



UPPSALA
UNIVERSITET

Degree project 45 credits

June, 2025

Displaced Muon Detection Using Machine Learning

ATLAS L0 Muon Trigger Upgrade

Anastasios Kakos

Master's Program in Physics



UPPSALA
UNIVERSITET

Displaced Muon Detection Using Machine Learning

Anastasios Kakos

Abstract

In the search of new physics scenarios, many modern particle physics theories have proposed the extension of the Standard Model (SM), the current understanding of the elementary particles and their forces constituting the universe. Some of these extensions predict the existence of long-lived particles (LLPs), which decay further away from the Interaction Point (IP) producing displaced particle signatures. Unfortunately, studies around them have been hindered by a severe lack of available statistics. To address this issue, this analysis focuses on utilising machine learning (ML) algorithms in the form of Convolutional (CNN) and Recurrent Neural Networks (RNN) with the goal of triggering not only on prompt, but also on displaced muon event signatures using hit data from the Monitored Drift Tube (MDT) chambers of the ATLAS Muon Spectrometer (MS). The algorithms developed took into account the available hardware memory and latency limitations, achieving in some cases a simultaneous background rejection and signal detection efficiency for both prompt and displaced muons greater than 90%, although failing to make a distinction between the two. Training and validation on simulated data from dark photon samples of varying mass and lifetimes, as well as real background data, indicated good generalisation capabilities. The models were successfully converted into High Level Synthesis (HLS) for implementation on Field Programmable Gate Arrays (FPGAs) and post-conversion results showed a viability in detector-level deployment, despite the loss in performance. This study serves as a foundation for ML-based triggering on displaced muon signatures, showing the potential for integration in future ATLAS upgrades, if the necessary optimisation standards are met.

Faculty of Science and Technology

Uppsala University, Place of publication Uppsala

Supervisors: Maria Carnesale - Rimsky Rojas Caballero

Rebeca Gonzalez Suarez

Subject reader: Richard Brenner

Examiner: Andreas Korn

Sammanfattning

Vid jakten på nya fysikscenarion så har många moderna partikelfysik teorier föreslagit en extension av Standard Modellen (SM), den nuvarande bästa modellen för elementär partikelfysik och dess krafter som bygger upp universum. Några av dessa extensioner förutsäger existensen av lång livade partiklar (LLPs), som sönderfaller längre ifrån interaktions punkten (IP) vilket producerar förskjutna partikelsignaturer. Tyvärr har studier kring dessa LLPs varit starkt begränsade av statistik. För att hantera detta problem har denna analys utforskat möjligheten att använda maskin inlärnings (ML) algoritmer, mer specifikt så kallade "Convolutional" (CNN) och "Recurrent Neural Networks" (RNN) med målet att utlösa inte bara omedelbara utan även förskjutna muon event signaturer med hjälp av träff-data från den så kallade "Monitored Drift Tube" (MDT) kammarna av ATLAS muon spektrometer (MS). De algoritmer som har blivit utvecklade tog bland annat hårdvaruminne och latentbegränsningar till hänsyn, vilket i vissa fall uppnådde en bakgrundssuppression och signal detektions effektivitet för både omedelbara och förskjutna muoner på mer än 90% - med nackdelen att skillnaden mellan de två sorters muoner ej lyckas separeras. Träning och validering på simulerad data från mörka fotoner av varierande massa och livstid - samt riktig bakgrundsdata - indikerar att generaliseringsmöjligheten av denna studie är god. Vidare så har modellen blivit konverterad till ett "High Level Synthesis" (HLS) format för att kunna bli implementerad på Field Programmable Arrays (FPGAs), och resultat på efter-konversionen visade att det är implementerbart på en detektor nivå - trots en förlust i prestanda. Denna studie utgör en grund för ML-baserad utlösning på förskjutna myon-signaturer och visar potential för integration i framtida ATLAS-uppgraderingar, förutsatt att nödvändiga optimeringskrav uppfylls.

Acknowledgement I would like to thank our supervisors, Maria Carnesale and Rimsy Rojas Caballero, for all their care, guidance and assistance throughout this project as well as Rebeca Gonzalez Suarez for her dedication, always to the point feedback and precious advice. I would also like to express my sincere gratitude to Verena Martinez Outschoorn and Oliver Kortner, leads of the ATLAS NGT WP2.2, who also provided their insight and along with Rebeca gave us the opportunity to visit and work at CERN making it one of the most memorable experiences of my academic life. Special thanks also to Axel Gallen and Giulia Ripellino for their helpful input.

Contents

Acronyms	vii
1 Introduction	1
2 Introductory concepts	3
2.1 The LHC	3
2.2 The ATLAS detector	3
2.3 ATLAS coordinate system	4
2.4 ATLAS Components	4
2.4.1 Inner Detector	4
2.5 Central Solenoid magnet	5
2.5.1 Toroid magnet system	5
2.5.2 Calorimeter system	6
2.5.3 Muon Spectrometer	6
2.6 The ATLAS Trigger System	7
2.6.1 L1Muon Trigger	9
2.7 Muon detection in the barrel	10
2.7.1 Resistive Plate Chambers	10
2.7.2 Monitored Drift Tube chambers	11
2.7.3 RPC Layout	12
2.7.4 MDT layout	13
2.7.5 Drift distance	14
2.8 The High-Luminosity LHC	14
2.8.1 Level-0 Muon Trigger	15
2.8.2 The ATLAS NextGenTriggers Project	17
2.9 Physics concepts	17
2.9.1 Dark matter	17
2.9.2 Long Lived Particles	17
2.9.3 Dark photon	18

3	Method	19
3.1	Project goal	19
3.2	Constraints	20
3.2.1	Sources of background	20
3.2.2	ATLAS asymmetry	21
3.2.3	FPGA memory limitations	21
3.2.4	Latency limitations	21
3.3	Sample generation	21
3.4	ROOT	22
3.5	Neural network methods	22
3.5.1	CNN	22
3.5.2	RNN	24
3.6	FPGAs and hls4ml	25
4	Implementation	26
4.1	Data preparation	26
4.2	CNN image format	29
4.3	RNN image format	32
4.4	Neural network implementation	33
4.4.1	Model preparation	33
4.4.2	Dataset splitting	33
4.4.3	Displacement threshold	34
4.4.4	Normalisation	35
4.4.5	Model training and testing	35
4.4.6	CNN architecture	35
4.4.7	RNN architecture	39
4.5	Converting models with hls4ml	40
5	Results and Discussion	40
5.1	CNN results	40

5.1.1	Detecting all muons	41
5.1.2	Detecting only displaced muons	61
5.2	RNN results	71
5.2.1	Detecting all muons	71
5.2.2	Detecting only displaced muons	73
5.3	hls4ml implementation	74
5.3.1	CNN hls4ml models	74
5.3.2	RNN hls4ml models	75
5.4	Comparison between CNN and RNN	78
6	Conclusions	78
6.1	Potential future work	79
A	Appendix	84
A.1	ATLAS MDT Tables	84

Acronyms

ASIC Application Specific Integrated Circuit. 15, 25

ATLAS A Toroidal LHC ApparatuS. i, 1, 3, 5, 6, 17, 20, 21, 27, 28, 78, 79, 84

AUC Area Under Curve. 50, 75

BI Barrel Inner. 7, 12, 13, 15, 26, 35, 37, 59, 61, 71, 78, 84

BM Barrel Middle. 7, 12, 13, 26, 35, 37, 41, 50, 61, 71, 78

BO Barrel Outer. 7, 9, 12, 13, 26, 35, 37, 61, 71, 78

BSM Beyond the Standard Model. 1, 79

BT Barrel Toroid. 5, 7

CL Combined Loss. 41

CM Centre-of-Mass. 3

CMS Compact Muon Solenoid. 1, 3

CNN Convolutional Neural Network. i, 1, 2, 20–24, 28–31, 33, 35, 37, 41, 48, 59, 61, 71, 73, 74, 78, 79

CTP Central Trigger Processor. 8

DAQ Data Acquisition. 8, 15

DL Dice Loss. 41

DV Displaced Vertex. 17, 18, 26, 27, 59

ECT Endcap Toroid. 5

FL Focal Loss. 41

FPGA Field Programmable Gate Array. i, 15, 17, 19–21, 25, 26, 28, 38–40, 78, 79

FPR False Positive Rate. 48

GPU Graphical Processing Unit. 22

GRU Gate Recurrent Unit. 24, 25, 39, 40

HAHM Hidden Abelian Higgs Model. 1, 19, 21, 27, 28, 31

HL-LHC High Luminosity - Large Hadron Collider. 1, 12, 14, 15, 18, 19

HLS High-Level Synthesis. i, 26

HLT High Level Trigger. 8, 9, 19, 59, 78, 79

HPC High-Performance Computing. 25

IBL Insertable B-Layer. 4

ID Inner Detector. 4, 5

IP Interaction Point. i, 4, 14, 17, 59

L0 Level-0. 15–17, 19, 21, 59, 79

LAr Lead/Liquid Argon. 6

LHC Large Hadron Collider. 1, 3, 14, 17

LLP Long Lived Particle. i, 1, 15, 17, 18, 21

LO Leading Order. 21

LSTM Long Short Term Memory. 24, 25, 39, 40

MDT Monitored Drift Tube. i, 6, 11–15, 17, 19, 21, 24, 26–30, 32, 33, 37, 59, 71, 78, 84

ML Machine Learning. i, 1, 17, 19–22, 25, 26, 33, 35, 78, 79

MM Micromegas. 6

MS Muon Spectrometer. i, 1, 6, 7, 11, 20, 21, 78, 79

MUCTPI Muon-to-Central Trigger Processor Interface. 8

NGT NextGenTriggers. 1, 17, 19, 79

PDG Particle Data Group. 26, 27

PV Primary Vertex. 17, 18, 26, 27

RNN Recurrent Neural Network. i, 1, 2, 20, 22, 24, 28, 32, 33, 39, 40, 71, 73, 75, 78, 79

ROC Receiver Operating Characteristic. 48–50, 61, 73

RoI Region of Interest. 8, 15, 79

RPC Resistive Plate Chamber. 6, 9, 10, 12, 15, 17, 21, 79

SCT SemiConductor Tracker. 5

SL Sector Logic. 15, 19, 78

SM Standard Model. i, 1, 17–19

sMDT small-diameter Monitored Drift Tube. 15

sTGC small-strip Thin Gap Chamber. 6

TGC Thin Gap Chamber. 6, 9

TNR True Negative Rate. 72, 74

TPR True Positive Rate. 48, 71, 74, 75

TRT Transition Radiation Tracker. 5

1 Introduction

The Large Hadron Collider (LHC) at CERN is the most powerful particle accelerator ever built. It consists of various experiments across its 27 km accelerator ring and is able to produce around 1.7 billion collisions per second [1].

The ATLAS [2] (A Toroidal LHC ApparatuS) experiment is the largest general-purpose particle detector at the LHC which contributed alongside the CMS (Compact Muon Solenoid) experiment to the discovery of the Higgs boson in 2012, cornerstone of the Standard Model (SM), our best understanding of how the elementary particles, the basic building blocks of the universe, and the forces they are governed by are related to each other.

The ATLAS collaboration [3] consists of more than 6000 participants from over 40 countries and over 180 institutions, making it one of the largest scientific collaborations.

The billions of collisions in ATLAS have a combined data volume of more than 60 million MB per second, but only some of these will contain interesting characteristics that might lead to new discoveries. To reduce data flow to manageable levels, ATLAS uses a special event selection system, the “trigger” [1], which picks events with distinguishing characteristics for physics analysis.

The High Luminosity LHC (HL-LHC) is a major upgrade of the LHC, aiming to increase the number of collisions by a factor of 5 – 7.5 [4] with respect to the nominal LHC design, thus significantly increasing the amount of data produced.

For the data acquisition system to be able to handle this increase in data volume, an upgrade to the trigger system is also required.

The HL-LHC will be especially useful to explore rare Beyond the Standard Model (BSM) processes characterised by low statistics. Some of the most interesting cases of BSM physics involve signatures of exotic Long Lived Particles (LLPs), like for example, the dark photon, a hypothetical particle which according to the Hidden Abelian Higgs Model (HAHM) [5] could potentially decay to SM leptons and more specifically, to two collimated, displaced muons.

At ATLAS, muons are detected in the Muon Spectrometer (MS) and while there are dedicated triggers regarding prompt muons, this is not the case for displaced ones.

One of the projects focused on the upgrade of the trigger system for the upcoming HL-LHC is the NextGenTriggers (NGT) [6] project of ATLAS. NGT works to improve the trigger acquisition system of the ATLAS MS by making use of state-of-the-art Machine Learning (ML) techniques and it is in this context that the design of a dedicated trigger algorithm for displaced muon signatures could be of great interest. The project, funded by donations, aims to develop this technology in the next 5 years during the upcoming LHC Long Shutdown 3 scheduled from July 2026 to June 2030 [7].

This thesis focuses on contributing to the enhancement of the first level trigger system of ATLAS in its barrel region by developing ML models that filter events generated from pp - collisions in the ATLAS detector and selecting those displaying signatures of displaced muons originating from the decay of LLPs, such as dark photons. This project was carried out in collaboration with Erwan Larié as part of the Uppsala University Nuclear and Particle Physics Master’s Programme for the NGT project of ATLAS.

The tasks related to the Convolutional Neural Network (CNN) aspect described in this report were carried out by the author, and the tasks relevant to the Recurrent Neural Network (RNN)

were performed by the collaborator Erwan Larié, with the latter providing assistance in the initial stages of the CNN development due to the RNN development beginning at a later stage.

2 Introductory concepts

2.1 The LHC

The LHC [8] is the world's largest and most powerful particle accelerator located beneath the France–Switzerland border near Geneva at a depth of up to 175 m. It first started operation during 2008, and remains, up to this day, the latest addition to the accelerator complex of CERN. It hosts 9 experiments [9], the biggest of which are ATLAS, CMS, ALICE and LHCb, while the smallest ones are TOTEM, LHCf, MoEDAL-MAPP, FASER and SND@LHC. The LHC consists of a 27 km ring of superconducting magnets with a number of accelerating structures to boost the energy of the particles along the way. The accelerated particles are protons or heavy ions.

Particle beams travelling in opposite directions using separate beam pipes at ultra-high vacuum [10] ($10^{-10} - 10^{-11}$ mbar pressure) are accelerated to speeds close to the speed of light with the help of a strong magnetic field maintained by superconducting electromagnets operating at -271.3°C , a temperature preserved through a distribution system of liquid helium.

So far, the LHC has had three data taking periods, referred to as Runs. During Run 1 (2010 – 2012) the LHC operated at a Centre-of-Mass (CM) energy of 7 and 8 TeV, while during Run 2 (2015 – 2018) it delivered up to 7 times more data [11] by operating at a CM energy of 13 TeV, delivering in the second half of 2017 an instantaneous luminosity of $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ and 156 fb^{-1} of integrated luminosity by its end [4].

Here, it is mentioned that *instantaneous luminosity*, $\mathcal{L}$, is the measure of the number of potential collisions per surface unit (or cross section σ) over a given period of time.

$$\mathcal{L} = \frac{1}{\sigma} \frac{dN}{dt} \quad (1)$$

Integrated luminosity, $\mathcal{L}_{\text{int}}$, is a measure of the collected data size and defined as the time integral of the instantaneous luminosity measured in inverse femtobarns (fb^{-1}).

$$\mathcal{L}_{\text{int}} = \int \mathcal{L} dt \quad (2)$$

Run 3 (2022 – 2026) [12] is currently on-going where a peak CM energy of 13.6 TeV has been achieved and the dataset available for analysis relative to Run 2 is expected to be tripled.

2.2 The ATLAS detector

The ATLAS experiment (Figure 1) at the LHC is the largest detector [13] ever constructed for a particle collider. It is a multi-purpose particle detector with a forward-backward symmetric cylindrical geometry, dimensions of 46 m in length and 25 m in diameter and a near 4π coverage in solid angle.

ATLAS is designed to record the billions of high-energy proton or ion collisions taking place at its center using more than 100 million electronic channels. It consists of an inner tracking detector surrounded by a thin superconducting solenoid providing a 2 T axial magnetic field, a toroid magnet system providing a magnetic field of up to 3.5 T, electromagnetic and hadronic

calorimeters, and a muon spectrometer. Each of these components is described in the following subsections.

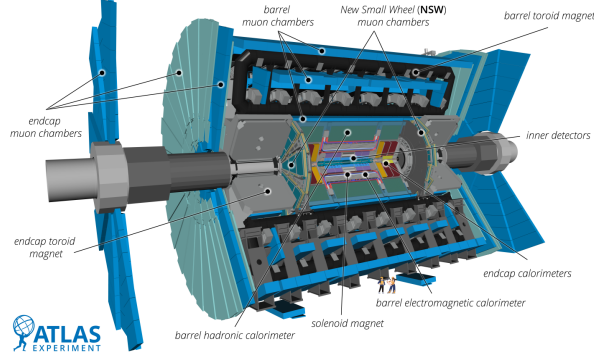


Figure 1 Schematic of the ATLAS experiment [14].

2.3 ATLAS coordinate system

ATLAS uses a right-handed coordinate system with its origin at the nominal Interaction Point (IP) in the centre of the detector and the z -axis along the beam pipe.

The x -axis points from the IP to the centre of the LHC ring, and the y -axis points upwards.

Polar coordinates (r, ϕ) are used in the transverse plane, ϕ being the azimuthal angle around the z -axis.

The pseudorapidity is defined in terms of the polar angle θ as:

$$\eta = -\ln \tan \left(\frac{\theta}{2} \right) \quad (3)$$

And is equal to the rapidity in the relativistic limit. The rapidity is defined as:

$$y = \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right) \quad (4)$$

Where E is the particle energy and p_z the modulus of the particle momentum along the z -axis. Angular distance is measured in units of:

$$\Delta R \equiv \sqrt{(\Delta y)^2 + (\Delta \phi)^2} \quad (5)$$

2.4 ATLAS Components

2.4.1 Inner Detector

The Inner Detector [15] system (ID) is immersed in a 2 T axial magnetic field and provides charged-particle tracking in the range $|\eta| < 2.5$. The high-granularity silicon pixel detector covers the interaction region and typically provides four measurements per track, the first hit generally being in the Insertable B-Layer (IBL) [16]. It is followed by the SemiConductor

Tracker (SCT), which usually provides eight measurements per track. These silicon detectors are complemented by the Transition Radiation Tracker (TRT), which enables radially extended track reconstruction up to $|\eta| = 2.0$. The TRT also provides electron identification information based on the fraction of hits (typically 30 in total) above a higher energy-deposit threshold corresponding to transition radiation.

2.5 Central Solenoid magnet

The ATLAS solenoid [17] surrounds the ID at the core of the experiment. This powerful magnet is 5.8 m long, 2.56 m in diameter and weighs over 5 tonnes. It provides the 2 T axial magnetic field of the ID in just 4.5 cm thickness. This is achieved by embedding over 9 km of niobium-titanium superconductor wires into strengthened, pure aluminum strips, thus minimising possible interactions between the magnet and the particles being studied. Its coil is indirectly cooled using the forced flow of two-phase helium through 6 parallel cooling circuits.

2.5.1 Toroid magnet system

The ATLAS toroids [17] use a series of eight coils to provide a magnetic field of up to 3.5 T, used to measure the momentum of muons. There are three toroid magnets in ATLAS: two at the ends of the experiment, the Endcap Toroids (ECTs), and one massive toroid surrounding the centre of the experiment, the Barrel Toroid (BT).

The BT [18] uses over 56 km of superconducting wire, weighs about 830 tonnes and consists of 8 coils. Each coil has the shape of a flat racetrack frame enclosing two double pancake windings. The windings are housed in an aluminium alloy casing that guarantees cooling, rigidity and transfer of the magnetic forces.

Each coil has its own cryostat that transfers the forces between the coils. The toroidal structure is essentially built up with 8 cryostats and the linking elements between the cryostats, which provide mechanical stability of the whole structure. The coils are cooled in the indirect mode by circulating liquid helium in tubes welded onto the coil casings.

The two ECTs [18] extend the magnetic field to particles leaving the detector close to the beam pipe. Each endcap is 10.7 m in diameter and weighs 240 tonnes. The ECTs are inserted in the BT at each end and line up with the superconducting solenoid surrounding the ID. Each ECT consists also of 8 racetrack coils with pancake windings in aluminium alloy coil casing. The 8 coil casings are cold-linked with a web structure and all assembled as a single cold mass in one large cryostat. The internal forces in the toroids are taken by the cold supporting structure between the coils, which is essentially different from the warm structure design of the BT. Due to the magnetic forces, the ECT magnets are pulled into the BT and the corresponding axial forces are transferred to the BT warm structure via the ECT cryostats and axial transfer points linking both magnet systems.

2.5.2 Calorimeter system

The calorimeter system [15] covers the pseudorapidity range $|\eta| < 4.9$.

Within the region $|\eta| < 3.2$, electromagnetic calorimetry is provided by barrel and endcap high-granularity Lead/Liquid-Argon (LAr) calorimeters, with an additional thin LAr presampler covering $|\eta| < 1.8$ to correct for energy loss in material upstream of the calorimeters. Hadronic calorimetry is provided by the steel/scintillator-tile calorimeter, segmented into three barrel structures within $|\eta| < 1.7$, and two copper/LAr hadronic endcap calorimeters. The solid angle coverage is completed with forward copper/LAr and tungsten/LAr calorimeter modules optimised for electromagnetic and hadronic energy measurements, respectively.

2.5.3 Muon Spectrometer

The Muon Spectrometer (MS) [19] (Figure 2) forms the outer shell of the ATLAS detector and occupies by far the largest part of its volume. It is located on the outside of the calorimeter modules and covers the space between approximately 4.5 m and 11 m in radius and 7 m and 23 m longitudinally on both sides of the interaction point. The total volume is approximately 16 000 m³.

The MS comprises separate trigger and high-precision tracking chambers measuring the deflection of muons in a magnetic field generated by the superconducting air-core toroidal magnets. The field integral of the toroids ranges between 2.0 and 6.0 T m across most of the detector. Three layers of precision chambers, each consisting of layers of Monitored Drift Tubes (MDTs), cover the region $|\eta| < 2.7$. These are complemented, since Run 3, in the innermost layer of the endcap by layers of small-strip Thin Gap Chambers (sTGCs) and Micromegas (MM) chambers, both providing precision tracking in the region $1.3 < |\eta| < 2.7$. The muon trigger system covers the range $|\eta| < 2.4$ with Resistive Plate Chambers (RPCs) in the barrel region, Thin Gap Chambers (TGCs) in the endcap regions, and the aforementioned sTGCs and MM chambers in the innermost layer of the endcap.

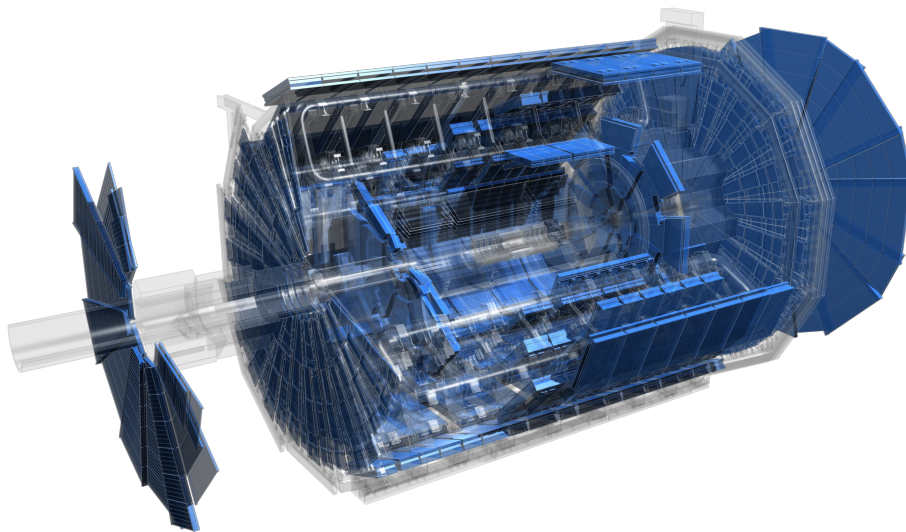


Figure 2 Highlight of the ATLAS Muon Spectrometer [20].

In the barrel region, the muon chambers are arranged in three concentric cylinders around the beam axis, whereas the endcap chambers form four disks on each side of the interaction point,

concentric around the beam axis. The chambers are placed such that particles that originate at the interaction point traverse the three chamber stations. The position of these stations has been chosen to take optimum advantage of the magnetic field configuration. Since the analysis described in this thesis focuses exclusively on the barrel region of the MS, an emphasis will be placed toward the description of the barrel chamber layout rather than that of the endcap. The cylindrical barrel stations are referred to as Barrel Inner (BI), Barrel Middle (BM) and Barrel Outer (BO).

Each of the barrel stations is arranged in large and small sectors. The large sectors cover the region between the BT coils, while the small sectors cover the azimuthal range of the barrel toroid coils. In the barrel, a 16-fold azimuthal segmentation has been chosen in such a way that each of the barrel stations is divided into 8 large and 8 small φ sectors (Figure 3). There is also a division with respect to pseudorapidity along the beam axis (Figure 4), and thus each barrel station spans across up to 14 η stations, indexed from -7 to $+7$, excluding the value of 0 for symmetry reasons, although, depending on the barrel station, η stations -7 and $+7$ might not be occupied by detection units due to geometry or design reasons.

The distinction between large and small chamber barrel stations is made using a station index value from 0 to 5, as shown in Table 1.

Station Index	0	1	2	3	4	5
Barrel Station	BIS	BIL	BMS	BML	BOS	BOL

Table 1 Station Indices correspondence to Barrel Stations

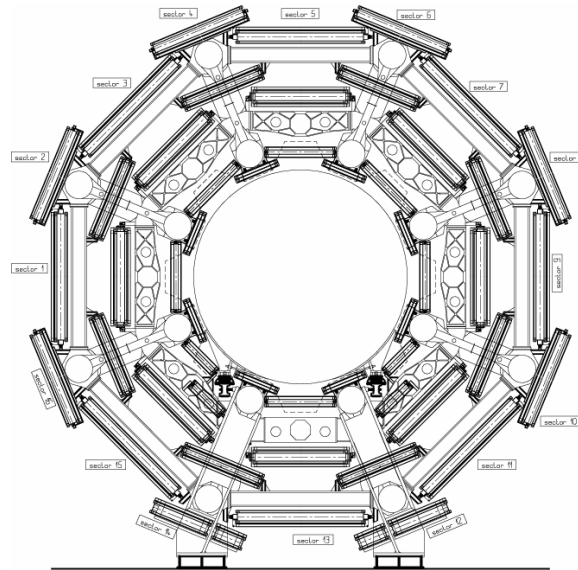


Figure 3 Cross sectional view of the ATLAS Muon Spectrometer showing the division in φ sectors [21].

2.6 The ATLAS Trigger System

The trigger system [23] (Figure 5) is an essential component of the ATLAS experiment as it is responsible for deciding whether or not to permanently record data from a given collision. It aims to minimise the rate of the recorded events while maintaining an excellent and unbiased efficiency for physics processes of interest.

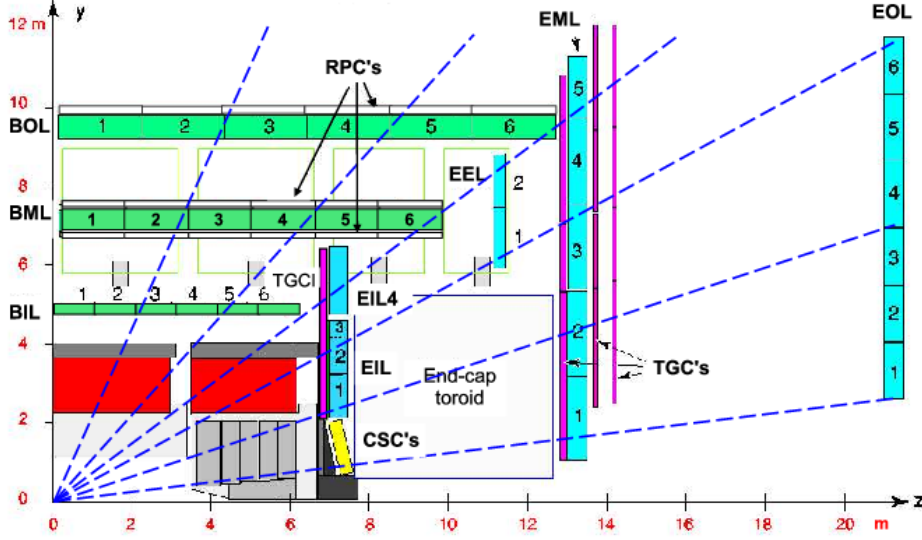


Figure 4 Cross sectional view of the ATLAS Muon Spectrometer in the $r - z$ projection at $\varphi = \pi/2$, showing the division in η stations.
Cathode Strip Chambers (CSCs) have been replaced by sTGCs and MM since Run 3 [22].

The first-level (Level-1, L1) trigger is mainly based on two independent systems which use custom electronics to trigger on reduced-granularity information from either the calorimeters (L1Calo) or the muon detectors (L1Muon). The L1 topological processor (L1Topo) system uses kinematic information from objects reconstructed in the L1Calo and L1Muon systems and applies topological selections. The L1 trigger decision is formed by the Central Trigger Processor (CTP), based on the inputs received from the L1Calo trigger system, the L1Muon trigger system through the Muon-to-Central Trigger Processor Interface (MUCTPI), the L1Topo system, as well as several other subsystems. The CTP is also responsible for applying *dead time*, a mechanism to limit the number of close-by L1 accepts.

Events that satisfy the trigger menu requirements based on object type, threshold and multiplicity are accepted at a rate up to the maximum detector read-out rate of 100 kHz (down from the bunch crossing rate of about 40 MHz) at a fixed latency (detector read-out time window) below $2.5 \mu\text{s}$.

Up to 512 distinct L1 trigger items may be configured in the CTP.

If accepted by the L1 trigger, events are then sent to a software-based High Level Trigger (HLT). Here online algorithms reconstruct the event at progressively higher levels of detail than at L1, either in the full detector volume or in restricted Regions of Interest (RoIs), which are detector regions in which candidate trigger objects have been identified by the L1 trigger. The physics output rate of the HLT during an ATLAS data-taking run is expected to be 3 kHz on average.

Finally, the Data Acquisition (DAQ) system transports data from custom subdetector electronics to offline processing, according to the decisions made by the trigger. Data to be used for offline analysis are compressed prior to processing from the original event size of about 3 MB to below 2 MB.

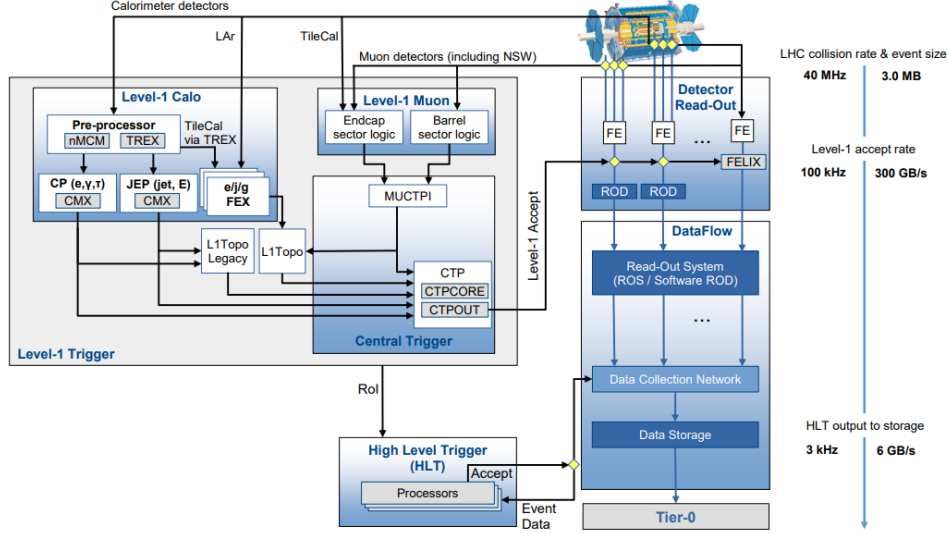


Figure 5 The ATLAS Trigger and DAQ system in Run 3 with emphasis on the components relevant for triggering as well as the detector read-out and data flow. L1Calo, L1Muon and the CTP all send data to the Read-Out System [23].

2.6.1 L1Muon Trigger

The L1Muon trigger uses hits from the RPCs (in the barrel) and the TGCs (in the endcaps) to determine the deviation of the hit pattern from that of a muon with infinite momentum. Due to the fact that the emphasis of this analysis is placed on the muon trigger regarding the barrel region, an extensive description of the L1Muon trigger in the endcap region is not provided, but more information for the curious reader can be found at [23].

The muon trigger in the barrel region [4] ($|\eta| < 1.05$) is based on three trigger stations. Two stations are used for low- p_T (6–10 GeV) muon triggers, while the third station is used in addition for high- p_T (8–40 GeV) triggers (Figure 6).

Each station is composed of two detector planes, and each detector plane is read out, as already mentioned, in two orthogonal projections, η and φ , which are also commonly referred to as the bending and non-bending projections, respectively. The muon trigger p_T resolution is mainly determined by the information provided by the detectors in the bending projection.

However, the detector information provided in the non-bending view helps to reduce the background trigger rate from noise hits in the chambers produced by low-energy photons, neutrons and charged particles, as well as localising the track candidates in space as required for the HLT.

In addition, the trigger chamber information in the non-bending view provides a second coordinate measurement for offline reconstruction of muons, in contrast to the precision chambers that give information only in the bending projection.

Muon triggering is achieved with RPC doublets in each station and is based on a coincidence logic. For the low- p_T thresholds, the coincidence of at least one hit in each of the two inner station RPC doublets is required (since Run 3 [23]). The high- p_T trigger thresholds require the low- p_T trigger logic to be satisfied, as well as an additional hit in one of two layers in the outer barrel station (BO). There are a total number of six thresholds, three for the low- p_T muons and

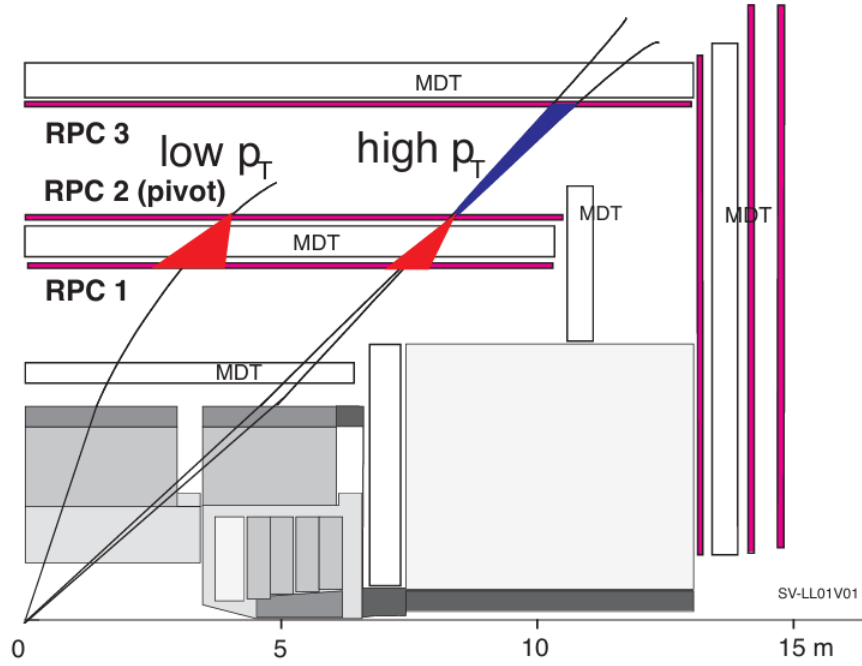


Figure 6 The ATLAS L1Muon trigger scheme in the barrel region [4].

three for the high- p_T ones.

2.7 Muon detection in the barrel

2.7.1 Resistive Plate Chambers

RPCs have already been mentioned, while providing an overview of the ATLAS detector and its trigger system, but no comment was made on their functioning principles.

RPCs are gaseous detectors and their operation is based on the detection of the gas ionization produced by charged particles when traversing the active area of the detector, under a strong uniform electric field applied by resistive bakelite plates acting as electrodes. The plates are kept spaced by insulating spacers.

RPCs used in the ATLAS experiment [24] (Figure 7) are characterized by plates of 2 mm thickness, while the gap between them (often referred to as the *gas gap*) is at 2 mm as well. The spacers are cylindrical with 12 mm diameter and are placed every ≈ 10 cm in both directions. The gas mixture consists of $\text{C}_2\text{H}_2\text{F}_4:\text{C}_4\text{H}_{10}:\text{SF}_6$ (with percentages 94.7:5:0.3), while the electric field between the plates is typically at 4.5 kV mm^{-1} .

Due to the strong electric field, when the incident particle ionises the gas molecules, the primary ionisation electrons are multiplied into avalanches.

The discharge electrons drift in the gas and the signal is induced on a series of pick-up copper strips, which is read out via capacitive coupling and detected by the front-end electronics. There are a series of strips on both sides of the detector, placed orthogonally to each other, so that they match the η and φ coordinates. On one side, the η -strips provide the bending view of the trigger detector, while the φ -strips provide the second-coordinate measurement which is also

required, as already mentioned, for the offline pattern recognition.

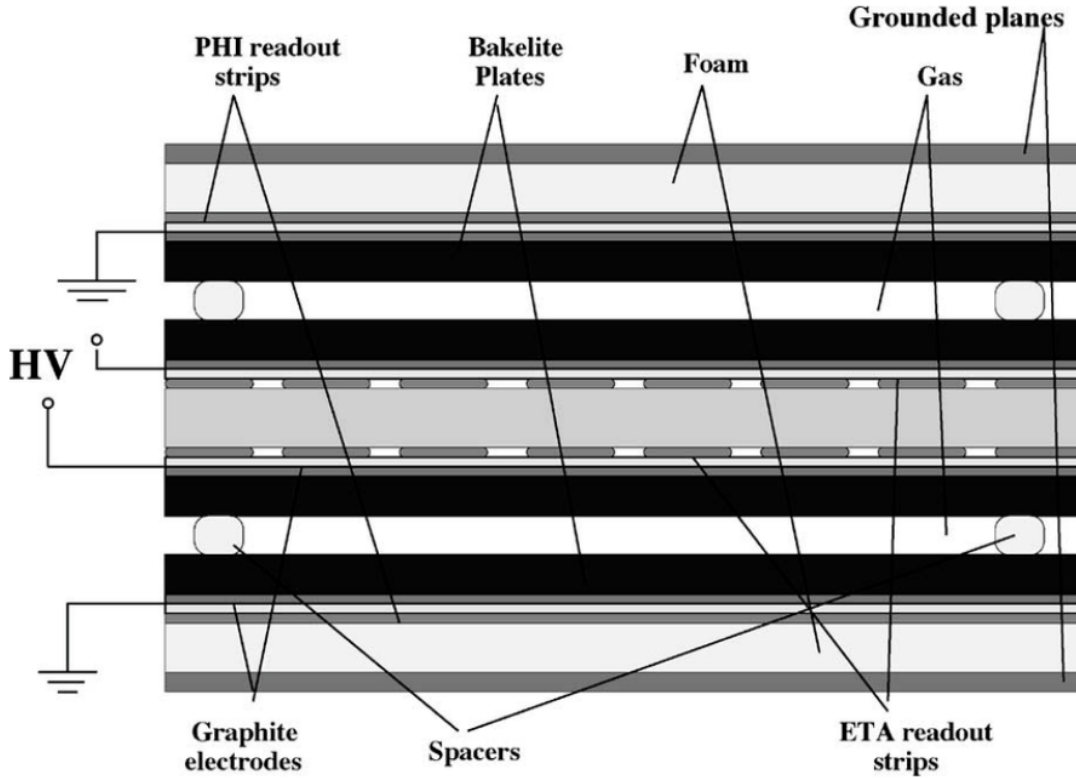


Figure 7 Schematic view of the two gas volumes of an ATLAS RPC unit [25].

2.7.2 Monitored Drift Tube chambers

MDT chambers [26] are one of the main types of precision chambers used for muon tracking and therefore provide the muon momentum measurement both in the barrel and endcap regions of the ATLAS MS.

The primary sensitive element of the MDT chambers is a cylindrical aluminium drift tube with 30 mm diameter, in the center of which an anode wire of 50 μm diameter is held by the end-plugs from the sides of the tube. The length of the drift tube ranges from 70 cm to 630 cm, depending on the MS geometry requirements.

Each MDT chamber consists of 2×3 or 2×4 monolayers of drift tubes, while the drift tubes are glued on three cross plates connected by two longitudinal beams (Figure 8). The tubes are closely spaced so that each 'triple layer' or 'multilayer' has a thickness of about 82 mm.

MDTs also take advantage of the avalanche multiplication operating principle and by registering the drift times (to the wire) of the ionization electrons in the gas (Figure 9), one determines six coordinates of a typical track in the plane of the layer and in the direction across the tubes. This results in a measurement of effectively one coordinate with 40 μm precision and one angle with 3×10^{-4} precision.

The drift signal is processed with a current-sensitive amplifier followed by a shaper and a discriminator set at a threshold, which corresponds to the 22nd electron when the avalanche amplification is 2×10^4 . With these parameters the average single tube resolution has a value of 80 μm .

The gas mixture consists of Ar:CO₂ (93:7) with a few hundred ppm of water at 3 bar.

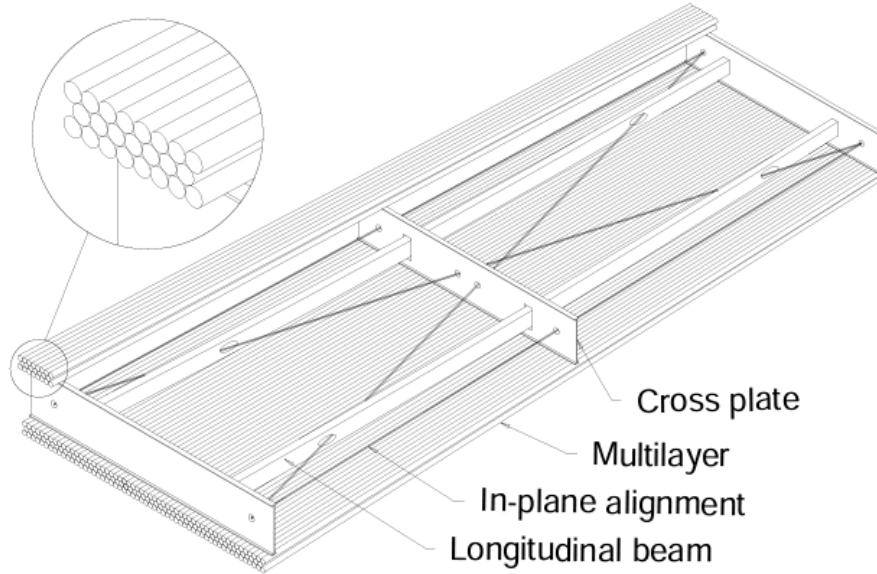


Figure 8 Schematic drawing of an MDT chamber. Structural components are indicated [21].

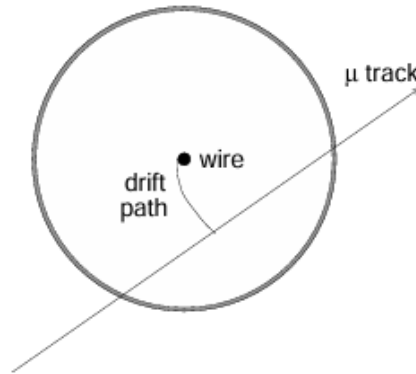


Figure 9 Drift tube operation in a magnetic field with curved drift path [21].

2.7.3 RPC Layout

RPCs are placed in doublets [21] in the barrel. Each chamber is made up of two modules (Figure 10) overlapping in the z direction (to achieve full acceptance), while each module is subdivided into two units along the azimuthal direction (there is a dead area of 14 mm between the two units). The locations of the three RPC stations are above and below the MDTs in the BM and above the BOL or below the BOS MDT chambers. The longitudinal and lateral dimensions of the RPCs have been chosen such that their active areas match the ones of the corresponding MDTs.

To this day, the BI does not possess any RPC layer, however in the context of the HL-LHC upgrade, a triplet of RPCs will be added to the BI, which will increase acceptance to 96% [27].

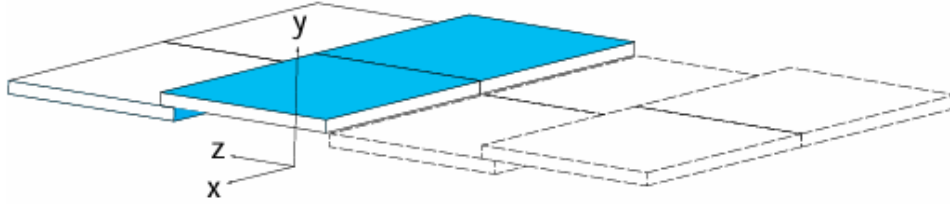


Figure 10 Schematic view of two adjacent RPCs (one drawn with a solid line, the other with a broken line). Each chamber is made of two modules overlapping in the z direction and each of the modules is again subdivided into two units in the azimuthal direction [21].

2.7.4 MDT layout

Each MDT chamber consists of two multilayers and three layers of drift tubes in each multilayer in the BO and BM, while in the BI, each multilayer consists of four layers of tubes (Figure 11).

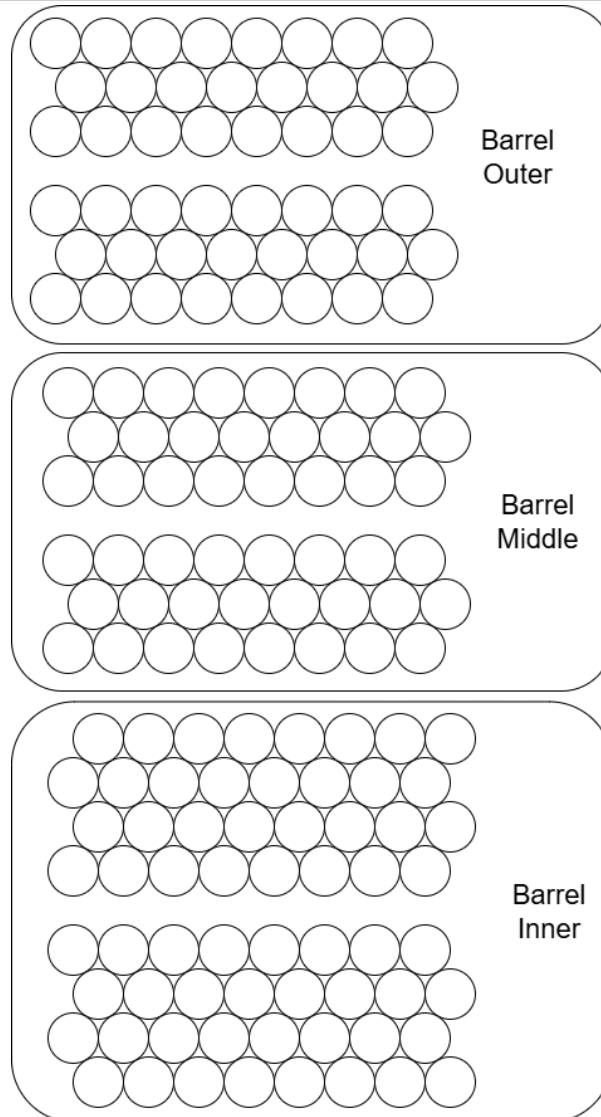


Figure 11 Representation of the MDT chamber layers across the different barrel stations

2.7.5 Drift distance

Drift distance [28] corresponds to the distance between the central MDT wire and the point of first ionization of a passing charged particle and possesses a crucial role towards vertex reconstruction, as it allows for the tracking of a charged particle based on the calculated displacement value originating from several MDTs.

The hit time data are recorded in $1.2\,\mu\text{s}$ windows. The drift time of a signal pulse is obtained by subtracting from the raw pulse arrival time (calibrated in conjunction with the relative bunch crossing) an offset time t_0 , which takes into consideration various phenomena causing latency, such as the time of flight of the particle (muon) from the pp IP to the tube, the signal propagation time along the tube wire, plus various fixed electronics and cable latencies.

The drift distance (or drift radius) is obtained from the drift time using a time-to-distance transfer function $R(t)$, property of a single MDT chamber taking into account the operating conditions of the chamber.

2.8 The High-Luminosity LHC

The HL-LHC [29] is a major upgrade of the LHC and is expected to start operations in the middle of 2026, and to reach nominally a peak instantaneous luminosity of $\mathcal{L} = 5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, corresponding to approximately 140 inelastic proton-proton collisions per bunch crossing, which could be maintained for the duration of the HL-LHC project. In this configuration, approximately 250 fb^{-1} will be delivered per year, for a total of 3000 fb^{-1} by the end of Run 6, ten times the integrated luminosity of the LHC Runs 1 – 3 combined.

There is even potential for the HL-LHC to operate to its *ultimate configuration* [4], where the instantaneous luminosity can reach the value of $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, resulting in 200 inelastic collisions per bunch crossing, $300\text{--}350 \text{ fb}^{-1}$ of data per year or a total of 4000 fb^{-1} .

There are various upgrades scheduled for the LHC to be able to cope with the new luminosity standards, some of which are listed below:

1. More powerful focusing magnets and new optics to focus the particle bunches before they meet with higher magnetic fields
2. “Crab cavities” for tilting the beams to increase the probability of collisions
3. Reinforced machine protection with new, more performant collimators
4. New crystal collimators to improve the cleaning efficiency of the insertions for ion beam operation
5. Innovative superconducting links to carry currents of record intensities
6. An upgraded accelerator chain
7. An upgraded Trigger and Data Acquisition System

Since the focus of this thesis is on contributing to the triggering system of ATLAS in the barrel region, only the very last item of the list above is discussed.

An upgrade to the ATLAS Trigger and DAQ system is essential to fully exploit the physics potential of the HL-LHC.

The current minimum muon p_T threshold for correct triggering on rare interesting events is at 20 GeV. The L1Muon trigger system has a rate of 100 kHz, however, it would be impossible to maintain that rate at HL-LHC conditions without increasing the threshold to at least 50 GeV. In order to maintain the current threshold, a rate of 1 MHz is required.

In general, high- p_T thresholds have 5 – 10% lower efficiency than the low- p_T thresholds in the barrel region and this is due to the coincidence requirements discussed in Section 2.6.1, so keeping the p_T threshold low ensures better detection efficiency as well as less rejection of the most interesting signatures in the sub-50 GeV p_T range.

2.8.1 Level-0 Muon Trigger

The Level-0 (L0) Muon trigger is an upgrade to the current L1Muon trigger of ATLAS.

The L0 trigger candidates will be sent to the new ATLAS Global Trigger (which will replace and extend the functionality of the previously mentioned L1Topo system [4]) within a latency of approximately 5 μ s. The readout buffers, which store data until the L0 trigger decision is delivered, have to be large enough to accommodate a maximum latency of 10 μ s. Compared to the current 2.5 μ s latency, this extra time allows for much more complex algorithms to be implemented.

The L0Muon trigger will operate at a rate of 1 MHz, thus meeting the aforementioned rate requirement to be able to trigger at low p_T thresholds and thus, it will be able to maintain good acceptance for the most important physics channels despite the luminosity increase.

For this reason, triplets of new generation RPCs will be installed in the BI region, in order to increase trigger acceptance. The barrel trigger system will thus, benefit from increased redundancy (9 layers of detectors instead of 6), allowing for more coincidence logic schemes and a geometrical acceptance of 96% instead of 80% [30]. For the new BI RPCs to be installed, the replacement of the BIS MDT chambers with small-diameter MDTs (sMDTs) is required. sMDTs have a diameter of only 15 mm, approximately four times shorter maximum drift times (compared to regular MDTs), eight times lower occupancy and are in general designed to cope with the high background rates of the HL-LHC. These upgrades are depicted in Figure 12.

A replacement of the DAQ and readout electronics is foreseen as well. Instead of the presently used on-detector custom Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) will be used to collect and optically transfer hit data from the frontend to the backend boards, where the L0Muon will be executed. The better time resolution of the FPGAs, when it comes to RPC hit sampling can be used for more accurate time of flight measurements, greatly benefitting searches for LLPs. The L0Muon barrel trigger algorithm (Figure 13) will be performed off-detector by 32 FPGA-based boards [30], called Sector Logic (SL), one per each ATLAS φ sector.

To maximize the performance and flexibility of the trigger system, the full hit data of RPCs and MDTs will be sent directly to the barrel SL boards and the new MDT Trigger Processors, which will match MDT hits with RoIs formed by the RPC candidates, allowing for better momentum resolution and thus, improved rejection of fake muon hits that could potentially bypass the triggering selection even when not meeting the minimum p_T threshold requirement.

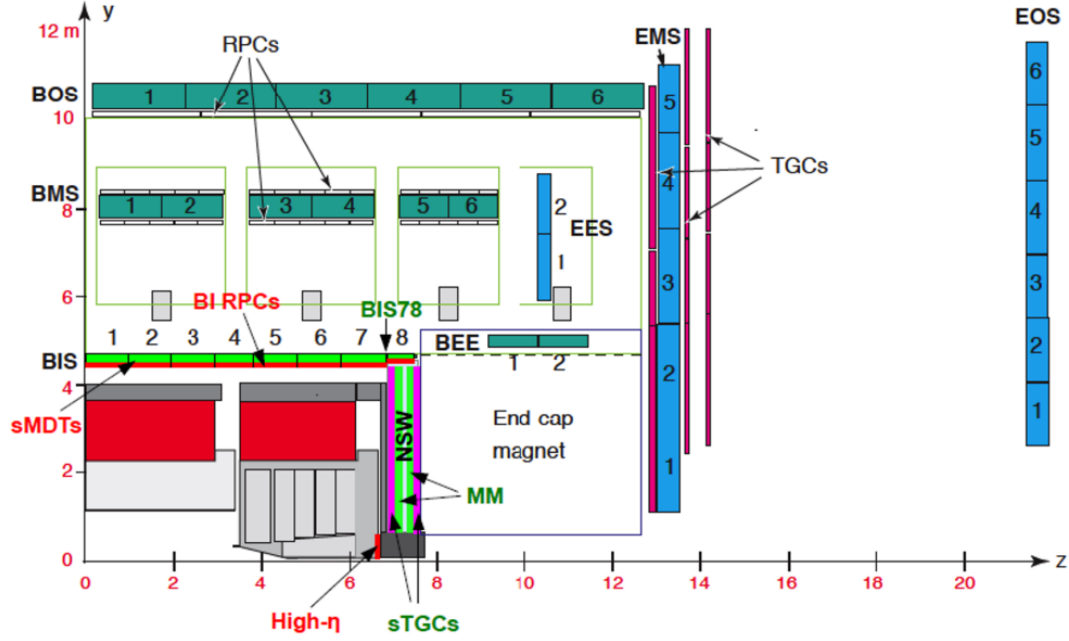


Figure 12 Layout of the foreseen detector upgrades in the Muon Spectrometer for Phase-2 (for Small sectors only) [31].

The combination of the fact that this matching is done only in small regions of the detector per event (making it computationally cheap) with the higher latency of the new first-level trigger allows this selection to be done even at L0.

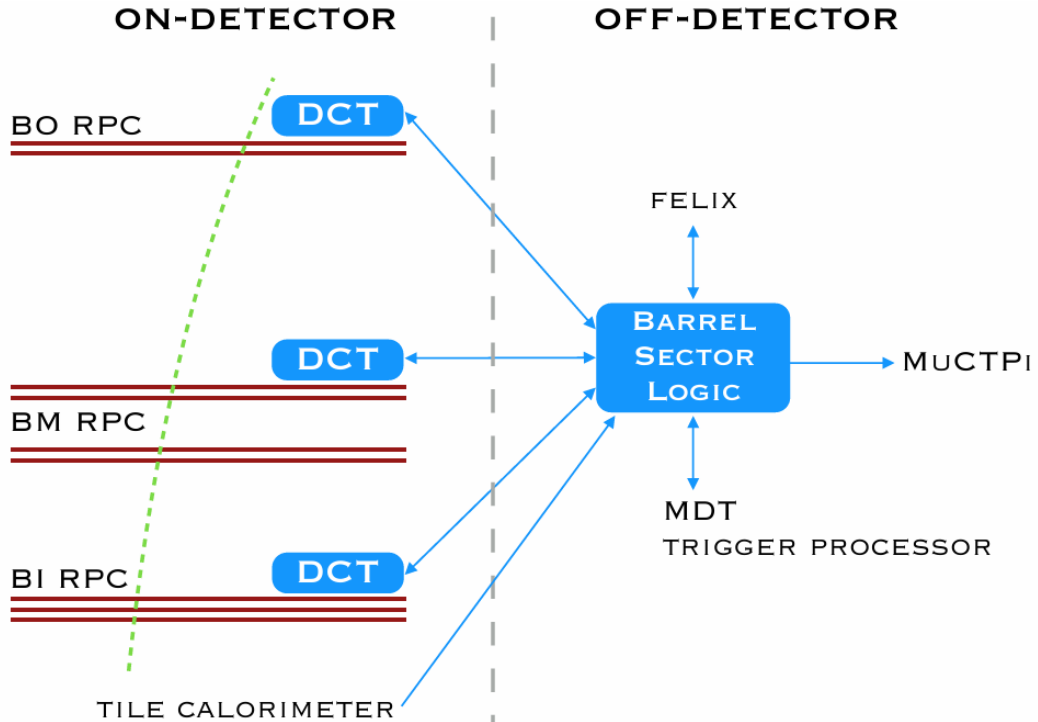


Figure 13 Scheme of the L0Muon Barrel Trigger system [31].

2.8.2 The ATLAS NextGenTriggers Project

The NGT project [7] of ATLAS was "born" in 2022, when a group of private donors visited CERN. The project aims at uncovering as-yet-unseen phenomena by more efficiently selecting interesting physics events using new technologies that take advantage of neural network optimisation, quantum-inspired algorithms, high-performance computing and FPGA techniques.

One of the work packages related to the goals of the ATLAS NGT project is the enhancement of the L0Muon trigger system [32].

The current proposal is to further take advantage of the superior spatial resolution of the MDT chambers by reducing the amount of RPC chambers required to provide seed information to the MDT Trigger Processor to as little as one. This could lead to better performance stability of the L0Muon trigger system, as the degradation, due to aging, of the RPC detectors would be minimised.

Another goal of the L0Muon trigger enhancement work package, which is also where the contribution of this thesis is found, is to directly trigger on non-pointing signatures from the decay of exotic LLPs.

To achieve this, ML techniques will be utilised in compliance with the available hardware resources, which, with the current proposal, would require the completion of the muon trigger tasks within a maximum latency of $1.3\,\mu\text{s}$.

2.9 Physics concepts

2.9.1 Dark matter

According to estimations, ordinary matter makes up less than 5% of the universe, while the rest is made up of dark matter [33] and dark energy. Dark matter does not emit or reflect light, while there has been no observed interaction of dark matter with any known particle so far.

The lack of dark matter signals in different experiments points towards alternative explanations like hidden sectors. Hidden sectors connected very feebly with the SM via "mediator" particles could explain dark matter, being then called dark sectors, and other unexplained phenomena. These "mediator" particles, acting as *portals*, could be produced by the interactions of SM particles in the LHC and thus, should be able to decay back into the same particles or a pair of dark matter particles. The spin of the particle characterises the type of the portal.

2.9.2 Long Lived Particles

Particles in the SM have various lifetimes, depending on factors like their mass and couplings. Even though the LHC was designed for new heavy, prompt particle detection that decay almost immediately, new BSM theories predict the existence of LLPs [34] with longer lifetimes and the ability to travel from a few mm to even up to several meters inside the detector before decaying.

As a result, the decay point of these particles may often not trace back to the Primary Vertex (PV), or the pp IP, and thus a Displaced Vertex (DV) signature is formed.

Displacement here refers to the distance between the PV and the DV formed by an LLP decay or in other words, the distance covered by an LLP inside the detector until the point of its decay.

At the LHC, LLPs are hard to identify, especially to trigger, due to the busy environment and reconstruction bias towards prompt particles, but it is believed that the higher luminosity and updated trigger technology of the HL-LHC will be able to provide a significant boost towards these studies, due to the increased detector acceptance and higher trigger efficiency.

2.9.3 Dark photon

Dark photons [35] are hypothetical new particles from a hidden sector that could be connected to dark matter. They are the mediators of a broken dark $U(1)$ gauge theory that kinetically mixes with the SM hypercharge and can be long-lived [36]. Several modes of production are shown in Figure 14 as well as possible decay modes with SM decay products as shown in Figure 15.

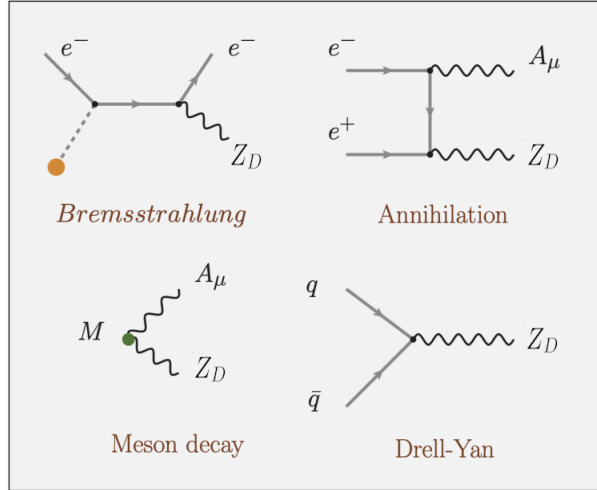


Figure 14 Production of dark photons: Bremsstrahlung, Annihilation, Meson decay and Drell-Yan [35].

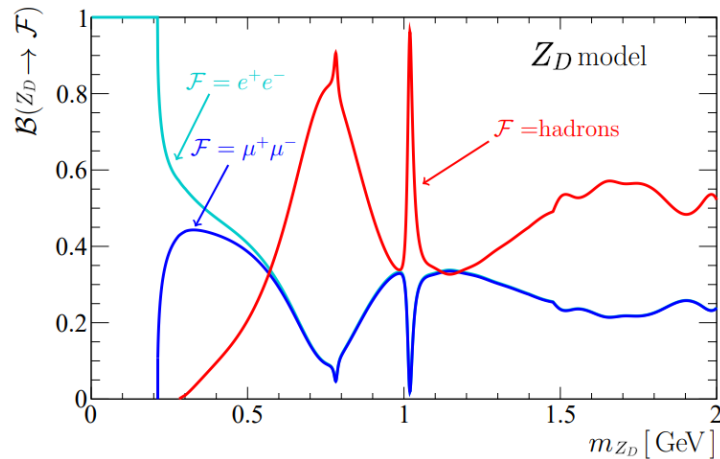


Figure 15 Dark photon decay branching fractions [37].

The Hidden Abelian Higgs Model (HAHM) is an extension of the SM that contains an extra dark gauge field $U(1)_D$, which mixes kinetically with the hypercharge SM gauge field (“vector portal”) and gives rise to a spin-1 mediator known as the dark photon Z_D . The symmetry of $U(1)_D$ is broken by a new dark Higgs field H_D , which mixes with the SM Higgs boson H (“Higgs portal”) and gives mass $m(Z_D)$ to the dark photon.

If there are no hidden-sector states with masses smaller than $m(Z_D)$, the mixing through the vector portal with the SM photon and Z boson causes the dark photon to decay exclusively to SM particles, with a sizable branching fraction to leptons [38] (Figure 16).

Depending on the coupling values, dark photons can be long lived producing *displaced leptons*.

This project uses simulated samples of dark photons produced via HAHM process with varying $m(Z_D)$, in which dark photons decay to a pair of displaced, collimated muons (Figure 16). This is a signature particularly interesting to the NGT working group as it sets the stage for the development of a dedicated algorithm for displaced muon triggering. Something that could be very important towards boosting the available statistics of the related ongoing (or future) and currently statistically limited analyses.

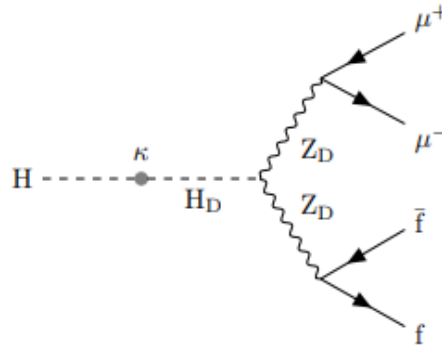


Figure 16 The HAHM dark photon production process.
Here κ is the mixing parameter [38].

3 Method

3.1 Project goal

The goal of this project is developing ML algorithms to be used towards the enhancement of the ATLAS L0Muon trigger in the barrel region, in which neural networks are trained to identify displaced muon candidates, decay products of dark photons produced from the HAHM process. This is done using information from the MDT precision chambers, which will subsequently be passed to the HLT for relevant event selection.

The overall purpose of this application is to advance studies for these exotic signatures by providing a dedicated trigger for corresponding events, many of which would get rejected from the current algorithm due to not meeting the trigger requirements.

The ML models developed in the context of this project are intended for implementation on the FPGAs to be used in the barrel SL as part of the HL-LHC upgrade.

The current proposal is the use of two ML models. One for displaced muon classification, trained on different dark photon samples (in terms of $m(Z_D)$ and displacement) and one which takes into account the output of the first model (to use as positional information) and the event data to estimate the displacement values of the predicted displaced muon candidates.

Two types of neural networks were considered in the context of this analysis:

1. Convolutional Neural Network (CNN)
2. Recurrent Neural Network (RNN)

The different model architectures and functioning principles are covered in their corresponding sections (Sections 3.5.1 and 3.5.2). The main incentive behind the use of these algorithms is that CNNs have already been studied and implemented on FPGAs making them an obvious choice, however the use of RNN model architectures could be proved more suitable for a trigger implementation, due to having the advantage of not relying on sparse image data inputs, which impacts both performance and resource use.

3.2 Constraints

3.2.1 Sources of background

One of the most important challenges, when it comes to muon detection and track reconstruction from the MS data, especially when studying such rare processes, is the presence of background sources, unrelated to the actual collision, generating noise to the data.

These typically are:

- Beam Induced Background [39] (BIB) originating from collision between protons and residual gas in the beam pipe can generate residual particles such as muons to be picked up by the MS.
- Cosmic Ray Background [40] (CRB) generated from cosmic ray showers from which produced muons can penetrate the detector and be picked up by the MS.
- The ATLAS Cavern background [41] which refers to the diffuse background from $p - p$ collisions filling the ATLAS detector cavern. It mainly consists of delayed hits caused by low-energy neutrons and photons. These particles can stay in the detector cavern, creating a *long-lived neutron gas* that can cause background signals even long after the initial event - sometimes even after earlier bunch crossings.

Simulating these sources of background artificially is possible, however, it is not always realistic. More importantly though, background simulations are not yet available for the ATLAS Phase-2 geometry simulations, on which this analysis is based on. Thus, the cleanest way of compensating for this along the lines of this analysis is using actual data from ATLAS itself, after using the muon object reconstruction algorithm to remove muons.

3.2.2 ATLAS asymmetry

Due to restrictions originating from the geometry of the ATLAS MS, the number of tubes (or even monolayers of tubes) in the MDT chambers may vary depending on the barrel station, φ sector and η station. This directly affects the processing of the data used as input to the neural networks (e.g. as will be demonstrated later, different image sizes for the CNN per barrel station), which also causes differences in performance and resource usage depending on the barrel station the model is implemented on.

For these reasons, a case-by-case treatment was necessary to follow the detector's geometry and maintain acceptable performance levels across all barrel stations.

3.2.3 FPGA memory limitations

Since the final stage of this analysis involves the implementation of the ML models on the FPGAs, their limited memory has to be taken into account.

Initially, the neural networks tested were created with best performance in mind, despite their size and computational cost. However, the upper parameter limit of the model, before being converted to "FPGA language", was estimated to be < 1000 and as a result had to be compressed to meet the FPGA requirements, while maintaining performance at acceptable levels.

3.2.4 Latency limitations

As already mentioned in Section 2.8.2, one of the goals of the enhanced L0Muon trigger is to directly trigger on exotic LLP signatures without necessarily relying on RPC seeds using data directly from the MDT chambers.

In order to do that efficiently, the whole "decision making" algorithm to be implemented in the context of this analysis has to meet the upper latency limit requirement of $1.3 \mu\text{s}$.

3.3 Sample generation

For the project needs, Monte Carlo simulated data of the HAHM process were used after passing the ATLAS detector simulation, which uses Run 4 geometry.

The samples were initially generated at Leading Order (LO) using MADGRAPH5_aMC@NLO version 3.5.7, which simulates the hard scattering process, so it includes the calculation of matrix elements and the cross section for the relevant partonic interactions [42].

Then these parton-level events are being passed to PYTHIA 8.314 [43], which takes care of the following parton showering, hadronisation, and underlying event simulation. For PYTHIA 8.314, the NNPDF2.3LO parton distribution functions were used, and the A14 set of tuned parameters was applied [44].

After the event generation is complete, they have to be processed accordingly and converted to n-tuples containing detector hit-level information after passing through the ATLAS detector simulation. Truth level information is also provided, especially useful for particles either not reaching or not being picked up by the detector, e.g. information regarding the dark photons,

such as coordinates of their production and decay vertices, particularly useful for the displacement determination.

The samples available for the purposes of this analysis are demonstrated in Table 2 with respect to their mass and lifetimes.

$m(Z_D)$ [GeV]	0.4	0.4	15	15
$c\tau$ [mm]	25	250	500	1000

Table 2 Available dark photon sample masses with their corresponding lifetimes

3.4 ROOT

ROOT [45] is a unified software package from CERN for the storage, processing, visualisation and analysis of scientific data.

ROOT was used for various purposes throughout this project, some of which are related to the direct visualisation of the detector geometry using the available data or the verification of the correct processing of the data to be used as input for the neural networks.

In Figures 17a and 17b examples are being provided of how n-tuple data could be used for better understanding of the detector's geometry. Apart from gaining a better understanding, these visualisations were used as a cross-check method to make sure the detector simulation worked as expected.

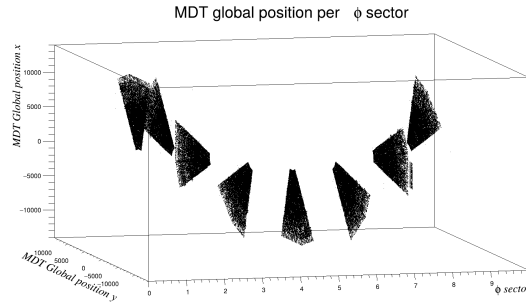
3.5 Neural network methods

A neural network [46] is a ML program, or model, that makes decisions in a manner similar to the human brain, by using processes that mimic the way biological neurons work together to identify phenomena, weigh options and arrive at conclusions.

Every neural network consists of layers of nodes or artificial neurons, an input layer, one or more hidden layers, and an output layer (e.g. as shown in Figure 18). In this analysis, use of CNNs and RNNs is made, so the focus of this section is centered around these sub-categories of neural networks.

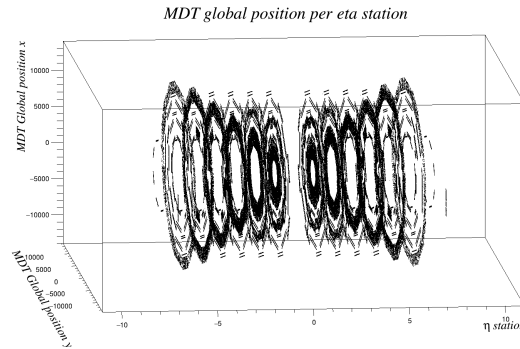
3.5.1 CNN

CNNs [47] are utilized primarily for image recognition, pattern recognition, and/or computer vision. These networks harness principles from linear algebra, particularly matrix multiplication, to identify patterns within an image. With that said, it becomes evident why a CNN could be proved very useful towards this study. The signal studied here is of muons, which tend to leave a track when passing the detector, and, more specifically, collimated displaced muons, meaning this track pattern is usually expected to be observed twice in an image depiction of an event. However, one downside regarding the implementation of a CNN is that they can be computationally demanding, requiring Graphical Processing Units (GPUs) to train models. In the context of using a CNN for triggering, where decisions need to be made almost instantaneously, this caveat is not to be neglected.



(a) Plot of the global positions of simulated hits on MDTs separated by φ sectors, allowing the visualisation of the 8 φ sectors.

(For clarity, here a φ sector includes both large and small ones, the separation is done using the Station Index seen in Section 2.5.3)



(b) Plot of the global positions of simulated hits on MDTs separated by η sectors, allowing the visualisation of the η stations ranging from -7 to $+7$.

Figure 17 Visualisation plots of the detector obtained through the use of ROOT from simulated HAHM process sample hits on MDTs

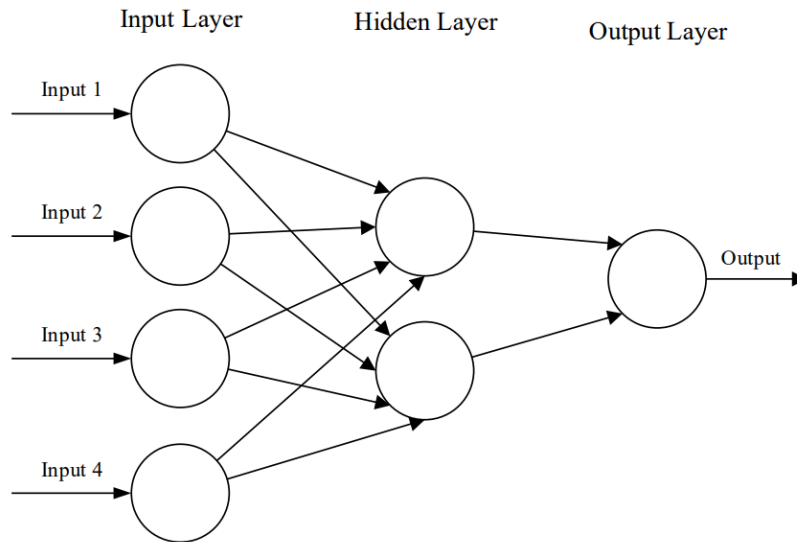


Figure 18 An example of a feed-forward neural network with an input, a hidden and an output layer with their corresponding nodes [47].

Regarding their architecture, as noted earlier, CNNs primarily focus on the basis that the input will be comprised of images. This focuses the architecture to be set up in a way to best suit the need for dealing with these specific type of data.

One of the key differences with respect to conventional feed-forward neural networks is that the neurons that the layers within the CNN are comprised of are organised into more than one dimension. In this analysis, 2-dimensional convolutional layers are used, so the overall shape goes along the lines of (*height, width, number of features*), where in the case of this project, the input images have one feature and that is the drift distances measured across the hit MDTs. The specific inputs and neural network architectures used are covered at the corresponding section regarding implementation.

When the input data are passed to the CNN, a number of filters (also referred to as *kernels*) has also to be set (determining also the depth of the output), which act as feature detectors.

A feature detector is a 2-dimensional array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3×3 matrix. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter [48]. This dot product is then fed into an output array. The filter then shifts by a stride, repeating the process until the kernel has swept across the entire image. This process is referred to as *convolution*. The final output from the series of dot products from the input and the filter is known as a feature or activation map.

Finally, a *fully-connected* layer will then perform the same duties found in standard neural networks and attempt to produce prediction scores from an activation function used according to the goal of the network.

3.5.2 RNN

RNNs [49] are a type of neural network architecture that is mainly used to detect patterns in a sequence of data. In feed-forward neural networks, usually, the information flows in one direction from input to output. However, in an RNN information is fed back into the system after each step.

Even if the uses of RNNs are more commonly known for numerical time-series data (e.g. in the stock market or sensors), they are also applicable to images, if these get respectively decomposed into a series of patches and treated as a sequence.

In neural networks, the overall objective after each epoch is finding the global minimum of a loss function (related each time to the task at hand). For this purpose, gradients are being introduced and a gradient descent algorithm is applied in order to find that minimum, reach the output and update the weights among the neurons to continue to the next iteration.

In RNNs though, information is passed through time and the output of each node in the sequence is propagated back to update the weights (not just the one before the output layer), meaning one could estimate the loss function not only once the output layer is reached, but rather after each step in the sequence [50]. The problem with this arises when in that potentially long sequence of neurons, there are very small weights. Because, due to the matrix multiplication taking place, having to multiply by a small weight continuously will eventually lead to very small values for the gradient and in the end vanish, making it impossible to update the weights per epoch.

To avoid such issues in this analysis, some specific types of RNNs were used that take care of this caveat. More specifically, Long Short Term Memory (LSTM) and Gate Recurrent Unit (GRU) RNN architectures were tested.

LSTMs make use of a memory cell which is controlled by three gates: the *input* gate, the *forget* gate and the *output* gate. These gates decide what information to add to, remove from and output from the memory cell allowing LSTM neural networks to selectively retain or discard information as it flows through the network which allows them to learn long-term dependencies. The network has a hidden state which is like its short-term memory. This memory is updated using the current input, the previous hidden state and the current state of the memory cell [51].

GRUs simplify LSTMs by combining the input and forget gates into a single update gate and streamlining the output mechanism. This design is computationally efficient, often performing similarly to LSTMs, and is useful in tasks where simplicity and faster training are beneficial [52]. Since this is crucial when it comes to the design of a trigger algorithm, as will be seen in the corresponding section, GRUs will prove to be very useful towards minimising resources.

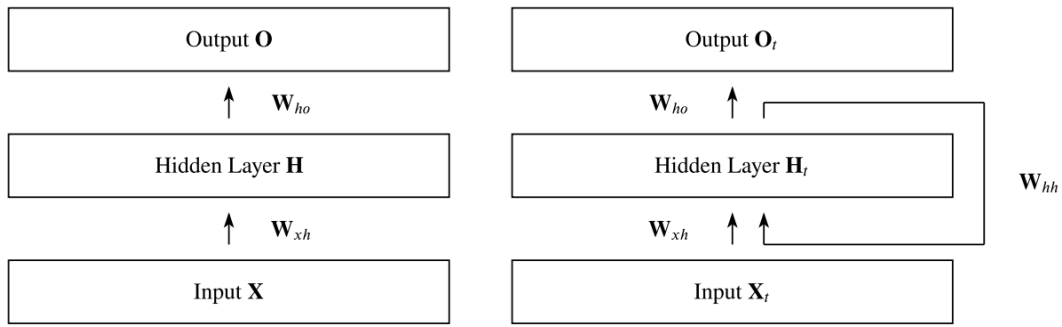


Figure 19 Visualisation of the differences between feed-forward neural networks and RNNs [49].

3.6 FPGAs and hls4ml

So far, reference has been made on FPGAs, however, not much information regarding its characteristics has been provided.

An FPGA is a versatile type of integrated circuit, which, unlike traditional logic devices such as ASICs, is designed to be programmable (and often reprogrammable) to suit different purposes, notably High-Performance Computing (HPC) and prototyping [53].

FPGAs can be adapted to suit multiple use cases, including emerging or experimental purposes, without the need to physically modify or alter their hardware. This reconfigurable versatility is achieved through an array of fixed programmable logic blocks and flexible interconnects that can be configured to perform either complex operations or to serve as simple logic gates. FPGAs also include memory elements, ranging from single-bit flip-flops to very dense memory arrays, for digital data storage within the device.

FPGAs are highly valued for their combination of high performance and extreme versatility. They are particularly useful in applications requiring high performance, low latency and real-time flexibility such as in this case, a trigger for high-energy physics phenomena.

In order to be able to take advantage of what an FPGA can provide and implement the designed ML models, the Python package `hls4ml` has to be used [54] [55]. `hls4ml` provides an efficient and fast translation of ML models from open-source packages (like Keras and PyTorch)

for training ML algorithms to High-Level Synthesis (HLS) code that can then be transpiled to run on an FPGA.

The workflow when making use of `hls4ml` involves converting the trained ML models to HLS code, which is then synthesized to Verilog/VHDL, using Vivado HLS [56].

4 Implementation

4.1 Data preparation

The simulated data, after being extracted from the simulation, are provided as `.root` file n-tuples. They are stored in the form of a ROOT `TTree` the branches of which contain important variables needed to define the properties of detected hits. Some of these variables are related to the local position of the hits in the MDTs (crucial for determining the corresponding drift distances), geometrical properties such as the station index, the η station and the φ sector, the tube number of the registered hits and their corresponding layer in the MDTs etc., as well as information related to the Particle Data Group (PDG) ID, a unique number used for identifying the type of particle producing the hit or even PDG IDs of the particles from the decay of which the particles producing the hits came from (also referred to as *mothers*). "Truth-level" information is also provided, such as positions of PVs and DVs of the particles generated from the simulation (especially useful for the calculation of displacement).

These ROOT files are subsequently being processed accordingly to store relevant information to the analysis. This kind of information is saved in the form of keys in an HDF5 (`.h5`) file. During this processing, the information saved regarding each hit is:

- **Event number:** a unique ID associated with each event, allowing for an event-by-event data selection as well as inspection of the total implementation process to ensure correct application.
- **η station:** η station associated with each hit.
- **φ sector:** φ station associated with each hit.
- **Station index:** barrel station associated with each hit.
- **MDT tube number:** an integer from 1 up to the total number of tubes in the MDT chamber referring to the position of each tube across the layer it is associated with in the chamber (see Figure 11).
- **MDT multilayer number:** an integer of either 1 or 2 referring to which triplet (for BM, BO) or quadruplet (BI) of layers the tube is located in the MDT chamber (see Figure 11).
- **MDT layer number:** an integer ranging from 1 – 6 for BM, BO or from 1 – 8 for BI, referring to the layer of the tube that each hit is associated with in the MDT chamber it was detected (see Figure 11).
- **isMuon array:** An array with length equal to the amount of muon hits per event, if any.

- **drift distance**: calculated drift distance values associated with each hit.
- **displacement**: calculated displacement values associated with the hits produced by displaced muons.

The drift distance calculation is done using the information from the local position coordinates of each hit, with respect to the MDT wire it was detected. With that said, after applying the time correction mentioned in 2.7.5 to the local positions, then the drift distance is obtained from:

$$R_{drift} = \sqrt{x_{loc}^2 + y_{loc}^2} \quad (6)$$

Where R_{drift} represents the drift distance and x_{loc}, y_{loc} the local hit coordinates with respect to the wire of the MDT in which the hit was produced.

The displacement calculation, on the other hand, is a less straightforward calculation as the corresponding *mother* dark photon particles from which the displaced muons came from need to be traced.

For this reason, one needs the PDG ID of the muon *mother* to correspond to the one of the dark photon as it was defined in the simulation (in this project, the dark photon PDG ID was set to: $PDG_{Z_D} = 3000001$). Then from all the "*truth-particle*" *barcodes*, numbers unique to each particle for each event, the ones that match the corresponding "*mother-particle*" *barcodes* are being selected. After knowing each individual *mother*, information is extracted regarding their "*truth-production*" and "*truth-decay*" vertices. There is ID information available regarding all vertices in general, PVs, as well as DVs. Using the *mother-particle barcodes* the PV and DV IDs are being extracted which one can match with the corresponding "*general*" *vertex* ID (per event) and extract their positional coordinates with respect to the ATLAS coordinate system.

The displacement is then calculated as:

$$L_{xy} = \sqrt{(x_{pv} - x_{dv})^2 + (y_{pv} - y_{dv})^2} \quad (7)$$

Where the displacement is noted as L_{xy} , the PV coordinates as x_{pv}, y_{pv} and the DV coordinates as x_{dv}, y_{dv} .

Any analysis in experimental physics has to take into consideration the interference of possible background. As already mentioned in Section 3.2.1, in order to best make up for its presence, real data from ATLAS were taken in n-tuple form. At a point in this project, the models are also trained when single prompt muons are present. For this reason, muon-gun n-tuple samples were also taken.

Both the background and muon-gun samples were processed in the same way as the signal HAHM samples, keeping the same type of information in their corresponding .h5 files.

The only exceptions were that in the background data, muon hits detected from the muon reconstruction algorithm were removed during the processing, meaning the muon array was kept empty for all events in the final background .h5 file.

As for the displacement values, for both the muon-gun and the background samples, since all hits come from prompt particles, displacement was always equal to zero.

With this information available, the events are being processed again one by one and input images in the form of `numpy arrays` are being formatted according to the neural network used

each time. More information on the exact format approach is provided in the corresponding sections for CNN image format and RNN image format (Sections 4.2 and 4.3), but it is worth mentioning here that all different formats of images cover a single φ sector and either large or small barrel stations each time. This is due to the fact that each φ sector hosts one FPGA.

Images containing the following information are being created:

- `events` images, in the form of arrays containing `float` drift distance values in the corresponding hit positions (position is determined by the tube layer and tube per MDT)
- `event_labels` images, having the same shape as the `events` images, but containing the value of 1 wherever a desired hit is to be identified by the neural network and the value of 0, wherever the corresponding entry represents an empty tube or a background hit.
- `displacement` images, having the same shape as the `events` images, but containing displacement values only for hits that correspond to displaced muons.

These images are created for all three sample types, namely the HAHM process, the real background and the muon-gun samples. Then, the `events` images of the HAHM process events are being merged with the corresponding ones of the real background and, if necessary, muon-gun samples.

At this point, the strategy first mentioned in Section 3.1 is provided as a reminder to the reader. The general idea is for two models to be trained and tested. The first one is to perform a classification task on the corresponding entries (pixels) of the input images and the second is to use the predictions of the first neural network as positional information in order to regress on displacement. Three variations of the first model are trained and tested with the overall purpose being to see which one performs better toward displaced muon identification.

These are:

1. **Case 1:** identification of muon hits in general on event images containing displaced muons from the HAHM process, single prompt muons from a muon-gun sample and real background data from ATLAS.
2. **Case 2:** identification of only displaced muon hits on event images containing displaced muons from the HAHM process, single prompt muons from a muon gun sample and real background data from ATLAS.
3. **Case 3:** identification of displaced muon hits on event images containing displaced muons from the HAHM process and real background data from ATLAS.

Depending on the case listed above, the `event_labels` array is being processed according to Table 3.

Label	Case 1	Case 2	Case 3
1	All muon hits	Displaced muon hits	Displaced muon hits
0	real background hits	real background + prompt muon hits	real background hits

Table 3 `event_labels` array values according to the neural network goal case

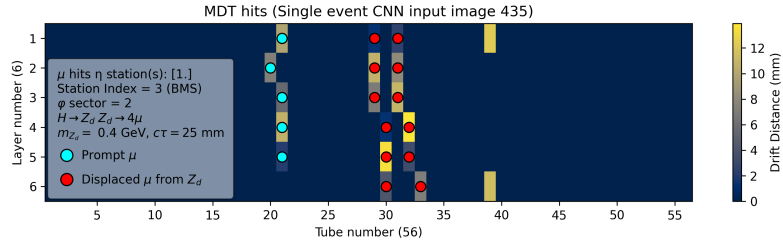
After the preparation of the input images, the final step is downsampling the dataset. This is done because of the following reasons:

- There are a lot of images containing only zero labels
- Need to train the neural networks on both finding where the labelled hits are and also if there are any
- Use an optimal $\frac{\text{images with signal hits}}{\text{all images}}$ ratio for the final input dataset to the neural network.

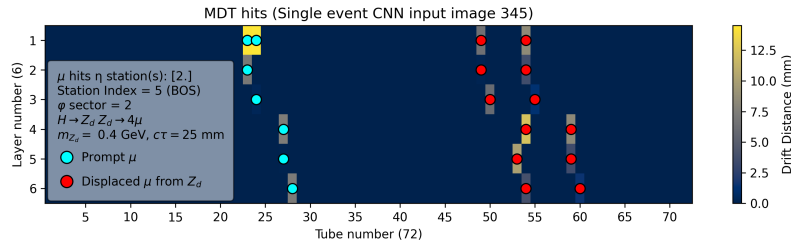
With the downsampling process done, an `.npz` file is created containing all the relevant images mentioned above, as well as some extra information mainly for visualization purposes not related to the training of the neural networks (e.g. η stations, φ sectors, barrel station, etc.).

4.2 CNN image format

The input images (described in the previous section) for the CNN models are representations of the MDT chambers in a specific η region, one φ sector and a specific barrel station, where one pixel corresponds to a single tube. The overall shape of the arrays containing the images is $(N, N_{layers}, N_{tubes}, 1)$, where N is the number of event images, N_{layers} is the number of total tube layers found in an MDT chamber of the region, N_{tubes} is the total number of tubes found in the η region covered and the final 1 refers to the fact that all these images contain one feature (e.g. drift distance, label or displacement). Examples of visual representation of CNN images are provided in Figures 20a and 20b.



(a) A visual representation of an example CNN input image for BMS, η station 1, φ sector 2. (φ sector 2, η station 1 MDT chambers of BMS have 2×3 layers of 56 tubes each)



(b) A visual representation of an example CNN input image for BOS, η station 2, φ sector 2. (φ sector 2, η station 2 MDT chambers of BOS have 2×3 layers of 72 tubes each)

Figure 20 Visualisation of input CNN images depending on covered detector geometry

A representation of how the labelling is done for the CNN images according to the neural network goal case (listed in Section 4.1) is provided in Figures 21a, 21b and 21c.

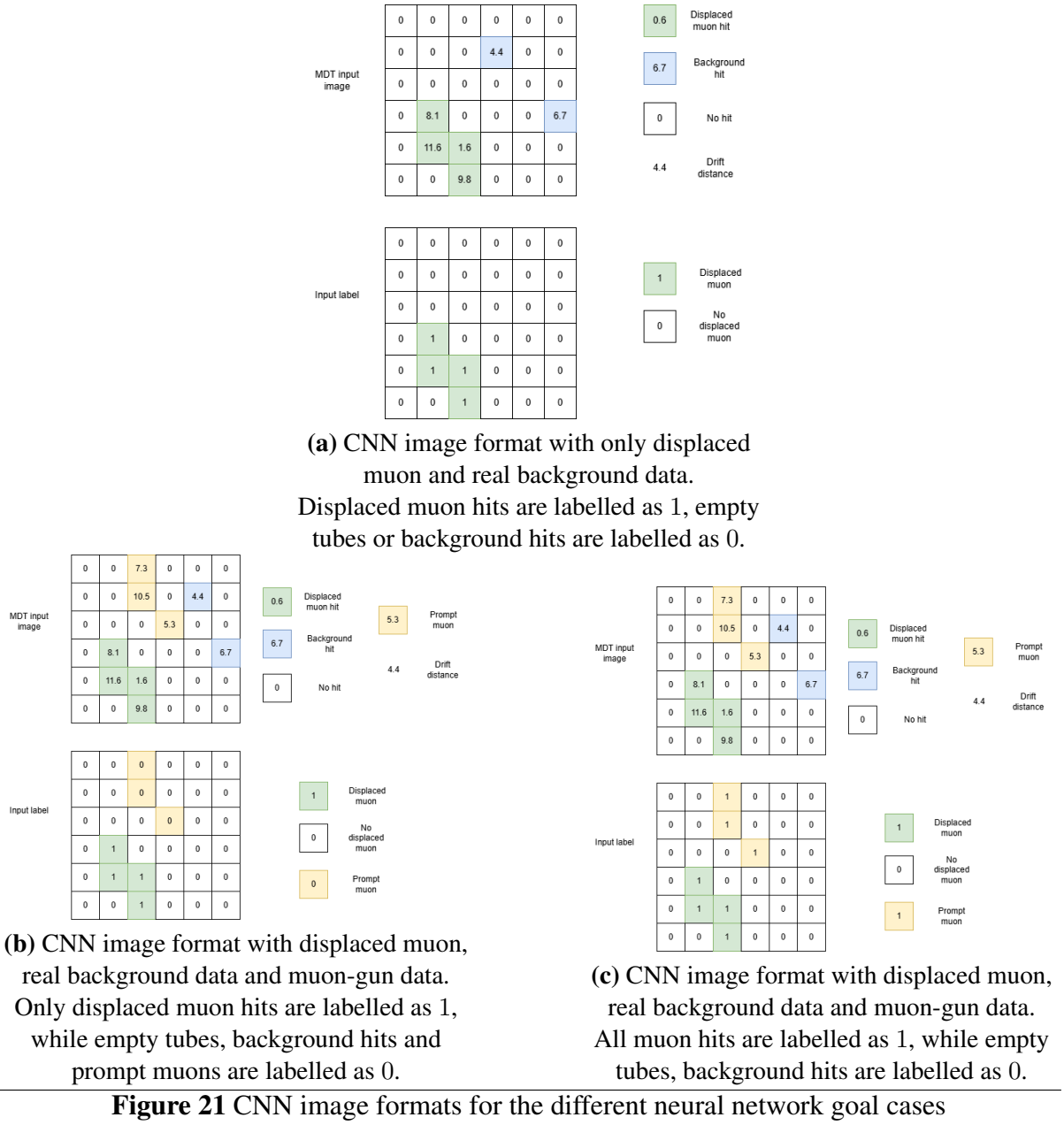


Figure 21 CNN image formats for the different neural network goal cases

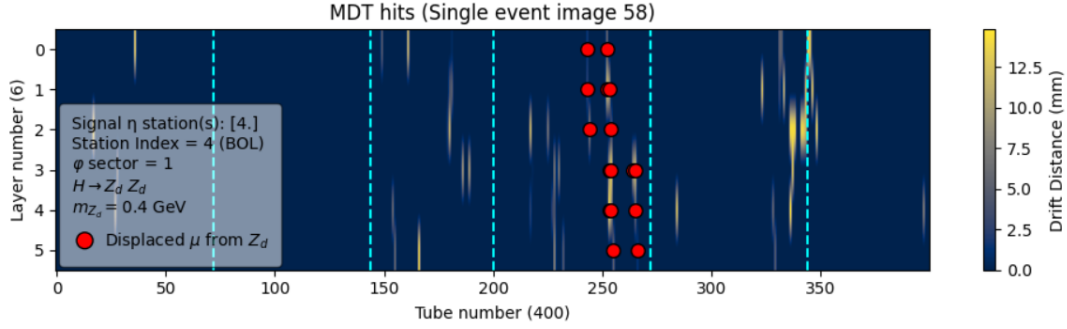
Two approaches were chosen for the CNN images. These were:

- Images of tube dimension equal to the maximum N_{tubes} encountered in a single φ sector and side of the full η range, meaning either only positive or only negative η stations, of the represented barrel station (will be referred to from now on as **half eta** images).
- Images of tube dimension equal to the maximum N_{tubes} seen across all MDT chambers of a specific barrel station in a single η station and φ sector. These images cover only either a single η station or an overlap region between two η stations (from now on referred to as **single eta** images).

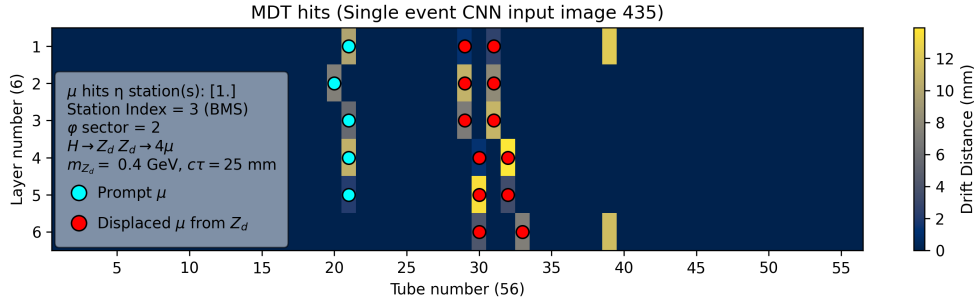
Whenever N_{tubes} in the area covered by the image is smaller than the tube dimension, *padding* is used, meaning the extra entries are filled with zeroes.

Detailed information regarding the N_{tubes} depending on barrel station, η station and φ sector

is provided in Tables 14 - 19 of the *Appendix*. Examples of *half eta* and *single eta* images are provided in Figures 22a and 22b respectively.



(a) Example of a CNN *half eta* input image covering positive η stations. The dashed lines represent the borders between the different η stations.



(b) Example of a CNN *single eta* input image.

Figure 22 Visual comparison between CNN *half eta* and *single eta* images.

The motivation behind having two options for the CNN is related to the consumption of computational resources, data sparsity as well as having enough tube area to cover the relevant event properties, i.e. the collimated behaviour of a pair of displaced muons originating from an HAHM dark photon.

For example, initially only *half eta* images were taken, because they contain more information regarding the event they are covering, but for low-displacement samples the relevant information was found in only a few tubes (less than a single η station), thus leading to unnecessary resource consumption, which negatively affects latency. However, as shown later especially for heavier dark photon samples, sometimes *single eta* images may not be enough to capture the pair of collimated displaced muons mainly due to the opening angle of the pair highly depending on the dark photon mass.

Also, a major issue in the CNN approach is the data sparsity encountered in each image. In every image, the CNN has a lot more entries that are labelled as 0 compared to those labelled as 1, which is also where the interest is on. This means that it has a much harder time identifying "ones" compared to "zeroes". Thus, *half eta* images, even though they provide more information, they also increase resource consumption and data sparsity as they contain much more "zeroes".

On the other hand, *single eta* images contain less information, but reduce drastically the consumed resources, as well as the data sparsity.

4.3 RNN image format

The input images for the RNN include information from all 20 MDT layers ($BI + BM + BO \equiv 2 \times 4 + 2 \times 3 + 2 \times 3 = 20$) and cover only one η station of a single φ sector each time. They are in the form of 20×5 arrays where the dimension of size 5 is used to store the relevant information of up to 5 hits per layer. This information is the tube of the hit on the layer of detection and the corresponding feature, i.e. the drift distance for the `events` images, the labels (either 0 or 1) for the `event_labels` images and the displacement for the `displacement` images.

The number of 5 hits was chosen to limit the consumption of resources and also because usually one encounters less than 5 labelled hits per layer, again depending on the neural network goal case.

After creating the images and downsampling them, an extra step is followed for the RNN, which is to chronologically order them, so as to take advantage of the capability of the RNN to extract information from sequential data. An example of an RNN input image is provided in Figure 23.

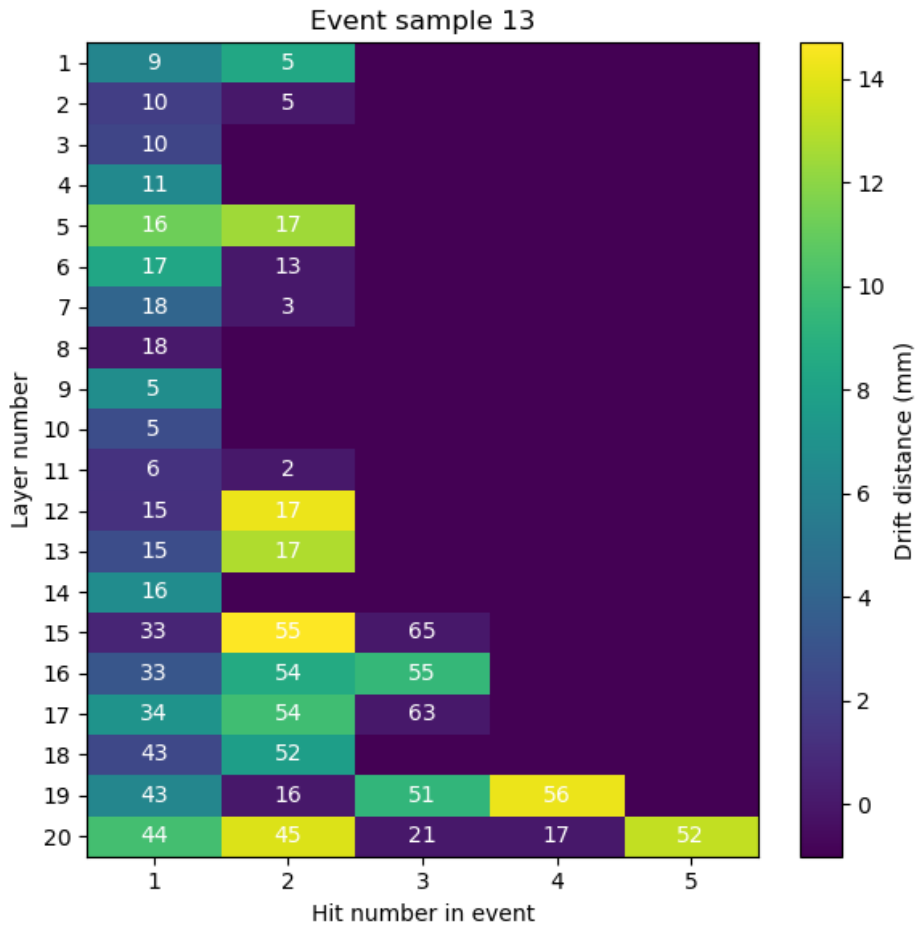


Figure 23 Visualisation of a formatted RNN input image.

4.4 Neural network implementation

4.4.1 Model preparation

After preparing the input data accordingly for each neural network type, a model architecture is set using an input shape that matches the images to be trained and tested on. Here, the layers comprising the model are defined. Exact architectures for the CNN or the RNN used in this analysis are provided in Sections 4.4.6 and 4.4.7.

The model is then compiled with the following properties set:

- **Loss function** [57]: a type of function used to measure the performance of an ML model by calculating the deviation of its predictions on the given input from the actual value or *ground truth*.
As discussed in the previous section, the data sparsity is severe in the images of this analysis, thus in order to best deal with it the choice of a suitable loss function is essential.
- **Optimizer** [58]: an essential component of neural networks that is responsible for updating the weights of the neural network during the training process.
- **Learning rate** [59]: a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. It determines the size of the steps taken towards a minimum of the loss function during optimization. An optimal learning rate should be low enough for accurate convergence while high enough for reasonable training time. Smaller rates require more training epochs, potentially yielding better final weights, whereas larger rates can cause fluctuations around the optimal solution.

With these defined the model is ready to receive input data to fit its predictions on.

4.4.2 Dataset splitting

The goal of the neural networks are to receive as input the `events` images containing the drift distance values of the detected hits, as well as the `event_labels` images, containing the labelled data and try to learn how to recognise the labelled hits based on the recorded drift distances. However, in order to do that, before training the neural network, it is important that the input dataset is split into three separate datasets. These are:

- **Training dataset**: a partition of the dataset used for the training process of the neural network. The neural network tries to learn the relations between the `events` and the `event_labels` images, so that it can predict the labelled hits when knowing the drift distances in the MDTs.
- **Validation dataset**: a partition of the dataset, used to test the network's performance after each training epoch, in order to fine-tune its parameters and weights, before the training procedure is completed. This dataset allows the neural network to generalise better and thus perform better on unseen datasets.

- **Test dataset:** a partition of the dataset saved for testing its final performance on previously unseen data. During testing, the model uses the `events` images of the test dataset and makes predictions on the unseen `event labels` of the test dataset. The comparison between the true `event labels` and the model's predictions allow for an estimate of its performance.

The exact percentages of each dataset with respect to the full dataset are not fixed, but a general practice that is also followed here is 70% for training, 10% for validation and 20% for testing.

4.4.3 Displacement threshold

Since the main purpose of this analysis is to provide a dedicated trigger for displaced muons, it is important that the models are also tested on identifying displaced muon hits, when the displacement is above a certain threshold value. This is because displaced muons originating from the decay of dark photons characterised by low displacement, are very similar to prompt muons and thus very difficult to distinguish.

For this reason, as an additional part of this analysis, when the neural network goal case is to identify only displaced muons, neural networks will also be trained and tested after the application of a displacement threshold value.

When applied, a second downsampling will occur so that again the desired $\frac{\text{images with signal hits}}{\text{all images}}$ ratio is met. A plot of the displacement distribution across the different dark photon mass samples is provided in Figure 24, on which the choice of threshold will be based on.

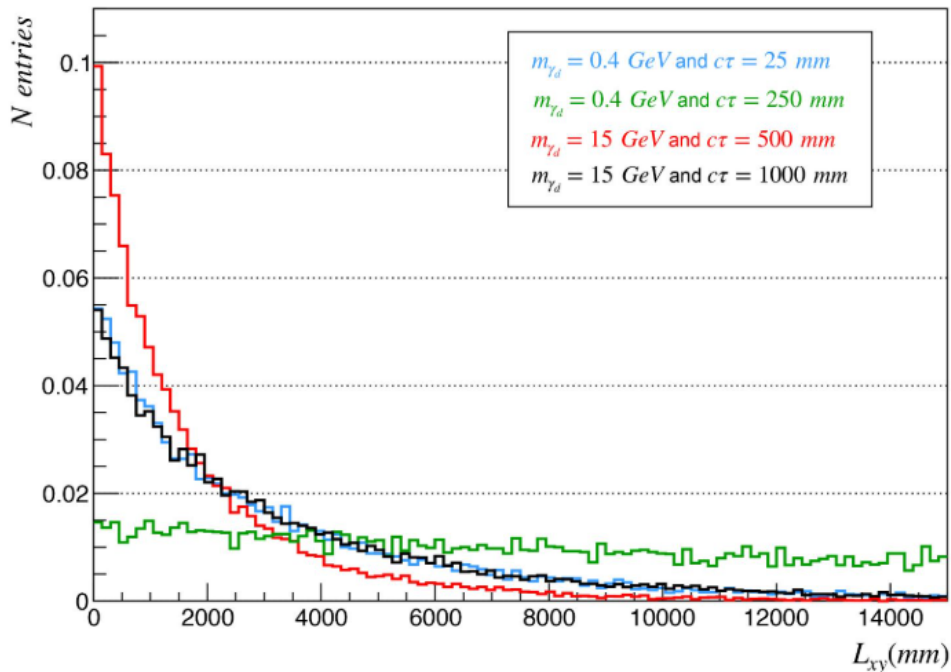


Figure 24 Displacement distribution of the various various dark photon mass samples normalised to unity.

4.4.4 Normalisation

Another important practice that should be followed before training and testing any model ML model is making sure the input dataset is normalised.

Normalisation is a mapping, and scaling method that helps forecasting and prediction models become more accurate by transforming the initial input data range into a new, standardized range and is especially useful when the input data are characterised by different units or magnitudes across different features. Finding a common scale for the data while maintaining the intrinsic variations in value ranges is the main goal of normalisation and this usually entails rescaling the features to a standard range, which is typically between 0 and 1.

In the context of this thesis, the labelled data are already normalised, since they contain only values of 0 and 1, so normalisation is applied on the drift distances of the `events` images. This is done following the formula provided below, also known as mean normalisation:

$$X' = \frac{X - \mu}{\sigma + \varepsilon} \quad (8)$$

Where X and X' are the initial and normalised values respectively of the input feature, μ is the mean value of the input feature across the whole dataset, σ is the standard deviation of the input feature across the whole dataset and ε is a small number to avoid division by zero.

4.4.5 Model training and testing

At this point, the model is now ready to receive the normalised training data as input and initiate the training procedure. Validation data are also passed here for the validation process after every epoch and other hyperparameters are also defined, such as the ***number of training epochs*** and a ***batch size***, determining the division of the input training data into smaller batches each of which is processed independently before the model parameters are updated.

Depending on the model, additional settings may be used such as the ability to shift the ***sample weights*** based on entries of greater importance, i.e. entries of the `events` images with a non-zero drift distance value (usually indicating a particle hit rather than an empty tube), or useful callbacks, such as ***early stopping*** and ***learning rate reduction*** based on the monitoring of a specific statistic, e.g. the loss of the validation dataset, which might help the model converge better to the desired performance.

With these set, the model is subsequently trained and then tested on the normalised and previously unseen test dataset.

4.4.6 CNN architecture

The initial CNN architecture involved the use of *half eta* images as input and was in general more complex. The `tensorflow.keras` library was used to create, train and test the neural networks. The code used to define the initial architecture of the CNN is provided in Figure 25 for BM and BO image inputs, while the corresponding one for BI is shown in Figure 26:

```

1  def create_cnn_model(input_shape):
2      p_drop = 0.1
3      model = tf.keras.models.Sequential(name = 'CNN_BM_BO')
4      model.add(tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
5          padding='same', input_shape=input_shape))
6      model.add(tf.keras.layers.BatchNormalization())
7      model.add(tf.keras.layers.Dropout(p_drop))
8      model.add(tf.keras.layers.MaxPooling2D((2, 2), padding='same'))
9      model.add(tf.keras.layers.Conv2D(64, (3, 3), activation='relu',
10         padding='same'))
11     model.add(tf.keras.layers.BatchNormalization())
12     model.add(tf.keras.layers.Dropout(p_drop))
13     model.add(tf.keras.layers.MaxPooling2D((3, 2), padding='same'))
14
15     model.add(tf.keras.layers.Conv2D(128, (3, 3), activation='relu',
16         padding='same'))
17     model.add(tf.keras.layers.BatchNormalization())
18     model.add(tf.keras.layers.Dropout(p_drop))
19
20     model.add(tf.keras.layers.UpSampling2D((3, 2)))
21     model.add(tf.keras.layers.Conv2D(64, (3, 3), activation='relu',
22         padding='same'))
23     model.add(tf.keras.layers.BatchNormalization())
24     model.add(tf.keras.layers.Dropout(p_drop))
25     model.add(tf.keras.layers.UpSampling2D((2, 2)))
26     model.add(tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
27         padding='same'))
28     model.add(tf.keras.layers.BatchNormalization())
29     model.add(tf.keras.layers.Dropout(p_drop))
30
31     model.add(tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid',
32         padding='same'))
33
34     return model
35

```

Figure 25 Code snippet for the initial CNN architecture used on BM and BO images

Both architectures follow a structure of what is also referred to as *convolutional autoencoder*, which in this case involves two downsampling (encoding) steps and two upsampling (decoding) steps. The convolutions were performed with different amounts of filters per downsampling/upsampling step, while each convolutional layer was followed by two layers used for regularisation purposes.

Namely:

- **Batch Normalisation** [60]: makes normalization a part of the model architecture and performs the normalisation for each training mini-batch. Batch Normalization allows one to use much higher learning rates and be less careful about initialization.
- **Dropout** [61]: during training, some number of layer outputs are randomly ignored or "dropped out". This has the effect of making the layer look-like and be treated-like a layer with a different number of nodes and connectivity to the prior layer. Dropout makes the training process noisy, forcing nodes within a layer to probabilistically take on more or

```

1  def create_cnn_model_BI(input_shape):
2      p_drop = 0.1
3      model = tf.keras.models.Sequential(name = 'CNN_BI')
4      model.add(tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
5          padding='same', input_shape=input_shape))
6      model.add(tf.keras.layers.BatchNormalization())
7      model.add(tf.keras.layers.Dropout(p_drop))
8      model.add(tf.keras.layers.MaxPooling2D((2, 2), padding='same'))
9      model.add(tf.keras.layers.Conv2D(64, (3, 3), activation='relu',
10         padding='same'))
11     model.add(tf.keras.layers.BatchNormalization())
12     model.add(tf.keras.layers.Dropout(p_drop))
13     model.add(tf.keras.layers.MaxPooling2D((2, 2), padding='same'))
14
15     model.add(tf.keras.layers.Conv2D(128, (3, 3), activation='relu',
16         padding='same'))
17     model.add(tf.keras.layers.BatchNormalization())
18     model.add(tf.keras.layers.Dropout(p_drop))
19
20     model.add(tf.keras.layers.UpSampling2D((2, 2)))
21     model.add(tf.keras.layers.Conv2D(64, (3, 3), activation='relu',
22         padding='same'))
23     model.add(tf.keras.layers.BatchNormalization())
24     model.add(tf.keras.layers.Dropout(p_drop))
25     model.add(tf.keras.layers.UpSampling2D((2, 2)))
26     model.add(tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
27         padding='same'))
28     model.add(tf.keras.layers.BatchNormalization())
29     model.add(tf.keras.layers.Dropout(p_drop))
30
31     model.add(tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid',
32         padding='same'))
33
34     return model
35

```

Figure 26 Code snippet for the initial CNN architecture used on BI images

less responsibility for the inputs. This conceptualization suggests that perhaps dropout breaks-up situations where network layers co-adapt to correct mistakes from prior layers, in turn making the model more robust. In CNNs, a dropout probability p_{drop} of 0.1 – 0.2 is usually applied, if applied at all.

The downsampling was performed using `MaxPooling` layers, which have the effect of reducing the dimensions of each image and thus allow the model to learn more complex aspects, while the corresponding upsampling was done using `UpSampling` layers, allowing to get final images of same dimensions as the input across all filters. The difference between the two models lies at the kernel size of the pooling and upsampling operations due to the dimensionality difference between BI and BM, BO input images which stems from the difference in the N_{layers} found in the BI MDT chambers, when compared to BM or BO.

Finally, the last convolutional layer of 1 filter and kernel dimensions (1×1) with the *sigmoid* activation is used to produce output images of identical shape as the input images, while also

mapping the predicted label values into values between 0 and 1. The predicted label values will also be referred to as $\text{CNN}_{\text{score}}$.

The *sigmoid* activation is provided for context in equation (9):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

Its name originates from its shape when plotted, which is demonstrated in Figure 27.

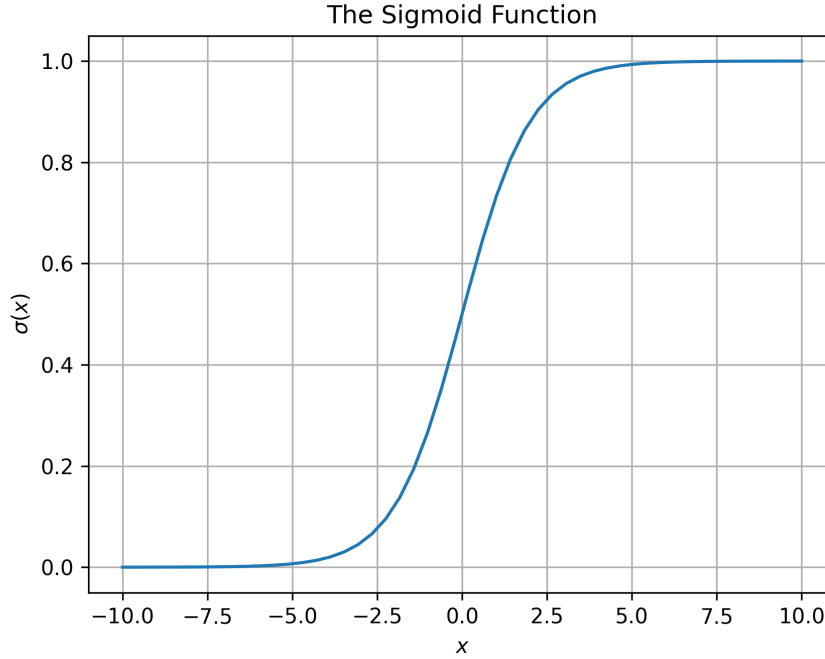


Figure 27 Plot of the sigmoid function.

The results of this model are provided in the corresponding section (Section 5.1), although a comment is provided here in order to motivate the need for a new approach.

Even though, the model showed promising results for the tested barrel stations and dark photon samples, unfortunately, it was too complex (over 180000 parameters) to be implemented on FPGAs, due to their currently limited memory capabilities. However, the take-away message was that the goal itself was achievable, if complexity and computational resources could meet acceptable levels set by the FPGA.

To reduce model complexity and computational resources, a new approach was decided and that involved two key changes.

- Model size reduction, keeping only two convolutional layers of 8 filters each (batch normalisation and dropout are still applied) and the final convolutional layer with the *sigmoid* activation.
- Use of *single eta* images to focus more on η station areas containing useful information (signal) and greatly reduce computational resources due to the smaller pixel count on the images.


```

1  def create_cnn_model(input_shape):
2      p_drop = 0.1
3      model = tf.keras.models.Sequential(name = 'CNN')
4      model.add(tf.keras.layers.Conv2D(8, (3, 3), activation='relu',
5          padding='same', input_shape=input_shape))
6      model.add(tf.keras.layers.BatchNormalization())
7      model.add(tf.keras.layers.Dropout(p_drop))
8
9      model.add(tf.keras.layers.Conv2D(8, (3, 3), activation='relu',
10         padding='same'))
11     model.add(tf.keras.layers.BatchNormalization())
12     model.add(tf.keras.layers.Dropout(p_drop))
13
14
15     model.add(tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid',
16         padding='same'))
17
18     return model
19

```

Figure 28 Code snippet for the final CNN architecture

The new model architecture, which is common for all barrel stations is provided in Figure 28.

As one can see the new model architecture is much less complex and is light enough for potential use on an FPGA. The total parameter count of this model is at 737 parameters, which meets the < 1000 parameters, the estimated requirement mentioned in Section 3.2.3

4.4.7 RNN architecture

As already mentioned in Section 3.5.2, the RNN implementation along the lines of this thesis involved the use of LSTM and GRU architecture. The exact models are provided in Figures 29 and 30.

```

1  def create_rnn_model(input_shape, labels_shape):
2      input_layer = Input(shape=input_shape)
3      x = LSTM(8, return_sequences=True)(input_layer)
4      x = Dropout(0.2)(x)
5      x = LSTM(8, return_sequences=True)(x)
6      x = Dropout(0.2)(x)
7      output = TimeDistributed(Dense(labels_shape[1],
8          activation='sigmoid'), name='time_distributed')(x)
9      model = Model(inputs=input_layer, outputs=output)
10
11     return model
12

```

Figure 29 Code snippet for the LSTM RNN model architecture

In both architectures, two LSTM/GRU layers are used with a dropout layer applied after each

```

1  def create_rnn_model(input_shape, labels_shape):
2      input_layer = Input(shape=input_shape)
3      x = GRU(8, return_sequences=True)(input_layer)
4      x = Dropout(0.2)(x)
5      x = GRU(8, return_sequences=True)(x)
6      x = Dropout(0.2)(x)
7      output = TimeDistributed(Dense(labels_shape[1],
8          activation='sigmoid'), name='time_distributed')(x)
9      model = Model(inputs=input_layer, outputs=output)
10
11     return model
12

```

Figure 30 Code snippet for the LSTM RNN model architecture

layer and a time distributed output dense layer with the *sigmoid* activation.

For the RNN, both architectures yield in general very similar results, but the LSTM one is heavier in terms of resource consumption as it breaks the estimated upper limit of 1000 parameters, specifically reaching 1157 parameters. On the other hand, the GRU architecture seems to be lighter at 957 parameters and since minimal performance difference was noticed, it was the one kept for implementation on FPGAs.

4.5 Converting models with hls4ml

The final step towards the implementation of the developed algorithms is centered around conversion to the language used by the FPGAs through the use of `hls4ml`.

After validating the models on previously unseen and unfiltered data to avoid any input bias, the models are converted using the `hls4ml` library and their performance is compared to their original prior to conversion.

5 Results and Discussion

5.1 CNN results

Here the results of the trained and tested models are presented prior to `hls4ml` conversion. Analytical plots are provided only for one barrel station in this section, in order to avoid repetition of identical practices. The overall performance of the networks across all the tested samples and barrel stations depending on whether they were trained to identify all muons or only displaced muons is captured in Tables 4 - 6 and 7 - 9 respectively.

5.1.1 Detecting all muons

Below the results on BMS are provided for all dark photon samples, while the goal of the CNN is to identify muon hits in general. The loss function used is a combination of the *Focal Loss* (FL) [62] and of the *Dice Loss* (DL) [63], both particularly useful for sparse, imbalanced datasets like the one encountered in this analysis.

Their definitions are provided below:

$$FL = \alpha(1 - p_t)^\gamma \log(p_t) \quad (10)$$

Where α is a balance parameter, γ is a focus parameter for hard, misclassified examples and p_t is the model's estimated probability for the class.

The definition of DL is also provided:

$$DL = 1 - \frac{2y \otimes p_t + \varepsilon}{y + p_t + \varepsilon} \quad (11)$$

Where y are the true labels and p_t is the model's prediction, while ε is a small number to avoid division by 0.

The combination of the two loss functions, referred to as *Combined Loss* (CL) was applied as demonstrated in (12):

$$CL = \alpha \cdot FL + (1 - \alpha)DL \quad (12)$$

Where α is a balance parameter between the two.

In an attempt to tackle high class imbalance two additional measures were taken. A 4 times higher weight was set across the data sample on entries that showed a drift distance value greater than zero, while also a scheduler was used to gradually increase the weight of the true class (labelled hits) after each epoch. Initially both classes start with identical weights and until the end 3 times more weight is placed on the labelled hits.

The models were trained for 50 *epochs*, a *batch size* of 64 and with a starting *learning rate* of 0.01 with the *Adam* optimizer. As a safety mechanic to mitigate overfitting, *early stopping* was used as well as a learning rate reduction mechanic, both while monitoring the *validation loss*.

After training for 50 epochs, the training and validation loss and accuracy curves are provided in Figures 31 and 32 for *single eta* image inputs.

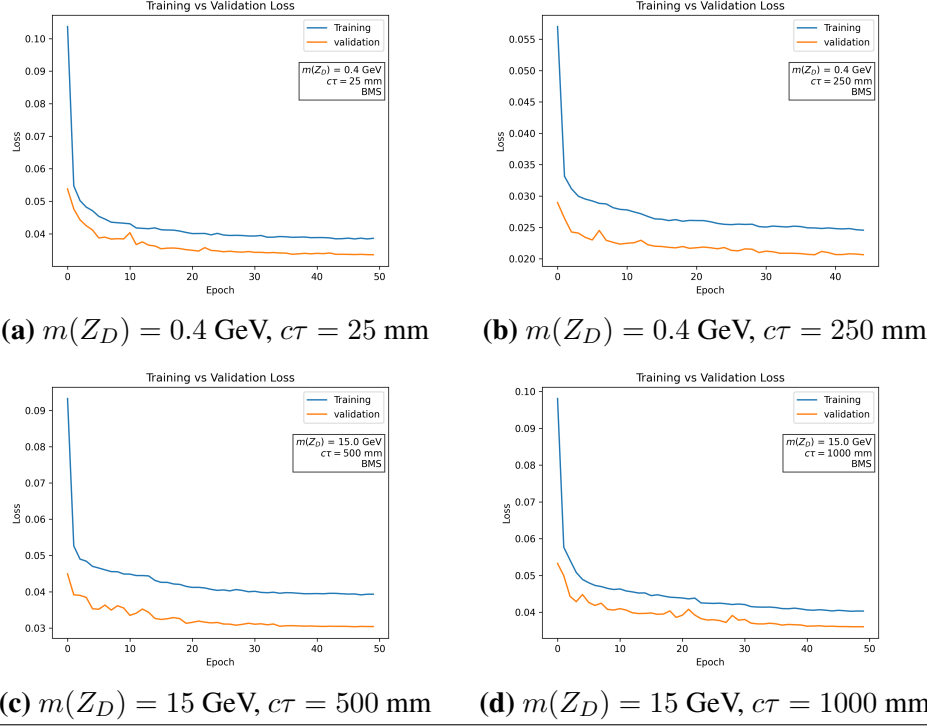


Figure 31 CNN training and validation loss plots over the number of epochs for BMS using *single eta* image input, when the model is trained to identify all muons.

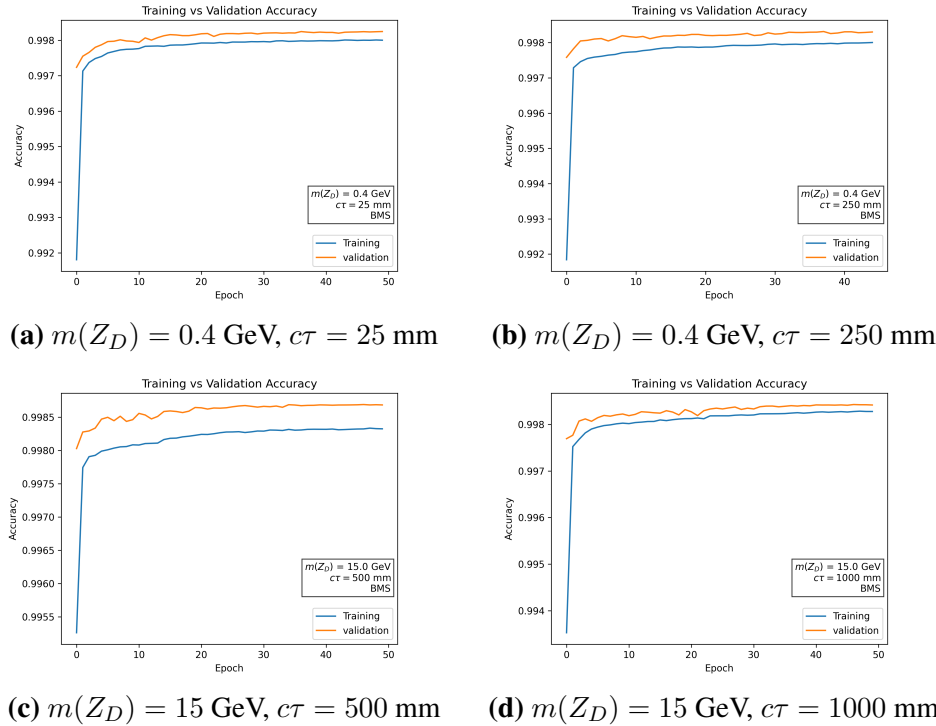


Figure 32 CNN training and validation accuracy plots over the number of epochs for BMS using *single eta* image input, when the model is trained to identify all muons.

From the training and validation loss and accuracy plots, the model seems to be performing similarly both during training and validating as no extreme divergence is observed, an indicator that the regularization methods used take action.

Also both are for all dark photon mass and lifetime samples at sufficiently good levels, but in order to be able to tell whether or not the model is performing well more statistics are required. The $\text{CNN}_{\text{score}}$ distributions are provided across Figures 33 - 35.

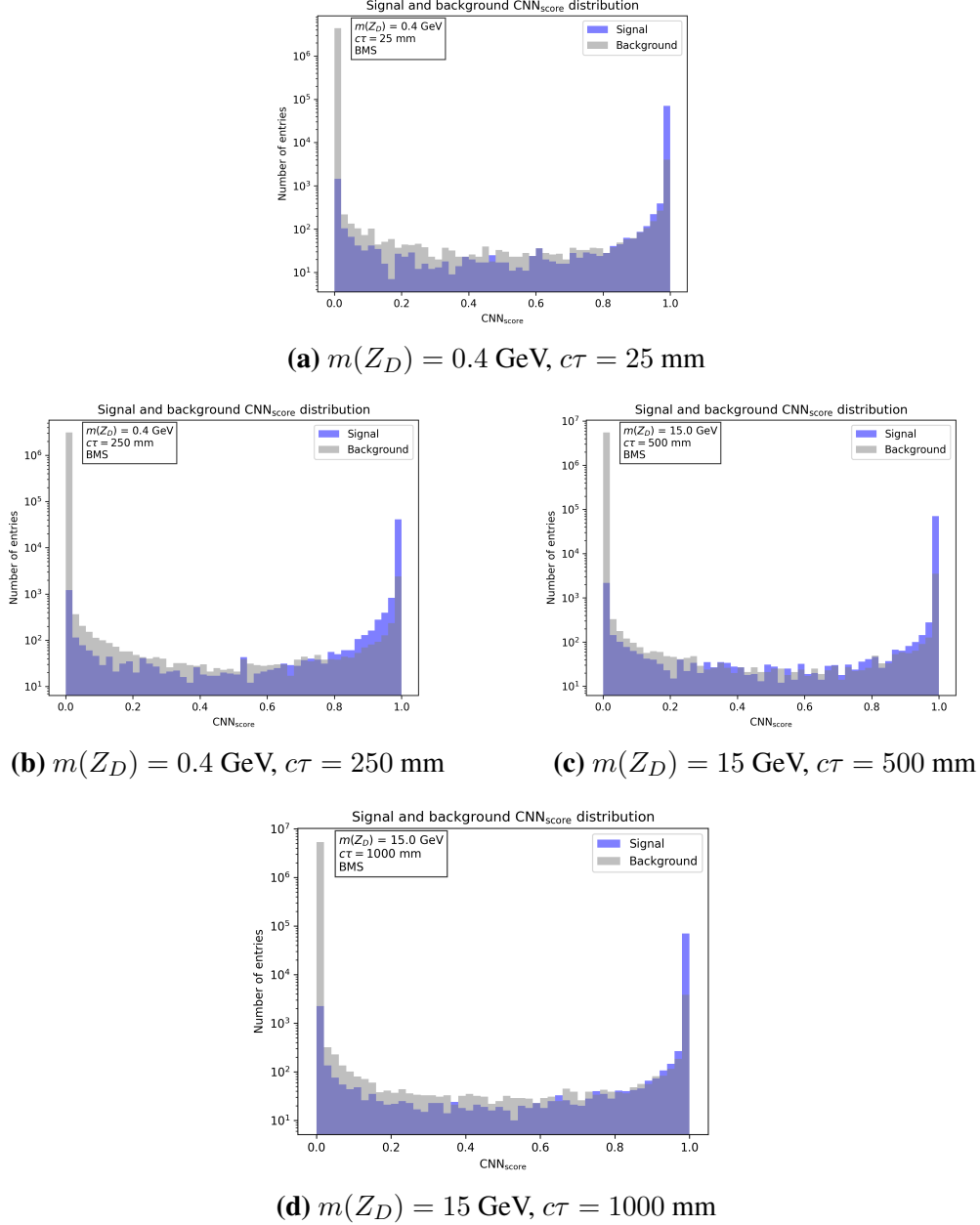


Figure 33 $\text{CNN}_{\text{score}}$ distribution plots for all signal and background entries for BMS using *single eta* image input, when the model is trained to identify all muons.

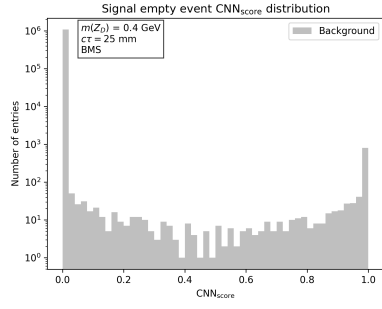
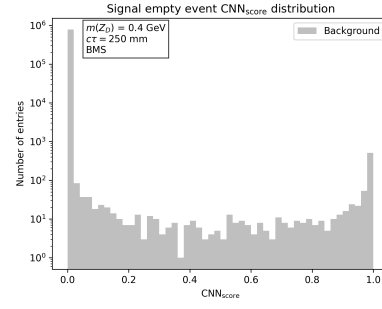
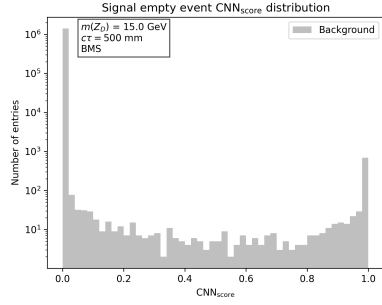
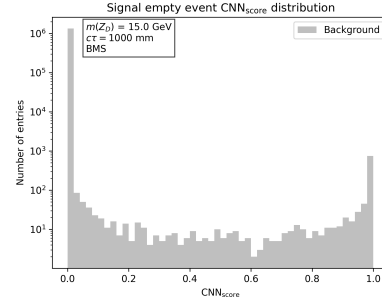
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 34 $\text{CNN}_{\text{score}}$ distribution plot for all entries in empty labelled images for BMS using *single eta* image input, when the model is trained to identify all muons.

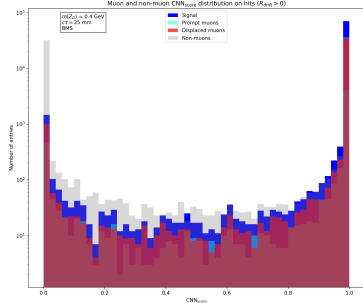
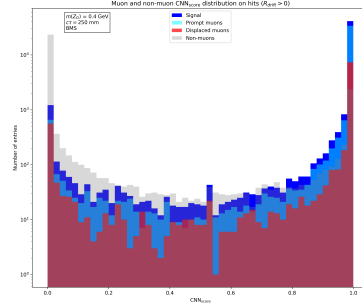
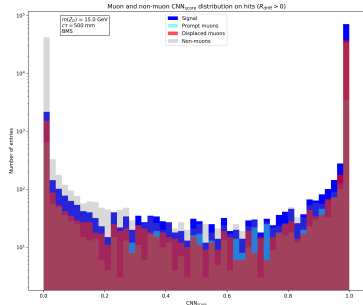
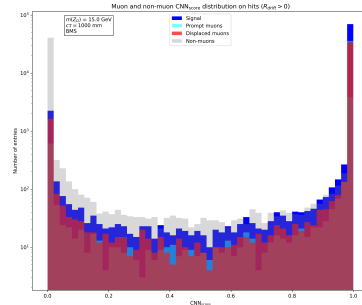
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 35 $\text{CNN}_{\text{score}}$ distribution plots for all entries that correspond to hits (non-zero R_{drift} value) for BMS using *single eta* image input, when the model is trained to identify all muons.

From the $\text{CNN}_{\text{score}}$ distribution plots, one can notice that roughly 1 every 100 muon hits gets hard misidentified as a background hit, while roughly 1 every 1000 entries that is classified as background gets misidentified as a muon.

The most interesting plot though is the one where only hits with a $R_{\text{drift}} > 0$ are only taken into consideration where one notices that roughly 1 out of 10 background hits gets misclassified as a muon.

As for the displaced and prompt muon performance, the network seems to perform similarly for both.

The claims above are also confirmed by the calculated confusion matrices, provided in Figures 36 - 39.

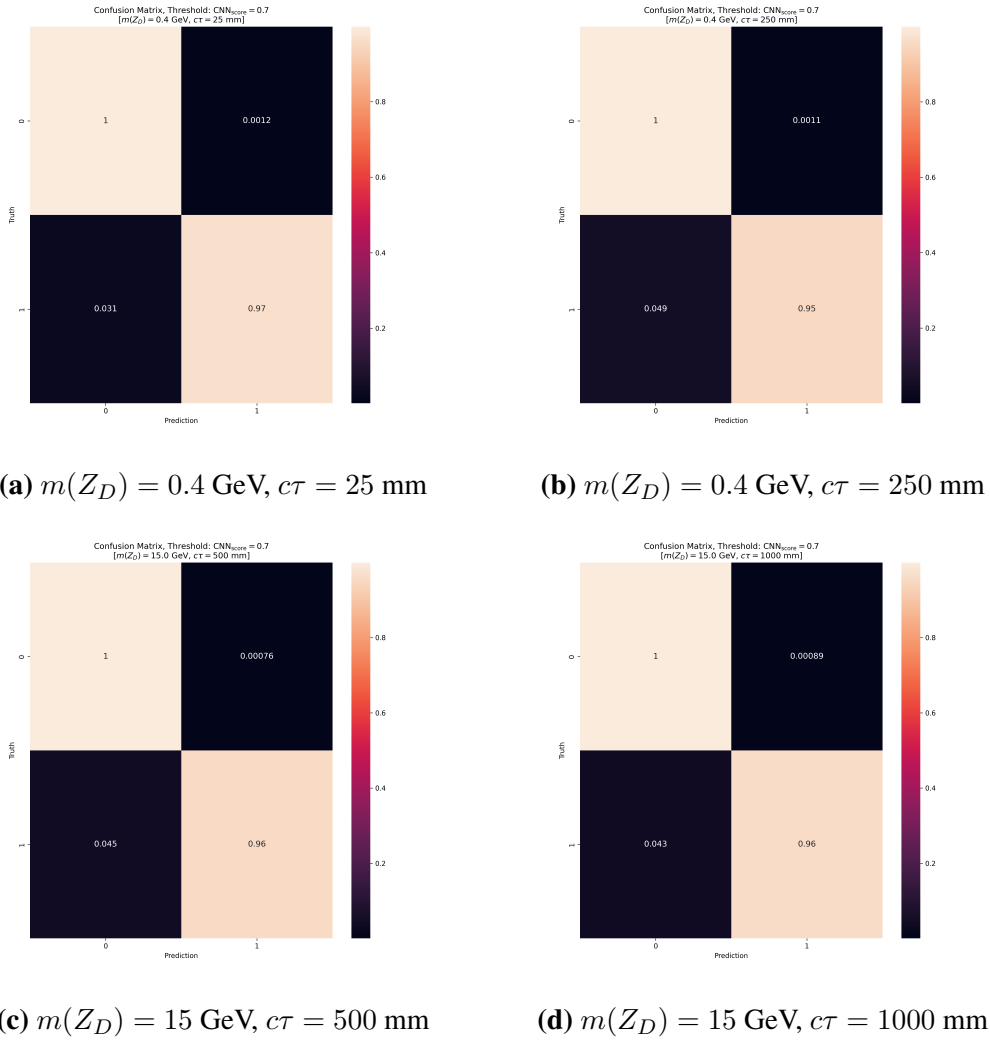


Figure 36 CNN confusion matrix for signal and background classes for BMS using *single eta* image input, when the model is trained to identify all muons.

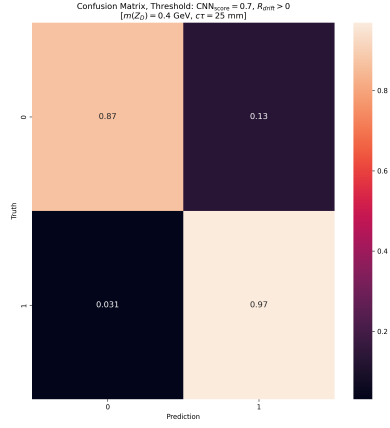
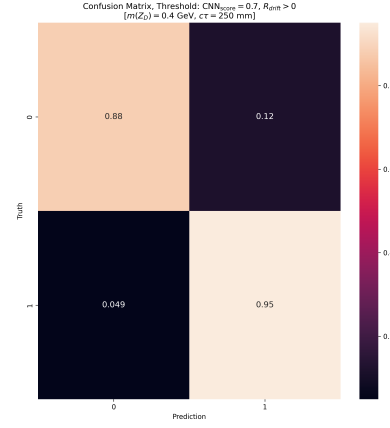
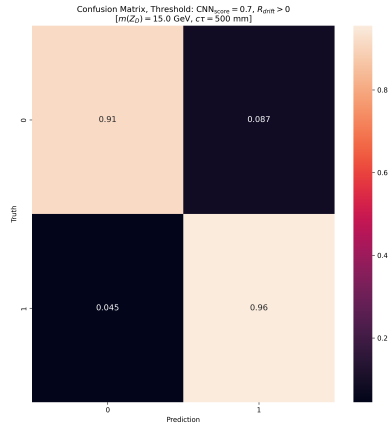
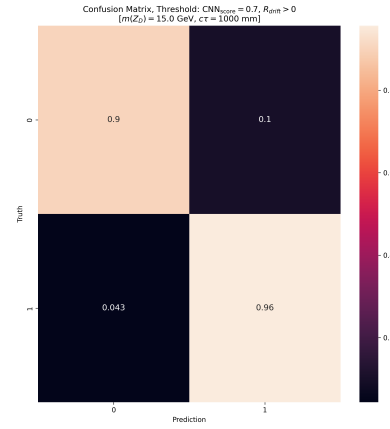
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 37 CNN confusion matrix for signal and background classes, when only counting hits with $R_{\text{drift}} > 0$ for BMS using *single eta* image input, when the model is trained to identify all muons.

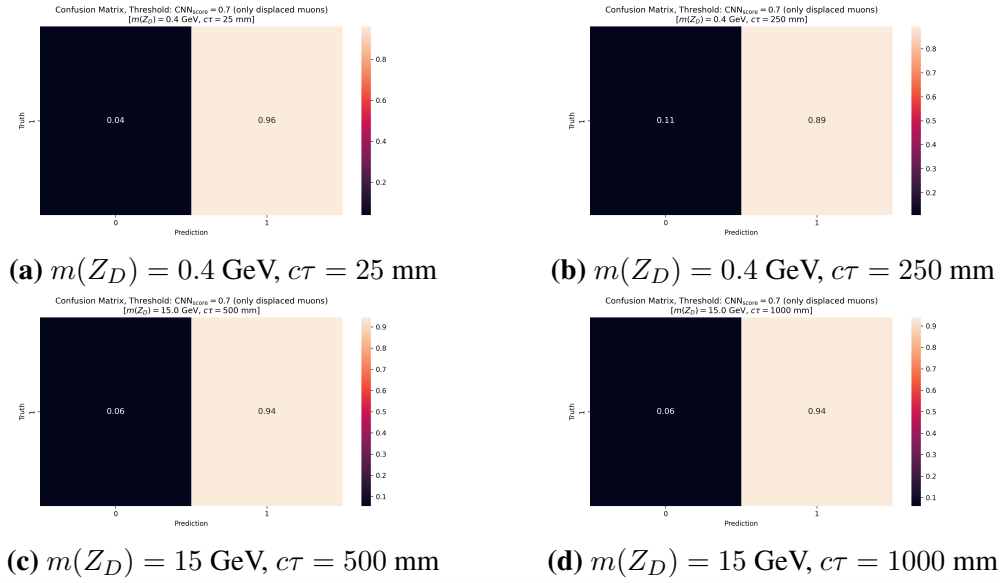


Figure 38 CNN confusion matrix only for displaced muons for BMS using *single eta* image input, when the model is trained to identify all muons.

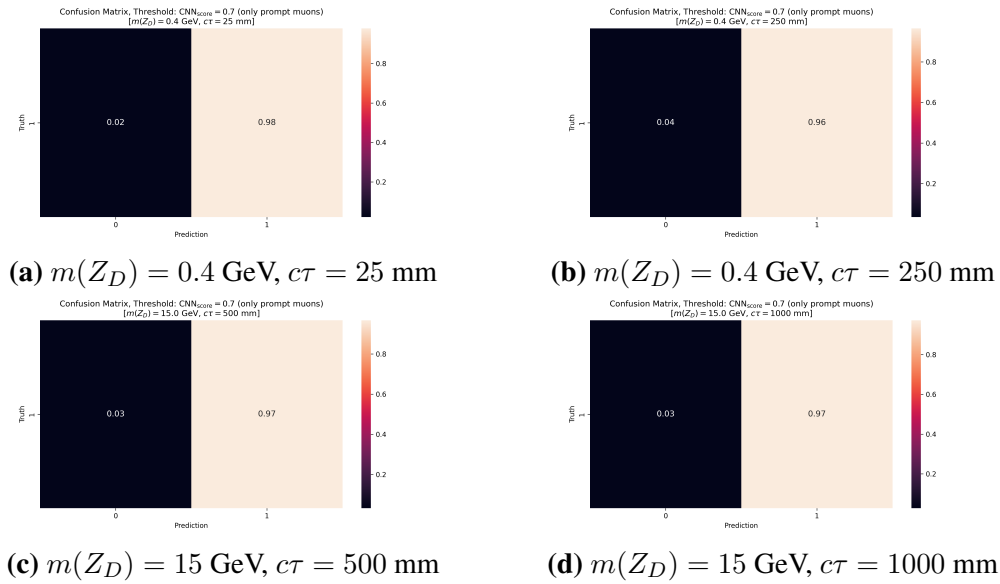


Figure 39 CNN confusion matrix only for prompt muons for BMS using *single eta* image input, when the model is trained to identify all muons.

In order to perform a check on why those hits are getting misidentified the distance in tubes of the hard misclassified hits ($\text{CNN}_{\text{score}} > 0.95$) from the closest muon hit on the same layer was measured, if any. The corresponding plots are provided in Figure 40.

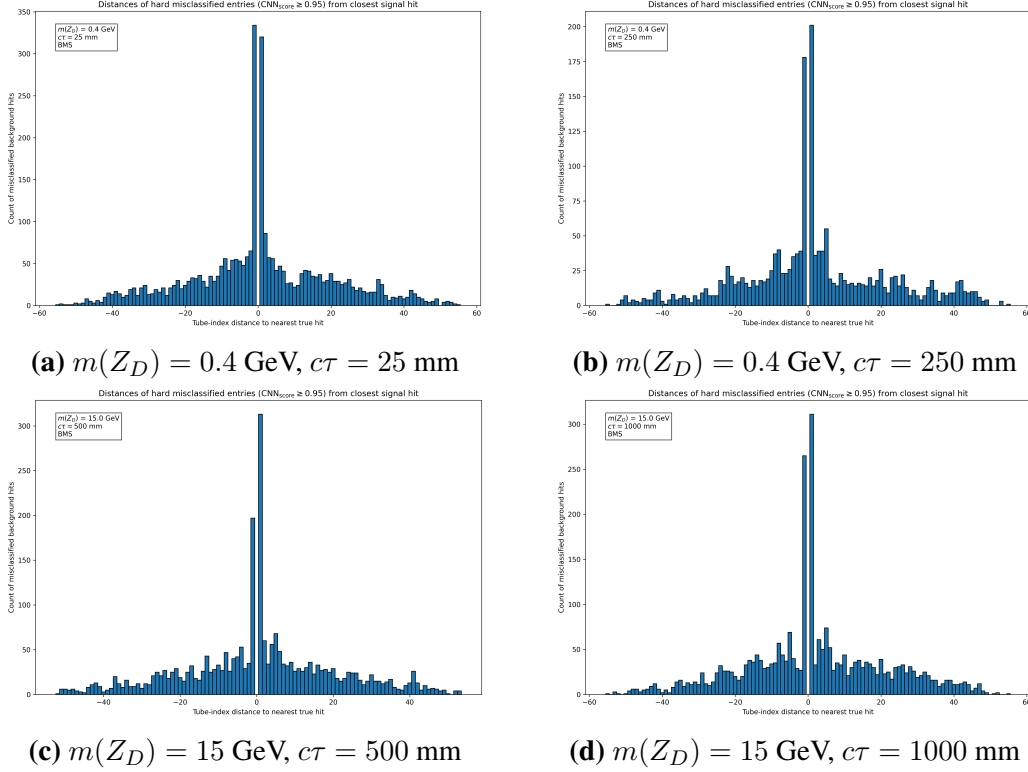


Figure 40 Distances of hard misclassified hits ($\text{CNN}_{\text{score}} > 0.95$) from the closest labelled hit on the same layer, if any, for BMS using *single eta* image input, when the model is trained to identify all muons.

From the distance plots it is evident that, since they are all centered around zero, the most probable reason for hard misclassification is that they are next to a true labelled hit, thus confusing the CNN. The same behaviour is encountered across all samples with the total counts in these plots agreeing to the expected number of hard-misses.

To further inspect the performance of the CNN, metrics such as the the Receiver Operating Characteristic (ROC) curve, defined as the False Positive Rate (FPR) VS True Positive Rate (TPR) and the purity VS efficiency curve are also provided.

Before proceeding with the plots, the relevant definitions are provided:

$$TPR = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (13)$$

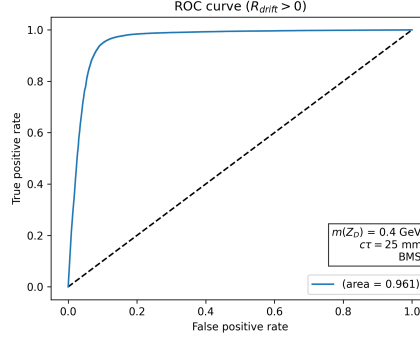
$$FPR = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \quad (14)$$

It is mentioned that the efficiency is equivalent to the TPR.

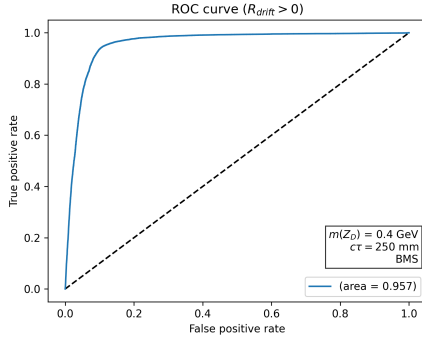
The purity is defined as:

$$\text{Purity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (15)$$

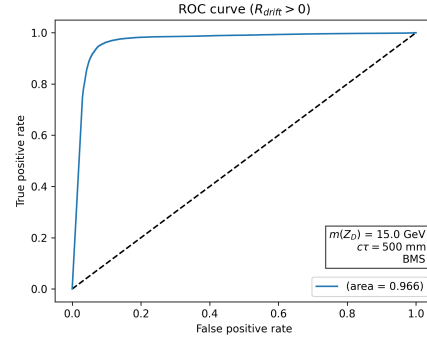
With this information provided, the ROC curve and the purity VS efficiency curves for both signal and background, when considering only hits with $R_{drift} > 0$ are shown in Figures 41 and 42.



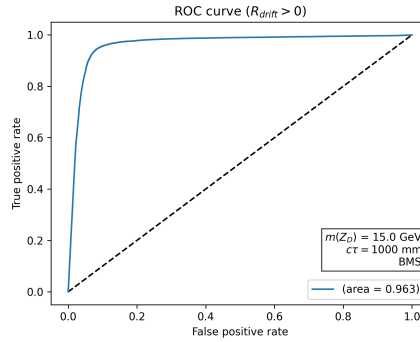
(a) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$



(b) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$



(c) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 500 \text{ mm}$



(d) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 1000 \text{ mm}$

Figure 41 ROC curve for hits with $R_{drift} > 0$ for BMS using *single eta* image input, when the model is trained to identify all muons.

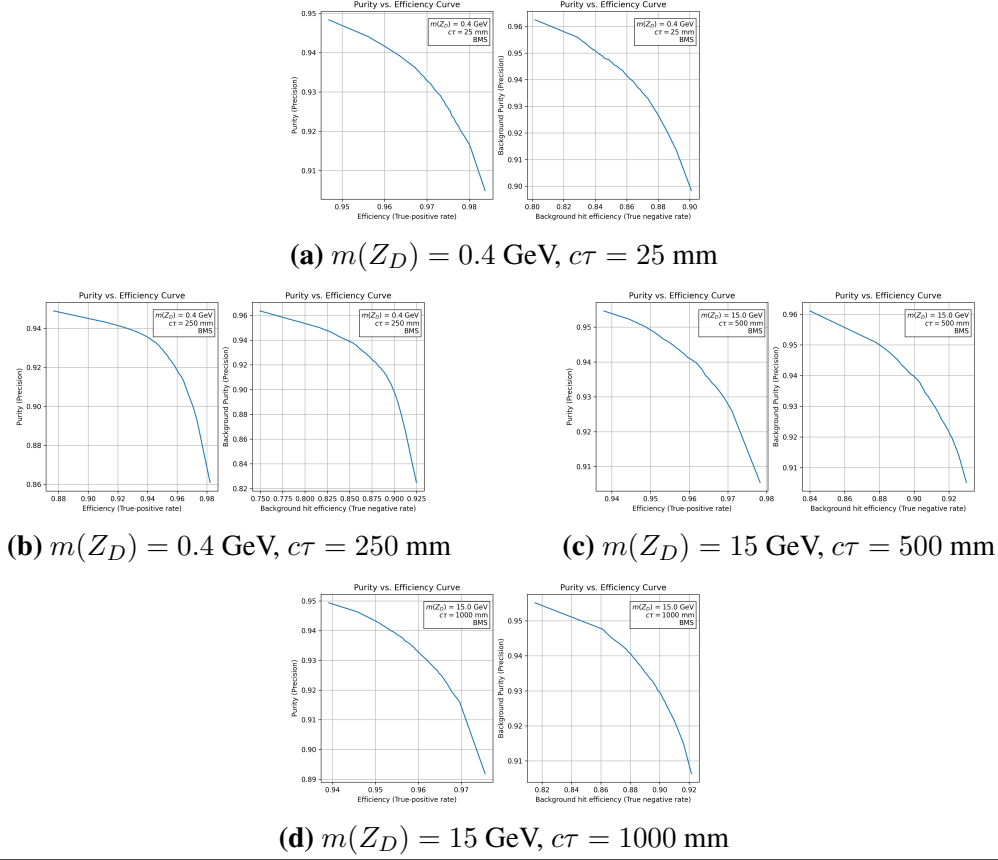


Figure 42 Purity VS efficiency curve for signal and background hit detection ($R_{drift} > 0$) for BMS using *single eta* image input, when the model is trained to identify all muons.

The ROC and purity VS efficiency curves show the expected trends from the results demonstrated so far for the BMS with the Area Under Curve (AUC) covering over 96% of the available plot space at hit-level ($R_{drift} > 0$), indicating strong signal detection and the purity staying at levels above 86% even when applying thresholds for maximal signal detection efficiency. Thus, an overall good muon detection is observed with most of the hard misclassified hits neighbouring actual labelled muon hits as indicated by the distance plots provided in Figure 40.

Some correctly identified examples are provided in Figures 43 - 46 for all samples. A general noticed trend is that the neural network seems to be able to capture the behaviour of a muon and provide generally accurate predictions. Some images with misclassified hits are also provided as examples in Figures 47 - 50.

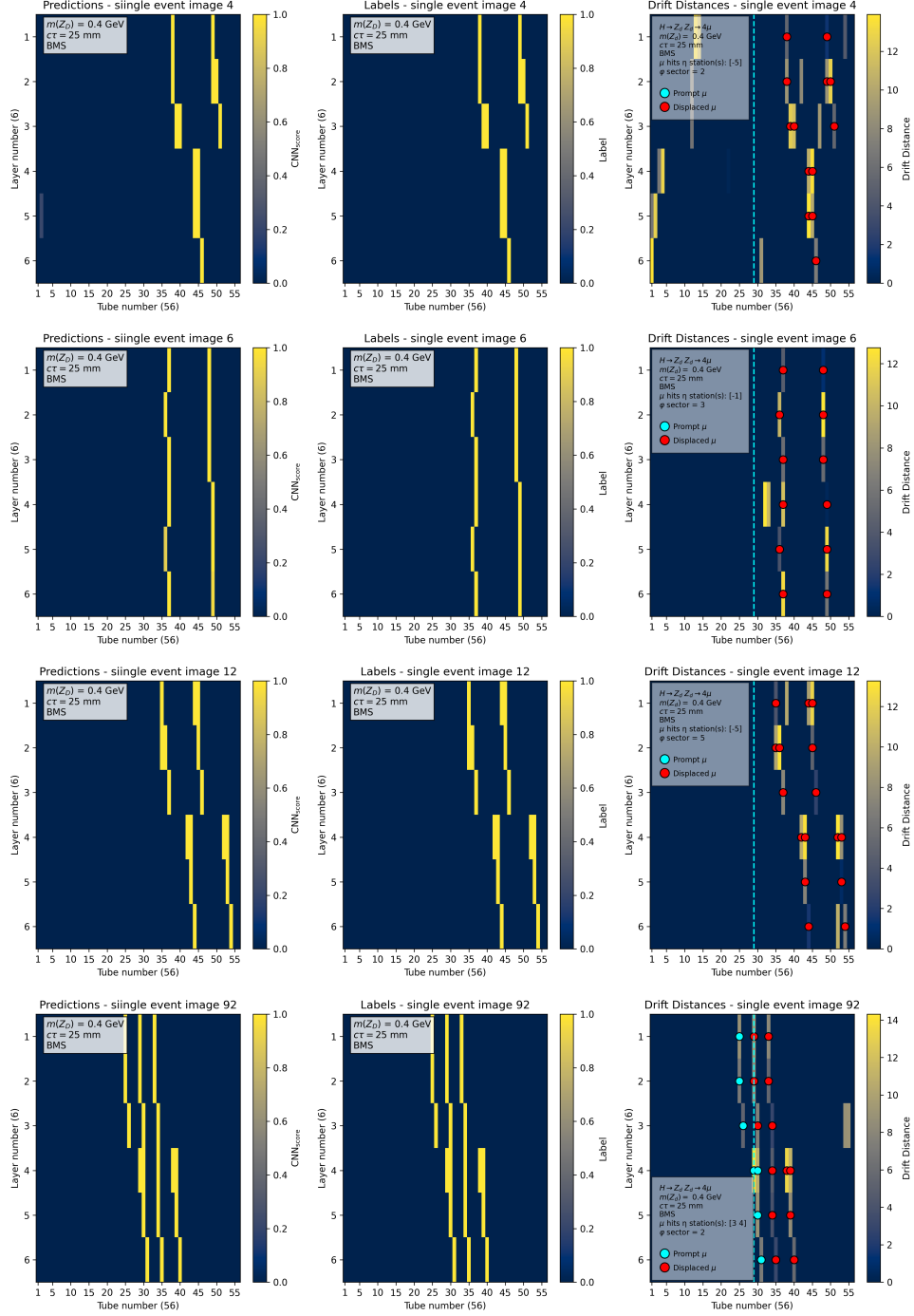


Figure 43 Correctly identified examples of event images from the dark photon sample with $m(Z_D) = 0.4$ GeV, $ct = 25$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

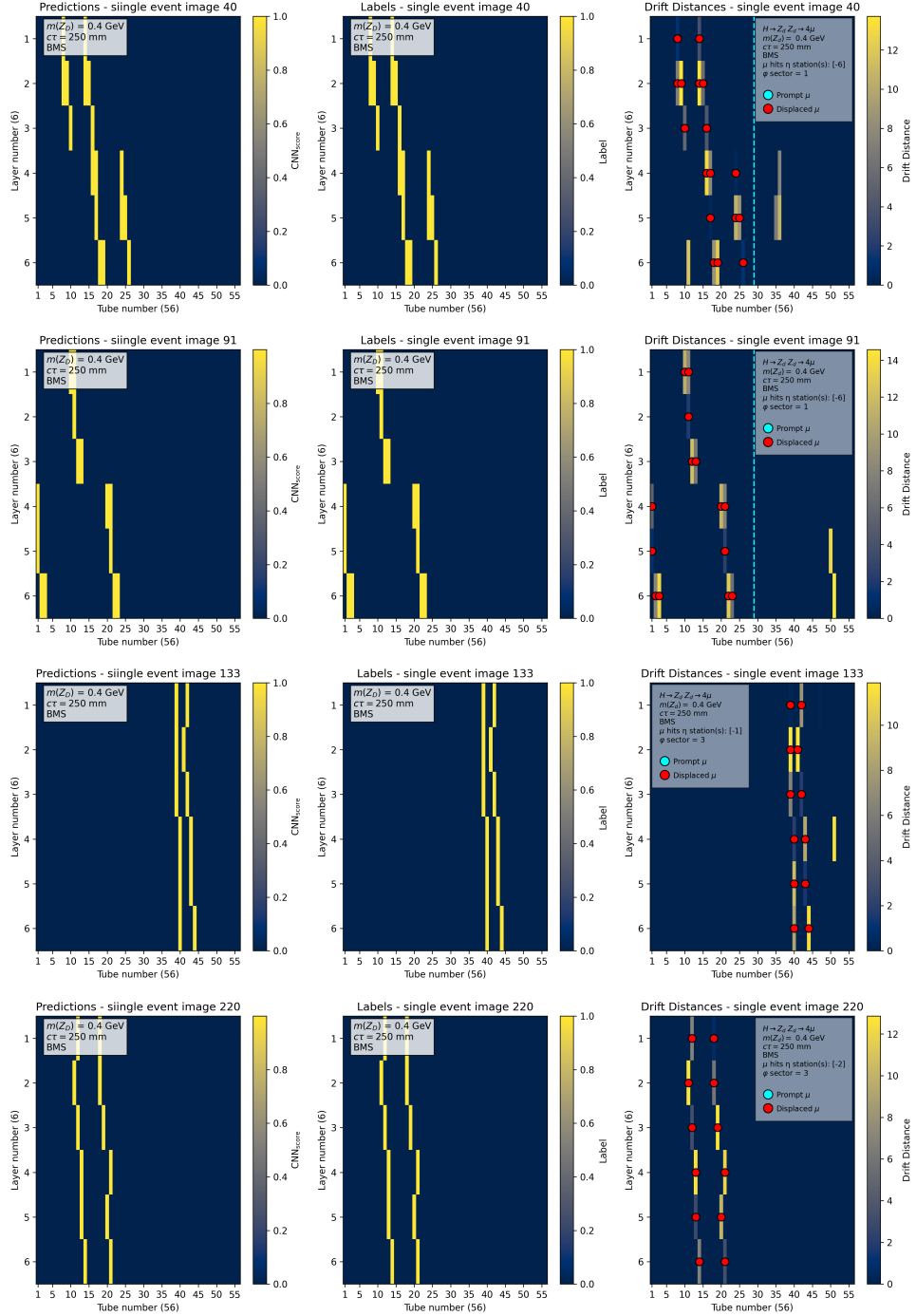


Figure 44 Correctly identified examples of event images from the dark photon sample with $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$ for BMS using *single eta* image input, when the model is trained to identify all muons.

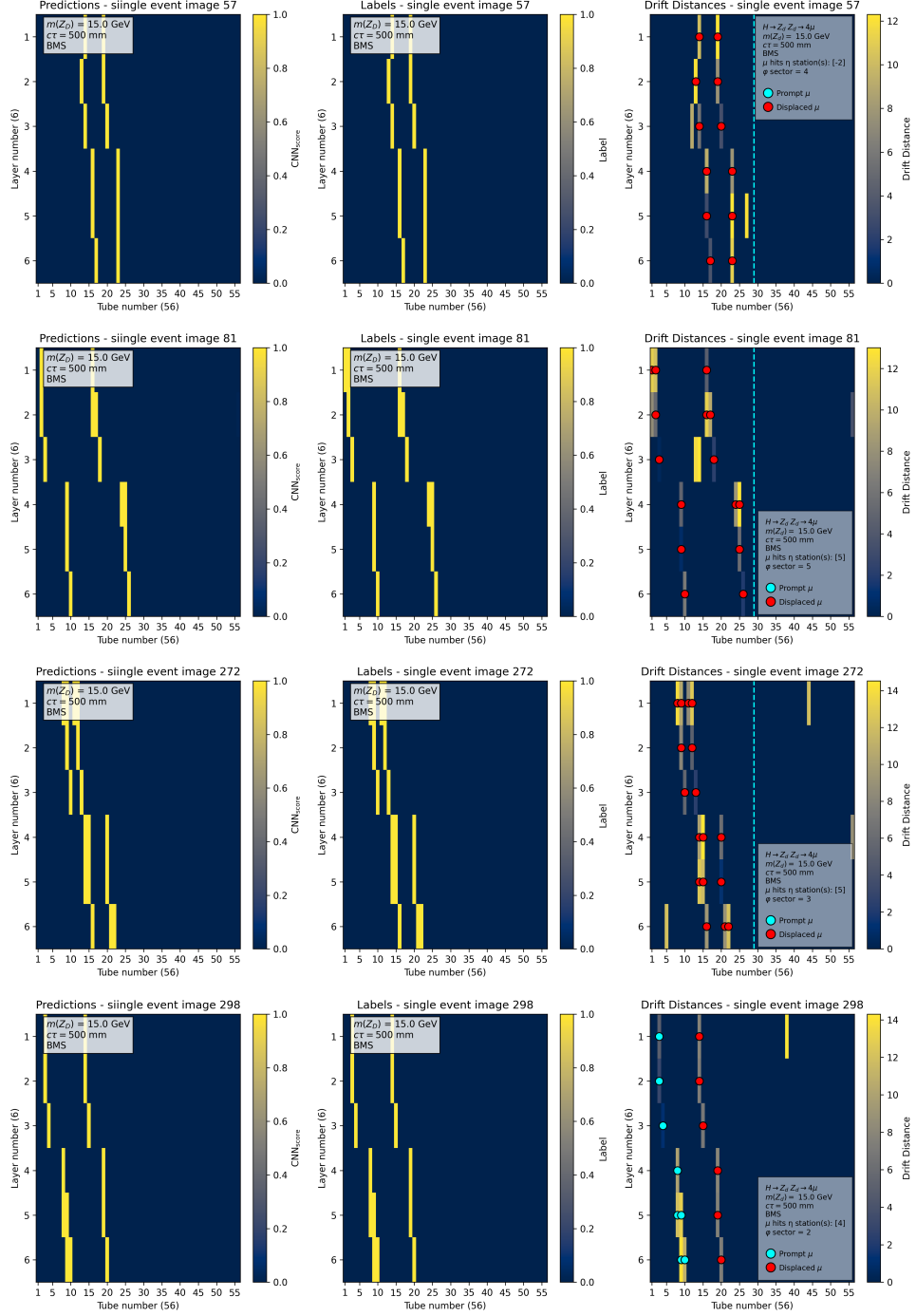


Figure 45 Correctly identified examples of event images from the dark photon sample with $m(Z_D) = 15$ GeV, $c\tau = 500$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

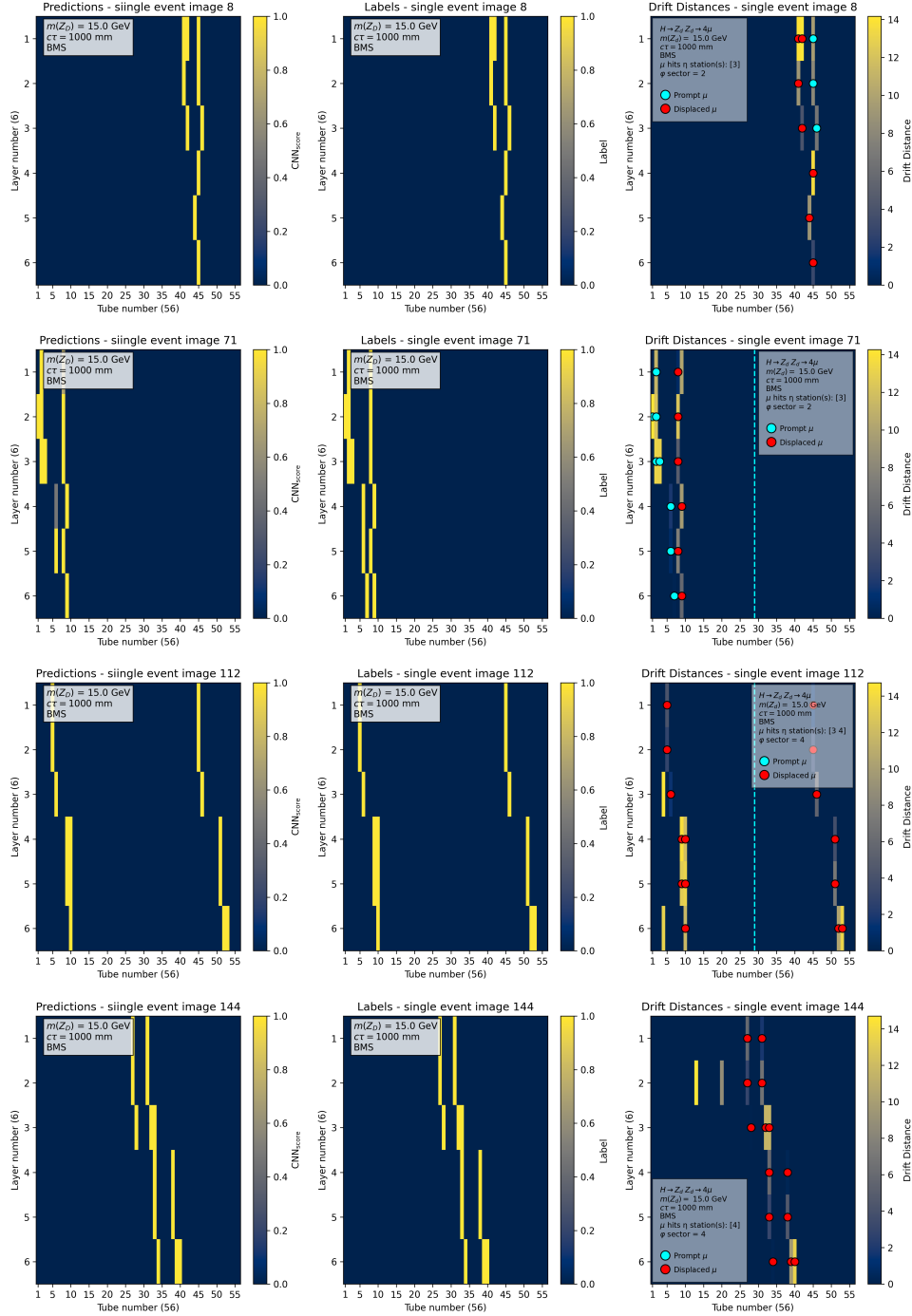


Figure 46 Correctly identified examples of event images from the dark photon sample with $m(Z_D) = 15$ GeV, $\tau = 1000$ mm for BMS using *single eta* input, when the model is trained to identify all muons.

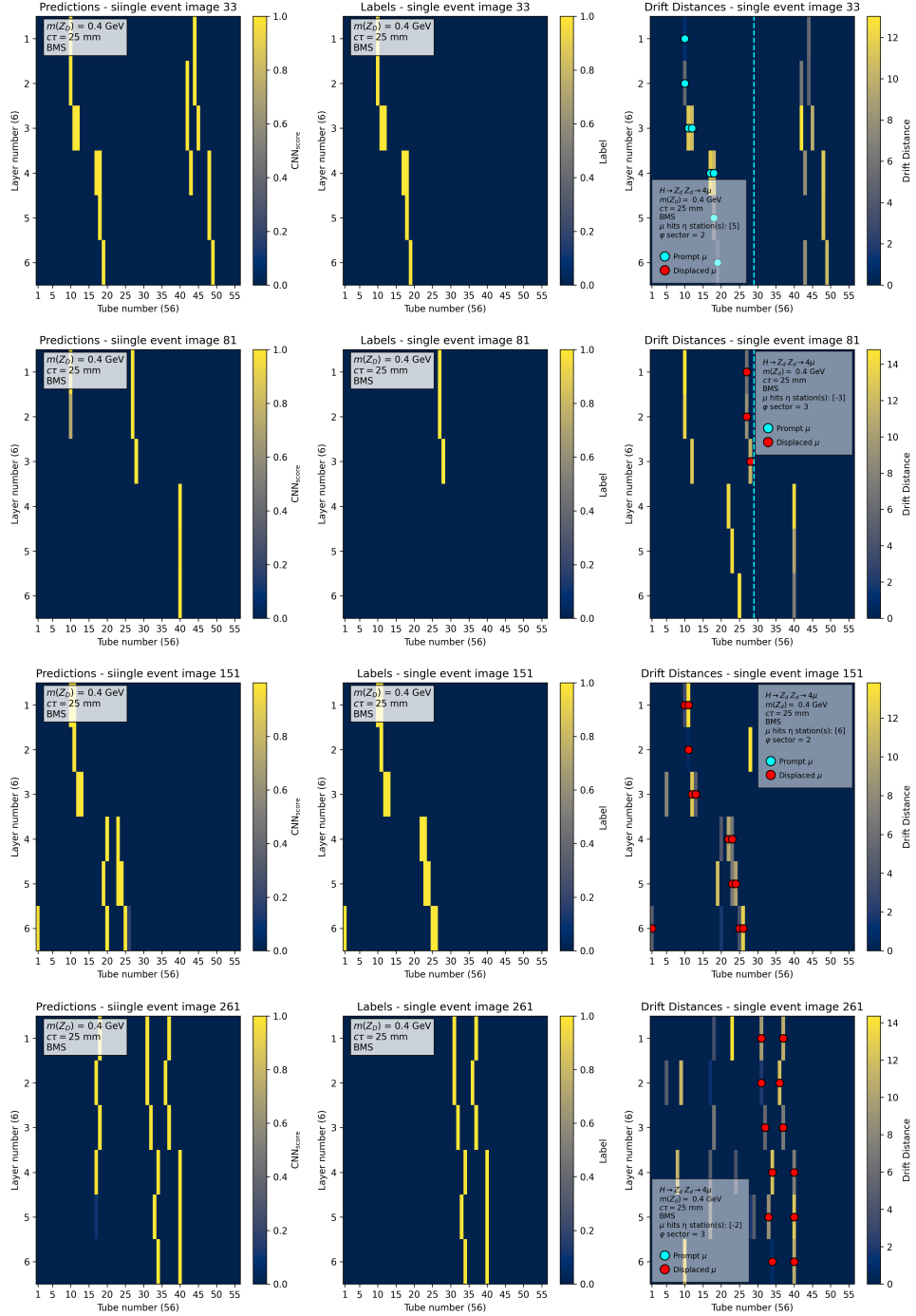


Figure 47 Misidentified examples of event images from the dark photon sample with $m(Z_D) = 0.4$ GeV, $\tau = 25$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

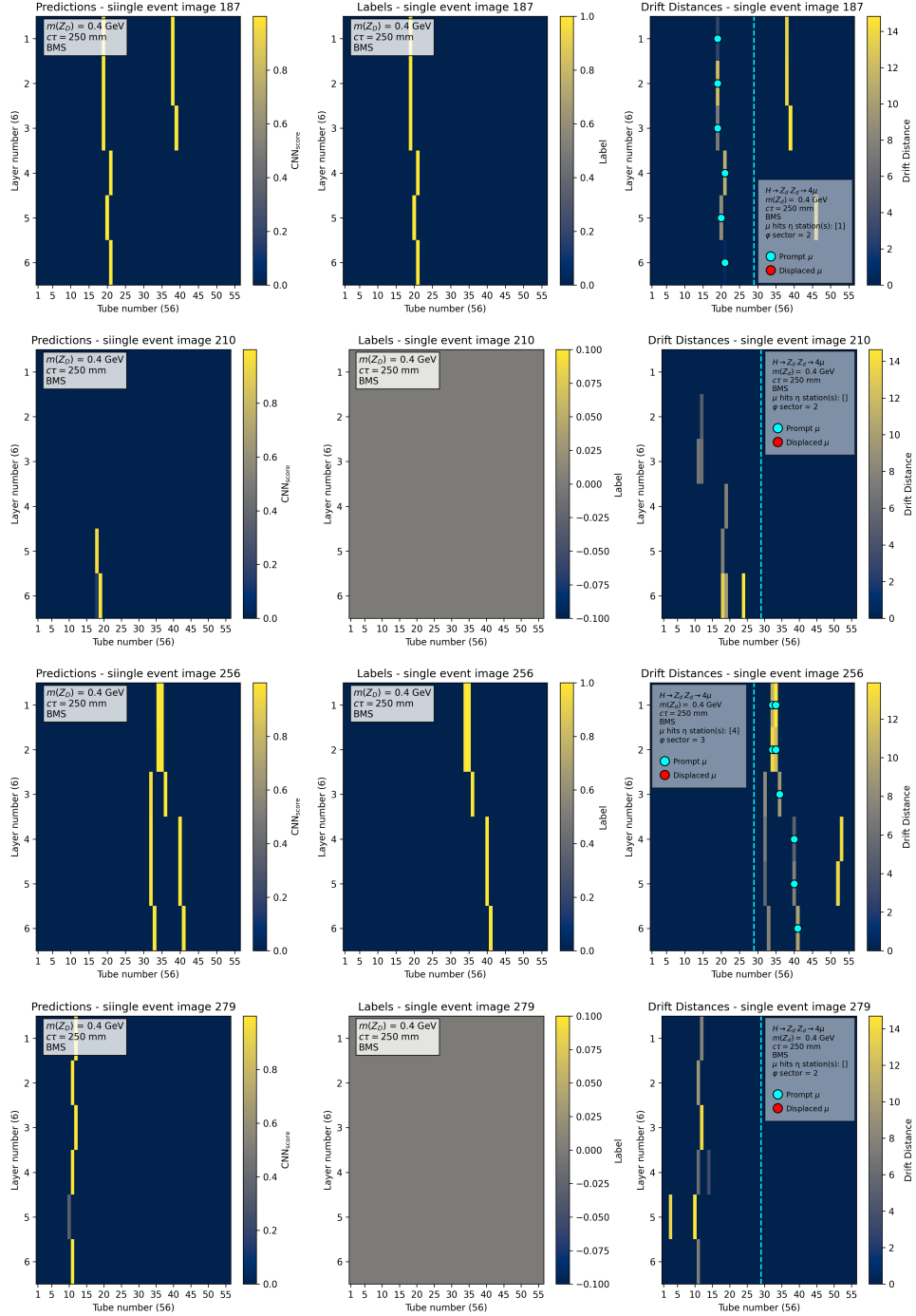


Figure 48 Misidentified examples of event images from the dark photon sample with $m(Z_D) = 0.4$ GeV, $c\tau = 250$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

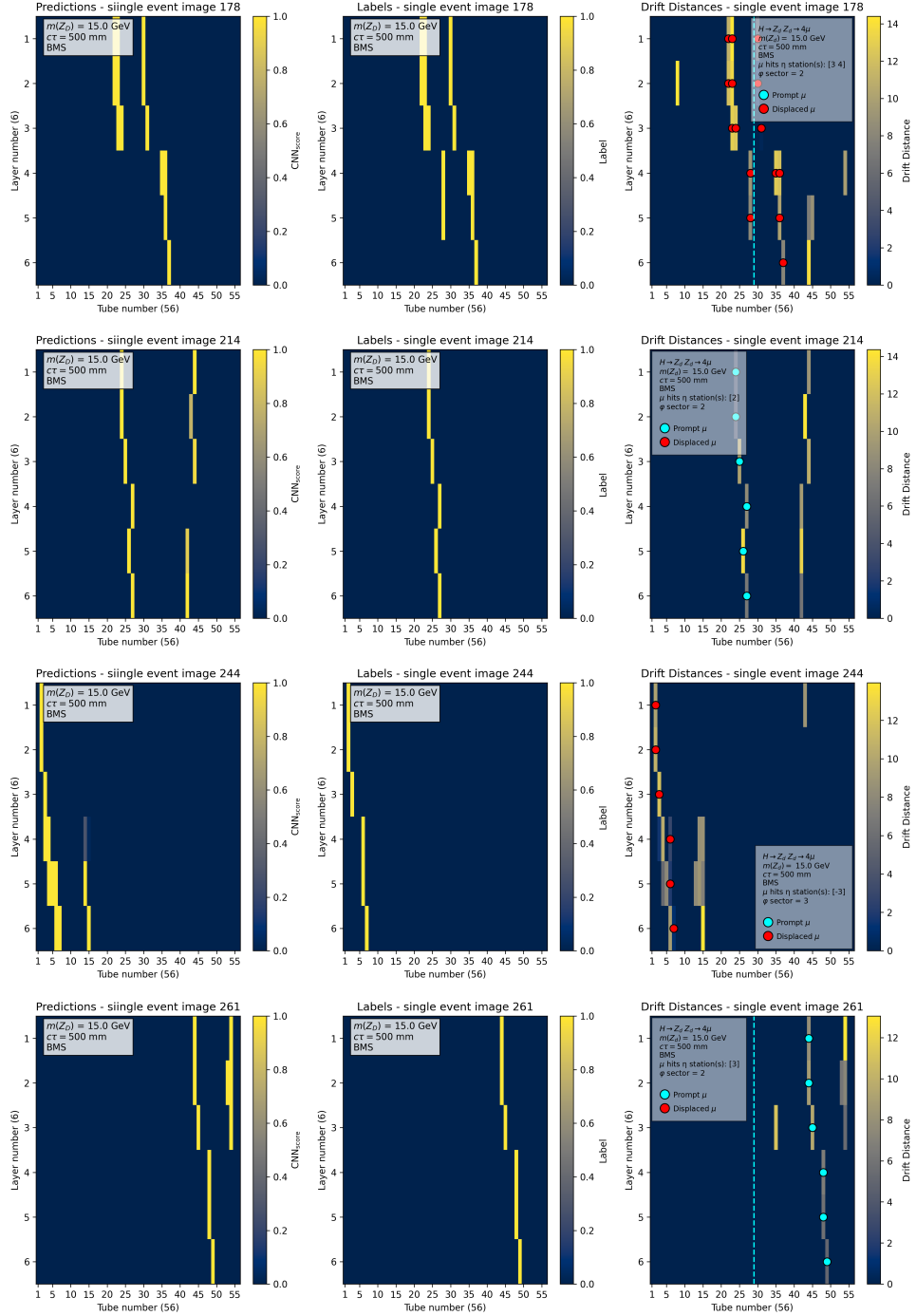


Figure 49 Misidentified examples of event images from the dark photon sample with $m(Z_D) = 15$ GeV, $c\tau = 500$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

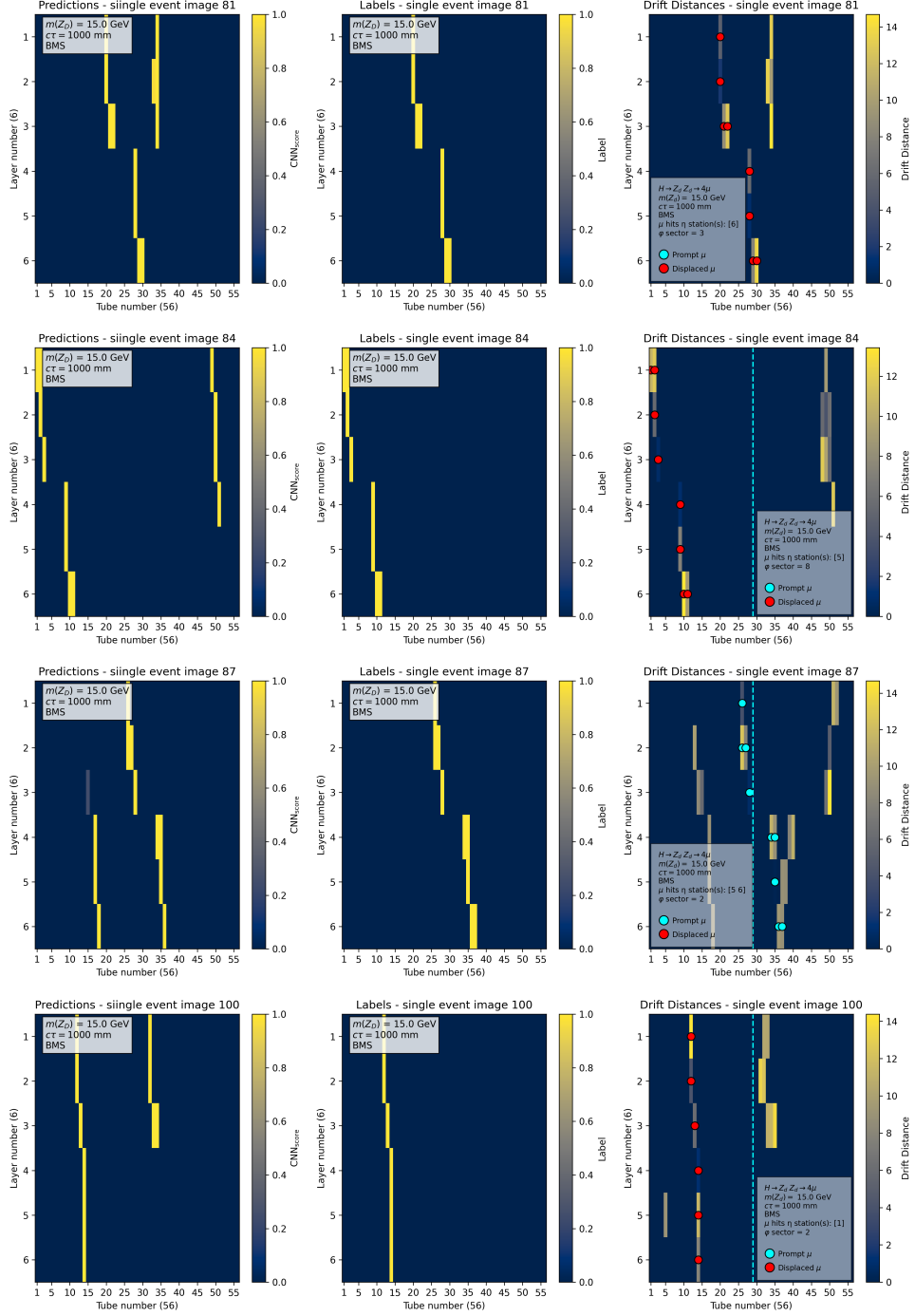


Figure 50 Misidentified examples of event images from the dark photon sample with $m(Z_D) = 15$ GeV, $ct = 1000$ mm for BMS using *single eta* image input, when the model is trained to identify all muons.

After closer observation of the misidentified examples, the neural network seems to follow the already expected behaviour of either misidentifying hits that are close to muon hits or misinterpret background particles that manage to form a track. It is not yet clear whether or not it can make distinction between displaced and prompt muons due to the fact it was not trained to distinguish between the two. Something to note is that displaced muons do show a generally collimated behaviour appearing in pairs, especially for the lower-displaced sample, however, for samples with larger lifetimes, less pairs tend to appear in *single eta* images, which leads to a strong indication that the neural network should fail upon trying to distinguish the displaced muons. This is covered by the next section. Another notable observation is that even when the CNN fails to classify correctly some image entries, the labelled muon hits themselves are usually captured, meaning that if used in the L0 triggering system, events that do have displaced muon information could still pass the triggering selection and be analysed at the HLT. Of course, for such an application the latency goal should be reached.

More results regarding the performance of the neural network across other barrel stations are provided in Tables 4 - 6, in which a model was also trained and tested to identify displaced muons without the addition of prompt muon data from a muon gun sample.

A general observation is that the CNN achieves good performance towards muon detection with the better performance encountered at the BIS, probably due to the fact that BI MDT chambers contain 8 total layers of tubes, meaning there is more space for the convolutions to perform a pattern recognition process. Also an important factor is probably the fact that images tend to be more populated with muons due to less separation as a result of being closer to the IP and perhaps also closer to the DV of the dark photons producing the displaced muons. It should be noted though that BI images are also larger in tube dimension, which affects data sparsity and may make their computation more heavy, thus mitigating `hls4ml` performance, when converting.

Signal sample	Added data	Input data size	True positive (%)	True negative (%)	True negative $R_{drift} > 0$ (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Background	38464	95.8	99.916	90
	Background + muon gun	80145	95.57	99.911	90
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Background	40588	95.2	99.92	90.8
	Background + muon gun	83445	95.5	99.924	91.3
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Background	5740	97.6	99.86	82
	Background + muon gun	47062	95.1	99.89	88
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Background	24312	98	99.86	83
	Background + muon gun	66229	96.9	99.88	87

Table 4 Results from CNN model training and testing targeting all muons for BMS.

Signal sample	Added data	Input data size	True positive (%)	True negative (%)	True negative $R_{drift} > 0$ (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Background	21461	95.6	99.982	95.9
	Background + muon gun	42884	98.6	99.982	96.1
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Background	53373	99.6	99.977	94.8
	Background + muon gun	94867	98.7	99.977	94.9
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Background	5901	98.2	99.972	93.4
	Background + muon gun	48303	98.4	99.977	95
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Background	26420	99.4	99.973	93.8
	Background + muon gun	69504	98.8	99.976	94.9

Table 5 Results from CNN model training and testing targeting all muons for BIS.

Signal sample	Added data	Input data size	True positive (%)	True negative (%)	True negative $R_{drift} > 0$ (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Background	41944	95.5	99.89	86
	Background + muon gun	85486	96	99.924	90.7
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Background	45372	96.2	99.89	86
	Background + muon gun	87615	96.1	99.9	88
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Background	7904	97.1	99.86	82
	Background + muon gun	51065	96.3	99.88	85
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Background	28412	97.1	99.87	84
	Background + muon gun	72236	96.6	99.903	88

Table 6 Results from CNN model training and testing targeting all muons for BOS.

5.1.2 Detecting only displaced muons

In order to check the network's performance on displaced muon hit identification, a threshold on displacement is placed and the data sample is downsampled again accordingly such that $\frac{\text{images with signal}}{\text{all images}} = 0.5$, so that a sufficient amount of images containing prompt muons is also kept for the evaluation.

The CNN was trained to identify only displaced muons, with a displacement of $L_{xy} > 200 \text{ mm}$ and the results were not surprising despite the still worse, but at acceptable levels loss and accuracy curves (Figures 51 and 52), which one should not neglect that do take into consideration easily classified entries, such as the empty ones. As demonstrated in the corresponding $\text{CNN}_{\text{score}}$ output plots (Figures 53 - 55) as well as the confusion matrices (Figures 56 - 59), the neural network basically cannot distinguish between prompt and displaced muons and, as expected, most misclassified hits originate either from prompt muons or displaced muons that do not meet the displacement threshold requirement. A severe performance reduction is also noticed in the corresponding ROC curve (Figure 60) as well as in the purity VS efficiency curves (Figure 61). Overall performance tables comparing performance on also other barrel stations, when targetting displaced muons with a dark photon mother of displacement greater than 200 mm are provided in Tables 7 - 9. One can notice that the results, when it comes to muon distinction are better in BMS, which probably originates from the fact that there is less data sparsity (compared to BI) and less muon separation when compared to BO, which allows the BM *single eta* images to contain more displaced, collimated muon pairs in the input image dimensions.

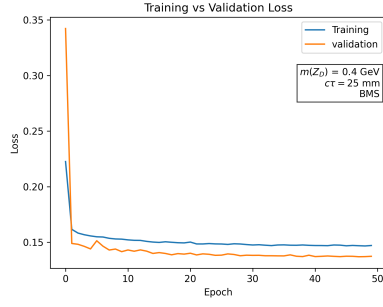
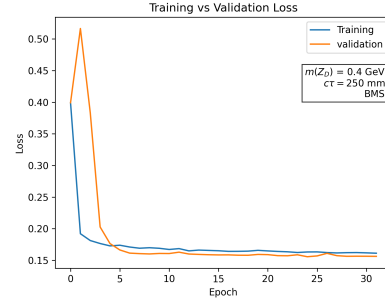
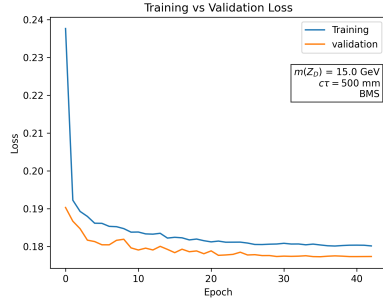
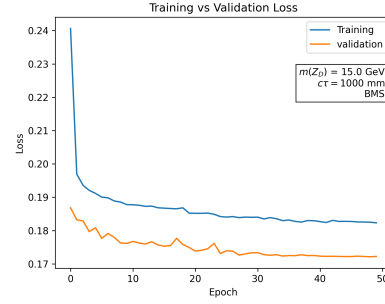
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 51 CNN training and validation loss plots over the number of epochs for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

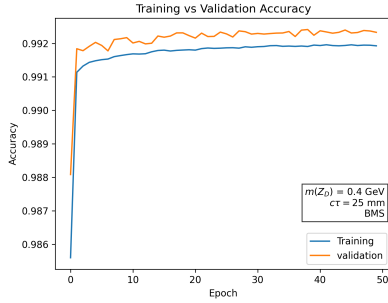
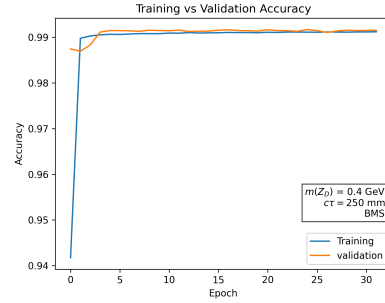
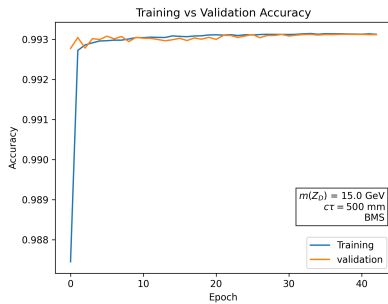
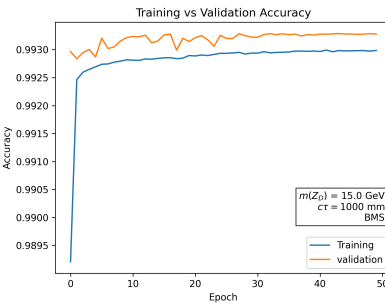
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 52 CNN training and validation accuracy plots over the number of epochs for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

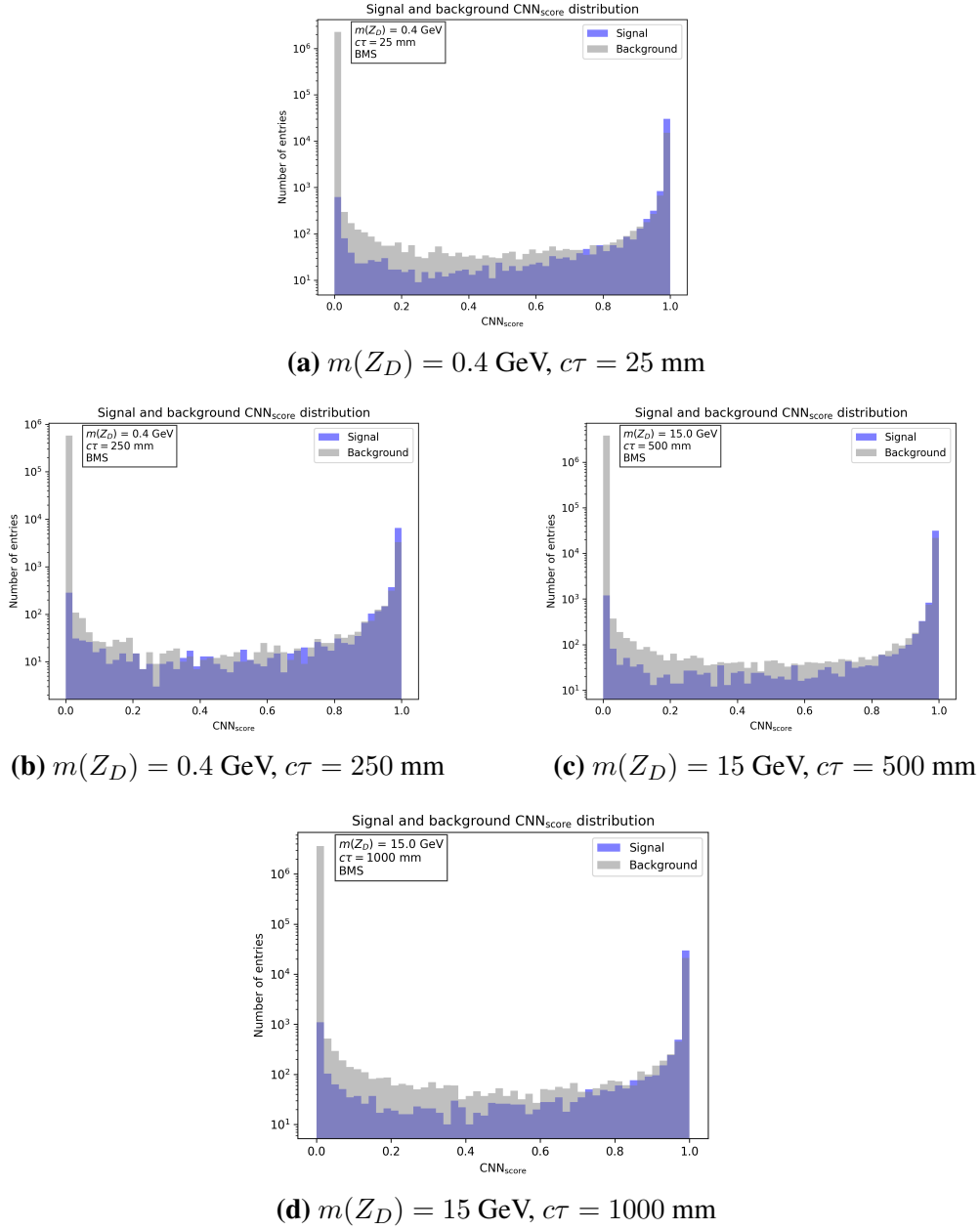


Figure 53 $\text{CNN}_{\text{score}}$ distribution plots for all signal and background entries for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

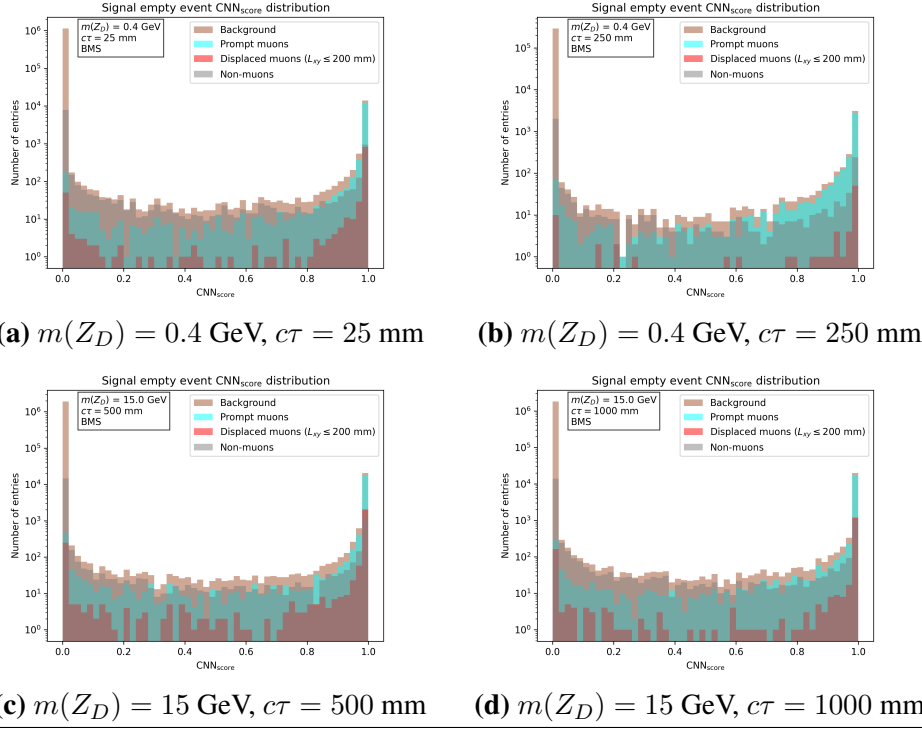


Figure 54 $\text{CNN}_{\text{score}}$ distribution plot for all entries in empty labelled images for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

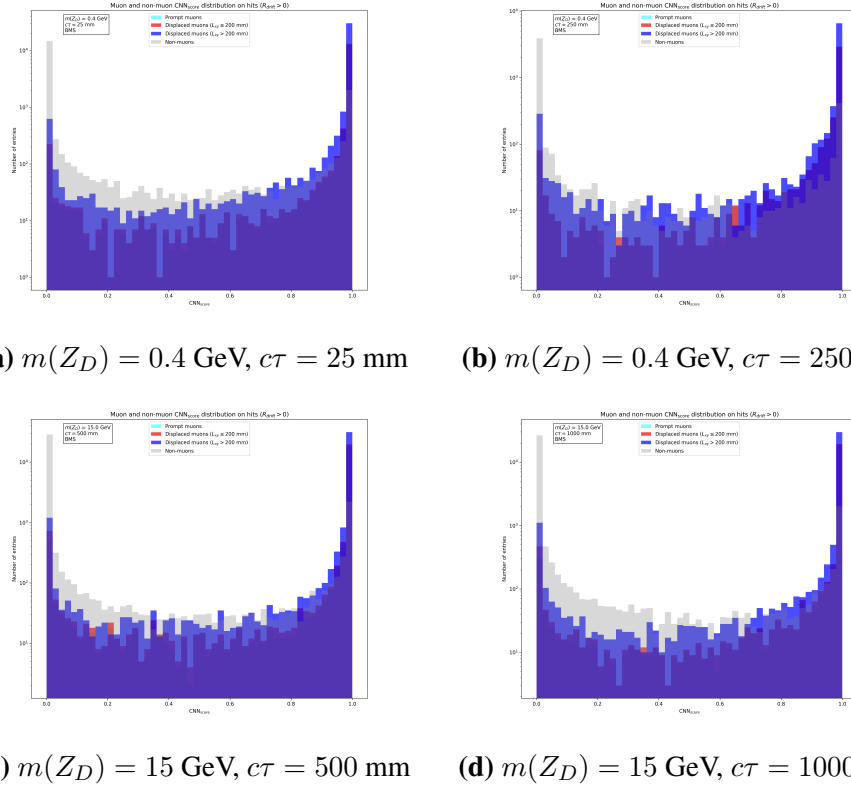


Figure 55 $\text{CNN}_{\text{score}}$ distribution plots for all entries that correspond to hits (non-zero R_{drift} value) for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

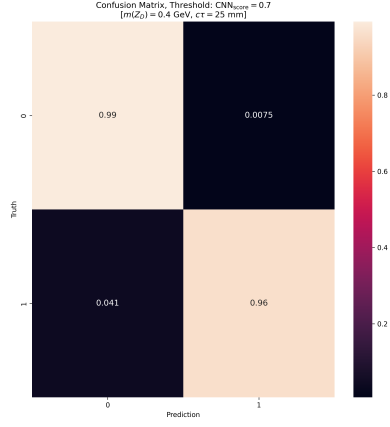
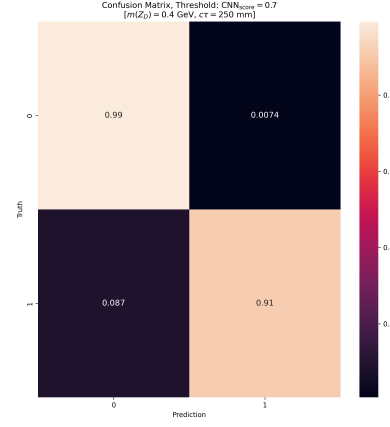
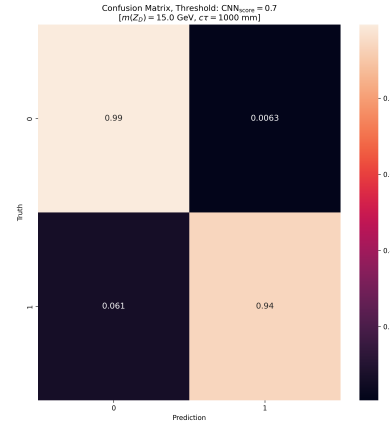
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 56 CNN confusion matrix for signal (only displaced muon hits with $L_{xy} > 200 \text{ mm}$) and background (prompt and displaced below displacement threshold muons, non-muon hits and empty entries) classes for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

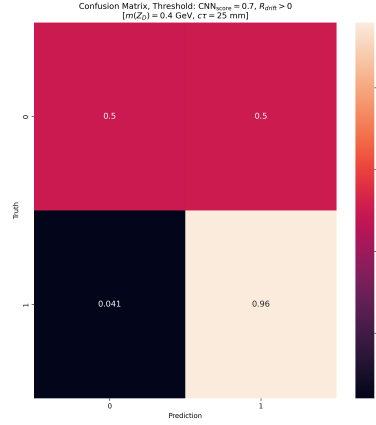
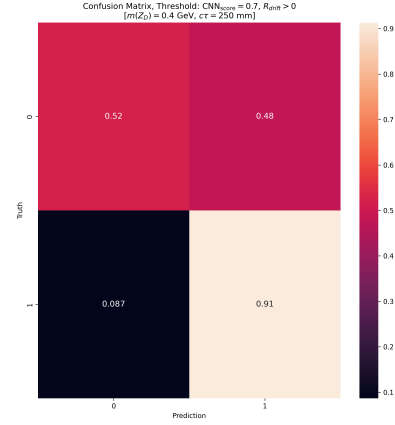
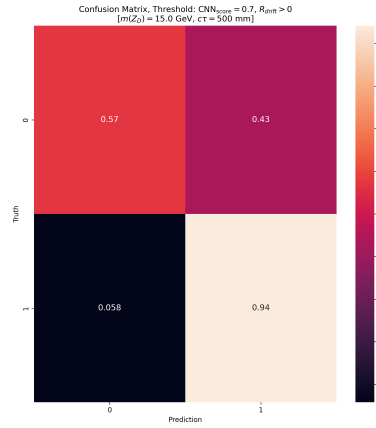
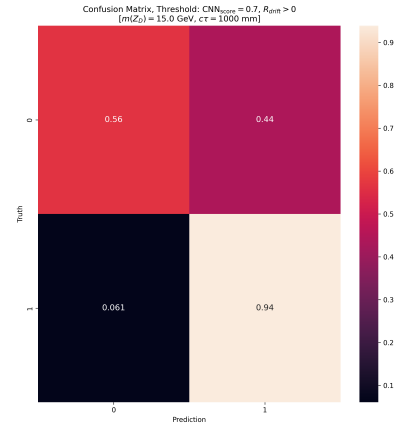
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 57 CNN confusion matrix for signal and background classes, when only counting hits with $R_{\text{drift}} > 0$ for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

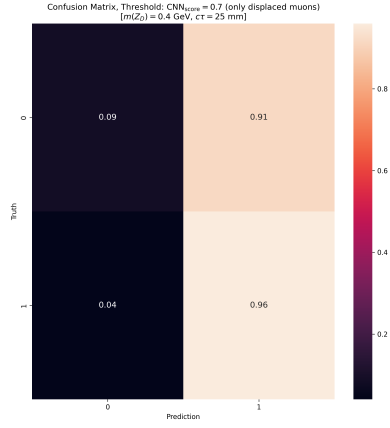
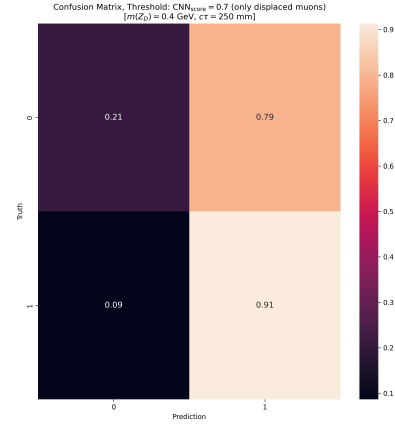
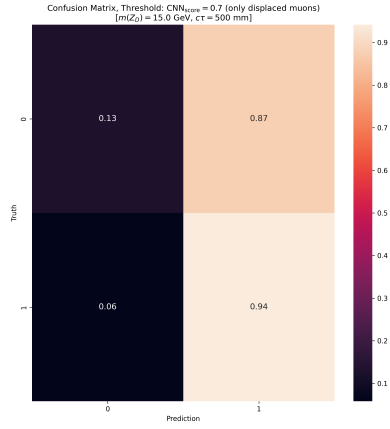
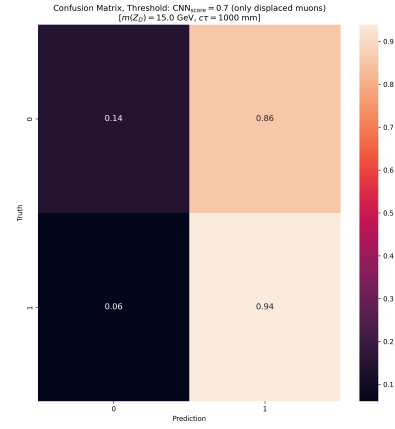

 (a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$

 (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$

 (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$

 (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 58 CNN confusion matrix only for displaced muons (displaced muons under displacement threshold are labelled as 0) for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

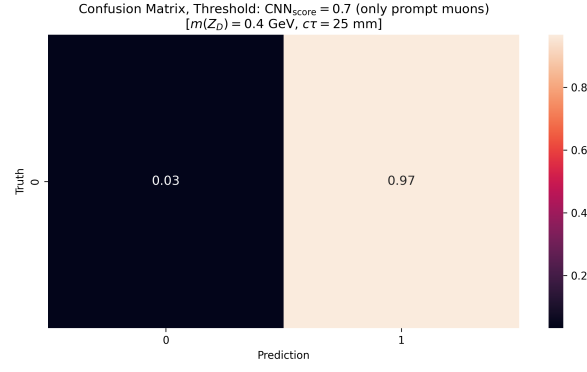
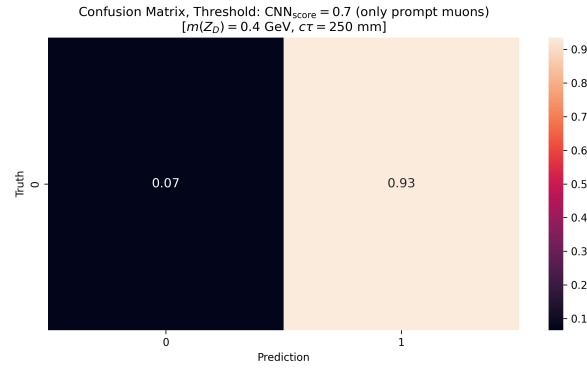
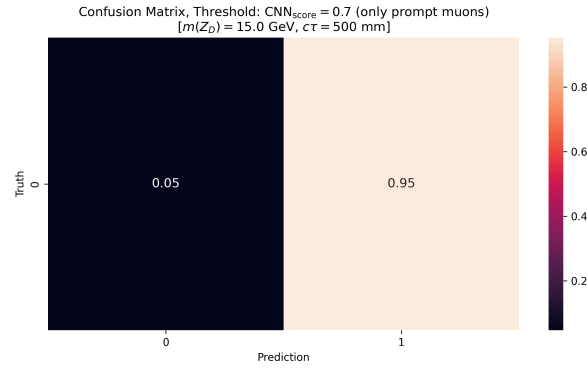
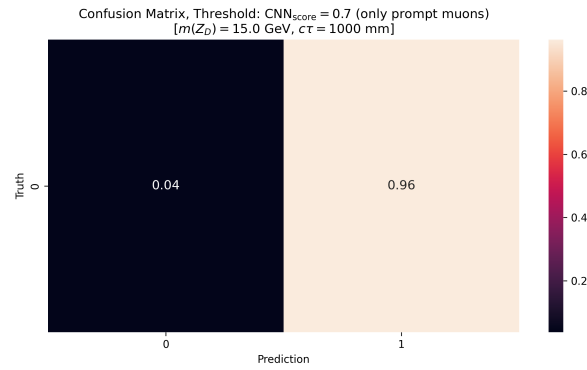
(a) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}, c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}, c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}, c\tau = 1000 \text{ mm}$

Figure 59 CNN confusion matrix only for prompt muons for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

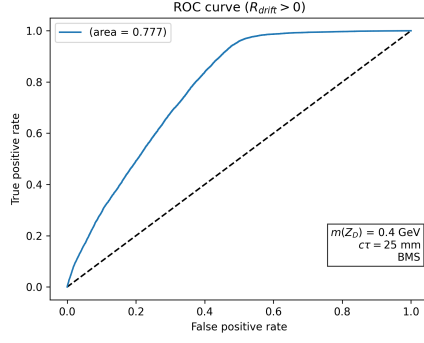
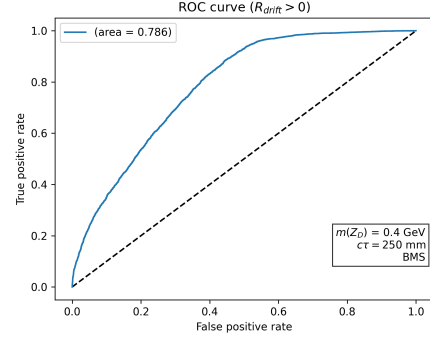
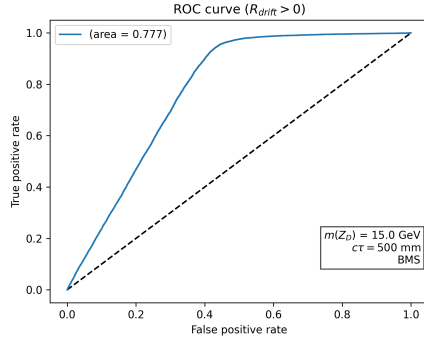
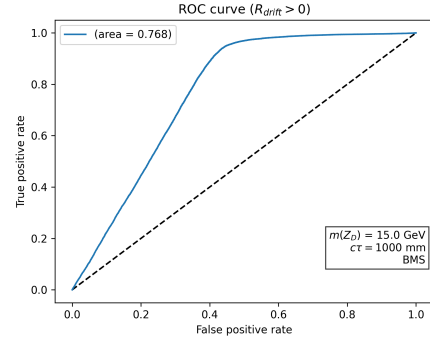
(a) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 1000 \text{ mm}$

Figure 60 ROC curve for hits with $R_{drift} > 0$ for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

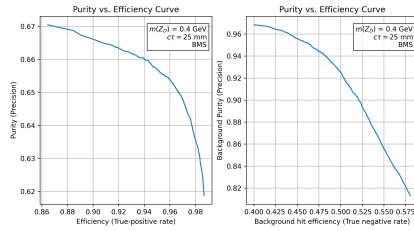
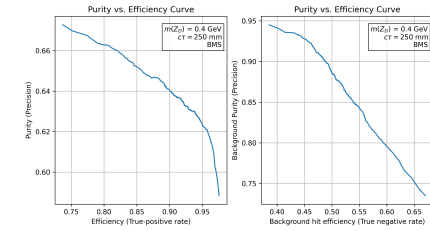
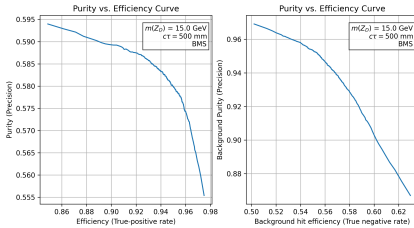
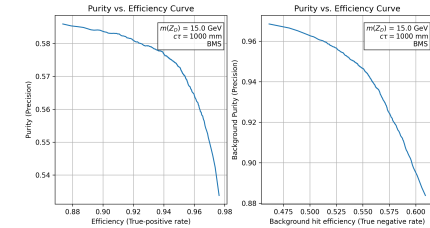
(a) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$ (b) $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$ (c) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 500 \text{ mm}$ (d) $m(Z_D) = 15 \text{ GeV}$, $c\tau = 1000 \text{ mm}$

Figure 61 Purity VS efficiency curve for signal and background hit detection ($R_{drift} > 0$) for BMS using *single eta* image input, when the model is trained to identify only displaced muons with dark photon displacement $L_{xy} > 200 \text{ mm}$.

Signal sample	Input data size	TP (%)	TN (%)	TN (%) $R_{drift} > 0$	TN (%) displaced μ $L_{xy} \leq 200$ mm	TN (%) prompt μ
$m(Z_D) = 15$ GeV $c\tau = 1000$ mm	54520	93.9	99.37	56	14	4
$m(Z_D) = 15$ GeV $c\tau = 500$ mm	57666	94.2	99.38	57	13	5
$m(Z_D) = 0.4$ GeV $c\tau = 250$ mm	8776	91.3	99.26	52	21	7
$m(Z_D) = 0.4$ GeV $c\tau = 25$ mm	34516	95.9	99.25	50	9	3

Table 7 Results for CNN model training and testing for displaced muons with a mother dark photon of displacement $L_{xy} > 200$ mm for BMS

Signal sample	Input data size	TP (%)	TN (%)	TN (%) $R_{drift} > 0$	TN (%) displaced μ $L_{xy} \leq 200$ mm	TN (%) prompt μ
$m(Z_D) = 15$ GeV $c\tau = 1000$ mm	29222	97.74	99.57	52	2	4
$m(Z_D) = 15$ GeV $c\tau = 500$ mm	69774	98.2	99.57	50	2	3
$m(Z_D) = 0.4$ GeV $c\tau = 250$ mm	7534	96.8	99.49	49	7	6
$m(Z_D) = 0.4$ GeV $c\tau = 25$ mm	35442	97.7	99.52	49	4	5

Table 8 Results for CNN model training and testing for displaced muons with a mother dark photon of displacement $L_{xy} > 200$ mm for BIS

Signal sample	Input data size	TP (%)	TN (%)	TN (%) $R_{drift} > 0$	TN (%) displaced μ $L_{xy} \leq 200$ mm	TN (%) prompt μ
$m(Z_D) = 15$ GeV $c\tau = 1000$ mm	59532	94.3	99.49	59	6	4
$m(Z_D) = 15$ GeV $c\tau = 500$ mm	61160	95.51	99.47	57	6	5
$m(Z_D) = 0.4$ GeV $c\tau = 250$ mm	11718	95.47	99.39	54	0	4
$m(Z_D) = 0.4$ GeV $c\tau = 25$ mm	40118	96.66	99.41	54	3	3

Table 9 Results for CNN model training and testing for displaced muons with a mother dark photon of displacement $L_{xy} > 200$ mm for BOS

5.2 RNN results

5.2.1 Detecting all muons

The results for the RNN are going to be presented in greater brevity now that the reader is more familiar with the plots used as performance metrics. With that said, the overall performance of the RNN on general muon detection is provided for all samples in Table 10. As a reminder, the RNN image input differs from that of the CNN as it can sustain information from all 20 tube layers encountered in the MDT chambers across BI, BM and BO.

Then the performance plots of the best performing model will be displayed and discussed.

Signal sample	Added data	Input data size	True positive (%)	True negative (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Background	9095	86.98	97.89
		18190	89.25	97.95
	Background + muon gun	20185	84.06	96.59
		42800	94.34	98.03
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Background	9780	88.46	97.87
		19780	92.75	97.97
	Background + muon gun	20840	93.51	98.14
		23450	91.95	98.08
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Background	1760	69.82	97.76
		3550	83.33	97.20
	Background + muon gun	13075	91.36	98.21
		28640	93.06	98.25
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Background	5045	80.39	97.65
		19355	91.09	97.30
	Background + muon gun	25265	90.11	98.02
		43880	89.51	98.41

Table 10 Results from RNN model training and testing targeting all muons in general.

From the results demonstrated in Table 10, all models achieve over 96% correct recognition of 0 classified entries, which leads to a strong resistance against triggering on images empty of signal hits. When it comes to signal hit detection, it is notable that the models trained on the statistically poorer samples are the ones that are less performant, such as the dark photon samples of $m(Z_D) = 0.4 \text{ GeV}$ and $c\tau = 250 \text{ mm}$ and 25 mm , which achieve a TPR of 69.82% and 80.39% respectively for their smaller sample sizes. Another important observation is that the models that were trained with the addition of prompt muons from muon gun samples tend to achieve either an almost identical performance ($< 1 - 2\%$ difference), which could be explained by statistical fluctuations, or a better one, but with a greater margin of difference at TPR when compared to those that were trained to recognise muons without the addition of prompt muons, such as the samples with $c\tau = 25 \text{ mm}$ and 1000 mm in which a difference in TPR of $5 - 8\%$ is observed. This is attributed to the greater muon abundance the RNN has in its disposal, when prompt muons from a muon gun enter the input data. The most performant model when it comes to muon recognition efficiency and background rejection is the one that corresponds to the dark photon sample of $m(Z_D) = 15 \text{ GeV}$ and $c\tau = 1000 \text{ mm}$, where a TPR of 94.34% and

TNR of 98.03 is achieved.

The performance plots of this model are provided in Figures 62 - 67.

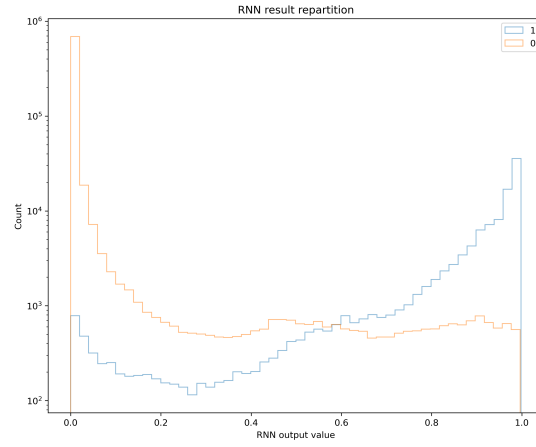
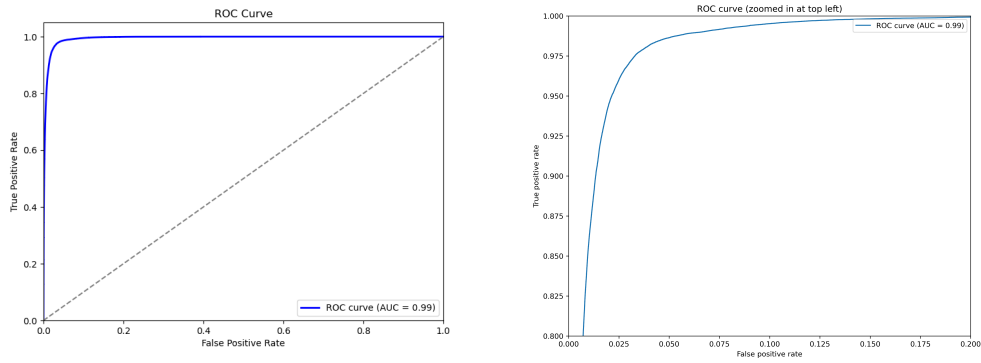


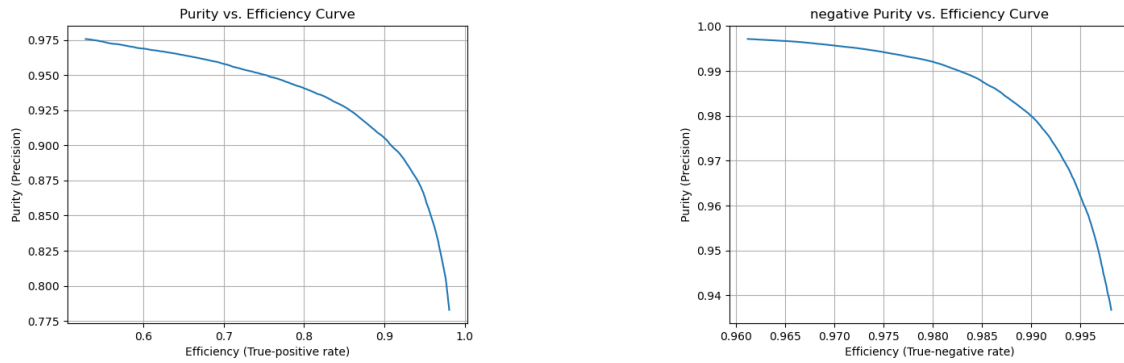
Figure 62 Distribution histogram of the results from testing the RNN model trained on input data from the $m(Z_D) = 15$ GeV and $c\tau = 1000$ mm sample.



(a) ROC curve trend general overview.

(b) ROC curve trend zoomed in

Figure 63 ROC curve of the RNN model trained on input data from the $m(Z_D) = 15$ GeV and $c\tau = 1000$ mm sample.



(a) Purity VS efficiency curve for signal data.

(b) Purity VS efficiency curve for 0 labelled entries.

Figure 64 Purity VS efficiency curves of signal and non-signal data after testing the RNN model trained on input data from the $m(Z_D) = 15$ GeV and $c\tau = 1000$ mm sample.

Upon inspection of the RNN performance plots, an overall optimal performance is demonstrated upon signal detection efficiency, as the RNN classifies most muon hits correctly with a confidence greater than 50%. An overall better classification is achieved on non-signal entry classification and generally, the performance listed in Table 10 seems to be validated from the network output distribution plot seen in Figure 62 as well as the ROC curves depicted in Figure 67.

Upon closer inspection a relatively high purity is achieved, close to 80% for high efficiency values. As for the non-muon class an even better performance is achieved.

5.2.2 Detecting only displaced muons

A similar test as the corresponding one for the CNN was also performed for the RNN model case and the drop in performance is not surprising, as the RNN struggles to differentiate between displaced and prompt muons. A displacement threshold of 200 mm was applied, similarly to the CNN case, in order to label the displaced muon data with high enough displacement so that a distinction against prompt muons can be observed. Results of the RNN models on the available samples are demonstrated in Table 11.

Signal sample	Added data	Input data size	True positive (%)	True negative (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Background	1743	83.02	96.90
		3294	88.24	97.65
	Background + Muon gun	3966	10.17	99.72
		8414	28.79	99.43
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Background	1934	84.23	96.87
		3424	91.65	97.12
	Background + Muon gun	4147	22.19	99.27
		8737	11.15	99.66
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Background	280	73.82	97.47
		547	78.03	96.95
	Background + Muon gun	2978	40.42	99.82
		5586	47.71	99.81
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25$	Background	935	81.58	96.12
		1810	88.20	96.48
	Background + Muon gun	3505	62.07	99.36
		6888	69.77	99.38

Table 11 Results from RNN model training and testing targeting only displaced muons where the dark photon mother shows a displacement higher than 200 mm.

Inspecting the results for the RNN models trying to detect displaced muons shows severe performance reduction and high fluctuations when it comes to detection efficiency of the displaced muons. Similarly to the CNN case, the models cannot confidently distinguish between prompt and displaced muons.

The output of the RNN shows this randomness when it comes to displaced muon classification, as it cannot confidently classify the provided hit data. This is shown in Figure 65.

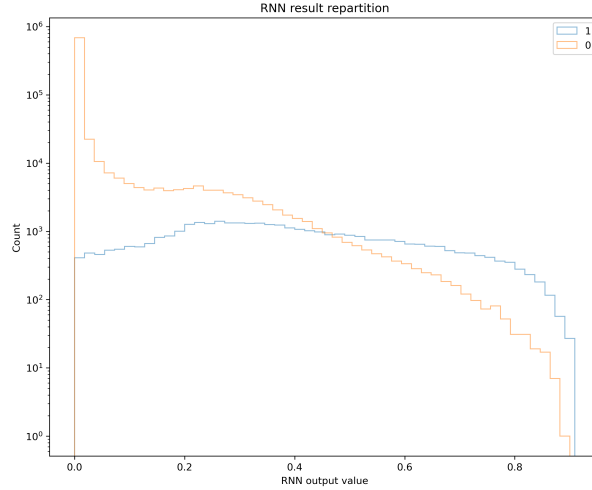


Figure 65 Distribution histogram of selected RNN model when targeting only displaced muons above the displacement threshold

5.3 hls4ml implementation

5.3.1 CNN hls4ml models

As a final step in this analysis both types of neural network models have to be validated on unfiltered data. As a reminder to the reader, it is mentioned that when it comes to muon identification in general, all input data samples consisted of a $\frac{\text{images with signal}}{\text{all images}}$ ratio of 0.75, in order to assist the model to recognise the muon signals. However, an unfiltered test has to be made in order to inspect the model's ability to generalise.

The results from the CNN validation process prior and post hls4ml conversion are provided in Table 12.

By inspecting the performance of the CNN upon the various validation samples one can notice a reduction in performance towards muon detection, when it comes to the heavier dark photon samples, compared to the initial results encountered in Section 5.1, however the most notable observation is with respect to background recognition. A severe inconsistency in background rejection is observed especially in the lightest dark photon mass samples. This is probably caused by the inability of the CNN to generalise on the randomness the background data can show. Since the CNN faces a large sparsity issue, more work is required for the models to perform sufficiently well as well as training them on data with higher statistics, in order to help counter sparsity. A main reason behind this failure could also be the inability to contain information from all three barrels in its images, which greatly limits the available information per event.

As for the hls4ml conversion, severe inconsistencies are observed with much smaller TPRs, but much higher TNRs, which leads to the assumption that this is a conversion issue rather than an architecture related problem. Of course further investigation is required to support this claim. The overall outcome of the validation process is that the CNN may not be the best trigger candidate, unless it is further optimised for sparsity.

Sample Trained Model	Sample	Python TP (%)	Python TN (%)	hls4ml TP (%)	hls4ml TN (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Sample 1	79.39	39.77	6.18	98.06
	Sample 2	79.89	39.77	6.02	98.05
	Sample 3	80.82	39.7	18.98	82.82
	Sample 4	67.73	39.69	21.67	82.83
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Sample 1	79.52	35.32	29.2	85.89
	Sample 2	80.65	35.38	31.24	85.86
	Sample 3	81.46	35.28	31.75	58.97
	Sample 4	72.81	35.33	32.07	58.9
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Sample 1	92.19	20.5	20.1	82.85
	Sample 2	94.58	20.41	18.98	82.81
	Sample 3	93.98	20.41	19.01	82.82
	Sample 4	88.21	20.4	21.67	82.83
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Sample 1	98.31	6.83	34.33	58.94
	Sample 2	99.28	6.82	32.06	58.98
	Sample 3	99.29	6.8	31.74	58.97
	Sample 4	98.11	6.79	32.07	58.9

Table 12 Comparison of performances between Python and hls4ml CNN models for hits ($R_{drift} > 0$) on different samples.

Sample 1 - $m(Z_D) = 15 \text{ GeV}$, $c\tau = 1000 \text{ mm}$

Sample 2 - $m(Z_D) = 15 \text{ GeV}$, $c\tau = 500 \text{ mm}$

Sample 3 - $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$

Sample 4 - $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$

5.3.2 RNN hls4ml models

In the RNN case, compatibility with hls4ml shows more optimistic results, as seen in Table 13. The signal detection efficiency is at a range of 89 – 96% across all samples, while the background rejection is at 90 – 97% prior to conversion.

After conversion, performance dips mostly for the lower lifetime samples for both dark photon masses, a trend probably explained from the quantization process that goes along the lines of hls4ml conversion. The most notable performance drop is again encountered for the $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$ sample, as the TPR drops to approximately 55%.

Inspecting the two models that perform better, the most consistent one towards both signal efficiency and background rejection is the one trained and tested on the sample with generally larger dark photon displacement (and longer lifetime), marking for both goals an approximately > 90% efficiency.

The drop in performance mentioned when it comes to hls4ml conversion is depicted across the comparative performance plots seen in Figures 66 - 69. The output distribution is far more spread across the prediction range, indicating the reduction in confidence towards classifying the passed hits correctly. This is further supported by the dip in the AUC value, when compared to the Python model, as well as the drop in purity even below 50%, when aiming for the highest signal detection efficiency.

Sample Trained Model	Sample	Python TP (%)	Python TN (%)	hls4ml TP (%)	hls4ml TN (%)
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 1000 \text{ mm}$	Sample 1	94.29	96.84	92.36	94.20
	Sample 2	94.46	96.83	92.72	94.15
	Sample 3	94.51	96.85	92.77	94.17
	Sample 4	90.51	96.88	88.05	94.31
$m(Z_D) = 15 \text{ GeV}$ $c\tau = 500 \text{ mm}$	Sample 1	94.06	96.66	75.29	96.55
	Sample 2	94.40	96.68	75.82	96.57
	Sample 3	94.42	96.69	75.89	96.57
	Sample 4	89.36	96.58	71.29	96.45
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 250 \text{ mm}$	Sample 1	96.17	90.35	93.40	87.59
	Sample 2	96.40	90.43	93.75	87.61
	Sample 3	96.44	90.51	93.82	87.62
	Sample 4	94.08	90.37	90.20	87.53
$m(Z_D) = 0.4 \text{ GeV}$ $c\tau = 25 \text{ mm}$	Sample 1	92.05	97.47	54.83	97.39
	Sample 2	92.22	97.48	55.18	97.38
	Sample 3	92.27	97.49	55.21	97.36
	Sample 4	93.20	97.28	54.61	96.93

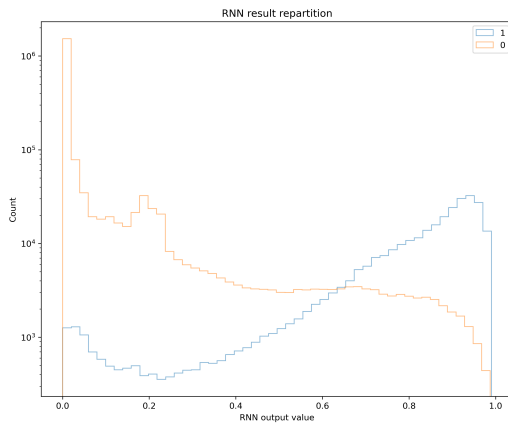
Table 13 Comparison of performances between Python and hls4ml RNN models on different samples.

Sample 1 - $m(Z_D) = 15 \text{ GeV}$, $c\tau = 1000 \text{ mm}$

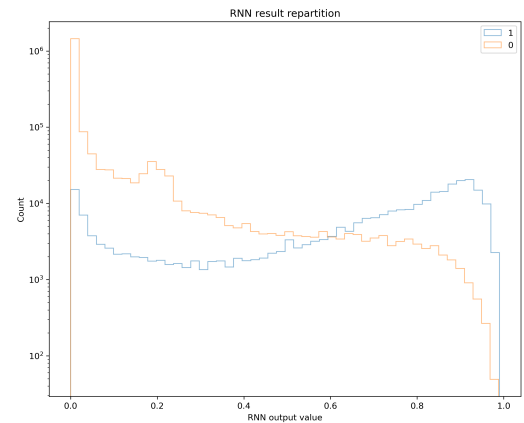
Sample 2 - $m(Z_D) = 15 \text{ GeV}$, $c\tau = 500 \text{ mm}$

Sample 3 - $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 250 \text{ mm}$

Sample 4 - $m(Z_D) = 0.4 \text{ GeV}$, $c\tau = 25 \text{ mm}$

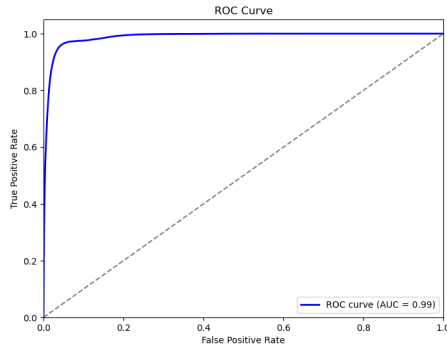


(a) RNN results distribution from validation tests before hls4ml conversion.

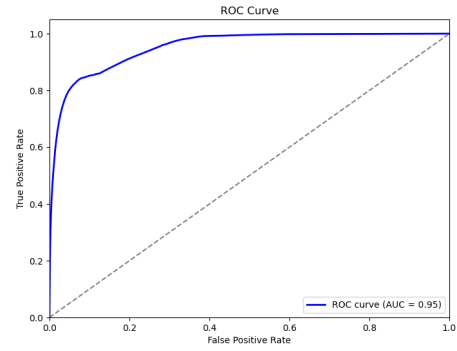


(b) RNN results distribution from validation tests after hls4ml conversion.

Figure 66 RNN results distribution from validation tests before and after hls4ml conversion.

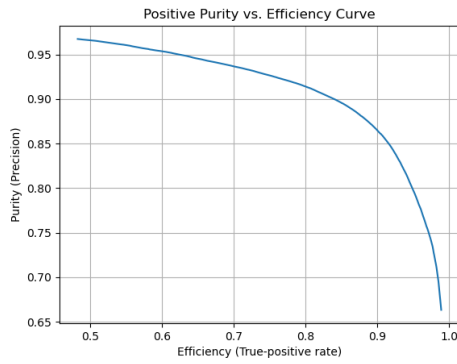


(a) ROC curve from RNN validation test
before hls4ml conversion

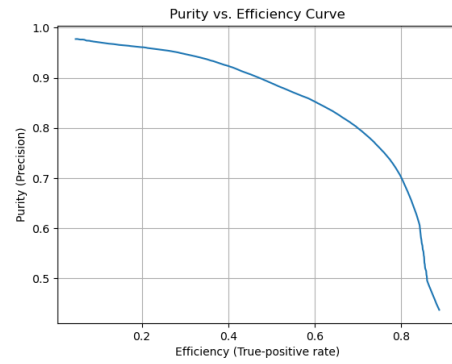


(b) ROC curve from RNN validation test after
hls4ml conversion

Figure 67 ROC curve from RNN validation test before and after hls4ml conversion

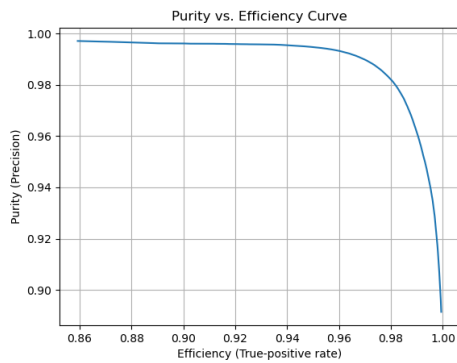


(a) Purity vs efficiency for true positives of
RNN validation tests results before hls4ml
conversion

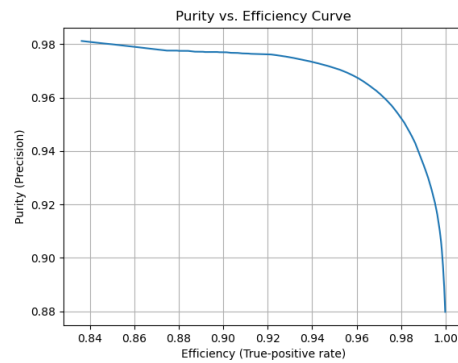


(b) Purity vs efficiency for true positives of
RNN validation tests results after hls4ml
conversion

Figure 68 Purity vs efficiency for true positives of RNN validation tests results before and after hls4ml conversion



(a) Purity vs efficiency for true negatives of
RNN validation tests results before hls4ml
conversion



(b) Purity vs efficiency for true negatives of
RNN validation tests results after hls4ml
conversion

Figure 69 Purity vs efficiency for true positives of RNN validation tests results before and after hls4ml conversion

5.4 Comparison between CNN and RNN

As demonstrated in the previous sections, both models seem to be performing very well when it comes to muon detection, with the CNN performing better towards this task based on initial results. However, its ability to generalise on the randomness of background as well as deal with the sparsity issues related to its input data, indicate the need for further optimisation in order to reach acceptable performance levels across all encountered cases. Training on larger datasets is required as well as even a better hyperparameter fine-tuning in order to increase background rejection efficiency and signal detection. A further investigation on the `hls4ml` conversion is also necessary if a CNN is to be used on FPGAs, in order to meet both the latency and performance requirements.

On the other hand, the RNN is able to generalise better on unseen datasets as its input does not suffer from the sparsity nature when compared to the CNN.

Conversion to `hls4ml` shows a far better consistency on the RNN side, meaning that if tuned correctly, the RNN should be the better candidate for an FPGA use.

The inability to distinguish displaced and prompt muons in both models is expected, as the models, at classification level do not have sufficient information to perform a pattern recognition specifically for displaced muons. Especially on the CNN side, including the whole sectors (BI, BM and BO) would certainly help the model make assumptions on the displaced muon opening angle, which is currently not possible with the currently used image formats. However, it was seen that increasing the image dimensions severely impacted the use of resources, which cannot be afforded at trigger level.

A good indication in both cases is that even though distinction between displaced and prompt muons cannot be made, both types still pass the event selection, which is currently not possible by the current muon trigger algorithm. If an event is triggered on, further possible analysis is possible at HLT, which still accomplishes the goal of this analysis.

6 Conclusions

This thesis presented the development of ML algorithms in the form of a CNN and an RNN tailored towards displaced muon triggering on hit data from MDTs of the ATLAS MS.

Throughout the timespan of this work, several model architectures were considered in order to meet the hardware challenges set by the SL of the ATLAS MS, as well as its trigger implementation nature, which both require lightweight models for fast decision making under the upper set latency limit.

In both cases, high performance models were achieved, when it comes to muon detection, reaching muon detection efficiency of even up to 99% on the CNN side, as well as an up to 96% actual hit background rejection.

On an attempt to distinguish displaced muons over a dark photon displacement threshold of 200 mm, both models failed the task as over 80% (at best case) of muons under the displacement threshold were being misclassified.

This outcome was expected due to the nature of the input images used (especially on the CNN side, not taking advantage of the full detector coverage each time), as well as the lack of infor-

mation the neural networks could have for making accurate assumptions on displacement. The implementation of a displacement regressor to provide a displacement score, as was mentioned in the current solution proposal in Section 3.1 is necessary to trigger more accurately on displaced muons, but proved to be out of the scope of this work, due to the increased complexity and limitations faced. However, it is worth mentioning that the current developed algorithms provide a necessary and fundamental basis for the implementation of such a displacement regression algorithm.

It is also notable that even the current algorithms allow for displaced muon triggering up to a level of accuracy, since they allow displaced muons to pass event selection and thus can make them available for analysis at the HLT.

Validation results on unfiltered data, showed better generalisation capabilities on the RNN side, further indicating the additional challenges a CNN implementation has to overcome in order to deal with the sparsity nature of its input data, while perhaps the most important reason for this generalisation failure is the inability to use images containing information from the whole barrel, which automatically leads to less globalised information per input image.

Conversion to `hls4ml` showed a superior compatibility for the RNN, achieving a better in overall consistency prior and after `hls4ml` conversion, however, both models suffered from decrease in performance, which can be attributed up to a degree to the quantization process occurring during the conversion.

Both models failed to meet the latency requirements, but through bit parallelisation practices, the RNN model could potentially be a candidate for displaced muon triggering, with the CNN requiring further investigation in order to reach optimal conversion levels and be functional for FPGA use.

6.1 Potential future work

The work that took place during this analysis serves as groundwork for the implementation of a muon trigger at L0 for the ATLAS MS utilising ML techniques. The completion of this task to its full extent was beyond the scope of this analysis, however the work done in this thesis will be further optimised by the NGT team to reach implementation level.

Challenges faced throughout this analysis prevented the optimisation of the `hls4ml` implementation as well as the implementation of a displacement regressor, that could be a very useful metric towards the detection of displaced muon signatures. It should be noted that `hls4ml` implementation could not reach its full extent also due to hardware limitations that required higher computing power, which was not available during the span of this work.

The use of RPC data could potentially be a very useful resource, as RoIs could be set and minimise the area that the algorithm should perform its check on.

Better performance could also be potentially achieved with different ML strategies that were not covered in this analysis, as well as even with a better optimisation of the current algorithms. More investigation can certainly be made in the context of this work that could potentially lead to a breakthrough for BSM physics studies in the future.

References

- [1] ATLAS Collaboration. Trigger & Data Acquisition. [Online]. Available: <https://cds.cern.ch/record/2775198/files/Trigger%20and%20Data%20Acquisition.pdf>
- [2] ———. The ATLAS Experiment. [Online]. Available: <https://atlas.cern/about>
- [3] ———. The ATLAS Collaboration. [Online]. Available: <https://cds.cern.ch/record/2759656/files/The%20ATLAS%20Collaboration.pdf>
- [4] A. Collaboration, “Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System,” CERN, Geneva, Tech. Rep., 2017. [Online]. Available: <https://cds.cern.ch/record/2285584>
- [5] T. H. Nady Bakhet, Maxim Yu. Khlopov, “Phenomenology of Hidden Abelian Higgs Model at Hadron Colliders using Neural Networks and Monte Carlo Techniques (Gauge Sector Z_{HAHM} Scalar Sector H),” *Inspire HEP*, 2015.
- [6] Next Generation Triggers (CERN). WP2: Enhancing the ATLAS Trigger and Data Acquisition. [Online]. Available: <https://nextgentrigger.web.cern.ch/wp2/>
- [7] Antonella Del Rosso. (2024) The next-generation triggers for CERN detectors. [Online]. Available: <https://home.cern/news/news/computing/next-generation-triggers-cern-detectors>
- [8] CERN. The Large Hadron Collider. [Online]. Available: <https://home.cern/science/accelerators/large-hadron-collider>
- [9] CERN. Experiments. [Online]. Available: <https://www.home.cern/science/experiments>
- [10] ———. A vacuum as empty as interstellar space. [Online]. Available: <https://home.web.cern.ch/science/engineering/vacuum-empty-interstellar-space>
- [11] Andreas Hoecker. (2025) A transformative leap in physics: ATLAS results from LHC Run 2. [Online]. Available: <https://atlas.cern/Updates/Feature/Run-2-Physics>
- [12] ATLAS Collaboration. ATLAS Run 3 Resources. [Online]. Available: <https://atlas.cern/de/node/37367>
- [13] ———. The Detector Overview. [Online]. Available: <https://cds.cern.ch/record/2759657/files/Detector%20Overview.pdf>
- [14] Bianchi, Riccardo Maria (CERN). (2022) Schematic of the ATLAS Experiment. [Online]. Available: <https://cds.cern.ch/record/2837191>
- [15] A. Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider,” *JINST* 3 S08003, 2008.
- [16] F. A. B. Abbott, J. Albert, “Production and integration of the ATLAS Insertable B-Layer,” *JINST* 13 T05008, 2018.
- [17] ATLAS Collaboration. Magnet System. [Online]. Available: <https://atlas-public.web.cern.ch/Discover/Detector/Magnet-System>

-
- [18] H. Ten Kate, “Superconducting magnet system for the atlas detector at cern,” *IEEE Transactions on Applied Superconductivity*, vol. 9, no. 2, pp. 841–846, 1999.
- [19] A. Collaboration, “ATLAS muon spectrometer : Technical Design Report,” CERN LHC ; ATLAS, Tech. Rep., 1997.
- [20] Dominguez, Daniel (CERN). (2021) Highlight of the ATLAS Muon Spectrometer. [Online]. Available: <https://cds.cern.ch/record/2777214/files/muon%20spectrometer%20highlight.png?version=1>
- [21] LHCC, “ATLAS Muon Spectrometer : Technical Design Report,” CERN, Geneva, Tech. Rep., 1997. [Online]. Available: <https://cds.cern.ch/record/331068/>
- [22] G. Aad, “Standalone Vertex Finding in the ATLAS Muon Spectrometer,” *Journal of Instrumentation*, 2014.
- [23] A. Collaboration, “The ATLAS Trigger System for LHC Run 3 and Trigger performance in 2022,” *JINST 19 (2024) P06029*, vol. 19, 2024.
- [24] G. Cattani, “The Resistive Plate Chambers of the ATLAS experiment: performance studies,” *Journal of Physics: Conference Series 280 (2011) 012001*, 2011.
- [25] M. A. Giulio Aielli *et al.*, “The RPC LVL1 Trigger System of the Muon Spectrometer of the ATLAS Experiment at LHC,” *IEEE Transactions on Nuclear Science 51(4):1581 - 1589*, vol. 51, 2004.
- [26] Song Xie, “A Gas Monitoring Chamber for ATLAS MDTs,” Ph.D. dissertation, Freiburg University, 2017.
- [27] S. Simsek, “The ATLAS RPC upgrade project for the High Luminosity LHC program,” CERN, Geneva, Tech. Rep., 2022. [Online]. Available: <https://cds.cern.ch/record/2805182>
- [28] A. Collaboration, “Resolution of the ATLAS muon spectrometer monitored drift tubes in LHC Run 2,” *JINST 14 P09011*, vol. 14, 2019.
- [29] CERN. High-Luminosity LHC. [Online]. Available: <https://home.cern/resources/faqs/high-luminosity-lhc>
- [30] A. Albano, A. Aloisio, M. Corradi, V. Izzo, and R. Vari, “A trigger demonstrator for the 10 muon barrel trigger of the atlas experiment for hl-lhc,” in *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, 2021, pp. 1–3.
- [31] ATLAS Collaboration, “Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer,” *Inspire HEP*, 2022.
- [32] CERN. Enhancing the Level-0 Muon Trigger. [Online]. Available: <https://nextgentrigger.web.cern.ch/t22/>
- [33] Caterina Doglioni, Dan Tovey. (2019) Searching for Dark Matter with the ATLAS detector. [Online]. Available: <https://atlas.cern/updates/feature/dark-matter>

-
- [34] ATLAS Collaboration. (2025) Displaced but not unnoticed: ATLAS in pursuit of Long-Lived Particles. [Online]. Available: <https://atlas.cern/Updates/Briefing/Displaced-but-not-Unnoticed>
- [35] G. L. Marco Fabbrichesi, Emidio Gabrielli, *The Dark Photon*, second edition ed. Springer, 2020.
- [36] S. G. J. S. David Curtin, Rouven Essig, “Illuminating Dark Photons with High-Energy Colliders,” *JHEP*, 2015.
- [37] M. W. W. X. Philip Ilten, Yotam Soreq, “Serendipity in dark photon searches,” *JHEP*, 2018.
- [38] CMS Collaboration, “Search for long-lived particles decaying to final states with a pair of muons in proton-proton collisions at $\sqrt{s} = 13.6$ TeV,” *JHEP 05 (2024) 047*, 2024.
- [39] G. e. a. Aad, “Beam-induced backgrounds measured in the atlas detector during local gas injection into the lhcb beam vacuum,” *Journal of Instrumentation*, vol. 19, no. 06, p. P06014, Jun. 2024. [Online]. Available: <http://dx.doi.org/10.1088/1748-0221/19/06/P06014>
- [40] A. Collaboration, “Beam-induced and cosmic-ray backgrounds observed in the atlas detector during the lhcb 2012 proton-proton running period,” *Journal of Instrumentation*, vol. 11, no. 05, p. P05013–P05013, May 2016. [Online]. Available: <http://dx.doi.org/10.1088/1748-0221/11/05/P05013>
- [41] G. Aielli, “Cavern background measurement with the ATLAS RPC system,” *Inspire HEP*, 6 2012.
- [42] S. F. J. Alwall, R. Frederix, “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations,” *JHEP 05 (2024) 047*, 2014.
- [43] N. D. Christian Bierlich, Smita Chakraborty, “A comprehensive guide to the physics and usage of pythia 8.3,” *SciPost Physics Codebases*, 2022.
- [44] A. Collaboration, “Atlas pythia 8 tunes to 7 tev data,” *Inspire HEP*, 2014.
- [45] CERN. About ROOT. [Online]. Available: <https://root.cern/about/>
- [46] IBM. (2021) What is a neural network? [Online]. Available: <https://www.ibm.com/think/topics/neural-networks>
- [47] R. N. Keiron O’Shea, “An introduction to convolutional neural networks,” *ArXiv e-prints*, 2015.
- [48] IBM. What are convolutional neural networks? [Online]. Available: <https://www.ibm.com/think/topics/convolutional-neural-networks>
- [49] Robin M. Schmidt, “Recurrent Neural Networks (RNNs): A gentle Introduction and Overview,” *ArXiv e-prints*, 2019.

-
- [50] SuperDataScience Team. Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem. [Online]. Available: <https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem>
- [51] Geeks for Geeks Team. Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem. [Online]. Available: <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>
- [52] ———. Introduction to Recurrent Neural Networks. [Online]. Available: <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
- [53] Josh Schneider, Ian Smalley. (2024) What is a field programmable gate array (FPGA)? [Online]. Available: <https://www.ibm.com/think/topics/field-programmable-gate-arrays>
- [54] FastML Team, “fastmachinelearning/hls4ml,” 2024. [Online]. Available: <https://github.com/fastmachinelearning/hls4ml>
- [55] T. Aarrestad *et al.*, “Fast convolutional neural networks on FPGAs with hls4ml,” *Mach. Learn. Sci. Tech.*, vol. 2, no. 4, p. 045015, 2021.
- [56] Vivado HLS. [Online]. Available: <https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html>
- [57] Dave Bergmann, Cole Stryker. (2024) What is a loss function? [Online]. Available: <https://www.ibm.com/think/topics/loss-function>
- [58] Abhishek Jaiswal. (2023) Understanding Optimizers in Neural Networks. [Online]. Available: <https://medium.com/@abhishek.jaiswaal1810/optimizers-in-neural-networks-4fb7adee4a63>
- [59] Geeks for Geeks Team. (2024) Learning Rate in Neural Network. [Online]. Available: <https://www.geeksforgeeks.org/impact-of-learning-rate-on-a-model/>
- [60] Sergey Ioffe, Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *ICML’15: Proceedings of the 32nd International Conference on Machine Learning*, 2015.
- [61] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.
- [63] L. W. V. T. O. S. J. C. M. Sudre, C.H., “Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, vol. 10553. Springer, Cham, 2017, p. 240–248.

A Appendix

A.1 ATLAS MDT Tables

The exact N_{tubes} encountered across all MDT chambers of ATLAS depending on barrel, φ and η station are provided across Tables 14 - 19. A slashed notation is used, whenever a difference is observed between positive and negative η stations presented as N_{tubes-}/N_{tubes+} , where N_{tubes-} and N_{tubes+} refer to the number of tubes encountered across negative and positive η stations respectively (e.g. for φ sector 7 in BIL one finds 36 tubes for η station -1 , but only 30 for η station 1). Wherever 0 tubes are encountered, those MDTs are part of barrel stations that are not covered in the scope of this thesis:

BIL MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	30	30	30	24	30	0	36/30	0
	2	36	36	36	36	36	0	36	0
	3	30	30	36	30	30	0	30	0
	4	36	36	36	36	36	0	30	0
	5	30	30	30	30	30	0	30	0
	6	36	36	36	30	36	0	30	0

Table 14 MDT counts at BIL by η station and ϕ sector

BIS MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	71	71	71	71	71	71	71	71
	2	58	58	58	58	58	58	58	58
	3	58	58	58	58	58	58	58	58
	4	58	58	58	58	58	58	58	58
	5	58	58	58	58	58	58	58	58
	6	58	58	58	58	58	58	58	58
	7	103	108	108/104	96/88	94/96	85/96	96	96

Table 15 MDT counts at BIS by η station and ϕ sector

BML MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	32/56	48/56	48	32	40/32	50	56/48	40
	2	56	56	56	56	56	56	56	56
	3	56	56	56	56	56	56	56	56
	4	40	40	40	40	40	40	40	40
	5	40	40	40	40	40	40	40	40
	6	48	48	48	48	48	48	0	48

Table 16 MDT counts at BML by η station and ϕ sector

BMS MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	56	56	56	56	56	0	0	56
	2	48	48	48	48	48	0	0	48
	3	48	48	48	48	48	0	0	48
	4	48	48	48	48	48	0	0	48
	5	32	32	32	32	32	0	0	32
	6	48	48	48	48	48	0	0	48

Table 17 MDT counts at BMS by η station and ϕ sector

BOS MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	48/72	48/72	48/64	48	56/48	56	72/56	56
	2	72	72	48/72	72	72	72	48	72
	3	56	56	56	56	56	56	48	48
	4	72	72	72	72	72	72	72	72
	5	72	72	72	72	72	72	72	72
	6	56	56	56	56	56	56	56	56
	7	-	-	-	-	-	-	72	-

Table 18 MDT counts at BOL by η station and ϕ sector

BOS MDT count		ϕ sectors							
		1	2	3	4	5	6	7	8
η stations	1	72	72	72	48	72	0	0	72
	2	72	72	72	72	72	0	0	72
	3	72	72	72	72	72	0	0	72
	4	72	72	72	72	72	0	0	72
	5	72	72	72	72	72	0	0	72
	6	64	64	64	64	64	0	0	64

Table 19 MDT counts at BOS by η station and ϕ sector