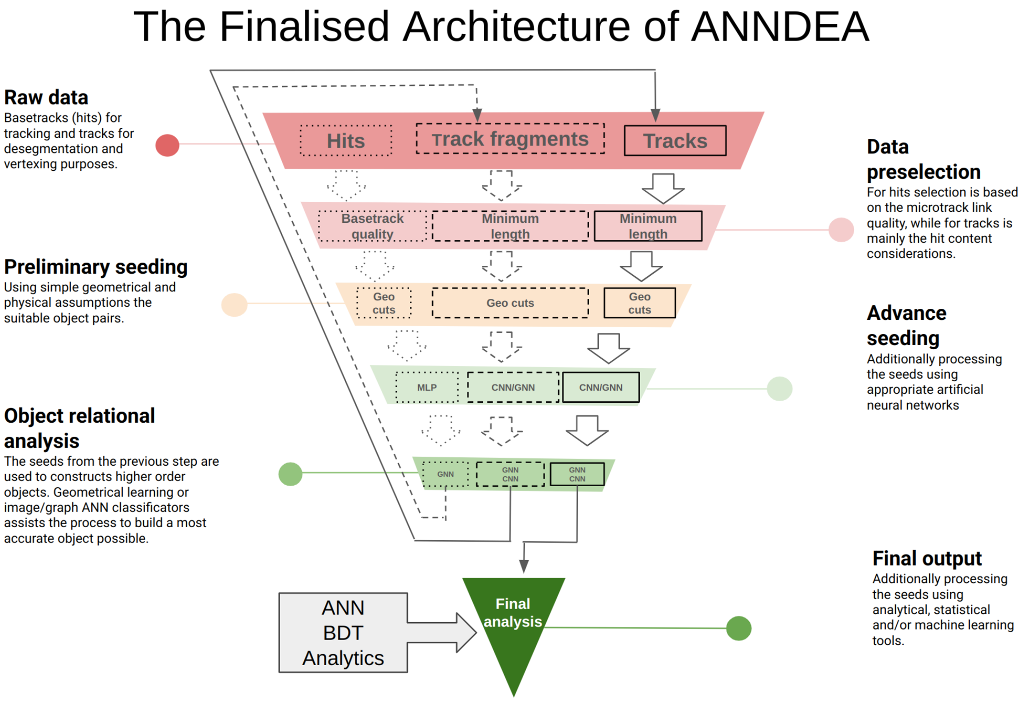
In the fifth and final study, geometric neural networks (GNNs) were applied to one of the most complex tasks: track reconstruction in emulsion data. This innovative method demonstrated compelling advantages, including higher reconstruction efficiency and adaptability to challenging scenarios like electromagnetic showers. These promising outcomes mark a meaningful step forward in developing flexible, ML-based tracking frameworks.

What makes this research particularly unique is its focus on underexplored ML frontiers in particle physics: evolutionary optimisation for detector components and deep learning applications for emulsion data. Collectively, these studies illustrate that ML not only offers theoretical advantages but also delivers practical benefits in terms of precision and data reconstruction quality. The clear message is that the integration of ML into high-energy physics is not only worthwhile – it is essential for continued progress.

Of course, this journey also revealed several practical challenges that currently limit the widespread adoption of ML in particle physics. These include steep learning curves, software maintenance burdens, and the substantial computational demands involved, particularly when dealing with data from emulsion detectors. However, these obstacles are not insurmountable. Instead, they serve as valuable signposts guiding the development of more accessible tools, better data pipelines, and more robust training datasets.

In response to some of these challenges, the ANNDEA framework was developed. This tool represents an important step forward, offering an easier and more intuitive interface for newcomers while automating many of the complex data manipulation steps. ANNDEA embodies the vision of democratising ML access in particle physics and lowering the barrier to entry for students and researchers alike.

This thesis also proposed a path forward by merging two complementary ML approaches within ANNDEA: one enhancing seed selection, and the other applying GNNs for event reconstruction. While neither method performs optimally in isolation, their combination, as illustrated in Figure 5.1, offers a balanced and effective solution for emulsion data reconstruction.


The Finalised Architecture of ANNDEA

Raw data
Basetracks (hits) for tracking and tracks for desegmentation and vertexing purposes.

Data preselection
For hits selection is based on the microtrack link quality, while for tracks is mainly the hit content considerations.

Preliminary seeding
Using simple geometrical and physical assumptions the suitable object pairs.

Advance seeding
Additionally processing the seeds using appropriate artificial neural networks

Object relational analysis
The seeds from the previous step are used to constructs higher order objects. Geometrical learning or image/graph ANN classificators assists the process to build a most accurate object possible.

Final output
Additionally processing the seeds using analytical, statistical and/or machine learning tools.

Figure 5.1: A schematic representation of the data pipeline proposed for the SND@LHC emulsion data reconstruction process.

Two future directions emerge from this research. First, enhancing traditional reconstruction pipelines with ML alternatives within the existing paradigm paves the way for more efficient, accurate physics event analysis. ANNDEA can be viewed as a natural evolution of FEDRA, shifting from analytical logic to adaptive ML algorithms.

Second, there lies exciting potential in exploring radically different approaches. For instance, convolutional neural networks may directly segment raw emulsion data to identify regions of interest without intermediate reconstruction steps. Likewise, integrating ML into the scanning process itself, using methods inspired by 3D medical imaging, could revolutionise how emulsion films are read, reducing bottlenecks and improving data quality.

In conclusion, this thesis affirms that, while challenges remain, the integration of ML into experimental particle physics, especially in novel areas like emulsion data analysis, offers remarkable opportunities. The road ahead is promising, and with continued research and collaboration, machine learning can become a cornerstone in unlocking new physics at the energy and intensity frontier.

Bibliography

[1] ATLAS Collaboration, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Phys. Lett. B*, vol. 716, no. 1, pp. 1–29, 2012.

[2] CMS Collaboration, "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC," *Phys. Lett. B*, vol. 716, no. 1, pp. 30–61, 2012.

[3] S. Weinberg, "The cosmological constant problem," *Rev. Mod. Phys.*, vol. 61, no. 1, pp. 1–23, 1989.

[4] Belle Collaboration, "Observation of large CP violation and evidence for direct CP violation in $B^0 \to \pi^+\pi^-$ decays," *Phys. Rev. Lett.*, vol. 93, no. 2, p. 021601, 2004.

[5] BaBar Collaboration, "Observation of CP violation in B $\to \eta/K^0$ decays," *Phys. Rev. Lett.*, vol. 98, no. 3, p. 031801, 2007.

[6] LHCb Collaboration, "Observation of Several Sources of CP Violation in $B^+ \to \pi^+\pi^+\pi^-$ Decays," *Phys. Rev. Lett.*, vol. 124, no. 3, p. 031801, 2020.

[7] L. Canetti, M. Drewes, and M. Shaposhnikov, "Matter and antimatter in the universe," *New Journal of Physics*, vol. 14, p. 095012, Sept. 2012.

[8] V. C. Rubin, J. Ford, W. K., and N. Thonnard, "Rotational properties of 21 sc galaxies with a large range of luminosities and radii, from ngc 4605 (r=4 kpc) to ugc 2885 (r=122 kpc)," *Astrophys. J.*, vol. 238, pp. 471–487, 1980.

[9] F. Zwicky, "Die rotverschiebung von extragalaktischen nebeln," *Helv. Phys. Acta*, vol. 6, pp. 110–127, 1933.

[10] L. S. Kisslinger and D. Das, "A brief review of dark matter," *Int. J. Mod. Phys. A*, vol. 34, no. 27, p. 1930013, 2019.

[11] R. Davis, D. S. Harmer, and K. C. Hoffman, "Search for neutrinos from the sun," *Phys. Rev. Lett.*, vol. 20, no. 21, pp. 1205–1209, 1968.

[12] The Super-Kamiokande Collaboration, "Evidence for oscillation of atmospheric neutrinos," *Phys. Rev. Lett.*, vol. 81, no. 8, pp. 1562–1567, 1998.

[13] Double Chooz Collaboration, "Indication for the disappearance of reactor $\bar{\nu}_e$ in the double chooz experiment," *Phys. Rev. Lett.*, vol. 108, no. 13, 2012.

[14] Planck Collaboration, "Planck 2013 results. I. Overview of products and scientific results," *Astron. Astrophys.*, vol. 571, p. A1, 2014.

[15] CERN, "CERN LHC website." https://home.web.cern.ch/science/accelerators/large-hadron-collider (accessed on 2023-10-10).

[16] CERN, "Future Circular Collider website." https://home.cern/science/accelerators/future-circular-collider (accessed on 2023-10-10).

[17] E. Bulbul, M. Markevitch, A. Foster, R. K. Smith, M. Loewenstein, and S. W. Randall, "Detection of an unidentified emission line in the stacked x-ray spectrum of galaxy clusters," *Astrophys. J.*, vol. 789, no. 1, p. 13, 2014.

[18] Fermilab, "Fermilab website." https://www.fnal.gov/ (accessed on 2023-10-10).

[19] G. Apollinari, O. Brüning, T. Nakamoto, and L. Rossi, "Chapter 1: High luminosity large hadron collider hl-lhc," *CERN Yellow Report*, no. 5, pp. 1–19, 2015. 21 pages, chapter in High-Luminosity Large Hadron Collider (HL-LHC) Preliminary Design Report.

[20] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3943–3956, 2020.

[21] Google, "Google voice search: faster and more accurate," 2015. https://blog.research.google/2015/09/google-voice-search-faster-and-more.html (accessed on 2023-10-10).

[22] MachineLearning.Apple.com, "An on-device deep neural network for face detection," 2023. https://machinelearning.apple.com/research/face-detection (accessed on 2023-10-10).

[23] SHiP Collaboration, "A facility to search for hidden particles at the cern sps: the ship physics case," *Rep. Prog. Phys.*, vol. 79, no. 12, p. 124201, 2016.

[24] SND@LHC Collaboration, "SND@LHC website." https://home.cern/science/experiments/sndlhc (Accessed on: 2024-07-27).

[25] P. Langacker, *The Standard Model and Beyond.* Taylor and Francis, 2017.

[26] C. W. Fabjan and H. Schopper, *Particle Physics Reference Library, Volume 2: Detectors for Particles and Radiation.* Springer, 2020.

[27] G. Degrassi, S. Di Vita, J. Elias-Miró, J. R. Espinosa, G. F. Giudice, G. Isidori, and A. Strumia, "Higgs mass and vacuum stability in the standard model at NNLO," *J. High Energy Phys.*, vol. 2012, no. 98, pp. 1–32, 2012.

[28] D. Buttazzo, G. Degrassi, P. P. Giardino, G. F. Giudice, F. Sala, A. Salvio, and A. Strumia, "Investigating the near-criticality of the higgs boson," *J. High Energy Phys.*, vol. 12, no. 12, p. 089, 2013. https://doi.org/10.1007/JHEP12(2013)089.

[29] Particle Data Group, "Particle data group website." https://pdg.lbl.gov (Accessed on: 2024-08-14).

[30] Z. Maki, M. Nakagawa, and S. Sakata, "Remarks on the unified model of elementary particles," *Prog. Theor. Phys.*, vol. 28, no. 5, pp. 870–880, 1962. https://doi.org/10.1143/PTP.28.870.

[31] M. Kobayashi and T. Maskawa, "CP-violation in the renormalizable theory of weak interaction," *Prog. Theor. Phys.*, vol. 49, no. 2, pp. 652–657, 1973. https://doi.org/10.1143/PTP.49.652.

[32] F. Curciarello, "Review on dark photon," *EPJ Web Conf.*, vol. 118, p. 01008, 2016.

[33] K. Bondarenko, A. Boyarsky, D. Gorbunov, and O. Ruchayskiy, "Phenomenology of GeV-scale heavy neutral leptons," *J. High Energy Phys.*, vol. 32, pp. 1–46, 2018.

[34] M. Heikinheimo, T. Tenkanen, and K. Tuominen, "WIMP miracle of the second kind," *Phys. Rev. D*, vol. 96, no. 2, p. 023001, 2017.

[35] The GAMBIT Collaboration, "Status of the scalar singlet dark matter model," *Eur. Phys. J. C*, vol. 77, p. 568, 2017.

[36] M. Fabbrichesi, E. Gabrielli, and G. Lanfranchi, *The Physics of the Dark Photon: A Primer*. Springer, 2021.

[37] M. Bauer, P. Foldenauer, P. Reimitz, and T. Plehn, "Light dark matter annihilation and scattering in LHC detectors," *SciPost Phys.*, vol. 10, no. 2, 2021.

[38] T. Asaka and H. Ishida, "Flavour mixing of neutrinos and baryon asymmetry of the universe," *Phys. Lett. B*, vol. 692, no. 2, pp. 105–113, 2010. https://doi.org/10.1016/j.physletb.2010.07.016.

[39] T. Asaka and M. Shaposhnikov, "The νMSM, dark matter and baryon asymmetry of the universe," *Phys. Lett. B*, vol. 620, no. 1–2, pp. 17–26, 2005.

[40] A. Fowlie, "CMSSM naturalness and the "fine-tuning price" of the very large hadron collider," *Phys. Rev. D*, vol. 90, no. 1, 2014.

[41] A. Boyarsky, O. Ruchayskiy, and D. Iakubovskyi, "A lower bound on the mass of dark matter particles," *J. Cosmol. Astropart. Phys.*, vol. 03, no. 005, 2009. https://doi.org/10.1088/1475-7516/2009/03/005.

[42] R. A. Krivonos, V. V. Barinov, A. A. Mukhin, and D. S. Gorbunov, "Strong limits on keV-scale galactic sterile neutrino dark matter with stray light from NuSTAR after 11 years of operation," *Phys. Rev. Lett.*, vol. 133, no. 26, p. 261002, 2024. https://doi.org/10.1103/PhysRevLett.133.261002.

[43] C. Adams, K. Alfonso, C. Andreoiu, E. Angelico, I. J. Arnquist, and J. A. A. t. Asaadi, "Neutrinoless double beta decay," *arXiv:2212.11099*, 2022.

[44] "Super-kamiokande website." https://www-sk.icrr.u-tokyo.ac.jp/en/sk/ (Accessed on: 2024-07-27).

[45] G. Bernardi, G. Carugno, J. Chauveau, F. Dicarlo, M. Dris, J. Dumarchez, M. Ferro-Luzzi, J.-M. Levy, D. Lukas, J.-M. Perreau, Y. Pons, A.-M. Touchard, and F. Vannucci, "Further limits on heavy neutrino couplings," *Phys. Lett. B*, vol. 203, no. 3, pp. 332–334, 1988.

[46] CHARM Collaboration, "A search for decays of heavy neutrinos," *Phys. Lett. B*, vol. 128, no. 5, pp. 361–366, 1983.

[47] NuTeV Collaboration, "Search for neutral heavy leptons in a high-energy neutrino beam," *Phys. Rev. Lett.*, vol. 83, no. 24, pp. 4943–4946, 1999.

[48] L3 Collaboration, "Search for heavy isosinglet neutrinos in e^+e^- annihilation at $130 < \sqrt{s} < 189$ GeV," *Phys. Lett. B*, vol. 461, no. 4, pp. 397–404, 1999.

[49] DELPHI Collaboration, "Search for neutral heavy leptons produced in z decays," *Z. Phys. C*, vol. 74, no. 1, pp. 57–71, 1997.

[50] ATLAS Collaboration, "Search for heavy neutral leptons in decays of w bosons produced in 13 tev pp collisions using prompt and displaced signatures with the atlas detector," *J. High Energy Phys.*, vol. 10, p. 265, 2019.

[51] ATLAS Collaboration, "Search for resonances in diphoton events at $\sqrt{s} = 13$ tev with the ATLAS detector," *J. High Energy Phys.*, vol. 9, p. 001, 2016.

[52] "ATLAS website." `https://atlas.cern/` (Accessed on 2023-10-10).

[53] "CMS website." `https://home.cern/science/experiments/cms` (Accessed on 2023-10-10).

[54] SHiP Collaboration, "SHiP Experiment Progress Report," 2019. Available at: `http://cds.cern.ch/record/2654870/files/SPSC-SR-248.pdf` (accessed on 2023-10-10).

[55] G. Bernardi, G. Carugno, J. Chauveau, F. Dicarlo, M. Dris, J. Dumarchez, M. Ferro-Luzzi, J.-M. Levy, D. Lukas, J.-M. Perreau, Y. Pons, A.-M. Touchard, and F. Vannucci, "Search for neutrino decay," *Phys. Lett. B*, vol. 166, no. 4, pp. 479–483, 1986. `https://www.sciencedirect.com/science/article/pii/0370269386916023`.

[56] CHARM Collaboration, "A search for decays of heavy neutrinos in the mass range 0.5–2.8 GeV," *Phys. Lett. B*, vol. 166, no. 4, pp. 473–478, 1986.

[57] SHiP Collaboration, "Sensitivity of the SHiP experiment to Heavy Neutral Leptons," *J. High Energy Phys.*, vol. 2019, no. 77, p. 16, 2019.

[58] SHiP Collaboration, "BDF/SHiP at the ECN3 high-intensity beam facility," tech. rep., CERN, Geneva, 2022. `https://cds.cern.ch/record/2839677`.

[59] G. D. Lellis, "Neutrino physics at cern," in *Proceedings of NOW 2024 (PoS)*, (Otranto (Lecce), Italy), p. 089, Sept. 2024. `https://inspirehep.net/files/7effa8b4ac226106888b34e4efdef700`.

[60] G. D. Lellis, "Neutrino physics with the SHiP experiment at CERN." Presented at: Neutrino 2024, Milano, Italy, Jan. 2025. Indico contribution.

[61] M. Al-Turany, D. Bertini, R. Karabowicz, D. Kresan, P. Malzacher, T. Stockmanns, and F. Uhlig, "The FairRoot framework," *J. Phys. Conf. Ser.*, vol. 396, no. 2, p. 022001, 2012.

[62] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, "An introduction to PYTHIA 8.2," *Comput. Phys. Commun.*, vol. 191, pp. 159–177, 2015.

[63] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, and D. Zschiesche, "Geant4 a simulation toolkit," *Nucl. Instrum. Methods Phys. Res. A*, vol. 506, pp. 250–303, 2003.

[64] C. Andreopoulos, A. Bell, D. Bhattacharya, F. Cavanna, J. Dobson, S. Dytman, H. Gallagher, P. Guzowski, R. Hatcher, P. Kehayias, A. Meregaglia, D. Naples, G. Pearce, A. Rubbia, M. Whalley, and T. Yang, "The GENIE neutrino monte carlo generator," *Nucl. Instrum. Methods Phys. Res. A*, vol. 614, no. 1, pp. 87–104, 2010.

[65] S. Alekhin, W. Altmannshofer, and e. a. Asaka, "A facility to search for hidden particles at the cern sps: the ship physics case," *Reports on Progress in Physics*, vol. 79, p. 124201, Oct. 2016.

[66] SHiP Collaboration, "The active muon shield in the SHiP experiment," *J. Instrum.*, vol. 12, no. 05, pp. P05011–P05011, 2017.

[67] DONuT Collaboration, "Final tau-neutrino results from the DONuT experiment," *Phys. Rev. D*, vol. 78, no. 5, p. 052002, 2008.

[68] SHiP Collaboration, "SND@LHC," *arXiv preprint: 2002.08722*, 2020.

[69] SND@LHC Collaboration, "Scattering and Neutrino Detector at the LHC," tech. rep., CERN, Geneva, 2020. Available at `https://cds.cern.ch/record/2729015`.

[70] SND@LHC Collaboration, "SND@LHC - Scattering and Neutrino Detector at the LHC," tech. rep., CERN, Geneva, 2021. Available at `https://cds.cern.ch/record/2750060`.

[71] SND@LHC Collaboration, "Observation of Collider Muon Neutrinos with the SND@LHC Experiment," *Phys. Rev. Lett.*, vol. 131, no. 3, p. 031802, 2023.

[72] FASER Collaboration, "Technical Proposal for FASER: ForwArd Search ExpeRiment at the LHC," *arXiv preprint: 1812.09139*, 2018.

[73] FASER Collaboration, "First direct observation of collider neutrinos with faser at the lhc," *Phys. Rev. Lett.*, vol. 131, no. 3, 2023.

[74] OPERA website. `http://operaweb.lngs.infn.it/spip.php?rubrique1`. Accessed on: 2023-10-10.

[75] ROOT website. `https://root.cern/`. Accessed on: 2023-10-10.

[76] DPMJET git repositary. `https://github.com/DPMJET/DPMJET`. Accessed on: 2023-10-10.

[77] SNDSW git repositary. `https://github.com/SND-LHC/sndsw`. Accessed on: 2023-10-10.

[78] W. Barkas, *Nuclear Research Emulsions: techniques and theory*, vol. 1 of *Pure and applied physics*. Academic Press, 1963.

[79] A. Ariga, T. Ariga, G. De Lellis, A. Ereditato, and K. Niwa, *Nuclear Emulsions*, pp. 383–438. Springer International Publishing, 2020.

[80] A. J. Herz and W. O. Lock, "Particle detectors. 1. Nuclear Emulsions," 1966.

[81] H. Becquerel, "On the rays emitted by phosphorescence," *C. R. Acad. Sci.*, vol. 122, pp. 501–503, 1896.

[82] C. Powell, "The use of stripped emulsions for recording the tracks of charged particles," *Phil. Mag.*, vol. 44, no. 349, pp. 219–222, 1953.

[83] K. Hoshino et al., "Proceedings of International Cosmic Ray Symposium of High Energy Phenomena (Tokyo 1974)," vol. 149, 1974.

[84] INFN Bologna website. `https://www.bo.infn.it/gruppo1/en/snd-experiment/`. Accessed on: 2023-10-10.

[85] Lebedev Institute website. `https://lebedev.ru/`. Accessed on: 2023-10-10.

[86] Università degli Studi di Napoli Federico II. `https://unina.it`. Accessed on: 2023-10-10.

[87] L. Arrabito, C. Bozza, S. Buontempo, L. Consiglio, M. Cozzi, N. D'Ambrosio, G. De Lellis, M. Serio, F. Di Capua, D. Di Ferdinando, N. Di Marco, A. Ereditato, L. S. Esposito, R. Fini, G. Giacomelli, M. Giorgini, G. Grella, M. Ieva, J. Janicsko, and T. Waelchli, "Track reconstruction in the emulsion-lead target of the opera experiment using the ess microscope," *J. Instrum.*, vol. 2, 2007.

[88] V. Tioukov, I. Kreslo, Y. Petukhov, and G. Sirri, "The FEDRA Framework for emulsion data reconstruction and analysis in the OPERA experiment," *Nucl. Instrum. Methods Phys. Res. A*, vol. 559, no. 1, pp. 103–105, 2006.

[89] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. Series D, pp. 35–45, 1960.

[90] "Measurement of the multi-TeV neutrino interaction cross-section with IceCube using Earth absorption," *Nature*, vol. 551, no. 7682, pp. 596–600, 2017.

[91] E. Khalikov and the SND@LHC Collaboration, "Recent results from snd@lhc," *arXiv*, 2024. Available at `https://arxiv.org/abs/2407.09241`.

[92] M. Bustamante and A. Connolly, "Extracting the Energy-Dependent Neutrino-Nucleon Cross Section above 10 TeV Using IceCube Showers," *Phys. Rev. Lett.*, vol. 122, no. 4, 2019.

[93] S. Buontempo, G. M. Dallavalle, G. De Lellis, D. Lazic, and F. L. Navarria, "CMS-XSEN: LHC Neutrinos at CMS. Experiment Feasibility Study," *arXiv preprint: 1804.04413*, 2018.

[94] M. Dittmar, "Testing the tau lepton universality at LEP," *CERN CDS*, pp. 45–51, 1991. Available at `https://cds.cern.ch/record/2624905`.

[95] The ATLAS Collaboration, "Precise test of lepton flavour universality in w-boson decays into muons and electrons in pp collisions at $\sqrt{s} = 13TeV$ with the atlas detector," *The European Physical Journal C*, vol. 84, no. 10, 2024. Available at `http://dx.doi.org/10.1140/epjc/s10052-024-13070-4`.

[96] The ATLAS Collaboration, “Test of lepton flavour universality in W-boson decays into electrons and tau leptons using pp collisions at $\sqrt{s} = 13 TeV$ with the ATLAS detector,” *Journal of High Energy Physics*, vol. 2025, no. 5, 2025. Available at http://dx.doi.org/10.1007/JHEP05(2025)038.

[97] R. D. Moise, “Test of lepton flavour universality in $b \to s\ell^+\ell^-$ decays,” 2021. Available at https://arxiv.org/abs/2105.06921.

[98] L. Collaboration, “Test of lepton flavour universality in $b \to s\ell^+\ell^-$ decays,” *Physical Review Letters*, vol. 131, no. 5, 2023. Available at http://dx.doi.org/10.1103/PhysRevLett.131.051803.

[99] CHARM Collaboration, “A precise determination of the electroweak mixing angle from semileptonic neutrino scattering,” *Z. Phys. C: Part.*, vol. 36, no. 4, pp. 611–628, 1987.

[100] MiniBooNE Collaboration, “Dark matter search in nucleon, pion, and electron channels from a proton beam dump with MiniBooNE,” *Phys. Rev. D*, vol. 98, no. 11, 2018.

[101] NA64 Collaboration, “Dark Matter Search in Missing Energy Events with NA64,” *Phys. Rev. Lett.*, vol. 123, no. 12, 2019.

[102] D. Crevier, *AI: The Tumultuous History of the Search for Artificial Intelligence*. BasicBooks, 1993.

[103] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959.

[104] M. Srikrishna, J. B. Pereira, R. A. Heckemann, G. Volpe, D. van Westen, A. Zettergren, S. Kern, L.-O. Wahlund, E. Westman, I. Skoog, and M. Schöll, “Deep learning from MRI-derived labels enables automatic brain tissue classification on human brain CT,” *Neuroimage*, vol. 244, p. 118606, 2021.

[105] Asos, “Asos launches chatbot.” https://www.drapersonline.com/news/asos-launches-chatbot (accessed on 2023-10-10).

[106] N. Laptev, S. Smyl, and S. Shanmugam, “Engineering extreme event forecasting at Uber with recurrent neural networks,” 2017. https://www.uber.com/en-GB/blog/neural-networks/ (accessed on 2023-10-10).

[107] I. K. Nti, A. F. Adekoya, B. A. Weyori, and O. Nyarko-Boateng, “Applications of artificial intelligence in engineering and manufacturing: A systematic review,” *J. Intell. Manuf.*, vol. 33, no. 6, pp. 1581–1601, 2022.

[108] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[109] B. Saavedra-Moreno, S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto, and A. Portilla-Figueras, “Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms,” *Renew. Energ.*, vol. 36, no. 11, pp. 2838–2844, 2011.

[110] W. P. Worzel, J. Yu, A. A. Almal, and A. M. Chinnaiyan, "Applications of genetic programming in cancer research," *Int. J. Biochem. Cell Biol.*, vol. 41, no. 2, pp. 405–413, 2009.

[111] W. Yan, J. Sun, Z. Liu, and Y. Hu, "A novel bi-subgroup adaptive evolutionary algorithm for optimizing degree of hybridization of HEV bus," *Cluster comput.*, vol. 20, no. 1, pp. 497–505, 2017.

[112] L. Jiang, Y. Li, Y. Liu, and C. Chen, "Traffic signal light control model based on evolutionary programming algorithm optimization bp neural network," in *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 564–567, IEEE, 2017.

[113] A. J. Keane, "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques," *J. Acoust. Soc. Am.*, vol. 99, no. 4, pp. 2599–2603, 1996.

[114] G. Hornby, A. Globus, D. Linden, and J. Lohn, "Automated antenna design with evolutionary algorithms," *Collection of Technical Papers - Space 2006 Conference*, vol. 1, 2006.

[115] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, 1959.

[116] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets* (S.-i. Amari and M. A. Arbib, eds.), (Berlin, Heidelberg), pp. 267–285, 1982.

[117] Tech Power Up, "Gpu specs database," 2023.

[118] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[119] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc IEEE*, vol. 86, pp. 2278–2324, 12 1998.

[120] K. Chellapilla, S. Puri, and P. Simard, "High Performance Convolutional Neural Networks for Document Processing," in *Tenth International Workshop on Frontiers in Handwriting Recognition* (G. Lorette, ed.), (La Baule, France), Université de Rennes 1, Suvisoft, Oct 2006. PDF: https://inria.hal.science/inria-00112631/file/p1038112283956.pdf.

[121] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[122] A. Leman and B. Weisfeiler, "The reduction of a graph to canonical form and the algebra which appears therein," 1968.

[123] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 20, no. 1, pp. 61–80, 2009.

[124] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," *Distill*, 2021. `https://distill.pub/2021/gnn-intro`.

[125] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[126] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 20, no. 1, pp. 61–80, 2009.

[127] D. Grattarola, D. Zambon, F. M. Bianchi, and C. Alippi, "Understanding Pooling in Graph Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–11, 2022.

[128] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint: 1207.0580*, 2012.

[129] Y. Li, W. Ma, C. Chen, M. Zhang, Y. Liu, S. Ma, and Y. Yang, "A Survey on Dropout Methods and Experimental Verification in Recommendation," *arXiv preprint: 2204.02027*, 2022.

[130] O. Lantwin, *Optimisation of the SHiP experimental design.* PhD thesis, Imperial College London, 2019. Available at: `https://cds.cern.ch/record/2693177`.

[131] D. Ginsbourger, R. L. Riche, and L. Carraro, "Kriging is well-suited to parallelize optimization," in *Computational Intelligence in Expensive Optimization Problems*, vol. 2, pp. 131–162, Springer Berlin Heidelberg, 2010.

[132] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 185, pp. 1–52, 2018.

[133] Y. Ren and Y. Wu, "An efficient algorithm for high-dimensional function optimization," *Soft Comput.*, vol. 17, pp. 995–1004, 2013.

[134] V. Bayliss, M. Courthold, and T. Rawlings, "Active muon shield - preliminary report." `https://cds.cern.ch/record/2027801/files/2015_06_MuonShield_design.pdf`, 2015. Accessed: 2023-04-20.

[135] P. Gorbunov, "On the field shape of the SHiP Active Muon Shield," 2017. SHiP Memo: Available on request.

[136] P. Hynek, V. Kreibich, and R. Firt, "Suitable production tools selection with the use of evolutionary algorithms," *Acta Polytech.*, vol. 60, pp. 56–64, 2020.

[137] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.

[138] T. Bäck, "Optimal mutation rates in genetic search," in *Proceedings of the 5th International Conference on Genetic Algorithms*, p. 28, Morgan Kaufmann Publishers Inc., 1993.

[139] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.

[140] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, 1999.

[141] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 16, no. 1, pp. 122–128, 1986.

[142] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, 2012.

[143] S. Shirobokov, V. Belavin, M. Kagan, A. Ustyuzhanin, and A. G. Baydin, "Black-box optimization with local generative surrogates," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 14650–14662, Curran Associates, Inc., 2020. Available at: `https://proceedings.neurips.cc/paper_files/paper/2020/file/a878dbebc902328b41dbf02aa87abb58-Paper.pdf`.

[144] M. M. G. Jr. and A. F. R. Araújo, "Diversity-based adaptive evolutionary algorithms," in *New Achievements in Evolutionary Computation*, ch. 1, IntechOpen, 2010.

[145] SHiP Collaboration, "Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks," *J. Instrum.*, vol. 14, no. 11, pp. P11028–P11028, 2019.

[146] DELPHI Collaboration, "Performance of the DELPHI detector," *Nucl. Instrum. Methods Phys. Res. B*, vol. 378, no. 1, pp. 57–100, 1996.

[147] X. Ai, H. M. Gray, A. Salzburger, and N. Styles, "A non-linear Kalman filter for track parameters estimation in high energy physics," *Nucl. Instrum. Methods Phys. Res. A*, vol. 1049, p. 168041, 2023.

[148] J.-C. József, *Study of interactions and background estimation in the OPERA Emulsion Film Detector.* PhD thesis, University of Neuchâtel, 2006. Available at `https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b9838bd9a8ce5e641dfef5acde555d1bd85b4238`.

[149] CHORUS Collaboration, "The CHORUS experiment to search for $\nu_\mu \to \nu_\tau$ oscillation," *Nucl. Instrum. Methods Phys. Res. A*, vol. 401, no. 1, pp. 7–44, 1997.

[150] DONuT Collaboration, "Final tau-neutrino results from the DONuT experiment," *Phys. Rev. D*, vol. 78, no. 5, 2008.

[151] DONuT Collaboration, "Momentum measurement of secondary particle by multiple coulomb scattering with emulsion cloud chamber in DONuT experiment," *Nucl. Instrum. Methods Phys. Res. A*, vol. 574, no. 1, pp. 192–198, 2007.

[152] A. Eide, T. Lindblad, T. Lindén, and C. S. Lindsey, "Neural networks for tracking," *Nucl. Instrum. Methods Phys. Res. A*, vol. 368, no. 3, pp. 855–858, 1996.

[153] P. Calafiura and S. Farrell, "HEP advanced tracking algorithms with cross-cutting applications (Project HEP.TrkX)," 2018. Available at: `https://heptrkx.github.io`.

[154] S. Farrell, D. Anderson, P. Calafiura, G. Cerati, L. Gray, J. Kowalkowski, M. Mudigonda, Prabhat, P. Spentzouris, M. Spiropoulou, A. Tsaris, J.-R. Vlimant, and S. Zheng, "The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking," *EPJ Web of Conferences (Online)*, vol. 150, 8 2017.

[155] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J.-R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati, L. Gray, J. Kowalkowski, P. Spentzouris, and A. Tsaris, "Novel deep learning methods for track reconstruction," *arXiv preprint: 1810.06111*, 2018.

[156] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[157] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertain. Fuzz.*, vol. 6, pp. 107–116, 04 1998.

[158] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[159] D. Baranov, S. Mitsyn, P. Goncharov, and G. Ososkov, "The Particle Track Reconstruction based on deep Neural networks," *EPJ Web Conf.*, vol. 214, 2019.

[160] G. DeZoort, S. Thais, J. Duarte, V. Razavimaleki, M. Atkinson, I. Ojalvo, M. Neubauer, and P. Elmer, "Charged Particle Tracking via Edge-Classifying Interaction Networks," *Comput. Vis. Sci.*, vol. 5, no. 1, 2021.

[161] A. Elabd, V. Razavimaleki, S.-Y. Huang, J. Duarte, M. Atkinson, G. DeZoort, P. Elmer, S. Hauck, J.-X. Hu, S.-C. Hsu, B.-C. Lai, M. Neubauer, I. Ojalvo, S. Thais, and M. Trahms, "Graph neural networks for charged particle tracking on fpgas," *Front. Big Data*, vol. 5, 2022.

[162] M. Andersson and T. Bäck, "Gamma-ray track reconstruction using graph neural networks," *Nucl. Instrum. Methods Phys. Res. A*, vol. 1048, p. 168000, 2023.

[163] X. Ju, D. Murnane, P. Calafiura, N. Choma, S. Conlon, S. Farrell, Y. Xu, M. Spiropulu, J.-R. Vlimant, A. Aurisano, J. Hewes, G. Cerati, L. Gray, T. Klijnsma, J. Kowalkowski, M. Atkinson, M. Neubauer, G. DeZoort, S. Thais, A. Chauhan, A. Schuy, S.-C. Hsu, A. Ballow, and A. Lazar, "Performance of a geometric deep learning pipeline for hl-lhc particle tracking," *Eur. Phys. J. C*, vol. 81, no. 10, 2021.

[164] S. Thais, P. Calafiura, G. Chachamis, G. DeZoort, J. Duarte, S. Ganguly,

M. Kagan, D. Murnane, M. Neubauer, and K. Terao, "Graph neural networks in particle physics: Implementations, innovations, and challenges," *arXiv preprint: 2203.12852*, 2022.

[165] M. Abdughani, J. Ren, L. Wu, and J. M. Yang, "Probing stop pair production at the lhc with graph neural networks," *J. High Energy Phys.*, vol. 2019, no. 8, 2019.

[166] J. Ren, L. Wu, and J. M. Yang, "Unveiling cp property of top-higgs coupling with graph neural networks at the lhc," *Phys. Lett. B*, vol. 802, p. 135198, 2020.

[167] The ATLAS Collaboration, "Search for dark matter produced in association with a higgs boson decaying to tau leptons at $\sqrt{s} = 13$ tev with the atlas detector," *J. High Energy Phys.*, vol. 2023, no. 9, 2023.

[168] P. Calafiura and S. Farrell, "HEP advanced tracking algorithms at the exascale (Project Exa.TrkX)," 2020. Available at: `https://exatrkx.github.io/`.

[169] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[170] T. Li, Y. Chen, S. Wang, K. Han, H. Lin, K. Ni, and W. Wang, "Reconstruction of the event vertex in the pandax-iii experiment with convolution neural network," *J. High Energy Phys.*, vol. 2023, no. 5, 2023.

[171] NEXT collaboration, "Demonstration of background rejection using deep convolutional neural networks in the next experiment," *J. High Energy Phys.*, vol. 2021, no. 1, 2021.

[172] K. Goto, T. Suehara, T. Yoshioka, M. Kurata, H. Nagahara, Y. Nakashima, N. Takemura, and M. Iwasaki, "Development of a vertex finding algorithm using recurrent neural network," *Nucl. Instrum. Methods Phys. Res. A*, vol. 1047, p. 167836, 2023.

[173] T. Suehara and T. Tanabe, "LCFIPlus: A framework for jet analysis in linear collider studies," *Nucl. Instrum. Methods Phys. Res. A*, vol. 808, pp. 109–116, 2016.

[174] K. Albertsson and F. Meloni, "Towards fast displaced vertex finding," *arXiv preprint: 1910.10508*, 2019.

[175] OPERA Collaboration, "More results from the opera experiment," *Journal of Physics: Conference Series*, vol. 1342, no. 1, p. 012036, 2020.

[176] "Machine learning for electromagnetic showers reconstruction in emulsion cloud chambers," *J. Phys. Conf. Ser.*, vol. 1085, no. 4, pp. 42025–, 2018.

[177] R. L. Workman and Others, "Review of particle physics," *PTEP*, vol. 2022, p. 083C01, 2022.

[178] CERN, "Lxplus service." `https://abpcomputing.web.cern.ch/computing_resources/lxplus/`. Accessed on: 2024-01-27.

[179] CERN, "Htcondor batch system." `https://abpcomputing.web.cern.ch/computing_resources/cernbatch/`. Accessed on: 2024-01-27.

[180] CERN, “AFS Service.” https://cern.service-now.com/service-portal?id=service_element&name=afs-service. Accessed on: 2024-01-27.

[181] CERN, “EOS Open Storage.” https://eos-web.web.cern.ch/eos-web/. Accessed on: 2024-01-27.

[182] CERN, “Collection point for AFS Phaseout material.” https://twiki.cern.ch/twiki/bin/viewauth/IT/AfsPhaseout. Accessed on: 2024-01-27.

[183] L. W. D. B. F. Fedotovs, W. Xie and H. Wilson, “Anndea.” https://github.com/FilipsFedotovs/ANNDEA, 2023.

[184] F. Fedotovs, “Anndea wiki.” https://github.com/FilipsFedotovs/ANNDEA/wiki, 2023.

[185] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, “Geometric deep learning on graphs and manifolds using mixture model cnns,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5425–5434, IEEE, 2017.

[186] A. Salvucci, “Measurement of muon momentum resolution of the atlas detector,” *EPJ Web Conf.*, vol. 28, p. 12039, 2012.

[187] G. DeZoort, “Gnns for tracking.” https://github.com/GageDeZoort/gnns-for-tracking, 2013.

[188] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and K. Kavukcuoglu, “Interaction networks for learning about objects, relations and physics,” in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

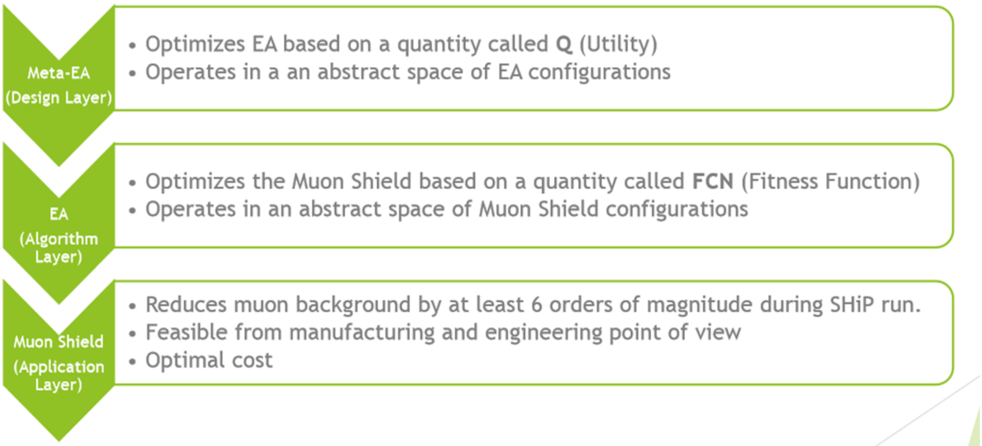
[189] SND@LHC collaboration, “Emulsion targets.” https://twiki.cern.ch/twiki/bin/view/SndLHC/EmuTarget, 2025.

[190] S. Van Stroud, P. Duckett, M. Hart, N. Pond, S. Rettie, G. Facini, and T. Scanlon, “Transformers for charged particle track reconstruction in high energy physics.” https://arxiv.org/pdf/2411.07149, 2024.

Appendix A

Finding the right EA architecture

Here, an arbitrarily chosen evolutionary algorithm is used to arrive at the best EA configuration for this particular problem. The whole procedure can be illustrated as:



For hyper-tuning studies, a reduced search space is used. The typical AMS full genetic representation with 42 parameters is taken, and 6 of them are chosen (in bold), while the rest are set to a constant value:

[**200**;170;**180**;207;**185**;248;305;242;**10**;40;150;150;2;2;80;80;150;150;2;2;**15**;51;29;46;10;7;54;38;46;192;14;9;**25**;31;35;31;51;11;3;32;54;24].

The setup can be simplified to:

$$[200;\ 180;\ 185;\ 10;\ 15;\ 25].$$

For easier processing by the evolution algorithm, the representation is changed using the formula:

$$[\frac{(x_1 - 175)}{5}; \frac{(x_2 - 175)}{5}; \frac{(x_3 - 175)}{5}; \frac{x_4}{5}; \frac{x_5}{5}; \frac{x_6}{5}] \tag{A.1}$$

This can now be expressed as [5; 1; 2; 2; 3; 5]. By restricting a range of values to (1-5) for each of the six parameters, a subset of search space with a size of 5^6 or 15625 points is obtained.

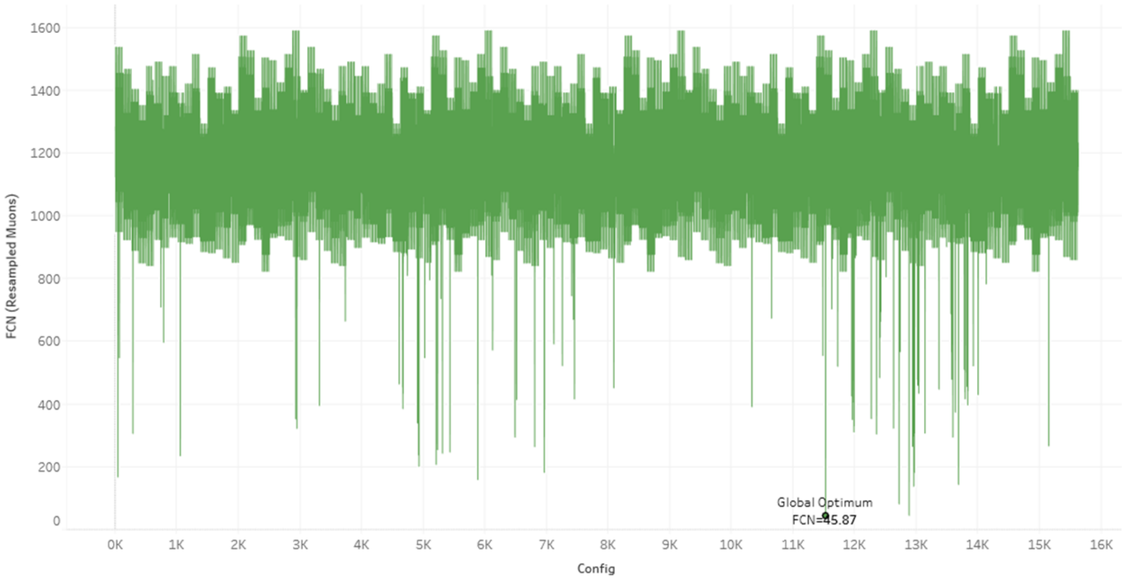


Figure A.1: The evaluated reduced space of muon shield configurations with fitness (or FCN) on the vertical axis and AMS configuration index number on the horizontal axis. The Global Optima (the point with the lowest fitness in the subset) is labelled with a value of 45.87.

By using a specially created submission script and the HTCondor batch service, each point in the reduced subspace is evaluated, resulting in the complete data set.

The next step is to run optimisation cycles using a simple Random Search Algorithm (RSA). The objective is to establish a reliable foundation against which to compare and evaluate EA settings. The RSA algorithm works as follows: RSA picks a random AMS configuration point and evaluates it using the data file. Every evaluation costs RSA one iteration point, but the algorithm is not penalised for evaluating the same point twice. The RSA continues to evaluate muon shield configurations one by one until it finds the global optimum. For better statistics, the whole procedure is repeated 100 times.

Once the RSA search profile is obtained, the meta-EA optimisation process begins:

1. The initial population of EA models with random configurations is generated.
2. Every EA individual runs 10 times over the data. Each of those 10 runs is capped to a maximum of 8041 iterations (the average number of iterations it takes for a random search algorithm to find GOP).
3. After each of those runs, a quantity called Q is evaluated, which is the utility of each EA model. It is a multi-objective metric that measures both EA performance versus RSA and its consistency in delivering those results from run to run.
4. Metagenetic operators are performed on the top selected EA configurations.

5. Steps 3 to 5 repeat until no further progress is made.

After several days of the meta-optimisation process, the run has been suspended and the top ten EA models have been re-evaluated by using 100 runs each to calculate a more accurate utility. Eventually, the top EA architecture has been selected for the muon shield optimisation process with little modifications to a mutation operator to accommodate the full 42-dimension vector space.

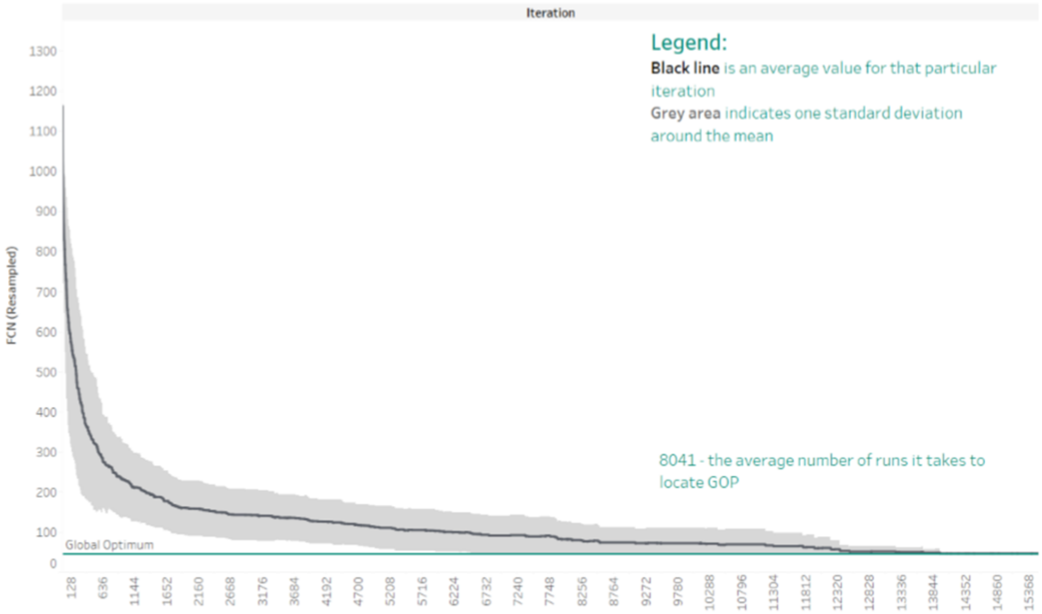


Figure A.2: The RSA search profile indicates an average best solution found at the evaluation iteration.

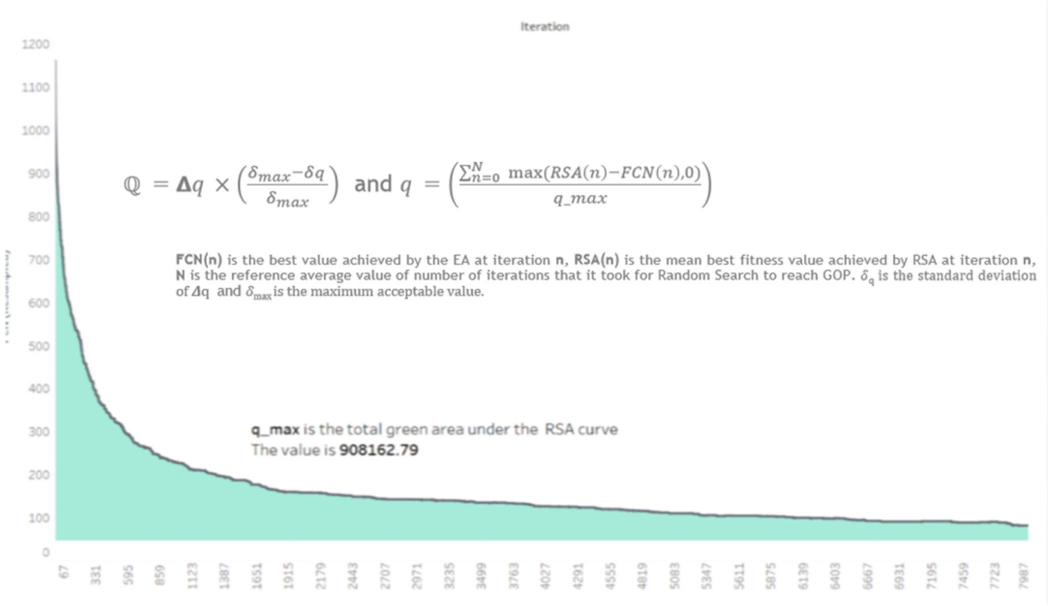
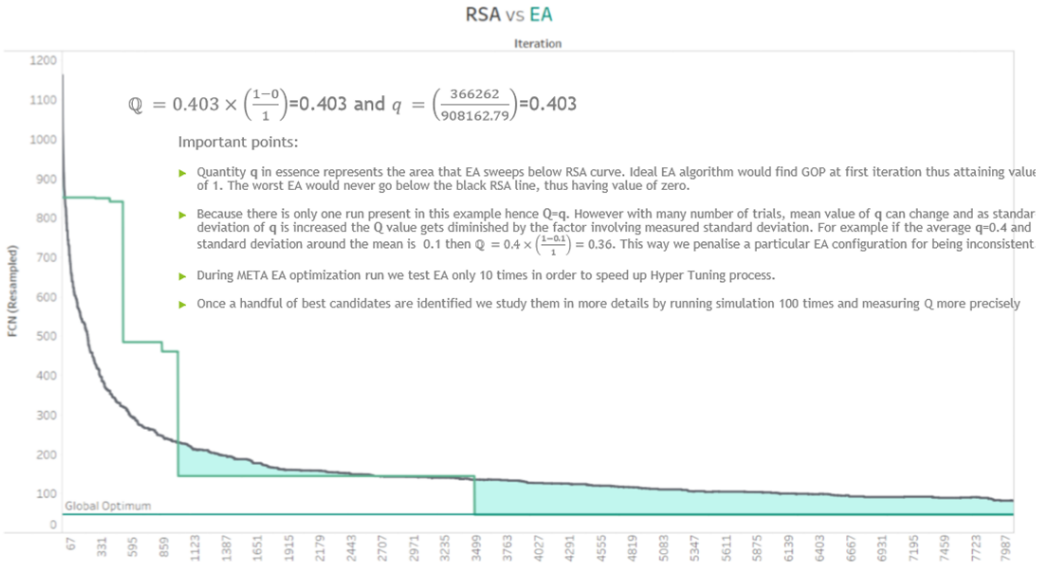


Figure A.3: The utility calculation methodology description.

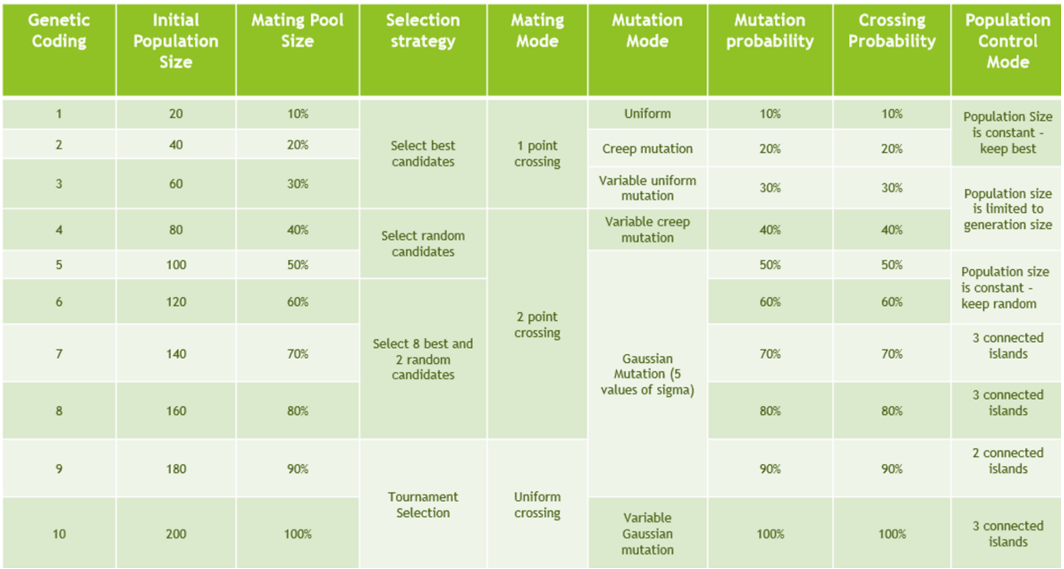


$\mathbb{Q} = 0.403 \times \left(\frac{1-0}{1}\right)$=0.403 and $q = \left(\frac{366262}{908162.79}\right)$=0.403

Important points:

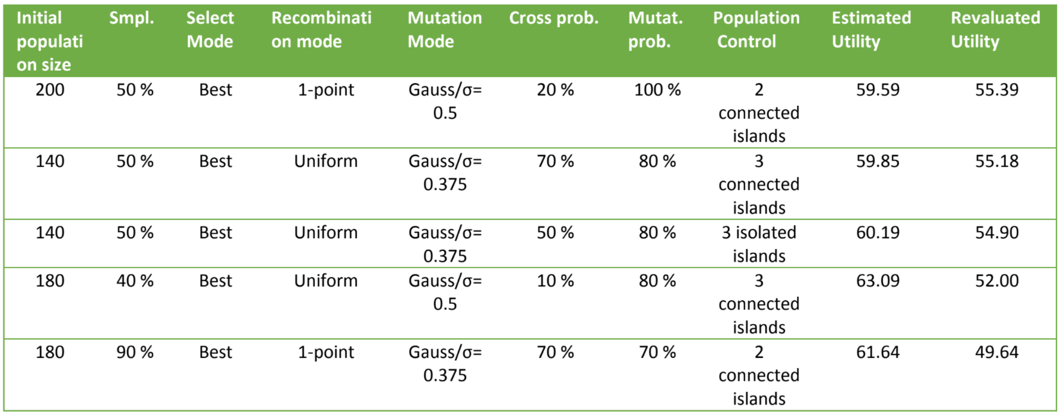
- Quantity **q** in essence represents the area that EA sweeps below RSA curve. Ideal EA algorithm would find GOP at first iteration thus attaining value of 1. The worst EA would never go below the black RSA line, thus having value of zero.
- Because there is only one run present in this example hence Q=**q**. However with many number of trials, mean value of **q** can change and as standar deviation of **q** is increased the Q value gets diminished by the factor involving measured standard deviation. For example if the average **q**=0.4 and standard deviation around the mean is 0.1 then $\mathbb{Q} = 0.4 \times \left(\frac{1-0.1}{1}\right) = 0.36$. This way we penalise a particular EA configuration for being inconsistent
- During META EA optimization run we test EA only 10 times in order to speed up Hyper Tuning process.
- Once a handful of best candidates are identified we study them in more details by running simulation 100 times and measuring Q more precisely

Figure A.4: An example of the utility calculation.



Genetic Coding	Initial Population Size	Mating Pool Size	Selection strategy	Mating Mode	Mutation Mode	Mutation probability	Crossing Probability	Population Control Mode
1	20	10%	Select best candidates	1 point crossing	Uniform	10%	10%	Population Size is constant - keep best
2	40	20%			Creep mutation	20%	20%	
3	60	30%			Variable uniform mutation	30%	30%	Population size is limited to generation size
4	80	40%	Select random candidates	2 point crossing	Variable creep mutation	40%	40%	
5	100	50%			Gaussian Mutation (5 values of sigma)	50%	50%	Population size is constant - keep random
6	120	60%	Select 8 best and 2 random candidates			60%	60%	
7	140	70%				70%	70%	3 connected islands
8	160	80%				80%	80%	3 connected islands
9	180	90%	Tournament Selection	Uniform crossing		90%	90%	2 connected islands
10	200	100%			Variable Gaussian mutation	100%	100%	3 connected islands

Figure A.5: The genetic encoding table that was used to define an EA model.



Initial population size	Smpl.	Select Mode	Recombination mode	Mutation Mode	Cross prob.	Mutat. prob.	Population Control	Estimated Utility	Revaluated Utility
200	50 %	Best	1-point	Gauss/σ= 0.5	20 %	100 %	2 connected islands	59.59	55.39
140	50 %	Best	Uniform	Gauss/σ= 0.375	70 %	80 %	3 connected islands	59.85	55.18
140	50 %	Best	Uniform	Gauss/σ= 0.375	50 %	80 %	3 isolated islands	60.19	54.90
180	40 %	Best	Uniform	Gauss/σ= 0.5	10 %	80 %	3 connected islands	63.09	52.00
180	90 %	Best	1-point	Gauss/σ= 0.375	70 %	70 %	2 connected islands	61.64	49.64

Figure A.6: Top 10 EA configurations following the meta-EA optimisation procedure.

Appendix B

The impact of reducing SLG cut

To estimate the impact of the reduction of the SLG cut from 7000 microns to 4000 microns that can affect the Run 1 data desegmentation performance, a short study has been conducted using the control domain data set and plotting the track length distribution and the background removal effect.

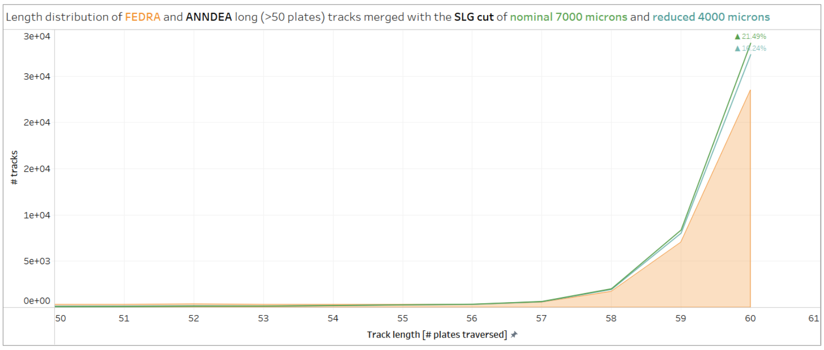


Figure B.1: Distribution of long (>50 plates) FEDRA original tracks (Orange filled area) for control domain simulation and ANNDEA Phase 2 tracks (Green line for nominal and Teal for reduced cut) with their length measured in number of plates traversed. The text labels indicate the percentage change in the number of tracks for this particular length.

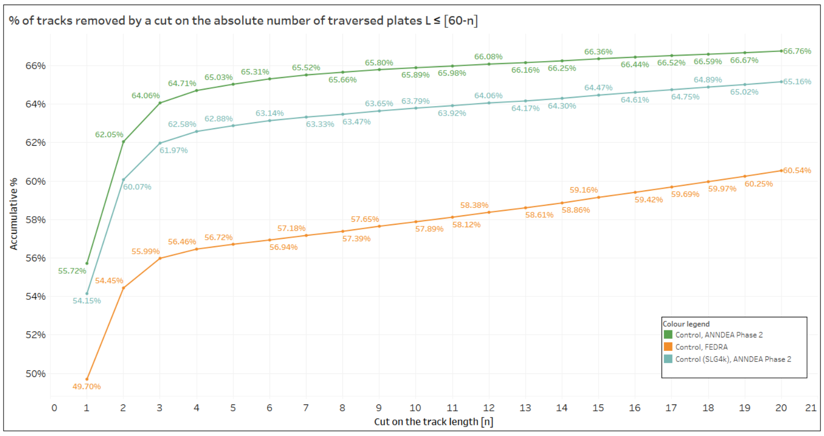


Figure B.2: The plots of the background removal effect on the control domain data that can be achieved thanks to de-segmentation with the lower value of SLG cut set to 4000 microns.

Although there is a drop in the desegmentation performance compared to the nominal ANNDEA configuration, it is still not sufficient to explain the gap observed in Run 1 data.

Appendix C

Classification of the truth particles

The classification used in SND@LHC research applies only to events simulated using the Monte Carlo method.

Table C.1: The particle group mapping based on the Particle, Mother Particle and Process ID.

Particle group	**Particle source**	**Particle ID**	**Mother Particle ID**	**Process ID**	**Additional comments**
Electrons	EM showers	-11, 11	13, -13	9	
	Delta rays	-11	13, -13	8	
Muons	External muons from IP1	-13, 13	0		Mother ID is designated as zero
	Muon neutrino interaction	-13	14		CC interactions only
Hadrons	Muon neutrino interaction	±xxx	14		Particle ID consists of 3 digits
	Hadronic interaction	±xxx	±xxx		Mother/Particle ID consists of 3 digits
	μ/e/γ nuclear interaction	±xxx	±xxx	25, 26, 27, 46	Mother/Particle ID consists of 3 digits

List of Figures

List of Tables

Glossary

νMSM Neutrino Minimal Standard Model. 13

AMS Active Muon Shield. 65, 66, 68, 71–73, 75, 79, 202
ANN Artificial Neural Network. 56, 89, 95, 103, 108, 120, 183, 212

BAU Baryon Asymmetry of the Universe. 1, 12
BSM Beyond Standard Model. 3, 4, 10, 48

CC Charged Current. 16, 31, 47, 208
CEL Cross Entropy Loss. 108, 121, 137, 153, 213–215
CNN Convolutional Neural Network. 55, 91, 103, 118, 152
CP Charge Parity. 12
CPU Central Processing Unit. 75, 103, 105

DAQ Data Acquasition. 34, 42
DIS Deep Inelastic Scattering. 20, 44, 49
DM Dark Matter. 48, 133
DOCA Distance Of Closet Approach. 117, 157

EA Evolutionary Algorithm. 71, 72, 76, 78, 80, 202
ECC Emulsion Cloud Chamber. 34, 38, 41, 81, 83, 90, 96, 115

FIP Feebly Interacting Particles. 31, 39

GAN Generative Adversarial Network. 76
GNN Geometrical Neural Network. 57, 91, 103, 118, 136, 137, 152, 171
GO Grain Oriented steel. 68

HNL Heave Neutral Lepton. 11, 19–21, 209
HS Hidden Sector. 24

IP Interaction Point. 31, 32, 35, 178

LEP Large Electron-Positron collider. 2, 19
LHC Large Hadron Collider. 2, 3, 19, 30, 39, 45, 81, 91
LHCC LHCC - LHC Experiments Committeet. 30

MC Monte Carlo simulation. 86
ML Machine Learning. 4, 49, 52, 118, 136
MLP Multi-Layer Percepetron. 57, 58, 171, 184

NC Neutral Current. 16, 47
NGO Non-Grain Oriented steel. 68

QCD Qauntum Chromo-Dynamics. 19
QFT Quantum Field Theory. 5

RGB Red,Green and Blue. 50
RMSE Root Mean Square Error. 148
RNN Reccurent Neural Network. 91
RSA Random Search Algorithm. 203

SHiP Search for Hidden Particles project. 3, 4, 19, 30, 32, 45
SLG Segment Longitudinal Gap. 128, 151
SM Standard Model of particle physics. 22, 30
SND Scattering Neutrino Detector. 22, 24, 32
SND@LHC Scattering Neutrino Detector at the Large Hadron Collider. 3, 4, 30, 42, 51, 79, 81, 83, 90, 183, 208
SNR Signal-to-Noise Ratio. 117, 154, 184
SPS Super Proton Synchrotron. 23
STG Segment Transverse Gap. 151

THRE Track Hit Reconstruction Efficiency. 111, 124, 126, 174
THRP Track Hit Reconstruction Purity. 112, 124, 126, 141, 143
TRE Track Reconstruction Efficiency. 111, 174

VO Vertex Origin. 151, 157
VRE Vertex Reconstruction Efficiency. 112, 156

VTRE	Vertex Track Reconstruction Efficiency. 112, 156, 157
VTRP	Vertex Track Reconstruction Purity. 113, 156