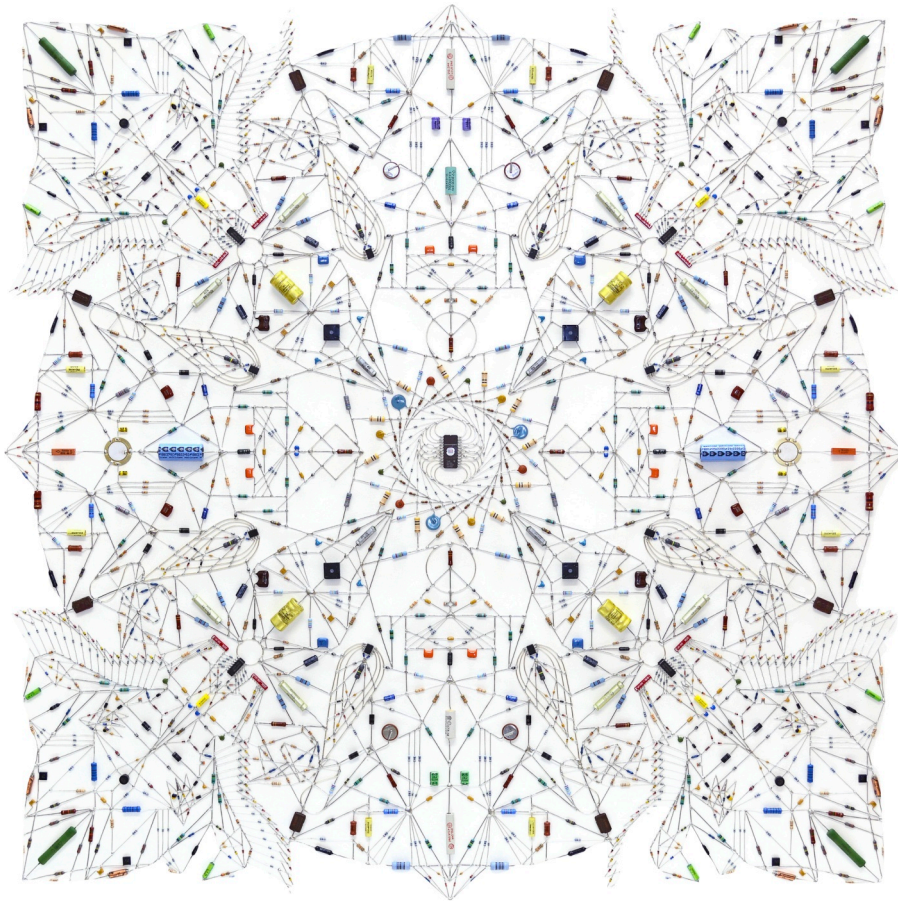


Silicon Tracking and a Search for Long-lived Particles

Rebecca Carney



Silicon Tracking and a Search for Long-lived Particles

Rebecca Carney

Academic dissertation for the Degree of Doctor of Philosophy in Physics at Stockholm University to be publicly defended on Thursday 13 June 2019 at 09.00 in sal FB42 AlbaNova universitetscentrum, Roslagstullsbacken 21.

Abstract

The ATLAS Detector, below the surface of the Swiss-French border, measures the remnants of high-energy proton-proton collisions, accelerated by the Large Hadron Collider (LHC) at CERN. Recently the LHC paused operations, having delivered an integrated luminosity corresponding to 150 fb^{-1} of data at a centre-of-mass energy of 13 TeV. This thesis describes a search for physics beyond the Standard Model using that dataset as well as the charged particle tracking detector technology that renders it possible. The analysis searches for long-lived, massive particles identified by a characteristic decay displaced from the interaction point and produced in association with high momentum jets.

Searching for rare processes requires sifting through a large amount of data, which stresses the ATLAS computing infrastructure. As such, measures are taken to reduce unnecessary computations and supplement our existing resources with, for example, inherently parallel computing architectures. Early adoption of these new architectures is necessary to understand the feasibility of their potential integration, including porting existing algorithms. A popular algorithm used in track reconstruction, the Kalman filter, has been implemented in a neuromorphic architecture: IBM's TrueNorth. The limits of using such an architecture for tracking, as well as how its performance compares to a non-spiking Kalman filter implementation, are explored in this thesis.

In 2026 the LHC will enter a High Luminosity phase (HL-LHC), increasing the instantaneous luminosity by a factor of five and delivering 4000 fb^{-1} within twelve years. This will impose significant technical challenges on all aspects of the ATLAS detector, resulting in the entire ATLAS Inner Detector being replaced by an all-silicon tracker. ITk (the new "Inner TracKer") will be comprised of Strip and Pixel detectors. The layout of the Pixel and Strip detectors was optimised for the upgrade to extend their forward coverage. To cope with the increased number of hits per chip per event and explore novel techniques for dealing with the conditions in HL-LHC, an inter-experiment collaboration, RD53, was formed, tasked with producing a front-end readout chip used in Pixel detectors. This thesis will briefly outline the author's contribution to both of these projects.

ITk silicon sensors will undergo significant damage over their lifetime due to non-ionising energy loss (NIEL). This damage must be incorporated into the detector simulation both to predict the detector performance and to understand the effects of radiation damage on data taking. The implementation of NIEL radiation damage in the ATLAS simulation framework is discussed in this thesis.

Keywords: *ATLAS, silicon, silicon tracking, radiation damage, neuromorphic, neuromorphic computing, long-lived particles, susy, rpvll, displaced vertices, pixel, pixel detector.*

Stockholm 2019

<http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-168230>

ISBN 978-91-7797-733-9
ISBN 978-91-7797-734-6

Department of Physics

Stockholm University, 106 91 Stockholm



SILICON TRACKING AND A SEARCH FOR LONG-LIVED PARTICLES

Rebecca Carney

Silicon Tracking and a Search for Long-lived Particles

Rebecca Carney

©Rebecca Carney, Stockholm University 2019

ISBN print 978-91-7797-733-9

ISBN PDF 978-91-7797-734-6

Cover image courtesy of Leonardo Ulían.

Printed in Sweden by Universitetservice US-AB, Stockholm 2019

Sammanfattning

ATLAS-detektorn, belägen på gränsen mellan Frankrike och Schweiz, mäter de partiklar som skapas i högenergetiska kollisioner mellan protoner som accelererats av CERN:s Large Hadron Collider (LHC). Efter att ha levererat en integrerad luminositet motsvarande 150 fb^{-1} vid en kollisionsenergi på 13 TeV, har LHC nu tillfälligt stängts ner för uppgraderingar.

Denna avhandling beskriver sökandet efter fysik bortom standardmodellen med hjälp av detta dataset samt den spårningsdetektorteknik för laddade partiklar som möjliggör sådana mätningar. Analysen söker efter långlivade, massiva partiklar som kan identifieras genom att de sönderfaller en bit ifrån själva kollisionspunkten.

Sökandet efter sällsynta processer kräver att stora datamängder analyseras, vilket ger en hög belastning på ATLAS e-infrastruktur. Därför behöver vi vidta åtgärder för att minska onödiga beräkningar och komplettera våra befintliga resurser med till exempel parallella datorarkitekturer. Att tidigt pröva dessa nya arkitekturer är nödvändigt för att förstå huruvida det är genomförbara att integrera dem i framtida detektorkoncept, samt i vilken utsträckning det är möjligt att överföra befintliga algoritmer. En populär algoritm som används i spårrekonstruktion, Kalman-filtret, har implementerats i en neuromorfisk arkitektur: IBM:s TrueNorth.

Begränsningarna för användningen av en sådan arkitektur för spårning, liksom dess prestanda i förhållande till en icke-spikande Kalman-filterimplementering, undersöks i denna avhandling. Under 2026 kommer LHC att gå in i en fas av högre luminositet, High Luminosity LHC (HL-LHC), vilket kommer att öka luminositeten med en faktor fem och generera en integrerad luminositet motsvarande 4000 fb^{-1} inom tolv år. Detta kommer att innebära betydande tekniska utmaningar, och hela ATLAS innersta detektorlager kommer att behöva ersättas av en spårningsdetektor med kisel i alla lager. ITk (den nya "Inner TracKer") kommer att bestå av en Strip- och en Pixeldetektor.

Layouten hos ITk:s Pixel- och Stripdetektorer har optimerats för att öka täckningen i detektorns framåtriktning. För att hantera det ökade antalet träffar per chip som en kollision i HL-LHC genererar, samt utforska nya tekniker för att hantera förhållandena vid HL-LHC bildades ett samarbete mellan flera experiment, RD53, som fick till uppgift att producera ett front-end-utläsningschip för användning i pixeldetektorer. Denna avhandling kommer kortfattat att beskriva författarens bidrag till båda dessa projekt.

ITk:s kiselnsensorer kommer att genomgå omfattande skador under sin livstid på grund av icke-joniserande energiförluster (non-ionizing energy loss, NIEL). Sådana skador måste inkorporeras i detektorsimuleringen både för att förutsäga detektorns prestanda och för att förstå effekterna av strålskador vid datainsamling. Implementeringen av NIEL-strålskador i ATLAS-simuleringsstrukturen diskuteras i denna avhandling.

Abstract

The ATLAS Detector, below the surface of the Swiss-French border, measures the remnants of high-energy proton-proton collisions, accelerated by the Large Hadron Collider (LHC) at CERN. Recently the LHC paused operations, having delivered an integrated luminosity corresponding to 150 fb^{-1} of data at a centre-of-mass energy of 13 TeV.

This thesis describes a search for physics beyond the Standard Model using that dataset as well as the charged particle tracking detector technology that renders it possible. The analysis searches for long-lived, massive particles identified by a characteristic decay displaced from the interaction point and produced in association with high momentum jets.

Searching for rare processes requires sifting through a large amount of data, which stresses the ATLAS computing infrastructure. As such, measures are taken to reduce unnecessary computations and supplement our existing resources with, for example, inherently parallel computing architectures. Early adoption of these new architectures is necessary to understand the feasibility of their potential integration, including porting existing algorithms. A popular algorithm used in track reconstruction, the Kalman filter, has been implemented in a neuromorphic architecture: IBM's TrueNorth. The limits of using such an architecture for tracking, as well as how its performance compares to a non-spiking Kalman filter implementation, are explored in this thesis.

In 2026 the LHC will enter a High Luminosity phase (HL-LHC), increasing the instantaneous luminosity by a factor of five and delivering 4000 fb^{-1} within twelve years. This will impose significant technical challenges on all aspects of the ATLAS detector, resulting in the entire ATLAS Inner Detector being replaced by an all-silicon tracker. ITk (the new "Inner TracKer") will be comprised of Strip and Pixel detectors.

The layout of the Pixel and Strip detectors was optimised for the upgrade to extend their forward coverage. To cope with the increased number of hits per chip per event and explore novel techniques for dealing with the conditions in HL-LHC, an inter-experiment collaboration, RD53, was formed, tasked with producing a front-end readout chip used in Pixel detectors. This thesis will briefly outline the author's contribution to both of these projects.

ITk silicon sensors will undergo significant damage over their lifetime due to non-ionising energy loss (NIEL). This damage must be incorporated into the detector simulation both to predict the detector performance and to understand the effects of radiation damage on data taking. The implementation of NIEL radiation damage in the ATLAS simulation framework is discussed in this thesis.

Contents

Preface	iii
1 Introduction	1
1.1 Particle physics	2
1.2 Experimental overview	7
1.3 Event reconstruction	25
1.4 ATLAS upgrades for HL-LHC	45
2 Radiation damage modeling in Pixel detector sensors	49
2.1 Silicon detectors	49
2.2 Energy deposition in silicon detectors	55
2.3 Digitization in Athena	64
2.4 Radiation damage simulation	73
2.5 Conclusion and outlook	80
3 Kalman filter in IBM’s TrueNorth	81
3.1 Neuromorphic computing	82
3.2 IBM’s TrueNorth	85
3.3 Implementation and test setup	95
3.4 Results and discussion	110
3.5 Conclusion and outlook	119
4 Search for long-lived, massive particles in multi-jet events with displaced vertices in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector	121
4.1 Introduction	121
4.2 Analysis overview	126
4.3 Event selection	128
4.4 Background rejection and estimation	150
4.5 Conclusion and outlook	158
5 Conclusion	161
Appendices	163
A Radiation damage modeling	165
A.1 Pseudocode snippets	165
A.2 A brief overview of semiconductor physics	168
A.3 List of variable name changes	175
B TrueNorth	177
B.1 Complete neuron description	177
B.2 Neuron parameter search	180

C	Additional information for DV plus jets	183
C.1	DV plus jets simulated signal grid	183
C.2	Data stream rates	183
C.3	DRAW studies	188
C.4	Displaced vertex reconstruction	190

Preface

This thesis compiles the projects I've contributed to during my degree and this preface serves to clarify my contributions to each project, as well as introducing the papers attached to the main text. As such the preface will be arranged in the order of the papers presented.

In the first year of my degree I spent a significant amount of time working on two projects which do not have dedicated chapters in this thesis, although both have resulted in publications. Instead, the three projects from my second year onwards form the main body of this thesis. The publications associated with the work described here are listed at the end of the preface.

Paper I

The first project I worked on explored the cluster properties of ATLAS IBL Pixel modules placed at shallow angles in a test beam. The modules were oriented in the beam such that the particles passed through $50\text{ }\mu\text{m}$ of silicon per pixel. As such, the front-end readout chips had to be tuned to low thresholds ($\sim 1000\text{ e}^-$) to record the signal, which required a custom tuning scheme. I also prepared tunings at 1500, 2000, and 3000 e^- . The modules used had broken low-voltage regulators on the front-end readout chip, which I manually bypassed. Overall, I prepared six Pixel modules for the extended layout testbeam which included bypassing the broken regulators on-chip, IV characteristics to determine safe operating parameters, and front-end configuration files.

For the testbeam itself I worked with the data acquisition team at SLAC to write a bit-stream converter for the output data packet, and wrote a framework for clustering to be used in data analysis. With the rest of the extended-layout working group, I helped set up the testbeam at SLAC, took shifts monitoring data-taking, and assisted in early data analysis. However, the final analysis and write-up was performed by other members of the group.

Paper II

My second project was working with the first 65 nm demonstrator produced for a front-end readout chip in HL-LHC. The chip, FE65-p2, was produced with the RD53 collaboration and allowed me to work directly with three chip designers. In that project I performed verification of the digital logic of the readout chip. This involved writing testbenches that simulated hit patterns that might be produced in the detector and testing how the chip processed them. My work in chip verification revealed discrepancies in the matching of the analogue and digital pixel matrix mapping and uncovered a couple of overlapping register definitions.

Following verification, several FE65-p2 chips were produced. FE65-p2 contains variants of a radiation-hard analogue amplifier, specifically designed to operate under the high doses received in the ITk. To test the suitability of these chips, I prepared six of the them to be irradiated at the LANSCE facility in Los Alamos National Lab. I performed measurements of the amplifier currents before and after irradiation as well as physically mounting passive components to each testboard. At Los Alamos I wrote a basic monitoring and control application over GPIB for the Keithley power supplies and took shifts over the course of the irradiation to remotely operate the chips from 30 m away.

However, as with the extended layout testbeam studies, the final analysis and paper writing were performed by other members of the group, so I chose not to include a write-up of it in my thesis.

About this thesis

The work included in this thesis represents three projects in which I have made a significant contribution throughout. The first two listed below, were also included in my licentiate thesis. The write-up of Paper III in Chapter 2 has been rewritten since the publication of the licentiate thesis whereas the write-up of Paper IV, in Chapter 3 has only minor modifications.

Paper III

The first piece of work presented in this thesis is a software project that simulates radiation damage in silicon detectors that undergo high particle fluences in the ATLAS detector. I was assigned the task of migrating a non-ionising energy loss (NIEL) damage model, written in a standalone simulation framework called Allpix, into the Athena simulation framework used by the ATLAS experiment. My task involved assessing where in the framework the simulation would be best placed, and then implementing it. During this task I noticed that portions of the Pixel digitization package would benefit from restructuring, and so in addition I restructured and rewrote significant parts of the package to optimize for clarity and performance.

After my licentiate I continued to work on the package for around 6 months, during which I continued to develop the radiation damage simulation portion of the software and began interfacing it with a centralized database of detector conditions, such that the package could be used in large-scale simulations. Ultimately, the contributions I made in those last six months were not used because the scope of the project changed.

Chapter 2 will detail both the implementation of a NIEL damage model in the Athena simulation framework and the restructuring of the Pixel digitization package. The attached paper details the software development work and a summary of the simulation's comparisons to measurements made using the ATLAS Pixel detector.

Paper IV

The IBM TrueNorth project presented a unique opportunity to work with cutting edge technology developed in the South Bay. I worked on implementing a Kalman filter in TrueNorth with David Clark, an undergraduate at UC Berkeley, supervised by Dr. Paolo Calafiura. David was tasked with producing a numerical simulation in Python to produce toy data and simulate characteristic features of the chip. This numerical simulation provided a baseline with which to compare the TrueNorth implementation to. David and I tried several approaches to implement the Kalman filter, before settling on the one described in this thesis. Whilst the formulating of crossbars was truly a joint effort, I wrote the entirety of the code for TrueNorth as well as the top-level design of the Kalman filter. I also wrote the analysis framework, conceived and performed the tests and measurements, and wrote the attached proceedings for CHEP 2016, in which I presented our work in a 15 minute presentation. Chapter 3 will detail the design, implementation, and performance of the Kalman filter in TrueNorth.

Chapter 4

In late 2017 I started working on a data analysis project, searching for physics beyond the Standard Model in data collected by the ATLAS experiment. The analysis searches for long-lived particles in the Inner Detector volume by running dedicated reconstruction for massive,

multi-track secondary vertices. I took on a several significant roles in the analysis.

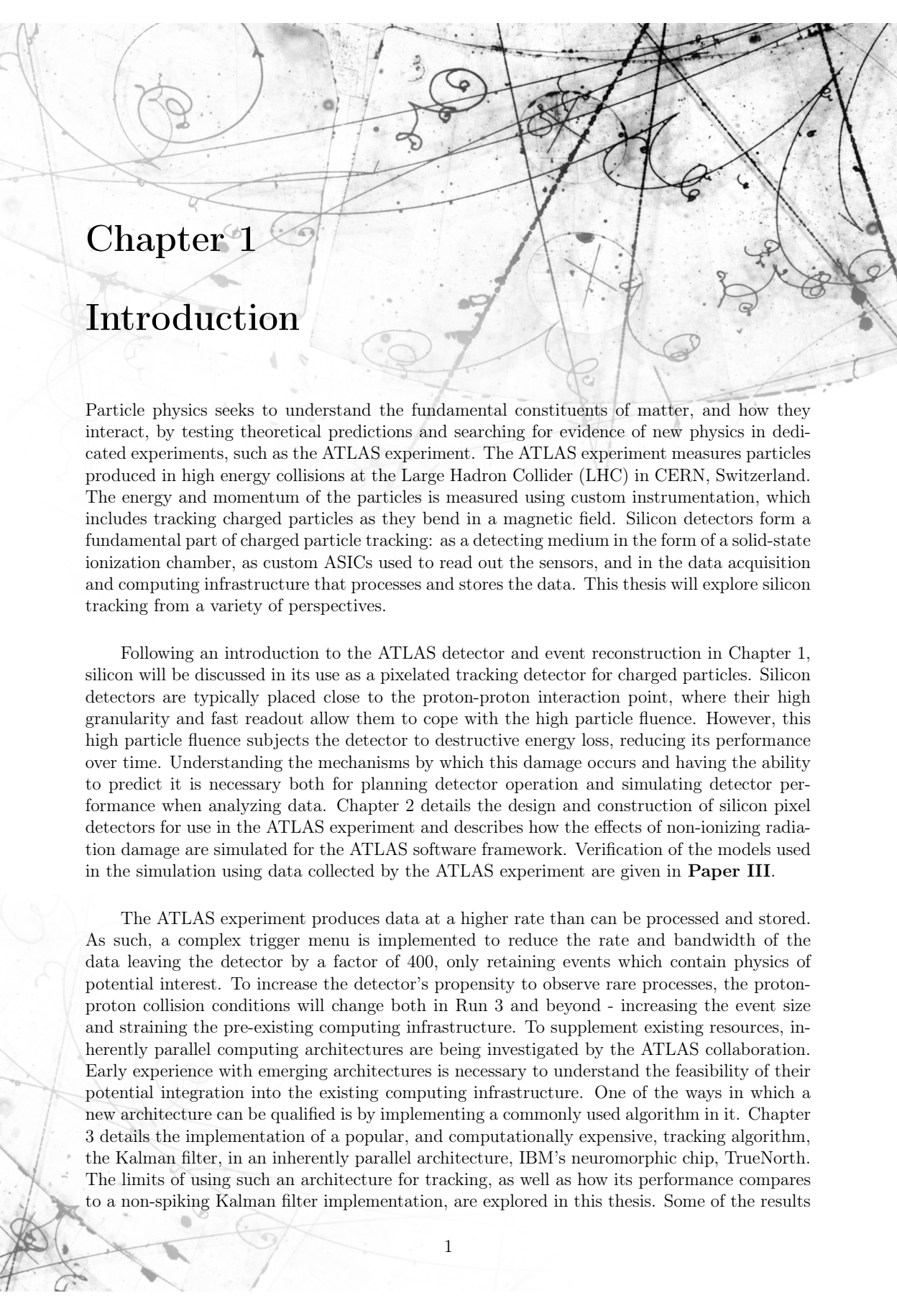
The entry-point to the analysis is a DRAW filter, which selects events of interest to undergo custom reconstruction. During the first few months I studied the filter, increasing its signal acceptance while halving its bandwidth. Whilst I did not design the data-flow for this analysis myself, I wrote the custom classes that manipulate reconstructed physics objects for analysis, as well as the the analysis framework. The long-lived particle model this analysis uses as a benchmark was simulated over several centralised campaigns that I organised and tested. The displaced vertex is a key signature that the analysis is built around and when I joined the analysis the software package that reconstructs it was undergoing major modifications. Whilst I did not contribute directly to the package development I analysed its performance for the scope of the analysis. I also studied the event- and object-level selections, particularly with regards to jet overlap-removal and cleaning. I am also responsible for one of the three major background studies in the analysis, secondary vertices promoted to a high mass by an accidental crossing.

At the time of writing there are two other students working on this analysis. Jennifer Roloff, who has studied R-hadron simulation, jet-vertex association for secondary vertices, and is responsible for the remaining two backgrounds, and Filip Backman who produced a data-driven map of the material in the detector used to remove secondary vertices from material interactions. Additionally, several students working on adjacent analyses worked to improve the secondary vertexing software that is central to this analysis.

This part of the thesis does not have an associated publication as of yet.

List of publications

- **Paper I** Simon Viel, et. al. “Performance of Silicon Pixel Detectors at Small Track Incidence Angles for the ATLAS Inner Tracker Upgrade”. In: *Nucl. Instrum. Meth.* A831 (2016), pp. 254–259. DOI: 10.1016/j.nima.2016.03.099 .
- **Paper II** Maurice Garcia-Sciveres, et. al. “Results of FE65-p2 Pixel Readout Test Chip for High Luminosity LHC Upgrades”. In: *PoS* 282 (2016), pp. 272–278. DOI: 10.22323/1.282.0272
- **Paper III** G. Aad, et. al. “Modeling Radiation Damage Effects for Pixel Sensors in the ATLAS Detector” Submitted to *JINST* (2019). arXiv: arXiv:1905.03739
- **Paper IV** Rebecca Carney, et. al. “Neuromorphic Kalman filter implementation in IBM’s TrueNorth”. In: *J. Phys. Conf. Ser.* 898. 4 (2017), p. 042021. DOI: 10.1088/1742-6596/898/4/042021



Chapter 1

Introduction

Particle physics seeks to understand the fundamental constituents of matter, and how they interact, by testing theoretical predictions and searching for evidence of new physics in dedicated experiments, such as the ATLAS experiment. The ATLAS experiment measures particles produced in high energy collisions at the Large Hadron Collider (LHC) in CERN, Switzerland. The energy and momentum of the particles is measured using custom instrumentation, which includes tracking charged particles as they bend in a magnetic field. Silicon detectors form a fundamental part of charged particle tracking: as a detecting medium in the form of a solid-state ionization chamber, as custom ASICs used to read out the sensors, and in the data acquisition and computing infrastructure that processes and stores the data. This thesis will explore silicon tracking from a variety of perspectives.

Following an introduction to the ATLAS detector and event reconstruction in Chapter 1, silicon will be discussed in its use as a pixelated tracking detector for charged particles. Silicon detectors are typically placed close to the proton-proton interaction point, where their high granularity and fast readout allow them to cope with the high particle fluence. However, this high particle fluence subjects the detector to destructive energy loss, reducing its performance over time. Understanding the mechanisms by which this damage occurs and having the ability to predict it is necessary both for planning detector operation and simulating detector performance when analyzing data. Chapter 2 details the design and construction of silicon pixel detectors for use in the ATLAS experiment and describes how the effects of non-ionizing radiation damage are simulated for the ATLAS software framework. Verification of the models used in the simulation using data collected by the ATLAS experiment are given in **Paper III**.

The ATLAS experiment produces data at a higher rate than can be processed and stored. As such, a complex trigger menu is implemented to reduce the rate and bandwidth of the data leaving the detector by a factor of 400, only retaining events which contain physics of potential interest. To increase the detector's propensity to observe rare processes, the proton-proton collision conditions will change both in Run 3 and beyond - increasing the event size and straining the pre-existing computing infrastructure. To supplement existing resources, inherently parallel computing architectures are being investigated by the ATLAS collaboration. Early experience with emerging architectures is necessary to understand the feasibility of their potential integration into the existing computing infrastructure. One of the ways in which a new architecture can be qualified is by implementing a commonly used algorithm in it. Chapter 3 details the implementation of a popular, and computationally expensive, tracking algorithm, the Kalman filter, in an inherently parallel architecture, IBM's neuromorphic chip, TrueNorth. The limits of using such an architecture for tracking, as well as how its performance compares to a non-spiking Kalman filter implementation, are explored in this thesis. Some of the results

described are summarized in **Paper IV**.

Searching for exotic physics processes can require processing detector data differently than in standard physics measurements. Chapter 4 describes a search for exotic, massive long-lived particles, identified by a characteristic decay displaced from the interaction point and produced in association with high momentum jets. The specialised reconstruction techniques used to search for long-lived particles in the search are detailed in the event reconstruction section in chapter 1. This analysis demonstrates the necessity for high-granularity silicon trackers and efficient computing to reconstruct the long-lived particle decay. There is no publication as of yet, but the parts of the analysis most relevant to this thesis has been completed: the efficient and accurate reconstruction of the displaced decay through a dedicated data filter and reconstruction package.

1.1 Particle physics

Modern particle physics is built upon the foundations of the Standard Model, one of the most precisely tested and successful theories in the history of experimental science. Despite this, there are unexplained observations in nature that the Standard Model is not able to address. In this section the constituents of the Standard Model will be introduced, followed by a brief outline of suggestions for physics beyond the Standard Model, and then an outline of a theory that extends the Standard Model to account for some of these observations, supersymmetry.

1.1.1 The Standard Model of particle physics

The Standard Model (SM) is a quantum field theory that combines Quantum Chromodynamics, Quantum Electrodynamics, and the weak interaction. Individually, each theory describes the interactions of particles with a specific quantum number but unified in the SM they provide a comprehensive description of physical phenomena.

Particles can be described by their electric charge, color charge, and spin. Particles with a half-integer spin are classed as fermions, and those with an integer spin are bosons. The SM describes matter as being comprised of fundamental fermions, quarks and leptons, which interact with each other via the exchange of spin-1 bosons, see Figure 1.1. As such, these bosons are also known as the force carriers of the SM. The electromagnetic force is mediated by the photon, a massless particle, that interacts only with electrically charged particles. The strong force is mediated by massless gluons that only interact with particles that have colour charge. The weak force is mediated by the $W^\pm$ and Z bosons and interacts with all fundamental particles in the SM.

The way in which quarks, charged leptons, and some of the force carriers interact with the Higgs field via the Brout-Englert-Higgs (BEH) mechanism accounts for their mass. The BEH mechanism introduces a new particle, the Higgs boson, whose mass and self-interaction are free parameters in the theory but were measured for the first time in 2012 at LHC experiments, ATLAS and CMS [2] [3].

The fundamental fermions of the SM can be further categorized into those that interact with the strong nuclear force, quarks, and those that do not, leptons. Quarks have an additional quantum number that leptons do not, color charge. This means that in addition to having an electric charge, a quark can also be classed as red, green, or blue. Due to the principle of asymptotic freedom, bare quarks are never observed and instead are measured in colorless, ‘bound’ states. These bound states are called hadrons and are classed as either baryons or mesons depending on their makeup. Baryons consist of either three quarks or three anti-quarks, whilst mesons are formed from a quark-antiquark pair. Bound states with more than

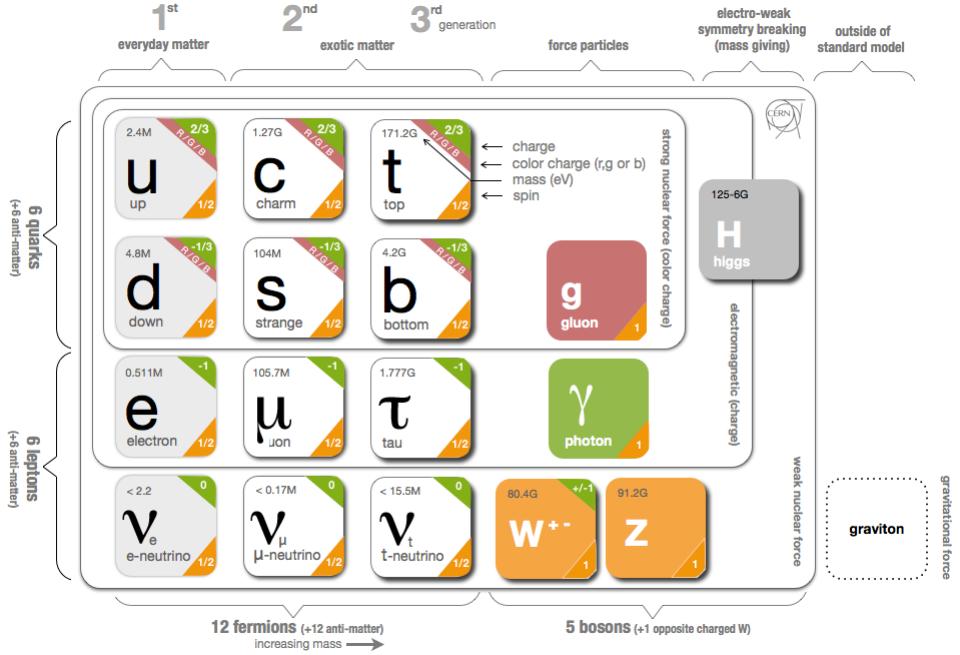


Figure 1.1: The fundamental particles of the Standard Model [1], including the hypothetical graviton.

three quarks have been observed as tetraquarks and pentaquarks [4] [5], but are far less common than baryons and mesons and quickly decay. In fact, the only stable hadron is the proton, which is comprised of two up and one down quark. Hadrons are formed from quarks through a complex process sometimes referred to as fragmentation. If a bare quark is produced in an interaction it emit gluons or other quarks until its energy falls below the hadronization scale and it hadronizes into a colorless bound state. This process leads to a cascade of hadrons which together form an object known as a ‘jet’. Jets will be discussed in more detail in section 1.3.

Quarks and leptons can be grouped into three generations, each heavier than the last. All of the stable matter in the universe is made from the first, and lightest, generation of these particles. If any of the second and third generation come into existence, they quickly decay to the first. There are a pair of fermions in each generation, for quarks the first generation is an up quark with a charge of $+\frac{2}{3}$ and a down quark with charge $-\frac{1}{3}$, whilst for leptons the first generation consists of an electron of charge -1 and an electron neutrino that is electrically neutral. There are six flavours of quarks: up, down, charm, strange, top, and bottom, and three flavours of lepton: electron, muon, and tau. The neutrinos paired with each charged lepton in their generation are labelled with the equivalent flavor, e.g. the third generation neutrino is the tau neutrino, however, unlike the other fundamental particles, the neutrino flavor eigenstates do not correspond to their mass eigenstates. Neutrinos are unusual in that there are three of them with different masses but that the flavour of each oscillates between all three generations as a function of time. This means that an electron neutrino produced in nuclear fission could be detected a few kilometres away as an electron neutrino, but also as a muon or tau neutrino [6]. The mechanism for flavor oscillation also implies that neutrinos have mass but since it is not clear if they interact with the Higgs field, it is not clear how this mass mechanism fits with what we know about the SM, nor indeed what the masses of the neutrinos actually are,

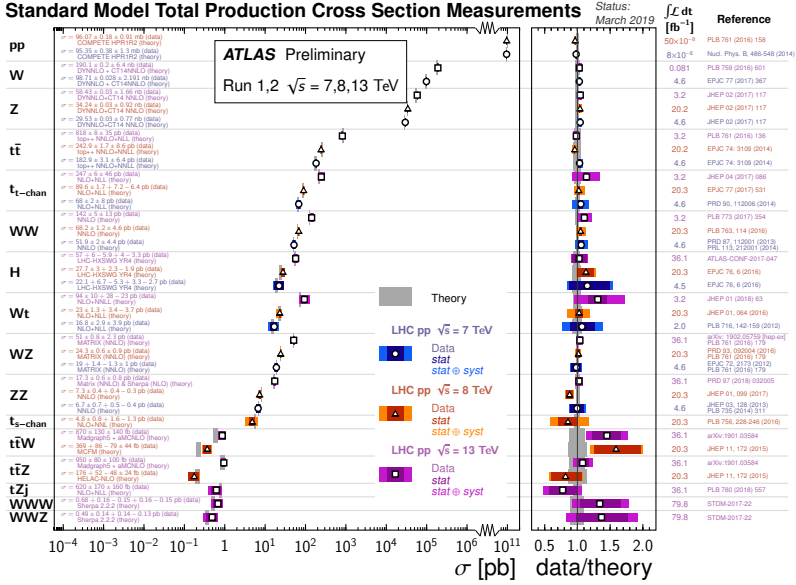


Figure 1.2: Measurements of the proton-proton total production cross-section compared to the corresponding theoretical expectations from the Standard Model [7].

although there are currently experiments that seek to answer both of these questions.

The dynamics of the twelve fundamental fermions are described by the Dirac equation, a relativistic formulation of quantum mechanics that describes fermion properties such as spin and the magnetic moment. The Dirac equation also predicts that fermions all have anti-matter counterparts, so named because, despite having the same mass and spin, they have opposite fundamental charges. The W^+ and W^- bosons are each other's antiparticle and the photon and Z boson are their own anti-particle. There are 8 gluons with various combinations of colour charge that are each other's antiparticles. It is still unclear whether the neutrinos are also their own antiparticles or if there exists a separate anti-neutrino for each matter neutrino.

All particles predicted by the SM have been observed and the SM has successfully described particle interactions over a vast range of energies. One way this can be verified is by measuring the production cross-section of a particle, which is analogous to the probability that it is produced in some interaction. Figure 1.2 shows the total production cross-section of various SM particles measured in proton-proton collisions by the ATLAS detector, as compared to their predicted cross-section by the SM, with a ratio of the two inset. The figure shows that the measured cross-sections show excellent agreement with the SM for strong, weak, and electromagnetic interactions and for bosons and fermions alike.

However, the SM cannot account for all observations in nature. For example, of the four fundamental forces, all are incorporated into the SM but gravity. And although the SM includes antiparticles, it does not account for the asymmetry of matter to anti-matter observed in the universe. Whilst neutrinos are incorporated in the SM, observations of their flavour oscillation requires them to be massive which the SM cannot account for. In addition, the SM only explains the makeup of approximately 5% of the universe as around 27% is taken up by a massive substance observed only by inference, dark matter. The existence of these unexplained phenomena indicates that there may be physics beyond the SM (BSM) that we have yet to discover.

Chapter 4 of this thesis describes a search for BSM physics that is benchmarked against a signal model from a variant of a theory called Supersymmetry. The next section will briefly describe some attributes of supersymmetry, focusing on the particles considered in the benchmark signal model and an attribute relevant to the search.

1.1.2 Supersymmetry

Supersymmetry (SUSY) is a BSM theory that extends the existing symmetries of the SM to include a fermion-boson symmetry; this combined phase-space is known as a ‘superspace’. A rotation in superspace transforms a spin- $\frac{1}{2}$ SM fermion into a spin-0 SUSY boson, a ‘sparticle’ and a spin-1 SM boson into a spin- $\frac{1}{2}$ SUSY fermion, a gaugino. For example, transforming a gluon, g , in superspace leads to a spin- $\frac{1}{2}$ gluino, $\tilde{g}$. If SUSY were a perfect symmetry, then the only difference between SM particles and their SUSY counterparts would be spin. However, to date, no evidence for SUSY particles has been found, so if they do exist their mass degeneracy with their SM super-partners must be broken. As such, the most simple, viable form of supersymmetry, the Minimal Supersymmetric Standard Model (MSSM), includes additional terms to account for the masses of SUSY particles as well as additional Higgs bosons and their superpartners, higgsinos. The superpartners to the electroweak boson, B^0 , and the neutral weak boson, W^0 , whose fields mix to form the SM photon and Z boson fields, are the bino and wino. These gauginos mix with the higgsinos to form eight mass eigenstates: four neutralinos, $\tilde{\chi}^0$, and four charginos, $\tilde{\chi}^\pm$.

Baryon and lepton symmetry is conserved in SM interactions and can be imposed in supersymmetry via the concept of R-parity, which is defined as:

$$P_R = (-1)^{\{3(B-L)+2s\}} \quad (1.1)$$

where B is the baryon number, L is the lepton number, and s is the particle’s spin¹. This definition means that SM fermions have R-parity of +1 and supersymmetric sparticles have R-parity of −1. Whilst either lepton or baryon number must be conserved to prevent rapid proton decay (which is not observed), there is no requirement that both must be. Variants of the MSSM extend the theory to include R-parity violation (RPV). The term below can be added to the MSSM superpotential to account for baryon number violating interactions:

$$\frac{1}{2}\lambda''_{ijk}\bar{u}_i\bar{d}_j\bar{d}_k \quad (1.2)$$

where λ''_{ijk} is the coupling strength, and $\bar{u}$ and $\bar{d}$ denote the up- and down- type adjoint quark spinors for quarks of generations $\{i, j, k\} = \{1, 2, 3\}$. If the coupling strength is non-zero then SUSY particles can decay to quarks, violating baryon number. This has a number of implications to the theory, including that the lightest particle in a SUSY decay chain is no longer stable and that SUSY particles need not be produced in pairs.

1.1.3 Production and decay

To verify the predictions of the SM and search for BSM physics, particles are collided at high energies in purpose-built detectors. Colliders bring particles within close proximity, causing them to interact. These interactions produce particles at measurable, and predictable rates which can be described by the particle’s production cross section, σ_p . The cross section for colliding protons can loosely be defined as:

$$\sigma_p = \sum_{ij} \int_0^1 dx_1 dx_2 f_i(x_1) f_j(x_2) d\hat{\sigma} \quad (1.3)$$

¹Unless explicitly stated otherwise, all formulae in the thesis will assume natural units where $\hbar = c = 1$.

where i, j are the indices of the partons that comprise the proton, x is the fractional momentum of the parton relative to the proton and f is probability distribution function of x [8]. The parton-parton cross-section, $\hat{\sigma}$, is calculated from the matrix element of the specific interaction process.

Particles, including those produced in a collision, that are unstable will decay into n particles at a rate Γ , that is described by the following formula:

$$\frac{1}{\tau} = \Gamma \propto \frac{1}{2M} \int |\mathcal{M}|^2 d\Phi_n(P; p_1, \dots, p_n) \quad (1.4)$$

where M is the mass of the particle decaying, $\mathcal{M}$ is the Lorentz invariant matrix element that contains information about the wavefunctions of the decaying particle and its products, P is the four-momentum of the decaying particle and p_i are the four momentum of its decay products, and Φ is the four-momentum dependent phase-space that is integrated over [8]. A particle can have multiple decay modes, in which case the total decay rate is the sum of the individual decay rates from each mode. The inverse of the total decay rate is the particle's mean proper lifetime τ , if the particle undergoes a Lorentz boost its lifetime in the lab frame will be longer. Particle decay can be described by Poisson statistics:

$$P(t) = e^{-t/\gamma\tau} \quad (1.5)$$

where P is the probability a particle lives for a time t before decaying.

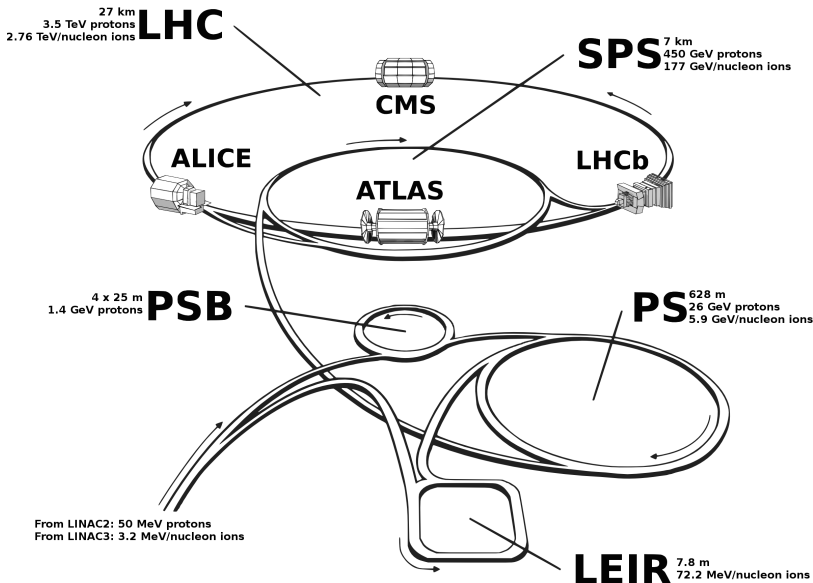


Figure 1.3: The LHC and part of the CERN accelerator complex.

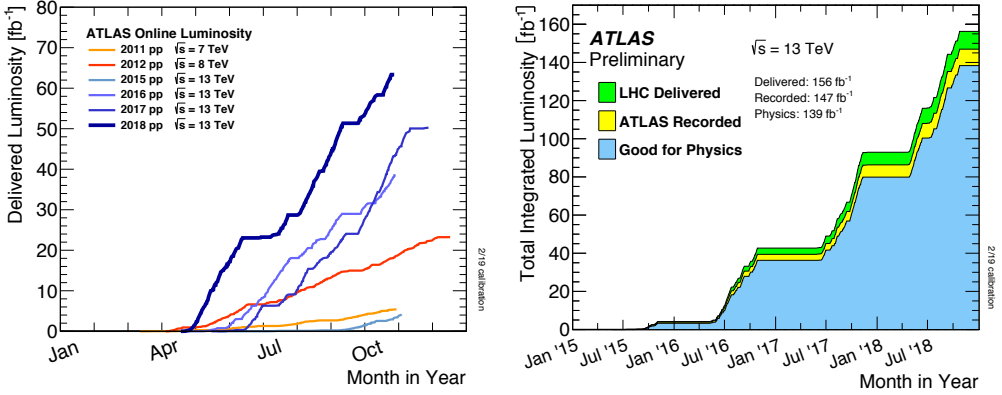
1.2 Experimental overview

To verify predictions of the Standard Model and search for BSM physics, protons are accelerated to high energies and collided in dedicated experiments. This part of the chapter will first describe the Large Hadron Collider (LHC) in section 1.2.1 before moving onto an overview of the ATLAS experiment in section 1.2.2. In section 1.3 the use of the ATLAS experiment's associated subdetectors to reconstruct properties of particles produced in LHC collision events will be discussed. The experimental overview will conclude in section 1.4 with a brief outline of planned upgrades to the ATLAS experiment for the High-Luminosity LHC, with particular focus on those topics relevant for this thesis.

1.2.1 The Large Hadron Collider

The LHC is a 26.7 km, dual-ring, superconducting, hadron accelerator and collider situated at CERN [9]. The LHC began colliding protons in 2010 at a centre-of-mass energy of $\sqrt{s} = 7$ TeV, ramping up to 8 TeV in 2012. This initial three-year period is known as Run 1. After a three-year shutdown, in which upgrades and maintenance were performed on both the collider and the experiments within it, the LHC ran for three more years, from 2015–2018, colliding protons at a centre-of-mass energy of $\sqrt{s} = 13$ TeV, a period known as Run 2.

The LHC primarily collides protons but for a fraction of its run-time also produces lead-lead, proton-lead, and xenon-xenon collisions. The LHC is fed by the CERN accelerator complex, which, along with the LHC is shown in Figure 1.3. The complex consists of several smaller accelerators starting with a linear accelerator, LINAC2, which accelerates the protons to 50 MeV. The beam is then injected into a cascade of synchrotrons: the Proton Synchrotron Booster (PSB), the Proton Synchrotron (PS), and the Super Proton Synchrotron (SPS), which accelerate the protons to 1.4 GeV, 25 GeV, and 450 GeV, respectively. From there, the protons are transferred to two beam pipes that circulate the beams in opposite directions around the LHC, accelerating them up to an energy of 6.5 TeV each.



(a) The integrated luminosity delivered by the LHC as a function of time for each year of LHC Runs 1 and 2, where each year is denoted by a different line.

(b) An overlay of the integrated luminosity in Run 2 as delivered by the LHC, recorded by the ATLAS experiment, and that is good for physics analysis.

Figure 1.4: The integrated luminosity delivered by the Large Hadron Collider (a) by year and (b) for Run 2 [10].

In the synchrotrons, particles are accelerated in an oscillating electric field amplified in a radio-frequency (RF) cavity. The nature of this acceleration method means that particles are grouped together in bunches, where a bunch consists of $\mathcal{O}(1 \times 10^{11})$ protons. The LHC has eight straight and curved segments. The curved segments house separate twin bore dipole magnetic fields and vacuum chambers that bend the beams. The eight straight segments contain regions where the two circulating beams can be brought into contact. Interaction points 1, 2, 5, and 8 house the four major LHC experiments: ATLAS, ALICE, CMS, and LHCb, respectively. The material in this thesis will focus on instrumentation and analysis of data taken at the ATLAS experiment. Points 3 and 7 contain collimation systems that ‘clean’ the beams of particles deviating from the desired bunch shape and momentum. Point 4 contains RF cavities used to accelerate the beams and point 6 contains the beam dumps.

The LHC stores and collides the two proton beams for an average of around 12 hours (a run) before a new fill is injected near to points 1 and 8. The bunches are not equally spaced but, rather, are structured according to the fill schemes from the previous accelerators in the chain that feeds the LHC. After being accelerated to the desired energy the beams are brought into collision in what is known as a bunch crossing. During Run 1, the LHC collided bunches in the ATLAS experiment every 50 ns, which was halved to 25 ns in Run 2. Each bunch is squeezed laterally by quadrupoles before each collision, to maximise the instantaneous luminosity of the bunch-crossing. The instantaneous luminosity is defined as:

$$\mathcal{L} = f_c \frac{n_1 n_2}{4\pi\sigma_x^* \sigma_y^*} \quad (1.6)$$

where f_c is the collision frequency, n_1 and n_2 are the number of particles in each bunch, and σ_x and σ_y are the root-mean-square transverse beam sizes in the x-y plane at the interaction point [8]. During a run, the time period for which the instantaneous luminosity is expected to remain constant is called a Luminosity Block (lumi-block, LB), which is typically less than 2 minutes.

An experiment records events of interest, where an event is defined here as recording the output of a detector for some time-period, producing a snapshot of the experiment. An experiment

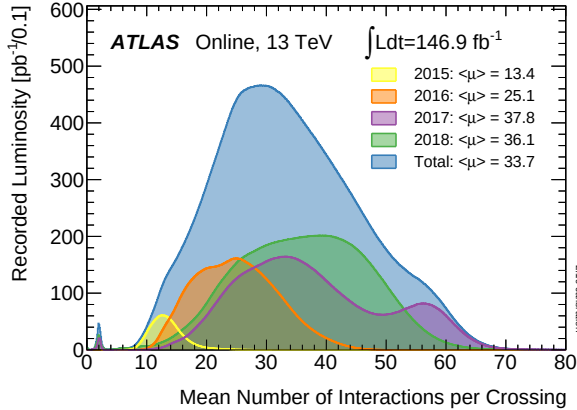


Figure 1.5: The average number of interactions per bunch-crossing, $\langle\mu\rangle$, for each year in Run 2 [10].

can record events from beam-collisions or from a process unrelated to the beam, for example cosmic-rays. The luminosity delivered by the LHC for both Run 1 and Run 2, measured by the ATLAS experiment, is shown in Figure 1.4a, divided up by year. However, the amount of luminosity delivered by the LHC does not correspond exactly to the amount recorded by the detector. Figure 1.4b shows the integrated luminosity for Run 2 as a function of time for three different scenarios: the luminosity delivered by the LHC, the luminosity delivered during which the ATLAS experiment was in a ready-state, i.e. able to record, and the luminosity delivered that the detector recorded that satisfied various data-quality requirements related to detector performance. As such, of the $\sim 150 \text{ fb}^{-1}$ of data delivered by the LHC in Run 2, $\sim 139 \text{ fb}^{-1}$ of it is fit for data analysis.

For a given physics process with production cross-section, σ_p , the number of events, N , produced from colliding beams of instantaneous luminosity, $\mathcal{L}$, that will contain that process is given by:

$$N = \sigma_p \times \int \mathcal{L}(t) dt \quad (1.7)$$

where the instantaneous luminosity is integrated over time. There can be multiple proton-proton inelastic scatters per bunch-crossing, μ , in an event. Typically only one of these results in an energetic interaction that produces physics of interest to experiments, whilst the particles produced in the other interactions can ‘pile-up’ on the collision of interest. For a given lumi-block the average number of interactions per bunch-crossing, $\langle\mu\rangle$, can be measured which is used to define the pile-up of the event. Figure 1.5 shows the average number of interactions per bunch-crossing in data taken by the ATLAS experiment in Run 2. The next section will describe the ATLAS experiment, followed by a discussion of how data recorded by the experiment’s detectors is used to reconstruct and identify particles and other objects utilized in physics analysis.

1.2.2 The ATLAS experiment

The ATLAS experiment is a roughly cylindrical, close to hermetic, multi-purpose detector, shown in Figure 1.7. The ATLAS detector was designed to achieve a number of physics goals, including but not limited to: performing detailed measurements on properties of the top quark, including its spin and couplings, understanding jet quenching in quark-gluon plasmas from

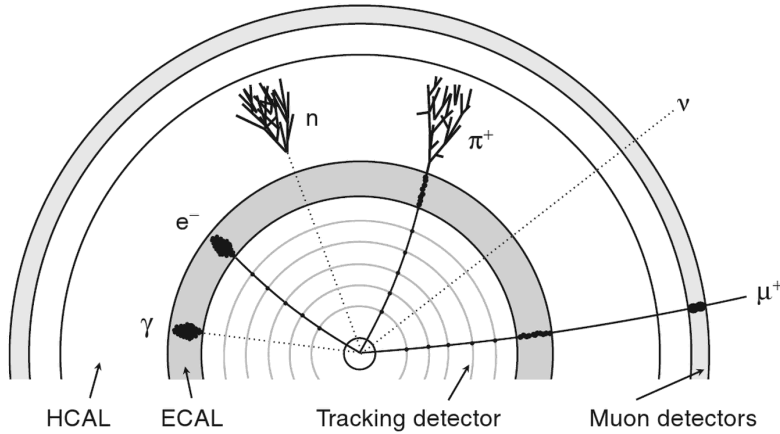


Figure 1.6: A generic general-purpose detector with concentric layers of subdetectors. Various particles that could be produced in a proton-proton collision are shown interacting with it [11].

heavy-ion collisions, and to continue the search for new physics, including supersymmetry and, at the time, the Higgs boson. Given that the scale of some of these processes was unknown, but occurred rarely enough that the new particles had not been observed, a large integrated luminosity was needed which imposed further technical requirements on the detector. With these goals in mind, the ATLAS collaboration needed a detector that had the ability to:

- Efficiently track particles at high instantaneous luminosity - this would require a trigger system to save only events of interest as the QCD jet production cross-section dominates rare physics processes.
- Measure a range of particle momenta and fully reconstruct events.
- Distinguish electrons and photons by their material interactions.
- Have as-close-to-full calorimetric coverage as possible, with enough segmentation to find unbalanced energy in the event.
- Precisely measure muon momentum in a standalone detector.
- Reconstruct and distinguish long-lived particles, such as tau leptons and b-hadrons, requiring detectors as close to the interaction point as possible, with high granularity.
- Search for rare processes, necessitating high instantaneous and integrated luminosities which require the detector to have fast and radiation-hard sensors and electronics, particularly in the detectors closest to the interaction point.

These choices [12] build up the design strategy of the ATLAS detector: concentric layers of sub-detectors with the beam pipe at their centre. The detector layers arranged in cylindrical shells are referred to as being part of the ‘barrel’, whilst the detector layers capping the cylinder are known as endcaps.

In Fig. 1.6 the interactions of some collision fragments are shown for a generic general-purpose detector, similar to the ATLAS experiment. Particles radiate out from the interaction point in the centre of the image. They first pass through a multi-layer tracking detector and, if the particles have an electric charge, deposit a small amount of energy in the detector elements.

Tracking layers are usually ionisation chambers, where the energy deposited by the particle liberates charge carriers that can be measured by dedicated electronics. The elements that have been hit can then be used to reconstruct the trajectory of the particle by connecting the dots between layers. To measure the momentum of the charged particles, a strong magnetic field is applied through the extent of the tracker. The procedure for track reconstruction and momentum measurement in the ATLAS experiment will be summarised in Section 1.3.1.

After leaving the tracker, which in ATLAS is known as the Inner Detector (ID), the particles

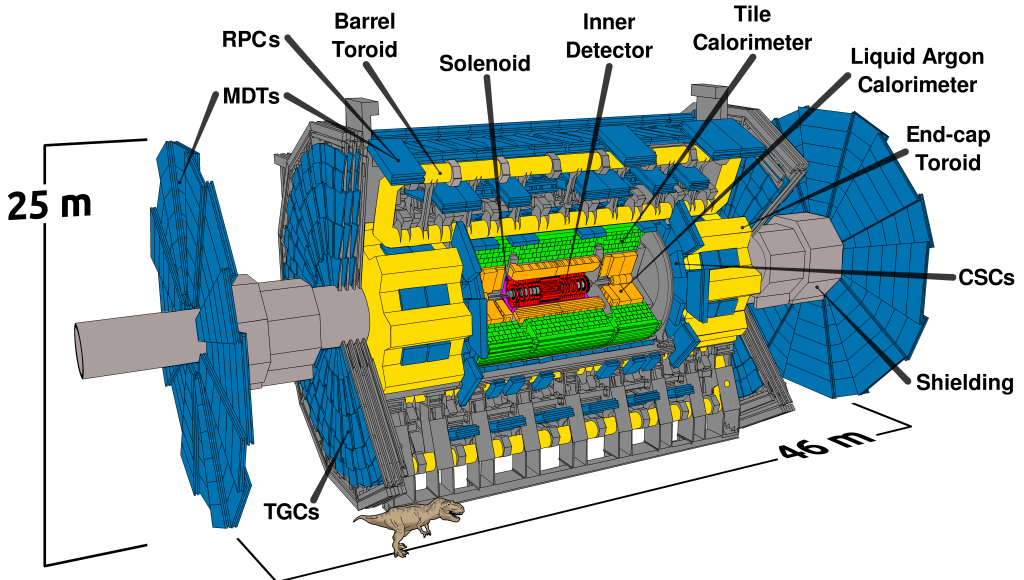


Figure 1.7: The ATLAS detector with T-Rex for scale [13]. The beam pipe runs through the axis of the detector and the particles collide within.

pass into calorimeters, built from a material with high atomic number which causes the particle to lose energy quickly, leading most to stop within the subdetector volume. The first calorimeter in the particles' path is an electromagnetic (EM) calorimeter which is made of a material designed to force electrons and photons to shower and most will be contained within it. Other particles will also lose some energy in the EM calorimeter and then penetrate into the hadronic calorimeter. The hadronic calorimeter is designed to have enough material such that even the most energetic particles are stopped within it, to ensure all the energy in the event can be measured.

However, two particles still escape the calorimeter, and detector, volume: neutrinos and muons. Neutrinos only interact via the weak force and will predominantly travel through the detector without leaving a trace. Muons do interact with the tracker but because of their mass are not stopped by the calorimeters and so an additional detector layer is placed outside the calorimeter volume to characterize them. Since muons are the only charged particles to make it through the calorimeter, their known mass and measured momentum can be used to measure their energy. This detector system therefore consists of more tracking detectors and an additional magnet system.

The ATLAS detector makes use of a cylindrical coordinate system, see Figure 1.8. The LHC beamline runs down the central axis of the cylinder and bunches are designed to interact close to the middle of this central axis, which is defined as the origin of the coordinate system. The beamline defines the z -axis and the x - y plane is transverse to the beam direction. Unless

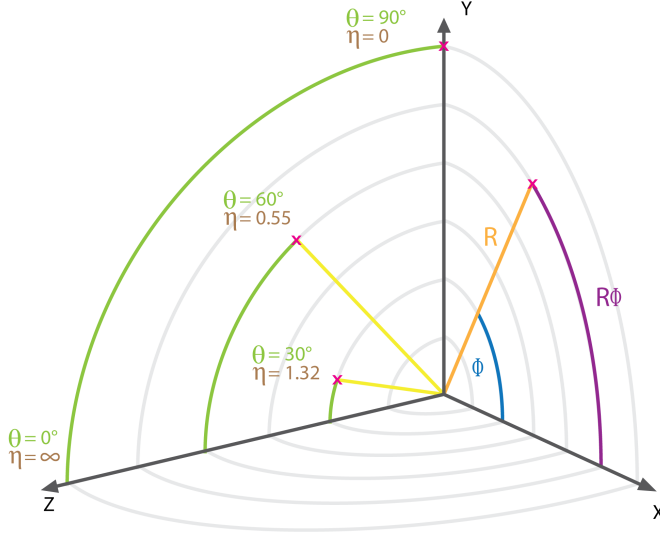


Figure 1.8: The coordinate system used by the ATLAS experiment, where the beam pipe that runs through the centre of the ATLAS detector lies along the z -axis [14].

stated otherwise, all quantities that are ‘transverse’ are defined in this plane, e.g. transverse momentum, p_T . Positive x points to the centre of the LHC ring and positive y points up to the surface, away from the cavern floor. The z -axis is defined to be positive pointing towards Geneva Airport and negative pointing towards the Jura mountains. The part of the detector in the positive- z axis is called the ‘A’-side and the other is the ‘C’-side. The azimuthal angle, ϕ , is measured with respect to the x -axis in the transverse plane. In general, R will refer to the radial distance in the transverse plane, where $R = \sqrt{x^2 + y^2}$. The polar angle, θ , is defined with respect to the z -axis in the longitudinal z - y plane. However, typically, the polar angle is not used in the atlas coordinate system and pseudorapidity is used instead:

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right) \quad (1.8)$$

Regions of the detector with small η are typically referred to as being ‘central’ whilst those at higher η are said to be ‘forward’. Pseudorapidity is generally more useful than θ as the proton partons that are involved in the hard scatter will not have perfectly balanced momenta and will likely boost the interaction products along the beam line. The difference in pseudorapidity, $\Delta\eta = \eta_1 - \eta_2$ is invariant to longitudinal boost and so can be used to compare objects boosted along the z -axis. For reconstructed objects such as jets, that have mass comparable to their momentum, a different, also Lorentz-invariant quantity is used, the change in rapidity Δy , where y here is not the coordinate axis but instead:

$$y = \frac{1}{2} \ln \left[\frac{E + p_z}{E - p_z} \right] \quad (1.9)$$

where E is the energy of the object and p_z is the longitudinal component of the momentum. Using pseudorapidity, the angular separation $\Delta R = \sqrt{\Delta\eta^2 + \phi^2}$ can be defined in ϕ and η space. This quantity is also invariant under longitudinal boosts. With this coordinate system defined, the coverage and resolution of the ATLAS detector is summarised in Table 1.1.

In the next few pages, each of the ATLAS sub-detectors will be discussed in more detail.

Detector	Resolution	$ \eta $ coverage
Inner Detector	$\sigma_{p_T}/p_T = 0.05\% \cdot p_T \oplus 1\%$	(0–2.5)
Pixel Barrel & EC		(0–2.5) & (2–2.5)
SCT Barrel & EC		(0–1.5) & (1.3–2.5)
TRT Barrel & EC		(0–1) & (1–2)
EM Calorimeter	$\sigma_E/E = 10\%/\sqrt{E} \oplus 0.7\%$	(0–3.2)
Barrel	"	(0–1.475)
EC	"	(1.375–3.2)
Hadronic Calorimeter		(0–4.9)
Barrel	$\sigma_E/E = 56\%/\sqrt{E} \oplus 5.5\%$ (pions)	(0–1)
Extended Barrel	"	(0.8–1.7)
EC	"	(1.4–3.2)
Forward	$\sigma_E/E = 100\%/\sqrt{E} \oplus 10\%$ (pions)	(3.1–4.9)
Muon Spectrometer	$\sigma_{p_T}/p_T = 1\%$ at $p_T = 1$ TeV	(0–2.7)
MDT		(0–2.7)
CSC		(2–2.7)
RPC		(0–1.05)
TGC		(1.05–2.7)

Table 1.1: Detector resolution and coverage, where $\oplus$ denotes that the quantities should be summed in quadrature and energy and momentum are measured in GeV. Note that the momentum resolution is not defined for individual tracking sub-detectors as tracks are reconstructed using their combined position resolution [15][16][17].

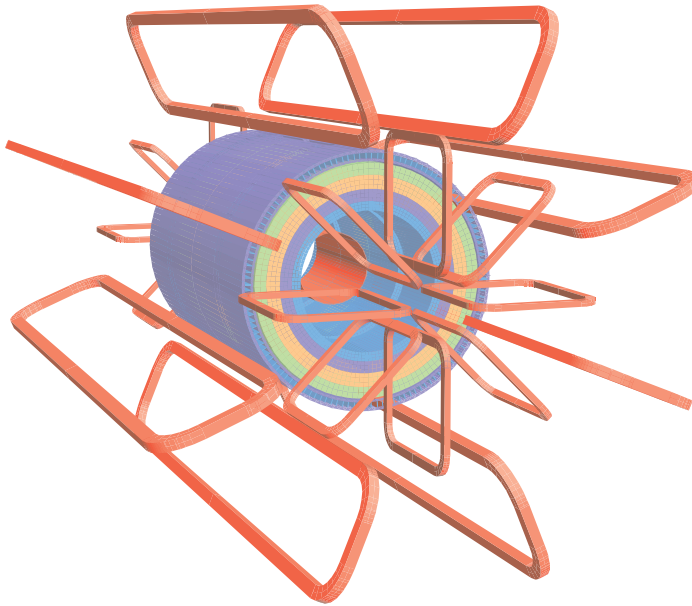


Figure 1.9: A rendering of the ATLAS magnet system. The red elements are the electromagnet windings and coils, note the solenoid in the centre of the barrel. Between the solenoid and barrel toroid is the steel in the Tile calorimeter, modeled by four different magnetic volumes (the four colors) [15].

Magnet system

The two magnet systems in the ATLAS experiment are one of the attributes that truly differentiate it from its sister experiment in the LHC, CMS. The ATLAS detector contains both a solenoid and three toroids (one surrounding the barrel, two on each end), which provide a magnetic field over $\sim 12,000 \text{ m}^3$, see Figure 1.9 [15]. The solenoid encases the Inner Detector in a $\sim 2 \text{ T}$ axial magnetic field and is $\sim 5.8 \text{ m}$ long with a radius of $\sim 1.3 \text{ m}$. The solenoid uses a superconducting niobium-titanium alloy in its windings which must be kept at $\sim 4.5 \text{ K}$. The magnetic flux is returned in the support structure and absorber of the hadronic calorimeter but the inhomogeneous distribution of materials leads to a large field gradient across the solenoid, which is discussed further in section 1.3.1. To minimise any material interactions between the ID and the calorimeters, the solenoid shares its vacuum vessel with the EM barrel calorimeter. The barrel toroid system sits outside the hadronic calorimeter barrel, surrounding both it and the endcap toroids, interspersed with the muon spectrometer. It consists of eight flat coils, wound with a niobium-titanium-copper alloy parallel to the beam axis in the longest stretches, contained within cryogenic vacuum vessels. The toroid provides a magnetic field to a much larger volume than the solenoid ($\sim 25 \text{ m}$ long with an inner and outer radius of $\sim 4.7 \text{ m}$ and $\sim 10 \text{ m}$, respectively) and as such there is a far greater range in field strength: between $0.15 - 2.5 \text{ T}$ [18]. An interesting property of the toroids is that whilst they are supported throughout their length they sag significantly due to their weight and the Lorentz-force they generate. In fact, in the centre of the top barrel toroids the structure deforms up to 26 mm , which was successfully accounted for in the design. In comparison the sag for the entire Inner Detector is less than $100 \mu\text{m}$. The endcap toroids are wound with the same alloy in eight flat coils separated by eight aluminium wedges, contained in a vacuum cryostat. The endcaps

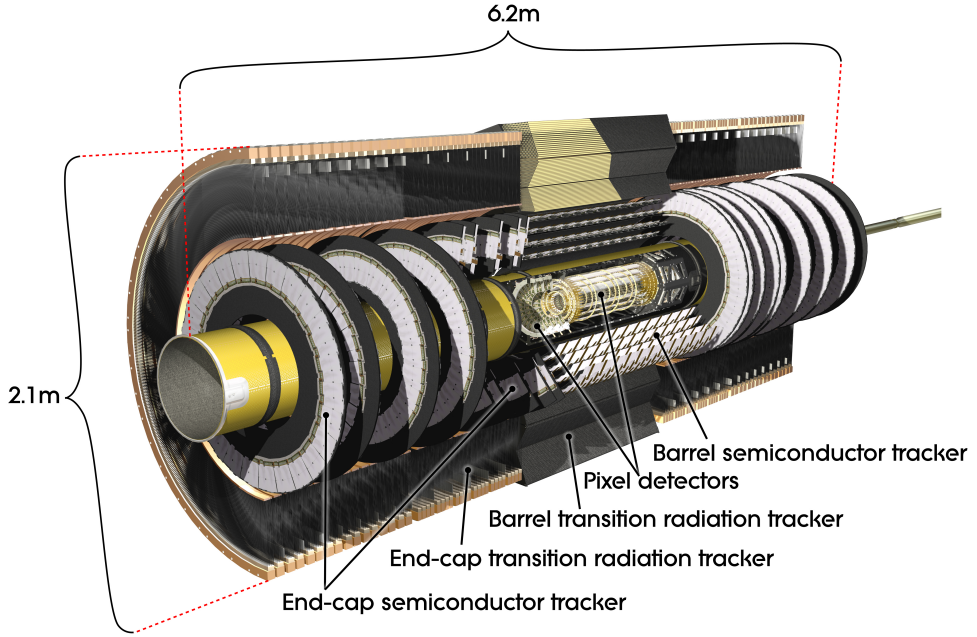


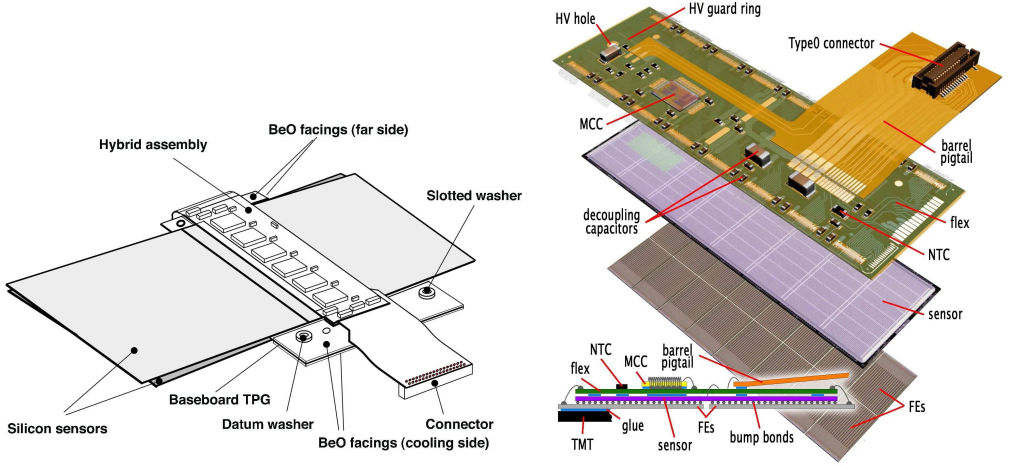
Figure 1.10: The ATLAS Inner detector, which consists of two silicon trackers: the Pixel and SCT detectors, and the Transition Radiation Tracker [19].

are arranged in a volume of inner and outer radius 0.8 m and ~ 5.4 m, respectively and also provide a wide range of field strength: $\sim 0.2 - 3.5$ T to the endcap Muon Spectrometers. The entire ATLAS magnet system is cooled by liquid helium.

Inner Detector

The ATLAS Inner Detector (ID) is a tracker comprised of three subdetectors: the Pixel detector, the Semi-Conductor Tracker (SCT), and the Transition Radiation Tracker (TRT), see Fig. 1.10.

- Pixel** The subdetector closest to the beampipe uses pixelated silicon sensors to measure the location of charged particles passing through it and is known as the Pixel detector. The Pixel detector has four barrel layers at radii of 33 mm, 50.2 mm, 88.5 mm, and 122.5 mm which are known as the Insertable B-Layer (IBL), B-Layer, Layer 1, and Layer 2, respectively. The barrel is also capped at each end by three layers of disks at $|z|$ of 495 mm, 580 mm, 650 mm [20]. While only spanning ~ 1.44 m in length and taking up 0.0075 % of the volume of ATLAS, the Pixel detector accounts for ~ 92 % of the 100 M readout channels in the entire detector due to its high granularity. The innermost layer of the Pixel detector, IBL, has $50 \times 250 \mu\text{m}^2$ pixels, the outer 3 layers and endcap modules have $50 \times 400 \mu\text{m}^2$ pixels. This high spatial resolution allows the detector to reconstruct the proton-proton interaction point with great precision: down to between 20–150 μm in x and y and between 35–300 μm in z , as well as the decays of long-lived particles such as B-hadrons [21]. This will be discussed further in the section on object reconstruction. A pixel module, sometimes referred to as a sensor assembly, loosely refers to the detector



(a) A sketch of a barrel SCT module. The hybrid assembly sits astride the front of the sensors, where the double-layers are offset by a small stereo angle. The entire module is supported on a thermally conductive structure. Information from the strips is read out via front-end chips wire-bonded to the sensors [15].

(b) A 3D model of a pixel module used in the outermost 3 layers. Sixteen readout chips are connected to the sensor by bump bonds. The readout chips are then wire-bonded to a flexible PCB glued atop the sensor, from which data is transported off-detector [22].

Figure 1.11: Pixel and SCT sensor assemblies, for barrel and endcap modules, respectively.

consisting of a pixel sensor and readout chip(s). The pixel sensor is a solid-state ionization chamber made of doped silicon and will be discussed in detail in Chapter 2, particularly with respect to its tolerance to a high particle fluence. The readout chips are mixed signal front-end CMOS ASICs designed to record and digitize the energy deposited in the sensor. The front-end and sensors are electrically and mechanically connected by lead solder bumps, see Fig. 1.11b. The Pixel detector not only records which pixels were hit in an event but also the amount of energy deposited by the particle. This is done with a charge sensitive amplifier in the front-end that integrates the charge induced in the pixel sensor and records how long the charge exceeds some pre-tuned threshold. This process effectively digitizes the analogue charge collected in the sensor into Time-Over-Threshold (ToT). The measured ToT is used, along with which pixel was hit in the interaction, to reconstruct the tracks produced. This will be covered further in section 1.3.

- SCT** The subdetector immediately outside the Pixel detector is also silicon-based and known as the SemiConductor Tracker (SCT). The SCT has four layers in the barrel at radii of 299 mm, 371 mm, 443 mm, and 514 mm, and nine endcaps at either end placed over a large range of $|z|$: 514 – 2727 mm. The SCT is sometimes referred to as the ‘Strips’ detector on account of its smallest elements being pixels with a much larger pitch in one direction than the other: $80\text{ }\mu\text{m} \times 6\text{ cm}$. Modules in the SCT contain four sensors, connected in pairs, with each pair mounted back-to-back, see Figure 1.11a. This means that each SCT layer is actually a double layer and a charged particle passing through all four subdetector layers could leave eight hits, two in each double-layer. The back-to-back sensors are mounted at a small stereo angle of 40 mradians [23]. This allows double hits on each double-layer to be combined into a 2D region with area of approximately 0.16 mm^2 , which is about $8\times$ the surface area of an individual element in the Pixel detector. The strips are read out using front-end chips connected to each strip by wirebonds, which

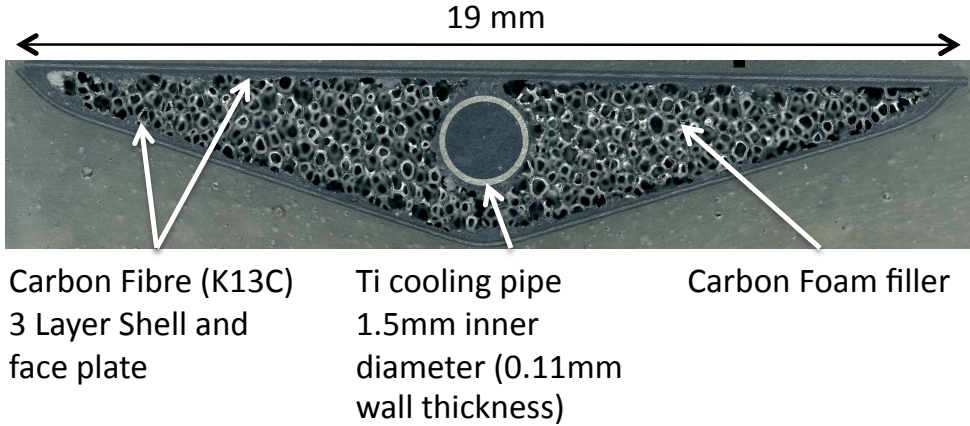


Figure 1.12: Cross-section of IBL stave showing the carbon composite foam contained in a pyrolytic graphite shell. The cooling pipe runs through the centre of the stave [24].

either sit astride the sensor in the barrel or at the end of the sensor in the endcaps on the ‘hybrid assembly’. Unlike pixels, only the presence of a hit-over-threshold, or binary hit information, is recorded.

- **Support and cooling** Both the SCT and Pixel detectors minimise their material budget to avoid altering the trajectory of the charged particle through multiple scattering interactions. As such, each use a low-mass, rigid, pyrolytic graphite composite that both supports and conducts heat away from their sensor assemblies. In the barrel, modules are arranged on staves and, in the endcaps, on petals. These support structures have pipes filled with C_3F_8 that cool the structures and in turn the modules. The IBL, the newest addition to the silicon tracker, uses CO_2 as its coolant which requires a smaller-diameter pipe than C_3F_8 , see Figure 1.12. This smaller pipe is embedded in the stave itself which is filled with a composite carbon foam, unlike the rest of the silicon tracker where the cooling pipe is only in contact with the surface of the support element.
- **TRT** Following the SCT is the Transition Radiation Tracker (TRT). The TRT is comprised of $\sim 350,000$ individual proportional drift tubes, or straws, with an anode running through their centre and an outer shell cathode. The TRT straws are flushed with a xenon-gas mixture which, paired with the large potential difference between the straw casing and the anode, causes cascades of electrons to drift towards the anode. The ToT is also recorded for the TRT, but in this case only to flag large charge deposits indicative of an electron. The TRT is able to discriminate between electrons and pions traversing the detector due to materials of different dielectric constants sandwiched between the straws. As particles with a high Lorentz factor pass through these materials, transition radiation is emitted in the form of x-rays which interact with Xe gas in the straws producing large signals. Such a setup allows the TRT to identify electrons, as due to their low mass, electrons are the only charged particles with Lorentz boost above the transition radiation emission threshold [25].

Calorimetry

The calorimeters in the ATLAS experiment serve many roles, hence the variety in their construction materials, granularity, and placement in the detector. One role of the calorimeters

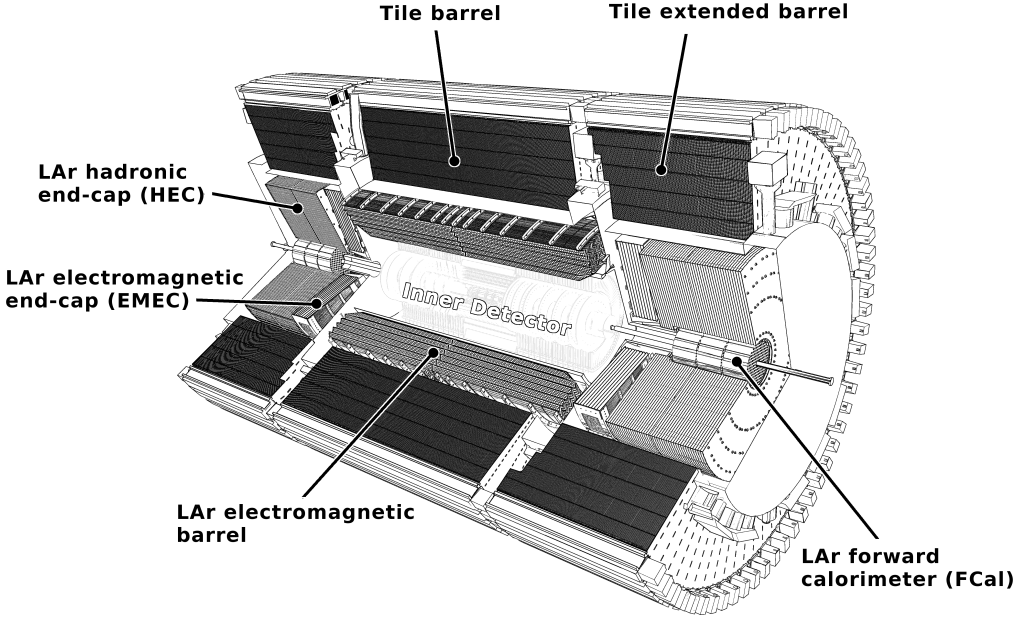


Figure 1.13: The ATLAS detector calorimeters [13].

is to contain the collision products so that their energy can be measured. This is done by constructing the calorimeters using materials with a large atomic number to promote hadronic and electromagnetic showering. The characteristic interactions of electrons and photons in a material can be used to define the radiation length:

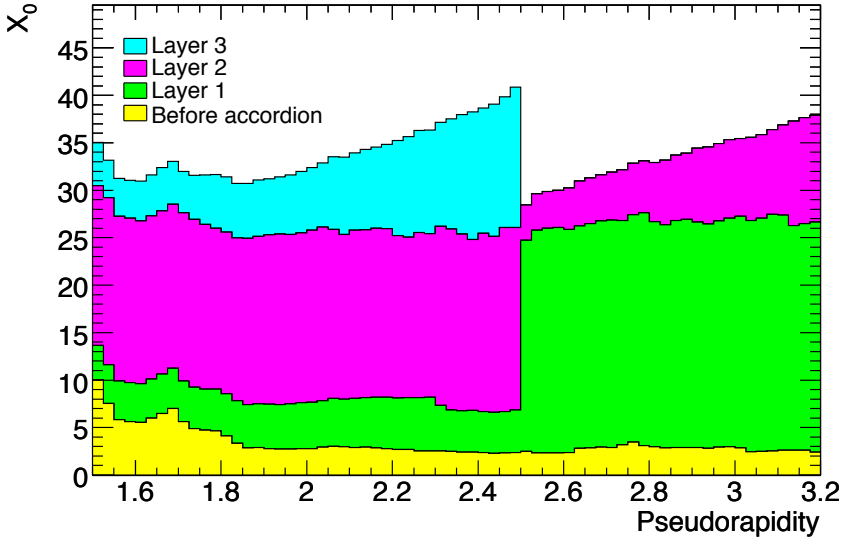
$$\frac{1}{X_0} \propto \frac{1}{A} \left(Z^2 [L_{\text{rad}} - f(Z)] + Z L'_{\text{rad}} \right) \quad (1.10)$$

where X_0 is the radiation length measured in g cm^{-2} , A is the atomic mass of the material in g mol^{-1} , Z is the atomic number of the material, L_{rad} and L'_{rad} are measured quantities that depend on the atomic number of a material, and $f(Z)$ can be approximated by an order-6 polynomial in Z [8]. Materials with greater atomic number have shorter radiation lengths. The radiation length is the mean distance over which a GeV electron has lost all but $1/e$ of its energy in Bremsstrahlung, or for a photon it is $7/9$ of the mean free path for pair-production to occur. For stopping hadrons, a different metric is used, the nuclear interaction length:

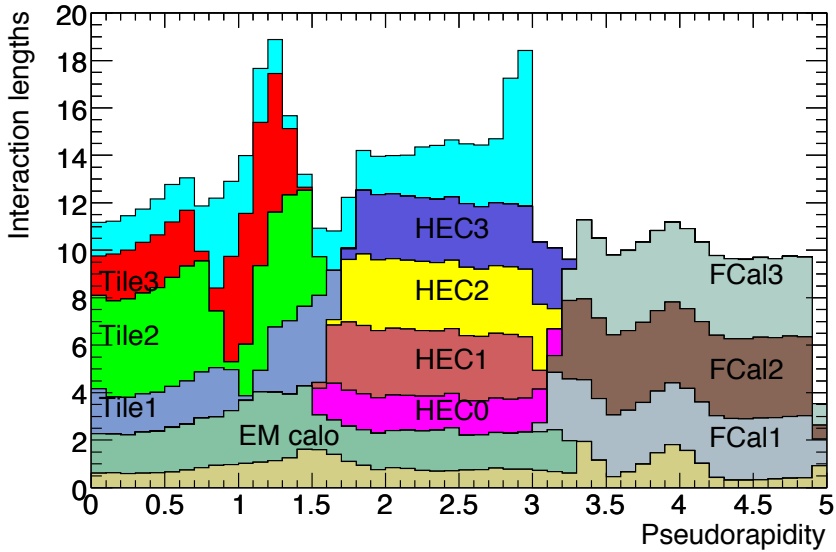
$$\lambda_I \propto A^{\frac{1}{3}} \quad (1.11)$$

where λ_I is also measured in g cm^{-2} . Both the radiation length and nuclear interaction length are tabulated for various materials in [8] and in table 1.2 for materials used in the calorimeters.

The EM calorimeter has the equivalent material for more than 22 radiation lengths in the barrel and > 24 in the endcaps - which is enough to contain all but 1×10^{-11} of an electron's energy. And for hadrons, the calorimeters have around 10 nuclear interaction lengths of material in barrel and endcap. The development of the shower is harder to predict than with electromagnetic interactions but through measurements is parameterised in terms of the energy, for example a 50 GeV pion needs $5.6 \lambda_I$ to contain $\sim 95\%$ of its shower [26]. In the ATLAS experiment the energy of most particles is measured using sampling calorimeters. Sampling calorimeters are constructed using repeating layers of two types of material: a passive absorber



(a) A stacked histogram of the number of radiation lengths in material before the hadronic calorimeters.



(b) A stacked histogram of the number of interaction lengths in each part of the ATLAS detector. From the bottom (unlabelled) is the Inner Detector, followed by the EM and hadronic calorimeters, and finally (unlabelled) the muon detector.

Figure 1.14: Properties of the ATLAS detector relevant to tracking and calorimetry [15].

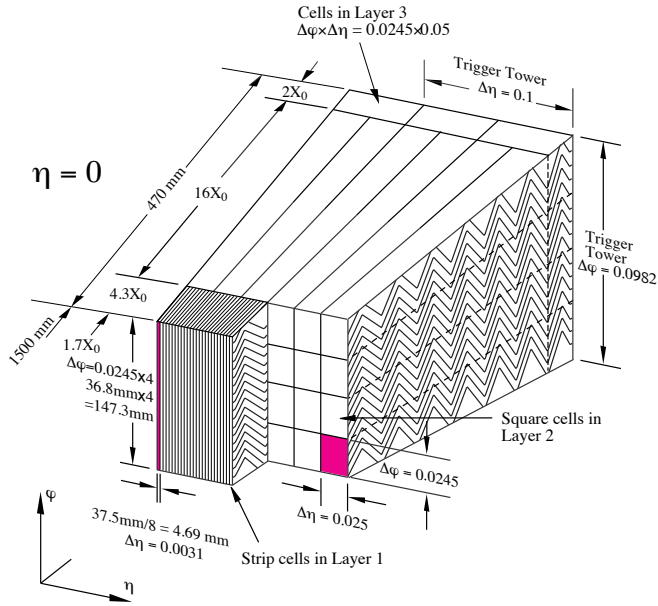
Material	X_0 g cm ⁻²	λ_I g cm ⁻²	ρ g cm ⁻³	X_0/ρ cm	λ_I/ρ cm	Usage
LAr	19.5	119.7	1.4	14.0	85.7	Sampling in FCal, HEC, EM
Cu	12.9	137.3	8.96	1.4	15.3	EM absorber in FCal and HEC
Fe	13.8	132.1	7.9	1.8	16.8	Tile absorber (in the form of steel)
Pb	6.4	199.6	11.4	0.56	17.6	Absorber in LAr EM barrel and EC
W	6.76	191.9	19.3	0.35	9.9	Absorber in FCal

Table 1.2: Radiation and nuclear interaction lengths for a selection of materials used in the ATLAS calorimeters. The lengths themselves (in cm) have also been calculated, using the density, ρ , of each material [8].

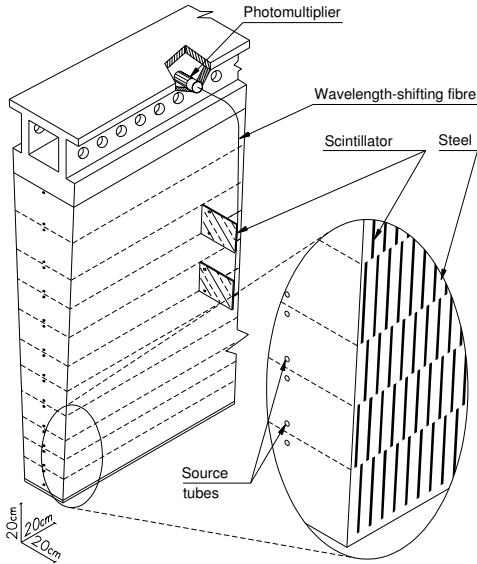
that causes the particle to shower, and an active detecting material or sampler. The ATLAS calorimeters are shown in Figure 1.13.

- The **Electromagnetic calorimeter** in the ATLAS detector is separated into barrel and endcap segments. The calorimeter uses alternating layers of lead absorber and a liquid Argon (LAr) sampler. Particles shower in the lead and, if charged, ionize the LAr. The charge is collected at copper-kapton electrodes that span the gap between the absorber with the LAr flowing in between. The segments of the EM calorimeter are tiled like an accordion to provide full coverage in ϕ , see Figure 1.15a. This fine segmentation allows for greater spatial precision when reconstructing data and the presampler allows fine pointing resolution for photons and electrons. The presampler provides an estimate of the energy lost before reaching the EM calorimeter [27].
- There are three **Hadronic calorimeters** in the ATLAS detector: Tile, the LAr endcap (HEC), and the forward LAr (FCal).
 - The **Tile calorimeter** is made of layers of iron absorbers between layers of scintillators. The scintillators are lined with fibre-optic cables connected to photomultiplier tubes that transduce the signal into a measurable current [28], see Figure 1.15b.
 - As the name suggests the **LAr endcap** sits in the high η region of the detector, after the LAr EM endcap calorimeter. LAr is used in this region as a collection medium instead of scintillator due to the higher particle flux which would damage the scintillator and lead to a much shorter lifetime. The absorber in the endcap hadronic calorimeter is copper rather than lead, as it has a smaller nuclear interaction length per unit mass. The nuclear interaction length is shown for the entire detector in Fig. 1.14b. This clearly shows that the LAr HEC calorimeter has many more interaction lengths in its volume than the EM calorimeter.
 - The **FCal** is in the very forward region of the detector, adjacent to the beam line. It's divided into 3 layers moving outwards from the centre of the detector, each with half as many sampling channels as the last. FCal layers 2 and 3 use a tungsten absorber whilst layer 1 uses copper, optimised for hadronic and electromagnetic interactions, respectively. Circular holes are bored into the absorber material and electrodes are placed inside, leaving an $\mathcal{O}(100)$ μm gap that is filled with the sampling LAr, see Figure 1.15. The FCal is very dense, coming in at 10 interaction lengths over only 1.35 m [29].

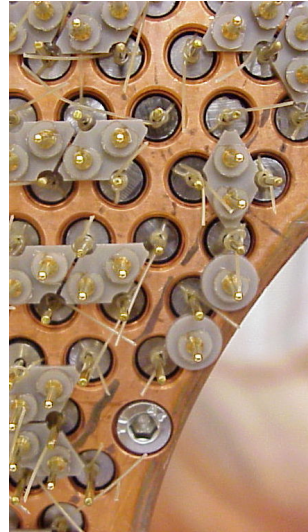
Elements of the calorimeters are clustered into towers of fixed size which are used in the trigger system. This will be discussed further in the trigger and DAQ section, later in the chapter.



(a) A sketch of part of the barrel EM calorimeter in the η - ϕ plane, showing the three different granularities in each layer, including trigger towers. The radiation lengths of each layer are also shown. The zig-zag, corrugated, accordion shape ensures full coverage in ϕ .



(b) A section of the tile calorimeter showing steel interleaved with scintillator which is transported to photo-multiplier tubes at the top of the assembly by wavelength-shifting fibre-optic cables.



(c) A photo of FCal Layer 1, showing the small clearance between the electrode and its housing, which is filled with the LAr sampler.

Figure 1.15: Details of the ATLAS calorimeter [15].

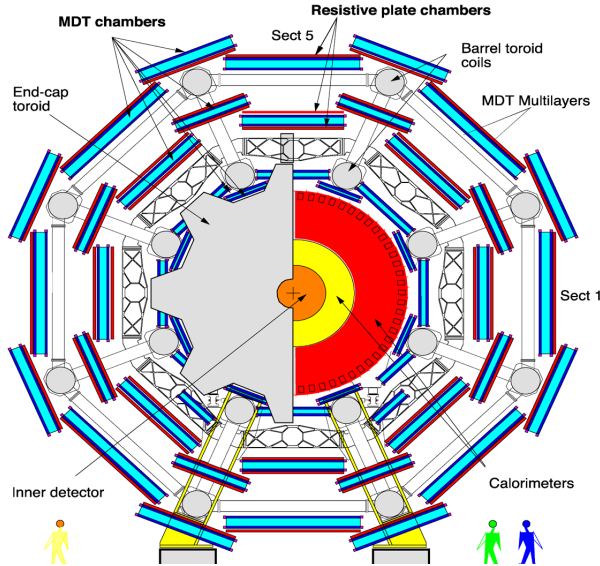


Figure 1.16: A side view of the ATLAS detector with the MDT and RPC of the muon detector clearly visible [30].

Muon spectrometer

Muons penetrate both calorimeters so an entire detector, to track and trigger on muons, is placed outside the calorimeter, called the Muon Spectrometer (MS). The MS consist of five elements: the three toroids in the ATLAS magnet system, two precision-tracking subdetectors, and two subdetectors used for triggering, see Fig. 1.16. The MS is designed to measure muon momenta between $\mathcal{O}(1) - \mathcal{O}(1000)$ GeV with subdetectors filling the space between each octant of the barrel toroid and in front of and behind the endcap toroids. The barrel subdetectors are arranged in three concentric layers at approximately: 5 m, 7.5 m, 10 m, with adjacent planes staggered to accommodate the toroids. At each endcap, muon detectors are 7.4 m, 10.8 m, 14 m, and 21.5 m from the interaction point, in $|z|$. The subdetectors are designed with radiation tolerance and excellent tracking and timing resolution leading to wide variety of technologies, outlined below:

- **Monitored Drift Tubes (MDT) chambers** provide precise tracking information, down to around $80 \mu\text{m}$, using argon/ CO_2 -filled drift tubes. An alignment system built with computer-vision software and lasers to track the detector position ensures the MDT's position is known to within $40 \mu\text{m}$ [31].
- **Cathode Strip Chambers (CSC)** track muons in the forward region of the detector, due to their higher rate capability and time resolution when compared with the MDT, which would not cope with the higher particle fluence [32]. The CSC is a multi-wire proportional chamber with segmented strips which allows precision position measurements down to $40 \mu\text{m}$.
- **Resistive Plate Chambers (RPC)** provide the first-level muon trigger and have a reduced position measurement, when compared with the MDT, in the barrel region of the detector. They consist of ionisation chambers filled with an ethane-butane based gas mixture. The high electric field and relatively short drift distance give a timing jitter of

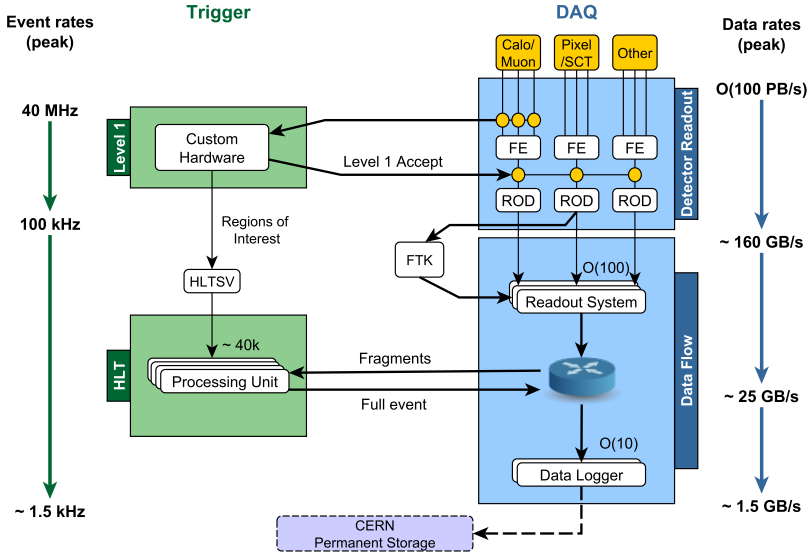


Figure 1.17: The ATLAS trigger and data acquisition system [34].

less than 2 ns [33].

- **Thin Gap Chambers (TGC)** also act as muon triggers for the forward region and provides a second position measurement to complement the CSC. The TGC's are also multi-wire proportional chambers, the thin gap refers to the separation of the anode and cathode [15].

The timing and position information from the RPC and TGC are passed into the trigger system which decides which events to retain. The next section will discuss how data from the experiment is stored and how the decision is made to store an event.

Trigger and data processing

During Run 2, the LHC produced collisions with instantaneous luminosities of up to $2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ every 25 ns, 60% higher than the peak luminosity measured in Run 1. This, combined with the higher centre-of-mass collision energies, leads to an average of 10 more interactions per bunch crossing, $\langle \mu \rangle$, i.e. a higher detector occupancy per event. This change to the run conditions has two effects: first, that the detector occupancy per event increases as result of the increased pile-up, and hence the amount of data needed to be stored per event, and secondly the rate at which events need to be stored. However, not all events produced can be read out, due to the limited bandwidth of the data acquisition system and, even if all were able to be read out they could not all be processed and stored due to the limited processing and storage facilities. As such the detector employs a complex trigger system that selects which events to read out and process, and which to flush. This section will briefly outline the ATLAS trigger system followed by a discussion of the computing and data processing of each event.

The ATLAS trigger system serves the dual purpose of selecting events that contain interesting physics, whilst reducing the amount of data recorded in an event by 4 orders of magnitude. This is done in a two-tier system, see Fig. 1.17, with a hardware-based level (Level 1, L1), and a software based trigger (the High-Level Trigger, HLT) [35]:

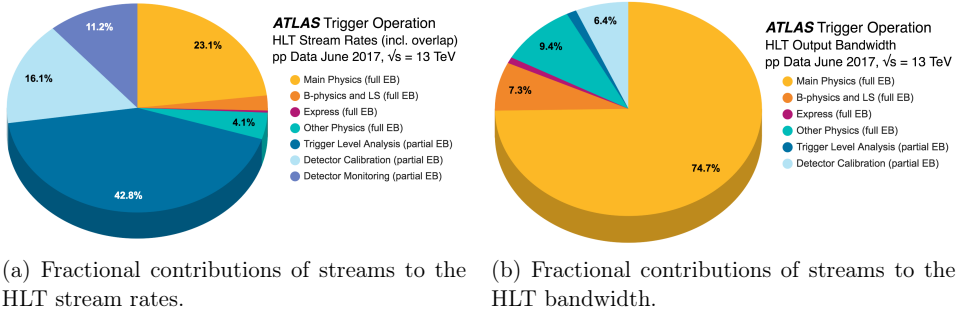


Figure 1.18: Contributions of individual streams to the HLT (a) rate and (b) bandwidth for a fill taken in June 2017 with a peak instantaneous luminosity of $L = 1.53 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and a peak pile up of 43 interactions per bunch crossing [37].

1. The **L1 trigger** reduces the event rate from 40 MHz to 100 kHz and bases its decisions on detector information from the MS and calorimeter subdetectors. In the MS, the RPC and TGC provide information about high energy muons. In the calorimeters, a sliding window of fixed size is slid over cluster of calorimeter cells [36]. If the window exceeds some energy threshold, the Region of Interest (RoI) is passed, along with the energy to the L1 trigger. The L1 trigger decision is then made by processing kinematic and RoI information from either subdetector by the Central Trigger Processor (CTP). The data is held in buffers on-detector until the L1 trigger decision is reached and has a fixed latency of $2.5 \mu\text{s}$. When the CTP decides to trigger an event, the criteria for making that decision as well as RoI, are supplied to the HLT.
2. The **HLT** reduces the event rate from 100 kHz down to 1.5–2 kHz. The HLT uses the full resolution of the calorimeter and muon detector data to reconstruct physics objects (e.g. tau particles, jets, missing energy). The HLT contains several thousand algorithms for triggering an event based on these objects. The analysis described in Chapter 4 utilizes a trigger based on L1 calorimeter deposits and HLT reconstructed jets. The sliding window technique used in calorimeter trigger towers and a description of how jets are ultimately reconstructed will be detailed in section 1.3. The HLT decision is typically made within 300 ms, after which the triggered event is read out into a specific data stream.

Data triggered in the event is divided into specific streams, some of which are ‘prescaled’ to reduce their rate. The Main Physics stream contains triggers used for data analysis, including jets. The B-physics and light states (LS) stream contains triggers specific to B-physics analyses. The express stream records events at a low rate for data quality monitoring. Finally, there are other minor streams used to monitor events not used in physics analyses, such as streams to monitor detector performance and calibration as well as the trigger-level analysis (TLA) stream. The TLA stores partial event information e.g. only four-vectors of jets for a jet-based analysis, at a higher rate than most physics streams providing analyses with a large dataset that has only a fractional impact on the trigger bandwidth because of its size, as can be seen in Figure 1.18. As such, whilst TLA’s dominate the total rate it is the Main Physics stream that is responsible for the majority of the bandwidth.

The triggered experimental data is then reconstructed into physics objects and processed for analysis within the ATLAS Distributed Computing system. The system forms a part of the Worldwide LHC Computing Grid (WLCG, often simply the ‘Grid’) which is divided into four

‘tiers’ [38]:

- **Tier 0:** Data from the HLT trigger streams is first passed to Tier 0 which performs event reconstruction at two dedicated centres. It then distributes both the raw and reconstructed data to the Tier 1 sites.
- **Tier 1:** consists of thirteen computing centres that store raw and reconstructed data, perform large-scale reprocessing, and communicate with Tier 2 facilities.
- **Tier 2:** consists of 160 smaller computing sites and can store data and typically run data analysis computing tasks, as well as running and reconstructing simulated data.
- **Tier 3:** typically refers to small, local computing clusters which are not formally part of the Grid.

Computing jobs on the grid are also supplemented by High Performance Computers, which will be discussed further in Chapter 3.

Data reconstruction and post-processing for data analysis is done in the software framework, Athena [39]. In addition to reconstruction, Athena also contains suites for simulating event generation and how those events interact with the detector. ATLAS simulates events to compare measured physics processes to theory and as a tool in analyses, but also to model how the detector interacts with particles produced in those processes. Both simulation and raw data are reconstructed in Athena using the same packages, allowing direct comparisons between the two in data analysis. The way in which Athena models detector interactions will be discussed in detail in Chapter 2. The next section will explore how raw data from the various ATLAS subdetectors can be used to construct these objects.

1.3 Event reconstruction

After passing the trigger, an event of interest is reconstructed using the full detector resolution. Reconstruction involves combining raw detector information into objects useful for physics analysis, for example, associating calorimeter deposits with a charged particle track to reconstruct an electron. This section will briefly outline how ATLAS detector data is reconstructed, with particular focus on charged particle tracks, vertexing, and jet physics, all of which are particularly important to the search for new physics discussed in Chapter 4.

1.3.1 Track reconstruction and vertexing

One of the most fundamental objects that can be reconstructed from raw data is a charged particle track. Tracks parameterise the trajectory of the particle, incorporating the sign of its charge, and allow us measure its momentum. Tracks can be clustered together to find their common point of origin, or vertex, and associated with calorimeter deposits to help identify particles. In this section, the considerations and strategy for track reconstruction in the ATLAS detector will be briefly described.

A track is formed by a charged particle depositing energy in the tracking detector in the form of ionisation. This mechanism, and the means by which the energy is collected and stored will be described in more detail in Chapter 2. With no magnetic field, and no gas or detectors present, a particle produced in the proton-proton collision would describe a straight line through space. As the charged particle moves in the ATLAS tracking detectors however, the magnetic fields present cause it to describe a quasi-helical trajectory. If we were to assume a homogenous

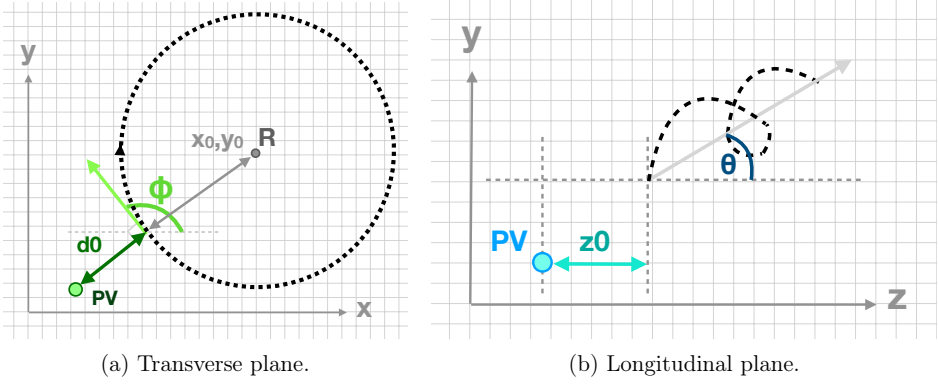


Figure 1.19: A charged particle moving in a homogenous magnetic field describes a helical trajectory, which is circular in the transverse plane and cycloidal in the longitudinal plane. In the transverse plane the particle's momentum at the distance of closest approach, d_0 , along with the angle the tangent makes with respect to the x -axis, ϕ can be used to parameterize the trajectory. Also required for perigee parameterization is the direction of curvature to measure charge sign, the distance to d_0 in the longitudinal plane, z_0 , and the constant angle the helix describes with respect to the z axis, θ .

magnetic field in the Inner Detector, then the particle's trajectory could be parameterised by a circle in the plane transverse to the magnetic field. This can be seen more clearly in Figure 1.19. By solving for this motion, an equation can be written that links the particle's momentum with the radius of the circle, R :

$$\begin{aligned}
 R &= \frac{p_T}{q \cdot e \cdot |\vec{B}|} \\
 &= \frac{p_T}{0.3 \cdot |\vec{B}|}
 \end{aligned}
 \tag{1.12}$$

where p_T is the component of the particle's momentum transverse to the magnetic field, q is the sign of the particles charge (± 1), e is the charge of an electron, $|\vec{B}|$ is the magnitude of the applied magnetic field. The factor of 0.3 can be substituted if the equation is written in natural units, assuming the particle has unit charge.

If the radius is not known however, the momentum can be reconstructed by sampling points along the trajectory, for example by setting up concentric detection layers around the interaction point. By measuring three points along the track in the transverse plane, an arc can be described. The length of the base of the arc, L , is measured as the distance between the outermost points, and the distance from centre of the base to the arc, s , can be calculated. s is also known as the sagitta of the arc. Using L and s the radius of the circle, and hence the momentum of the track, can be recovered as follows. Please refer to figure 1.20 for reference.

First, redefine R in terms of L :

$$\begin{aligned}
 \sin\left(\frac{\alpha}{2}\right) &= \frac{L/2}{R} \\
 \alpha &\approx \frac{L}{R}
 \end{aligned}
 \tag{1.13}$$

where the small angle approximation is assumed. Then, define the length u as the difference between R and the sagitta. Then, by Pythagora's:

$$u = R \sqrt{1 - \frac{L^2}{4R^2}}
 \tag{1.14}$$

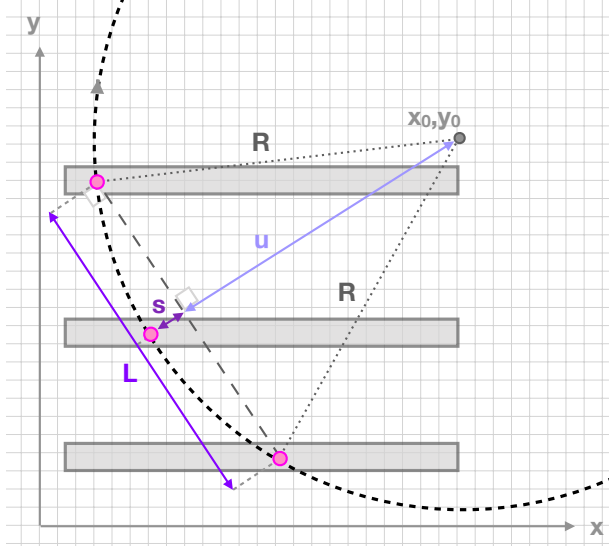


Figure 1.20: A cartoon showing the a track in the transverse plane interacting with 3 detector layers. Knowing the values of the sagitta, s , the lever arm, L , and the magnetic field strength the momentum of the particle can be calculated.

Take the binomial expansion of u :

$$\begin{aligned}
 u &\approx R \left[1 - \frac{L^2}{8R^2} + \dots \right] \\
 \text{Given } s &= R - u : \\
 s &= R - R \left[1 - \frac{L^2}{8R^2} \right] = \frac{L^2}{8R} \\
 \therefore R &= \frac{L^2}{8s}
 \end{aligned} \tag{1.15}$$

Then, subbing this expression for R back into equation 1.12:

$$p_T = 0.3 |\vec{B}| \frac{L^2}{8s} \tag{1.16}$$

By simple propagation of the measurement error, the uncertainty on the momentum measurement can be defined as:

$$\frac{\sigma(p_T)}{p_T} = \frac{8p_T}{0.3 |\vec{B}| L^2} \sigma(s) \tag{1.17}$$

This relationship shows that momentum resolution degrades linearly with p_T , which makes intuitive sense as a particle with higher momentum will bend less in the magnetic field. Equation 1.17 governs the design of both the ID and MS trackers, motivating, for example, the large separation of the innermost and outermost tracking layers to increase the lever arm L , which reduces the momentum uncertainty by $1/L^2$.

However, in reality, the magnetic field in particle detectors is inhomogeneous, the extent to which varies depending on the detector. For example, the magnetic field strength of the ATLAS Inner Detector solenoid goes from 2 T at the centre down to 0.8 T at 3 m from the centre, along its central axis: a difference of 60% [18]. The non-uniformity of the field means

that the particles trajectory is no longer a simple helix, instead its path must be deduced from solving the transport equation using numerical methods:

$$\frac{d^2 \vec{r}(l)}{ds^2} = \frac{q}{p} \left(\frac{d\vec{r}(l)}{dl} \times \vec{B}(\vec{r}) \right) \quad (1.18)$$

where $\vec{r}(l)$ is the position vector of the particle having travelled some path length, l , and the magnetic field is now position dependent. The numerical method used is a variation on the Runge-Kutta method [40], which divides the integration interval into small parabolas that are solved independently in an iterative procedure. This algorithm propagates the track through the magnetic field from one layer of the detector to the next.

Further complications to track reconstruction arise from the detector material itself, as well as its associated support structures. Charged particles lose energy via ionization, changing their momentum. The mean rate of energy loss is proportional to the material's atomic number, Z , and is well described by the Bethe formula [8], which is discussed in more detail in Chapter 2. In addition, electrons in dense material lose energy predominantly by Bremsstrahlung, at a mean rate proportional to $1/X_0$, where X_0 is the radiation length of the material as described in the previous section on calorimetry. Particles other than electrons may also undergo Bremsstrahlung but at a reduced rate as the cross-section is inversely proportional to the rest mass of the particle squared, see equation 33.30 in [8].

Whilst these energy losses change the momentum of the particle and can be accounted for in the propagator, it is multiple scattering that drastically changes the particle's trajectory. Multiple scattering is the cumulative effect of the particle undergoing many, small, Coulombic interactions from nuclei in the detector material. The resulting scattering angle, θ_{mult} , is proportional to:

$$\theta_{\text{mult}} \propto \frac{1}{p} \left(\frac{x}{X_0} \right)^{\frac{1}{2}} \quad (1.19)$$

where p is the particle's momentum and x is the distance it travels through the material [8]. To reduce the rate of energy loss due to Bremsstrahlung and deviation due to multiple scattering, it is imperative to reduce the relative radiation length x/X_0 of the material by using low mass detectors and support structures, as was described in the previous section on the ATLAS ID.

The combination of material interactions, non-uniform magnetic field, and large number of hits in the detector require the particle tracks to be reconstructed iteratively using a series of steps. For each step, the parameters describing the track can be updated. Given that the track is measured at some detector surface, spanned by local coordinates $L = \{l_x, l_y\}$, it can be described by an additional three parameters:

$$\text{Track} = (l_1, l_2, \phi, \theta, \frac{q}{p}) \quad (1.20)$$

where ϕ is the angle the tangent to the track at the detector plane makes with respect to the x -axis, θ is the angle of the track tangent makes with respect to the z -axis, and q/p is the particle's charge over momentum as before. A useful version of this parameterization is at the perigee surface, which is the distance of closest approach to a point:

$$\text{Track} = (d_0, z_0, \phi, \theta, \frac{q}{p}) \quad (1.21)$$

where d_0 is the distance of closest approach in the transverse plane, and z_0 is the distance, in the longitudinal plane, to d_0 . This track parameterization is used in the reconstruction steps outlined below.

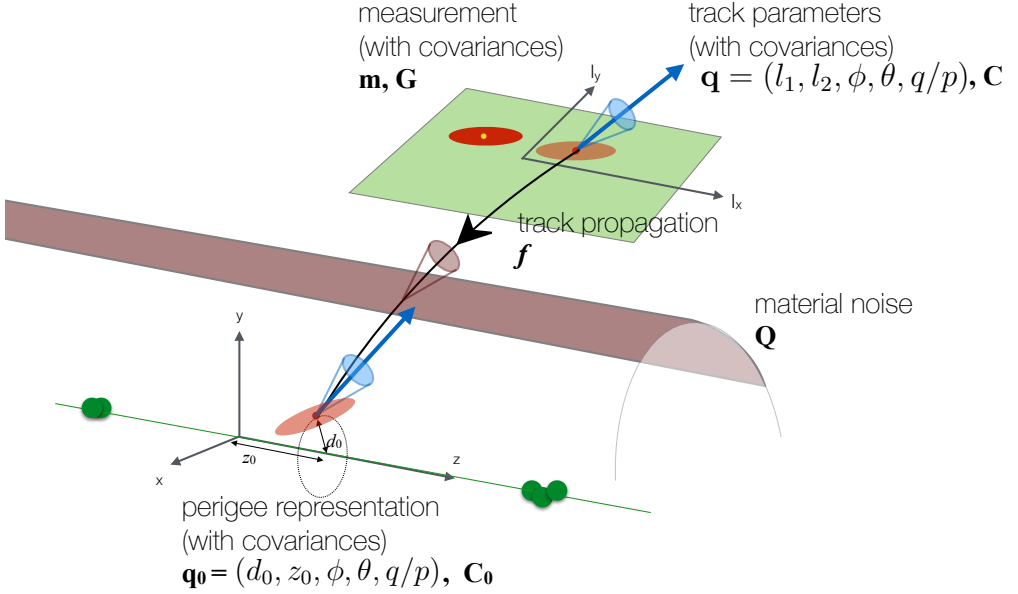
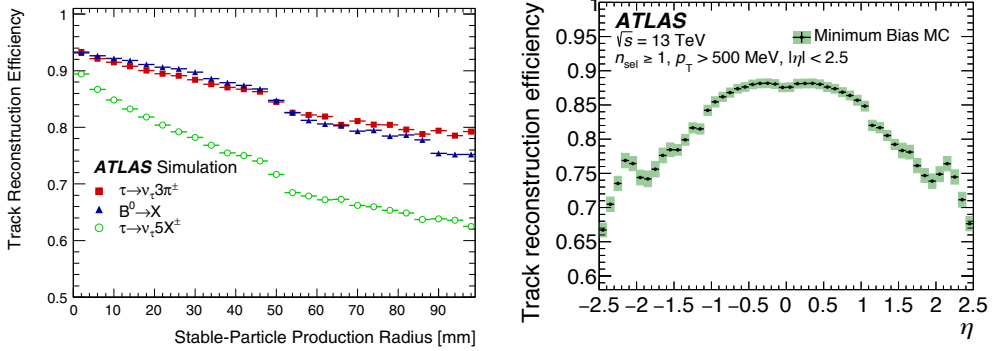


Figure 1.21: An illustration of the track propagation as part of the track reconstruction process in silicon trackers [42].

- **Creating space-points** is the first step in reconstructing particle tracks. Pixels and strips that have recorded a significant energy deposition in each layer are clustered together. For pixel clusters, a neural network is then used to identify and flag clusters that were merged but that actually come from distinct tracks [41]. The cluster centroid is formed by weighting the first and last pixel in a cluster parallel to beam line by the charge recorded. Combining the centroids of pixel clusters with the position of the associated layer of the detector creates a 3D space-point. The space-point and cluster centroid are both accessible for track reconstruction. In the SCT, space-points are made by pairing up strips with hits on back-to-back modules, and along with the layer they appear in. Since the front-end electronics in the SCT do not digitize the signal from the strips, no weighting is applied to the clustering.
- **Seed building** combines space-points, in adjacent layers, into triplets. These seeds can be built from three hits in the Pixel detector (PPP), three hits in the SCT layer (SSS), or a mix of the two (SPP). The seeds are built by connecting hits in azimuthal regions and throwing away seeds that do not make specific cuts on transverse momentum and distance from the beam spot. If the seed is matched to a hit in a fourth layer of the Inner Detector it is given a larger weight than a seed which does not.
- **Track building** starts off by estimating the track parameters from a seed and propagating them through the detector elements using the Runge-Kutta propagator, see Figure 1.21. The clusters of hits on each layer are then checked to see if their location matches any of the predictions for hit detector elements. Following this initial estimation a series of extrapolation and smoothing steps are made using an adaption the Kalman filter. Every seed leads to at least one hit collection, which is a track candidate. This process produces multiple track candidates that can share seeds or produce ‘ghosts’, which appear to exhibit



(a) Track reconstruction efficiency using standard tracking for simulated LLP [46].

(b) Track reconstruction efficiency using a soft-scattering (min-bias) simulation [47].

Figure 1.22: Track reconstruction efficiency in simulated ATLAS events for (a) SM LLP's and (b) charged particle tracks in soft-scatter events.

the properties of a track but are unphysical in some way. There is a compromise made between cuts that decrease the number of ghost tracks and that keep a good reconstruction efficiency.

- **Ambiguity solving** eliminates multiple tracks that share near-duplicate information and ghosts that are formed by valid seeds but that are not indicative of an underlying track. Ambiguity solving uses a scoring scheme that applies positive scores for unshared hits and good fit quality, and negative scores for holes, missing hits, or shared hits, and favours high transverse momentum tracks. At the end of the ambiguity solving process only tracks that pass a threshold for this scoring scheme will remain.
- **Tracks are then extended into the TRT** which increases the momentum resolution for the Inner Detector given its long lever arm [43]. The inside-out, forward tracking, methods uses a Kalman fitting-smoothing technique to extrapolate the silicon track to define a road through the TRT in which hits are added [44]. The outside-in, back-tracking, method first transforms the TRT data using the Hough transform [45]. In Hough space intersecting lines represent space-points in real space belonging to the same track. Once the TRT hits have been clustered they are extrapolated into the Pixel and SCT layers.

Whilst complex, this method yields an excellent reconstruction efficiency for a variety of particles and momenta, as seen in Figure 1.22. The plot shows the track reconstruction efficiency vs radius (a) and vs. η (b), for simulated Run 2 data. The track reconstruction efficiency is defined as the number of tracks reconstructed from simulated particles over the number of simulated particles in that bin. For both the low-momentum, prompt charged particles and the long-lived charged particles, track reconstruction efficiency is between 70–90% across the full pseudorapidity range of the detector and out to the third layer of the Pixel detector.

Once the tracks in an event are reconstructed, they are clustered together or ‘vertexed’. Vertexing, or finding the common point of origin of tracks, is a useful tool in identifying particle decays and physics processes, as well as improving the track measurement. Vertexing is a two-step process: vertices must be seeded and possibly associated tracks grouped to them, then the vertex and track positions are iteratively fitted and adjusted until a final track collection

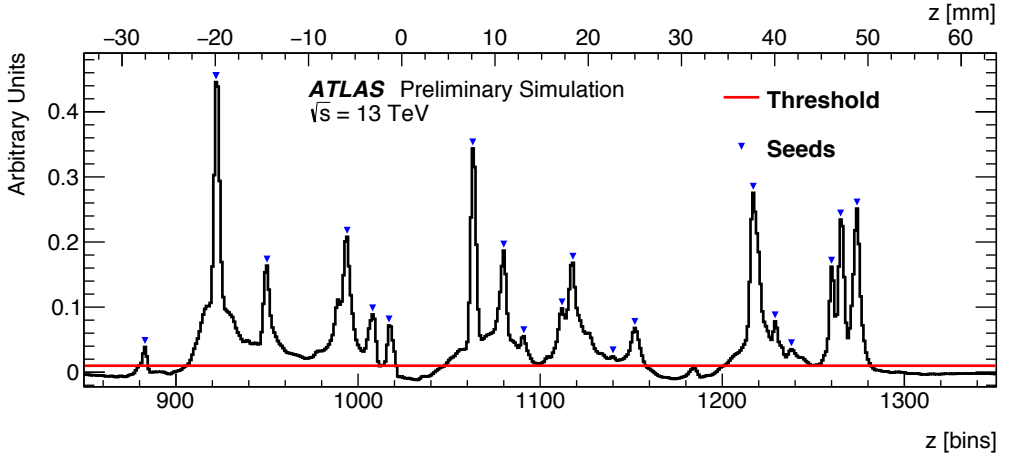


Figure 1.23: A simulated event with high track multiplicity showing all the seed primary vertices binned in z . In this event the peak at $z \approx 920$ bins is the modal value and will be used to seed the PV [48].

associated to a vertex, vertex position, and corresponding errors have converged. Vertex reconstruction can be done in various ways, but the most commonly used is the Iterative Vertex finding approach:

1. A collection of tracks seeming to originate from the beam spot, with various kinematic constraints, are compiled.
2. Using the centre of the beam spot as a reference point, the distance of closest approach, z_0 , for each track is binned. The estimated modal value is taken as the seed Primary Vertex for the initial pass, as can be seen in Figure 1.23.
3. Tracks in the collection are then iteratively fit with respect to the seed vertex, using the Kalman filter based Adaptive vertex fitting [49]. In this algorithm tracks are given weights according to their fit, which are incrementally updated as the vertex position is refitted. Using this method tracks that are a poor fit to the vertex, perhaps because they come from a different vertex or because they are in fact ‘ghost’ tracks, are gradually down-weighted. Once tracks drop below a certain weight they are removed from the vertex and can be used to construct a different one [50].
4. Steps 2 and 3 are repeated until all seed vertices have been fit.

The algorithm can be prone to merging vertices in close proximity (< 3 mm) so corrective post-processing is used to decide if a vertex should be split. Of this collection of vertices, the vertex with the largest $\sum_{\text{trk}} p_T$ is usually referred to as *the* Primary Vertex (PV) of the event. In the following sections a variation on the track and vertex reconstruction described here, that focuses on particle decays displaced from the primary vertex, will be discussed.

1.3.2 Large Radius Tracking

To reduce combinatorics and ghost tracks, various kinematic and geometric restrictions are placed on standard tracks at the seeding, track-finding, and vertexing stages. Whilst these con-

Cut type	Cut description	Standard tracking	Large Radius Tracking
Forward tracking	Min. p_T	500 MeV	900 MeV
	Max. η	2.7	5.0
	Max. $d0$	10 mm	300 mm
	Max. $z0$	250 mm	1500 mm
Clustering	Min Si hits, not shared	6	5
	Max. # of shared hits	1	2
Back-tracking	Min. p_T	1000 MeV	-
	Max. $d0$	100 mm	-

Table 1.3: Comparison of the standard and large radius tracking parameters [51].

Cut type	Cut description	Primary Vertexing	Secondary Vertexing
Track preselection	Min. p_T	400 MeV	1000 MeV
	Max. $d0$	4 mm	300 mm
	Min. $d0$	n/a	2 mm
	Max. $z0$	1000 mm	1500 mm
	Max $z0$ significance	5.0	10.0
	Min. # Pixel hits	1	0
	Min. # SCT hits	4	2 (7 if $p_T < 20$ GeV)
2-track compatibility	Max. $d0_{SV}$	-	50 mm
	Max. $z0_{SV}$	-	100 mm
	Max. χ^2/n_{dof}	-	5
	Max. SV position	-	563 mm
Track association	Max. track χ^2/n_{dof}	-	5
	Max. $\sigma(d0_{SV})$	-	5
	Max. $\sigma(z0_{SV})$	-	5

Table 1.4: Comparison of primary and secondary vertexing parameters. Note that the impact parameters and associated uncertainties for track selection are with respect to the seed vertex [50] [52].

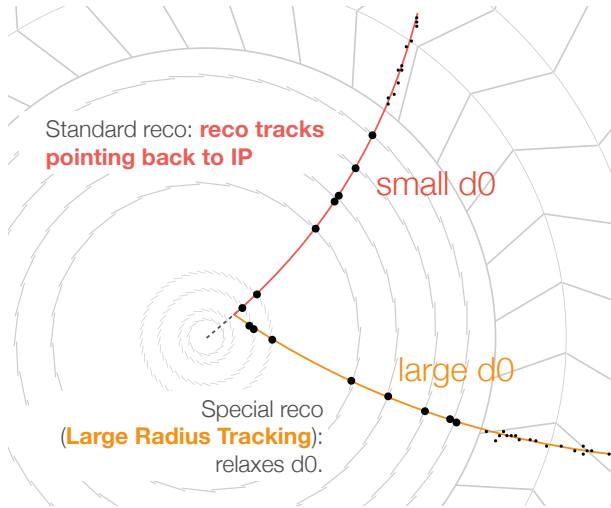


Figure 1.24: A cartoon depicting a neutral, long-lived particle that decays around 65 mm from the IP to two tracks. One track has a small transverse impact parameter, d_0 , and will likely be reconstructed by standard tracking. The other track has a much larger d_0 and would likely only be reconstructed using LRT.

straints efficiently reduce fake tracks, they restrict track reconstruction to particles originating from a relatively prompt decay. In section 1.1.2, various Standard Model and BSM particles were described that have a lifetime long enough such that some of these constraints would prevent them from being reconstructed, see Figure 1.24. As such, after standard tracking and vertexing are run, an additional set of tracks and vertices are built from a procedure optimised to search for displaced decays, known as Large Radius Tracking (LRT).

LRT builds tracks in much the same way as standard tracking, but with relaxed selections on impact parameters and silicon hits, as listed in Table 1.3. Because of these relaxed constraints, LRT is more computationally intensive than standard tracking and increases the time taken to reconstruct an event by a factor of ~ 2.5 for Run 2 data. However, not all analyses require large radius tracks so events are passed through a filter, to select only those that show additional signatures of interest, e.g. a high momentum lepton, or large energy deposits in the calorimeter. These filters will be discussed in more detail in Chapter 4.

Since LRT is run after standard track reconstruction, space-points are created with detector hits that were not used to form standard tracks. Space points are used to form seeds and those seeds are extended to tracks [51]. However, the progressive extension and fitting algorithm used for LRT tracking is a different type of Kalman filter that picks the most likely road for a track instead of creating multiple track candidates. Once the track candidates are created, ambiguity resolution and TRT extension proceed as with standard tracking but with the relaxed constraints from Table 1.3.

LRT shows significant improvement in reconstructing long-lived particles in simulation, when compared to standard tracking alone, as is shown in Figure 1.25. The figure shows the track reconstruction efficiency for a simulated, long-lived SUSY hadron, decaying at a range of radii in the ATLAS ID. The plot shows whether, for a given decay product of the LLP, the particle's track was reconstructed, binned by the LLP decay radius. The plot shows three distributions: the efficiency from using standard tracking, the efficiency from using LRT, and the combined efficiency if both are used. Small dips in the efficiency occur at radii corresponding to the Pixel detector barrel layers. This is caused by tracks in close proximity to each other at

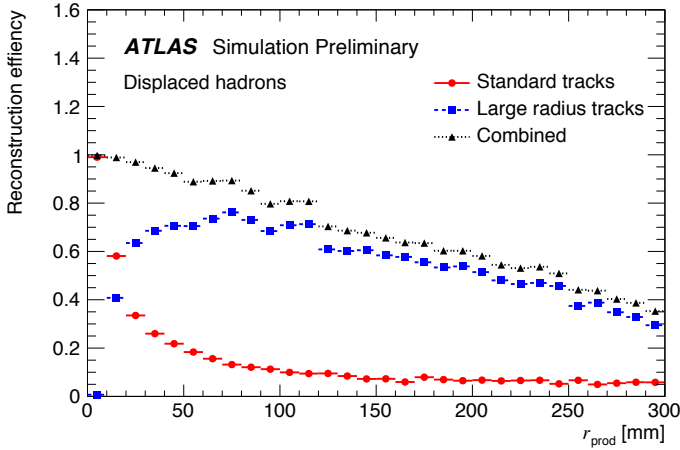


Figure 1.25: Track reconstruction efficiency comparing standard and LRT for long-lived, massive SUSY R-hadrons [51]

the LLP decay vertex being more likely to be within a pixel pitch of each other, causing them to fail the shared hit requirement from table 1.3. The plot shows that for a long-lived hadron that standard tracking has difficulty reconstructing, LRT shows significant improvement. The tracks produced from running LRT and standard tracking can both be used to reconstruct secondary vertices, displaced from the primary proton-proton interaction. The final part of this section will briefly outline this process.

Secondary vertex formation

Long-lived particles decaying within the Inner Detector volume to Standard Model particles will result in reconstructed tracks coming together at a vertex displaced from the beam spot. These vertices, depending on the mass and kinematics of the particle that formed them, can consist of a variety of tracks: some of which may point back to the primary vertex and could be reconstructed by standard tracking, and some that may have a large impact parameter (so not pointing back to the primary vertex) and may only be reconstructed by LRT. At this stage of event reconstruction, track collections from both standard and LRT have been formed and so both are used to seed the formation of the secondary vertex. The steps for secondary vertex formation are quite different to those of primary vertex formation, but they share some common parameters, which can be compared in table 1.4. Secondary vertexing proceeds as follows:

1. **Seed-track preselection** collects together tracks with $p_T \geq 1$ GeV, that have a good track fit according to their χ^2/n_{dof} value (passes back-tracking fit), and, for tracks with no hits in the innermost two Pixel layers, the track should have been extended to the TRT². Additionally other types of pre-selection such as hit-pattern matching can be applied but were not used at the time of writing.
2. **Two-track compatibility** proceeds by attempting to pair all possible combinations of tracks that pass the preselection requirements and establish whether the vertex they form passes various cuts:

²Recall that it is not a requirement in either of the track reconstruction schemes discussed previously that TRT extension is successful

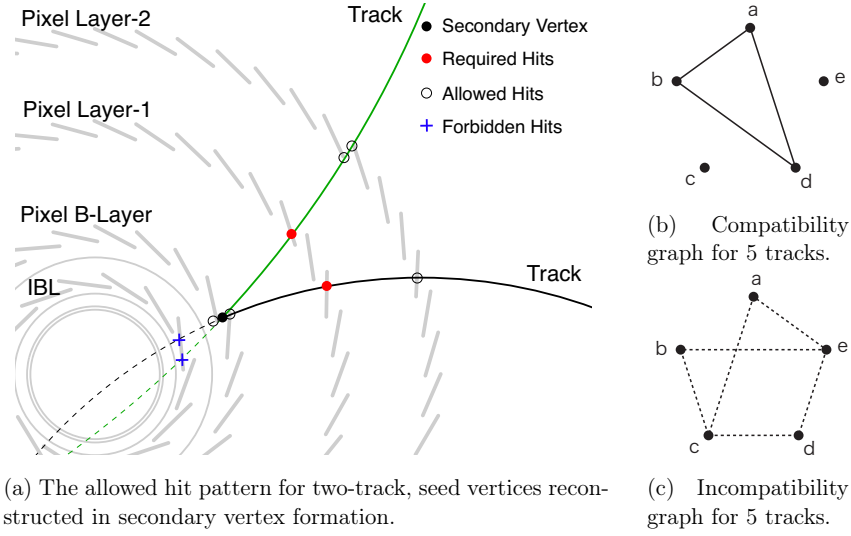


Figure 1.26: Hit pattern for seed-tracks in secondary vertex formation and a cartoon depicting the compatibility and incompatibility graphs for five tracks (nodes). The compatibility graph shows three of the tracks (a,b,d) form a vertex (arcs) and are compatible. If, instead, the incompatible nodes are connected by arcs, the incompatibility graph is produced [52].

- First, using the track parameters, the vertex position is approximated and the distance of closest approach of the two tracks to the secondary vertex (SV) is calculated. If either track has a $d0_{SV} > 50$ mm or $z0_{SV} > 100$ mm then the track pair is marked as incompatible.
 - The vertex is then fit. If the fit fails or its $\chi^2/n_{\text{dof}} > 5$, the track pair is also marked as incompatible.
 - The fitted 2-track vertex position must be within the inner radius of the TRT (563 mm) and have $|d_0| > 2$ mm, else the track pair is marked as incompatible.
 - Both seed tracks in the vertex must have a hit pattern that is consistent with the position of the vertex. This means there can be no hits-on-track in detector layers before the vertex and the next layer after the vertex must have a hit-on-track (or be a disabled module). Detector layers after the next adjacent layer may also have hits-on-track but it is not required. In the case where the vertex is in close proximity to a detector layer, the next adjacent layer is required to have a hit instead and the presence of a hit on the adjacent layer is optional. An example of this latter case is shown in Figure 1.26a. The decision on the compatibility of each track pair is saved to be used in the next stage of vertex formation.
3. The next step involves **forming the vertices**. The event now contains information about the compatibility of all possible combinations 2-track secondary vertices. Depending on the process that formed the physical vertex, some compatible 2-track vertices may have an additional compatible track or even more, i.e. the underlying vertex may be n -track where $n > 2$. To establish which tracks should in fact be clustered together in a vertex, an efficient graph theory technique is employed. The 2-track vertices are represented by

an incompatibility graph, an example of which is shown in Figure 1.26c. In this graph, tracks are represented by nodes and 2-track vertices by edges. If two nodes are connected it means that the tracks they represent were labeled ‘incompatible’ in the previous step when trying to form a seed vertex. This does not mean that the track should not be attached to the successfully formed 2-track vertex though, for example, a track may be incompatible with one track in the 2-track vertex but not with the other. The graph is solved by removing the minimum number of nodes such that all edges are likewise removed and hence all tracks in the vertex are compatible. If a track is compatible with more than one vertex the ambiguity is resolved with a goodness-of-fit metric.

4. **Vertex merging** is necessary to recombine vertices that were artificially split due to the incompatibility graph method in the previous step. First, any vertices with a separation significance³ less than 100 is not considered in the algorithms mentioned below. Vertex merging is done sequentially, starting by attempting to merge the vertex with the least number of tracks into that with the most, increasing the track multiplicity of the vertex being considered for merging with each iteration. The vertex algorithms considered for each iteration are listed below
 - *Merge by shuffling* is used to address the case in which the smaller vertex is part of the larger vertex but was reconstructed incorrectly due to its smaller position resolution. To do this, the smaller vertex is seeded with the position of the larger vertex and re-fit. If the updated position is compatible with the position of the larger vertex, the two are merged.
 - *Magnetic merging* employs a similar technique to merge-by-shuffle, but on a track-by-track basis. Tracks from the larger vertex are attached to the smaller vertex one-by-one, after which the smaller vertex is refit and if found to be compatible with the position of the larger vertex, the two are merged.
 - *Wild merging* takes all tracks in both vertices and refits them as a single vertex, seeded by the position of the larger vertex. The same criteria for merging as with the previous methods is applied.
 - *Proximity*: Any vertices within 1 mm of each other are forced to merge, regardless of the separation significance.
5. **Track attachment/association** is the penultimate step. The n-track vertices formed only contain tracks that passed the basic pre-selection criteria outlined in the first step, but they may still be a good fit to the newly formed vertices. If the tracks that did not pass pre-selection satisfy the requirements from table 1.4 they will be attached to the vertex and marked as being ‘associated’, as opposed to being used in the initial vertex formation.
6. **Final vertex fitting** is now performed, similarly to primary vertex fitting, and all attached tracks are also refit with this updated point.

Secondary (or displaced) vertices are discussed further in Chapter 4, where they form a key signature in the search for long-lived particles. More information and detailed studies on the performance of the secondary vertexing algorithm for various simulated exotic particles can be found in [52].

³The separation weighted by the covariance matrix.

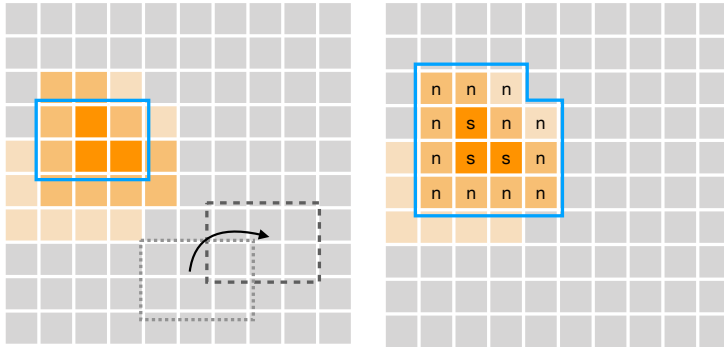


Figure 1.27: A cartoon showing the sliding window clustering method on the left, and the topological clustering method on the right. The solid line shows the cells that were clustered together in each case. For the sliding window, the fixed window is shown in dashed lines and an arrow indicates its motion. For the topo-clusters the seed cells above threshold are denoted by ‘s’, and any neighbor cells added are labelled ‘n’. Any cells not added were below threshold to be a secondary neighbor.

Calorimeter clustering

Just as tracks are reconstructed from hits in the ID and MS, calorimeter deposits are clustered into deposits of interest to combine with the other subdetector systems and identify particles. Clustering algorithms group calorimeter cells and calibrate their summed energy to account for losses due to the clustering or material constraints and depending on what particle they’re being used to identify [53]. ATLAS uses two types of clustering algorithms, see Figure 1.27:

- The **sliding windows** approach groups calorimeter cells into elements in 2 planes: the azimuthal, or transverse plane, where cells are arranged into towers of size $\Delta\phi_{\text{cluster}}$ and the longitudinal plane, where cells are grouped by $\Delta\eta_{\text{cluster}}$, the size of which are listed in Table 1.5. A ‘window’ of fixed size $(5 \times \Delta\eta) \times (5 \times \Delta\phi)$ is then moved across each cell in steps of size $\Delta\eta \times \Delta\phi$. A cluster of cells is formed if the sum of the transverse energy from the towers in the window is above 3 GeV or 15 GeV depending on whether seeding is being done only in the EM calorimeter or in both EM and hadronic. The energy deposits are used to weight the centroid of the cluster, much like with Pixel space-point centroid finding mentioned in the previous section. Clusters in each layer are then further clustered together, although this step is usually done in conjunction with particle identification which will be discussed later.
- The **topological cluster** approach instead clusters any cells that have a relatively high signal-to-noise ratio, resulting in clusters of various sizes, unlike in the sliding window method. Clusters are seeded by cells with a signal-to-noise ratio greater than the seed threshold listed in Table 1.5. Neighboring cells are then added to the cluster, if their threshold is greater than another signal-to-noise threshold, the neighbor threshold, and these cells can then have *their* neighbors provided those neighbors exceed the ‘additional neighbor threshold’.

These methods of calorimeter cluster seeding are very efficient, ranging from 95% for an electron with transverse energy of $E_T = 7$ GeV, to > 99% for electrons with $E_T > 15$ GeV [54]. Regions of interest around an EM calorimeter cluster are defined as having $\Delta R < 0.3$, with respect

Clustering method	Parameter	EM only	Combined
Sliding window	$ \eta_{\max} $	2.5	5.0
	# towers in η	200	100
	# towers in ϕ	256	64
	Window size ($\Delta\eta \times \Delta\phi$)	5×5	5×5
	$\Delta\eta_{\text{cluster}}, \Delta\phi_{\text{cluster}}$	0.025	0.1
Topological	Seed threshold	6 S/N	4 S/N
	Neighbor threshold	3 S/N	Cell added to cluster regardless of energy
	Additional neighbor threshold	3 S/N	2 S/N

Table 1.5: Key parameters for calorimeter cell clustering [53].

to the seed centroid. Finally the clusters themselves are calibrated at two different scales: either at the cell level where the cluster’s energy is the sum of its constituents’ (known as EM calibration) or with some additional weighting applied (local cell weighting or LCW), which accounts for the different types of energy loss in the EM and hadronic calorimeters. These calorimeter clusters, alongside tracks, are then used to identify particles in the detector which will be briefly outlined for the most common Standard Model particles in the next section.

Electrons & photons

Electrons and photons leave very similar signatures in the ATLAS detector as both interact with materials with a high atomic number and small radiation length, which means they both produce showers in the EM calorimeters. In addition, electrons will ionize the ID and their path can be reconstructed as tracks. Electron reconstruction starts in the track reconstruction process. By default, standard tracking assumes the particle being reconstructed is a pion, however if a track seed cannot be successfully extended to a track and is within the region of interest of an EM calorimeter cluster, track extrapolation is attempted again but this time allowing for the greater energy loss that an electron would suffer from material interactions. During the track fitting process too, the track can be fit to allow for larger energy loss assuming it’s actually an electron if it fails the fit as a pion. Reconstructed tracks are matched to calorimeter clusters using ΔR values on the order of the cell resolution, and if they match then the electron is tagged [54].

In the case of photons, a photon may make it to the calorimeter before showering or may undergo conversion into an electron-positron pair in the ID. As with electron reconstruction, a calorimeter cluster seeds a region of interest in the ID. However, in the case of the photon there can be no track, two tracks, or a single track for the cases of no pair-production, pair-production where both tracks are reconstructed, and pair production where only one of the tracks were reconstructed. In addition, there is a need to distinguish between prompt photons, photons from jets, and photons from π_0 decays. This is largely done by using information about the cluster shape and size in the EM calorimeter as well as how much the cascade leaks into the hadronic calorimeter [55]. In case of pair-production, if both tracks are reconstructed their invariant mass must be consistent with a massless particle and the TRT can be used to confirm the electron ID.

For both photons and electrons, a sliding window clustering approach is used as the showers tend to be more collimated than for jets or hadrons.

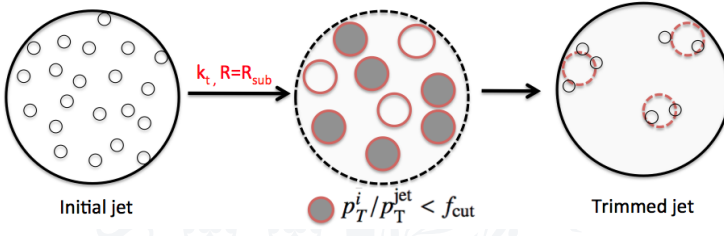


Figure 1.28: A cartoon showing the procedure for jet trimming [57].

Jets

Due to color confinement, quarks and gluons cannot exist in isolation and promptly hadronize. As such, the majority of inelastic proton collisions in ATLAS result in collimated cascades of particles which can then be clustered together as jets. For the rest of this thesis, jets will refer to those seeded from topological clusters in the EM and/or hadronic calorimeters, although they can also be seeded from tracks or combinations of tracks and clusters. Jets are grouped into two categories that are reconstructed and calibrated differently, distinguished by having either a large or small angular separation, ΔR . Small- R jets are reconstructed with $\Delta R = 0.4$ and are typically used to find hadronized quarks or gluons. Large- R jets are typically used to reconstruct the decay products of massive hadrons and have $\Delta R = 1.0$. Small- R jets are seeded with EM calibrated topo-clusters, whilst large- R jets are seeded with LCW topo-clusters.

In the ATLAS experiment, jets are reconstructed using the anti- k_T algorithm [56] which is another clustering algorithm that proceeds according to a distance metric, d_{ij} :

$$d_{ij} = \min \left[k_{Ti}^{2p}, k_{Tj}^{2p} \right] \frac{\Delta_{ij}^2}{\Delta R^2} \quad (1.22)$$

where i and j are the indices of the two entities being considered, k_T is the p_T of the topo-cluster under consideration, the exponent p is set to -1 , $\Delta_{ij} = \Delta y^2 + \Delta \phi^2$, where y is the rapidity and ϕ is the azimuthal angle, and ΔR is the angular separation as before. The algorithm proceeds as follows, first the distance metric, d_{ij} is calculated for two entities in a list, which may be calibrated topo-clusters, jets from a previous iteration, or a combination of the two. Then, if $k_{Ti}^{2p} < d_{ij}$ then the entity i is considered a standalone jet and is removed from the list of entities. Else, the two entities are combined into a new jet. This continues until there are no entities left in the list. The advantage of this algorithm is that it clusters the hardest, or largest p_T , entities first and as such is less prone to seeding from noisy elements.

Whilst jets formed from topo-clusters already have pile-up and noise rejection inherent in their formation, large- R jets are more prone to including them simply on account of their size. As such, trimming is employed to remove large- R jets' softer constituents, see Figure 1.28. Jet trimming also uses equation 1.22, but sets $p = +1$, which changes the clustering to start with the softest p_T elements first instead, and also has a much smaller reach, setting $R = 0.2$. The algorithm is applied to a jet and results in dividing it into a group of sub-jets that form it. Then, for each sub-jet, if its p_T is less than 5% of the p_T of the overall jet, it is removed from the jet. This method gives stable pile-up rejection in Run 2 and in simulation up to an average of 200 $\langle \mu \rangle$. For large- R in $\langle \mu \rangle \sim 50$, trimming may remove up to 10% of the jet mass whilst for small- R jets between 30–50% of the mass may be trimmed away and so proves too aggressive to mitigate pile-up. As such, for small- R jets the average pile-up energy is instead calculated per event and its relative contribution assuming homogeneity in space, is subtracted from the jet's energy instead.

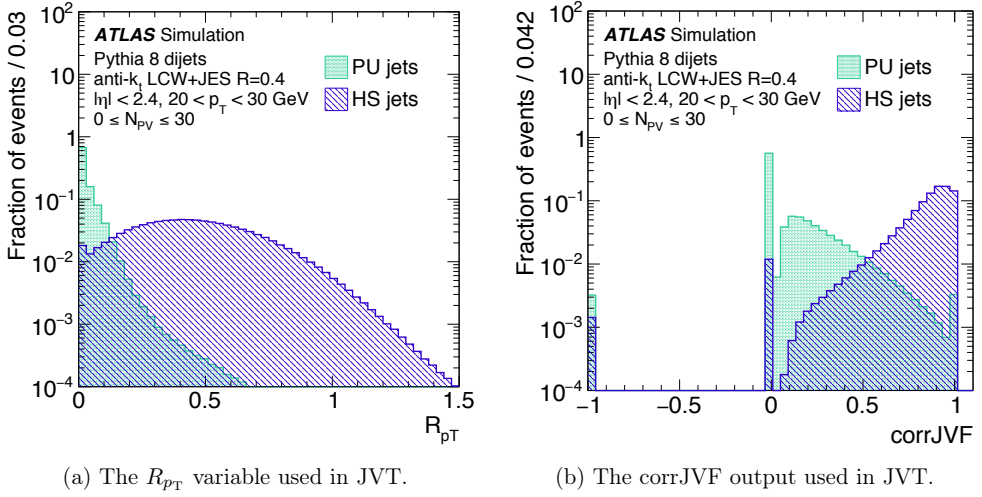


Figure 1.29: The output of R_{p_T} and corrJVF for pile-up (PU) and hard-scatter (HS) jets in simulation, see text for details [58].

A final technique used for pile-up suppression is the Jet Vertex Tagger (JVT) [58]. JVT is a two-dimensional likelihood with corrJVF in one dimension and R_{p_T} in the other. Although the interplay of these two variables in the likelihood is complex the variables themselves can be understood intuitively. First, the quantity sum-track- p_T ($\Sigma^{\text{trk}} p_T$) of a jet can be defined:

$$\Sigma^{\text{trk}} p_T(\text{PV}_x) = \sum_i p_{T,i}^{\text{track}}(\text{PV}_x) \quad (1.23)$$

where i is the index of the sum over the p_T of tracks associated both to the jet in question and some primary vertex, PV_x , where usually $x = 0$ denotes the hard-scatter vertex. Tracks are associated to jets using proximity in η and ϕ , using ΔR . The exact cut on ΔR is a function of track p_T , e.g. low p_T tracks are more likely to be associated to less collimated particle sprays and hence a wider jet cone. The JVT variable R_{p_T} of a jet, then, is the $\Sigma^{\text{trk}} p_T$ of the jet divided by the calibrated jet p_T , where various jet calibrations will be discussed shortly:

$$R_{p_T} = \frac{\Sigma^{\text{trk}} p_T}{p_T^{\text{jet}}} \quad (1.24)$$

This variable is plotted for simulated hard scatter and pile-up jets in Figure 1.29a. R_{p_T} peaks at zero for pile-up jets as they are rarely associated to the hard scatter vertex. However, for the jets that are from the HS, the R_{p_T} value is equivalent to the charge fraction coming from the hard scatter vertex and has a wide distribution of values. The other JVT variable, corrJVF , has the same numerator as R_{p_T} but a more complicated denominator, which is easier to understand after defining a few additional terms. One of which, p_T^{PU} , is a variable similar to $\Sigma^{\text{trk}} p_T$:

$$p_T^{\text{PU}} = \sum_{n \geq 1} \sum_i p_{T,i}^{\text{track}}(\text{PV}_n) \quad (1.25)$$

which sums the $\Sigma^{\text{trk}} p_T$ over all pile-up vertices. The variable n_{p_T} , which is the number of tracks associated with all pile-up vertices in the event, is used to define a linear correction A :

$$A = \frac{1}{k \cdot n_{p_T}} \quad (1.26)$$

where k is some fitted parameter, usually 0.01. Then corrJVF is defined as:

$$\text{corrJVF} = \frac{\Sigma^{\text{trk}} p_T}{\Sigma^{\text{trk}} p_T + A \cdot p_T^{\text{PU}}} \quad (1.27)$$

The distribution for this variable for both pile-up and hard scatter jets in simulation can be seen in Figure 1.29b. CorrJVF is defined in the range $[0, 1]$ but a jet is assigned the value -1 if it has no tracks associated to it at all. With R_{p_T} and corrJVF combined in the JVT, hard-scatter jet-tagging efficiencies of over 90% can be obtained for a fake-rate⁴ of $\sim 1\%$ and are efficient over a wide range of $\langle\mu\rangle$.

Following pile-up mitigation the jets are calibrated to account for energy lost in the detector that was not recorded. This loss is due to insensitive material in the sampling calorimeters, gaps in detector coverage, and the activation energy of the cascades. This correction is known as the Jet Energy Scale (JES) [59]. JES correction calculation is first done in simulation, comparing reconstructed topo-cluster jets to jets formed from simulated particle track four-vectors. JES is then calibrated to the data, to account for differences in simulation and real physical processes, using two methods. The first method takes the same topo-cluster-to-track ratio from the initial calculation and compares it to the same quantity from data. The second method compares di-jet events in different parts of the detector and uses their differences to correct JES. Large- R jet invariant mass is also calibrated via the Jet Mass Scale (JMS) which is refined by adding the invariant mass of tracks associated to the jet. The JES of small- R tracks can be further improved by the Global Sequential Calibration (GSC) that accounts for the detector's non-uniform response in η , shower development, and punch-through - in which not all of the shower is contained to the calorimeters.

With jets reconstructed and calibrated, additional particle identification can be performed, as is described in the following section.

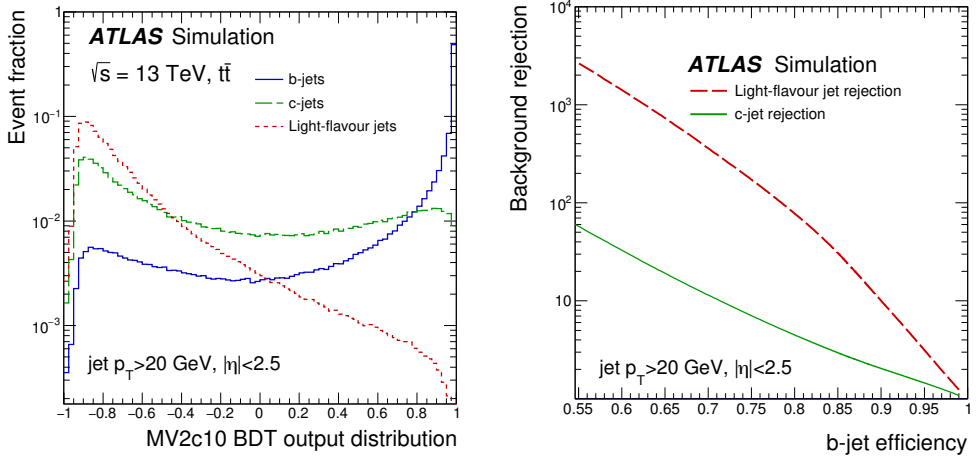
B-tagging

Bottom quarks fragment into excited mesons, which promptly decay strongly or electromagnetically to stable b-hadrons, which decay to D-mesons via the weak force. The strength of the weak coupling constant for B-hadron decays suppresses them and displaces them from the primary vertex on the order of mm. The decays are reconstructed using a similar secondary vertex algorithm to the one described in 1.3.1 but with tighter IP cuts ($|d0| < 3.5 \text{ mm}|$) and silicon hit requirements (at least one hit in the Pixel detector, no more than one hit shared with another track). Vertices are seeded from two-track pairings and their invariant mass is used to rule out other long-lived decays such as K-short mesons. Because these secondary vertices are only being built from standard tracks there is no need for the complicated ambiguity solving employed in the secondary vertexing method described earlier. Instead, tracks are added by proximity to the seed vertex unless they do not improve the χ^2 fit result across the vertex.

B-tagging is the identification of a b-hadron and its decay products using tracks and jets. B-tagging in the ATLAS experiment is done with several multivariate techniques including likelihood discrimination, boosted decision trees, and neural nets which make use of various reconstructed properties of b-hadron candidates [60]:

- The significance of the tracks' impact parameters, which gives more weight to tracks with smaller measurement errors is fed into a likelihood ratio for both transverse and longitudinal IP's.

⁴In this case incorrectly tagging a pile-up jet as a hard-scatter jet.



(a) The output of the MV2c10 Boosted Decision Tree for b-jets, c-jets, and light-flavor jets.

(b) The MV2c10 BDT background rejection vs efficiency for b-jet backgrounds.

Figure 1.30: Rejection and discrimination of the B-tagging BDT [60].

- Various properties of the secondary vertex candidate are then fed into a 3D likelihood ratio known as SV1: the invariant mass of the tracks associated to the vertex, the ratio of $\sum_{\text{vtx trks}} \text{energy}^{\text{trk}} / \sum_{\text{jet trks}} \text{energy}^{\text{trk}}$, and the number of two-track vertices in the event.
- Finally a Kalman filter is used to extend the b-hadron candidate decay vertex to a D-hadron candidate vertex, and the flight path for the b-hadron decay is recovered.

The outputs of these three algorithms, along with the jet η and p_T , are passed into a boosted decision tree known as MV2c10 which allows B-hadrons, D-hadrons, and light flavor jets to be tagged, see Figure 1.30.

Muons

Muons will ionise the ID and the MS and may lose energy in the EM and hadronic calorimeters, although at a much lower rate than electrons. As such their reconstruction can proceed starting from either tracker, see Figure 1.31 for how muons move in the ATLAS magnet system. Track reconstruction in the MS is a little different than in the ID because of the reduced granularity, number of layers, and occupancy of the detector. First a pattern-finding algorithm called a Hough transform is used to search for track candidates in the MS and nearby trigger chambers. This is done by translating the 3D position of detector elements registering hits into a phase-space where those more likely to be tracks are clustered together. An example of this for a pair of two-dimensional, intersecting tracks is shown in Figure 1.32. Within the MDT, hits are fit into straight-line segments. Segments in the CSC are built from a combinatorial search over binned η - ϕ hits. These segments are equivalent to the seeds used in ID track reconstruction and are extrapolated into full tracks, and fitted, also using a similar method to ID tracks. Segments from the central detector layers (recalling that each subdetector in the MS has three layers) are used to seed the tracks and then extensions are made to the outer and inner layers. Ambiguity resolution, using a point-based system like in the ID that is based around number of hits, is used to remove any duplicate/shared tracks.

The MS tracks can then be combined with ID tracks in four ways [62]:

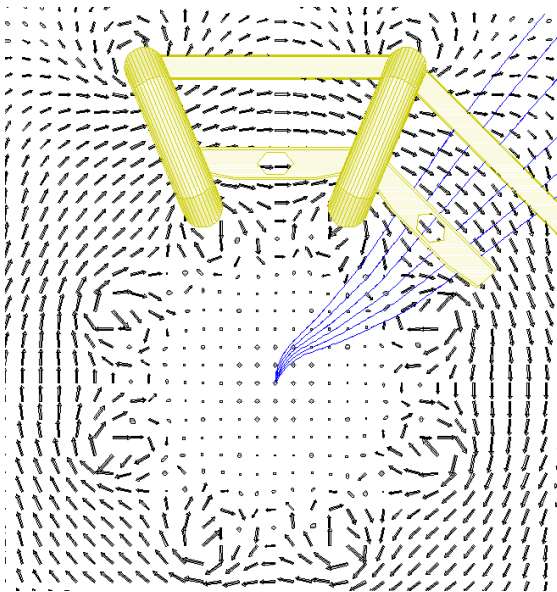
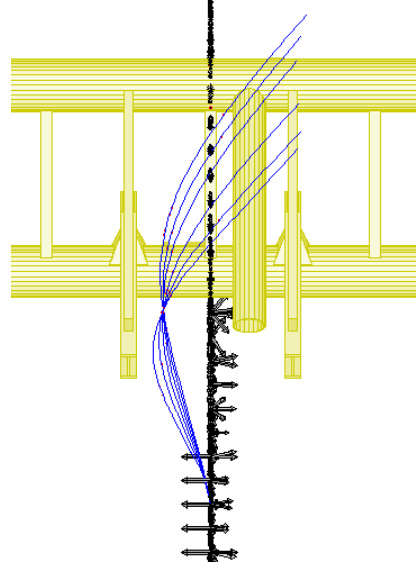
(a) Trajectory of a high- p_T muon in the r - ϕ plane.(b) Trajectory of a high- p_T muon in the r - z plane.

Figure 1.31: Muons moving through the solenoidal and toroidal magnetic fields of the ATLAS detector [15].

- A **Combined Muon** uses a track reconstructed in the MS and extrapolates it inwards to see if it matches any ID tracks. A combined track is then formed by refitting using the hits from both detectors, and MS hits can be added or removed to improve the overall fit. Combined muons can also be formed by taking an ID track and extrapolating to the MS but this is less common due to the sheer multiplicity of ID tracks.
- If an ID track is extrapolated to the MS and only one local track segment in the MDT or CSC is found to match the track it is classed as a **Segment Tagged** muon. A muon may only form a segment in one layer of the MS subdetectors if it has low p_T or if the acceptance in that region of the MS is poor.
- If an ID track is matched to a calorimeter cluster compatible with the amount of energy deposited by a minimally ionizing particle then it is classed as a **Calorimeter Tagged** muon. Whilst this method is prone to fakes it manages to recover acceptance in the very central region of the detector ($|\eta| < 0.1$) where the MS has coverage gaps for detector access.
- For muons that are within the acceptance of the MS, but not the ID, the muon track can be **Extrapolated** from the MS to the beam line to see if it's compatible with the interaction point. This is typically done for muons in the range $2.5 < |\eta| < 2.7$.

Muons are reconstructed with the following priority: combined, segment tagged, and calorimeter tagged.

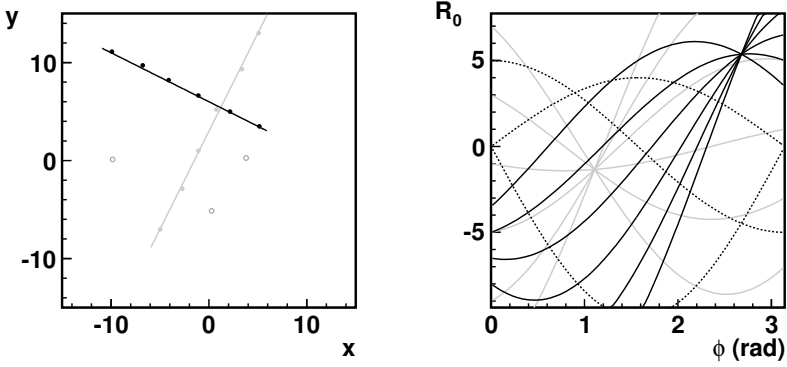


Figure 1.32: Two tracks in the x-y plane (left) and their corresponding points in Hough space (right). In Hough space each line represents a point in real space and the intersection of the lines indicate that the points likely belong to the same track [61].

Missing energy

With various collision fragments reconstructed into particle and composite objects, the event can be analysed for signs of interesting physical processes. One such quantity that combines these objects is the Missing Transverse Energy (MET). MET is calculated by taking the magnitude of the vector sum of momentum in the transverse plane from reconstructed objects (e.g. electrons, photons, muons, and jets). Assuming that the incident partons had a negligible amount of momentum in the transverse plane, this quantity should be zero within measurement uncertainties unless a particle has escaped the experiment without detection. MET can be used to infer the presence of undetectable SM particles such as neutrinos or massive stable BSM particles, such as neutralinos.

The final part of this chapter will briefly outline select upgrades to the ATLAS detector and associated infrastructure for the high-luminosity LHC.

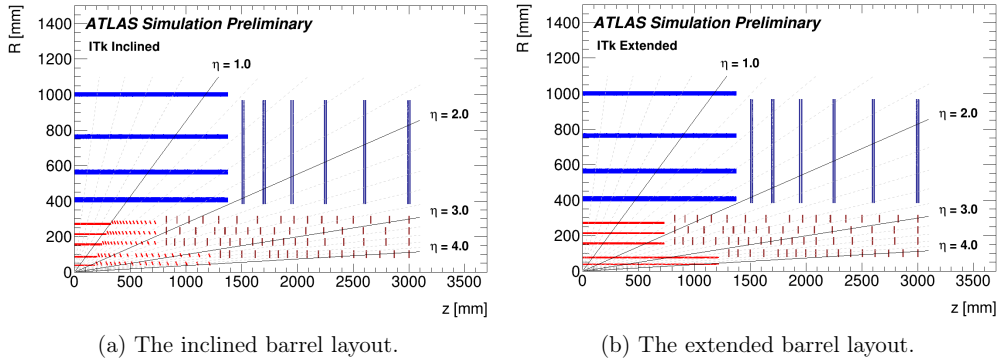


Figure 1.33: The two ITk Pixel layouts that were considered for HL-LHC [64]. The layers shown below $R = 300$ (in blue) are Pixel layers, those above (in red) are Strip layers.

1.4 ATLAS upgrades for HL-LHC

In 2026 the LHC will enter a High Luminosity phase (HL-LHC) increasing the average instantaneous luminosity by a factor of five and delivering 4000 fb^{-1} within twelve years. This will impose significant technical challenges on all aspects of the ATLAS detector but particularly the Inner Detector, trigger, and data acquisition systems. HL-LHC is projected to deliver collisions with up to 200 interactions per bunch crossing, saturating the TRT and parts of the current Pixel detector’s hit occupancy, rendering them inoperable. As such the entire Inner Detector will be replaced by an all-silicon tracker, the new Inner TraKer (ITk). The section below briefly outlines aspects of the ITk Pixel project relevant to this thesis, the entire technical design report is available here [63].

ITk layout

ITk will have five barrel layers of pixel detectors surrounded by four layers of strip detectors. Both the Pixel and Strip detector barrels will be capped by layers of ring structures. The new detector is required to extend tracking coverage from $|\eta| = 2.5$, in the current ID, to $|\eta| = 4$. In 2014, detailed studies of the ITk layout were performed. At the time, two layouts were studied by the collaboration: an ‘extended’ layout, which increased the reach of the two innermost Pixel barrel layers out to $|z| \approx 1.2 \text{ m}$, and the three outermost to $|z| \approx 0.6 \text{ m}$, and an ‘inclined’ layout, which instead extends the Pixel barrel layers out to $\eta \approx 1$ and fills the remaining space with Pixel modules inclined at an angle to the barrel, in the range $1 \leq |\eta| \leq 4$, see Figure 1.33.

The extended layout was estimated to provide an additional handle for track reconstruction, utilising the near-parallel placement of modules with respect to high- η tracks to measure long clusters of hits, which in turn could be used to establish the incident angle of tracks crossing the layer. **Paper I** of this thesis describes a measurement, made using Pixel modules from the ATLAS IBL detector, to investigate this concept. The pixel assemblies were placed in an electron testbeam at precise angles between 2° and 15° , and the size of pixel clusters was measured and found to be distinguishable at small angles.

Ultimately, the collaboration selected the ‘inclined’ ITk layout for the final design, which is discussed further in the technical design report.

Condition	ITk Pixel Inner, flat modules	IBL modules	Difference
Lifetime integrated luminosity*	2000 fb ⁻¹	300 fb ⁻¹	×7
Lifetime fluence*	1.3×10^{16} neq/cm ²	5×10^{15} neq/cm ²	×2.6
Total Ionising Dose*	7 MGy	2.5 MGy	×2.8
Avg. hits/chip/bunch-crossing	223 @ $\langle\mu\rangle = 200$	27 @ $\langle\mu\rangle = 75$	×8
Average hit density	0.6 hits per mm ²	0.08 hits per mm ²	×8
Current consumption	< 8 μ A/pixel	20 μ A/pixel	×0.4
Minimum stable threshold	600 e	1500 e	×0.4

Table 1.6: A subset of the expected conditions and specifications for the ITk Pixel readout chip in the flat portion of the innermost layer, as compared to the readout chip in the innermost layer of the current ATLAS detector [65] [63]. (*) Denotes values assuming the innermost layer is replaced after an integrated luminosity of 2000 fb⁻¹ is delivered by HL-LHC.

1.4.1 ITk Pixel modules

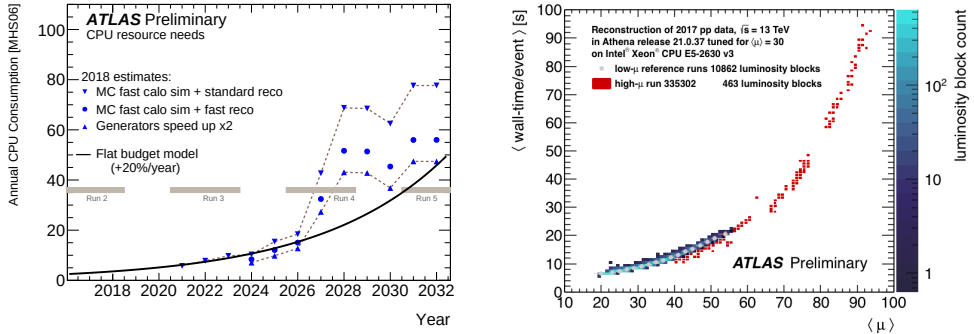
Along with optimising the ITk layout, the Pixel detector modules need to be upgraded to cope with the increased particle fluence, radiation damage, and data transmission requirements, which surpass the conditions in any previous run in the ATLAS detector, see table 1.6.

ITk pixel detectors have the same basic structure as those in the ATLAS ID: a pixelated silicon sensor connected to a mixed-signal, front-end readout chip. RD53, an inter-experiment collaboration between the ATLAS and CMS experiments, was formed with the task of designing tools and strategies for front-end readout chips to be used with hybrid pixel detectors in HL-LHC conditions [65]. In late 2015, RD53 produced a test-chip in 65 nm CMOS: FE65-p2. FE65-p2 features $50 \times 50 \mu\text{m}^2$ pixels, to reduce the per-pixel hit rate in the HL-LHC environment. This reduction in pixel area, when compared with the smallest pixels in the current ID, is made possible by the use of 65 nm technology, which provides the necessary logic density.

FE65-p2 is also the first instance of a readout chip employing analog logic blocks surrounded by a sea of synthesised digital logic. Previous front-end readout chips tiled blocks of analog and digital logic over the entire chip in a step-and-repeat procedure, which is a more labour intensive design process.

The ITk Pixel front-end chip is expected to receive a lifetime total ionising dose of 7 MGy, which is approximately three times the expected lifetime dose of the IBL. To test the radiation-hardness of various designs, FE65-p2 was equipped with eight columns of different analogue front-end flavours and then irradiated in a proton beam to see how each flavour’s performance was affected. These tests and their results are detailed in **Paper II**. The tests on FE65-p2 resulted in three analogue front-ends being selected in a full-size demonstrator chip, RD53A. In 2017 RD53A, which has been studied over the course of two years to select a final analogue front-end variant, amongst other things, for the ITk readout chip.

Pixel modules in the innermost layer of the ITk will experience a particle fluence around 2.6 times higher than the IBL over the course of their lifetime. Non-ionising energy loss in the sensor elements will detrimentally affect its electrical properties, reducing its efficiency as a detector. The precise mechanism by which high particle fluence affects silicon sensors, strategies for coping with that damage, and simulations to predict its affect on sensor properties is detailed in Chapter 2. The validation and usage of these simulations for the current Pixel detector are discussed in **Paper III**.



(a) The estimated CPU resources needed for the ATLAS computing framework during HL-LHC, for both data and simulation processing. The blue points show the improvements possible in three different scenarios: (1) assuming a fast-calorimeter simulation is used for 75% of simulations; (2) a fast event reconstruction on simulation (3) event generation is halved. The solid line shows the amount of resources expected to be available if a flat funding scenario is assumed, which implies an increase of 20% per year, based on the current technology trends.

(b) The dependency of reconstruction wall time per event on the average number of interactions per bunch crossing, shown for a couple of luminosity blocks of data from 13 TeV proton-proton collisions recorded by ATLAS in 2017. The blue scatter plot shows the reconstruction time per event for running conditions typical to Run 2, whilst the red boxes show the performance during a dedicated high- μ run. The running conditions and the way the data was processed in each case was not identical, but the overall trend should be representative of performance in HL-LHC.

Figure 1.34: Projected computing performance plots for HL-LHC [66].

1.4.2 Track reconstruction in ITk

The HL-LHC will deliver between five and ten times the luminosity that was delivered by the LHC in Run 2, increasing the amount of data recorded per event and hence the amount of data processed per second. Accounting for a baseline growth of 20% pa in Grid resources, projections indicate that they would need to increase by an additional factor of five by 2025 to maintain current performance. Since it is not feasible to grow the Grid by that amount in time, there is a shortfall in both CPU power and storage for the HL-LHC, see Figure 1.34a. One of the most time-intensive steps of data processing in HL-LHC will be track reconstruction [67], see Figure 1.34b.

The ATLAS collaboration is attacking this problem in several ways. Firstly by parallelizing its existing codebase and improving the overall efficiency of the software workflow, and secondly to supplement the WLCG further with high performance computing, which is already a part of the ATLAS computing framework. This second approach requires understanding both how to seamlessly parse data to and from the HPC and also how to select and optimise computing on the HPC itself. For example, implementing a common tracking algorithm, the Kalman filter, on an inherently parallel architecture such as FPGAs, Intel's Xeon-Phi architectures, and GPU's [68]. Future technology, that may become commodity computing in a decade's time, is also being explored. Chapter 3 and **Paper IV**, describe a study to implement the Kalman filter on an inherently parallel architecture: IBM's neuromorphic chip, TrueNorth.

Chapter 2

Radiation damage modeling in Pixel detector sensors

Since the 1950's, silicon has dominated the semiconductor industry for its abundance, ease of use, and its physical properties: ideally suited for creating layered junctions of varying conductivity, the building blocks of transistors.

The physical properties of silicon that make it desirable as a component in industry are not dissimilar to those that make it an excellent detector in particle physics. This chapter will first explore the design of silicon pixel detectors in section 2.1. This section relies on a baseline understanding of the properties of semiconductors and pn-junctions and whilst a complete description is beyond the scope of this thesis, Appendix A.2 provides a concise overview of these topics for those unfamiliar with the properties of pn-junctions. Section 2.2 describes energy deposition models for silicon detectors, including mechanisms that create irreversible damage. Section 2.3 describes the way in which these detailed mechanisms are simulated in the ATLAS software framework, Athena. This section serves both as context for the changes described later in the chapter and also as a detailed reference, as no documentation of this part of the software yet exists. Finally, section 2.4 describes how the simulated effects of radiation damage are incorporated into Athena. The verification of the models and techniques used in this simulation are detailed in **Paper III**.

2.1 Silicon detectors

The reverse-biased silicon diode forms an excellent basis for a solid-state ionisation chamber. The potential difference across the junction depletes charge carriers from the diode. This means that a small amount of injected charge will produce a measurable current that would otherwise be lost in thermal noise from the undepleted motion of thermally liberated charge carriers. A charged particle traversing the diode's bulk will impart energy to the lattice, liberating charge carriers which move in the electric field and produce a current. The current is then integrated by a charge sensitive amplifier and recorded. In this section, specific modifications to the basic diode, described in the previous section, to build a particle detector will be discussed.

Junction manufacturing

Most of the behaviour of pn-junctions is defined assuming that the physical dimensions of each dopant used in the pn-junction are the same. However, the silicon detectors used in modern particle physics consist of a bulk of one dopant type and a thin implant of the other, which is related to how the devices are manufactured. In the years following the invention of the junction transistor, which, like the diode, is based around the properties of the pn-junction,

methods to quickly produce devices with controlled doping concentrations were explored [69]:

- **Grown junction method** involves extracting semiconductor from a melt in a single crystal ingot. The silicon melt is infused with a specific carrier concentration and halfway through the dopant type is switched by adding a new material to the melt. Each ingot must then be cut around the junction. This technique was discontinued as cutting was not scalable for mass production.
- **Alloy junction method** starts with a singly-doped crystal which is sliced to a desired thickness. A pellet of the opposite dopant type is placed on the crystal and the materials are heated until a molten boundary is formed at the junction. When the crystal cools it recrystallises with the doped material in the pellet infused in the bulk as an alloy. However, the amount of recrystallisation that occurs is hard to control and because the electrical properties of pn-junctions are highly dependent on the concentration of the dopant, this lack of control meant the method was unreliable and was discontinued.
- **Planar technology** uses solid-state diffusion of a dopant into the surface of the bulk. The depth to which the dopant penetrates is easily controlled and sections of the bulk can be masked off using a SiO_2 mask.

Modern commercial semiconductor processes consist of a bulk-doped material with a planar layer of the opposite dopant material diffused on top. This forms the basis for a silicon sensor.

Ohmic contacts

To bias a semiconductor diode, and create a depleted space-charge region, metal electrodes are attached to the front and backside of the junction. When a metal and a semiconductor are brought into contact the relative difference between the work function¹, ϕ_{metal} , of the metal and electron affinity², $\chi_{\text{semiconductor}}$, of the semiconductor create a band-gap characterised by the barrier potential, ϕ_{barrier} :

$$e\phi_{\text{barrier}} = e(\phi_{\text{metal}} - \chi_{\text{semiconductor}})$$

For good conductance between the semiconductor and the metal, the metal should be chosen such that the barrier potential is as small as possible. This means finding a metal work function that closely matches the semiconductor's electron affinity. If the barrier potential is large compared to the semiconductor's conducting band, a Schottky rectifying junction will be formed where current will only flow by applying a forward bias between the metal contact and the semiconductor.

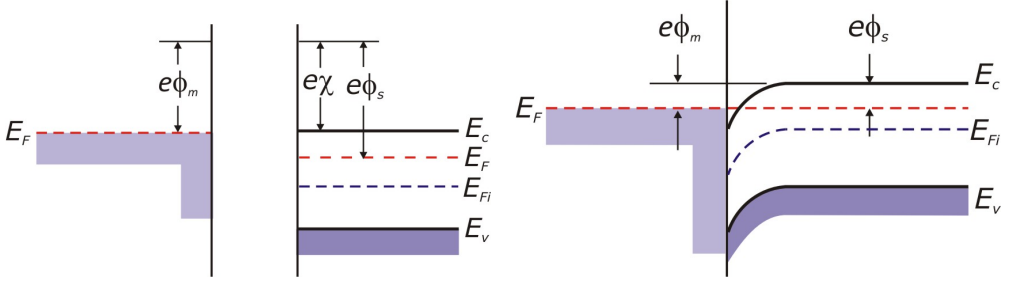
Ideally, for an n-type bulk, the metal contact must have a work function, $e\phi_{\text{metal}}$, that is less than or equal to the electron affinity, $e\chi_{\text{semiconductor}}$ of the semiconductor giving barrier energy that is less than or equal to zero, in other words an Ohmic³ contact is formed, see Figure 2.1. This means that electrons can flow freely from the semiconductor into the electrode and vice versa, i.e. the metal-semiconductor junction will conduct under forward or reverse bias.

Aluminium is a commonly used metal for Ohmic contacts as it has a relatively low work function of 4.08 eV when compared to silicon's electron affinity, 4.05 eV. Similarly, for a p-type electrode, the metallic work function must exceed electron affinity and band gap of the

¹The work function is the minimum energy required to remove an electron from a material and into the vacuum.

²The electron affinity of a semiconductor is the energy required to release an electron from the conduction band into the vacuum.

³An Ohmic contact is a conductive contact with a linear voltage-current relationship.



(a) The energy bands in a metal (left) and an n-type semiconductor (right), compared side-by-side. The work function of the metal is less than or equal to the electron affinity of the semiconductor.

(b) An Ohmic junction formed by a metal and n-type semiconductor. The semiconductor's conductance lower bound drops towards the junction, allowing charge carriers to easily flow across the boundary.

Figure 2.1: The formation of an Ohmic junction from a metal and an n-type semiconductor. In the diagrams E_c is the bottom of the conduction band, E_F is the Fermi level of the n-type material. E_{Fi} is the Fermi level of the intrinsic semiconductor, E_v is the top of the valence band, ϕ_m , ϕ_s are the work functions of the metal and semiconductor, and χ is the electron affinity of the semiconductor. The purple blocks indicate continuous bands [70].

semiconductor. Thus, to increase the Fermi energy and produce a good Ohmic contact with aluminium, a high dopant concentration exceeding 10^{19} cm^{-3} is used at each side of the sensor [71]. Since this doping concentration is orders of magnitude greater than the bulk, it is denoted with a +, e.g. n^+ , p^+ .

Pixel isolation strategy

Pixels consist of highly doped implants in the bulk, segmented in x and y. By design, the pixels need to be isolated from each other, electrically, to provide spatial resolution. This section explores how additional electrical isolation is provided for the n^+ pixel implants used in the Pixel detector sensors in the ATLAS experiment.

When charged particles pass through the sensor the SiO_2 layer on the detector surface is ionised and builds up a positive space-charge. The positive space-charge in the oxide layer causes a localised electric field between the pixel implants which causes a layer of electrons to accumulate at the oxide-bulk interface. If the implants conduct electrons, in other words if these are n^+ implants, then the electron accumulation layer can short pixels together. Therefore electrical isolation is needed.

This is done either by using p-stops, which are isolated p-doped islands in between n^+ implants, see Fig. 2.2a, or p-spray, in which the p-dopant is simply sprayed on top of the implants and coats the bulk in a diffuse layer, see Figure 2.2b. The doping concentration of the n-implants is high enough that an additional masking layer is not needed in the case of the p-spray. P-stop isolation was used to isolate pixels in early sensors, as the size and concentration of the dopant can be easily controlled with common lithographic techniques. However, the high potential of the p-stop/bulk boundary leads to lower breakdown voltage as the sensors are subjected to a higher particle fluence [73]. This means that the p-stop isolation strategy is less commonly used in silicon sensors that will be exposed to a high particle fluence over their lifetime.

When using p-spray isolation, however, care must be taken to pick the correct dopant concen-

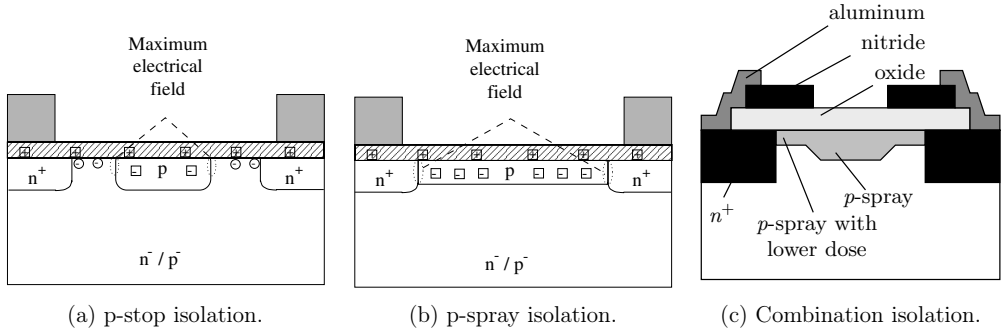


Figure 2.2: p-type isolation strategies for n^+ pixel implants. The squares in (a) and (b) show fixed space-charge regions, the circles are free charge carriers. Not shown are the additional metal and nitride layers. Figure (c) shows the combined p-stop, p-spray technique [72].

tration. If the p-spray dopant concentration is too low, the electric field from the oxide layer will still penetrate to the bulk and short the pixels together. If the dopant concentration is too high, the junction formed at the p-spray and implant boundary breaks down at lower voltages like the p-stop [74].

A moderated p-spray technique that has a thicker p-spray layer between pixels is used to combat the disadvantage of the p-spray. The thick, inter-pixel layer isolates the bulk from the oxide whilst the concentration adjacent to the n^+ implants is made just sufficient to isolate the bulk whilst not causing early breakdown in the device.

An additional advantage of using p-spray isolation instead of p-stop is that the application of a higher p-spray dose between the pixels is done by etching a deposited nitride layer later in the process and reapplying the p-spray. This does not count as an additional processing step, unlike the p-stop oxide etching, as nitride application is a commonly used process in semiconductor fabrication, for example in the passivation layer. The final metallisation and isolation strategy for an individual n^+ pixel is shown in Fig. 2.2c, where n^+ implants are read out by aluminium electrodes, a p-spray isolates the pixels from each other and shields the bulk from any positive-charge build-up in the oxide layer, and the nitride is used as a mask for the edge p-spray region.

Guard ring design and the bias grid

During operation, the pixel sensor is reverse-biased to deplete the bulk. The front-end chip which reads out the sensor, digitizes, and stores information about the pixel hit, is also biased. The front-end chip bias voltage is orders of magnitude smaller than the sensor, thus it makes sense to keep the side of the sensor that is in electrical contact with the front-end at a similar voltage to avoid sparking.

During fabrication the silicon sensors are diced from wafers into single units. This cutting process causes mechanical damage to the edge of the lattice and effectively causes it to become conductive. If the depletion region reaches this edge, the lattice damage acts like a generation centre and increases the leakage current. If a metal contact, at a different potential than the cutting edge is nearby, an electron accumulation layer can form and create a high field leading to early breakdown, as was the case with the p-stop implants. To prevent this, the potential difference between backside and frontside is stepped down in stages, using structures called guard-rings on the implant side where the junction is located.

Guard rings are based on the principle of punch-through. A floating implant, like that shown in the right side of Fig. 2.3a, is at V_{bi} , which is very close to the potential of the

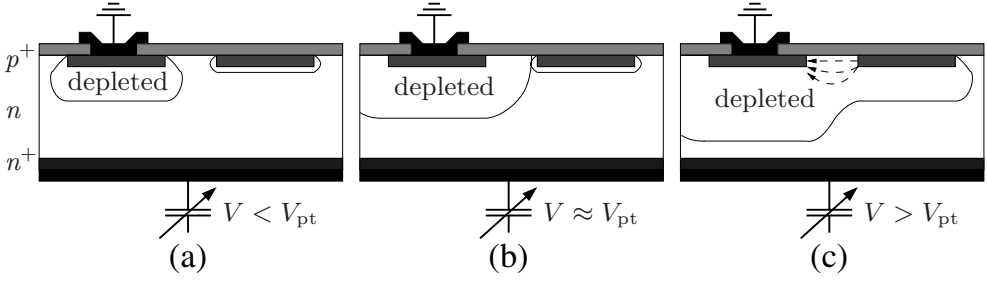


Figure 2.3: The effect of doping concentration and electric field on charge carrier mobility [72].

bulk. As the backside bias is increased there comes a point where the depletion region around the grounded electrode makes contact with the floating implant: this is known as the punch-through voltage, V_{pt} and the floating electrode will remain at this voltage even as the backside bias is increased, see Fig. 2.3b. In Fig. 2.3c, V_{pt} is sufficient that any holes in the floating implant flow to the grounded electrode. More floating implants can be daisy chained together to the edge of the sensor, with a potential drop between each subsequent ring. The spacing of the rings determines their voltage drop and is usually non-uniform over the surface of the sensor. For example, the innermost ATLAS Pixel layer has planar sensors with 13 guard rings. Ramping down from high voltage to ground the ring spacing is: 3 rings $\times 25 \mu\text{m}$, $50 \mu\text{m}$, 5 rings $\times 25 \mu\text{m}$, 2 rings $\times 50 \mu\text{m}$, and $100 \mu\text{m}$. A bias grid is present in modern pixel sensors to test the functionality of pixels before the expensive process of bump-bonding them to the front-end readout chip. The bias grid is essentially a net connecting every pixel in the sensor. The pixel implants are not connected to the bias grid directly, instead part of the pixel implant is removed and an isolated implant is introduced, called a bias dot, see Fig. 2.4. Before bump-bonding the bias dot can be coupled to the pixel implant via punch-through, like the guard rings. After bump-bonding the bias grid is either connected to ground via a capacitor or a resistor depending on its design such that noise is sent to ground. The design shown in Fig.

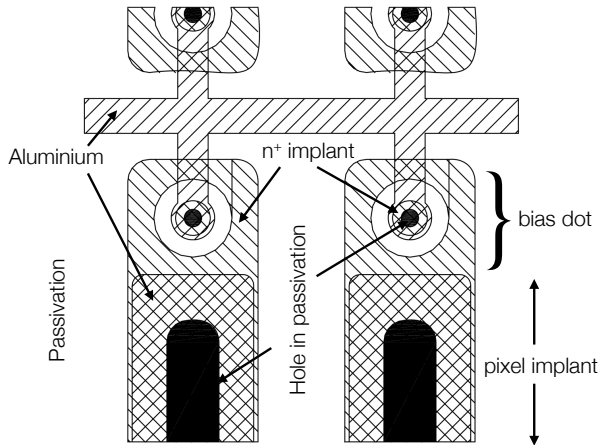


Figure 2.4: The bias grid connecting four pixels. Solid black sections are holes in the passivation, forward hatching is aluminium, and backwards hatching is the pixel implant. Adapted from [72].

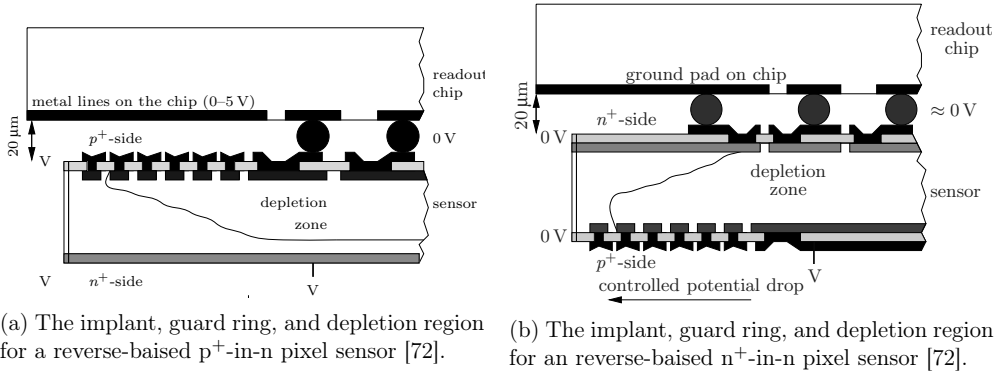


Figure 2.5: Bulk and implant designs

2.4 minimises the charge lost by this space in the pixel implant as any charge that falls into this area will drift towards the surrounding implant.

Pixel and bulk dopant pairings

In the previous parts of this section the basic components of a silicon Pixel sensor have been described including the bulk, ohmic implants, isolation to stop those implants shorting, guard rings to ramp down the bias voltage, and bias grid to test the sensors before bonding them to the front-end readout chip. Whilst these latter features are necessary to test and use the sensor, the majority of its properties - particularly those that are of interest to the topic of radiation damage on which this chapter focuses - stem from the choice of bulk and pixel implant dopant type. In the discussion that follows an implant, at the readout electrode, of type A in a bulk of type B will be referred to as an A-in-B design.

There are four possible combinations of implant and bulk: n^+ -in- n , n^+ -in- p , p^+ -in- p , and p^+ -in- n . Ideally, the depletion region will always grow *from* the pixel implant *towards* the backside of the detector so that even when under-depleted some charge can still be collected. This constraint would also enable single-sided processing. If the pn-junction grows from the back of the sensor, backside processing is needed to produce guard rings. However, if the guard rings are on the front-side they can be produced in the same steps as the pixel implants. Single-sided processing is not only more cost-effective but allows for thinner sensors. This is shown for p^+ -in- n and n^+ -in- n sensors in Fig. 2.5.

Two bulk and implant pairings with the pn-junction on the front-side are the n^+ -in- p and p^+ -in- n designs. The use of p^+ -in- n sensors in particle physics goes back to the first planar silicon sensors used in the NA11 and NA32 fixed target experiments at the SPS in the early 1980's, and extends through to the current strip sensors in the ATLAS SCT [75]. Additionally, because of the inherently isolated p^+ implants, not only was single-sided processing possible but additionally no p-stop isolation was needed. However, modern Pixel sensors do not use the p^+ -in- n design. The reasons for this stem from the effects of radiation damage on the sensor bulk that causes the net doping concentration in the n-type bulk to change to a p-type bulk, which will be described more completely in section 2.2.2. Bulk-type inversion causes the depletion region, in a p^+ -in- n sensor, to grow from the back of the sensor. This means the detector has to be fully depleted to avoid signal loss, which requires an increasingly high voltage as the radiation damage continues.

Using an n^+ -in- n sensor instead of p^+ -in- n , in environments which experience a large amount of radiation damage, has the advantage that after bulk-type inversion the pn-junction

will grow from the pixel electrode towards the backplane so the detector can be operated below full depletion. However, before type inversion, the pn-junction is located on the backside of the sensor, so the guard rings are also placed on the backside, see Fig. 2.5b. Although placing the guard rings on the backside introduces an additional processing step, having the pn-junction on the backside of the detector reduces the possibility that the high voltage sensor bias could arc to the grounded front-end which is only 10's of microns from the sensor and easily exceeds the critical value in air of 1 kVmm^{-1} [76]. Recent studies however show that introducing a passivation layer of Benzocyclobutane (BCB), which has a critical breakdown value of 300 kVmm^{-1} , allowed early prototypes of n^+ -in-p sensors to be biased to 1 kV for several hours [77]. With this in mind n^+ -in-p sensors combine the benefits of the depletion region growing from the frontside of the detector⁴ even after high radiation doses with the single-sided processing seen in p^+ -in-n detectors, see Fig. 2.5a. n^+ -in-p sensors will be used in future silicon detectors, including ITk.

2.2 Energy deposition in silicon detectors

The ATLAS Pixel and SCT detectors operate on the principle that high energy particles lose energy and ionise the detector volume. However the very particles that these sensors detect can damage the sensing medium. Over time, these damages accumulate and change the microscopic structure of the silicon detector, ultimately changing its macroscopic properties and affecting its performance.

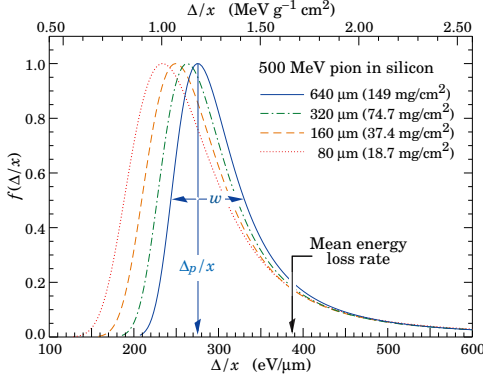
This section will first introduce ionising energy loss mechanisms before describing radiation damage and its effects. Although the latter half of the section is only concerned with radiation damage in silicon sensors, the front-end readout chip must also be carefully designed to cope with its effects. However, the modifications to make the front-end radiation ‘hard’ are outside the scope of this document. Discussions on the topic can be found, most recently for the HL-LHC front-end test-chip FE65-p2 and demonstrator RD53A [78] and for the current Pixel detector [79].

Understanding the cause of the changes in detector performance is essential to engineering solutions. It also provides a way to generate realistic simulation data for detectors in high-radiation environments. Simulation can then be compared to data collected in the experiment, which is an essential part of data analysis.

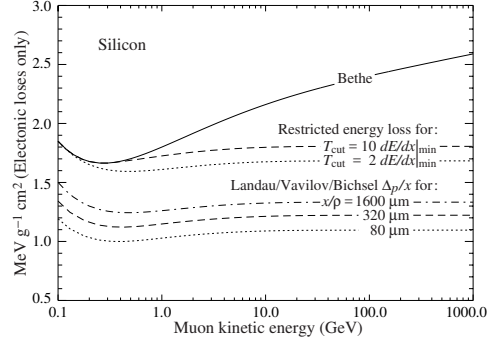
2.2.1 Ionising energy loss

High energy particles traveling through matter lose energy through single-particle interactions. In relatively thin detectors, fewer collisions will take place and there is a high relative variance in energy loss [8]. The Bethe formula, which gives the mean rate of energy loss, is ill suited to describe thin detectors. This is because the energy loss probability distribution function, or straggling function $f(\Delta; \beta\gamma, x)$, is highly skewed with a long tail from single-collision large energy transfers. As such the mean of the distribution is in the tail, and thus the modal value would be a more informed choice. In thin silicon detectors, the most probable energy loss over a thickness, x , is far smaller than the mean energy loss, see Fig. 2.6a. Only when x is around 15 cm thick does the Bethe formula become relevant. This thickness is an order of magnitude greater than the maximum path length a particle could traverse in a single silicon detector used in ATLAS, as is illustrated in Fig. 2.6b. A straggling function that does well to describe the energy loss p.d.f. is the Landau-Vavilov function with Bichsel corrections, see Figure 2.7.

⁴P-type silicon does not undergo space-charge sign inversion.

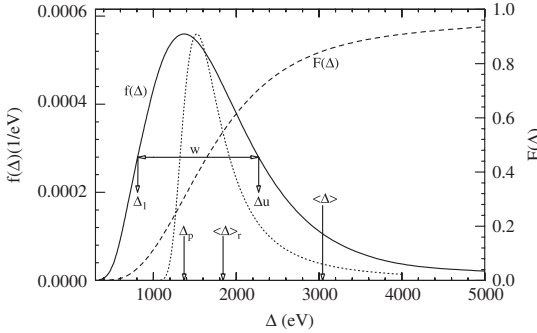


(a) Straggling functions for the energy loss distribution of a 500 MeV pion in various thicknesses of silicon, with each distributions normalised at its modal value. The full width half maximum, w , decreases with increasing thickness.

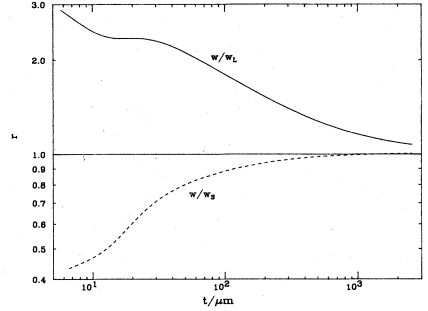


(b) Energy losses in silicon, for muons of various energies. The solid line is the mean energy loss from the Bethe formula. The first set of dashed and dotted lines show the mean loss if $T \leq T_{\text{cut}}$. The final three lines show the most probable energy loss for three different thicknesses.

Figure 2.6: Comparisons of the mean and most probable energy losses for minimum ionising particles in silicon [8].



(a) A comparison of the straggling function (solid line) and the Landau function (dotted line) for a particle of $\beta\gamma = 3.6$ in $10 \mu\text{m}$ of silicon. The dashed line shows the cumulative straggling function. The labels indicate the straggling function m.p.v, Δ_p , the mean energy loss from Bethe function, $\langle\Delta\rangle$, and the mean energy loss calculated from the subset of energies in the range of the plot, $\langle\Delta\rangle_r$, [80].



(b) The ratio, r , of the full width half maximum, w of the straggling functions in a silicon detector compared to w_L in the Landau-Vavilov distribution, for increasing thickness, t (solid line). The dashed line compares w and w_S , the FWHM of the Landau distribution with the Shulek correction. This correction was dropped following its divergence from experimental data [81].

Figure 2.7: Comparisons of the Landau function's modal value and FWHM with detector straggling functions.

Landau's formula derives from the Rutherford cross-section [80] and focuses on the problem of energy loss in finite material thickness with two conditions. The first being that the most probable energy loss would be large relative to the lowest orbital bound electrons, or *s*-shell electrons, and the second that the energy loss would be very small when compared to the maximum energy loss in a single collision, which was later removed by Valivov [82]. This first condition, however, is not applicable to most MIPs interacting with silicon detectors where, for example, in 250 μm of silicon the most probable energy loss is on the order of 10 keV, whereas the 1s orbital electrons have binding energy on the order of 1.8 keV [83]. This condition was, at least partially lifted, by Blunck and Leisegang who convolved the Landau-Vavilov distribution with a Gaussian distribution of width equal to the Rutherford cross-section [80]. Whilst the 'LanGau' is a good approximation to the underlying straggling function, differing by $\sim 5\%$ at 200 μm in silicon, it breaks down for thicknesses below this.

Bichsel added corrections to the Landau-Vavilov distribution by accounting for the density effect. As a particle's energy increases, its electric field changes shape, increasing the contribution to the energy of distant collisions [8]. Additional corrections, that added resonances related to the electron orbitals in atomic structure, were added which improved upon similar efforts from Blunck, Leisegangs, and Shulek [84]. Bichsel, in his 2006 review of straggling functions for particle tracking and ID [80], makes the case that producing analytical straggling functions from historical first principles is computationally expensive and instead they should be calculated ahead of time with a Monte Carlo simulation. This will be explored further in section 2.3.4.

The straggling functions provide a probability density function (p.d.f.) to sample and extract a collision energy for a given interaction. The Bichsel model also provides a deposition model of how to distribute the charge in simulation. In Section 2.3.4, a charge deposition model based on Bichsel's recommendations was added to the ATLAS simulation framework. This ensures the shape of Pixel clusters is correct and allows these effects to be decoupled from the effect of radiation damage on cluster size.

2.2.2 Radiation damage in silicon

Radiation damage in silicon sensors can broadly be divided up into surface and bulk effects. This is because the thin metal, oxide, and nitride layers at the surface of a silicon sensor exhibit very different behavior following irradiation than the thick layer of low-dopant concentration silicon that forms the bulk of the detector.

Surface effects stem from the Total Ionising Dose, TID, which is the cumulative effect of ionisation in the surface dielectrics. Although the mechanisms that cause bulk damage can occur in the surface layers, the structural damage to the material affects its properties less than ionising radiation losses. Despite the relatively large ionisation energy in SiO_2 of 17 eV [85], high energy particles produced in collider experiments can still ionise the material as they pass through the detector. Many of the charge-carrier pairs produced by ionisation quickly recombine, however some will not and so drift apart in the electric field. The electrons in charge-carrier pairs created in the oxide layer will quickly be collected at the n^+ side of the detector due to their high mobility, $\approx 20 \text{ cm}^2\text{V}^{-1}\text{s}^{-1}$ [72]. Holes have a much lower mobility than electrons in SiO_2 , $\approx 2 \times 10^{-5} \text{ cm}^2\text{V}^{-1}\text{s}^{-1}$, due to interstitial traps⁵ and are unable to escape, leading to a positive charge accumulating in the oxide layer.

This effect is responsible for the electron accumulation layer that can short n^+ pixel implants in the silicon detector and necessitates the use of a p-spray or p-stop to mitigate its effect, see section 2.1.

⁵Traps and structural defects will be discussed further in the next section.

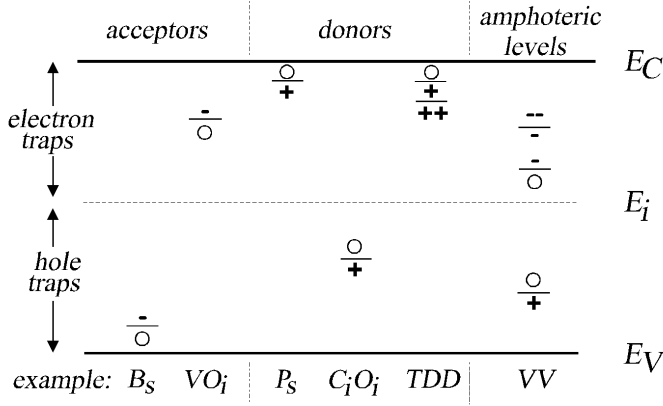


Figure 2.8: Energy levels of a variety of hole and electron traps. The energy levels in the trap are denoted with a negative (-), neutral (o), or positive (+) charge when they are filled with an electron or not [87].

Non-ionising radiation damage

Ionising energy loss in the bulk does not change a detector's electrical properties thus there is no concept of TID for the majority of the sensor, instead, the Displacement Damage Dose, which is the product of the particle fluence and the Non-Ionising Energy Loss (NIEL) is the dominant cause of radiation damage. This relationship is known as the NIEL scaling hypothesis and depends on both the particle interacting with the material and its energy [86]. NIEL is defined as the energy lost through elastic Coulombic and nuclear, and inelastic nuclear interactions between the incident particle and the lattice atoms, to produce vacancy-interstitial pairs and phonons [87]. NIEL damage is caused by a particle displacing an atom from its lattice position in the bulk. Often the displaced atom has enough energy to displace others in a 'knock-on' effect, in such cases the subject of the initial NIEL damage is the Primary Knock-on Atom (PKA). The clusters of damage that can form from a PKA consist of several types of lattice defects. A stable defect, consisting of a displaced, interstitial atom (I) and a vacancy (V) in the lattice, is known as a Frenkel pair (FP).

The average number of Frenkel pairs is given by:

$$\langle N_{FP} \rangle = \begin{cases} 1, & \text{for } E_d < E < \frac{2E_d}{\zeta} \\ \zeta \frac{E}{2E_d}, & \text{for } E > \frac{2E_d}{\zeta} \end{cases} \quad (2.1)$$

where E_d is the minimum energy required to displace an atom from its lattice site, E is the energy imparted to the lattice atom, and ζ is the displacement efficiency. This can be used to estimate the average number of Frenkel pairs created from a particle of known E_d and ζ , for example, in neutrons: $E_d = 25$ eV and $\zeta = 0.8$ [86]. Neutrons only transfer energy to silicon detectors in non-ionising interactions and so are used as a standard to measure how many Frenkel pairs and hence how much NIEL damage is done to the bulk. A hardness factor, κ , is used to convert between the a fluence of any particle (charged or neutral) and the fluence of 1 MeV neutron per square *cm*, also referred to as the 'neutron equivalent fluence' or $n_{eq} \text{cm}^{-2}$.

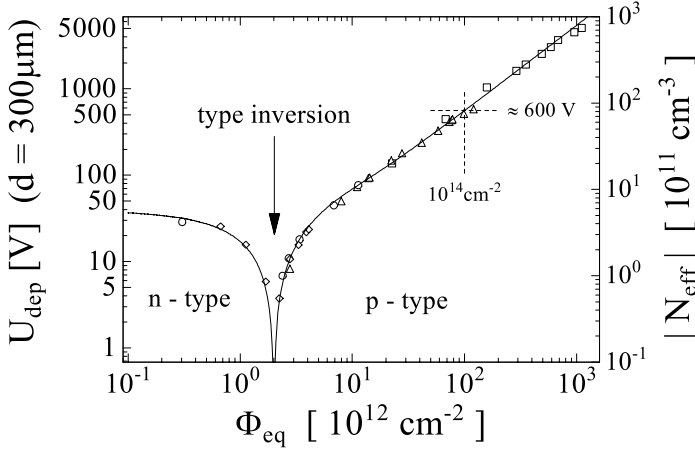


Figure 2.9: The change in effective carrier concentration (N_{eff}) and full depletion voltage (U_{dep}) with neutron equivalent fluence for a 300 μm n-type bulk silicon strip sensor [72].

2.2.3 Effects of NIEL damage on silicon detectors

The specific effects covered in this section are those relevant to the radiation damage modeling undertaken for this work, and are not a complete list. The production of charge carrier traps will first be considered, followed by the multiple effects of space-charge sign inversion including changing the electric field shape, the mobility of the charge carriers, and how annealing can be used to reverse some of the damage.

Charge trapping

If a Frenkel pair does not immediately recombine it can migrate in the lattice and react with other defects to become a trap. A trap causes a charge carrier to become stuck in its electrostatic potential well. In a shallow trap the charge carrier can usually escape simply due to thermal motion, in other words the trap's energy lies close to the respective charge carrier's conduction band. If a trap is deep then the operating conditions of the detector usually have to change for the charge carrier to be freed, see the discussion on annealing that follows. Conversely a deep trap is closer to the respective charge carrier's valence band.

Acceptors denote traps that will become negatively charged when filled and donors are traps that become positively charged when filled. Amphoteric defects introduce both donor and acceptor levels into the band gap. A common trap in silicon is a di-vacancy, VV . A di-vacancy, amphoteric defect introduces several acceptor states very close to the intrinsic Fermi energy in the band gap. In n-type silicon these are easily filled and contribute a negative space-charge to the bulk. Another common defect is a vacancy capturing an interstitial oxygen atom and forming one or two bonds, generally this trap is referred to as VO_i , which is an acceptor. These and other common traps are shown in Fig. 2.8.

If enough traps are present then free charge carriers traversing the bulk do not make it all the way to the electrode and instead become stuck in the trap.

Space-charge sign inversion

A common reaction in n-bulk silicon is the VP defect, which is formed by the n-dopant phosphorus (P) atom reacting with a lattice vacancy (V). This particular trap differs to the other

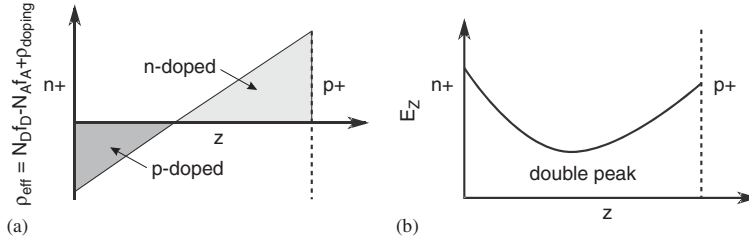


Figure 2.10: The effective charge density (a) and resultant field shape (b) in an irradiated module with trapping [94]

traps shown and discussed in the previous section in that it does not trap free charge carriers but rather the interstitial defect in the Frenkel pair.

Aside from removing the Frenkel pair's recombination centre, it also removes an electron donor from the bulk, changing the Fermi energy of the material [88]. Donor removal, along with acceptor addition from V_iO formation will result in changing the effective carrier concentration from n-type, with a positive space-charge, to p-type with a negative space-charge. This process is known as space-charge sign inversion, or type inversion, and affects a number of macroscopic detector properties.

A subfield of silicon detector physics attempts to offset space-charge sign inversion using defect engineering, where impurities, such as oxygen, are introduced to the bulk. A popular theory as to why this works is that V_2 defects are less likely to form than VO defects in a higher oxygen concentration and since VO is a more shallow acceptor trap, an electron is less likely to be trapped for long and contribute to the negative space-charge [89] [90] [91]. In the past decade there has been a particular focus in exploring the use of crystal growth techniques that introduce impurities during growth, for example magnetic Czochralski (MCz) has a homogenous distribution of oxygen, whereas Diffusion Oxygenated Float Zone (DOFZ) diffuses a layer of oxygen into the surface of the bulk after crystal growth. The DOFZ technique has been used successfully in the most recent iteration of ATLAS Pixel sensors for the innermost barrel layer, the Insertable B-Layer (IBL) [92] and will likely be used in ITk (D. Muenstermann, personal communication, March 2017), ITk prototypes are currently produced in the Float Zone method (FZ) [93].

The change in space-charge can also be viewed as a change in the effective doping concentration N_{eff} , as seen in Fig. 2.9. In the figure, an n-type material is irradiated with particles of a high hardness factor, leading to a decrease in the effective doping concentration before type inversion, and an increase after it. In p-type materials the combination of donor removal and acceptor introduction means that no type inversion will occur. This fact allows for the n^+ -in-p sensor designs used in ITk, see section 2.1.

Changes to electric field shape

Unirradiated sensors have an electric field shape that increases linearly with an increasing bulk depth, reaching a peak at the pn-junction. This shape is due to the relatively uniform space-charge density present in the material. However, once a sensor experiences NIEL damage, traps near both collecting electrodes become filled due to the generation-recombination current near each electrode [94]. The filled traps generate a space-charge near the electrodes: electrons are trapped at the n^+ implant causing a region of negative space-charge and holes are trapped in the p^+ implant, causing a region of positive space-charge. When the detector is fully depleted the n^+ implant usually has a positive space-charge, so the introduction of the space-charge region

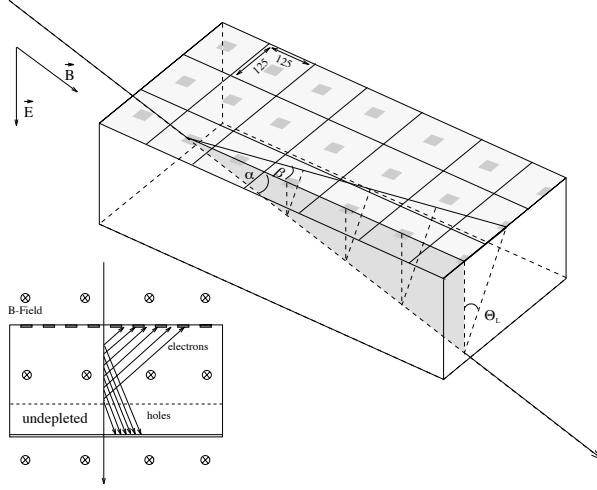


Figure 2.11: A 3D view of the various angles associated with a particle traversing a silicon detector in a perpendicular magnetic field. A slice of the volume, across the front face where the particle exits, is shown in the bottom of the figure and the corresponding drift due to the Lorentz force is shown for both charge carriers. α is the grazing angle of the particle, Θ_L is the Lorentz angle [95].

from trapped electrons creates an additional pn-junction, see Fig. 2.10. The introduction of these space-charge regions also creates a relatively neutral region in the sensor, which is usually skewed from centre. The double pn-junction and low space-charge region create a highly non-linear electric field. This causes the charge carriers to move quickly near the electrodes, and very slowly near the centre of the detector, see Fig. 2.12b. In simulation the non-uniform electric field shapes are modeled offline in a TCAD simulation, utilising the methods outlined in [94], which considers traps with two-defect levels in the band-gap. The number of traps and the number of filled traps changes with fluence, the results of this simulation for an IBL planar sensor irradiated at various fluences is shown in Fig. 2.12a. The simulation shows that even after $1 \times 10^{14} \text{ n}_{\text{eq}}\text{cm}^{-2}$ the junction position has moved from the front to the back of the sensor. At larger fluences, as more traps become filled near the electrodes, the double junction becomes more prominent, leaving a low-field region in the centre of the sensor.

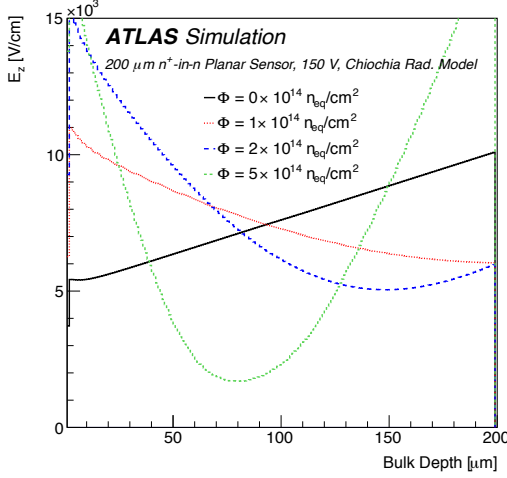
Drift due to the Lorentz force

The non-uniform shape of the electric field, including features like the double junction, changes the mobility of the charge carrier, according to the relationship in equation A.13. This affects the time taken to reach the electrode and recombine which also changes the likelihood of a charge carrier being trapped. The change in carrier mobility affects both the thermal diffusion of the charge carrier and the drift it experiences in a magnetic field, subtended by what is known as the Lorentz angle, see Fig. 2.11. The Lorentz angle is defined in equation 2.2:

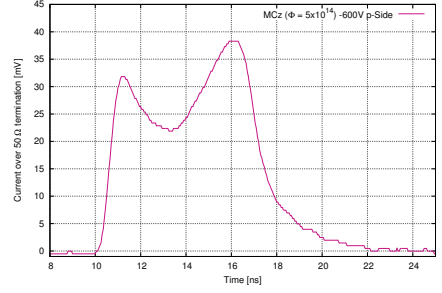
$$\tan(\Theta_L) = \mu_H(\vec{E}) \cdot |\vec{B}| = \mu(\vec{E}) \cdot r \cdot |\vec{B}| \quad (2.2)$$

where Θ_L is the Lorentz angle, B is the applied magnetic field⁶, μ_H is the Hall mobility which is related to the drift mobility, μ , by the Hall factor, r . The magnetic field is assumed to be

⁶In the ATLAS Inner Detector the magnetic field is applied by a superconducting solenoid, see section 1.2.2.

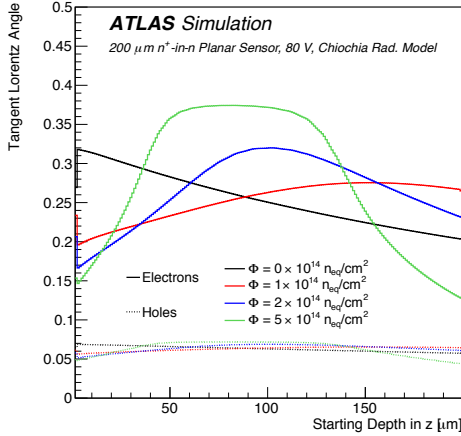


(a) Simulated electric field for 200 μm planar IBL sensor biased at 150 V at various fluences as a function of distance from the collecting electrode, where the contributions are averaged over the x - y plane [96].

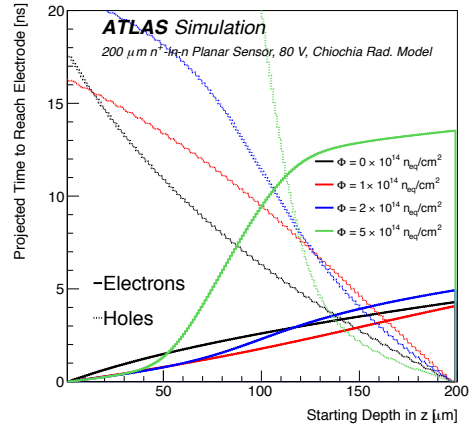


(b) The signal produced by electrons traversing an n^+ -in- n diode from backside to collecting electrode. The diode has been irradiated to $5 \times 10^{14} \text{ neq cm}^{-2}$. The double junction is clearly visible.

Figure 2.12: The double junction phenomenon seen in irradiated diodes and sensors (b) is well modeled in simulation (a) by the Chiochia model.



(a) The value of the Lorentz angle due to the inhomogenous electric field at various fluences.



(b) Time taken by a charge carrier to travel its respective collecting electrode at various fluences.

Figure 2.13: The electric field is used to calculate the charge carrier mobility and hence the Lorentz angle, simulated using a TCAD model in (a) for various fluences. This is used to determine total time taken to reach the electrode (b). Both plots were simulated with a 200 μm IBL planar sensor, biased at 80 V, where the contributions are averaged over the x - y plane and are displayed as a function of the charge carrier's starting distance from the collecting electrode [96].

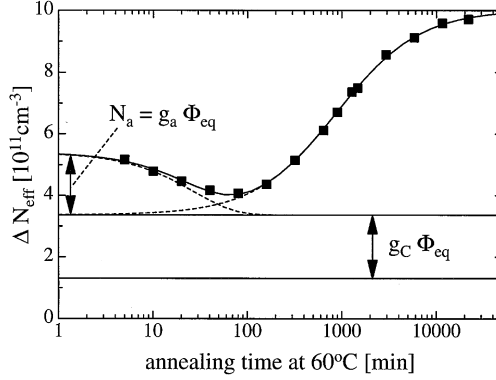


Figure 2.14: Beneficial and reverse annealing in silicon irradiated to $1.4 \times 10^{13} \text{ n}_{\text{eq}}\text{cm}^{-2}$. N_a is the contribution to the effective carrier concentration, N_{eff} , and is parameterised by g_a and the fluence Φ . The fit through the data points is employed using the Hamburg model [98].

perpendicular to the electric field in the detector volume. The Hall factor, r , accounts for the scattering mechanisms present in silicon and is dependent on temperature [97]. The Lorentz angle for an unirradiated module is effectively uniform but this changes with fluence. The change in Lorentz angle after various fluences at different depths in the sensor is shown in Fig. 2.13a.

For simulation, plots such as these are integrated to produce cumulative distributions, which can be used as look-up tables at runtime. The maps shown here are preliminary versions that will be used to simulate radiation damage. Work is currently ongoing to establish the most accurate conditions with which to build the maps. For example, the rate of trap generation, and the capture cross-section of traps are currently being explored to best represent the characteristics of irradiated sensors for a given fluence [96]. The change in carrier mobility also affects how long it takes for a charge carrier to reach the collection electrodes. Since charge carriers can be produced anywhere in the bulk it is again useful to produce a look-up histogram, like the one shown in Fig. 2.13b. These maps are also produced, along with the Lorentz angle maps, for every desired fluence.

Annealing

Traps are composed of defects, some of which can be dissociated by applying thermal energy to the bulk. This can cause defects to travel through the bulk, reacting with other defects, or recombine. Applying thermal energy to the radiation damaged bulk to change its electrical properties is known as annealing.

Annealing is a time and temperature dependent process. Although the exact mechanisms underlying phenomena observed following annealing are still an active topic of investigation, the behaviour is well characterised. The Hamburg model [87] is a phenomenological model that parameterises the effects of annealing. The Hamburg model describes the effective carrier concentration of a material according to five time and temperature dependent processes:

- (a) The removal of electron donors in n-type silicon.
- (a) The constant addition of stable, non-annealable defects that act as acceptors.
- (a) Changing defects that act as acceptors to defects that are neutral.

- (a) The introduction of neutral defects, which have the potential to become acceptor defects, into the bulk.
- (a) Changing neutral defects into acceptors.

Processes (a) and (b) occur during irradiation. Keeping irradiated silicon at 300 K on the order of days results in process (c), this is known as beneficial annealing. Beneficial annealing is so called because it increases the effective carrier concentration in n-type silicon before type inversion, see Fig. 2.14, which appears to reverse some of the damage. However, if the annealing is allowed to continue on the order of weeks, processes (d) and (e) will take over, this is called reverse annealing. The Hamburg model is implemented by fitting to detector data. Beneficial and reverse annealing effectively change the detector properties such that it behaves as if it were irradiated with a different fluence.

To implement this in simulation, the Hamburg model is fitted to experimental data and used to establish the effective doping concentration. The effective doping concentration is then supplied to a TCAD model of the sensor and used to recalculate the electric field map. At this time the change in trapping rate due to annealing is not accounted for in the simulation.

2.3 Digitization in Athena

Digitization is a simulation stage in Athena that takes hits generated by Geant4 and produces ‘digits’, or the corresponding digital response of a detector to the Geant4 stimuli. This is done in several stages: a detailed model of each subdetector is incorporated into the code which allows the subdetector community to fine-tune the analogue signal produced. This analogue signal is then combined with detector specific additions such as thermal noise and subsequently digitized. It is crucial that the digitization code accurately reflect the signal formation and processing in the actual detector so that the digits produced by the simulation can be compared directly to the data read out of the detector.

This section will provide an overview of the Athena PixelDigitization package which digitizes all four layers of the ATLAS Pixel detector. Following this section will be a discussion about changes made to the PixelDigitization package to reduce repeated code and improve the overall structure. In addition, the energy deposition model has been updated to better represent the spatial distribution of charge carriers in thin sensors using the Bichsel model. Finally, changes to the signal formation model to include radiation damage modeling will be described.

2.3.1 Athena overview

The Athena framework is an implementation of the Gaudi architecture which was originally designed for LHCb [99]. The Gaudi architecture is a high-level structure with well-defined interfaces and functionality upon which the software framework is built. Each structure in the architecture is easily decoupled and can be developed independent of the other structures. The Gaudi architecture produces ‘DataObjects’ and algorithms that operate on those objects. This approach is echoed in the Athena framework, shown in Fig. 2.15, where each stage of simulation produces a new data object [39]. Gaudi also defines a collection of ‘Services’ which provide the algorithms with a standard way to perform tasks common to the architecture, e.g. filling and reading a data container, processing job options, and event selection.

Athena is a complex framework that extends and specialises Gaudi. This includes the addition of ‘Tools’ and the ToolSvc. Services are setup and initialised once per job by the framework, with the same instance of a Service used by many Algorithms. A Tool acts as a sub-Algorithm,

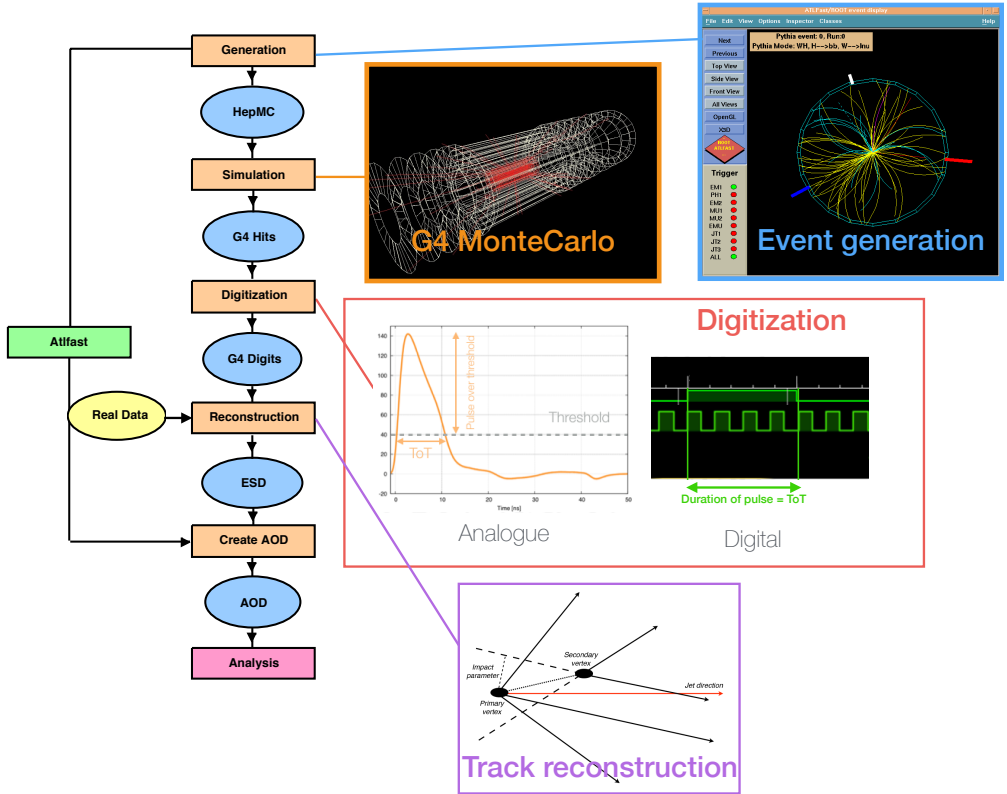


Figure 2.15: The Athena simulation workflow. Each stage of the simulation is shown with a box and the data object produced after each step is shown as an oval. The simulation data is processed until it is in the same format as the incoming detector digits. Thus the real data and the simulation data both enter the workflow at reconstruction.

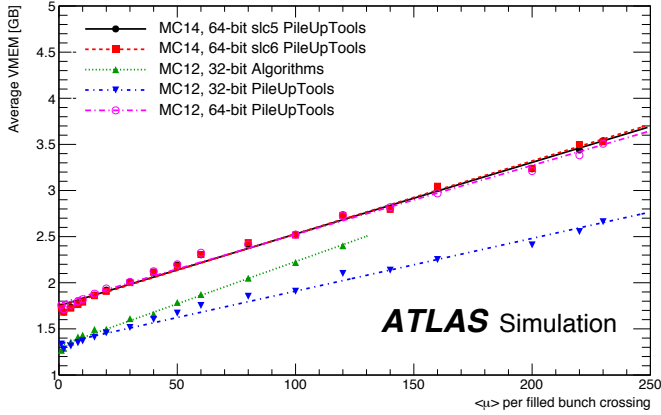


Figure 2.16: The average memory use for several digitization configurations as a function of the pile-up, $\langle\mu\rangle$. In comparing the 32-bit Algorithms (green triangles) and PileUpTools (blue inverted triangles) it is clear that the Algorithm approach diverges from the PileUpTools memory usage. The Algorithms data is cut short as it reaches the memory limit of the machines [100].

performing operations on specific subsets of a data object. In 2014, development in Athena moved away from basing functionality in Algorithms into Tools [100]. This is because of the different approaches both classes have to storing events, which become imperative to performance when processing collisions with a large amount of pile-up. Algorithms keep a cache of background events in memory for each event, which requires increasingly large memory usage as pile-up increases, whereas the PileUpTools only load those events that are needed, which increases the CPU consumption dealing with I/O, see Fig. 2.16. The average memory use for several digitization configurations as a function of the pile-up, $\langle\mu\rangle$. The figure shows that the 32-bit Algorithms (green triangles) memory usage increases with pile-up at a higher rate than any of the Tools. The Algorithm data is cut short due to jobs at higher pile-up failing after reaching the memory limit of the cluster nodes the simulation was run on. Whilst these differences are minimal for the pile-up conditions in Run 1, they are prohibitive for simulating data for Run 2 and HL-LHC. As such, the restructure and additions described in this chapter will focus on Tools instead of Services.

2.3.2 Pixel digitization package

Hits simulated in Geant4 that pass through Pixel detector elements are digitized in the Pixel digitization package. The package contains two algorithms, one for debugging and one used in production. The latter makes a call to a single Tool, that contains a set of other Tools that are called sequentially to perform each part of the simulation. Various tools within the package make call to services outside it, such as the Pixel Calibration Service, which performs the charge-to-ToT conversion. The Pixel digitization Tool provides functions to deal separately with pile-up and hard-scatter signal events, which are beyond the scope of this thesis and detailed here [100] [101]. The core functionality of the tool for the scope of this work is summarised below:

1. **DigitizeAllHits** loops over the hits in an event with a function called **DigitizeElements** which creates a collection of pixels and their corresponding collected charges for the event

in a container called `chargedDiodes`. This container is then digitized from charge to ToT in a function that looks up the operating conditions of the front-end, such as the discriminator threshold, and then creates the an RDO.

2. **DigitizeElements** uses information in the Geant4 hit pointer to look up the identity of the detector element the hit passed through, in this case a Pixel sensor. The element stores information about the location of the module in the detector, for example whether it is a barrel or end-cap module and what its η and ϕ coordinates are. The detector element is then passed, along with the `chargedDiodes` collection, to the **SurfaceChargesTool::process** function.
3. **SurfaceChargesTool::process** uses the detector element information to pick the correct SubCharges Tool. Whilst the majority of the sensors in the Pixel detector have a planar geometry, the IBL also contains 3D sensors, both of which are represented by a different SubCharges Tool. In addition, a different SubCharges tool is used to distinguish sensors in the IBL, the three outermost layers of the Pixel detector barrel, and the end-caps. The Tools retain information about physical properties of the sensor, for example sensor thickness and pixel pitch. Once the correct SubChargesTool has been selected, the function **SubChargesTool::charge** is called.
4. **SubChargesTool::charge** contains the code that uses the information stored from Geant4 (the total energy deposition, entry point, exit point) to calculate the charge induced on the sensor's collecting electrode.

The majority of the changes made to the Pixel digitization package were made on functionality in the SubCharges Tool. As such the next section will describe the design of the Tool before modification, followed by a discussion and outline of the modifications made during the restructure.

2.3.3 SubCharges Tool

The modifications made to the Pixel digitization package discussed in this thesis are concerned with depositing energy and recording charge in each pixel, i.e. energy deposition and signal formation. In the description of the class below, variable names may differ from those used in the original digitizer code, for clarity, these are explained in Appendix A.3. The names used here are those used in the modified code.

Energy deposition

The Geant4 step of the simulation provides three pieces of information accessible during digitization:

- Energy lost by a particle in the detector volume.
- Particle entry point, measured in *local* η , ϕ , and depth direction.
- Particle exit point, measured in *local* η , ϕ , and depth direction.

The entry and exit point are used to calculate the path length of the particle through the sensor:

$$\text{pathLength} = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2 + (\Delta\text{depth})^2} \quad (2.3)$$

where the coordinate variables are defined for the 'local' with a coordinate basis in the sensor as opposed to the centre of the ATLAS detector. The energy loss is then deposited uniformly

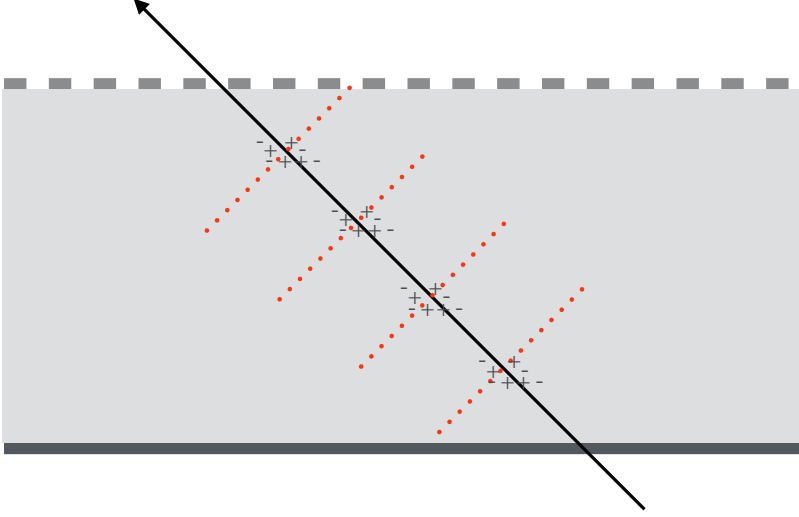


Figure 2.17: The uniform deposition of energy lost by the traversing particle in the sensor bulk. The red lines show the even divisions of the path length with equal amounts of charge-carrier pairs created per division.

across the path length in steps, see Fig. 2.17. The step-size is defined by the sensor thickness and an initial number of steps (typically 100). The number of steps to process the energy deposition is then scaled by the path length:

$$\begin{aligned} \text{stepsize} &= \frac{\text{sensor thickness}}{\text{initial nSteps}} \\ \text{scaled nSteps} &= \text{int} \left(\frac{\text{path length}}{\text{stepsize}} \right) + 1 \end{aligned} \quad (2.4)$$

The energy deposited in each step is further divided into ‘chunks’. To keep the processing time the same regardless of path length, the initial number chunks is scaled by path length. If the path length is large relative to the sensor thickness, the number of chunks per step will be smaller, i.e. each chunk will represent a larger amount of energy loss than if the path length were smaller. Equation 2.5 defines the scaled number of charge chunks in each step.

$$\text{scaled nChunks} = \text{initial nChunks} \cdot \left(\frac{\text{initial nSteps}}{\text{scaled nSteps}} \right) + 1 \quad (2.5)$$

The amount of energy deposited by the particle into the bulk is then equally split between all the charge chunks along the path:

$$\text{energyLoss per chunk} = \frac{\text{total energy loss}}{(\text{scaled nChunks}) \cdot (\text{scaled nSteps})} \quad (2.6)$$

Which is then converted to charge by dividing by the average energy needed to ionise silicon: 3.6 eV.

To summarise, this portion of the simulation places a scaled number of charge chunks, with each chunk representing an integer number of fundamental charge carriers, at intervals along the path known as steps. Whilst this energy deposition is easy to simulate and understand it does not accurately represent distribution of energy along the path length that was described by the Bichsel model in section 2.2.1. Modifications to the simulation to reflect this are discussed in section 2.3.4.

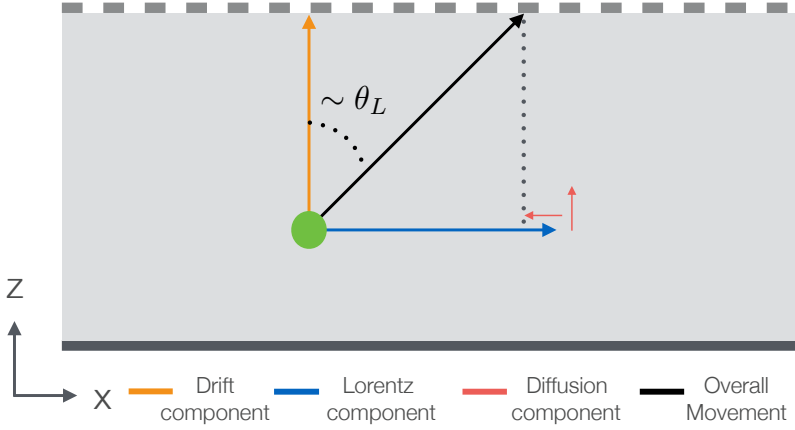


Figure 2.18: For each cluster of charge carriers, drift and diffusion are applied to identify the collecting electrode. The cartoon shows how these contributions are broken down in the code, the contributions are not to scale.

Signal formation

After energy has been deposited uniformly in the bulk, the subCharges tool applies drift and diffusion to assess which electrode will collect the charge from the chunks and be added to the chargedDiodesCollection. The charge carriers drift towards their respective electrodes, perpendicular to the sensor surface, due to the applied bias. In a magnetic field the Lorentz force will also cause the charge carriers to drift, in barrel modules in the ATLAS detector this motion is parallel to the pixel surface. The charge carriers will not follow the electric field lines exactly but exhibit some thermal motion around the path, known as diffusion. These contributions to the final position of the electron in the charge carrier pair are shown in Fig. 2.18.

This is implemented by first iterating over each step along the path length, and within that loop iterating over each chunk of charge carriers. In the first step, the distance (in sensor depth) from the charge chunks to their respective collection side is calculated. Then, in the second step, the location of the collecting pixel is calculated by using the lateral displacement due to the Lorentz angle and the distance from the first step. The diffusion is added as a random number throw, scaled by the depth of the pixel over $300 \mu m$. The signal formation logic is summarised Appendix A.1 in the pseudocode snippet 1.

Once all the collecting pixels for each charge chunks have been calculated, the final position of each is converted to a pixel index and stored for digitization. It is assumed that each chunk of charge is collected at its corresponding pixel with perfect efficiency, so the equivalent amount of charge per chunk recorded in each pixel is then digitized.

Discussion

The SubCharges Tools are relatively short pieces of code which nevertheless simulate a lot of functionality. However, some of the assumptions made may affect the accuracy of the simulation.

The energy deposition part of the code assumes a uniform distribution in space, whereas the spatial deposition is actually governed by a Poisson process, as described in section 2.2.1. The Geant4 simulation prior to digitization samples the straggling function for the detector volume

to obtain the energy loss but does not provide a spatial distribution. The code was modified to correct for the spatial distribution by rerunning the energy distribution as a Monte Carlo simulation, which is described in section 2.3.4.

The signal formation portion of the code only drifts and diffuses the carrier collected by the readout-side electrode, which in the Pixel detector is electrons. The holes are essentially ignored but the implied behaviour is that they drift by the same amount as electrons and are collected on the backside. Although both holes and electrons are generated in equal amounts during ionisation, electrons make up the dominant portion of the signal, as will be explained in Section 2.4.2. Since holes have a smaller mobility than electrons, they drift less under the Lorentz force and so could end up underneath a different pixel than their partner electron. At maximum the two could differ by one pixel, so it is unclear if this has a significant effect on the cluster shape. However, the code was modified to allow a separate treatment as holes and electrons must be treated separately when simulating radiation damage, as they have different trapping rates.

The only difference between the treatment of each charge chunk in a given step is the diffusion applied. The diffusion is scaled according to a pre-calculated value that assigns an average diffusion length of $7\text{ }\mu\text{m}$ for a $300\text{ }\mu\text{m}$ piece of silicon and scales it according to the actual path length. However, diffusion, as was seen in equation A.7b, is dependent on both temperature and charge carrier mobility (and hence electric field) and is not constant. The newer versions of the signal formation model thus implements the Einstein diffusion relation via:

$$\sigma = \sqrt{2Dt} \quad (2.7)$$

where σ is the spread of the arrival position, D is the diffusion constant, and t is the time the particle is in motion for. This, however, does not account for effects at large bias voltages which affect the mean free energy of the electron and hence the mean diffusion [102]. Studies are currently underway to decide how best to model this specific dependance.

2.3.4 Digitization restructure

This section describes efforts to reduce the number of classes and duplicated code in the PixelDigitization package and simplify the workflow. This discussion is preceded by a brief summary of recent modifications, by another student, to the energy deposition portion of the code, included here to motivate the restructure [103].

New energy deposition model

In section 2.2.1, it was shown that energy is not deposited uniformly throughout the bulk during ionisation, instead the particle traversing the bulk undergoes random collisions with lattice electrons and can deposit a range of energies in each collision. Bichsel suggests a good way to simulate the spatial deposition of energy in the 2006 review [80], which is the approach followed in the Athena implementation.

The Bichsel model does not start off with a pre-defined number of steps or charge chunks but first runs a Monte Carlo over the path length of the particle in the detector volume. This Monte Carlo proceeds by simulating each collision in the bulk and then deciding on the energy loss of that collision. This is done with a Poisson distribution describing the probability, $P(n)$, of a particle undergoing n collisions with the detector material in a segment, x :

$$P(n) = \frac{m_c}{n!} e^{-m_c} \quad (2.8)$$

where m_c is the average number of collisions for all particles in x :

$$m_c = x \Sigma_t = \frac{x}{\lambda} \quad (2.9)$$

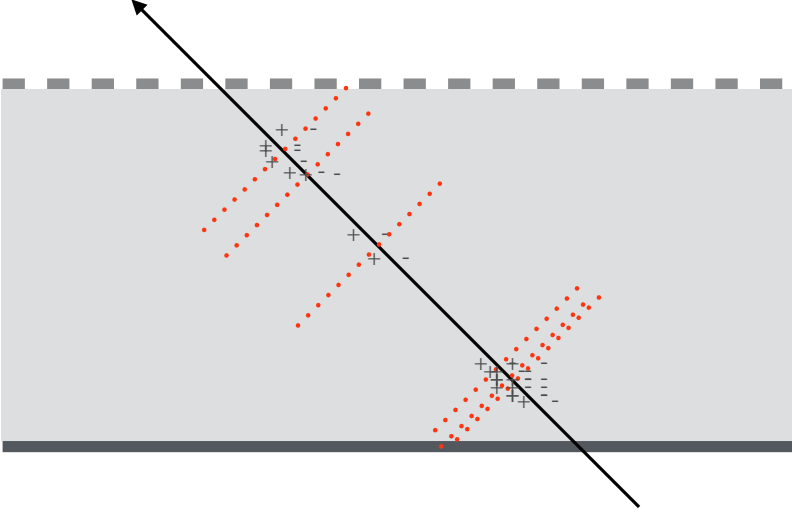


Figure 2.19: A Poisson-distributed spatial distribution of ionisation. The energy lost by the traversing particle in the sensor bulk is sampled from a distribution that includes the Landau-Vavilov-Bichsel corrections. The red lines show the collision points of the particle with the bulk, note that the amount of energy, and hence the number of charge carriers, deposited per collision is not the same from collision to collision.

where Σ_t is the integral of the differential cross-section multiplied by the density of the material and λ is the mean free path between collisions. The differential cross-section is usually computed from the Fermi Virtual Photon (FVP) approach, which requires the $\beta\gamma$ of the incident particle, and various properties of the detector, such as the dielectric function and the atomic number. As such, look-up tables for 6 particles including the electron, muon, proton, pion, and kaon, produced in the event generation have been added to the package. If a particle produced in the event generation is not in that list then the default uniform deposition model is used. The look-up tables contain entries for a particle's $\beta\gamma$, Σ_t , and the most probable collision energy $ColE$. The procedure for the Bichsel deposition model then goes as follows:

1. Obtain the traversing particle's type (pdgID), e.g. electron, proton, etc, and $\beta\gamma$ from the **genPart_4V** handle, which contains the four-vector of the particle from the event generation.
2. Using this information, access the look-up tables and determine the integrated cross-section, Σ_t (and hence the mean-free path before a collision occurs, for a particle of this type, this energy, in this material).
3. Enter a simulation loop. For each step in the loop throw a random number for the distance travelled by the particle before a collision happens.
4. Check to see if the particle has travelled outside the detector volume.
5. If not, throw another random number, between 0 and Σ_t and interpolate the collision energy based on this value and $\beta\gamma$. Check that the energy loss is less than the amount needed to produce a delta ray. If not, ignore it as delta rays are already simulated by Geant4.

6. Record the position and energy loss and save to the running total.

This is also summarised in Appendix A.1 in the pseudocode snippet 2. One potential continuity issue with this method is that it does not use the total energy loss provided by Geant4, however since most of the particles traveling through the detector are minimum ionising, the differences between what Geant4 propagates to the calorimeters and the particle energy loss simulated in the pixel layers is negligible. This simulation also always uses the particle's starting energy from the event generator, and so may overestimate $\beta\gamma$, however for the same reasons above this discrepancy is likely negligible.

Although not shown explicitly here, it's worth noting that the initial number of chunks and steps are still used to cluster the energy deposition via a weighted mean such that the signal formation part of the code still has the same number of steps.

This model has recently been validated against data showing it to more accurately represent cluster shapes seen in data [103]. More verification will be made with recent data once the radiation damage model, which also changes cluster shape, is implemented in Athena.

Restructure of subCharge tools

At the time of writing the Pixel digitization package contains nine, near-identical classes which derive from the SubCharges Tool: DBM, IBLPlanar, IBL3D, PixelBarrel⁷, PixelEC, IBLPlanarBichsel, IBL3DBichsel, PixelBarrelBichsel, and PixelECBichsel. The differences between most of the classes are minimal:

- The PixelBarrel and PixelEC classes are identical.
- The IBLPlanar and PixelBarrel/EC tools are identical except for an additional 20 lines of code in the IBL tool that accounts for the reduced collection efficiency in IBL planar sensors due to their slim-edge design.
- The DBM⁸ tool differs from the PixelBarrel/EC tools only by its hardcoded physical constants given that its sensor is made from diamond instead of silicon (e.g. diffusion constant, number of charge-carrier pairs produced per eV) and a rudimentary treatment of trapping to change the charge collection efficiency. The code is not commented, so it's not clear where the hard-coded values come from. Since they are not dependent on any parameters drawn from the conditions information it is assumed this trapping treatment is not related to radiation damage. The DBM code was not within the scope of this project and will not be discussed further.
- The signal formation portion of the 3D charge tools differ quite extensively from its planar equivalent but shares the same energy deposition model. The 3D tool is not studied in this thesis but it should be noted that the techniques and methodology described in this chapter were used to implement a radiation damage simulation of 3D sensors into Athena in 2018. A standalone simulation of radiation damage in 3D sensors is discussed in the thesis of Veronica Wallangen (2019, unpublished at time-of-writing).
- All the comparisons above, excluding DBM, extend to the Bichsel equivalent tools.

Within the planar-sensor classes, the differences between the tools are either non-existent or minimal. All tools, regardless of material or sensor dimensions, share the same energy deposition

⁷The naming convention here is to refer to the pixel layer closest to the beam-pipe as IBL and the outer 3 layers as 'Pixel'.

⁸Diamond Beam Monitor, a luminometer forward of the Pixel detector.

model. Moreover, the Bichsel tools incorporate the uniform treatment of energy deposition as a backup in case the Bichsel deposition fails, so the energy deposition model is also duplicated in the Bichsel tools. The situation is summarised in the left cartoon of Figure 2.20, where the

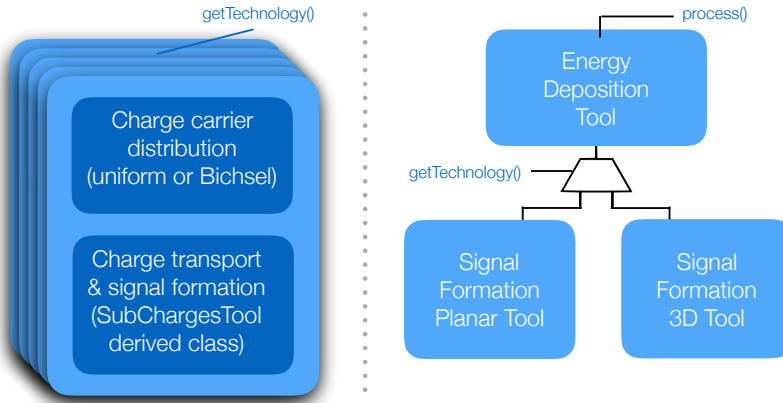


Figure 2.20: A graphical representation of how different types of sensor geometry are handled in the current PixelDigitization chargeTools release (left) and how the restructured tools handle different sensor geometry (right).

SurfaceChargesTool picks one of these ‘technology’ classes based on what part of the detector the module resides. To increase code comprehension and reduce the number of classes in the PixelDigitization package, the class-structure on the right side of Figure 2.20 was implemented.

In the pre-existing code, all SubCharges Tools implement energy deposition and signal formation separately, moreover all share the same energy deposition model which is independent of sensor geometry. As such, in the restructure, this portion of the code was extracted into a separate class, the Energy Deposition Tool and is flagged as either silicon or diamond.

The signal formation models were then divided into planar and 3D sensor geometries. The energy deposition selects which Tool to use with the same getTechnology() function call that determines the sensor material. Then, only two signal formation tools are needed: one for 3D implants (the SignalFormation 3D Tool) and one for planar implants (the SignalFormation Planar Tool).

The way the SignalFormation Tools work means that pixel geometry, sensor thickness, and sensor size are stored in a database of sensor geometries currently obtained by the Silicon Detector Element class. So the SignalFormation Planar Tool can be used for IBL, the three outer layers, and the DBM. To deal with the additional constraints on IBL and DBM, a flag is passed into the tool to assert the relevant code blocks. In this way nine classes is reduced to three. This is also useful for extending to future technologies such as monolithic CMOS modules or Low-Gain Avalanche Detectors, by simply adding an additional SignalFormation class.

2.4 Radiation damage simulation

This section will describe how specific features of radiation damage are modeled and were integrated into the Athena SignalFormation Tools. The radiation damage simulation study predates this thesis by over a decade, in part by using a standalone Geant4 package called Allpix [104]. Whilst the various aspects of simulation were studied in Allpix, the final implementation,

structure, and logic were original work.

Radiation damage is implemented in the Signal Formation simulation by:

- Trapping charge carriers at a rate related to the fluence the sensor has experienced.
- Changing the electric field shape of the sensor to account for changes to its structure, which in turn changes its mobility.
- Accounting for the smearing of induced charge when a carrier is trapped.
- Treating holes and electrons separately.

Surface charge effects are considered indirectly by the change in electric field shape near the electrode implants. The remainder of the chapter will discuss each of these points, concluding with the structure of their implementation in the Signal Formation Tools.

2.4.1 Trapping time and position

Charge trapping can be simulated according to a characteristic lifetime, which is dependent on fluence. The fraction of charge carriers that are *not* trapped after some time, $N(t)$, is given by equation 2.10.

$$\frac{N(t)}{N(0)} = e^{-\frac{t}{\tau}} = e^{-\beta_d \Phi t} \quad (2.10)$$

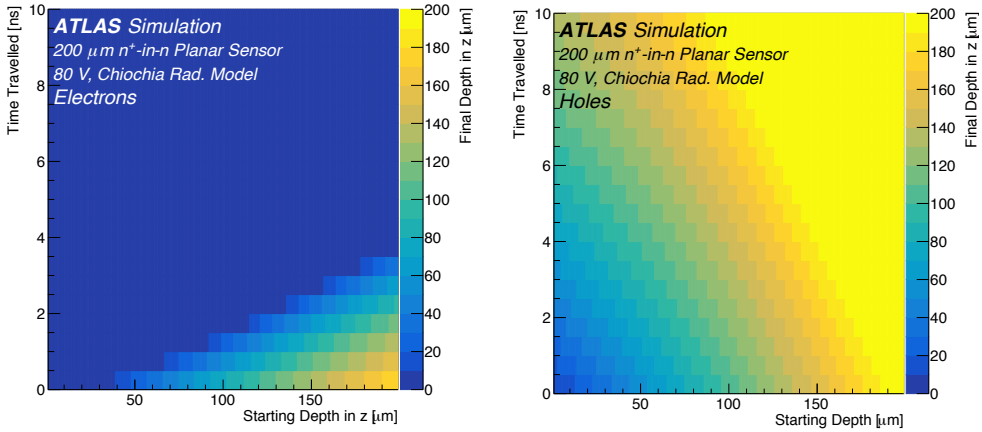
where $N(0)$ is the initial number of free charge carriers, τ is the characteristic lifetime which is given by the the damage parameter, β_d , and the fluence, ϕ , [87]. Trapping is simulated by throwing a random number for the fraction carriers not trapped and deducing for how long the chunk of charge carriers would have to move in the bulk for that to happen.

$$t_{\text{trap}} = -\frac{1}{\beta_d \Phi} \ln(u) \quad (2.11)$$

where $u \in [0, 1]$ is a random number drawn from a uniform distribution, and t_{trap} is the time the charge carriers move for until they are trapped, also known as the trapping time. To determine if a chunk of charge carriers is trapped, the trapping time is compared to the time-to-electrode, which is looked up in the stored values shown in the previous section. If the trapping time is less than the time to electrode, the chunk is said to be trapped. Once the charge chunk is trapped, the trapping time can be used, along with the Lorentz angle and carrier mobility, to find the position of the trap. It is assumed that charge carriers trapped in the signal formation process do not de-trap within the collection period of the readout chip and so parts of the signal from ionisation will be lost. The position of the trap in the bulk is used to calculate the charge induced on the collecting electrode, which is explained in the following section. The trap-location calculations are performed before running the simulation and stored in look-up tables, an example of the trap location table for an unirradiated IBL planar sensor is shown in Fig. 2.21.

2.4.2 Signal formation in a damaged bulk

Charge-carrier pairs, created by a high energy particle ionising the bulk of a silicon detector, when coupled with a charge sensitive amplifier, induce a signal as they move in the bulk. Signal formation does not begin when the charge carriers recombine at their respective electrodes but instead from when they begin to move in the electric field, immediately after ionisation. Initially, it seems that modeling charge collection only as recombination would not affect the simulation.



(a) The location of an electron trap, given initial position and trapping time.

(b) The location of a hole trap, given initial position and trapping time.

Figure 2.21: The location of a trap given the charge carrier starting position and trapping time. The electrons are collected at 0 and the holes at $200\ \mu\text{m}$ in this figure. The plots were produced using the simplified model for unirradiated IBL planar sensors [96].

The integrated induced charge is the same if the charge is drifted in the electric field as if it is just assumed to recombine at the electrode. There would be some offset in the timing: if an electron-hole pair was formed near the backside of an n^+ -in- n detector, the free electron would have to travel the entire depth of the bulk before it recombined, which takes on the order of 10 ns. If instead a majority-holes sensor was used, like the SCT p^+ -in- n , a hole travelling the full width of the bulk could take on the order of 50 ns or 2×25 ns bunch-crossings to arrive. However, since the shape of an induced signal is not simulated, this would not affect the simulated charge in the absence of trapping.

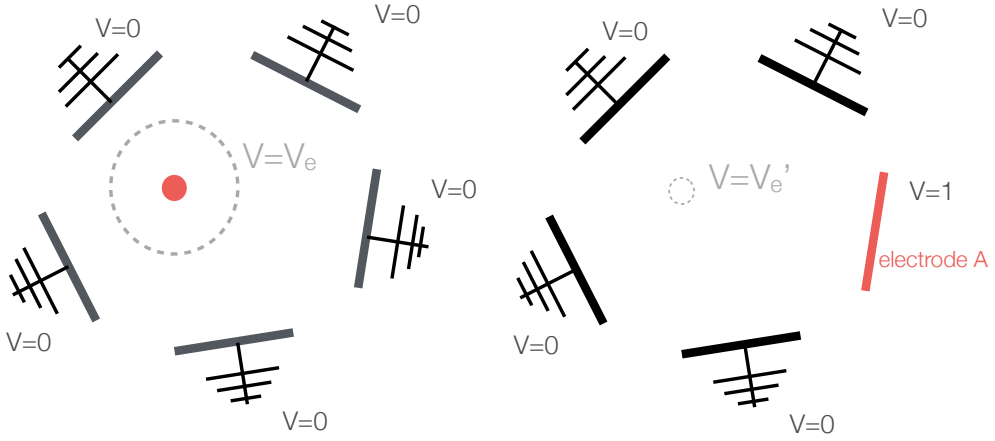
However, if a charge carrier is trapped it induces a signal on both electrodes up to the moment it is trapped. In this case, the induced charge on the electrodes is not equal to the collected charge and has to be calculated separately. This section will discuss the induced charge is calculated and how this procedure has been optimised for simulation.

Prior to the 1930's, induced charge was calculated by moving an electron along the electric field lines of the detector in small increments and calculating the corresponding change of voltage on the detector surfaces by assuming the total energy of the system had not changed. This form of iterative 'energy balancing' did not account for defects in the lattice nor space-charge [105].

Another way of calculating the charge induced by a moving charge carrier on an electrode is to solve Gauss's law. Doing so requires a detailed Electric field map that takes into account the bias voltages, space-charge, and moving charges at every time instance. Whilst this may be possible for simple geometries such as parallel plate capacitors it becomes a lengthy and impractical calculation for larger, more complex geometries.

Ramo-Shockley weighting potential

In the late 1930's Shockley and Ramo independently published a means to calculate the induced charge on an electrode [106] [107]. The resulting 'weighting field' provides a means of calculating the charge induced on an electrode simply from knowing the geometry of the system, and the displacement of the charge carrier over its full range of motion. The weighting field can be



(a) A test charge in a complex geometry of grounded electrodes. (b) The test charge in figure (a) is removed and an electrode is raised to unity potential.

Figure 2.22: A graphical representation of the derivation of the weighting field from Green's theorem.

derived from setting up the following two cases. First a single charge carrier is surrounded by some complex electrode geometry, see Fig. 2.22. If all the electrodes are grounded, Gauss's law can be used to calculate the flux of the carrier through a Gaussian surface, shown as a sphere of potential V_e in Fig. 2.22a. Starting from Green's theorem [108]:

$$\int_{\text{vol.}} (V' \nabla^2 V - V \nabla^2 V') dv = - \int_{\text{surf.}} \left(V' \frac{\delta V}{\delta n} - V \frac{\delta V'}{\delta n} \right) ds \quad (2.12)$$

where V is the potential of the electric field between the conductors, $\frac{\delta}{\delta n}$ is the differential with respect to the normal of a surface. If the test charge is removed from the geometry and one of the electrodes, historically called electrode A, is raised to have a unit potential, the potential of the field is then V' , see Fig. 2.22b.

Green's theorem is then evaluated for these cases. If the volume evaluated on the left hand side is bounded by all conductors, both the Laplacians evaluate to zero as there is no net charge. The right hand side of the equation can then be evaluated for three cases, where a surface integral is performed over:

1. All electrodes, except A, will, by definition, evaluate to 0.
2. Electrode A will result in the first term in equation 2.13.
3. The equipotential sphere will result in the second term in equation 2.13.

Adding the results of these three cases together to form the complete picture results in equation 2.13. From equation 2.12, the induced charge, Q_A , on electrode A can be derived:

$$\begin{aligned} 0 &= - \int_A \frac{\delta V}{\delta n} ds - V_e' \int_{\text{sphere}} \frac{\delta V}{\delta n} ds \\ &= 4\pi Q_A + 4\pi e V_e' \\ \therefore Q_A &= -e V_e' \end{aligned} \quad (2.13)$$

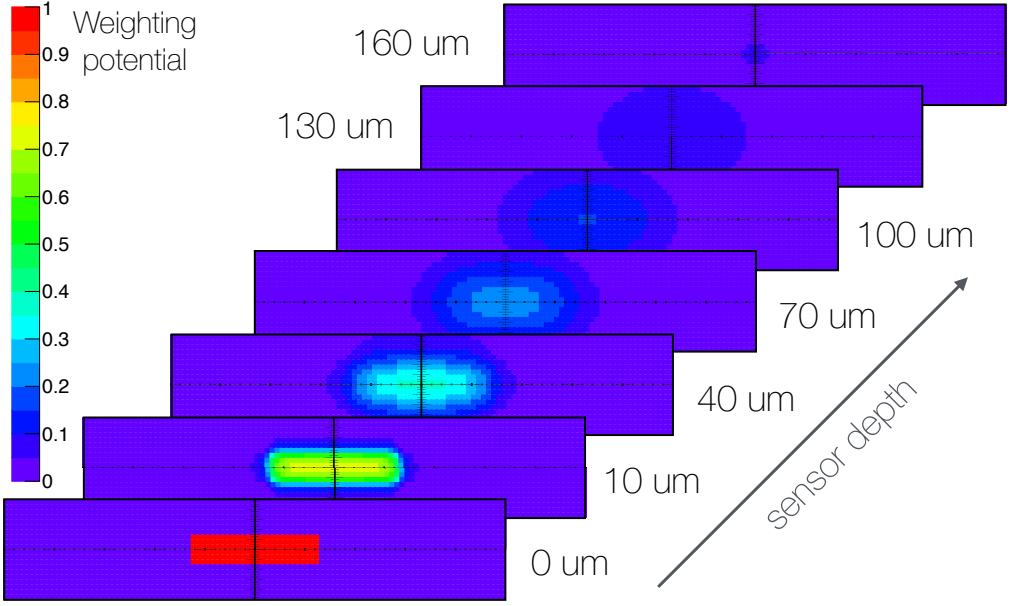


Figure 2.23: Slices of the generated 3D weighting potential map shown at different depths in the sensor. The weighting potential maps is of a quarter of an IBL pixel ($50 \times 250 \mu\text{m}$) which extends for 2.5 pixels but is shown here for the full pixel.

where V'_e is the weighting potential [106]. The weighting potential, unlike the electric potential, depends only on the geometry of the electrodes and not operational conditions such as bias voltage or temperature. This is a powerful tool as the weighting potential can be calculated and stored ahead of time, reducing the need to perform calculations during simulation.

For radiation damage modeling the weighting potential is used to find the induced charge on an electrode if the charge carrier is trapped. This is done by looking up the value of the weighting potential at the starting position of the charge carrier and the value of the weighting field at the position where the charge carrier is trapped:

$$Q_{\text{induced}} = e \left(\phi(\vec{s}_{\text{end}}) - \phi(\vec{s}_{\text{start}}) \right) \quad (2.14)$$

where $\vec{s} = (x, y, z)$ is the position of the charge carrier in the bulk, ϕ is the weighting field, and e is the charge of an electron. An example of the weighting potential for a single pixel at increasing sensor depth is shown in Fig. 2.23, the maps shown in the figure are those used in the simulation of IBL planar sensors.

Induced charge on neighbouring electrodes

The weighting potential can also be used to calculate the induced charge on adjacent electrodes. In Strip and Pixel detectors, collecting electrodes are closely packed and it is feasible that a free charge carrier could pass close to more than one before either recombining or being trapped. Without trapping, the charge carrier travels to an electrode and recombine and any signal induced on an adjacent electrode integrates to zero [109]. The current pulse on the adjacent electrodes is bipolar, see Fig. 2.24, which means that if the charge carrier is trapped a non-zero charge will be integrated on that electrode. This effect is accounted for by looping over all nearest neighbour pixels, see Fig. 2.25. A weighting potential calculation on the geometry

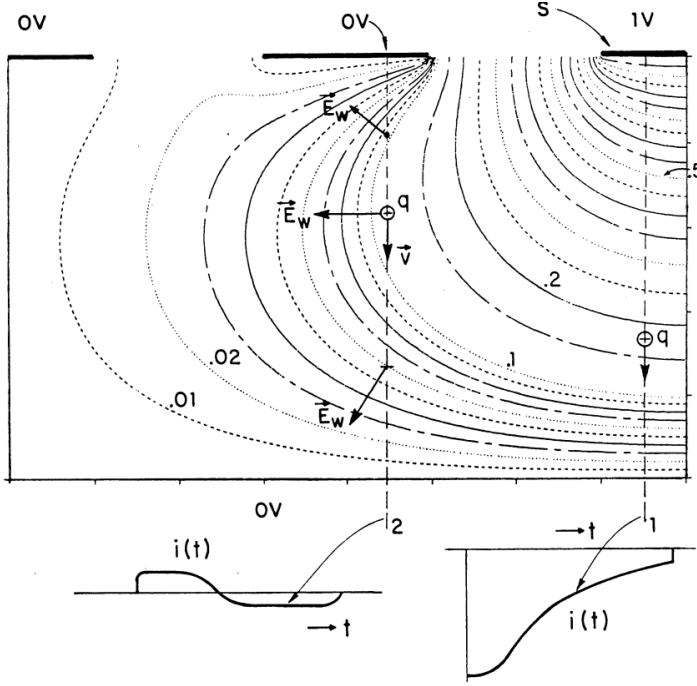
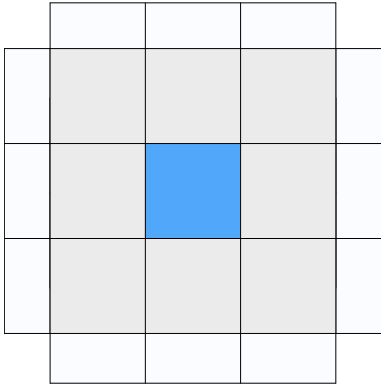
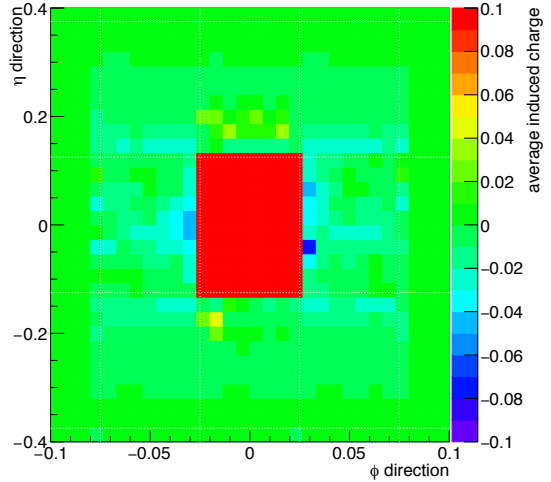


Figure 2.24: A charge carrier moving in the weighting potential, with the field norm. labelled $\vec{E}_w$ (main image) and resulting current flow in strip detector electrodes (bottom). The weighting potential shown here is calculated for the collecting electrode, labelled '1'. Adjacent electrodes are labelled '2' [109].



(a) Charge is trapped in the blue pixel, so loop over neighboring pixels (grey) and look up charge induced. The white pixels are part of the weighting potential map but since the induced charge with this pixel geometry is effectively zero (see (b)), they are not checked.



(b) The induced charge on a 3.5×3.5 pixel region when charge is trapped in the central pixel (red) for an n^+ -in-n, or electron-collecting electrode. The simulated fluence is set to $5 \times 10^{14} \text{ n}_{\text{eq}}\text{cm}^{-2}$. The pixels are $50 \times 250 \mu\text{m}$, as in IBL. Produced in preliminary Athena implementation, courtesy of Ben Nachman.

Figure 2.25: The effect of trapping on induced charge on neighbouring pixels.

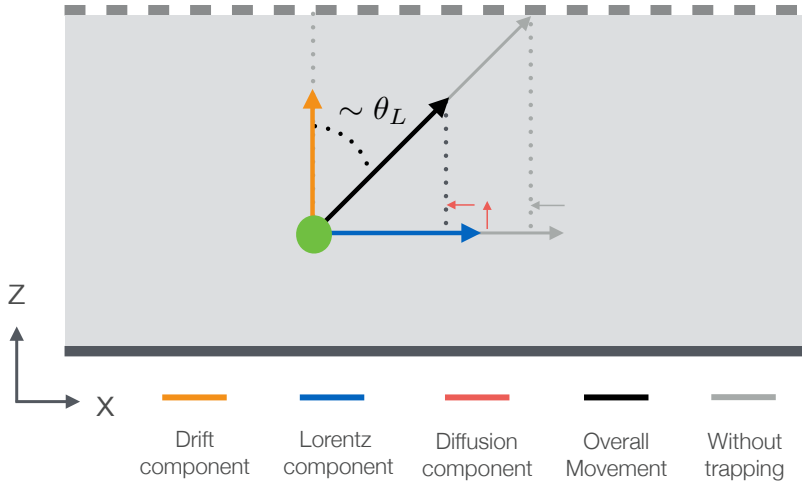


Figure 2.26: The contributions to signal formation from a trapped chunk.

of ~ 2.5 pixels is made beforehand and stored in a histogram, which is essentially used as a look-up table at runtime. By storing the weighting field for more than one pixel, the charge induced on neighbouring electrodes can be checked easily.

2.4.3 Athena implementation

The entry point to radiation damage modeling occurs after energy deposition. As before, there is a loop over each step, where energy has been deposited, and a loop over the charge chunks in each step. In the radiation damage signal formation, there is an additional loop that runs twice over the signal formation code, once for electrons and once for holes which is necessary as they experience trapping at different rates. The various contributions to the motion of the charge carrier before being trapped are shown in Fig. 2.26. The radiation damage simulation is then comprised of a few steps that include all the considerations mentioned in the previous section and is also summarised in the pseudocode snippet 3 in Appendix A.1.

1. Starting from the charge carrier's position in the bulk simulated by the Energy Deposition Tool, the simulation looks up the time it would take the charge carrier to reach its respective electrode. This is done by using the maps that take into account the modified electric field shape and Lorentz angle for the given operating conditions and fluence, which are both taken from the conditions database for this run and module.
2. Throw a random number and calculate the trapping time, as was detailed in section 2.4.1.
3. If the trapping time is less than the time taken to reach the electrode, the charge carrier will be trapped and so enter the trapping loop. Else, drift and diffuse the charge carrier to its respective electrode, as was done in the original Tool.
4. For the trapped charge the next step is to look up the trap depth based on the charge carrier's initial depth in the bulk and the trapping time. The trap depth is independent of the lateral drift from the Lorentz angle which is also looked-up.
5. A random number is thrown to apply diffusion to the drifted charge carrier.

6. The final charge position is then used to find the weighting potential value at this position. The weighting potential at the initial position in the bulk is also found.
7. Next the nearest neighbour pixels are looped over and the weighting potential value at the centroid of the neighbor pixel is looked up for the start and end position of the charge carrier in each.
8. The induced charge is then calculated for the 3×3 pixel region and the unsmeared factor is again applied before adding the induced charge to the chargedDiodes container.
9. Any areas with a low electric field, such as the IBL slim edges can be applied as a scaling factor to the induced charge in the same way it was applied in the original chargeTools.

Although there are many additional steps in simulating the signal formation, many of them involve looking up values in pre-calculated maps, for example the weighting potential. However there are also two additional loops: one to treat electrons and holes separately and one to find the nearest neighbour induced charge. As this code is still in development, performance tests within Athena have not yet been run but care is being taken to minimise the number of calculations needed, for example by incorporating the drift due to the Lorentz angle into the distance maps and only entering the simulation loop when a charge chunk is trapped.

The radiation damage code can be included in any of the current signal formation tools by adding a flag that allows the radiation damage simulation portion to be accessed.

2.5 Conclusion and outlook

In this project, a radiation damage simulation under development in the standalone Geant4 simulation framework, Allpix, was implemented in Athena. This involved identifying the correct stage of Athena simulation (digitization), where the simulation should be placed in the flow of digitization (subChargesTool/signalFormation), and implementing the model. In doing so, various physical and technical aspects of the simulation were also improved, including a restructure of the Charge Tools in the package.

However this is only the first step in the Athena implementation, which, at the time of writing must be validated within the ATLAS framework both using low-level objects (pixel cluster shapes, charge) and at a higher level (effect on space-points, tracks). As for the overall structure of the Athena implementation, at the time of writing it is unclear where the maps used in the simulation will be stored in the final release. Following the work detailed here, the author investigated ways to interface the Tools with a database of run conditions such that the maps could be loaded by a service rather than as a run-time parameter. Ideally, the majority of the maps should be stored in a database of run conditions as their values rely heavily on the detector operation (bias voltage, temperature history) and how much fluence it has been exposed to. The Signal Formation Tool should not decide which fluence and map to access, rather this should be decided by some external service given the module ID and Run number associated with the simulated hit. The weighting potential is only dependent on sensor geometry and so should not be kept in the conditions database but rather the service or tool that picks the module technology, for example planar or 3D, IBL sensor or outer layer sensor.

Aside from outlining the radiation damage simulation, **Paper III** also details validation of various aspects of the Allpix simulation with respect to data collected by the ATLAS experiment. This included comparing the cluster shapes at specific angles between data and simulation, validating the modeling of the Lorentz angle, and comparing the electric field strength simulated in TCAD to that measured in irradiated sensors.

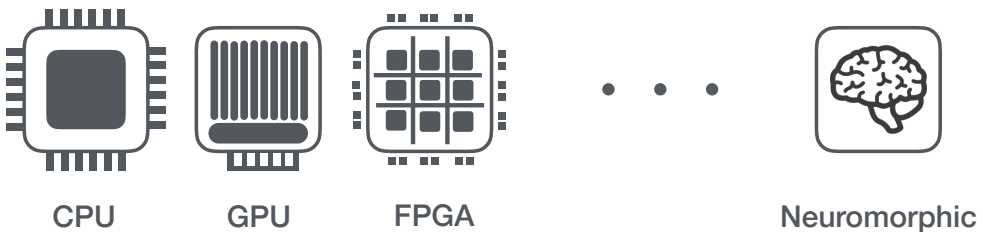
Chapter 3

Kalman filter in IBM's TrueNorth

With future colliders attaining unprecedented luminosities and data rates, new methods of computation and new architectures on which to perform them are being explored. With the advent of a post-Moore's law field of computation the range of technology and infrastructure available in the next decade is less easy to foresee and thus plan for. Future commodity computing architectures may take a form that the current HEP infrastructure must adapt to, so early explorations into the portability of common algorithms to novel architectures is necessary. An algorithm key to track reconstruction in multiple HEP experiments over the past 50 years is the Kalman filter, see its usage in ATLAS track reconstruction in Chapter 1. As such the implementation of this algorithm on novel architectures is a promising step to understanding the usability of such a device and projecting the feasibility of its inclusion in future HEP computing.

In recent years, neuromorphic architectures have become more prominent in commercial companies such as IBM. In 2008 IBM entered a Defense Advanced Research Projects Agency (DARPA) project called SyNAPSE that involved the design of a scalable, low power neuromorphic architecture and in 2015 released the first version of this chip to select institutes for testing and development. Neuromorphic architectures are, quite literally, brain or neuron-inspired computers.

In this chapter the implementation of a Kalman filter in IBM's neuromorphic architecture, TrueNorth, will be explored and its performance used to evaluate its usage in HEP computing. A brief introduction to the concept of neuromorphic computing will be covered in section 3.1, IBM's TrueNorth chip will be introduced in section 3.2, and the implementation of a Kalman filter in section 3.3 with the results of that implementation discussed in section 3.4. Some of the results discussed in this chapter are summarised in **Paper IV**.



3.1 Neuromorphic computing

Neuromorphic computing is computing based on our knowledge of how the brain processes and distributes information. The brain processes data in small elements called neurons. Complex processes in the brain arise from the amalgamation of these simple units. Efforts in brain inspired computing broadly fall into two categories¹:

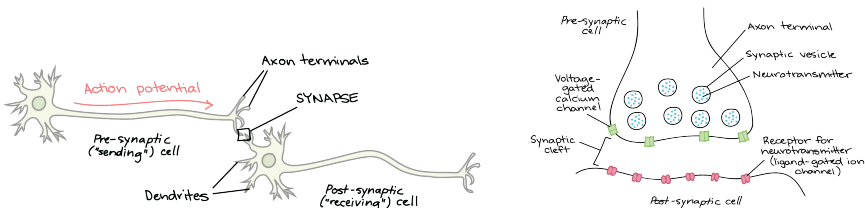
1. For the understanding of biological information processing through synthesis. This is a top-down approach to understanding the intricacies of the brain. International collaborations such as the Human Brain Project (HBP) have been formed with the explicit purpose of top-down modelling.
2. To produce future computing architectures based on biological information processing.

The latter approach seeks to take what mechanisms are understood about the brain and apply them to conventional computing methods. Despite our knowledge of how the brain operates, on the whole, still being largely incomplete, there are some key attributes that make neuromorphic computing desirable.

Brains are incredibly efficient in terms of energy consumed vs. number of operations performed when compared with a standard CPU, both in terms of computations per second and because of the information density of those computations [110]. The brain volume has an optimal wire-to-processor ratio [111]. It is fault tolerant - we know that large parts of the brain are regenerated everyday with no loss in performance [112] and that the brain exhibits learning. The latter design approach seeks to pick and choose the attributes of brain computation that cause these desirable features and use them to build a computer with deterministic output. To understand the fundamentals of neuromorphic computing then, is to understand some basic principles about the fundamental building block in neuroscience: the neuron.

3.1.1 Biological neurons

The neuron is a cell that can be deconstructed into the following substructures. Dendrites are projections from the body of the neuron that act like wires, transmitting electrochemical charge-carriers towards the soma, the cell body. The soma is also connected to an axon, a long chain of specialised cells that terminate in a series of protrusions known as the axon terminal, see Fig. 3.1a. Neurons form long chains in the body that communicate both electrically, with



(a) A very simplified cartoon of the biological neuron and its connectivity.

(b) A simplified cartoon of the synapse and post-synaptic cell.

Figure 3.1: Simplified cartoons of the neuron and synapse [113].

direct conducting channels between the cells, and chemically via synapses: chemical bridges between the axon terminal, in the pre-synaptic neuron, and the dendrites, in the post-synaptic

¹From discussion with Karlheinz Meier, March 2017.

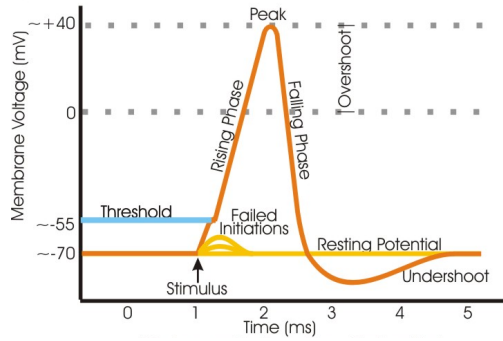


Figure 3.2: A schematic of the action potential, or spike, emitted by the soma.

neuron, see Fig. 3.1b. The consequence of synaptic transmission is a change in the properties of the cell membrane of the post-synaptic neuron. The transmission can either increase or decrease the potential of the exterior of the cell membrane with respect to the interior of the cell.

Since a single neuron is likely connected to multiple axons, its membrane potential is determined by the integration of all of these inhibitory and excitatory post-synaptic potential changes. The membrane potential has an associated time constant which describes the rate at which charge leaks away till the membrane reaches some resting potential. When the membrane potential reaches a specific threshold in the neuron, an avalanche of ion channels will open creating what is known as an action potential, where the neuron potential rises rapidly, falls off with a characteristic time, and undershoots as is seen in Fig. 3.2. This produces a current pulse, also called a spike, which is propagated along the axon to the synapses.

The axon is an insulated conductor, whose diameter can also vary and affect the transmission of electrical impulses produced by the soma. A popular model for the relationship between the membrane potential and the post-synaptic current is the Hodgkins-Huxley model [114] that considers each of the three dominant ion channels at the synapses as parallel RC circuits where the semi-permeable cell membrane is represented in each case with a resistor-capacitor pair

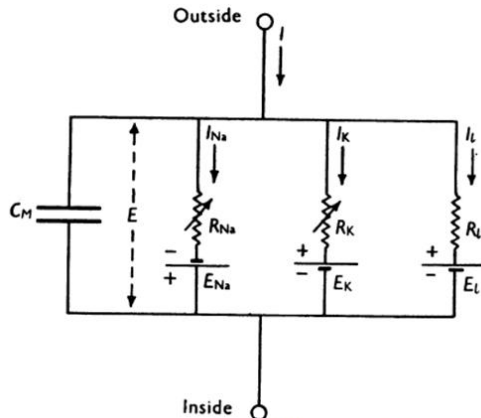


Figure 1: Circuit Model of an Axon [1]

Figure 3.3: The Hodgkins-Huxley model of the current passing across a semi-permeable cell membrane by three ion channels.

with a probabilistic model assigned to each, see Fig. 3.3. This results in a series of differential equations with 20 tuneable parameters.

Simplified versions of the Hodgkins-Huxley model exist, for example the leaky-integrate-and-fire model, which replaces the separate treatment of the ion channels by a single RC circuit with no probabilistic switching.

3.1.2 Neuromorphic networks and artificial neural networks

Neuromorphic architectures take key characteristics of the spiking biological-neuron model and create an electrical simulation of that model. Neuromorphic devices have a vast morphology but they share the following attributes:

- **Spikes to encode data** Unlike conventional computing architectures that encode instructions and data in binary words, neuromorphic devices encode data in temporally separated spikes.
- **Distributed, simple, parallel processing** All neuromorphic architectures share the concept of a neuron performing computations on inputs. The computations it performs have little variety and so complexity arises from the connectivity between the neurons and the number of tuneable parameters each neuron has.
- **Thresholding a weighted sum** Neurons calculate a weighted sum of their inputs, store that value as a membrane (or neuron) potential, and check it against a set threshold. Some models include a temporal element where the membrane potential can leak away at a given rate.
- **Spiking when over threshold** If a neuron potential is over threshold, the neuron emits a spike of its own and the neuron potential is reset. In some of the more biologically plausible architectures the reset includes a refractory period during which the neuron cannot fire again.

These devices differ from a traditional von Neumann architecture in that they have no complex instruction set, but only a model file to update the neuron parameters and connectivity of the neurons. Instead of a single cluster of device memory and a CPU, each neuron has its own memory and IO blocks such that each neuron can operate independently for truly parallel computation. In many ways neuromorphic architectures more closely resemble Artificial Neural Networks (ANN) but there are some important differences. The concept of thresholding in a neuromorphic neuron could be analogous to the activation function in an ANN neuron, aside from the concept of the neuron potential retaining values for multiple timesteps. Neuromorphic neurons that seek to closely resemble biological neurons usually contain some analogue circuitry to better model the shaping of the post-synaptic spikes as they enter the membrane which can lead to more complex behaviour than is typically exhibited by ANN neurons. Neuromorphic neurons may also be programmed to exhibit time-dependent plasticity over short or long time periods, in which neurons fire more or less frequently depending on how often spikes are sent to them. This concept has been shown to translate well to ANN vision tasks [115].

3.2 IBM's TrueNorth

The architecture considered in this study is the first version of IBM Research's fully digital neuromorphic chip, TrueNorth. Whilst many other hardware realisations of neuromorphic architectures focus on simulating biological neuron behaviour, TrueNorth's focus is on low-power, event-driven computation. The chip was developed in response to a DARPA project called SyNAPSE [116]. SyNAPSE challenged participants to meet a set of criteria including spike rate encoding, hardwired crossbars with programmable parameters, operating speed > 10 Hz, and that any hardware produced would, at scale, consume $< 10^{-12}$ J per synaptic operation. With the focus on scalability, low power, and processing elements operating with a slow clock, IBM designed TrueNorth.

In 2015 IBM Research released TrueNorth NS1e evaluation boards, see Fig. 3.4, to a select

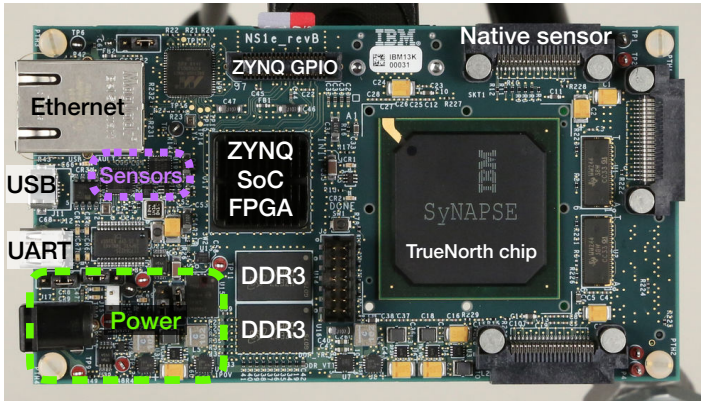


Figure 3.4: The NS1e TrueNorth evaluation board.

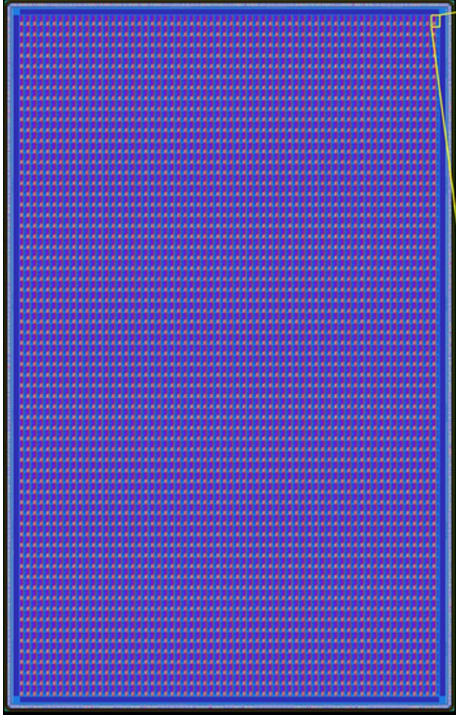
number of institutes in the USA, including Lawrence Berkeley National Laboratory, to develop algorithms, perform alpha testing on the software released, and to use the supplied TrueNorth testboards. The NS1e boards contain a single TrueNorth chip, a ZYNQ System-on-Chip ARM core and FPGA, which runs linux and interfaces directly with TrueNorth, and optional native sensor IO to send spikes directly into the chip. The software released at the time consisted of a Hardware Description Language (HDL) called the Corelet Programming Environment (CPE), a TrueNorth Neuro-Synaptic Chip Simulator (NSCS), and an extension to the popular Caffe² deep learning framework called Tea. Tea supported the constraints of the TrueNorth architecture and allowed users to train feed-forward neural networks and utilise other standard layer designs in Caffe's library. Early in 2016, support for Tea was discontinued and IBM released a framework known as the Energy-Efficient Deep Neuromorphic network (Eedn) [117]. Eedn shifted TrueNorth's focus from feed-forward and deep neural nets to convolutional and all recent publications affiliated with the project have focused on vision tasks.

The rest of this chapter will detail the implementation of a simple tracking algorithm in TrueNorth and evaluate its suitability and ease-of-use in future collider experiments.

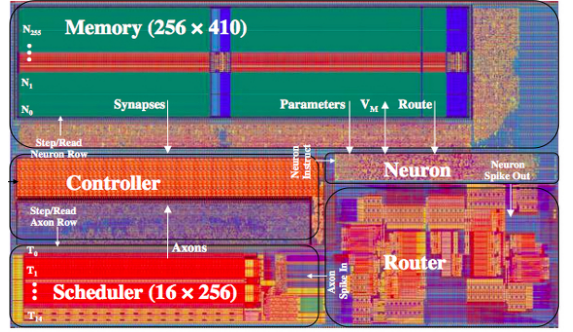
²Available at <http://caffe.berkeleyvision.org/>.

3.2.1 Top level design: topology, timing, and packets

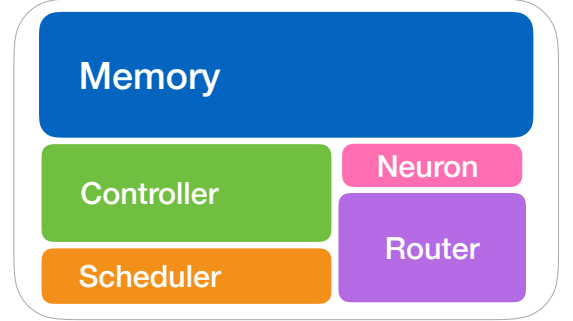
The TrueNorth architecture is comprised of parallel processing units called cores, of which there are 4,096 on a single chip. TrueNorth's cores are tiled in a 64×64 grid, as seen in Fig. 3.5a, where each core can be connected to any other core on the chip by a two-dimensional mesh network. Packets enter the chip via one of the IO ports and hop from one core to another until they reach the correct one. Every core contains a router which communicates data packets between cores and delivers them to the core's scheduler when they reach the correct destination.



(a) The TrueNorth chip top-level layout. Each highlighted rectangle on the chip is one of 4,096 cores.



(b) An annotated layout of the TrueNorth core.



(c) A schematic of the top-level blocks in a TrueNorth core, where the size of each block is proportional to the area taken up in the real layout.

Figure 3.5: IBM's TrueNorth chip layout and core detail.

TrueNorth cores contain a router, scheduler, token controller, a core memory block, and the computational element: a neuron block, as seen in Fig. 3.5b and 3.5c. The need for both a scheduler and token controller is due to TrueNorth's mixed Synchronous-Asynchronous design. Instead of a fast global clock net, which consumes idle power, communication between cores on the chip happens asynchronously with a four-phase handshake protocol. Packets passed from the router to the scheduler are stored in the scheduler's 16×256 SRAM. The dimensions of the SRAM correspond to the 256 inputs to the programmable core³ and the 4 bits of delay that can be programmed into a packet and thus inputs to the router are stored in the corresponding cell of the SRAM. The scheduler, too, is event-driven and queues packets as they arrive from the router asynchronously.

³Programming the core is covered in Section 3.2.2

The actual computations in the core happen in the 256 neurons in the neuron block and are synchronous. The clock supplied to each neuron in the neuron block is generated by the token controller and so is local to each core. The clock pulses for performing computation are only generated if the input to the neuron has a packet waiting and if the connection to that input is active, thus although the logic in a neuron is synchronous its usage is event driven.

Once the token controller has established that a neuron will be active it sends the exact number of clock pulses needed to perform the computation, it requests the neuron model parameters stored in the core memory block and combines them with the input packets stored in the scheduler and sends both of these to the neuron, it takes any outputs produced by the neuron block at the end of the computation and sends them to the router and updates the neuron parameters in the core memory block.

Although each core performs a combination of synchronous and asynchronous tasks, these are all initiated by the arrival of a signal edge in the token controller called a tick. The sending of data packets from a source core to its destination, along with any processing done by the core upon the packet's arrival, must be completed before the arrival of the next tick. As such it is not called a clock in the traditional sense but rather a global synchronization point. If communication or computation is not completed within one tick a number of errors will be flagged by the chip. These will be addressed in Section 3.4.4.

3.2.2 The programmable core: neurons, axons, and the crossbar

The elements of TrueNorth discussed up to this point are necessary to understand the chip's unique constraints, but are also fundamental to the chip's architecture and cannot be changed by the user. In this section the programmable elements of the TrueNorth core will be discussed.

Alluding to the biological counterparts they imitate, the inputs, connections within the core, and computational elements are known as axons, synapses, and neurons, as shown in Fig. 3.1a. A single neuron can have multiple axons connected to it, as shown in Fig. 3.6a and likewise a single axon can be connected to multiple neurons as in Fig. 3.6b. Combining these two concepts results in a crossbar: a connectivity map of the axons and neurons where connections are denoted by the presence of a binary synapse, as shown in Fig. 3.6c.

Axons, aside from being inputs to the core, have two additional properties. They can buffer an incoming packet for up to 15 ticks before it is sent to the crossbar, this is the 16×256 SRAM described in section 3.2.1, and they can be assigned a label, $G \in \{0, 1, 2, 3\}$. Both of these values can be set by the user. The axon label is stored, along with synaptic connectivity and neuron parameters, in the core memory. The axon label is used by the neuron to indicate how packets arriving from an axon with that label should be handled, thus a single neuron can handle packets up to four ways.

Once a neuron receives and processes an incoming packet from an axon, it can emit one of its own. A neuron's output can be fed back to a new axon in either the same core or another core but may only be connected to a single axon. Splitting the output of a neuron over multiple axons will be covered in section 3.2.5. The user sets both the crossbar, by indicating whether a synapse is present or not, and neuron output connectivity.

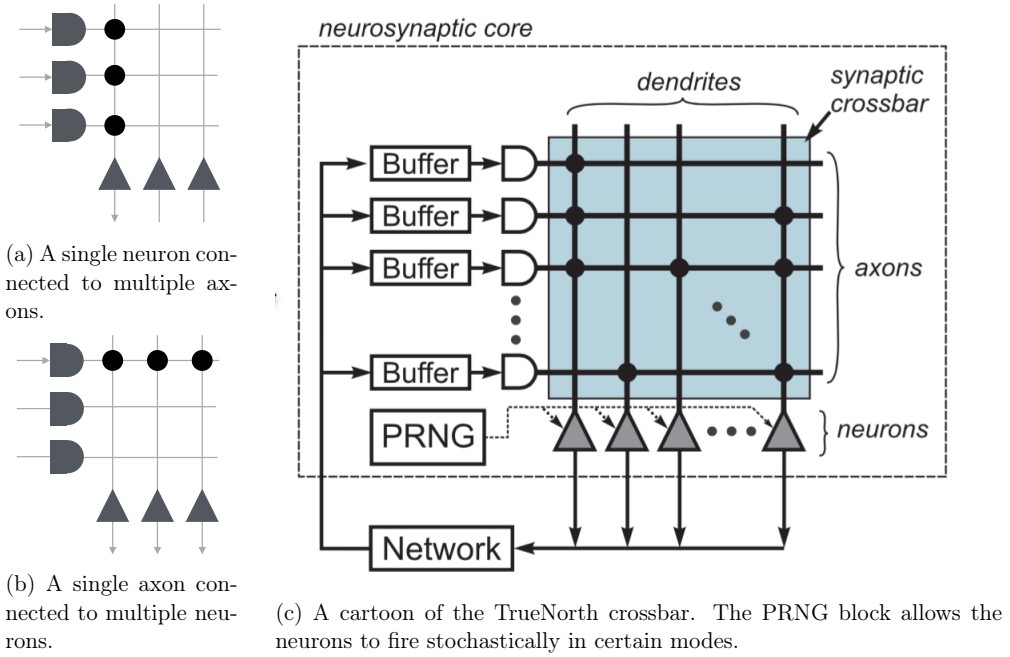
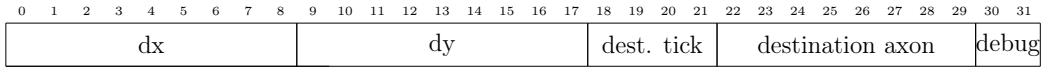


Figure 3.6: TrueNorth elements and the crossbar. Axons, inputs, are represented as semicircles. Neurons, which perform computations and output the result, are represented as triangles. Synapses, which indicate if a neuron and axon are connected, are shown as black circles.

3.2.3 Spikes and the neuron update equation

TrueNorth communicates and processes data packets called spikes. A spike is a 32 bit packet containing routing information to indicate which core it belongs in, a destination tick which indicates when it should be injected into the crossbar, a destination axon in the crossbar, and some debugging information, see Fig. 3.7. Once a spike is injected into the network it first



Removed by routing network when handed to scheduler

Figure 3.7: Composition of a 32 bit TrueNorth spike, as processed by the routers in each core. When the spike reaches its intended core the first 18 bits are stripped and only the tick, axon, and debug bits remain.

moves in the dx direction till it reaches the router in its destination column. At this point the first 9 bits are stripped away and the spike continues moving in the dy direction of the core network till it reaches the router in its destination core, where the second 9 bits are stripped away. The remaining 14 bit packet is sent into the scheduler to be processed by the core, as described in the previous section.

Once a spike packet has been routed through the crossbar to the correct neuron, it is weighted according to the label of the axon it came from, since there are 4 possible axon labels a single neuron can store up to 4 weights. The weighted input is added to the neuron potential, V_j , where $j \in [1, 256]$ indexes neurons on the core. Once all inputs to the neuron have been

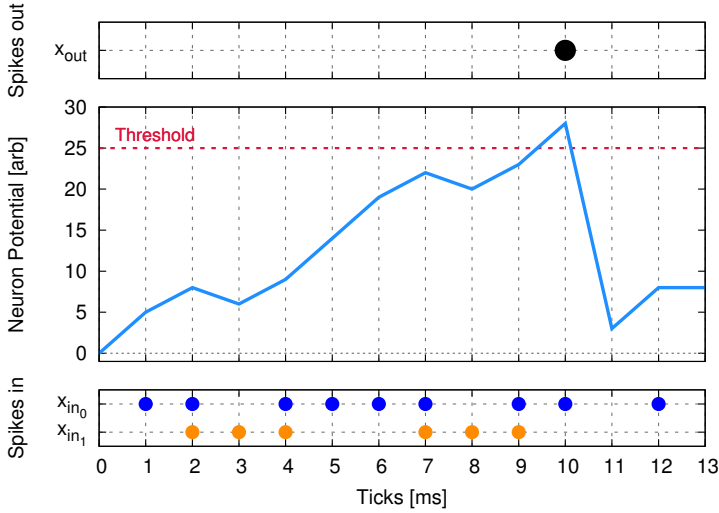


Figure 3.8: The evolution of a neuron potential over multiple ticks. The neuron has two inputs which are weighted differently: one increases the value of the neuron potential and the other reduces it. After ten ticks $V_j > \alpha_j$ and the neuron emits a spike. V_j is then reset by $V_j - \alpha_j$.

processed in this way V_j is changed by a fixed bias, or leak, and then compared to a threshold, α_j .

If $V_j \geq \alpha_j$ the neuron emits a spike, this is more commonly referred to as the neuron firing. Once the neuron has fired V_j will be changed according to a programmed reset scheme. The neuron potential is also compared to a negative threshold, however if below this threshold it will not spike but will be also be reset. The process described here is illustrated in Fig. 3.8. A cartoon of the neuron circuit is shown in Fig. 3.9. The sequence of operations above constitute the neuron update, which is described more completely by the equation 3.1. The parameters in this equation are described in table 3.1. Both this update equation and table are a subset of the complete neuron update, which has additional parameters to control leak and stochasticity that are not included in the Kalman filter, this is described in appendix B.1.

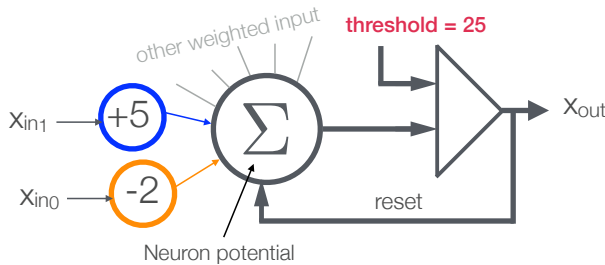


Figure 3.9: A cartoon of the neuron circuit. Input spikes are weighted according to the label assigned to the axon they entered the crossbar on and summed. This value is then checked against a fixed threshold and if greater than or equal to threshold emits a spike and is reset according to some scheme.

The neuron potential update at each timestep:

$$V_j(t) = V_j(t-1) + \sum_{i=1}^{256} A_i(t) \times w_{i,j} \times s_j^{G_i} \quad (3.1a)$$

The potential, V_j is capped at a ceiling and floor defined as:

$$\begin{aligned} \text{ceiling} &= \begin{cases} \alpha_j + M_j, & \text{if } \gamma_j = 2 \\ 393216, & \text{otherwise} \end{cases} \\ \text{floor} &= \begin{cases} -\beta_j - M_j, & \text{if } \gamma_j = 2 \\ -393216, & \text{otherwise} \end{cases} \end{aligned} \quad (3.1b)$$

The neuron will cross the positive threshold and fire if:

$$V_j > (\alpha_j + \eta_j) \quad (3.1c)$$

The neuron will cross the negative threshold if:

$$V_j(t) < -\beta_j \quad (3.1d)$$

If V_j crosses the positive threshold it will be reset according to:

$$V_j(t) = \begin{cases} V_{\text{rst}_j}, & \text{if } \gamma_j = 0 \\ V_j(t) - \alpha_j - \eta_j, & \text{if } \gamma_j = 1 \\ V_j(t), & \text{if } \gamma_j = 2 \end{cases} \quad (3.1e)$$

If V_j crosses the negative threshold it will be reset according to:

$$V_j(t) = \begin{cases} -V_{\text{rst}_j}, & \text{if } \gamma_j = 0 \\ V_j(t) + \beta_j + \eta_j, & \text{if } \gamma_j = 1 \\ V_j(t), & \text{if } \gamma_j = 2 \end{cases} \quad (3.1f)$$

scope	symbol	range	description
Basic neuron update	i, j	$[1, 256]$	Axon and neuron indices, respectively
	$V_j(t)$	$[-393216, 393216]$	Neuron potential at tick t
	$A_i(t)$	$[0, 1]$	Presence of spike in axon i at tick t
	$w_{i,j}$	$[0, 1]$	Presence of synapse connecting axon i and neuron j
	G_i	$[1, 4]$	Axon label
	$s_j^{G_i}$	$[0, 2^8 - 1]$	The weight assigned to spikes incident on neuron j from axons with label G_i
Threshold and reset	α_j	$[0, 2^{18} - 1]$	Positive threshold. If V_j exceeds this the neuron will fire and be reset.
	β_j	$[0, 2^{18} - 1]$	Negative threshold. If V_j drops below this the neuron will not fire but will be reset.
	M_j	$[0, 2^{18} - 1]$	Acts as a mask for the reset stochasticity and also extends the ceiling for the special reset case where $\gamma_j = 2$
	γ_j	$[0, 2]$	Membrane potential reset mode
	η_j	$[0, 2^{18} - 1]$	Masked stochastic threshold.
	V_{rst_j}	$[0, 2^{15} - 1]$	Reset voltage in special reset mode

Table 3.1: A partial list of the changeable neuron parameters in TrueNorth. The complete list is available in table B.1.

The entirety of the spike packet is information about when and where the packet should be injected into the network, it contains no additional information and so is essentially a digital 1. Digital spiking is a key difference between TrueNorth and other neuromorphic or ANN chips and affects the way data is encoded and processed. For example non-spiking architectures usually encode data in binary and can represent non-integer values using floating or fixed point representations. The neuron has no native way of handling binary encoding so unary is used. In unary, the ordering of ones and zeros in a data word does not change the meaning of the word. For example, in binary a word with 3 bits can encode 2^3 different values whereas in unary only 3 different values can be encoded. To illustrate this consider a word with two 0's and one 1.

In binary:

$$100 \neq 010 \neq 001$$

However, in unary:

$$100 = 010 = 001$$

Encoding in unary can be done by defining a data word in terms of a fixed number of ticks

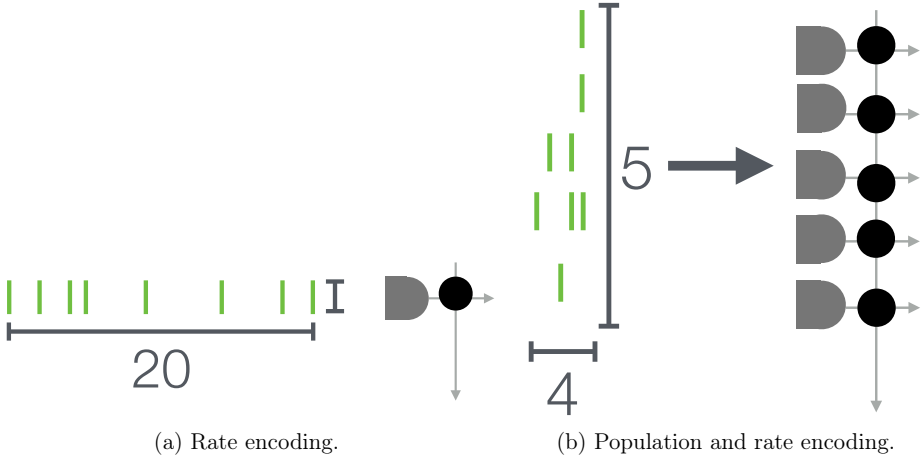


Figure 3.10: Rate and population encoding with axons. The same value (0.4) is encoding in a 20-spike word using 8 spikes. In rate-only encoding this takes 20 ticks and 1 axon to complete. In rate-and-population encoding this takes 4 ticks and 5 axons.

(the above example would take 3 ticks to encode) which is known as rate encoding or by a combination of rate and population encoding. In this context population encoding describes splitting up a word over multiple axons, as is shown in Fig. 3.10b. Splitting up a word over multiple axons allows it to be processed in less ticks but takes up more of the crossbar. This becomes an important consideration when many inputs are needed to the same crossbar and hence there is limited space to speed up a computation. This topic will be explored further in section 3.4.

3.2.4 Arithmetic in TrueNorth

With an understanding of the basic functionality of neurons and connectivity, a TrueNorth user can begin to program more complex operations. This section will outline how to implement basic arithmetic concepts in TrueNorth.

Adding two spike trains together simply involves connecting them to the same neuron on the crossbar, as in Fig. 3.11a. This is because weighted addition is intrinsic to the neuron functionality. Likewise, subtracting spike trains uses the same concept but assigns a different axon type to change the sign of the weight, as is illustrated in Fig. 3.11b. Multiplying incoming spikes by a weight and dividing that weight by comparing it to a threshold are also operations intrinsic to the neuron.

To multiply a spike train by an integer, the neuron weight alone can be applied. For example, if a spike train was connected to a neuron with $s_j^{G_i} = 2$ and $\alpha_j = 1$ then the output of that neuron would spike twice the rate of the input train, so if there were 20 spikes in an input word of there would be 40 spikes in the outgoing word. Setting $s_j^{G_i} > \alpha_j$ multiplies the incoming spike rate, and setting $s_j^{G_i} < \alpha_j$, divides the incoming spike rate. To multiply an incoming spike train by a numerical weight, the weight must be represented as a rational number of the form:

$$\text{weight} = \frac{s_j^{G_i}}{\alpha_j} \quad (3.3)$$

The neuron weight register is only 8-bits, so the precision of weights that can be represented is

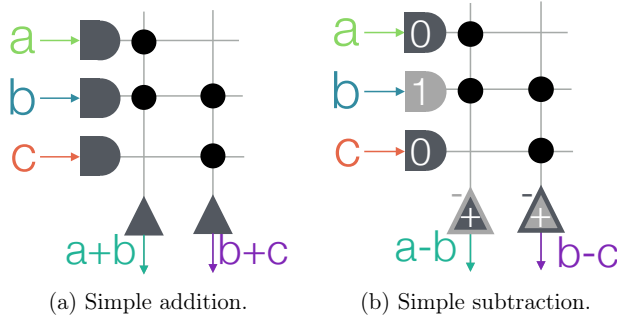


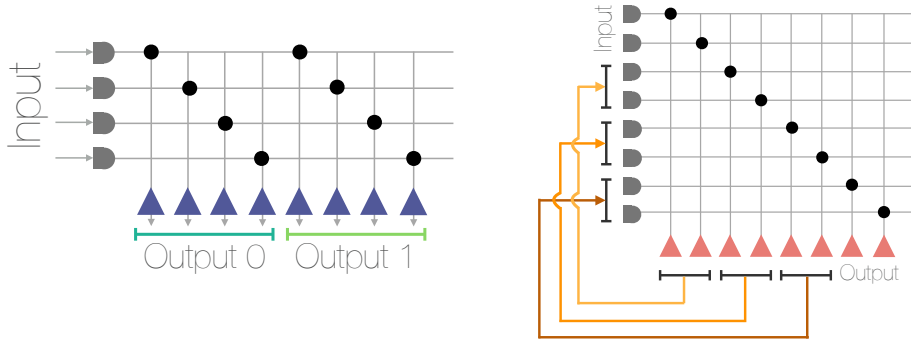
Figure 3.11: Crossbars showing basic signed addition of multiple spike trains in a core. In the case of subtraction two axon labels are used so the neuron knows which inputs to add and which to subtract. Note that each neuron can interpret these labels differently.

limited. The range of weights can be extended by duplicating a spike train over multiple axons and adding those inputs together, as is shown in Fig. 3.12a.

Increasing the range of the threshold, or denominator of the rational weight representation requires splitting it into two factors: threshold = factor a $\times$ factor b, and using two neurons. The first neuron has whatever neuron weights are needed assigned and α_j = factor a. The first neuron's output is fed into a second neuron with $s_j^{G_i} = 1$ and α_j = factor b, as shown in Fig. 3.12b. Increasing the range of the threshold, however, is rarely needed since it is already an 18 bit register. A combination of these two methods can be used for high precision weights.



Figure 3.12: Extending the range of values the neuron weight and threshold is able to represent is possible to increase representation precision. In the case of extending the neuron weight range, at least two axon types are needed. The first (in red in Fig. (a)) has the max value of 255, the second (in grey) will have the value (weight % 255).



(a) A simple 1:2 splitter corelet for a 4-pin Connector. As with the delay corelet, the crossbar is a simple diagonal and adding additional splits would just require duplicating this pattern along additional neurons.

(b) A simple delay corelet for a 2-pin Connector. With each recurrent connection the input can be delayed up to 15 ticks, so the min. and max. programmable delay for this corelet is 4 ticks and 46 ticks, respectively.

Figure 3.13: A couple of simple corelet concepts.

3.2.5 Key corelet concepts

With many ways to organise cores and parameters in TrueNorth, IBM has defined a workflow for programming and deploying models. The connectivity, axon and neuron parameters, and crossbar in TrueNorth are programmed using a Hardware Description Language (HDL) called the Corelet Programming Environment (CPE), which is built and written using Matlab. The CPE defines several objects including the neuron, core, and a top level container called a corelet. Corelet objects can contain crossbars for multiple cores and define multiple neuron types. They can also be built from other corelets, so a single top-level corelet can be defined from multiple smaller ones, see Fig. 3.14. A couple of popular corelets in the corelet library, supplied with CPE, include a splitter, that takes a corelet with an output Connector of size n and produces x copies, also of size n , as shown by the crossbar in Fig. 3.13a. Another popular corelet is the delay corelet, which simply delays some input for t ticks, see Fig. 3.13b. These corelets have in common that they compensate for limitations in the TrueNorth architecture: the delay corelet compensates for the maximum delay per input being only 15 ticks, and the splitter corelet allows for neuron outputs to be sent to multiple destinations, something also not supported by the native architecture.

One of the complications of programming a corelet is that the inputs and outputs to the object cannot simply be assigned to individual pins directly but must be interpreted through a class called the Connector. A Connector must be defined to have a finite size. This becomes a problem when only a few of the pins in the Connector are desired in some cases but all are needed in others, there is no option in the CPE that allows pins to remain unused or unconnected for a specific instance.

For example, consider the case where a corelet's output Connector was size four. Later in the design process it was decided that the output of that corelet should be used in two places in the design, but only two of the pins in the Connector were needed in the second instance. Because only a partial number of pins in the Connector are needed the user would either have

⁴Outputting a different sized Connector than was input is not supported by the corelet library version of the splitter, so a custom one had to be written since the need for this functionality arose often.

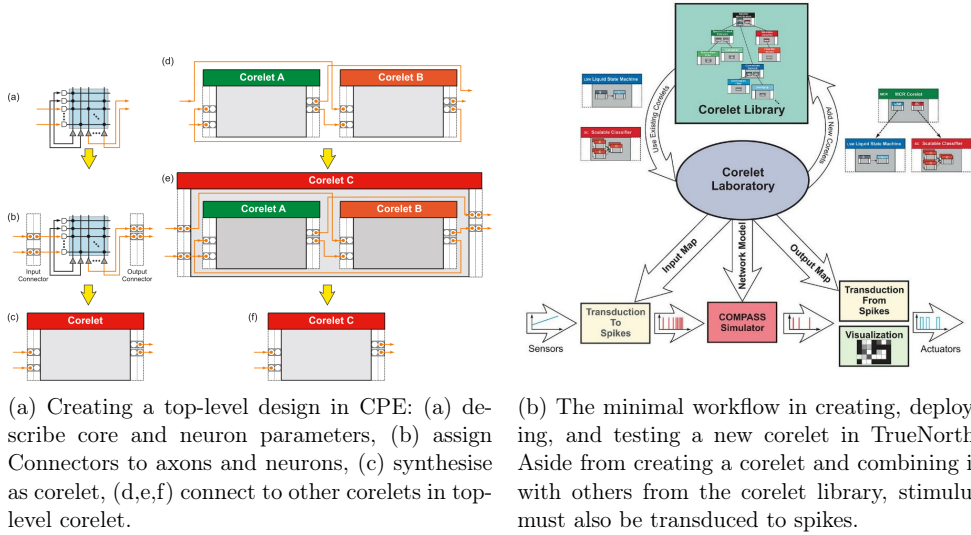


Figure 3.14: Corelet creation and deployment workflows.

to decide to reserve two extra axons in the recipient corelet and connect the size four Connector, effectively wasting two axons, or would have to write a custom splitter class to split a Connector of size four into two Connectors of size two⁴. Naturally this second solution involves using an additional corelet, and hence core, and will add latency to the design - all of which could easily be averted if the user could connect pins directly, rather than having to go through Connectors. At the time of writing it is not possible to force multiple corelets, which take up less than one core, to be deployed on a single core. In other words there is no way to optimise core usage by placing multiple corelets on a single core. This is a feature that IBM have spoken of including in future updates to CPE. Thus, at the time of writing, the number of cores used in a design is greater than or equal to the number of corelets in the design.

Once a corelet has been described it is synthesised into a model file that can be run by the TrueNorth NSCS or loaded onto the chip. When the model is deployed it is not possible to change any of the parameters or connectivity - so no online-learning is possible with this architecture. The significance of this will become apparent in section 3.3.2.

3.3 Implementation and test setup

In Chapter 1.3, the Kalman filter was introduced and its use in track reconstruction in the ATLAS experiment was described. In this section the implementation of a subset of the Kalman filter equations in the TrueNorth architecture will be detailed.

3.3.1 The Kalman filter and the steady-state Kalman filter

The Kalman filter [118] is a recursive set of equations that produces an estimator of the state of a discrete-time controlled process in a way that minimises the mean squared error of the estimator with respect to the true state. The Kalman filter operates on a stream of noisy measurements and essentially ‘filters’ out the noise to recover the underlying state. The state,

$x \in \mathbb{R}$, is defined according to the linear stochastic difference equation:

$$x_t = \mathbf{F}x_{t-1} + \mathbf{D}u_{t-1} + w_{t-1} \quad (3.4)$$

Where $\mathbf{F}$ is the state transition matrix which relates the state at the previous timestep, $t - 1$, to the state at the current timestep t , u is the control input, $\mathbf{D}$ relates the control input to the state, and w is a white, wide-sense stationary, normally-distributed stochastic noise called the process noise. The state is not measured directly but rather through a device that has some additional properties, thus measurements, y , of the state are given by:

$$y_t = \mathbf{H}x_{t-1} + v_{t-1} \quad (3.5)$$

Where $\mathbf{H}$ is the measurement matrix that relates the measurement to the state and v is also a white, wide-sense stationary, normally-distributed stochastic noise assumed to be independent of w , called the measurement noise. The covariance matrices of w and v are defined to be Q and R , respectively. Unlike similar state estimators such as the Weiner algorithm, the Kalman filter is light-weight to implement as it recursively incorporates a weighted history of the entire state at each time step, instead of requiring that all states in the history are retained and operated on together. Because the Kalman filter incorporates all past estimates of the state, and due to the statistical properties of the estimators, the Kalman filter estimators are more precise than predictions made using a single measurement.

The Kalman filter has its origins in an iterative linear least-squares estimator of the form:

$$S^2 = \sum_{i=1}^T \frac{(y_i - \mathbf{H}\hat{x}_i)^2}{\sigma_i^2} \quad (3.6)$$

Where y , x , and $\mathbf{H}$ are defined as before, S is the mean square error of the estimator and is distributed normally, $\sigma^2 = v$, and is the variance of the measurement, or noise on the measurement. This equation can be extended to n dimensions by writing it in matrix form as:

$$S^2 = (\vec{y} - \mathbf{H}\hat{x})^\top \mathbf{R}^{-1}(\vec{y} - \mathbf{H}\hat{x}) \quad (3.7)$$

where y, x are now assumed to be $n \times 1$ vectors and $\mathbf{R}$ is an $n \times n$ matrix of the covariance of the measurement noise. To solve for the estimator, $\hat{x}$, S^2 is minimised with respect to x :

$$\begin{aligned} \frac{dS^2}{dx} &= -2\mathbf{H}^\top \mathbf{R}^{-1}(y - \mathbf{H}\hat{x}) = 0 \\ \therefore \hat{x} &= \left(\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H} \right)^{-1} \mathbf{H}^\top \mathbf{R}^{-1} y \end{aligned} \quad (3.8)$$

Which can be simplified further by defining:

$$\mathbf{P} = \text{covar}[x] = \left(\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H} \right)^{-1} \quad (3.9)$$

and therefore equation 3.8 becomes:

$$\hat{x} = \mathbf{P} \mathbf{H}^\top \mathbf{R}^{-1} y \quad (3.10)$$

This formulation assumes the entirety of measurements have been made *a priori* and, depending on how many timesteps are used, can involve a large matrix inversion. To break the equation into smaller steps, and allow it to be evaluated iteratively, it can be rewritten as:

$$\hat{x}_t = \hat{x}_{t-1} + \mathbf{P}_t \mathbf{H}_t^\top \mathbf{R}_t^{-1} (y_t - \mathbf{H}_t \hat{x}_{t-1}) \quad (3.11)$$

with the state covariance given by:

$$\mathbf{P}_t = \left(\mathbf{P}_{t-1}^{-1} + \mathbf{H}_t^\top \mathbf{R}_t^{-1} \mathbf{H}_t \right)^{-1} \quad (3.12)$$

The part of the matrix multiplication in equation 3.11 which involves $\mathbf{P}$ is still computationally intensive at each iteration and can be reduced using the following identity:

$$\mathbf{K}_t = \left(\mathbf{P}_{t-1}^{-1} + \mathbf{H}_t^\top \mathbf{R}_t^{-1} \mathbf{H}_t \right)^{-1} \mathbf{H}_t^\top \mathbf{R}_t^{-1} = \mathbf{P}_{t-1} \mathbf{H}_t^\top \left(\mathbf{R}_t + \mathbf{H}_t \mathbf{P}_{t-1} \mathbf{H}_t^\top \right)^{-1} \quad (3.13)$$

where K is known as the Kalman gain matrix which minimises the *a posteriori* error covariance, $\mathbf{P}_t$ [119]. The Kalman gain is then incorporated into equation 3.11 and 3.12 so that they becomes:

$$\begin{aligned} \hat{x}_t &= \hat{x}_{t-1} + \mathbf{K}_t (y_t - \mathbf{H}_t \hat{x}_{t-1}) \\ \mathbf{P}_t &= (\mathbb{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t-1} \end{aligned} \quad (3.14)$$

Another way to think of this is that as the measurement noise covariance, $\mathbf{R}$, in the Kalman gain expression, tends to zero the measurement y is trusted more whilst the predicted state $\mathbf{H}\hat{x}$ is trusted less and vice versa. The equations in 3.14 constitute the discrete Kalman filter measurement update equations and combine a new measurement with an *a priori* state estimate to obtain an improved *a posteriori* measurement. Not included in these equations is the fundamental *prediction* step which is given by the linear stochastic difference equation 3.4 at the beginning of this section and the covariance prediction equation [120]:

$$\mathbf{P}_t^- = \mathbf{F} \mathbf{P}_{t-1} \mathbf{F}^\top + \mathbf{Q} \quad (3.15)$$

To distinguish between the update equations that use the state and covariance estimate from the previous timestep (equations 3.15 and 3.4), and the measurement update equations derived in this section (equations 3.13 and 3.14), the time update states are denoted with a ‘-’, as seen in equation 3.15. Thus the full discrete Kalman filter update equation is summarised below:

$$\hat{x}_t^- = \mathbf{F} \hat{x}_{t-1} + \mathbf{D} u_{t-1} + w_{t-1} \quad (3.16a)$$

$$\mathbf{P}_t^- = \mathbf{F} \mathbf{P}_{t-1} \mathbf{F}^\top + \mathbf{Q} \quad (3.16b)$$

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^\top \left(\mathbf{R}_t + \mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^\top \right)^{-1} \quad (3.16c)$$

$$\hat{x}_t = \hat{x}_t^- + \mathbf{K}_t (y_t - \mathbf{H}_t \hat{x}_t^-) \quad (3.16d)$$

$$\mathbf{P}_t = (\mathbb{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_t^- \quad (3.16e)$$

A point of interest for this set of equations is that since the covariance of all state estimator and measurement is wide-sense stationary, and white with zero mean, over time they will average out to be zero. This will quickly cause the estimation error covariance, $\mathbf{P}$, and the Kalman gain, $\mathbf{K}$, to stabilise and remain constant, hence both of these values can be pre-computed or trained depending in the application.

A subset of the Kalman filter estimator is the steady-state Kalman filter which assumes $\mathbf{P}$ and $\mathbf{K}$ to be constant. If we additionally simplify the system to omit any control input (so no $\mathbf{D}u$ in equation 3.16a), and set the measurement matrix to the identity (so $\mathbf{H} = \mathbb{I}$) the Kalman filter equations reduce to:

$$\hat{x}_t^- = \mathbf{F} \hat{x}_{t-1} + w_{t-1} \quad (3.17a)$$

$$\hat{x}_t = \hat{x}_t^- + \mathbf{K}_t (y_t - \hat{x}_{t-1}) \quad (3.17b)$$

where the state estimate covariance, $\mathbf{P}$, and Kalman gain, $\mathbf{K}$, are not shown as they are constants in time. This equation further simplifies to:

$$\begin{aligned}\hat{x}_t &= (\mathbf{F} - \mathbf{K}) \hat{x}_{t-1} + \mathbf{K} \vec{y}_t \\ &= \mathbf{A} \hat{x}_{t-1} + \mathbf{B} \vec{y}_t\end{aligned}\tag{3.18}$$

In this work, $\mathbf{A}$ is defined as the matrix multiplying the recurrent state estimate and $\mathbf{B}$ is the matrix multiplying the observations/measurements of the system. In this case, $\mathbf{B}$ is equal to the Kalman gain, $\mathbf{K}$. Since steady-state dynamics are assumed, both the $\mathbf{A}$ and $\mathbf{B}$ matrices can be pre-calculated, or trained.

Attempts were made to implement a full Kalman filter in the TrueNorth architecture in a couple of ways. There are many existing methods to implement a Kalman filter in an artificial neural network (ANN). These perform the covariance matrix and Kalman gain matrix updates using recurrency and/or backpropagation depending on the type of Kalman filter used [121]. This method could not be implemented in TrueNorth as no neuron parameters or connectivity can be changed during runtime and additionally any training of an ANN-style network in TrueNorth is extensive. For such a method to be implemented the chip would have to be stopped and started continuously to update the parameters, resynthesise the model file, and load it onto the chip.

Another method considered was that of the Neural Engineering Framework [122] which takes linear control systems and implements them in populations of neurons. In this definition, a neuron population refers to many neurons performing a very similar operation with slight variations. Unfortunately part of the formulation required that a non-linear leak be implemented in the neurons and that the post-synaptic current have a finite width, thus altering the value arriving at the neuron: essentially, this method does not work for digital spikes and so had to be abandoned.

After exploring tracking algorithm implementations in existing neuromorphic architectures we came to the conclusion that TrueNorth's neuron model is too simple to emulate their approaches. Instead, using the inherent ability of a neuron to do linear algebra, the Kalman filter was formulated using explicit matrix multiplication and addition. The following sections will describe this implementation.

3.3.2 Implementation and design

A steady-state Kalman filter was implemented in TrueNorth using linear algebra and was broke down into two processes: matrix element multiplication and addition which are easily implemented as was seen in section 3.2.4. For example, in a 2-dimensional state estimate produced

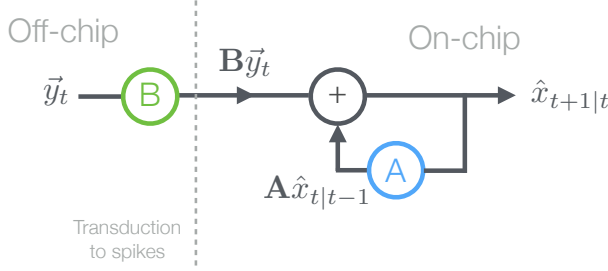
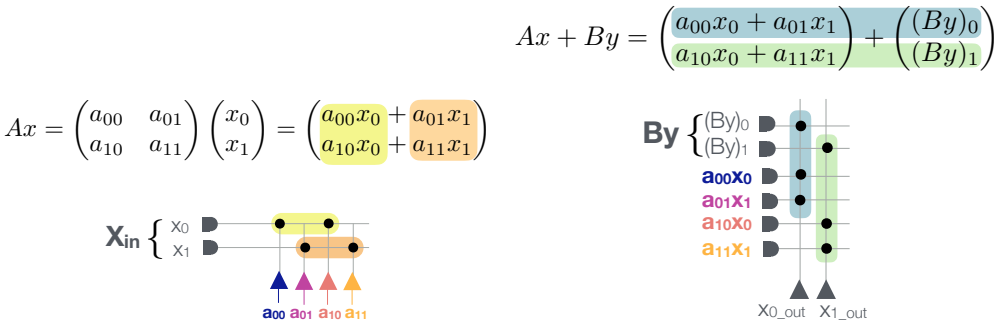


Figure 3.15: A circuit diagram for the steady-state Kalman filter implementation on TrueNorth, from equation 3.18.

by a Kalman filter, equation 3.18 would involve multiplying a (2×1) vector by a (2×2) matrix and adding it to the noisy measurement. These steps are broken down visually into crossbars as shown in Fig. 3.16.

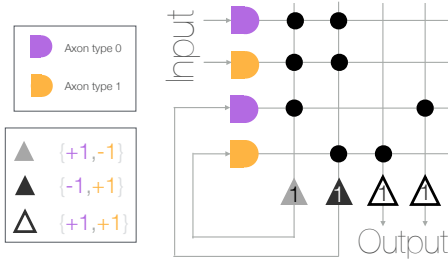
The system measurements, the input to the Kalman filter, are multiplied by matrix $\mathbf{B}$, however this step is not performed on the chip, as is shown in Fig. 3.15. This is because the measurements have to be pre-scaled to lie in the range $[-1, 1]$, off-chip, before they can be transduced to spikes, so an additional multiplication is simple to perform. It is possible to port this operation onto the chip but it will introduce both additional latency and take up additional cores to perform. It is easy to see how this formulation could be extended for higher-dimensional states. For example, the multiplication block in Fig. 3.16a would require n axons, and n^2 neurons and synapses, for an n -dimensional state. The addition block in Fig. 3.16b would require $n + n^2$ axons and synapses, and n neurons, for an n -dimensional state.



(a) Each matrix element is represented by a different neuron. This can be thought of as the columns of the resultant matrix corresponding to rows in the crossbar, as highlighted in the image.

(b) In the case of addition, values that should be added together are sent to the same neuron. Here the rows of the matrix correspond to columns in the crossbar.

Figure 3.16: The Kalman filter state estimate weighting and addition to the measurement step as shown in crossbars for a two-dimensional system. The outputs of the coloured neurons in Fig. (a) are sent to the corresponding axons in Fig. (b).



$V_{\text{Neuron \#0}}$	$V_{\text{Neuron \#1}}$	$V_{\text{Neuron \#2}}$	$V_{\text{Neuron \#3}}$	Spikes in	Spikes out
$0 + 1 - 1 = 0$	$0 - 1 + 1 = 0$	0	0	1,1	0,0
$0 - 1 = -1$	$0 + 1 - 1 = 0$	0	0	0,1	0,0
$-1 + 1 - 1 = 0$	$0 - 1 = -1$	0	$0 + 1 = 0$	1,0	0,1
$0 + 1 - 1 = 0$	$-1 + 1 - 1 = -1$	$0 + 1 = 0$	0	1,0	1,0
0	$-1 + 1 = 0$	$0 + 1 = 0$	0	0,0	1,0
Total spikes				+3, -2	$2 - 1 = 1$

(a) An example of the paired signed neuron logic.

(b) A table showing the evolution of the paired neuron V_j for various spike patterns.

Figure 3.17: Signed addition reset logic, if one of the neurons in the pair fires it will add +1 to the neuron potential of its pair, thus ensuring that the sum of the neuron potentials in the pair remain 0.

Positive and negative values

With the basic building blocks of the steady-state Kalman filter implementation formed, an additional consideration is how to deal with signed numbers. Whilst it may be possible to transform dynamical system measurements such that $y \in [0, 1]$, this does not guarantee that the corresponding **A** and **B** matrices will also be positive-valued, thus inputs are pre-scaled to the range $y \in [-1, 1]$. Standard binary words carry an extra bit for encoding the sign of the value and the logic that processes that word is designed to expect a sign-bit at a specific place in the packet. There is no such mechanism available to encode a sign bit in a spike train explicitly since spikes are integrated in time. One could envision an additional channel in the core reserved solely for a sign spike that would gate some special logic to treat the respective word as signed but the special logic quickly becomes complicated and was not pursued. Instead, positive and negative values are handled in parallel and combined in the addition core, as in equation 3.19:

$$\begin{aligned}
 V_j^+(t) &= V_j^+(t-1) + \sum_{i=1}^{256} A_i(t) \times w_{i,j} \times \begin{cases} +1, & \text{if } G_i = 0 \\ -1, & \text{if } G_i = 1 \end{cases} \\
 V_{k \neq j}^-(t) &= V_{k \neq j}^-(t-1) + \sum_{i=1}^{256} A_i(t) \times w_{i,k} \times \begin{cases} -1, & \text{if } G_i = 0 \\ +1, & \text{if } G_i = 1 \end{cases}
 \end{aligned} \tag{3.19}$$

These equations are very similar to the standard neuron update equation but here explicit neurons are assigned to encode the positive and negative values and at each tick this pair will have potentials with equal and opposite-sign; in other words the sum of the neuron potentials in the pair will always be zero. However if one of the neurons spikes, the pair becomes unbalanced, thus additional logic is used to reset the neuron that did not fire, this is illustrated by Fig. 3.17. This additional logic, for an n -dimensional system, will add $2n$ axons and synapses, and n neurons to a signed addition crossbar. It should be noted that the signed addition crossbar neurons should have a negative threshold set to at least 2α for the addition to be performed correctly.

The final crossbar of the n -dimensional serial Kalman filter, with this signed addition concept included, uses $2n(n+3)$ axons, $2n(n+2)$ neurons and $4n(2n+1)$ synapses. The complete crossbar for the case $n=2$ is shown in Fig. 3.18.

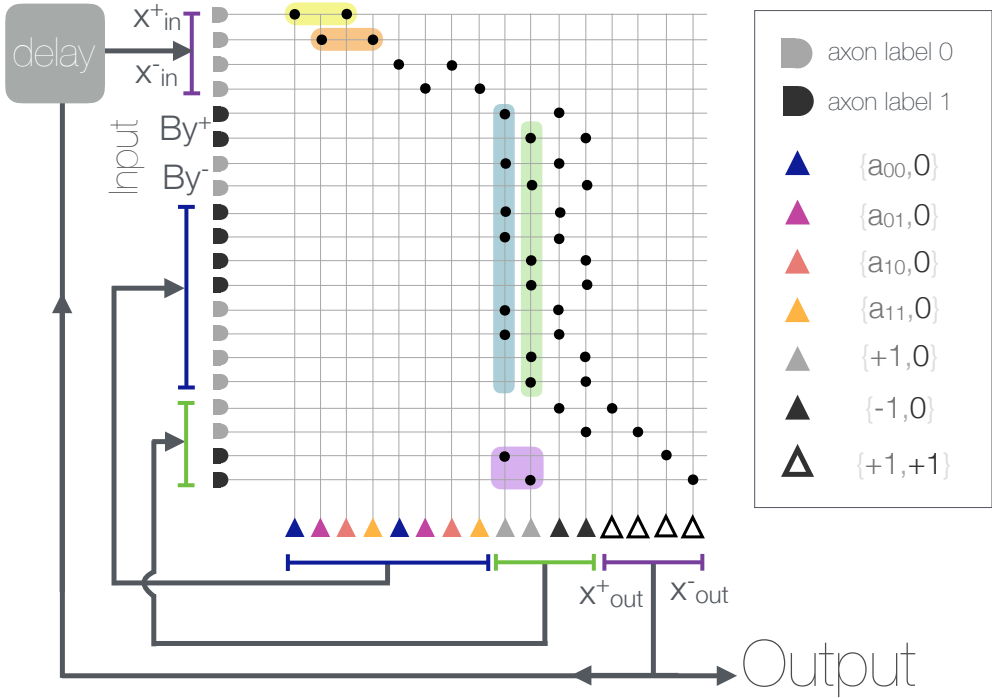


Figure 3.18: The complete 2-dimensional Kalman filter crossbar. Not shown explicitly is the delay corelet to correctly time in the state estimate with the measurements, as this depends on the size of the word used to encode the data. The split at the output is done using a splitter corelet, also not shown here. The highlighted portions show the multiplication, addition, and signed-reset logic for the positive channel described previously.

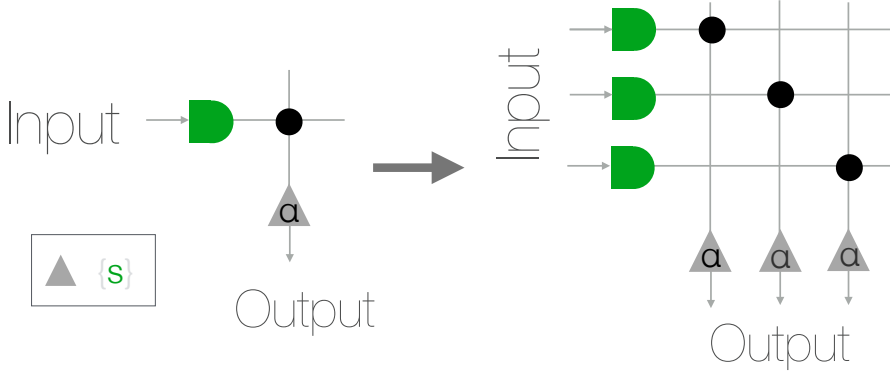


Figure 3.19: Naively, to parallelise a corelet simply replace each axon-neuron pair with b axons and neurons and a diagonal crossbar.

Serial and parallel implementations

Up till this point the crossbar logic described will only work with a purely ‘serial’ encoding, a rate encoding, as discussed in section 3.2.3, where a single axon-neuron pair will process a word and any variations in data are encoded in time-separated spikes. This section will explain the modifications necessary to the Kalman filter crossbars so that they can process data encoded across both time and multiple axons, this will be known as the parallel encoding.

First, the serial word must be parallelised by dividing it over a number of axons and neurons, hereafter called the blocksize, denoted by b . For example if a data word takes 32 ticks in the serial encoding, then it could be split into a parallel block of 8 axons and take 4 ticks to encode. A data word is thus defined as:

$$\text{data word size} = \text{window of time [ticks]} \times \text{blocksize [axons]} \quad (3.20)$$

The naive assumption, when formulating the parallel representation, would be to replace each synapse in the serial crossbar with a diagonal crossbar, and replicate each neuron and axon b times, see Fig. 3.19. To understand why this doesn’t work, first consider what is required of the parallel representation. For the equivalent stimulus, it must produce the same number of spikes for a given word. In the serial case this word is defined as taking a fixed number of ticks and the output is over 1 neuron. The equivalent parallel case must output the same number of spikes, when summed over the b neurons and the reduced number of ticks. Consider table 3.2 for a word of size 6 ticks, multiplying an incoming train by $\frac{s}{\alpha} = \frac{2}{3}$ and compare it to the output of the naive parallel solution in table 3.3 for $b = 3$ and 2 ticks. From the first word it is clear that the number of spikes emitted and, in fact, the total number of spikes is not the same. This is a problem for the Kalman formulation which is explicitly calculating the state estimate and for an accurate output relies on the multiplication and timing being accurate. The naive formulation does not work because not all neurons in the block receive the same input and so a different number of spikes are integrated over, which cannot be rectified without enforcing a binary-like encoding. Instead, consider a setup where every neuron receives the same input, i.e. with a full crossbar, as shown in Fig. 3.20. Although every neuron is receiving the same input, it is receiving b times too much. To deal with this, a linear ladder of thresholds is used. An analogous method is used in a linear ADC, where an input voltage is sampled over progressively increasing resistors. Using the crossbar and neuron parameters shown in Fig. 3.20 then, the stimulus applied in the serial and naive-parallel case produces an output as shown in table 3.4.

word	spikes in	spikes out	V_0	word	spikes in	spikes out	V_0
1	1	0	$0 + 2 = 2$	3	1	0	2
	1	1	$2 + 2 - 3 = 1$		0	0	2
	1	1	$1 + 2 - 3 = 0$		0	0	2
	1	0	$0 + 2 = 2$		1	1	1
	1	1	$2 + 2 - 3 = 1$		0	0	1
	1	1	$1 + 2 - 3 = 0$		0	0	1
2	1	0	2	4	0	0	1
	0	0	2		1	1	0
	1	1	1		0	0	0
	0	0	1		1	0	2
	1	1	0		0	0	2
	0	0	0		1	1	1
Total spikes in: 14, total spikes out: 9							

Table 3.2: The response of a single neuron to a serial, or rate-encoded, stream of 6-tick words. The neuron is set up to have $s = 2$, $\alpha = 3$, $\gamma = 1$, i.e. if the neuron fires V is reduced linearly by α . See table 3.1 for more details.

word	spikes in	spikes out	$V_{0,1,2}$
1	1, 1, 1	0, 0, 0	2, 2, 2
	1, 1, 1	1, 1, 1	1, 1, 1
2	1, 0, 1	1, 0, 1	0, 1, 0
	0, 1, 0	0, 1, 0	0, 0, 0
3	1, 0, 0	0, 0, 0	2, 0, 0
	1, 0, 0	1, 0, 0	1, 0, 0
4	0, 1, 0	0, 0, 0	1, 2, 0
	1, 0, 1	1, 0, 0	0, 2, 2
Total spikes in: 14, total spikes out: 8			

Table 3.3: The response of a block of three neurons to a parallel stream of 2-tick words. The neurons all have $s = 2$, $\alpha = 3$, $\gamma = 1$, i.e. if the neuron fires V is reduced linearly by α - the same as in table 3.2.

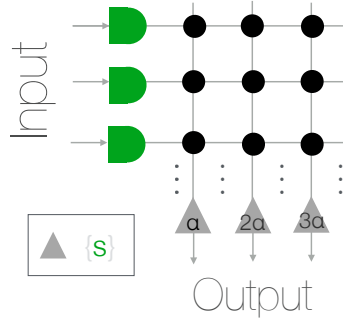


Figure 3.20: A full crossbar is used at the input axons so all neurons in the block receive the same input. The neurons are then setup to implement the same weights for the input axons but have a linearly increasing threshold from left to right. There is some additional logic, not shown in this diagram, to implement the correct reset across the neuron block.

Here, the total number of spikes emitted after each word is the same, so this is the axon/neuron setup that will be used in the parallel implementation.

An additional caveat is that for all the neurons in the block to sample the spikes correctly at each tick, they must all be reset by the same amount: by $(\alpha_0 \times \text{\#neurons fired in block})$, i.e. by α_0 times how many neurons fired in that timestep.

The first approach to the reset was to use neurons set up with $\gamma = 2$, which is the ‘no-self-reset’ mode for the neuron: when the neuron spikes its potential remains the same. The crossbar for this setup is shown in Fig. 3.21. The first part of the crossbar, spanning neurons and axons $\{1 - 3\}$, is the input block that sends all input spikes to all neurons in the block. The second full crossbar, spanning axons $\{4 - 6\}$ and neurons $\{1 - 3\}$, is the reset crossbar. Consider the case that the first neuron in the block fires, from Fig. 3.21 a spike will be queued in axon 4. In the next tick, the spike will be sent to neurons 1, 2, 3, 4. In neurons $\{1 - 3\}$ the spikes will be weighted by $-\alpha_0$, thus resetting all the neurons. The same principle holds if more neurons in the block fire.

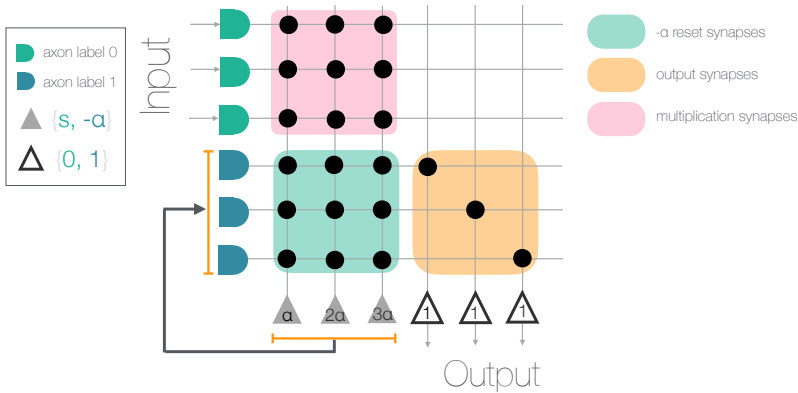


Figure 3.21: The parallel multiplication crossbar for $\gamma = 2$. This is no-self-reset-when-fire mode for the neurons, so if the neurons fire the spikes are both fed back to reset the neuron potential across the block and output from the core. This setup does not work as intended due to the cap on the neuron potential ceiling, see text for details.

word	spikes in	spikes out	$V_{0,1,2}$
1	1, 1, 1	1, 1, 0	0, 0, 0
	1, 1, 1	1, 1, 0	0, 0, 0
2	1, 0, 1	1, 0, 0	1, 1, 1
	0, 1, 0	1, 0, 0	0, 0, 0
3	1, 0, 0	0, 0, 0	2, 2, 2
	1, 0, 0	1, 0, 0	1, 1, 1
4	0, 1, 0	1, 0, 0	0, 0, 0
	1, 0, 1	1, 0, 0	1, 1, 1
Total spikes in: 14, total spikes out: 9			

Table 3.4: The response of a block of three neurons to a parallel stream of 2-tick words. The neurons all have $s = 2$. The thresholds are: $\alpha_0 = 3$, $\alpha_1 = 6$, $\alpha_2 = 9$. The neurons are set up such that all are reset by the same amount: ($\alpha_0 \times \#$ neurons fired in block).

The reason this setup doesn't work is because of the ceiling applied to the potential in the case that $\gamma = 2$, from equation 3.1b:

$$V_j = \begin{cases} \alpha_j + M_j, & \text{if } \gamma_j = 2 \\ 393216, & \text{otherwise} \end{cases}$$

The parameter M_j can be set to be non-zero to increase the ceiling but doing so causes stochastic behaviour in the neuron. Capping V_j in this way causes a bias and does not produce the correct output, thus the reset portion of the crossbar and neuron parameters had to be redesigned.

If, instead of the neurons in the block having $\gamma = 2$, all have $\gamma = 1$ then the setup shown in Fig. 3.22 could be used. Although the crossbar looks the same as for the previous case, note that two extra neuron settings and 2 extra axon labels used. In this setup if a neuron fires then its potential is reset linearly by the neuron itself. The spike emitted is then fed back into the

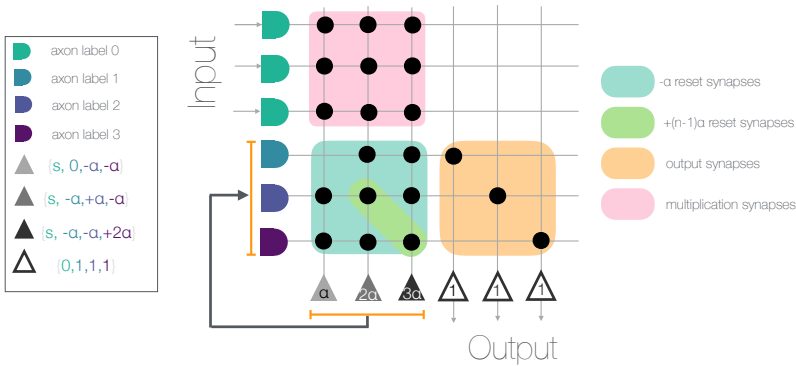


Figure 3.22: The parallel multiplication crossbar for $\gamma = 1$. In this mode neurons that fire reset themselves but are also reset by other neurons in the block firing. Unfortunately this particular crossbar setup only works for $b < 4$, see text for details.

crossbar to change the potential of the neurons that didn't fire. Note that all the block-neurons, except the first, also have a feedback loop that is connected to them which assigns a weight of $(j-1)\alpha_j$, this is to compensate for their potential already having been reduced by neurons to the left of them in the block.

However, although this crossbar produces the correct behaviour, it ceases to work for a blocksize > 4 as there are only 4 axon labels. If axons are desired to increase the range of the neuron weight this crossbar can only be used for blocksize < 3 . This constrain on the blocksize is compensated for by introducing an additional neuron-axon pair for each neuron in the block except the first, as seen in Fig. 3.23. This introduces additional complexity to the crossbar and almost doubles the number of neurons and axons but it does produce the correct behaviour.

One final problem is the limited size of the neuron weight register. The reset crossbar requires weights of up to $|(b-1)\alpha_0|$ to reset the last neuron on the block and the neuron weight register range is only $[-(2^8-1), 2^8-1]$, whereas the threshold range is $[-(2^{18}-1), 2^{18}-1]$. There are several ways to deal with this constraint. The first is making all $\alpha_j \leq 255$, but this lends little precision to the input weight applied at the top of the crossbar, recalling that a spike is weighted by $\frac{s_j}{\alpha_j}$ for each neuron.

Another, more practical solution, is to use the neuron weight extension method outlined in section 3.2.4. For example, if $\alpha_0 = 300$, then the axons feeding back spikes from neurons with $\alpha = 2 \times 300 = 600$ would be duplicated once: with the first axon having $s = 255$, and the second having $s = 45$, totaling $s = 300$, as desired. Then for the axons feeding back spikes from neurons with $\alpha = 3 \times 300 = 900$, the axons would correspond to weights $\{255, 255, 100\} = 600$ and so three axons would be needed, etc. The variable extending the range of thresholds that can be reset is p , i.e. if $p = 2$, there will be twice as many axons in the reset portion of the parallel multiplication crossbar, which extends the thresholds that can be used in the neurons up to $2 \times 255 = 510$.

Once again, however, the limited number of axon labels become a problem. Consider that

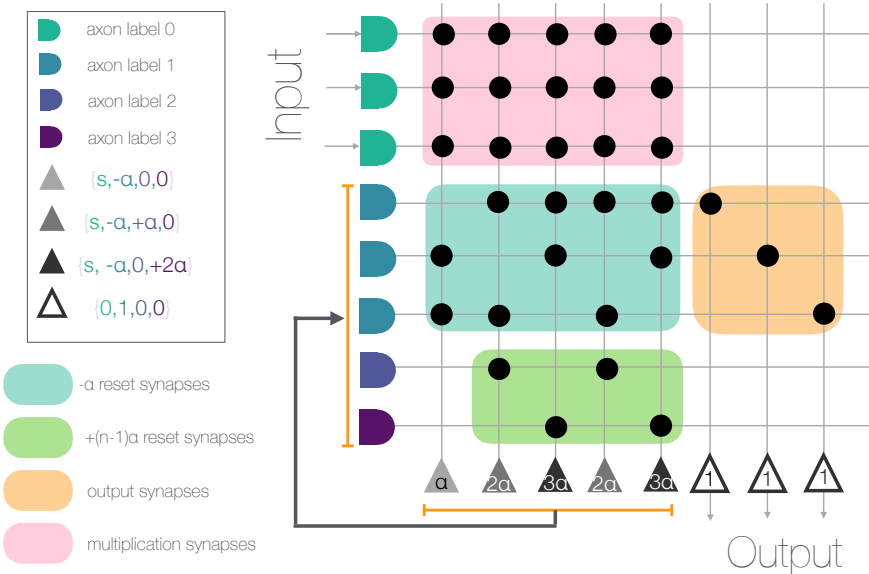


Figure 3.23: The parallel multiplication crossbar for $\gamma = 1$. In this mode, neurons that fire reset themselves but are also reset by other neurons in the block firing. This setup works for $b \geq 2$ and allows an extra axon label to extend the neuron weight precision. See text for details.

the weights at the input of the block likely need to have their range extended: thus at least 2 axon labels are needed to do this. An additional axon label is needed for the subtraction portion of the reset-crossbar, which leaves one remaining axon label for the addition portion of the reset crossbar whereas two are needed to extend the reset weight. The only way around this, if one still desires to use a neuron threshold > 255 , is to have $\alpha \% 255 = 0$, i.e. have the threshold be a multiple of 255.

Naturally this additional constraint reduces the precision with which the non-spiking weights can be represented but presents a viable way to program the parallel encoding and so was the setup used for the testing of the parallel Kalman filter. The implications of this will be explored in section 3.4.4.

The parallel corelet design, for multiplying an input spike train, takes up an entire core and replaces one axon-neuron pair in the original serial crossbar: so how does this design scale for the rest of the Kalman filter?

The $\mathbf{Ax}$ portion of the crossbar can be evaluated on an element-by-element basis, in other words replacing each axon-neuron pair in the multiplication portion of the crossbar (highlighted in yellow and orange in Fig. 3.18) will not affect the calculation.

Addition does not mix the state dimensions so it can be split, one dimension at a time, into same-sign addition and addition that combines positive and negative channels. These corelets are also built from the design concepts used in parallel multiplication corelet. For example, the same-sign addition corelet for $b = 3$ is shown in Fig. 3.24, which utilises the same reset scheme as the parallel multiplier but has no need to duplicate axons to extend the weight or threshold range of the neurons since they are both < 255 . This corelet scales, for adding two

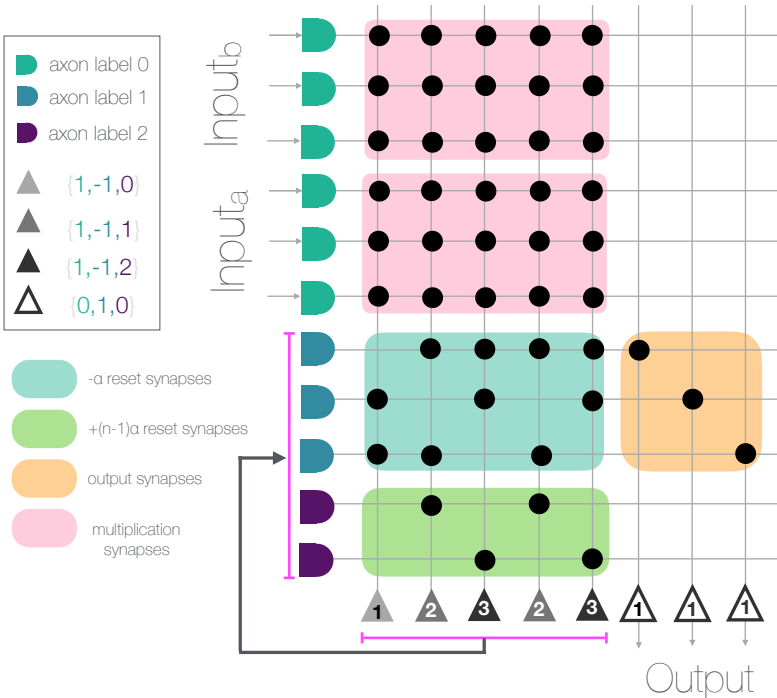
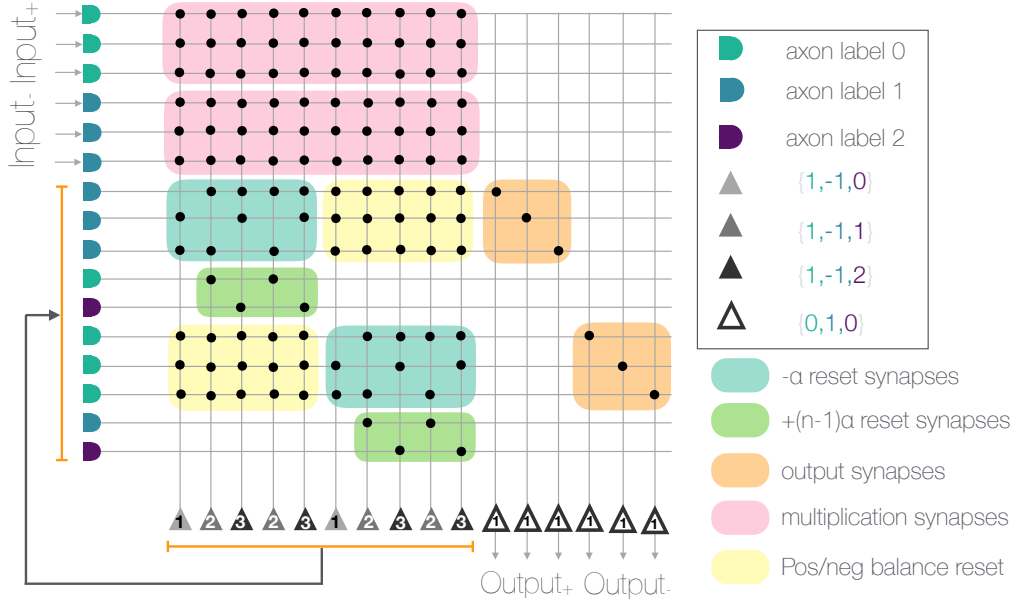


Figure 3.24: The parallel same-sign addition corelet for $b = 3$.

Figure 3.25: The parallel different-sign addition corelet for $b = 3$.

values together as:

$$\begin{aligned} \text{Axons: } & 4b - 1 \\ \text{Neurons: } & 3b - 1 \end{aligned} \tag{3.21}$$

So clearly the number of axons is the limiting factor for same-sign parallel addition.

In the case of combining positive and negative trains, or different-sign parallel addition, the crossbar is largely the same as for the same-sign addition corelet but with the extra addition-reset logic that was described in section 3.3.2. For example, the different-sign parallel addition corelet for $b = 3$ is shown in Fig. 3.25. The template for the axon-neuron usage in the general parallel case is shown in Fig. 3.25, from which it is clear that this corelet scales like:

$$\text{Axons, neurons: } 6b - 2 \tag{3.22}$$

So this corelet is always square. It is important to note that, because of the limited size of the crossbars in TrueNorth, this version of the Kalman filter cannot be put on a single core and requires multiple splitters to use. The number of splitters, as with the number of duplications of the various corelets needed, also scales with the dimensionality of the state being tracked. The limitations of the core size on the maximum achievable accuracy and speedup of the parallel version of the Kalman filter is explored in the next section.

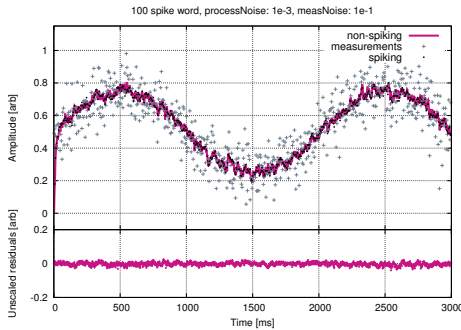
3.3.3 Test setup and simulation

To test these implementations of a spiking Kalman filter a variety of setups were used. The spiking implementation was written as a corelet and deployed on both the TrueNorth chip and NSCS simulator. Multiple tests were run to check that the output of the two matched exactly and this was always found to be the case, therefore many of the tests were run on NSCS as it is faster than running on TrueNorth with a 1 kHz tick.

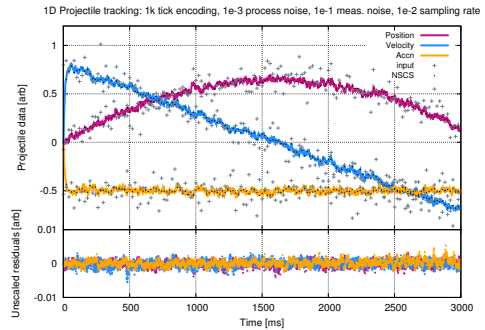
During runtime, there is no way to probe registers in TrueNorth, for example the neuron potential V_j , even when using NSCS, so to ensure the processes going on in the chip matched what we thought we understood, a spiking simulation was written in Python. This simulation took the update rules described in the previous section and wrote them explicitly, thus it also ran on spikes. This turned out to be very useful and helped us to debug a number of issues that we could not have with NSCS alone. It also allowed us to try things which weren't possible with the chip simulation, for example extending the range of the registers beyond what is physically possible on the chip.

A Python simulation was also used as a baseline to compare the TrueNorth spiking estimates to. The Python Kalman filter baseline was implemented using simple linear algebra techniques.

Two datasets were generated to test the Kalman filter: a 2D system which consists of the amplitude and phase of a sine wave and a 3D system that contains the position, velocity, and acceleration of a 1D projectile, examples of both of these are shown in Fig. 3.26. Data was generated for a variety of starting conditions and combinations of measurement and process noise. The default conditions, unless specified, were a process noise of 1×10^{-3} , measurement noise of 1×10^{-1} , a timestep of 1×10^{-3} and the simulation was usually run for 1000 timesteps.



(a) An example of the 2D dataset generated, tracking the amplitude and phase of a one-dimensional sine wave.



(b) An example of the 3D dataset, tracking the displacement, velocity, and acceleration of a projectile.

Figure 3.26: Examples of the 2D and 3D datasets generated to test the Kalman filter. The grey crosses are the noisy measurements input to the Kalman filter. The solid coloured lines show the non-spiking, or ideal, estimate of the state from the non-spiking Kalman filter. The black dots show the estimate of the state from the spiking Kalman filter.

3.4 Results and discussion

This section will evaluate the performance of the steady-state Kalman filter model implemented as described in section 3.3.2 and hence explore the limitations of the TrueNorth architecture in performing linear algebra. Most of the topics explored in this section are relevant to both the serial and parallel versions of the implementation, aside from the discussion on the compromises necessary to ensure speedup through parallelism.

3.4.1 Evaluation metrics

The difference between the state as estimated by the spiking Kalman filter, $\hat{x}_{spiking}$, and the noiseless state, x , was first considered as a metric with which to evaluate the performance of the TrueNorth Kalman filter. This metric directly evaluates how well the spiking implementation filters the measurement noise and recovers the underlying state, i.e. how well it performs as a Kalman filter. However, this was deemed less useful than comparing the spiking Kalman filter output to the non-spiking Kalman filter output, which evaluates how well the spiking version of the algorithm performs when compared with its non-spiking counterpart. The spread of the residuals was settled on as the evaluation metric as it averages the performance of the tracker over many timesteps. Information about the overall structure of the residuals is lost but any biases will be referred to explicitly in the write-up. The metric is defined as:

$$\frac{1}{N} \sum_{k=1}^N (x_{non-spiking} - x_{spiking})^2 \quad (3.23)$$

or the variance of the residuals. In this text it will be referred to as the Mean Square Error (MSE).

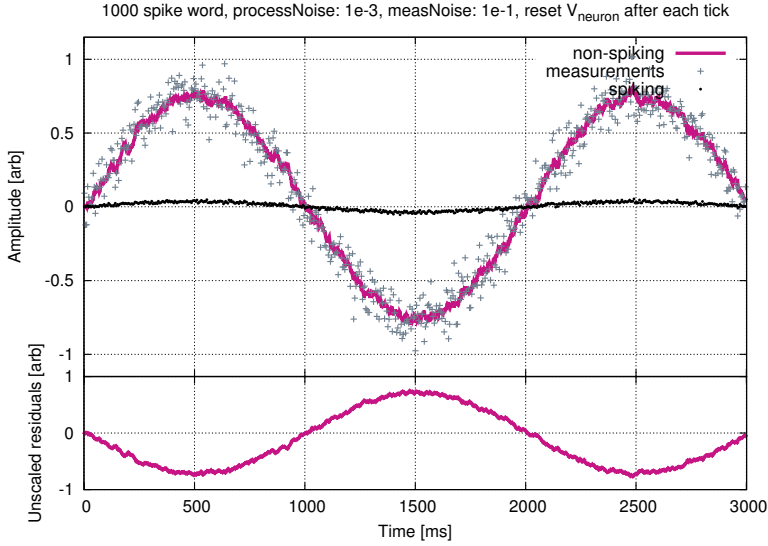
3.4.2 Neuron remainder and reset schemes

Picking the correct reset scheme for a parallel multiplier was discussed at some length in the implementation section of this chapter. In this section, the effect of other reset schemes on the behaviour of the Kalman filter will be discussed.

A reset scheme describes what happens to V_j after neuron _{j} fires. All neurons in the Kalman filter implementation use a ‘linear’ reset, where V_j is reduced by $(\alpha_0 \times \# \text{ neurons fired in block})$. In the linear reset scheme it is possible that the neuron weight is not a multiple of the threshold. In that case, after the neuron fires, there will be some remainder left over in the neuron potential which is effectively the quantization error of the weight representation in the neuron. The question, then, is whether to allow this remainder to (1) be thrown away before the next tick⁵, (2) be removed at the end of each word, (3) persist. The effect of these three reset schemes is shown in Fig. 3.27, on the 2D sine-curve dataset.

Clearly, reset scheme (1) is unusable as losing the remainder after each tick stops the filter being able to output the correct magnitude of the state, even though it shows some tracking capability - thus it is not explored further. However, previous uses of TrueNorth [123] show good performance for their intended task by setting the neuron potential to zero at the end of each word. This reset scheme, reset (2), is implemented by a complex crossbar involving set and reset pins. Although, from the single use-case shown in Fig. 3.27b it is clear that the residuals show a small bias when compared with Fig. 3.27c, how this scaled for larger encoding representations was explored for both the 2D and 3D datasets, shown in Fig. 3.28. It is clear that this reset scheme consistently produces a result that is further from the ideal non-spiking

⁵This would essentially force V_j to zero after it fired, something that could be implemented by setting $\gamma = 0$.

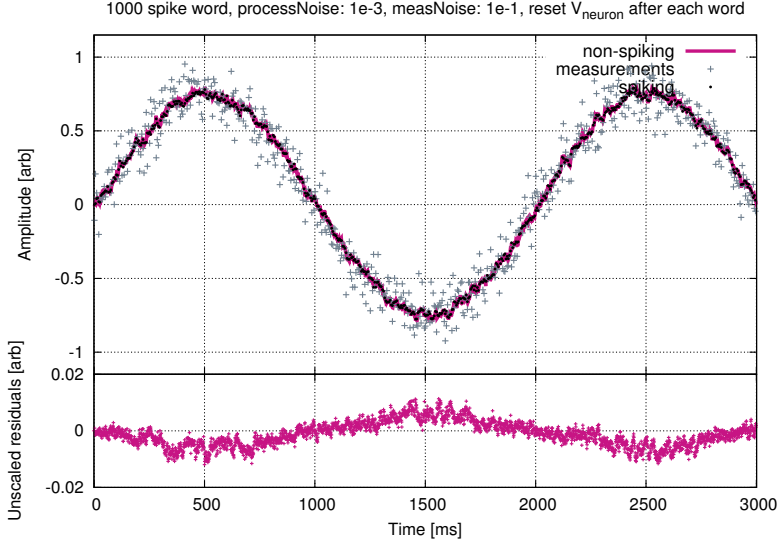


(a) V_j set to zero after every tick, i.e. remainder always discarded.

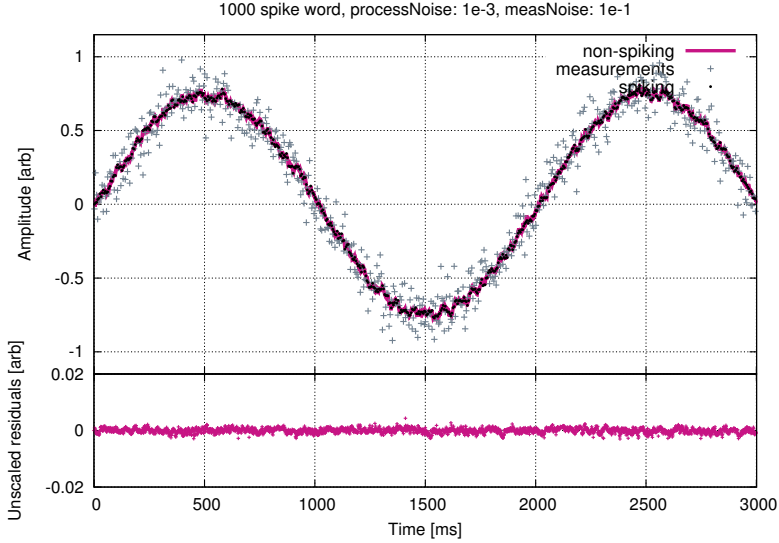
Figure 3.27: The effect of various reset schemes on the performance of the spiking Kalman filter on a 2D tracking problem, encoded with 1,000 spikes per word, using the maximum weight register size. Note the difference in the residuals scale to the other two plots in this figure.

estimate, regardless of the number of spikes used to represent the data word.

Since both reset schemes (1) and (2) showed worse performance when compared to the equivalent non-spiking filter than reset scheme (3), which preserves the neuron remainder, all performance tests were run with reset scheme (3) and the Kalman filter implementation was implemented using it.

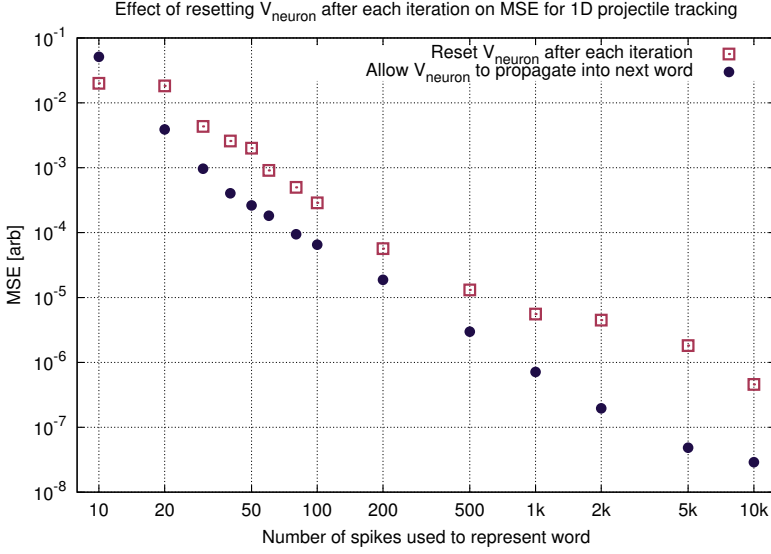


(b) V_j set to zero after every data word, i.e. remainder discarded after every word.

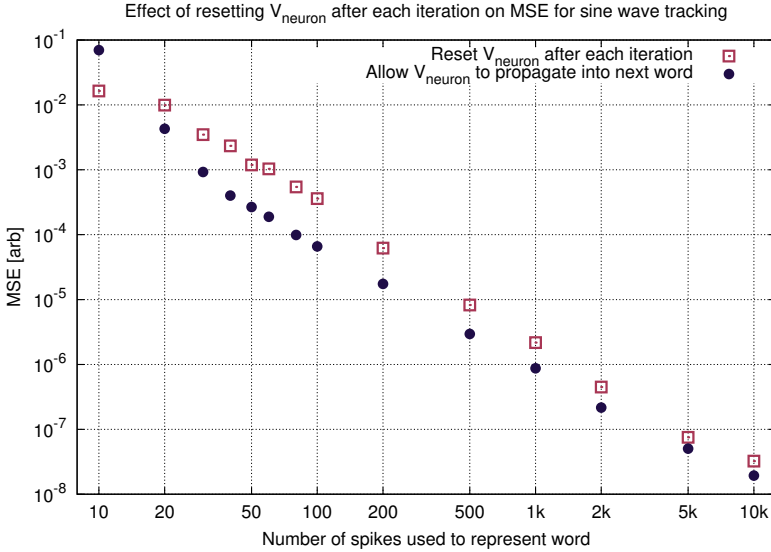


(c) V_j is linearly reset by $\alpha_0 \times \text{spikes fired in neuron block}$, i.e. the remainder persists between words.

Figure 3.27: The effect of various reset schemes on the performance of the spiking Kalman filter on a 2D tracking problem, encoded with 1,000 spikes per word, using the maximum weight register size. (*cont'd*)



(a) Comparison of reset schemes using the 3D projectile dataset. The y-axis is the variance of residuals between spiking and non-spiking Kalman filters.



(b) Comparison of reset schemes using the 2D sine wave dataset. The y-axis is the variance of residuals between spiking and non-spiking Kalman filters.

Figure 3.28: A comparison between reset schemes (2) and (3) for 2/3D datasets across a range of word-lengths measured in spikes. Consistently, for both datasets and across all ranges of encoding word, the reset (3) scheme, which allows the remainder in V_j to persist, outperforms resets scheme (2), which resets V_j after every word.

3.4.3 Input and weight representation precision

TrueNorth processes digital spikes which encode data in trains using either rate encoding or a combination of rate and population encoding. The number of spikes used to represent a given data word changes the accuracy with which that data is represented. For example, a 10-spike word will have a stepsize an order of magnitude larger than a 100-spike word. The more spikes used to encode data, the better the representation of that data and the closer the spiking Kalman filter performance approaches the non-spiking Kalman filter.

However, this effect is mitigated by how well the weights in the $\mathbf{A}$ and $\mathbf{B}$ matrices are represented. Incoming spikes in the neuron are multiplied by a neuron weight $s_j^{G_i}$ and divided by a threshold α_j , which are both integers. So the ratio of those integers is meant to be equivalent to the matrix weight - however, both the neuron weight and threshold are finite sized registers and so introduce an error on the state estimate. The combination of encoding representation and weight representation affects the accuracy of the state estimate in the Kalman filter, as is shown in Fig. 3.29 for both the 2D and 3D datasets.

Although there is a consistent improvement in the output of the spiking filter with respect to the non-spiking filter with using more spikes to encode a state, there is a limit to which this holds. This limit is dependent on how well the weight is represented in the neuron. For example, using the default neuron weight range of $[-255, 255]$, increasing the number of spikes used to encode a word beyond 1,000 does not improve the accuracy of the spiking Kalman filter relative to its non-spiking counterpart. This plot is capped at 10,000 spikes per word as, with the default tick rate, that equates to 10 s to parse an entire word into TrueNorth. The weight is capped at (1028×255) as it produces the same range of values as the threshold. The value of (256×255) is also significant as it is the maximum range that could be achieved by weight duplication, as described in section 3.3.2.

This plot shows that whilst picking a good encoding scheme will improve the accuracy of an output, this effect is mitigated by how well the weights in the system are represented.

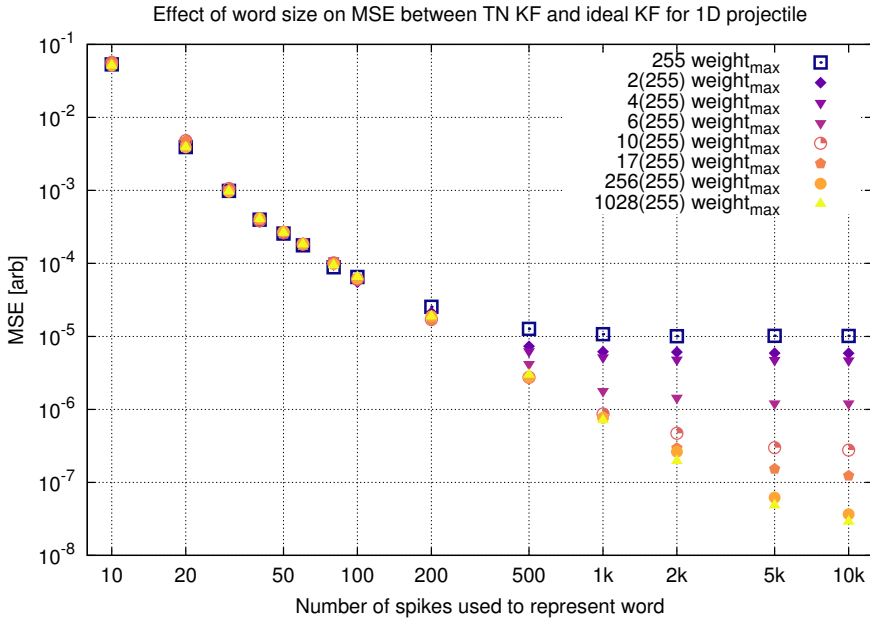


Figure 3.29: The effect of data word size, measured in spikes, on the Mean Square Error of the prediction spiking Kalman filter when compared with the equivalent non-spiking filter. The plot also shows the effect of effective weight register size, as the number of axon-neuron pairs assigned to increase the range of the weight register from 255, the performance of the spiking Kalman filter improves.

3.4.4 Latency and tick: timing limitations

Speedup through parallelism

In section 3.3.2, a way to implement the parallel Kalman filter was discussed, and the number of neurons, axons, and cores needed to implement it was explained. This section will explore effect of the finite size of the core on the available speedup and weight precision available in that parallel setup.

The limiting factor on the maximum speedup available from parallelising spike-train processing is the size of a TrueNorth crossbar. There are 256 neurons and axons available on a single core, so any design that requires more of them to be used cannot be synthesised. In the parallel Kalman filter design the corelet that uses the most neurons and axons for a given blocksize is the multiplication block, thus there is a constraint that:

$$b \left[q + \frac{p(b+1)}{2} \right] \leq 256 \quad (3.24)$$

Where b is the blocksize or parallelism, q is the amount of duplication of the input used to increase the neuron weight, and p is the amount of duplication used in the reset crossbar to extend the range of the neuron threshold that can be used.

Out of the three parallel corelets detailed: multiplication, same-sign addition, and different-sign addition, this corelet places the largest spatial constraints on the implementation. However, this equation assumes that the weight representation precision should be maximised. If we assume the minimum-possible weight representation precision, i.e. that:

$$p = q = 1 \text{ and also } \alpha_j < \frac{255}{b-1}$$

then this formula reduces to:

$$\begin{aligned} 3b - 1 &\leq 256 \\ \therefore b &\leq 85 \end{aligned}$$

However, in this scheme of minimum weight-precision, the different-sign parallel addition corelet becomes the limiting factor:

$$\begin{aligned} 6b - 2 &\leq 256 \\ \therefore b &\leq 43 \end{aligned}$$

So with the minimum possible weight representation in the multiplication core the maximum speedup that can be attained through parallelism is 43.

If the constraints on minimising the weight precision in favour of speedup begin to be relaxed then the condition:

$$p = q = 1$$

remains. In other words because the threshold is no longer less than $\frac{255}{b-1}$, the positive reset axons in the crossbar will have to be duplicated. In that case the maximum blocksize, and hence speedup attainable is limited by the size of the multiplication core:

$$\begin{aligned} \frac{b(b+3)}{2} &\leq 256 \\ \therefore b &\leq 21 \end{aligned}$$

If the constraint that $p = 1$ is also relaxed, i.e. that the threshold (and denominator of rational representation of matrix element weight) can be > 255 , then the maximum attainable speedup

changes again, according to equation 3.24. This is plotted for all possible values of p and b , given a 256×256 core in Fig. 3.30.

From this plot it can be seen that the maximum weight resolution attainable with a speedup of 2 is: 5×10^{-5} , with a speedup of 3 that reduces to 1×10^{-4} , etc. The reason there are bands of similar-resolution representation for different speedup, and vice-versa, is because of the constraint that $p, b \in \mathbb{Z}$, and that equation 3.24 is non-linear. Practically this means that there are bands where the latency of the system can be reduced by adding more neuron-axon pairs without penalising the precision of the weight representation. And equivalently, there exist ranges where the weight precision can be increased without reducing the speedup due to parallelism.

It is important to note that these plots represents the minimum stepsize in weight attainable for a given speedup, but that that stepsize might not always be the optimal representation of the weight element as some axons might need to be dedicated to increasing the range of the numerator in the rational representation of the weight beyond 255, i.e. setting $q > 1$. Appendix B.2 provides the p, q, s, α_0 search used for setting up the multiplication corelet for a given b . Note that if $b \in 2, 3$ this algorithm can be modified as there are enough axon labels available for relaxing the constrain that $\alpha \% 255 = 0$.

Increasing the tick rate

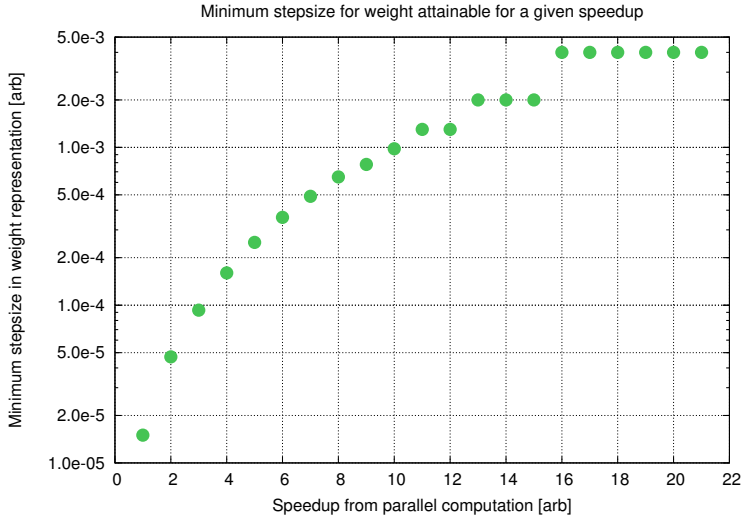
The tick is TrueNorth's global synchronization point and a number of operations must be completed within the span of a tick, else the chip will throw an error state and cease operation. These error states, and the effect of increasing the tick rate on the evaluation of the Kalman filter model, will be discussed in this section.

If the situation arises where there are a high density of spikes to compute for a given core, it is possible that the neuron will not finish computing all the spikes queued in the scheduler memory before the next spike arrives. In this case, an error is thrown by the token controller which shows up as a 'TOK' core error. One can imagine that if a dense spike-train coupled with a parallel implementation was implemented on a core, the dense crossbar and input, for a small enough tick, would cause this error to be thrown.

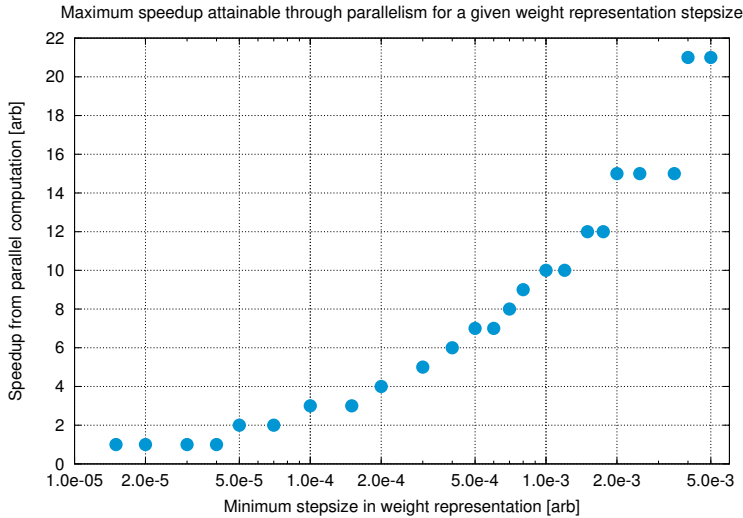
Another timing error that can occur is with routing spikes. If not all the spikes, issued from neurons in the previous tick, have arrived at the router of their destination core before the next tick is issued, a scheduler error or 'SCH' core error is thrown. However, these are not always caught, for example if a spike is off by a whole tick, no error will be thrown and it will simply be queued incorrectly in the wrong tick. Situations where this could occur are if the entirety of the chip is already being utilised and a spike packet has to traverse many cores to reach its destination, and/or if there are many spikes in the network that have to be routed within a given tick.

To try and understand the maximum speedup that could be achieved for this implementation of the Kalman filter, the tick rate was increased. The tick rate is generated externally and fed into the chip. On the NS1e board, the ZYNQ SOCFPGA generates the tick according to a parameter assigned in the model.conf file produced by CPE, the parameter is called "tick-Period" and is measured in ms.

The default value for this parameter is 1, i.e. a 1 kHz global tick. A test was set up to reduce the tick period in steps to see at what point the errors described began to occur and which ones occurred first depending on model, and serial or parallel implementation. The serial implementation of the 2D model as it has the lowest synaptic density and hence core usage of the models produced. To put that in perspective, the 2D serial Kalman filter can be synthesised in 3 cores: a single core for the Kalman filter corelet, and a core each for the delay and splitter corelets - which is $< 0.1\%$ of the entire cores in the chip. Also, since the 2D model was used,



(a) As more neuron-axon pairs are dedicated to speedup through parallelizing the input, the precision with which the weight is represented decreases. Because of the non-linearity of the relationship between speedup and the number of neuron-axon pairs for extending the weight representation, certain speedups can be used without penalising the precision of the weight representation, e.g. a speedup of 12 enforces the same penalty as a speedup of 14.



(b) Equivalently, increasing the precision of weight representation reduces the number of neurons and axons available to parallelise the data and there are also increases in precision that do not penalise the speedup.

Figure 3.30: The relationship between speedup achieved through parallelizing a spike train over a population of neurons and axons and the maximum resolution for weight representation using the neuron's weight and threshold.

spike trains oscillated between being sparse and dense as the sine wave being tracked moved around 0.

First the tick rate was double to 2 kHz, but any tick rate higher did not seem achievable, although no error was thrown by the chip. Upon further investigation it turned out that the limiting factor was the chip's bandwidth. Ideally, each of the 6 ports on TrueNorth is rated for 15 Mspikes/s, which equates to around 480 Mbps bandwidth. Only two of these ports are accessible on the NS1e board so the maximum total throughput of the board is limited to 480 Mbps. In practice, what happens is that if the ZYNQ core cannot match the data rate requested by the model file, because of the limited bandwidth on TrueNorth, it simply continues to run and throttles the tick rate to match - in that way the correct computations are performed but the performance is limited by the port bandwidth (Rodrigo Alvarez, personal communication, Aug. 2016).

Again it is worth emphasizing the point that this throttling arose just beyond a tick of 2 kHz, using $< 0.1\%$ of the total chip with a serial spike train. If this minimal model cannot be sped-up there is little hope that this version of the chip can be used for latencies beyond 1 kHz for more complicated models.

3.5 Conclusion and outlook

In this project a steady-state Kalman filter implementation was designed for the TrueNorth architecture and its prediction accuracy benchmarked against a numerical implementation in Python. The aim of this study was both to understand how to approach programming tracking algorithms in TrueNorth but also to understand the compromises made to achieve low power. This section will briefly summarise the study and recommend changes to make to this or a similar architecture more suitable for use in the ATLAS experiment.

Overall it was shown that a steady-state Kalman filter can be programmed in the TrueNorth architecture. Given that each neuron in a crossbar can add, subtract, multiply, and divide, a linear algebra based approach was chosen and in the case of encoding spikes serially, was simple to implement.

However, there are a number of factors that must reach a compromise when programming in TrueNorth: latency, data representation precision, and weight representation precision. It has been shown that to achieve a simple implementation that takes a minimal amount of real estate on the chip, there is a large latency as many ticks must be used to represent the data. This latency can be reduced by reducing the number of spikes used to represent the data, but this is done at the expense of the precision to which the data is represented.

If, instead, the latency is reduced by using a parallel implementation, then the accuracy of the weight representation can be compromised. If a parallel implementation is used, at least two spikes must be placed in parallel to accurately reset the neurons. If the complex neuron scheme described in the penultimate section is not employed, then a bias is introduced to the tracking. Whilst it is possible to speed up the serial implementation by increasing the global tick rate of the chip, even the most simple model could not be sped up by more than twice the default tick rate of 1 kHz, and any, more complex, models would be unlikely to perform better.

Aside from the problematic compromises that must be made with speed and accuracy there are specific implementation details that, if improved, would greatly simplify the implementation. If the number of axon labels was increased from four to five, then the constraint that the threshold has to be an integer multiple of 255 could be lifted. This would greatly simplify the weight selection scheme and allow for better weight representation in the parallel implementation.

Whilst spiking is a fundamental aspect of neuromorphic computing, the digital spikes

employed in TrueNorth lack the complexity to encode data in an efficient way and unary encoding must be used. Whilst unary encoding is an inefficient use of bits, a single bit-flip only changes the value being encoded by $+/-1$. In binary, by comparison, the placement of a bit-flip inside a word changes the value of the word by the placement of the flipped bit. Thus, unary encoding and hence spiking, is more resilient to small errors being introduced. For example, this can be seen with overflow in the Kalman filter. If two spike trains of length three ticks are added together, where one train contains three spikes and one train contains one spike, then 4 spikes total will be emitted. One spike will effectively leak into the next window but there is no register that will ‘roll over’, presenting the sum as zero. Spikes are inherently fault tolerant but it is questionable whether fault tolerance is a good compromise for the latency issues described earlier in this thesis.

In TrueNorth specifically, some issues with how the chip is programmed could be easily solved and save both space on the chip and a lot of time for developers. One of the most inefficient uses of crossbars is the splitter. The splitter is only necessary because neurons cannot send spikes to multiple axons and it is not clear why this feature cannot be added in a newer version of the chip.

Currently, to connect two corelets together, a Connector class must be used. The Connector classes must match in dimension, there is no way to connect, for example, two of the pins on a four pin Connector and leave the other two floating. The practice of not connecting all the pins in a model, although not always advisable, is at least possible to do in other hardware description languages, such as VHDL. Instead, additional corelets must be made to convert one Connector size to another, which is a waste of the chip’s resources and adds additional latency. However, one feature that would greatly improve the developer experience is in the chip simulation. There is no way to probe the internal registers of the chip simulation during run-time. In FPGA programming, register values are routinely probed over the course of a simulation to assist in debugging logic. In TrueNorth the most important hidden register is the neuron potential, V_j . With no way to know the value of the neuron potential during run-time, a python-based model of the chip was written to emulate its functionality. Because the neuron potential was just an integer number in Python, it could be printed out and used to debug timing and connectivity logic.

Whilst the implementation of a steady-state Kalman filter in a neuromorphic architecture is possible, in a chip like TrueNorth many compromises must be made to obtain accurate tracking at a reduced latency. If neuromorphic chips are to be explored for future implementations of the Kalman filter, it seems reasonable that chips which have a higher degree of freedom, for example that make computations with analogue spikes, or that have a true LIF neuron model would be better places to start. It is also imperative, when deploying a Kalman filter, that the chip has the ability to change its weight matrix at runtime. The most commonly used Kalman filter types in track reconstruction for the ATLAS experiment, are a combinatorial and extended Kalman filter both of which have their covariance matrices updated every iteration.

Thus, it seems unlikely that a digitally spiking neuromorphic chip, such as TrueNorth, would be used to implement tracking algorithms for use in track reconstruction in the ATLAS experiment. However, the lessons learnt with respect to fault tolerance, neuron model complexity limiting the algorithms that can be implemented, and optimal register size could be used to investigate, or even design, more suitable architectures going forward.

Chapter 4

Search for long-lived, massive particles in multi-jet events with displaced vertices in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector

4.1 Introduction

This chapter will outline a search for long-lived particles produced in physics processes beyond the Standard Model using data collected by the ATLAS experiment between 2015-2018. The chapter will begin by motivating these searches, defining what a long-lived particle is in the context of the analysis, and discuss how ATLAS, and other collider experiments, have searched for long-lived particles previously.

Section 4.2 will be dedicated to a long-lived particle search that is characterized by an event with a large multiplicity of high p_T jets and a reconstructed multi-track vertex, displaced from the interaction point. First, the general structure of the analysis and an outline of the selection will be discussed, followed by a more detailed section on background studies that are currently underway.

Following the overview, two studies on the benchmark signal model will be described, the first in section 4.3.1 on improvements to the DRAW filter, that selects events to undergo special reconstruction, and the second in section 4.3.5 on the cause of inefficiencies in displaced vertex mass reconstruction. The chapter will conclude with by discussing the expected reach of the analysis as well as considerations for future studies.

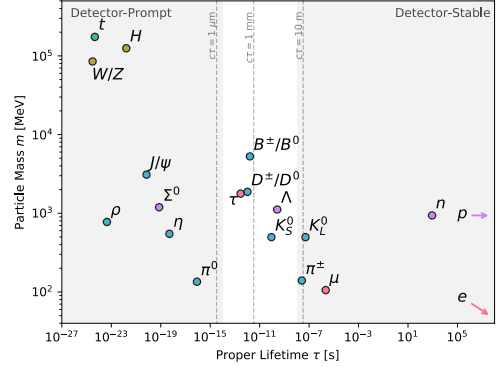
4.1.1 Long-lived particles

To date, whilst tensions exist, no analyses at collider experiments have observed evidence for physics Beyond the Standard Model (BSM). As ever more stringent limits are set on the wide variety of experimental signatures predicted by BSM models, and the models themselves, searches are narrowed down to less accessible regions of parameter space to account for our unexplained observations in nature, some of which were outlined in the introduction. One such region that is ripe with possibility is the search for BSM long-lived particles (LLP).

Whether a particle is considered ‘long-lived’ or not depends very much on the scope of the discussion. For example, whilst a muon could be considered long-lived on the scale of most detectors, its lifetime is insignificant when compared with that of the free neutron, see Figure 4.1 which shows a range of Standard Model particle’s lifetimes, masses, and the dominant mechanism for their decay mode. For the purposes of this chapter, a long-lived particle will

Particle	Lifetime	Decay length ($c\tau$)	Comment
n	880 s	$\mathcal{O}(10^8)$ km	$\ll \Delta m$
μ	$2 \mu\text{s}$	0.7 km	"
K_L	50 ns	15 m	"
$\pi^\pm$	26 ns	8 m	Mediator
Λ	0.3 ns	90 mm	Weak
K_S	90 ps	27 mm	"
B	1.5 ps	0.5 mm	"
π^0	90 as	27 nm	Strong

(a) A table of the average, proper lifetimes of a selection of Standard Model particles [8]. The comment field indicates the dominant mechanism for the particle's lifetime: small mass difference, heavy mediator, or a Weak or Strong coupling.



(b) A plot of the average, proper lifetimes and masses of a selection of Standard Model particles, indicating whether they would be considered prompt, long-lived but decaying within the detector volume, or stable on the scale of the ATLAS detector [124].

Figure 4.1: The range of lifetimes of Standard Model particles.

have a lifetime on the order of 10 ps or more.

In the introductory material in Chapter 1, the total decay rate of a particle was described as the sum of the individual decay rates for each decay mode. There are several mechanisms that can cause the decay to be suppressed: a heavy mediator, limited phase space, and a small coupling strength. If the boson that mediates the decay is much heavier than the particle decaying, then the mediator will be off-shell and the decay will be less likely to occur. For example, this is what causes the charged pion to have a lifetime that is significantly longer than the neutral pion. In a similar vein, if the mass of the particle decaying and its decay product are nearly degenerate, the phase space by which the decay can proceed is limited, also suppressing the decay; this is why the free neutron has a long lifetime. If the coupling strength of the decay vertex is very small, the decay is also suppressed, which is what causes SUSY particles decaying to SM quarks via an RPV coupling to be long-lived. It is also possible for a particle's lifetime to be suppressed by the combined effect of any of these mechanisms.

Table 4.1a shows that Standard Model particles have been observed to have a range of lifetimes, and these same mechanisms could also cause BSM particles to be long-lived. Various models that provide solutions to some of the observations that the Standard Model is unable to address predict long-lived particles. These include but are not limited to: dark matter models, magnetic monopoles and bound quirks, hidden valley models, scale-suppressed SUSY, split-SUSY, and SUSY with R-parity violation [124] [125].

The benchmark signal models studied for this analysis all contain particles that are long-lived due to this last mechanism. Figure 4.2 shows three of these benchmark models; in each case the vertex suppressed by a small RPV coupling is shown in blue and marked with the coupling strength, λ'' . Currently, the model in which a gluino decays to an LSP LLP neutralino, shown in Figure 4.2a, has been studied for the analysis whilst the remaining models will be added at a later date.

The next section will outline experimental signatures of long-lived particles and techniques to search for them at the ATLAS experiment.

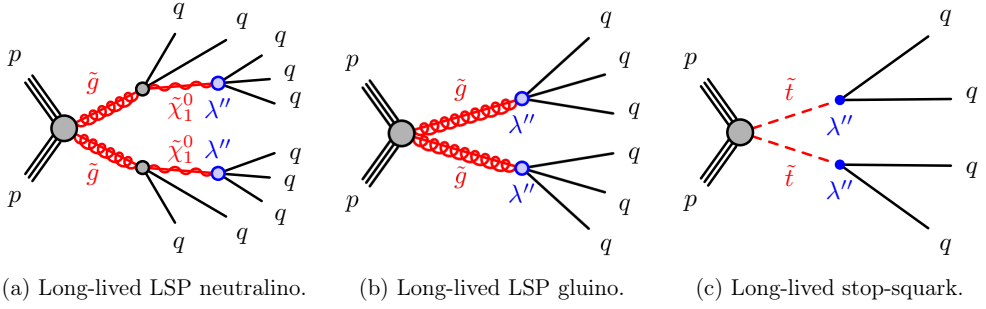


Figure 4.2: Benchmark SUSY signal models for the DV plus jets analysis. Note that in these Feynman diagrams no distinction is made between quarks and anti-quarks.

4.1.2 Long-lived particle searches

Long-lived particles can be detected directly in a collider experiment, under the assumption that they are massive and slow-moving with respect to Standard Model LLP's, using the following techniques. Some of the signatures described below are illustrated in Figure 4.3 in a cartoon of the transverse plane of the ATLAS detector:

- **Large dE/dx :** A heavy particle will have a lower $\beta\gamma$ factor than a less massive particle produced with the same momentum. As such the particle can be identified, if charged, by its increased rate of ionisation per unit length, dE/dx in ionisation chambers such as the ATLAS ID, Cherenkov detectors, and calorimeters with a small radiation length. If this anomalously massive particle is long-lived it can traverse one or more detectors and be identified by its large dE/dx signature [126] [127].
- **Time-of-flight:** A heavy, long-lived particle will take longer to move through a detector than a lighter one. If the particle is slow enough it may even stop in the detectors and decay long after it was produced. In the case of collision detectors, if the detector has timing resolution sufficient to distinguish a hit arriving later than expected (which could be on the order of ns or less, depending on the detector), it is possible to search for LLP's using this signal alone. In the ATLAS experiment both the Muon Spectrometer and calorimeters contain timing detectors that are used in such searches [128] [129] [130].

Long-lived particles can also be detected by their unusual trajectories, or inferred from detector behaviour:

- **Unusual calorimeter deposits** can occur when a long-lived particle decays far from the initial particle collision and either some or all the decay products interact with the calorimeters. In some models only part of an LLP's decay products interact with the calorimeter, for example some SUSY models predict an LSP gravitino that decays from an NLSP neutralino, alongside two photons. The SUSY particles do not interact with the detector but the two SM photons interact with the calorimeter, and balance the momentum of the heavy neutralino, leading to calorimeter deposits that may not point back to the interaction point. If the calorimeter is segmented in such a way that these unusual energy deposits can be reconstructed, the direction that the photons 'point' can be used to search for LLP's [131]. In another scenario, the majority of the decay products *do* interact with the calorimeter so the deposits may point back to the interaction point of the collider but, depending on the lifetime of the decaying particle, may only interact

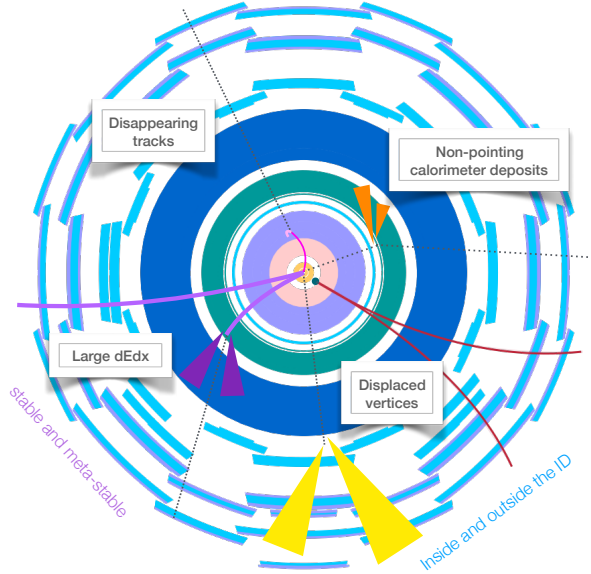
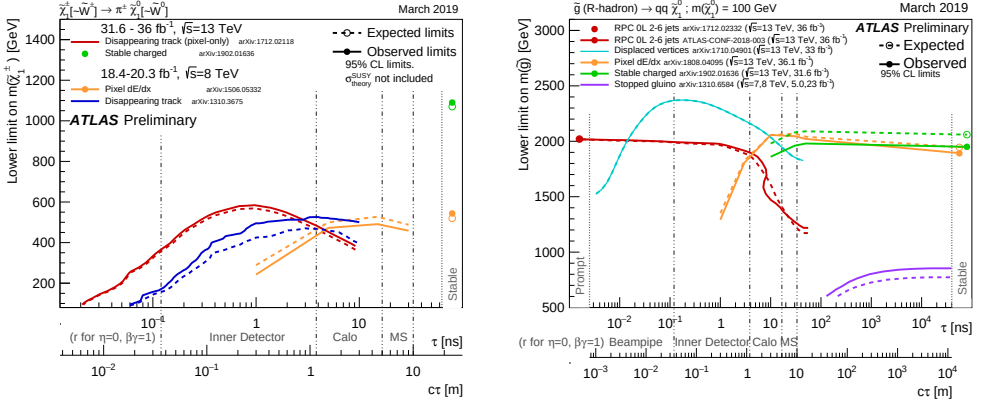


Figure 4.3: A cartoon of how select LLP signatures manifest in the ATLAS detector. Original graphic, courtesy of Laura Jeanty, was modified for this thesis.

with parts of the calorimeter, depositing, for example, a very small fraction of energy in the EM calorimeter with respect to the hadronic. Naturally, a combination of these two effects could also occur.

- **Unusual tracks** may occur if the long-lived particle interacts with the magnetic field of a tracking detector in a different way to SM particles. For example, the trajectory of magnetic monopoles will bend in the direction of the magnetic field, rather than perpendicular to it. For quirks, the BSM gluons that pair them causes their coupling to the magnetic field to change with time resulting in oscillating pairs of tracks [124] [132].
- **Disappearing tracks:** When a charged long-lived particle decays to a massive, neutral particle and the other charged decay products have a low enough momentum such that they may not be reconstructed as tracks, it can seem as if the charged track that was reconstructed from the long-lived particle, disappears. There is also the requirement that the long-lived particle decay in the tracking volume such that the track can be reconstructed. This is, for example, the signature that a long-lived chargino decaying to a mass-degenerate neutralino and a very soft pion would make in the detector [133].
- **Displaced vertices:** When the charged decay products of a long-lived particle are reconstructed, they can be clustered to form a vertex that is displaced from the interaction point. The vertex can be formed from displaced jets, or from displaced tracks. There is no requirement that the long-lived particle be charged or neutral, only that such a vertex can be reconstructed [134] [135]. All three models shown in Figure 4.3 could be reconstructed as displaced vertices.

The analysis described in this work uses the last signature listed, the displaced vertex, to search for long-lived particles decaying within the ATLAS ID and because a SUSY model is used to benchmark the search, the rest of this section will focus on long-lived particle searches using a



(a) The lower limits on the long-lived chargino mass for a range of lifetimes, compiled from four searches.

(b) The lower limit on gluino and gluino R-hadron masses across a range of lifetimes, compiled from five searches.

Figure 4.4: Summary plots showing the minimum mass of long-lived SUSY particles that compile various searches from the ATLAS experiment in Run 1 and 2 [136]. In both plots the x -axis shows both the mean proper lifetime of the long-lived particle and its average decay position, unboosted, with respect to the ATLAS subdetectors. In plot (b), the solid line marked ‘RPC’ is a reinterpretation of the prompt search shown by the red, circular marker in the figure.

displaced vertex signature and how their results can be interpreted for SUSY particles.

4.1.3 Displaced vertex searches

During Run 1 of the LHC, both the CMS and ATLAS experiments searched for LLP’s, initially by using dE/dx and time-of-flight measurements, later including displaced vertex, track, and anomalous calorimeter deposits to extend their searches, which have continued into Run 2. At this time both experiments have set lower limits on the mass of gluinos and stops in the $\mathcal{O}(\text{TeV})$ range, neutralinos, charginos, sbottoms, and staus in the $\mathcal{O}(100)$ GeV range; Figure 4.4b shows lower mass limits set for two of these particles across multiple lifetimes by multiple analyses performed by the ATLAS collaboration.

The most recent lower limits on the stop mass, as well as some of the most recent limits set on the gluino R-hadron mass and lifetime in Figure 4.4b, were set by analyses searching for events with displaced vertices in the ID and either MET or associated muons. In Run 1, the first search was performed on 4.4 fb^{-1} of $\sqrt{s} = 7$ TeV data collected by the ATLAS experiment for events containing displaced vertices with a high p_T muon associated to the vertex [137]. At the end of Run 1, a search for events with DV’s plus some signature, ‘X’, was performed on 20.3 fb^{-1} [138]. In this search ‘X’ was either displaced muons, di-leptons (combinations of electrons and muons), MET, or jets. The DV plus jets channel of that analysis was used to set an upper limit on the gluino and squark production cross-section for split-SUSY and LQ RPV SUSY models, respectively, across a range of LLP lifetimes. Two analyses were performed on 32.8 fb^{-1} of $\sqrt{s} = 13$ TeV, the first on the DV plus MET signature and the second on the DV plus di-lepton signature [139]. The most recent DV analysis focused on the DV plus muon signature and performed a search on 136 fb^{-1} of $\sqrt{s} = 13$ TeV data [140].

Whilst each of these DV analyses iterated and improved upon original techniques that are ultimately explored in this analysis, only the DV plus X search using the 8 TeV dataset

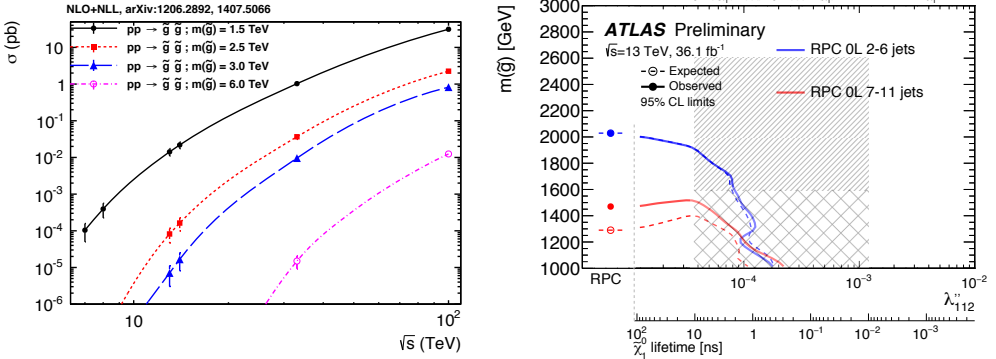
collected at the end of Run 1, included a DV plus jets signature, and so is the only one directly comparable to this search. The next section will describe this analysis, starting with a summary of the analysis and selections and background estimation techniques and outlining key differences between it and the Run 1 DV plus X search.

4.2 Analysis overview

The analysis described in this work searches for evidence of new physics processes producing a massive, multi-track displaced vertex and multiple jets in 139fb^{-1} of proton-proton collisions collected by the ATLAS experiment at a centre-of-mass energy of $\sqrt{s} = 13\text{ TeV}$. The structure of the analysis is dependent on the reconstruction of the displaced vertex which requires custom software that is too computationally expensive to apply to all raw events collected by the ATLAS detector. As such, events are pre-selected based on the presence of high momentum jets, to undergo special reconstruction. This pre-selection is made by filtering raw events leaving the HLT into a Derived Raw (DRAW) stream, which is discussed in section 4.3.1. With both the jet and secondary vertex collection reconstructed, event preselections are applied to ensure only well-measured data is used, and potential signal events are selected based on properties of both the jets and the displaced vertex, which are discussed in section 4.3.5.

The analysis does not require that the jets are linked to the displaced vertex, only that both are present in an event. There are no SM processes that produce high- p_T jets associated to a massive, multi-track displaced vertex. This means that the dominant source of background events are those where high momentum jets are produced in the hard-scatter and an unrelated background DV is present. There are three main background processes that can produce a displaced vertex signal: hadronic interactions, accidental crossing, and merged vertices. To remove the majority of hadronic interactions, a veto is applied to any DV's reconstructed in regions of known detector material. Estimation techniques for the remaining background DV processes are discussed later in section 4.4.

The analysis will benchmark its sensitivity to new physics against a SUSY process in which two strongly produced gluinos each promptly decay to two SM quarks and a long-lived neutralino, which decays via a small RPV coupling to three SM quarks, illustrated in Figure 4.2a. This signal model contains two displaced vertices and ten jets, four of which are produced promptly in the gluino decay, and the remainder of which are displaced. A range of gluino and neutralino mass points are studied, where $m(\tilde{g}) = [1.6, 2.6]\text{ TeV}$ and $m(\tilde{\chi}_1^0) = [10, 2500]\text{ GeV}$ and neutralino lifetimes across four orders of magnitude: $\tau(\tilde{\chi}_1^0) = [0.01, 10]$. The full grid of signal models in the mass plane is shown in appendix C.1. This benchmark model is simulated in stages. First the process is generated at the parton level to next-to-leading order in Madgraph [144], which contains information about the SUSY model used, can parse the information to specify how the particles decay and their lifetime. For this sample, the proton-proton interaction produces a pair of gluinos, each of which decay 100% of the time to a neutralino and a SM quark and antiquark. The neutralino is then decayed to three SM quarks/antiquarks via the RPV baryon-number violating coupling. It is assumed that all other RPV couplings are set to zero. The decays are simulated for light flavour quarks (up, down, charm, or strange), heavy flavor (strange, top, or bottom), and a fully democratic decay (in which all six quark flavours are possible). The majority of the initial signal models generated, which are models used in the studies described in this chapter, use the light flavour decay scheme. This analysis will explore the effect of heavy flavour decays on the signal efficiency at a later date. The simulation was also set up to include up to two initial state radiation partons. The Madgraph events are then passed to Pythia [145] which models showering, hadronization, and the underlying event. Pythia was set up using the A14 tuning and NNPDF23LO parton distribution function [146] [147].



(a) The calculated gluino production cross-section for gluinos of masses $\{1.5, 2.5, 3, 6\}$ TeV as a function of proton-proton centre-of-mass collision energy, $\sqrt{s}$. The calculations are made starting from constraints from the LHC Run 1 data and using MSSM SUSY, error bars shown for each marker quantify the effect of varying the renormalization and factorization scales [141] [142].

(b) The upper limit on gluino mass as a function of neutralino lifetime for a 200 GeV RPV neutralino. The solid lines are reinterpretations of the prompt search shown by the circular markers in the figure. Each line shows the reinterpretation for a different jet multiplicity event selection [143]. The hatched and cross-hatched areas show the expected reach of this analysis for different neutralino mass ranges, see text for details.

Figure 4.5: Predictions on the gluino production cross-section as a function of proton-proton centre-of-mass collision energy (a) and the reinterpretation of a prompt RPV SUSY decay for a cascade decay gluino (b).

Once the signal processes have been generated they are passed into a detailed simulation of the detector which models the interaction particles with each subdetector. This stage of the simulation is also set up to model the collision conditions, which include the pile-up and non-collision backgrounds observed in data taking during each run period.

4.2.1 Scope of this analysis

Although this analysis is not using the same benchmark signal model as the DV plus X search, a few statements can be made about scope of this analysis compared to that, based solely changes in run conditions.

Between Run 1 and Run 2 the instantaneous luminosity of the collisions increased from $8 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$, at the end of Run 1, to $2 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ in Run 2. All other things being equal the increase in integrated luminosity alone should allow the upper limit on each production mode cross-section to decrease by a factor of at least ~ 6.8 , assuming no events are recorded in the signal region, in this analysis compared to the Run 1 analysis. The proton-proton centre-of-mass collision energy also increases between Run 1 and 2, which increases the gluino production cross-section, allowing the analysis to search for models with higher mass gluinos. Figure 4.5a shows the gluino pair-produced cross-section calculated at six centre-of-mass collision energies, with different markers denoting different gluino masses. The plot shows that increasing the CoM energy from 8 to 13 TeV results in a 1.5-2 order of magnitude increase in the production cross-section for each mass.

As the full selection for this search is not yet prepared, the estimated reach of the analysis for a given signal model cannot yet be calculated, however given the factors mentioned above, a broad region of potential sensitivity can be outlined. Figure 4.5b shows the lower limit on gluino

mass for a given neutralino lifetime at a fixed neutralino mass of 200 GeV, for the benchmark cascade gluino decay to jets used in this analysis. Each contour is extrapolated from a prompt RPV multijet analysis [148] into the regions with measurable lifetimes. Each line represents a different jet multiplicity selection in the event. Overlaying the contours are two shaded areas showing the mass and lifetime points studied in this analysis. The hatched area shows the lifetimes and models studied across all 21 neutralino masses simulated for this signal model, whilst the cross-hatched area shows an extension of that region for the 200 GeV neutralino mass only. Assuming the same signal efficiency for the gluino search in the DV plus X Run 1 analysis but considering the increase in both the size and energy of the dataset, this analysis should be able to probe gluino masses at least 1 TeV larger with neutralino lifetimes across four orders of magnitude, as is shown in the shaded regions of the plot.

The next part of this chapter focuses on the analysis itself, starting with the event selections.

4.3 Event selection

The procedure for event selection can be grouped into four categories: triggering an event and selecting it for special reconstruction with the DRAW filter, followed by the preselections, baseline selections, and signal region selections:

1. **Trigger and DRAW filtering:** Events are triggered on using multi-jet flags. The DRAW filter then selects events, again using the presence of high- p_T jets, for special processing to reconstruct displaced vertices.
2. **Preselections** consist of basic checks on the quality of the dataset at the run, lumi-block, and event level. These checks are independent of the analysis.
3. **Baseline selections** are applied to events that pass preselections and also the following:
 - The event contains at least one displaced vertex with a good fit-quality.
 - That same vertex was reconstructed in a volume of the detector with good reconstruction efficiency.
4. **Signal region selections** are the final selections an event has to pass to be counted in the analysis. For an event to pass the signal region selections it must have already passed all the baseline selections and the following:
 - Calibrated jet p_T and multiplicity selections (informed by DRAW filter).
 - That same DV from the baseline selections cannot be in a region with known detector material.
 - That same DV must have an invariant mass of at least 10 GeV.
 - That same DV must have at least 5 tracks attached to it.

This part of the chapter will consider each part of the event selection described above. Starting with the multi-jet trigger and DRAW filtering in section 4.3.1, then the preselections and baseline selections in section 4.3.3, and finally the signal region selections including the material veto in section 4.3.4 and the DV mass and track multiplicity in section 4.3.5.

HLT nJets	nJets in L1 & p_T threshold [GeV]				HLT nJets	HLT jet p_T threshold [GeV]			
	2015	2016	2017	2018		2015	2016	2017	2018
3	1j, 100	1j, 100	1j, 100	1j, 100	3	175	175	200	200
4	3j, 40	3j, 50	3j, 50	3j, 50	4	85	100	100	120
5	4j, 20	4j, 15	4j, 15	4j, 15	5	60	70	70	85
6	4j, 15	"	"	"	6	45	60	60	70
7	-	"	"	"	7	-	45	45	45

(a) Lowest p_T L1 triggers and nJets for each corresponding HLT, by year.(b) Lowest p_T Run 2 HLT multi-jet triggers, by year.

Table 4.1: Lowest unscaled Level 1 and HLT multi-jet triggers in Run 2 [149] [150] [151].

4.3.1 Event trigger and DRAW filter

As was discussed in section 4.1, the long-lived neutralinos targeted in this search will decay to streams of hadrons that can be reconstructed as jets and ID tracks. In section 1.3.1 it was shown that tracks produced at a large transverse distance, R , have a higher reconstruction efficiency when both standard and Large Radius Tracking (LRT) are run. This is because large- R tracks have fewer silicon hits than a track originating from the centre of the detector and because the track has a large transverse impact parameter, d_0 ; the restrictions on both of these selections are relaxed in LRT compared to standard tracking. LRT and standard tracks are then used to form displaced vertices.

However, LRT increases the time taken to reconstruct an event with Run 2 pile-up conditions by a factor of around 2.5. Running secondary vertexing only increases the time taken to reconstruct an event between 200–400 ms, depending on the pile-up and track multiplicity of the event, which is only an $\mathcal{O}(1)\%$ level contribution to the overall event reconstruction time, so LRT is the dominating contribution. As the computing resources available to the ATLAS experiment are limited, which was briefly discussed in section 1.4, it is not possible to reconstruct every event using LRT and secondary vertexing. In addition, because the physics processes producing high-mass displaced vertices have relatively low cross-sections compared to Standard Model processes, it does not make sense that every event should undergo special reconstruction to search for displaced vertices. As such, a derived-RAW (DRAW) data format was created that contains a subset of the raw data from the detector. DRAW filters are used to siphon off data streams from the experiment, and LRT and secondary vertexing are only run on events passing those filters, see Figure 4.6. These filters act almost as a 4th layer of trigger after the HLT but are run offline on Tier-0. Because special objects, like displaced vertices, have not been constructed yet in data streams coming directly from the experiment, the DRAW filters act on other reconstructed objects, such as high p_T muons, MET, or jets to select events.

In this analysis the signal model studied will produce an event containing a reconstructed displaced vertex as well as high p_T jets of various multiplicities. As such, both the trigger and the DRAW filter target events with high p_T jets. The ATLAS Run 2 lowest unscaled triggers relevant to this analysis are shown in table 4.1, sorted by year. Table 4.1a shows the Level-1 triggers that must be passed for each corresponding HLT trigger. For example, for an event to be considered for the 2018 5-jet HLT trigger, it must first have passed the L1 4-jet trigger which requires that four jets in the event have p_T of at least 15 GeV. Table 4.1b then shows the jet multiplicity and p_T thresholds for events to pass the HLT. The lowest unscaled HLT triggers for Run 2 are used to inform the DRAW jet p_T thresholds discussed in the next section.

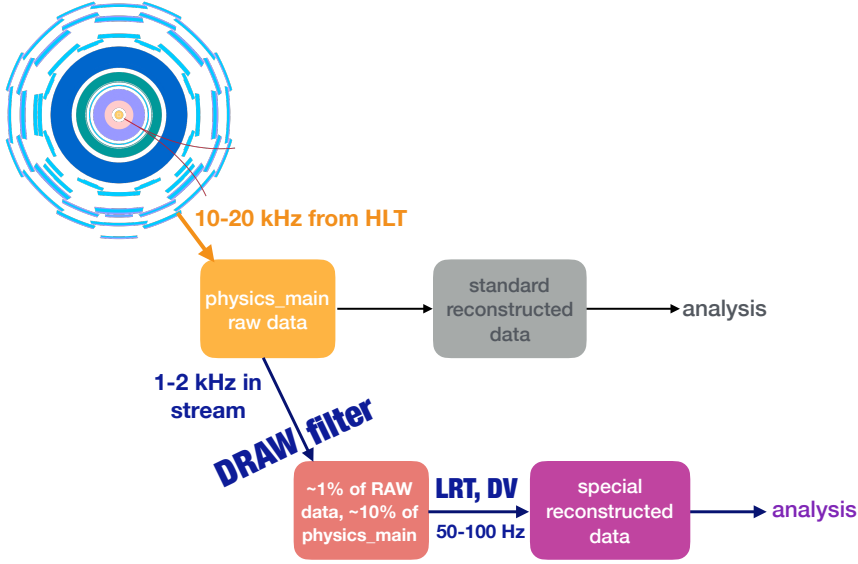


Figure 4.6: A cartoon showing the data flow from HLT through the DRAW filter to undergo special tracking and vertexing, the approximate data rates are also shown.

DRAW filtering

There are currently twenty-five DRAW streams used in ATLAS, twenty of which are used for performance studies, three for tracking studies, and two for BSM physics searches. Of these two, the DRAW RPYLL stream is used for this analysis.

The DRAW RPYLL filter incorporates twenty-two kernels which each use a different combination of HLT-level objects and triggers to select events. The kernels are, at time of writing, used by seven groups of signatures in the ATLAS experiment: disappearing tracks, highly ionizing particles, heavy neutral leptons, emerging jets, displaced jets, dilepton displaced vertices, displaced vertices in the MS, and multi-track displaced vertices (this analysis). In Run 2 the rate of data leaving the HLT in the Main physics stream is around 1–2 kHz, which is around 75% of the HLT bandwidth. Of that, 5–10% of the stream is filtered by the DRAW filters, of which the DRAW RPYLL stream accounts for around 60% of the bandwidth. A breakdown of the filter rates for a typical Run 2 run are shown in Appendix C.

This section will summarise a study to check the pre-existing Run 2 filter for inefficiencies and describe how they were mitigated for data collected by the ATLAS experiment in 2017–2018. In Run 1, events that had already undergone full reconstruction were selected based on the presence of [4–6] jets passing the trigger thresholds and at least one trackless, 45 GeV jet per event. In Run 2, the DRAW filter was modified first by increasing the jet p_T thresholds to scale with the higher multi-jet trigger thresholds. It was then made more complex by increasing the range of jet multiplicities that would be accepted into the filter to [3–7] and adding a flag based on jet $\Sigma^{\text{trk}} p_T$ (a trackless jet flag) to the DRAW filter. However, improvements to standard tracking at larger radii removed the necessity of requiring that one or more jets in an event be trackless to pass the filter for multi-jet events. The filter used in 2015–2016 data-taking is called the `DVMultiJetFinalTracklessFilter` and is comprised of three parts:

- The `DVMultiJetTriggerFlags` are a list of multi-jet trigger flags, and change between run periods. The lowest, unscaled triggers in Run 2 are listed in table 4.1.

Flag name	Min. # jets in event with these properties	Min. Jet p_T [GeV]	Max. jet $ \eta $	Max. jet $\Sigma^{\text{trk}} p_T$ [GeV]	Min. associated track p_T [MeV]
Single	1	70	2.5	5	500
Double	2	25			

Table 4.2: The Run 1 and 2015–2016 trackless jet flag requirements. The double trackless flag was changed for 2018 data-taking and 2017 reprocessing.

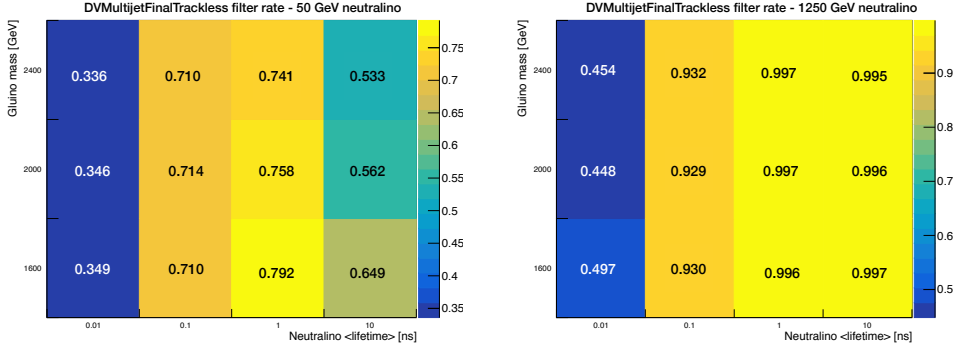
- The **jet multiplicity and p_T flags** check if an event has a certain number of jets, each with p_T larger than some threshold.
- The **trackless jets flags** are divided into two types: single or double. A jet is considered trackless based on the characteristics of the tracks associated with it. If the tracks associated to a jet have not been vertexed or if the sum¹ of their transverse momentum, $\Sigma^{\text{trk}} p_T$, is less than 5 GeV the jet is considered trackless. A minimum track p_T cut of 500 MeV is applied to any tracks being counted in the sum. A jet can also be considered trackless if it is not associated to any tracks at all. An event will pass the ‘single’ trackless jet filter if it contains at least one trackless jet with $p_T \geq 70$ GeV and $|\eta| < 2.5$ and the ‘double’ trackless filter if it contains at least two trackless jets with $p_T \geq 25$ GeV and $|\eta| < 2.5$. Table 4.2 summarises these requirements. A cartoon showing the various ways a jet can be trackless is shown in Figure 4.10b.

These three flags are AND’ed together to produce the `DVMultiJetFinalTracklessFilter`:

$$\begin{aligned}
 &\text{DVMultiJetFinalTracklessFilter} \\
 &= \left\{ \begin{aligned} &\text{tracklessJetFlags} \\ &\& [(2j \& 3j) \parallel 4j \parallel 5j \parallel 6j \parallel 7j] \\ &\& \text{DVMultiJetTriggerFlags} \end{aligned} \right\} \tag{4.1}
 \end{aligned}$$

such that for an event to pass the DV Multijet filter, it needs to contain several high p_T jets, at least one of which must be trackless. The next section will describe two studies: the first to understand the efficiency of the `DVMultiJetFinalTracklessFilter` in accepting signal events and the second to explore redundancy and pile-up mitigation in the filter.

¹The `sumPtTrk` of a jet is set in the `JetTrackMomentsTool` in the `Reconstruction/Jet/JetMomentTools` package.



(a) The DVMultiJetFinalTracklessFilter efficiency for a 50 GeV neutralino.

(b) The DVMultiJetFinalTracklessFilter efficiency for a 1250 GeV neutralino.

Figure 4.7: The DVMultiJetFinalTracklessFilter efficiency for two neutralino masses and range of lifetimes and gluino masses.

Signal efficiency of the DVMultiJetFinalTracklessFilter

To explore the effectiveness of the DVMultiJetFinalTracklessFilter for the benchmark signal model, a subset of points from the full parameter grid were studied, see Figure C.2. The subset explores a range of gluino masses ($\{1.6\text{--}2.4\}$ TeV), neutralino masses ($\{50\text{--}1250\}$ GeV) and all four neutralino lifetimes: $\{0.01\text{--}10\}$ ns. The grid points in which the neutralino and gluino mass are closer in value (e.g. the 1250 GeV neutralino) are often referred to as the ‘compressed mass’ regime in the analysis below. The DVMultiJetFinalTracklessFilter studies described use a metric called the ‘fiducial efficiency’. The fiducial efficiency uses information about the simulated neutralino’s decay vertex to define a region with high reconstruction efficiency and is bounded by $R < 300$ mm of the interaction point in the transverse plane and $|z| < 300$ mm in the longitudinal plane. Therefore, unless stated otherwise, the fiducial efficiency is defined as the number of events that pass the filter where the event must contain *at least one neutralino* that decays within the fiducial region.

The benchmark signal model for this analysis should contain at least ten jets per event: two from each gluino decay and three from each neutralino decay, which should be displaced from the interaction point. The gluino decays promptly, so jets from its decay are unlikely to be marked as trackless. The neutralino decay is long-lived and, whilst it is true that at long-lifetimes tracks are less likely to be reconstructed using standard tracking, and subsequently vertexed using the primary vertex algorithms, this is not the case for short lifetimes where the neutralino decays within the first 2 Pixel layers. This means that the requirement that a jet be trackless to pass the DVMultiJetFinalTracklessFilter is inefficient for short neutralino lifetimes, which is demonstrated in Figure 4.7. In the figure, events containing at least one 0.01 ns neutralino in the fiducial region only pass the filter at a maximum rate of 35% in the case of a 50 GeV neutralino, and up to 50% of the time in the case of a 1250 GeV neutralino.

The improved rate for the compressed regime is a combination of the effects of less collimated jets leading to poorer ghost-association and poorer track reconstruction due to higher angle decay products, hence more trackless jets in the event. In addition, for lifetimes longer than 0.01 ns the increased signal efficiency can be attributed to the second neutralino in the event being less likely to be boosted outside of the fiducial region in the compressed regime; this leads to more jets in the event which may again be trackless.

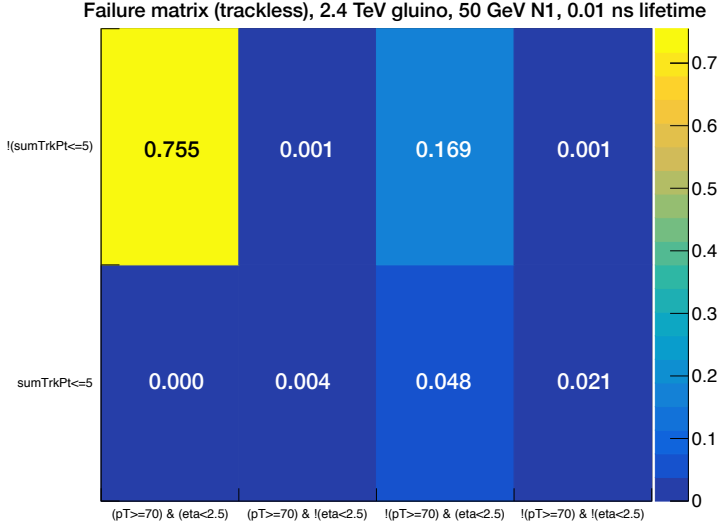
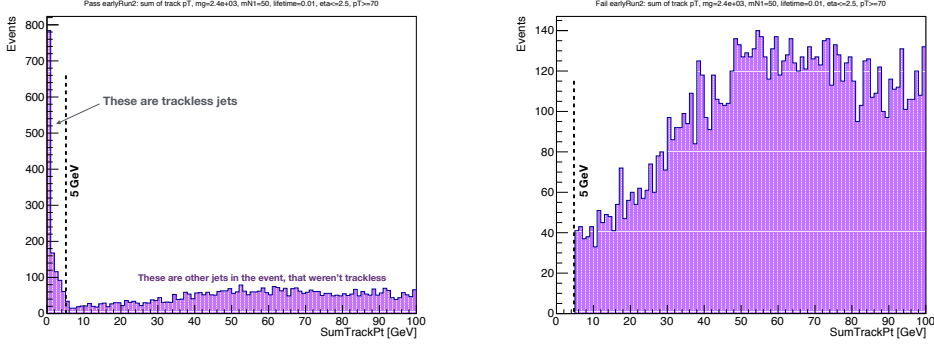


Figure 4.8: The decision matrix for jets in simulated signal events that fail the DVMultiJet-FinalTracklessFilter, where the decision on whether a jet passes each of the three flags that comprise the ‘trackless’ part of the single-trackless filter are recorded.

To interpret the poor efficiency at long lifetimes in Figure 4.7a it is first useful to understand the shortcomings of the filter in more detail. Figure 4.8 shows a decision matrix for simulated signal events that failed the DVMultiJetFinalTracklessFilter. The matrix is constructed from the three components of the ‘single’ trackless jet filter: jet p_T cut, jet η cut, and a cut on the $\Sigma^{\text{trk}} p_T$ of the jet. The matrix is filled by events from the signal parameters with the lowest fiducial efficiency: a 2.4 TeV gluino decaying to a 50 GeV neutralino with a lifetime of 0.01 ns. The matrix is constructed by checking if an event does not pass the filter and then looping through its reconstructed jets with $p_T < 25$ GeV, to mitigate soft jets. The matrix shows that around 93% of jets in events that fail, do not pass the single-trackless flag because they have $\Sigma^{\text{trk}} p_T > 5$ GeV, whilst only around 5% fail the single-trackless flag p_T requirement. These numbers vary slightly depending on the neutralino/gluino mass points that are studied but the clear pattern is that events failing the DVMultiJetFinalTracklessFilter do so because they are not meeting the $\Sigma^{\text{trk}} p_T$ requirement. This is further illustrated in Figure 4.9b which shows the distribution of $\Sigma^{\text{trk}} p_T$ for events that fail to pass the filter. To avoid confusing the reason for failure, the jets are required to have $|\eta| < 2.5$ and $p_T \geq 70$ GeV.

The reduced efficiency seen for the neutralino with an average lifetime of 10 ns can also be explained. The plot in Figure 4.9a shows that even within events that have trackless jets there are others that have a very large $\Sigma^{\text{trk}} p_T$. This implies that, for events with a 10 ns neutralino, the reduced fiducial efficiency may be due to the smaller number of events with two neutralinos in the fiducial region when compared with other lifetimes. It is likely that the neutralino does not in the fiducial region has more trackless jets than the one that is but not if the neutralino decays outside the calorimeter volume. This is further supported by comparing the 1250 GeV and

²The flags were added via the PhysicsAnalysis/SUSYPhys/LongLivedParticleDPDMaker package in Athena, specifically in the DVFlags.py file. These flags were used to create the new filter in the PhysDESDM_DV.py file in the same package.



(a) The $\Sigma^{\text{trk}} p_T$ spectrum for jets in events that pass the DVMultiJetFinalTracklessFilter filter.

(b) The $\Sigma^{\text{trk}} p_T$ spectrum for jets in events that fail the DVMultiJetFinalTracklessFilter filter.

Figure 4.9: Plots showing the $\Sigma^{\text{trk}} p_T$ spectrum for jets in events that (a) pass or (b) fail the DVMultiJetFinalTracklessFilter filter. The dotted line in both plots indicates the maximum $\Sigma^{\text{trk}} p_T$ value a jet can have to pass the trackless part of the filter.

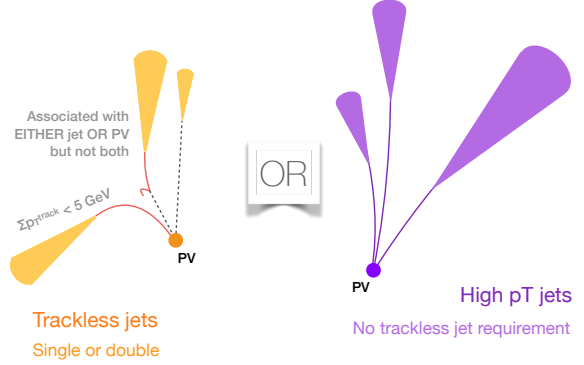
50 GeV neutralino plots for this lifetime, the heavier neutralino is less boosted and so it is more likely that the second neutralino decay before or within the calorimeter volume.

With these inefficiencies in mind, a new set of flags were introduced to the DV multijet filter². These flags are also based on jet multiplicity but have no trackless jet requirements. To keep the rate low, the p_T requirements are around double those in the DVMultiJetFinalTracklessFilter. The thresholds were chosen such that the rate of events passing the overall filter increased no more than 1 Hz in 2017 data-taking, see Figure C.7. The flags and lowest unprescaled trigger requirements for Run 2 are shown in table 4.10a. To entirely decouple the trackless jet requirement from the high p_T jet flag, the pre-existing filter is OR'ed with the new flags, also shown in Figure 4.10b:

$$\begin{aligned}
 &\text{DVMultiJetFinalFilter} \\
 &= \left\{ \text{DVMultiJetFinalTracklessFilter} \right. \\
 &\quad \left. \parallel [(2j_h \ \& \ 3j_h) \parallel 4j_h \parallel 5j_h \parallel 6j_h \parallel 7j_h] \right\}
 \end{aligned} \tag{4.2}$$

Figure 4.11 shows the fraction of events that passed exclusively due to the high p_T filter, i.e. the recovery rate of the filter. The DVMultiJetFinalTracklessFilter is already fairly efficient at filtering the intermediate neutralino lifetime events, so there is minimal recovery for these points but for the 0.01 ns lifetime recovers up to 66.4% of the signal and up to 45.5% of the signal for the 10 ns lifetime. With this addition, the overall DVMultiJetFinalFilter fiducial efficiency is shown in Figure 4.13. The addition of the high p_T flags improves the signal efficiency over all lifetimes leading to a minimum efficiency of only 93.3% and an average efficiency of 99.2% across the parameter grid. Figure 4.12 shows the fraction of events that pass the high p_T filter (regardless of whether they also pass the trackless filter). The figure shows that for this particular signal model, retaining the trackless jets filter increases the fiducial efficiency by at most 2%, compared with just using the high- p_T flags. However, early studies of other signal models with lower- p_T jets indicate that the DVMultiJetFinalTracklessFilter is necessary to retain sensitivity outside of this model and so it will be retained for later editions of the analysis that explore these models.

nJets	HLT [GeV]	Trk-less jet p_T [GeV]	High p_T cut [GeV]
2	-	220	500
3	175	120	180
4	85	100	220
5	60	75	170
6	45	50	100
7	45	45	75



(a) Jet multiplicity and p_T cuts for the lowest unprescaled HLT in Run 2, the thresholds set for the low- p_T threshold for jets which in events which also contain a trackless jet, and the high- p_T threshold for jets in events that do not.

(b) A cartoon showing the logic of the DVMultiJetFinalFilter. An event can pass the filter either because it contains trackless jets and other low- p_T jets or because it has higher p_T jets and the trackless requirement is removed. The cartoon also shows three scenarios whereby a jet could be considered trackless.

Figure 4.10: Trackless and high- p_T jets comprise the DVMultiJetFinalFilter.

Studies on the trackless-jet filter

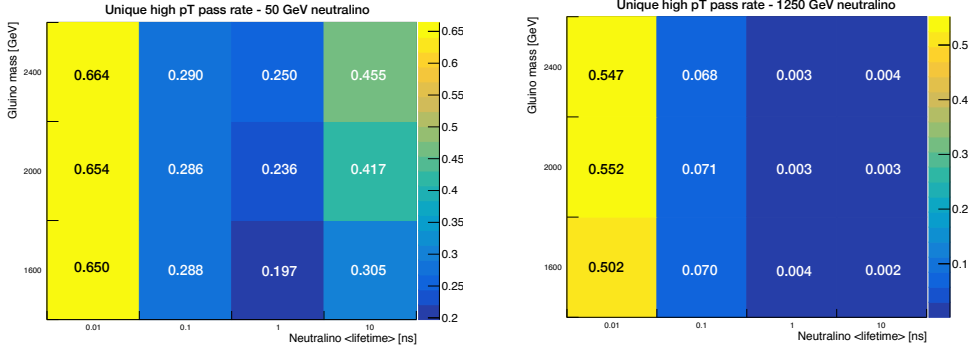
Fundamentally, the DRAW filter is a way to pre-select interesting events and minimise unnecessary computation, so in this section the relative rates of the different flags of the filter will be compared with their efficiency at recovering signal and a modification made to the DVMultiJetFinalTracklessFilter will be discussed.

Figure 4.14 compares the acceptance fraction of events for simulation and data. The DVMultiJetFinalFilter is broken down into its components: the light blue bins show the acceptance rates of individual flags both for the high p_T jets on the far right and for the trackless jets on the left. The darker shade of blue is the overall rate for those two branches of the filter and the dark blue bin is the overall DVMultiJetFinalFilter rate. The plot shows the acceptance rate for a 2.4 TeV gluino, 50 GeV neutralino with a lifetime of 0.01 ns. As was seen in the previous section, this is a model with 100% acceptance by the DRAW filter, and most of that is coming from the high p_T jet filters. The plot in Figure 4.14b shows the breakdown of the filter acceptance for one 2017 data-taking run, around 10 hours of data-taking. The number of events that pass the filter is compared to the number of events in the Main Physics stream from that run. The plot shows that most of the rate comes from the double trackless filter, which was not shown to dominate the pass rate in the signal sample and has redundancy with the high p_T flags³.

In addition, when plotting the flag acceptance rates vs. the average interactions per bunch crossing, $\langle \mu \rangle$, the double-trackless flag shows a clear pile-up dependance compared to both the single-trackless and high p_T flags, see Figure 4.15a. Based on these observations the double-trackless p_T threshold was doubled from 25 GeV to 50 GeV, removing a significant amount of the pile-up dependance and halving the overall rate of the filters, see Figures 4.15b and 4.16b and table C.1. The effect of the change on the signal efficiency of the 50 GeV neutralino grid results in a loss in efficiency of at most 1%, although the majority of points lose $\mathcal{O}(0.1\%)$.

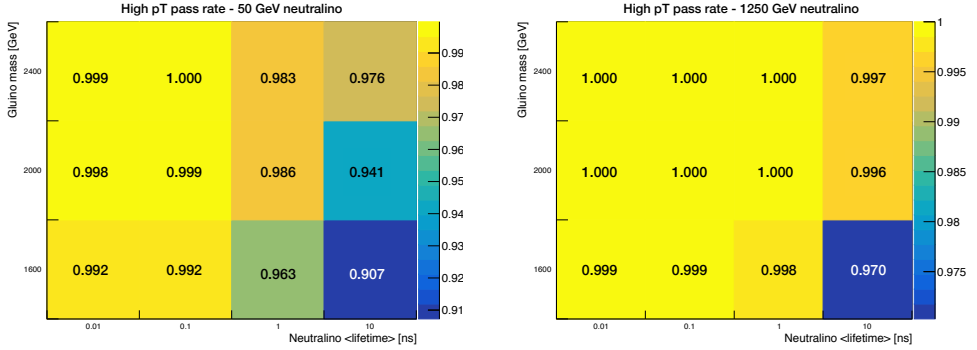
The outcome of these studies resulted in two changes: the increase of the double trackless

³This last point is more obvious by looking at a pairwise comparison of each flag in the filter, see Appendix ??.



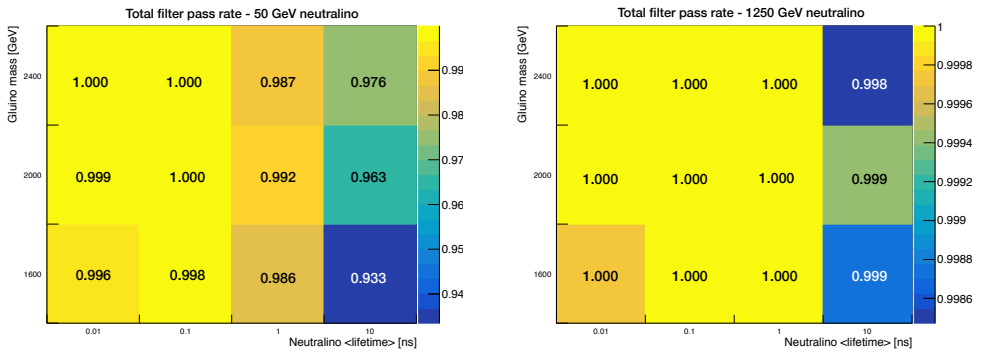
(a) Pass rate for 50 GeV, long-lived neutralino. (b) Pass rate for 1250 GeV, long-lived neutralino.

Figure 4.11: For events that pass the DVMultiJetFinalFilter, the two plots in this figure show the fraction that passed exclusively due to the addition of the high- p_T jet filters efficiency for two neutralinos of different mass.



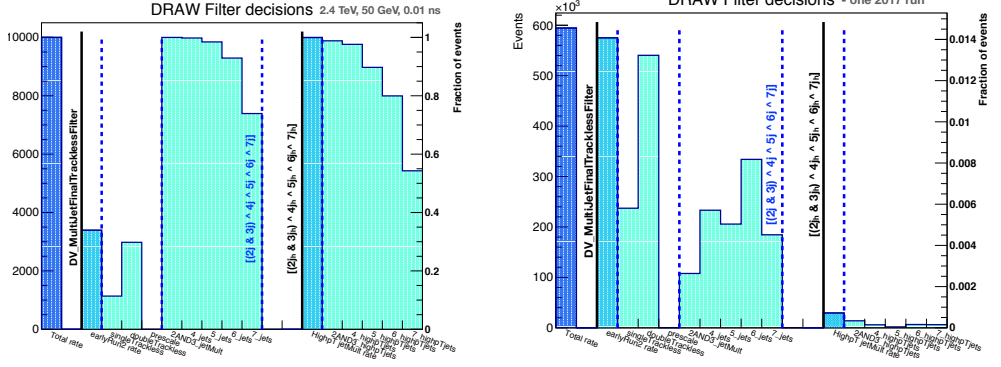
(a) High p_T filter efficiency for a 50 GeV neutralino. (b) High p_T filter efficiency for a 1250 GeV neutralino.

Figure 4.12: The high p_T jet filter efficiency for the two neutralinos of different mass.



(a) DVMultiJetFinalFilter efficiency for a 50 GeV neutralino. (b) DVMultiJetFinalFilter efficiency for a 1250 GeV neutralino.

Figure 4.13: The overall DVMultiJetFinalFilter efficiency for the two neutralinos of different mass.



(a) Event rate acceptance broken down into different filter flags. The signal simulated is a 2.4 TeV gluino decaying to a 50 GeV neutralino with a lifetime of 0.01 ns. Ten thousand events were passed through the filter.

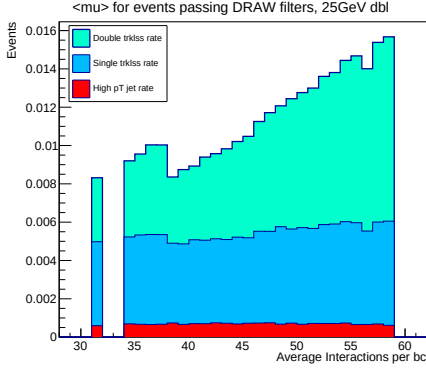
(b) Event rate acceptance broken down into different filter flags. This plot was made from one 2017 run lasting approximately 10 hours that produced around 46M events in the Main Physics stream.

Figure 4.14: Comparison of the rate of event acceptance in (a) simulation and (b) data. The light turquoise bins show the flags that make up the filter whilst the medium blue bins show the overall rate for first the trackless filter and then the high p_T filter. The dark blue bin is the DVMultiJetFinalFilter overall rate. Dashed vertical lines separate the flags from the summary rates or separate different aspects of the filter and solid vertical lines separate flags for the trackless filter and the high p_T filter.

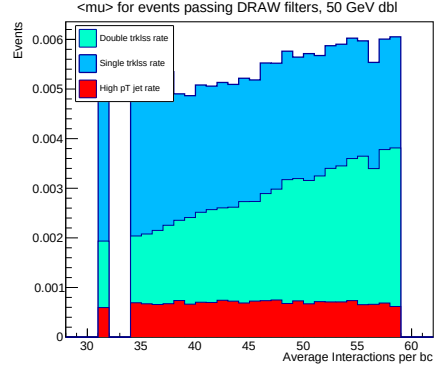
jet flag p_T threshold from 25 to 50 GeV as well as the introduction of the high p_T flags to produce the DVMultiJetFinalFilter. The high p_T flags were used to reprocess data collected by the ATLAS experiment from 2015–2017 and the DVMultiJetFinalFilter, including the increase of the double-trackless p_T threshold, was incorporated into the DRAW filter for 2018 data-taking. These changes improved the overall sensitivity of the filter to long-lived particles and helped reduce the pile-up dependance and overall rate of the filter.

4.3.2 Preselections and baseline DV-based selections

After the event has passed the DRAW filter and undergone LRT and secondary vertex reconstruction it undergoes preselections before being included in the analysis. First, a series of quality measurements are made that check whether the detector was functioning as expected during each run. This is done by combining information from online monitoring of the ATLAS detector during data-taking, and then twice after data-taking during offline monitoring [152]. Offline checks include monitoring each subdetector for abnormal occupancy, indicative of noise or failures, and also reconstructed objects such as jets or MET distributions. The first instance checks 10% of the data that is processed in an express stream and is typically reviewed within one day of the data run. Any consistent problems that hint at, for example, masking consistent detector noise or a different calibration, are noted and an attempt at correcting for them is made when processing the remaining 90% of the data. This remaining data is reviewed within two days of data-taking and any remaining defects that may cause the run, or specific LB's within it, to degrade the quality of the data are marked with a major defect. Entire runs or lumi-blocks within runs that do not contain a major defect are added to the 'Good Runs List', or GRL. If a run, or specific lumi-blocks within a run are not present in the GRL, then they

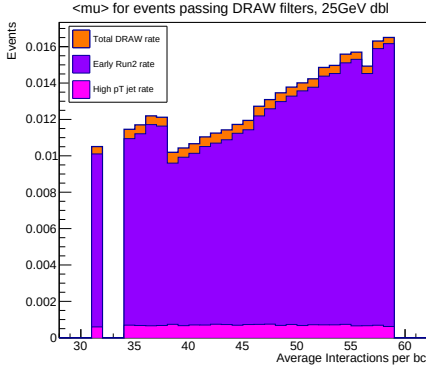


(a) Double-trackless flag has a minimum p_T threshold of 25 GeV.

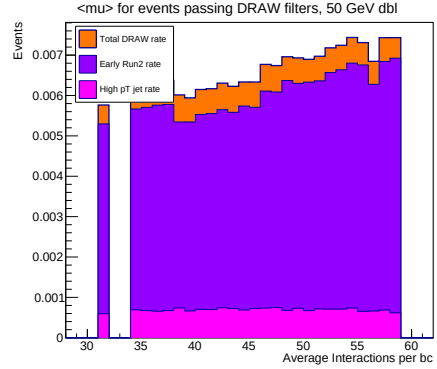


(b) Double-trackless flag has a minimum p_T threshold of 50 GeV.

Figure 4.15: Overlaid histograms showing the acceptance rate of three of the flags that comprise the DVMultiJetFinalFilter: the single-trackless low p_T , double-trackless low p_T , and high p_T flags, plotted against the average number of interactions per bunch crossing. The two plots differ in the minimum p_T threshold for each jet in the double-trackless flag.



(a) Double-trackless flag has a minimum p_T threshold of 25 GeV.



(b) Double-trackless flag has a minimum p_T threshold of 50 GeV.

Figure 4.16: Overlaid histograms showing the acceptance rate of the DVMultiJetFinalTracklessFilter, the high p_T flag, and the logical OR of the two, the DVMultiJetFinalFilter plotted against the average number of interactions per bunch crossing. The two plots differ in the minimum p_T threshold for each jet in the double-trackless flag.

are not selected for the analysis.

Vetoing at the event-level based on the defects set during data quality control does not happen when producing the GRL, so events are also checked during preselections for defects. This could be, for example, large amounts of noise in the calorimeter, or if the tracking detectors were not functioning for a couple of events but not an entire LB.

The remaining events are then checked for the presence of a reconstructed primary vertex (PV) to ensure a hard-scatter was recorded.

Baseline DV-based selections

First the event is checked for the presence of a reconstructed vertex with a high goodness-of-fit, $\chi^2/nDof < 5$, to minimise fake vertices. The reconstructed vertex must be within the fiducial region, which has an outermost boundary defined as $R < 300$ mm and $|z| < 300$ mm, a constraint set by the DV reconstruction efficiency. In addition, the DV must be at least 4 mm from any primary vertex in the event, both to avoid confusing the DV with one of those vertices and also to reduce the heavy flavour background which could contaminate the signal.

4.3.3 Offline jet calibration and selection

This section will briefly outline the calibrations applied to in events passing the DRAW filter and preselections, followed by a description of the event-level jet multiplicity and p_T cuts required for an event to pass the signal selections.

Offline jet calibration

Following preselection, the jets built from EM-calibrated, small- R (0.4), topo-clusters are corrected for pile-up dependency and other calibrations are applied:

- **Pile-up correction:** depending on how the jet is formed, additional corrections to account for additional activity in the event may be applied after jet formation. This is done by subtracting off a correction proportional to the jet area from the jet four-momentum vector [153]. The p_T of the jet will be modified by this correction.
- **JES and GSC:** the Jet Energy Scale (JES) and Global Sequential Calibration (GSC) first correct the jet energy according to where it was detected in η , and then apply a series of corrections to account for effects such as punch-through.
- **Smearing:** for simulated jets only, the four-momentum vector is smeared to account for the jet energy resolution of the detector.
- **Update JVT:** The Jet Vertex Fraction is then updated given the corrections applied to the the jet collection. The use of JVT in this analysis will be discussed in section 4.5.

A description of how these calibrations are applied is given in the section on object reconstruction: 1.3.

The calibrated jets are then checked against other reconstructed object collections, for example electrons and muons, to see if the objects could have been mis-identified. To determine this, each collection of objects is compared to one another in a pairwise fashion and, if within a given proximity, removed according to various conditions which are unique to each object. This is known as overlap removal (OR). In this analysis standard OR is applied to jets except with

⁴Recall that the $\Sigma^{\text{trk}} p_T$ is the sum of the p_T of all tracks associated both to the jet and, unless specified, the PV in an event, see 1.3 for details

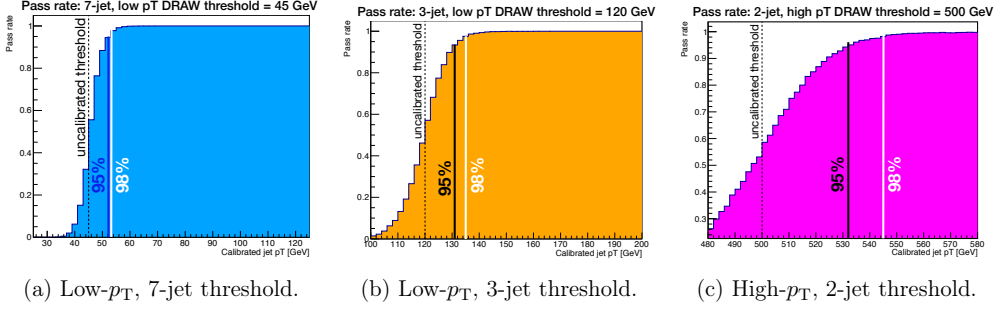


Figure 4.17: The ‘turn-on’ curves mapping the uncalibrated jet p_T (shown by dashed line) to the calibrated jet p_T , for various thresholds used by the DRAW filter flags. The 95% and 98% points are shown by vertical lines on each plot.

regards to muons. This is because the criteria for rejecting a jet, assuming it has been mislabeled as a muon, is based on the object’s associated number of tracks and the $\Sigma^{\text{trk}} p_T^4$. As some of the jets in the event may be trackless due to a LLP decay not being associated to the PV, this would cause jets to be removed and mis-labelled as muons. As such the jet removal, in jet-muon OR, is not used in this analysis by default. Studies to determine the rate at which this increases jet-misidentification and how this affects the analysis are currently underway.

The jet collection is then further modified in a process called ‘cleaning’. Cleaning is applied at various working points, which denote which specific flags are included in the process. Flags may suggest removing a jet from the collection because of properties in the jet indicative of noise in the calorimeters, for example, or that the energy deposits come from non-collision background events. This is broadly done in three ways: checking for abnormal pulse-shapes in the signal from the LAr calorimeter, comparing the amount of energy deposited in different layers of the calorimeters, and by using track-based variables [154]. For this analysis, some of the track-based cleaning flags used in the Loose and Medium working points use a variable called the charge fraction which is defined as:

$$chf = \frac{\Sigma^{\text{trk}} p_T}{p_T^{\text{jet}}} \quad (4.3)$$

and is the fraction of the jet p_T that is associated to PV-vertexed reconstructed tracks. The chf is used in jet cleaning in two ways:

- For jets with $|\eta| < 2$, where the fraction of energy deposited in the EM calorimeter (EMFrac) $< 5\%$, the jet is removed if $chf < 5\%$.
- For jets with $|\eta| < 2.4$, if the chf is less than 10% of the maximum fractional energy, f_{max} , deposited in a layer of the calorimeter the jet was reconstructed in.

As with OR, this is problematic for certain long-lived particle decays as the jets produced in the decay may have a low $\Sigma^{\text{trk}} p_T$, incorrectly causing the jet to be ‘cleaned’ from the event. As such, two working points for jet cleaning are currently being studied that do not employ chf as a discriminating variable: LooseBadLLP, and VeryLooseBadLLP. Whilst the former simply removes cuts using chf , the latter removes all cuts and only cleans jets using f_{max} and the fraction of energy deposited in the EM calorimeter. Studies are currently underway to understand the effect of both jet cleaning working-points on signal efficiency in this analysis. The combination of calibrations, OR, and cleaning lead to a collection referred hereafter as the calibrated jets.

Uncalibrated jet p_T [GeV]	Calibrated jet p_T [GeV]		DRAW filter that uses this threshold
	95% working point	98% working point	
45	52	53	Low- p_T 7-jet*
50	57	59	Low- p_T 6-jet, double-trackless
70	80	82	single-trackless
75	85	87	High- p_T 7-jet, low- p_T 5-jet
100	111	115	High- p_T 6-jet, low- p_T 4-jet
120	131	135	Low- p_T 3-jet*
170	185	189	High- p_T 5-jet
180	195	201	High- p_T 3-jet
220	237	244	High- p_T 4-jet, low- p_T 2-jet
500	532	545	High- p_T 2-jet*

Table 4.3: The uncalibrated jet p_T and calibrated jet p_T for various working points, used by the DRAW filter flags.. [*] denotes the turn-on curves plotted in Figure 4.17.

Jet-based, event-level signal selection

The multi-jet triggers inform the DRAW filter p_T and multiplicity thresholds, which in turn are used to the jet multiplicity thresholds as part of the offline signal selection. These selections differ to the p_T and multiplicity thresholds used in the DRAW filter as both can be altered during the jet calibration process, which is applied to uncalibrated jets that have already passed the filter. The thresholds are selected by picking a p_T that uncalibrated jets passing the DRAW p_T requirements have a high probability of passing once calibrated, which is shown for three different uncalibrated thresholds in Figure 4.17. The plots are normalized to the number of calibrated jets at each p_T . For example, the 6-jet, high- p_T DRAW threshold is 100 GeV but after calibration some of the jets are actually found to have a p_T as low as 90 GeV and about 5% of those pass the DRAW cut. So, a working point for the calibrated jet p_T threshold is picked that is well into the plateau of the sigmoid distribution in the plots. The value chosen for each multiplicity can be chosen to be more inclusive, and thus have a larger associated uncertainty, i.e. picking a lower calibrated jet p_T , or a larger value, in the plateau, that is less inclusive but has a smaller uncertainty. The calibration applied is different when comparing data and simulation. Table 4.3 shows two possible working points for the calibrated jet p_T cuts for a given event multiplicity, for simulated events, which are indicted in Figure 4.17 by vertical lines. The values were produced from reviewing the combined, fitted histograms for $\mathcal{O}(1M)$ jets in 166 different signal models, to give a full range of jet multiplicity and p_T . The values in this table may change slightly in the final analysis following the conclusion of the calibration studies for jet cleaning and OR discussed earlier.

With the offline jet thresholds set, the event can either pass the selection by passing one of the ‘high- p_T ’ multi-jet thresholds or by passing one of the low- p_T multi-jet thresholds AND either the single- or double-trackless jet requirement, both of which are discussed in detail in section 4.3.1. The offline jet thresholds are shown with their corresponding DRAW flags in Table 4.3.

4.3.4 Material veto of hadronic interactions

Hadronic interactions present a significant background as they can act as a seed to a high-mass vertex when paired with an accidental crossing track. As such, part of the event selection is to remove vertices reconstructed at places of known material with a 3D map. The map is constructed using both hadronic interaction data as well as detailed simulations of the detector material.

The construction is split into two schemes, which divide the map between the ‘Inner’ and ‘Outer’ volumes of the Pixel detector, where ‘Inner’ denotes material at radii, $R < 150$ mm, and ‘Outer’ denotes material within a radius of $(150 \leq R < 300)$ mm. Both sections extend out to $|z| < 300$ mm. The Inner volume has a high rate of material interactions allowing the map to be entirely data-driven, whilst the Outer volume has lower rate of material interactions and is supplemented by the detector simulation. The detector simulation is not a perfect facsimile of the real detector, so any apparent discrepancies between the simulated detector and what is observed in the Outer region always give preference to the real detector, e.g. if there is evidence of material interactions in data but the simulation does not indicate material, preference will be given to what is seen in data and that bin will be added to the veto.

The general technique to form the map involves pre-selecting low mass, low multiplicity displaced vertices in data, binning their location, applying a threshold to the histogram, and smoothing the resulting map, which favours high density vertices to build up the material structure. Vertex selections are as follows:

- Vertices must have an invariant mass less than 2.5 GeV to reduce contamination from high-mass background sources, such as accidental crossings, or vertices from the signal region.
- Next, vertices from the decay of Standard Model long-lived particles are removed. First, a specific cut focusing on reconstructed K-short mesons is applied by removing 2-track vertices with an invariant mass in the range $[0.45\text{--}0.55]$ GeV. Next a more general cut is applied to account for other long-lived Standard Model particles, such as Lambda baryons, K-long mesons, or partially-reconstructed B- or D-hadrons, and also photon conversions. The cut requires that any 3-track vertices used in the map must have an invariant mass greater than 0.95 GeV, and that the minimum opening angle between any two tracks in the vertex must exceed 0.15 radians.
- To reduce the number of fake vertices constructed from ghost tracks, none of the tracks in the vertex can have a hit missing after the vertex position.
- Additionally, the number of fake vertices is reduced by requiring the direction vector calculated from the momentum of the attached tracks and the direction vector that passes through the primary and displaced vertices are no more than 0.5 radians apart.

The location of selected vertices is then binned at different granularities, depending on their location in the detector with finer binning used at smaller radii. To increase the available statistics for the map, only fixed segments of the volumes are used to take advantage of the approximate ϕ -symmetry of the detector. The amount of folding that is done changes by radius to account for both the varying density of secondary vertices, and the varying symmetry of the detector. The four layers of the Pixel detector are folded: 7, 11, 19, and 13 times from the innermost to the outermost layers, respectively. As not all detector layers are perfectly centered with respect to the beamline, small translations between 0.1–1 mm in the transverse plane are required for different layers before folding.

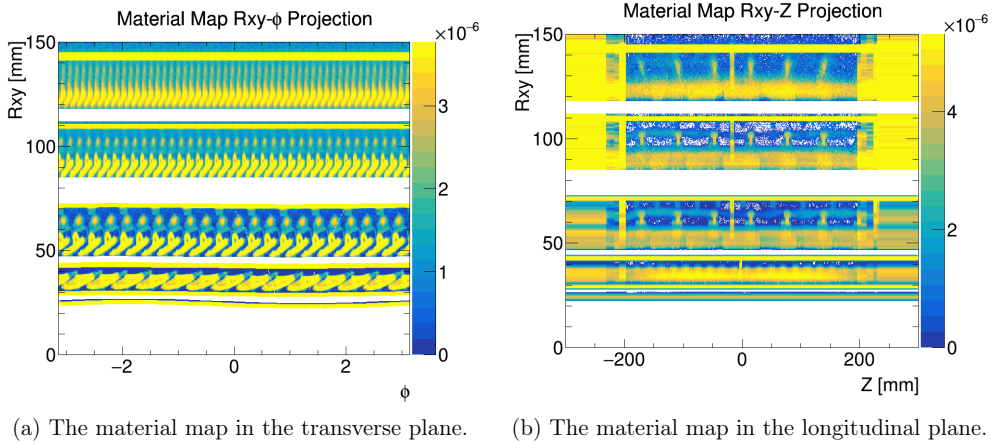
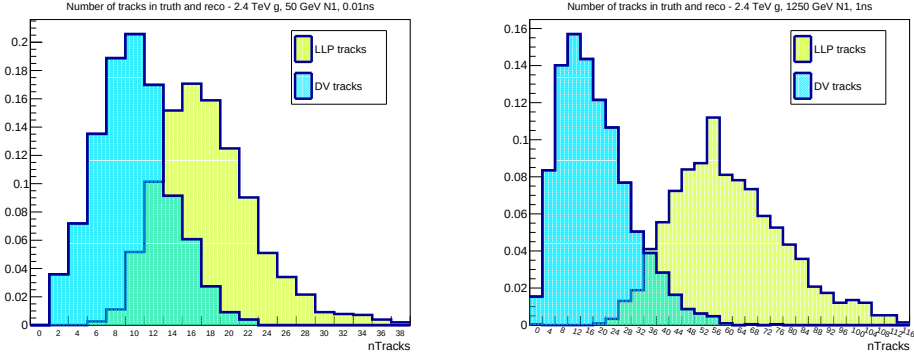


Figure 4.18: Inner volume material maps, the bin content is normalized to the number of vertices in the map. This map was constructed from data collected from the ATLAS experiment in 2018. Plots courtesy of Filip Backman.

A threshold is applied to the map and bins exceeding it are deemed to contain material. For the Outer volume, the simulated material is OR-ed with the data map. Bins passing these requirements are then included in the material veto which excludes approximately 42% of the fiducial volume. Figure 4.18 shows the material map in the transverse (a) and longitudinal slice (b) for the Inner volume. Structures in each Pixel detector layer, such as the titanium cooling pipes, can be clearly seen.



(a) Number of tracks attached to a 50 GeV neutralino with a 0.01 ns lifetime.

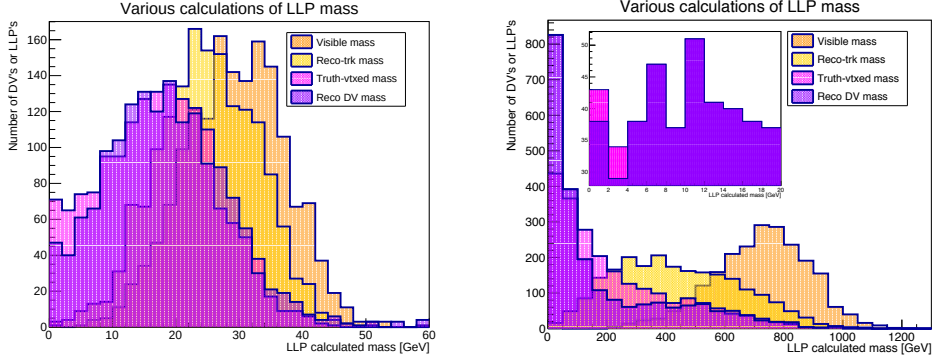
(b) Number of tracks attached to a 1250 GeV neutralino with a 1 ns lifetime.

Figure 4.19: The number of tracks attached to a simulated neutralino for two different mass and lifetime parameters. Each plot shows two histograms. The first, marked ‘LLP tracks’, is the number of charged, i.e. reconstructable, hadrons produced in the simulated decay. The second, marked ‘DV tracks’, is the number of tracks actually attached to the reconstructed displaced vertex.

4.3.5 Displaced vertex formation and mass recovery

In addition to passing the material veto, the DV-based signal selections require that at least one reconstructed DV is present with a mass greater than 10 GeV and at least 5 attached tracks. The selections applied to the DV are relatively simple, as there are limited physical processes in the Standard Model that could produce a massive, multi-track displaced vertex within a multi-jet event. The displaced vertex (DV) mass and track-multiplicity selections for this analysis use the working points from the upon the 2017 search for long-lived particles in events containing a DV and MET (DV+MET) as a baseline [139]. In this section some initial studies exploring multi-track DV reconstruction efficiency and efficacy, using the recently updated secondary vertex reconstruction package for the analysis’s baseline signal model, will be discussed. The package was altered between the DV+MET analysis and this work to improve reconstruction inefficiencies due to disabled modules, as well as changing the way vertices were formed to reduce vertex splitting [52], see Section 1.3.1 for details.

Figure 4.19 shows the number of attached tracks for neutralinos simulated at two different masses and lifetimes. Each plot shows two distributions labelled ‘LLP tracks’ and ‘DV tracks’. The first is produced using information about the simulated particles, or ‘truth’ information. The histogram shows the number of charged hadrons produced in the neutralino decay and can be thought of as the best possible track attachment the reconstruction software can achieve. The ‘DV tracks’ histogram shows the number of tracks actually attached to the reconstructed DV and does not make use of any truth information. It is clear that track reconstruction/attachment does not recover all truth particles in the decay but appears to perform more poorly for the higher-mass neutralino. Despite this, in both cases greater than 93% of the reconstructed DV’s pass the baseline track-multiplicity cut. But what about the mass cut? And why are there less tracks attached to the reconstructed DV than there were charged decay products in the event? These questions are explored in the section below.



(a) Various calculations of the mass of a 50 GeV neutralino with a 0.01 ns lifetime, where the 10 GeV baseline threshold visible.

(b) Various calculations of the mass of a 1250 GeV neutralino with a 1 ns lifetime, inset shows finer binning around 10 GeV baseline threshold.

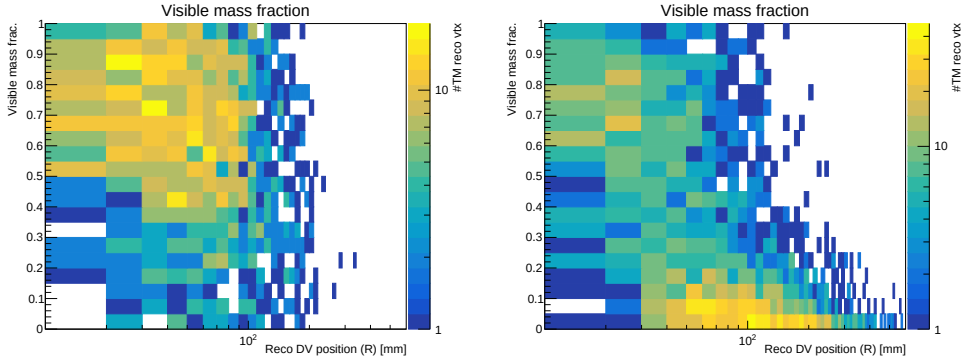
Figure 4.20: Plots with four histograms showing (from right to left) the: true visible mass, reconstructed track mass, truth vertexed mass, and the reconstructed DV mass - as defined in main text. The neutralinos simulated are produced in a 2.4 TeV gluino decay. For each mass range, the number of DV's (or LLP's in the case of the true visible mass) in the fiducial region with that mass are shown. The plot in (b) additionally contains an inset, zoomed in between $[0, 20]$ GeV, to better display the number of reconstructed DV's that do not make the 10 GeV baseline mass threshold.

Displaced vertex mass reconstruction

The mass of the reconstructed displaced vertex can be calculated at several steps during the reconstruction process, with the aid of simulated truth information:

- The true **visible mass** is calculated from the invariant mass of the simulated LLP decay products.
- The **reconstructed/recovered** track mass is calculated by taking the invariant mass of all reconstructed tracks truth-matched to LLP decay products, regardless of whether they are vertexed or not.
- The **truth vertexed** invariant mass of reconstructed & vertexed truth particles, i.e. not using the four-momentum calculated from the track fit but instead the truth four-momentum.
- The **reconstructed DV** mass which is the invariant mass of the reconstructed, vertexed tracks and only uses information from reconstructed objects, not the underlying truth information. Hence it is the only one of the four masses listed that is available without truth information, i.e. during data-taking.

These four masses are plotted in Figure 4.20 for the same two simulated gluino-neutralino parameters as in the previous section, for long-lived neutralinos that decay within the fiducial region. Each plot shows that whilst track reconstruction is not perfect at recovering all of the charged hadrons produced in the neutralino decay it is the track attachment (the difference between the 'Reco-trk' mass and the 'Reco DV' mass) that lowers the reconstructed DV mass. This is clearly a more significant effect in plot (b) which already suffers from less reconstructed decay products, as was seen in Figure 4.19. However the affect of mass loss affects the lower mass



(a) Visible mass fraction in R of a 50 GeV neutralino with a 0.01 ns lifetime.

(b) Visible mass fraction in R of a 1250 GeV neutralino with a 1 ns lifetime.

Figure 4.21: The change in visible mass fraction as a function of reconstructed long-lived particle decay position in the transverse plane, R , for two simulated long-lived particle signals. For each signal, a 2 TeV gluino decays to a long-lived neutralino in either an uncompressed regime with a 10 ps lifetime (a) or a compressed regime with a 1 ns lifetime (b).

neutralino more with regards to passing the baseline cutflow. If all of the 1,597 visible hadronic decay products simulated had been reconstructed and attached to a DV with 100% efficiency, less than half a percent would not have passed the 10 GeV mass cut. However, because of mass loss during the reconstruction process, the fail rate increases to $\sim 20\%$. For the 1250 GeV neutralino, assuming ideal reconstruction, none of the 2,214 visible displaced vertices will fail the baseline cut but, because of reconstruction inefficiencies, $\sim 10\%$ do. Appendix C.4 breaks down these efficiencies and shows plots that further support that track attachment leads to the most significant mass loss in the reconstructed DV.

Figure 4.21 shows the visible mass fraction for the two simulated neutralinos. Note that the fiducial requirement is placed upon the simulated LLP and not on the reconstructed DV it is associated with, hence some of the DV's are reconstructed to lie outside of $R < 300$ mm. The plots show the visible mass fraction, which is defined as the invariant mass of all tracks attached to the DV divided by the invariant mass of the LLP's truth decay products. A p_T cut of 1 GeV is applied to the truth particles to give reasonable accommodation to the minimum p_T requirements in forward and backward tracking in both standard and LRT reconstruction, see Table 1.4 for details, which might over-estimate the visible mass fraction but provides the same trend as with a lower p_T cut. In both plots the vertexing efficiency decreases with increasing R but at different rates. For example in the bin that contains LLP's decaying at 60–70 mm from the IP, the 50 GeV neutralino visible mass fraction peaks around 55–60% but has a broad spread of values up to 90%. For the heavier, 1250 GeV, neutralino the visible mass fraction in this bin peaks between 5–10%, with a narrower spread of values at around 10%. This trend is consistent with the observations the vertexing performance note for a hadronically decaying long-lived R-hadron.

The reason for poor track attachment was investigated by studying various properties of both the DV and the tracks that comprise it in cases where the DV mass was and was not faithfully reconstructed. A few plots showing the results of those studies can be found in Appendix C.4 but none pointed towards a clear reason for poor track attachment other than that the attachment becomes worse the more displaced the vertex. A rudimentary visualization of an event was made for cases with DV mass less than 10% of the true visible mass, shown

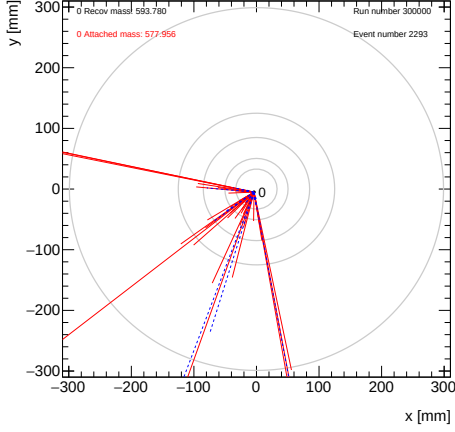
in Figure 4.22. The event displays plot tracks reconstructed from LLP decay products in the transverse plane with the Pixel barrel layers and first SCT barrel layer shown as concentric grey circles. The visualization displays reconstructed tracks as either attached (red, solid lines) or unattached (blue, dashed line) to the DV. The length of track is proportional to its p_T , which is calculated at the perigee with respect to the PV. All tracks are displayed as coming out of a single point (the fitted DV location) but in actuality this is only true for the attached tracks. The invariant mass of the DV, assuming all reconstructed tracks are attached, is written in purple on each plot, the actual DV mass, calculated using only the tracks attached to the DV, is written in red, both are in units of GeV. All tracks are displayed as coming out of a single point (the fitted DV location) but in actuality may differ slightly from this placement, this is particularly true for the unattached tracks. Angles are displayed as measured with respect to PV, in actuality may be slightly smaller due to boosted system. The tracks are truth-matched to the LLP decay products.

Plot (a) shows an event with one LLP decaying within the fiducial region that has several unattached tracks of various momentum, however these unattached tracks only cause minimal mass loss and 97% of the reconstructable DV mass is recovered. Plot (b) shows an event with one LLPs decaying within the fiducial region with significant mass loss (94%) due to track attachment. The reason for mass loss based on this event display alone seems clear: three sprays of particles with varying p_T are not attached to the DV. However, in both of these plots it is also not clear why the tracks are not attached to the DV in the first place: there are a range of p_T values so it is not due to momentum⁵, nor are only high angle tracks unattached, or tracks that point in a specific direction. This is particularly evident in plot (b) which shows a range of track momenta, directions that are unattached, including some tracks that are seemingly part of the same jet as attached tracks.

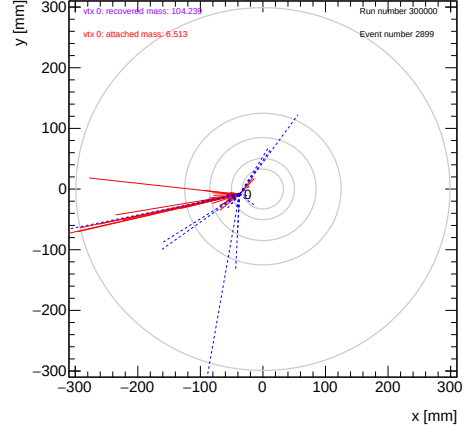
Plots (c) and (d) provide a hint that something is going wrong with the track fit in DV reconstruction. Track attachment proceeds by using the track kinematics fitted at the perigee point with respect to the seed DV. In Figure 4.22c the track parameters are calculated with respect to the PV perigee point and in 4.22d the track parameters are calculated with respect to the PV perigee point. It was found that for some tracks, the track fit failed resulting in a track p_T set to zero, which is why the attached tracks appear to vanish when calculated with respect to the DV. This does not explain why tracks are unattached but does hint at a problem in the reconstruction procedure for some tracks.

Figure 4.23 provides another clue as to why some tracks are not attached, using the significance of the impact parameter measurements with respect to the DV. Significance here is defined as the measured impact parameter, e.g. d_0 , weighted by the uncertainty on the measurement, e.g. $\sigma(d_0)$. The plots were produced by simulating 50 events for the same parameter grid as the event displays: a 2.4 TeV gluino decaying to a 1.25 TeV neutralino with a 1 ns lifetime. The events underwent a modified version of the secondary vertex reconstruction to calculate impact parameters of all truth-matched LLP tracks to the relevant reconstructed DV, regardless as to whether they were attached to it or not. The tracks are truth-matched to LLP children to ensure no fakes or PU tracks are present in the distribution, and have a p_T defined at the PV perigee point of at least 1 GeV. Additionally, only DV's within 1 mm of the truth LLP decay vertex were considered. Plot 4.23a is a scatter of the longitudinal vs. transverse impact parameter significances for attached (black, hollow markers) and unattached (red, solid markers) tracks that were reconstructed from LLP decay products. The dotted horizontal and vertical lines show the significance attachment threshold of 5. The scatter shows a broad spread of significance values for unattached tracks, those that have a significance less than both thresholds often fail the pattern requirement so it is not unexpected that unattached tracks

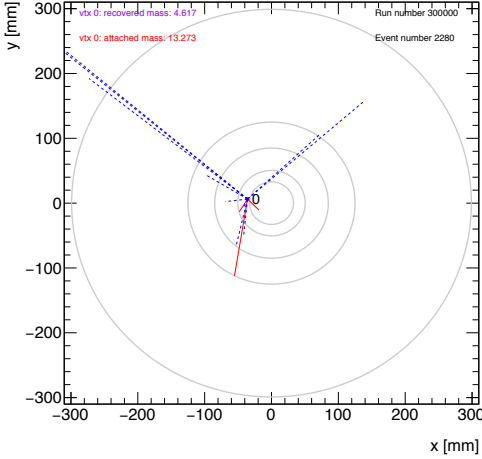
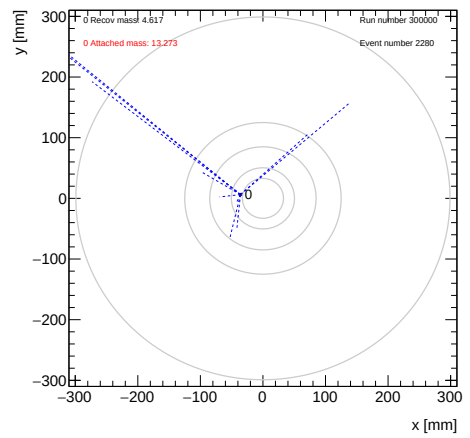
⁵Which is also clear from studies of each track's p_T spectrum, see Appendix ?? for details.



(a) DV with 3% mass loss.

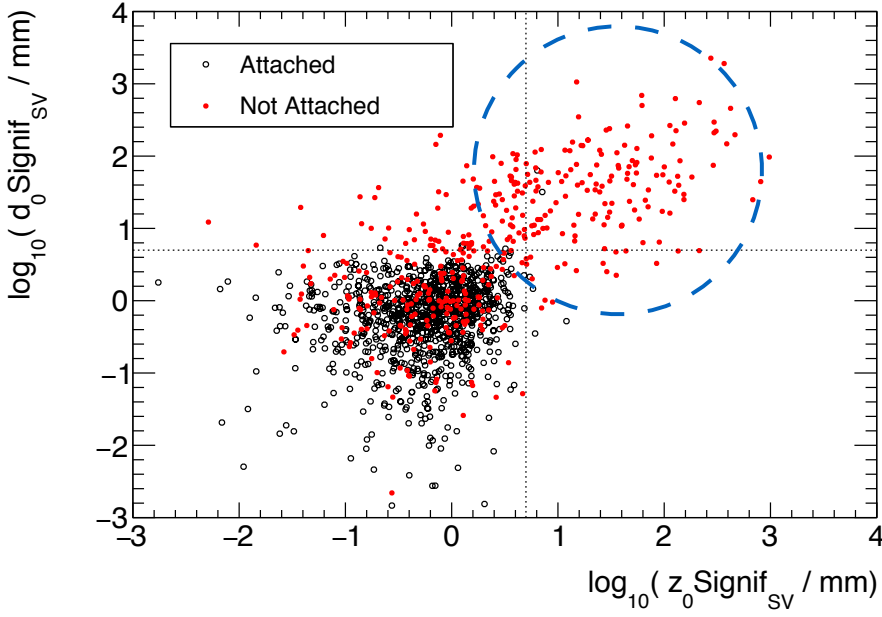


(b) DV with 94% mass loss.

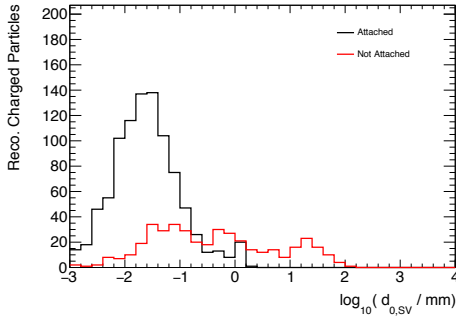
(c) This plot shows an event with mass loss of 66% from sprays of high- p_T tracks. Kinematics calculated with respect to PV perigee point.

(d) This plot shows the same event as (g) but with kinematics calculated using track-fit with respect to the DV perigee point.

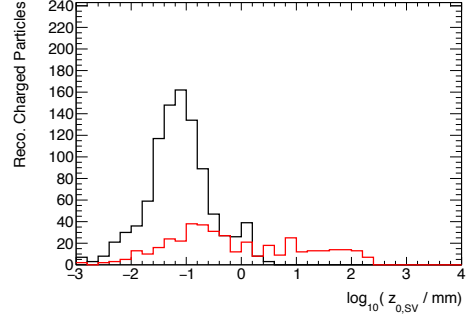
Figure 4.22: Event displays showing reconstructed tracks that are either attached (red, solid lines) or unattached (blue, dashed line) to the DV, where length is proportional to p_T . The inv. mass of the DV is written assuming all reconstructed tracks are attached (purple) and only red are attached (red). Both masses are given in units of GeV.



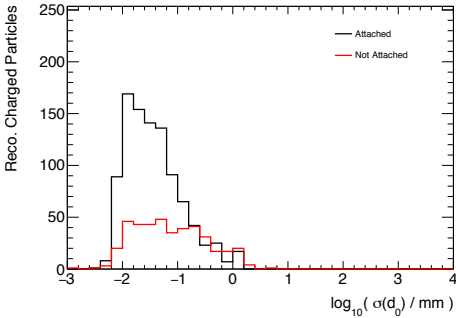
(a) A scatter plot showing the z_0 vs d_0 significance of reconstructed tracks either attached (solid, red markers) or unattached (black, hollow markers) to the DV. The dashed vertical lines show the significance threshold for track attachment.



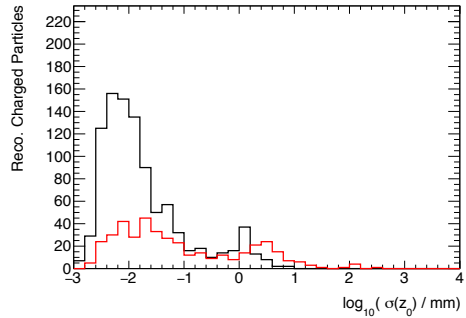
(b) The d_0 of reconstructed tracks.



(c) The z_0 of reconstructed tracks.



(d) The uncertainty on d_0 of reco. tracks.



(e) The uncertainty on z_0 of reco. tracks.

Figure 4.23: The significance and uncertainties of the track impact parameters, measured with respect to the DV for both attached and unattached tracks. All plots courtesy of Hide Oide.

end up in the bottom left hand corner of the plot. What is less expected is the magnitude of the significance for unattached tracks that exceed the threshold.

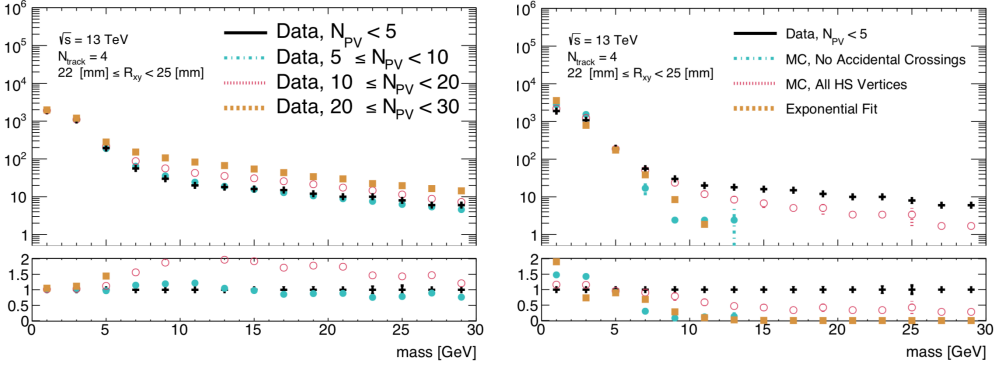
Plots 4.23b and 4.23c show the transverse and longitudinal impact parameters, respectively, whilst 4.23d and 4.23e show their uncertainties. Significance is inversely proportional to uncertainty, so if the uncertainty were underestimated it could lead to anomalously large significances seen in plot 4.23a. However the plots in 4.23d and 4.23e show that, for both impact parameters, attached and unattached tracks have very similar uncertainty distributions. The longitudinal impact parameter has a longer tail into larger uncertainties than the attached tracks which accounts for the occupancy of the left half of the scatter plot but does not explain the overall trend. Plots 4.23b and 4.23c however show that some of the unattached tracks have unreasonably large impact parameters, recalling that both the tracks and the DV position are truth-matched to LLP decay products. This indicates, as with the zero-valued p_T seen in Figure 4.22d, that there is a problem with the function that calculates the value of the track parameters at the DV perigee. At this stage it was deemed that further investigation was beyond the scope of the analysis and was passed on to the authors of the secondary vertex package and the tracking performance group. As it stands, the inefficiencies with track attachment reduce the DV acceptance by between 10–20%, depending on the parameters chosen. Understanding how this affects the event-level signal efficiency is currently in progress.

4.4 Background rejection and estimation

Searches for displaced, multi-track vertices benefit from a small background caused predominantly by low-mass vertices being promoted to a higher mass by an unrelated track. Using the baseline cuts described earlier, these backgrounds have been reduced to less than one event in recent DV analyses [138] [140] [139]. As such the background estimation technique first assesses the rate of background-DV formation for vertices that satisfy all baseline selections and then extrapolates to estimate the rate of background events for DV's satisfying all signal selections. The three main sources of background-DV formation are listed below:

1. **Hadronic interactions** occur when particles produced in the event interact with the nuclei of detector material or gas. These interactions can produce multi-track vertices that could be confused with a signal vertex and are dealt with in two ways: a veto of part of the fiducial region where there is known material and a background estimation technique for vertices from gas outside of the detector material.
2. **Merged vertices** occur when displaced, low mass vertices, such as reconstructed k-short decays, are in close proximity and end up being merged together to form a higher mass vertex.
3. **Accidental crossings** refer to any low mass, displaced vertex that would not satisfy the signal selections, that is promoted to a higher mass and track multiplicity by an unrelated track passing within attachment proximity. In previous analyses this source of background has dominated the other two in the signal region, once the hadronic interaction veto is applied.

The rest of this section describes the techniques used to reject and estimate these different sources of vertex background and concludes with an outline as to how these background vertex estimations are translated into an event-level background estimation in the signal region.



(a) The variation in the invariant mass of displaced vertices according to pile-up conditions, measured in the number of primary vertices in the event, N_{PV} .

(b) The invariant mass of displaced vertices in both data and simulation where the simulation is run both with and without accidental crossing contributions.

Figure 4.24: The invariant mass distribution of displaced vertices in the beam-pipe region in data and simulation. Plots courtesy of Jennifer Roloff.

4.4.1 Hadronic interactions background estimation

For hadronic interactions that are not vetoed due to imperfections in the material map or due to gas between the detector planes, an estimation is made as to how many will contaminate the signal region. The technique uses the fact that, for displaced vertices with mass less than 10 GeV, the background is dominated by hadronic interactions, whilst accidental crossings dominate at higher mass. The Run 1 version of this analysis did not include an hadronic interactions background estimation outside of the material region but this was later introduced to the DV plus MET analysis in Run 2. That analysis fit part of the DV mass distribution, below 10 GeV, with an exponential function and used the extrapolation of that, above 10 GeV, to estimate the hadronic interaction contribution to the DV background. However, that analysis found that even at low masses the data used was contaminated with accidental crossings and that there were large uncertainties on the exponential fit in the low mass region.

Based off of the findings of the Run 2 DV plus MET analysis, this analysis instead opted to use templates built from simulation that could then be normalized to data in the low mass region. The templates will be built for different R , given the decreasing rate in accidental crossings in increasing R , and for different track multiplicities. This method suffers from additional complications, namely that:

- There is an associated uncertainty modeling hadronic interactions processes in simulation, more specifically Geant4, that will add to the uncertainty on this background estimation method.
- There is some choice about where in the simulation process the template creation should happen: either directly following the Geant4 simulation but before digitization, or after full reconstruction. In the former case, only particle interactions with large detector volumes, not indicative of the real detector granularity, are available. The lack of digitization and reconstruction means that the template will have an associated uncertainty based on the efficiency of digitization (i.e. simulated detector response) and reconstruction to recover these vertices in the final event. In the latter case, when using a fully reconstructed event, additional measures must be taken to remove accidental crossing tracks from ver-

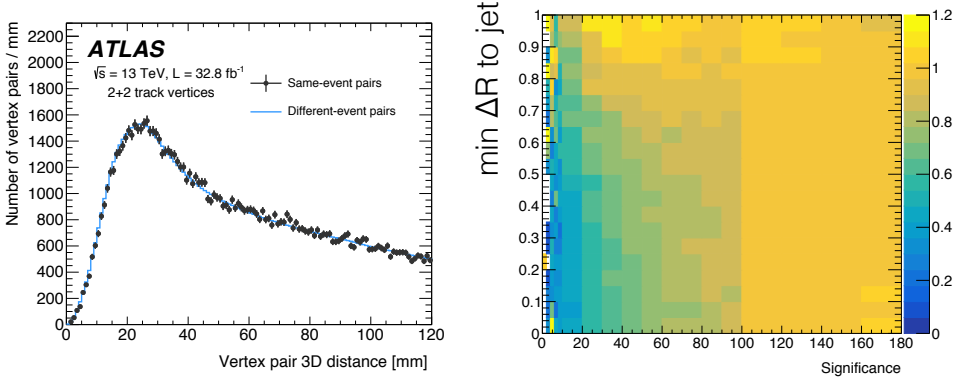
tices used in the distribution. However, since both data and simulation undergo the same reconstruction, using a fully reconstructed event reduces these uncertainties and provides the additional flexibility that the pile-up can be varied in the simulated event. Given these additional advantages the template method was built from reconstructed, low-PU, simulated events, normalized in the low-mass range.

The estimation is expected to be validated in two possible scenarios: using low-PU data events which reduce the rate of accidental crossing, and by inverting the material veto which should increase the number of hadronic interactions. However, recent studies have found that accidental-crossing must be taken into account even in the material region. The plot in Figure 4.24a shows the mass distribution of DV's, centered around the beam-pipe region, which should be dominated by vertices from hadronic interactions. Four sets of data are plotted in the figure, each with a different number of reconstructed PV's per event, analogous to the event PU. The plot shows that across all mass ranges, but particularly at masses above 5 GeV, a higher number of PV's yields a larger number of DV's for a given mass, indicating PU tracks affect the mass by accidental crossing even in the material region. This can also be seen in the plot in Figure 4.24b, which shows two simulated (MC) templates normalized to a single 2017 data run in the material-enriched region, an exponential fit of the data at the normalization range, [4–6] GeV, is also shown. The solid, blue circular markers in the plot show the number of purely hadronic simulated DV's in each mass bin, and the red, unfilled circular markers show all simulated hard scatter displaced vertices including those from accidental crossings. The data used in the plot only contains events with less than five PV's but still has better agreement with the more inclusive DV simulation. The exact normalization range and validation regions used for this background estimation method are still being studied but so far using a combination of reconstructed, simulated events to build mass templates is the expected strategy.

4.4.2 Merged vertices

It is possible that two unrelated displaced vertices, from background processes, could be merged due to proximity to form a high mass, high multiplicity vertex which would form part of the displaced vertex background. Previous analyses to this one used an earlier version of the reconstruction software. The software merged any vertices that were within 1 mm of each other, that may have been split artificially during reconstruction, at the risk of creating fake vertices [139]. As such, this background is estimated by comparing the rate at which n -track vertices from different events, i.e. two, uncorrelated, n -track vertices, will pass the merging criteria to understand how many uncorrelated vertices passed the merging criteria in the same event. Figure 4.25a shows the separation for 2-track vertex pairs within the same event and between different events, from a previous displaced vertex analysis that used the 1 mm separation as merging criteria. The plot is filled using pairs of displaced vertices, where the combined mass of the vertices must be greater than 10 GeV, to focus on DV's that pass the baseline selections other than track-multiplicity. The values in bins with separation less than 1 mm have no entries for the same-event histogram, as vertices that fulfill that criteria are always merged. As such, the number of entries in that bin for the mixed-events histogram gives an estimate of the rate of merging for uncorrelated vertices.

However, for this analysis the merging criteria in the most recent vertex reconstruction software is more complicated. In addition to vertices being merged if they are within 1 mm of each other, if they satisfy a number of criteria detailed in the section on tracking in chapter ??, they will also be merged, unless their separation significance is greater than 100. Here, the separation significance is the scalar displacement of the two vertices, weighted by the uncertainty on each the vertex's position [52]. There is a natural upper-limit to this uncertainty which is related to the vertex fit, as only vertices with a goodness of fit, $\chi^2/nDof < 5$ are retained during



(a) The binned separation between vertices in mixed- and same-event pairs of 2-track vertices in 2015+2016 data-taking, from earlier DV analysis [139].

(b) The ratio of same-event to mixed-event pairs as a function of proximity to the jet core and separation significance. Plot courtesy of Jennifer Roloff

Figure 4.25: Two separation metrics between pairs of displaced vertices used for the merged vertex background estimation.

reconstruction. If only the separation significance were used as a merging requirement, vertices that had a small separation but that were measured with high precision would not be merged. As the separation is not the only factor for deciding whether a pair of vertices will merge, using an identical method to the one shown is more complicated as additional correlations exist between vertex pairs in the same event. A study, currently underway, shows that vertices are more likely to be merged when reconstructed closer to the interaction point, or if within close proximity to a jet, see Figure 4.25b. The plot shows the proximity to the centre of the jet cone, ΔR , for a given vertex-pair significance. In this case, the same-event vertex pairs are normalized to mixed-events for significances > 100 , where merging is unlikely. Each bin is filled with the ratio of same-event pairs to mixed-event pairs. A smaller value in the bin means that there are less vertices in the same-event pairs than the mixed-event pairs, which indicates there were more events merged. If there were no correlations with significance and jet proximity, then this histogram should be close to one. This indicates that the separation significance should also be binned in minimum jet proximity. Studies to understand the correlations that need to be decoupled to use this method in light of the vertex reconstruction software modifications to estimate the merged vertex background are ongoing.

4.4.3 Accidental crossing, K_S estimate

A low-mass, displaced vertex can be promoted to a higher mass if it is crossed by an unrelated track at high angle that is attached to the vertex during reconstruction. This background is known as the accidental crossing background. The background is estimated using two methods, both of which were used in previous versions of displaced vertex analyses but which will have a different performance in this analysis due to the changes to the reconstruction software. In both cases, templates are produced for vertices crossed and not-crossed by a random track, normalized, and integrated in various slices of the fiducial volume which have different rates of accidental crossing. The calculated crossing factor assumes that the rate of crossing is independent of the track-multiplicity of the DV being crossed.

The first method uses a known displaced resonance, the k-short meson, K_S . The K_S has an

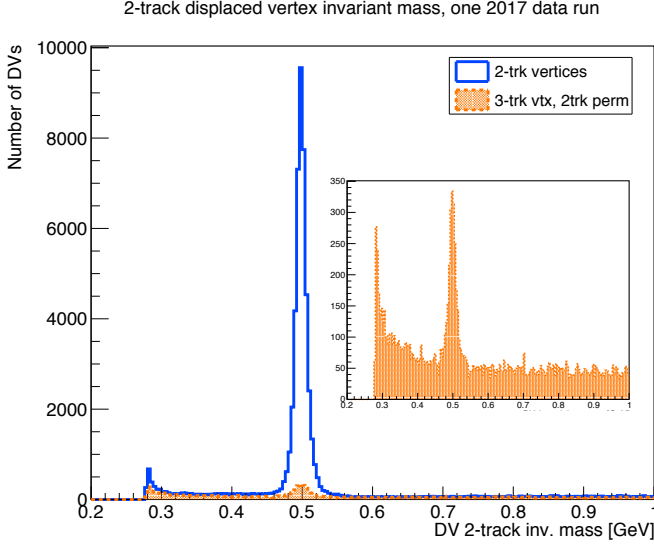


Figure 4.26: The invariant mass spectrum of two tracks in a vertex for 2-track vertices (solid, blue line, in overlaid histogram) and permutations of two tracks in 3-track vertices (dashed, orange line, both in overlaid histogram and inset) for 750k events from one 2017 data run. The inset histogram shows a subset of the y -range to show the overlaid histogram more clearly.

invariant mass of ~ 498 MeV and a mean lifetime of 90 ps, meaning that most decay within the first few layers of the Pixel detector volume. The K_S decays predominantly to two pions, and around 70% of those pions have an electric charge which results in displaced, 2-track vertices. The background estimation method first uses the invariant mass spectrum of 2-track displaced vertices to find the K_S mass peak. Integrating over the fitted peak gives the rate of reconstruction of K_S that have not undergone an accidental crossing. Next, the number of K_S that have undergone an accidental crossing is measured by plotting all permutations of two, of the three, tracks in 3-track vertices. If two of the tracks in the 3-track vertex actually come from the K_S decay, then the K_S invariant mass should show up in the mass spectrum. This means that the third track in the 3-track vertex is an accidental crossing track, illustrated in the cartoon inset of Figure 4.27. By integrating over the fitted peak in the 2-track permutations of the 3-track vertex mass spectrum, the portion of K_S undergoing an accidental crossing can be measured.

The invariant mass of two-track vertices in one 2017 data run is plotted in Figure 4.26. The vertices in the plot undergo all baseline selections imposed on the signal region. The plot shows two histograms, the blue histogram, with a solid line, is the invariant mass of 2-track displaced vertices and shows a clear peak centered on ~ 500 MeV, around the K_S mass. There is a kinematic cutoff at 280 MeV, which corresponds to the track mass being set to the pion mass during reconstruction. The second histogram, orange with a dashed line, is overlaid on first histogram and also shown inset with a smaller y -range. This histogram shows the two-track invariant mass of permutations of track pairs in 3-track displaced vertices. As such there are three entries made into this second histogram per vertex. The inset plot in the figure clearly shows the K_S peak, implying that some of the 3-track vertices are due to a K_S decay vertex being crossed by an unrelated track.

The integral of each of the fitted peaks, 50 MeV around the K_S mass, are combined to obtain

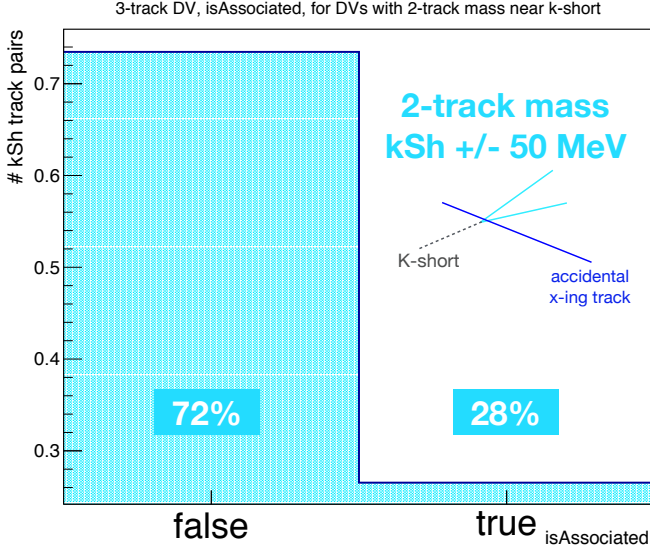


Figure 4.27: For 3-track vertices, where 2 of the 3 tracks have $(m - m_{K_S}) \leq 50$ MeV, it is assumed that the 3-track vertex is actually a K_S decay with an accidental crossing track attached. If either of the two tracks from the K_S decay were *not* used in the construction of the seed vertex but were instead attached after vertex formation, i.e. were *associated* to the vertex, the pair is marked as ‘true’, else the pair is marked as false. The cartoon, inset, illustrates the scenario.

the crossing factor:

$$\begin{aligned}
 a = K_S \text{ with accidental crossings} &= \int_{K_S - 50 \text{ MeV}}^{K_S + 50 \text{ MeV}} N[{}^3P_2(3\text{-track DV})](m) dm \\
 b = K_S \text{ without accidental crossings} &= \int_{K_S - 50 \text{ MeV}}^{K_S + 50 \text{ MeV}} N[2\text{-track DV}](m) dm \\
 \therefore \text{Crossing factor} &= \frac{a}{a + b}
 \end{aligned} \tag{4.4}$$

where the m is the invariant mass of the vertex, and 3P_2 is the permutation of two tracks in the 3-track vertex. The first equation corresponds to integrating over the orange histogram in Figure 4.26 and the second equation corresponds to the blue histogram.

This method assumes that the reconstruction efficiency for the 2-track and 3-track displaced vertices is the same, i.e. the accidental crossing should have no bearing on whether the vertex was reconstructed or not. Previously, a Run 1 DV analysis used this exact technique but had limited precision due both to the displaced vertex reconstruction efficiency and because it could not quantify the rate at which a K_S was reconstructed due to the presence of the accidental crossing track. Since that analysis, aside from the displaced vertex reconstruction efficiency improving, part of the new reconstruction procedure allows tracks to be attached to the vertex after the initial seed formation and merging has occurred. Tracks that are attached in this way are labelled as being ‘associated’ to the vertex, i.e. they were not part of any initial 2-track seed. As such, only 3-track vertices in which the two-tracks forming the k-short peak

are *not* associated are used in this measurement. This ensures that the measurement does not include vertices where the presence of the accidental crossing track caused the vertex to be reconstructed. Figure 4.27 shows whether either of the 2-tracks that, combined, have an invariant mass within 50 MeV of the K_S , are or are not associated to the DV for one 2017 data run. If either track was attached after seed vertex formation, the track-pair is added to the ‘true’ bin. The histogram shows that around 72% of the K_S would have been reconstructed, regardless of the presence of the accidental crossing track, and so can be used in the normalization of the accidental crossing background estimate.

Validation of this and the second accidental crossing estimation technique will be discussed at the end of the section.

4.4.4 Accidental crossing, track-template estimate

The second accidental crossing background estimation method uses a slightly different technique to estimate the crossing factor, which comes with additional assumptions. In this method, an artificial mass template is built. The template is then normalized to obtain the crossing factor. The artificial template is built from the invariant mass of 3-track displaced vertices, where one of the three tracks is manually attached at high angle to a pre-existing, 2-track vertex to mimic an accidental crossing. The template is built in stages:

1. First a collection of the four-momentum of high angle tracks is built from data. This is done by sampling displaced vertices with more 3 or more tracks above some mass threshold (previously 3 GeV [139]) in data events. A position vector can be calculated that connects the PV and DV. Tracks at a high angle to that vector, i.e. with $\Delta R > \text{cut}$ where $\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2}$ have their $\Delta\phi$, $\Delta\eta$, and p_T saved to the template. The ΔR upper limit is set such that only tracks at a high angle are saved to the template, in previous analyses the value of the cut has been between 0.7–1.0, and is currently being studied for this analysis as it can sculpt the shape of the high-mass region in the second step.
2. Next, for each 2-track reconstructed displaced vertex passing the baseline fiducial selections, a random track in the template is added to the vertex, and the mass of the vertex is recalculated including the properties of that track. This step is the reason that the relative ϕ and η values are saved to the track template, in the previous step, instead of the absolute values to ensure no unphysical vertices are produced. For example, if the absolute values were used, it is possible that a track could be saved that ends up back-to-back with the seed vertex; a scenario in which the vertex is unlikely to be reconstructed.
3. Next, the crossing factor is extracted. In the previous method this was extracted by using the invariant mass of the K_S to tag vertices that were or were not crossed. In this method, the assumption is made that, above some invariant mass threshold, all measured 3-track displaced vertices come exclusively from the accidental crossing background. Previous studies indicate that this is a reasonable assumption: a 3-track vertex cannot be the product of merging, as seed vertices need at least 2-tracks, and the rate of DV’s produced in hadronic interactions is significantly reduced above a given mass for displaced vertices of this multiplicity. As such, the artificial 3-track mass template is scaled to the measured 3-track mass distribution, above some threshold mass.

This scaling factor can be interpreted as the crossing factor in the following way: if all 2-track vertices always underwent exactly one accidental crossing to be promoted to a higher mass, then both the artificial and measured 3-track mass distributions should contain a close-to-

identical number of vertices above some mass threshold, i.e. a crossing factor of 1. What is actually seen is that far fewer 3-track vertices are measured than estimated in the mass template and, depending on the region of fiducial space being considered, crossing factors are between 0.1–0.3%.

4.4.5 Discussion on background estimation techniques

Unlike the K_S -based technique, the track-template method relies on a series of assumptions that are currently being studied for this version of the analysis. The first assumption is that the angular constraints placed on the tracks added to the template to avoid producing unphysical vertices are not sculpting the kinematics of the vertices in the mass distribution that is scaled in the third step. The next assumption is that adding tracks to low-mass vertices entirely at random will reproduce the mass distribution in real accidental crossing scenarios. One way in which this assumption can be studied is by attaching tracks to vertices using separation significance, e.g. if a vertex has a relatively large position uncertainty, an accidental crossing track may be more likely to be attached than for a vertex with a more precisely measured position. This effect could also be studied as a function of p_T or as a function of the opening angle between the two tracks, as highly-boosted 2-track vertices will have a smaller opening angle and a smaller position uncertainty than the same decay with a smaller boost. The K_S vertices provide an excellent candidate for these studies, since the original 2-track vertex can be tagged by mass. If some dependency on vertex kinematics is found for high-angle track attachment this can either be used to measure the uncertainty on the technique or to motivate a different attachment scheme.

Both the K_S and track-template accidental crossing techniques provide independent measurements of this background and can be used to check the consistency of each method. It is expected, given the improvements to secondary vertexing and Large Radius Tracking between this analysis and the previous displaced vertex analysis, that the K_S method will provide an accurate estimation of the crossing factor.

Each method described in the previous section is validated before using it to estimate background in the signal region. The validation techniques for each of the three backgrounds are still under discussion and depend somewhat on the final choice of selections used for the signal region. For example, the results from previous displaced vertex analyses found no background events when the full event selection was applied, but the displaced vertex track multiplicity was reduced from five tracks to four. Decreasing the number of tracks required by the displaced vertex in the search would improve the analysis efficiency but may come at the cost of a larger background, which needs to be studied before making a decision. The validation techniques for each background estimation also change depending on how the signal region is defined.

4.5 Conclusion and outlook

This chapter describes the foundations of a search for evidence of new physics processes producing long-lived particle that decays to a massive, multi-track displaced vertex and multiple jets in 139 fb^{-1} of proton-proton collisions collected by the ATLAS experiment in 2015-2018 at a centre-of-mass energy of $\sqrt{s} = 13 \text{ TeV}$. The trigger and DRAW filter which form the gateway to the analysis, the event selections including the mass reconstruction of displaced vertices within the ATLAS ID, have been carefully studied and estimation techniques for the DV background have been described.

Whilst the fundamental groundwork for the search has been laid, a few steps remain before the analysis is complete, including validating the background estimation methods in control regions, making a background estimation for the signal region, calculating the associated uncertainties for both the number of expected background and signal events for each signal model, and performing statistical tests on the results. As such, the acceptance of this analysis compared to the previous iteration is not a simple extrapolation from previous work. However, in addition to the larger, more energetic dataset delivered by the LHC, the acceptance should improve at the DV-level by virtue of the changes to the detector and the reconstruction software, whilst the acceptance at the jet-level should improve due to the changes to the DRAW filter. This final section considers some additional methods to improve the signal efficiency of the analysis that could be considered either for this work or in future studies.

One of the selections mentioned in the analysis overview was the track-multiplicity of the displaced vertex. Previous DV analyses have found that no background events were predicted down to four-track vertices, following all other selections, so expanding the signal region from five-track to four-track should increase the signal efficiency with minimal loss. One of the reasons this change was not made before is that, in previous analyses, this region has been to help validate the background estimation method. Using the four-track region is particularly useful for validating the track-attachment method for the accidental-crossing vertex estimation. To be able to include four-track vertices in the signal region requires coming up with a new validation region for this method. This could consist of a subset of four-track vertices that are excluded by other selections in the event, for example, only vertices within the material region could be used in the estimation, although this would rely on the two other sources of background: hadronic interactions and merged vertices, to be successfully accounted for in those regions. Reducing the track-multiplicity requirement will be one of the main focuses of this DV analysis.

Other handles that could help improve signal efficiency are related to the jet-track association. Currently, the analysis makes no requirement that the jets in the event come from hadronised partons from the LLP decay, but if the LLP is reconstructed as a DV then some of the high- p_T jets in the event should do⁶. One way this could be implemented is by creating an additional handle that checks whether one of the jets responsible for the event passing the jet multiplicity and p_T requirements is matched with a significant fraction of the tracks forming a DV in the event. This could be used, for example, to increase the fiducial region currently excluded by the material veto. The material map vetos around 42% of the fiducial volume, which is bounded by the tracking and vertexing efficiency. This is currently necessary as the regions are dominated by vertices formed by hadronic interactions with the material, some of which are already massive enough to pass the signal requirements for DV's and some of which are promoted into the signal region by an accidental crossing track. If these DV's could instead be vetoed by the requirement that they must be matched to high p_T jets, then the need for the material veto could be averted. However, removing the material veto would introduce

⁶In some signal models, such as the gluino cascade decay with an LSP LLP neutralino, some jets in the event will come from the promptly decaying gluino.

background into the signal region and the potential gain in statistics could be hampered by the uncertainty introduced in the DV-jet matching, which itself has been shown to be a non-trivial undertaking.

An additional way jet-track association could be explored is through the Jet Vertex Tagger (JVT). JVT is used to veto PU jets from the reconstructed jets in an event and uses a combination of jet attributes with respect to the primary vertex and pile-up (PU) vertices. Part of the tagger is trained on the $\Sigma^{\text{trk}} p_T$ of jets both with respect to the PV and PU vertices, see section 1.3. Long-lived particles with lifetimes greater than 0.1 ns are likely to have a significant fraction of tracks that are not associated to the PV giving a $\Sigma^{\text{trk}} p_T < 5$ GeV, as was seen in section 4.3.1. As such, using JVT to reject PU jets can cause jets from LLP decays to be removed. One way to reduce this effect could be to create a similar tagger that accepts jets that are *not* associated to a PU vertex, instead of rejecting those not associated to the PV. Another could be to create a tagger used alongside the JVT that checks the jets' association to the DV instead. In that scenario the JVT(PV) and JVT(DV) could be used together to keep jets that are either likely to come from a process at the PV *or* from a decay resulting in a DV. As $\Sigma^{\text{trk}} p_T$ has a complex relationship to the JVT, these proposed modifications may not be simple but could avoid losses in signal efficiency at longer lifetimes due to signal jets being rejected. Further study will show if such studies are necessary to improve signal efficiency for later analyses.

Chapter 5

Conclusion

Detailed measurements of SM particles and their interactions, as well as searches for BSM phenomenology, drive the field of particle physics. Run 3 of the LHC, the HL-LHC, and beyond present opportunities that are intimately tied to instrumentation development. The ATLAS data analysis described in this thesis is entirely dependent on both the performance of the silicon Pixel detector as well as the speed and accuracy with which tracks are reconstructed from it, both of which were explored in the instrumentation chapters of this thesis.

Development for HL-LHC pixel detectors are at the cutting edge of what is physically achievable with the energy budget available and data rates that can be read off the detector. Chapter 1 and **Papers I** and **II** highlighted challenges to both readout chips and track reconstruction that can be mitigated with technology (radiation-tolerant 65 nm front-end design) and algorithms (using cluster size to measure angle). Chapter 2 detailed the mechanisms which cause damage to silicon sensor elements in an environment with a high particle fluence such as the ATLAS Pixel detector and described how these effects were implemented in the experiment simulation framework, Athena. **Paper III** described how attributes of the simulation were validated in data collected by the ATLAS experiment in Run 1 and 2 and discussed how the simulation can be used to predict the conditions for detector operation in Run 3 and beyond. Whilst aspects of the simulation, such as its modeling of annealing are being improved, the implementation described in this thesis will serve as a baseline for running radiation damage simulation in Athena going forward. This simulation will be of particular use in preparing for HL-LHC, as the ITk Pixel readout chips will have 4-bit ToT as opposed to 8-bit in the current Pixel detector. The quantisation error, from having a smaller ToT register, will have to be decoupled from the effects on cluster size and shape from radiation damage, which will occur at a higher rate in ITk. As such, carefully modeling microscopic defects in silicon ensure that the behaviour of the detector is well understood and allows us to decouple simulation error from real anomalies in the signal that might be evidence for new physics. For example, analyses searching for highly-ionizing BSM particles must be able to distinguish the cluster shape and charge collected from radiation damage effects.

Chapter 3 and **Paper IV** described the implementation of a Kalman filter in the inherently parallel neuromorphic architecture, TrueNorth. Whilst this study concluded that the chip in its current form is not suitable for use as part of the ATLAS computing infrastructure, aspects of its programming could inspire data compression techniques. For example one of the tests performed in TrueNorth was the resilience of the neuromorphic Kalman filter implementation to truncated data, for speedup. It was found that the Kalman filter did not diverge from the mean value of the state being tracked, but produced a state estimate with a higher noise. Perhaps truncating data inside the chip, before performing computations would make online processing

more feasible to complete within the trigger latency. Part of the difficulty in processing data in a high-radiation environment, such as the ATLAS Pixel detector, is the rate of bit-flips that can occur. Although spike encoding is highly inefficient, it is very fault tolerant: a bit flip in a spike train will only change the encoded value by ± 1 whereas in binary a single bit flip can change the encoded value by $\pm$ the flipped bit. Although the low power offered by processing sparse spike trains might not be useful off-detector, on detector its fault tolerance or encoding inspired by sparse trains, could reduce the need to continually refresh the readout configuration, as is planned for ITk Pixel chips. However, if spikes were used they would have to be transduced or a native sensor used, both of which could offset any benefits and introduce more latency into the system. The question of how exactly these features could be added to the computing infrastructure are not within the scope of detectors at HL-LHC but could be of interest to future collider experiments.

Understanding detector performance at a fundamental level as well as maximizing computing resources to collect sufficient data were both found to be essential to the search for rare physics processes. Chapter 4 described a search for massive, BSM long-lived particles decaying to SM quarks and reconstructed as a displaced vertex produced in association with high p_T jets that necessitates these considerations. It was shown that reconstructing the long-lived particle decay vertex increases the time taken to reconstruct an event with Run 2 pile-up conditions by a factor of 2.5, which is not feasible to apply to all physics data triggered by the detector. Instead a subset of events of interest are filtered from the physics stream, before undergoing special reconstruction. Reconstructing these decays would also not be possible without the high-granularity Pixel detector that is able to reconstruct displaced decays out to 300 mm from the interaction point.

As this analysis shows, instrumentation and computing serve an essential role in measuring rare processes and searching for evidence of BSM physics in collider experiments, something that will become even more paramount in data-taking during Run 3 of the LHC, HL-LHC, and beyond.

Appendices

Appendix A

Radiation damage modeling

Additional material for Chapter 2. Section A.1 contains code snippets to supplement the descriptions of signal formation digitization, Bichsel energy deposition, and radiation damage. Section A.2 contains a concise summary of properties of semiconductors, doping, and the formation of a pn-junction. Section A.3 provides the difference in naming schemes between the digitization package in Athena releases 20.7 and 21 compared to the upgrade branch (22) where radiation damage is implemented.

A.1 Pseudocode snippets

A.1.1 Original Athena Pixel digitization code

The signal formation code logic in the original Athena implementation (see section 2.3.3), where x is a 3D position vector, `distElectrode` is the distance from the charge carrier to its collecting electrode, `rDif` is the distance travelled due to diffraction, θ_L is the Lorentz angle.

Algorithm 1 Original Athena signal formation in `SubChargesTool`.

```
1: diffConst := 0.007 [mm]
2: for  $i < \text{scaled nSteps}$  do
3:    $x_i = x_0 + \Delta x \left(i + \frac{1}{2}\right)$ 
4:    $\text{distElectrode} = \frac{1}{2} \cdot \text{sensorThickness} - \left(\text{module.readoutSide}() \cdot x_i[\text{depth}]\right)$ 
5:   for  $j < \text{scaled nCharges}$  do
6:      $\text{rdif} = \text{diffConst} \cdot \sqrt{\text{pathLength}/300 \mu\text{m}}$ 
7:      $x_f = x_i + \text{distElectrode} \cdot \tan(\theta_L) + \text{rdif} \cdot \text{rand}(\text{gaussian})$ 
8:   end for
9: end for
```

A.1.2 Bichsel simulation implementation in Athena

The Bichsel energy deposition simulation, the behaviour described here covers functions in both the Charge Tools and the BichselSimTool and described in section 2.3.4. In the pseudocode the particle species is denoted by pdgID , β and γ are the Lorentz variables, Σ_t is the integral of the differential cross-section multiplied by the density of the material, λ is the mean free path between collisions, x is the position of the collision and e is the energy imparted to the bulk.

Algorithm 2

```

1:  $\text{pdgID} := \text{genPart} \rightarrow \text{pdg\_id}()$ 
2: if  $\text{deltaRay}$  then
3:   break
4: else
5:    $\beta\gamma := \text{getBetaGamma}(\text{genPart})$ 
6: end if
7:  $\Sigma_t := \text{lookupMaxIntXsection}(\text{pdgID}, \beta\gamma)$ 
8:  $\lambda := \frac{1}{\Sigma_t}$ 
9: while 1 do
10:   $x_i = -\lambda \cdot \ln(\text{rand}[0, 1])$ 
11:  if  $(x_{total} + x_i) > \text{pathLength}$  then
12:    break
13:  end if
14:   $\text{index} := \text{rand}(0, \Sigma_t)$ 
15:   $e_i := \text{lookupEnergy}(\beta\gamma, \text{index})$ 
16:  if  $(e_{total} + e_i) > \text{deltaRayCut}$  then
17:    break
18:  else if  $(e_{total} + e_i) > \text{genPart} \rightarrow \text{getEnergy}()$  then
19:     $e_{total} += (\text{genPart} \rightarrow \text{getEnergy}()) - e_{total}$ 
20:    break
21:  else
22:     $x_{total} += x_i$ 
23:     $e_{total} += e_i$ 
24:  end if
25: end while

```

A.1.3 Radiation damage implementation in Athena

This code summarises the Athena implementation of radiation damage, described in section 2.4.3.

Algorithm 3 The signal formation simulation with radiation damage

```

1: [fluence, sensorBias, temperature,  $\langle Q \rangle$ ]  $\leftarrow$  getConditions( runNumber)
2: carriersPerChunk := totalEnergyDeposited  $\div$  (3.6  $\cdot$  nChunks  $\cdot$  nSteps)
3: for ( $i < \text{nSteps}$ ) do
4:   energyDeposited in this step :=  $0.5 \times \text{hitRecord}[i].\text{energy}()$ 
5:   initialPosition := hitRecord[i].position()
6:    $\phi_i \leftarrow \text{lookupWeightingPotential}(\text{initialPosition})$ 
7:   for ( $j < \text{nChunks}$ ) do
8:     if doRadDamage then
9:       for ( $k < 2$ ) do {Holes and electrons loop}
10:        timeToElectrode  $\leftarrow \text{lookupTimeToElectrode}(\text{isHole}, \text{sensorThickness}, \text{depth}_i)$ 
11:        trappingTime  $\leftarrow -(\beta_d \Phi)^{-1} \ln(\text{rand})$ 
12:        if (trappingTime < timeToElectrode) then {charge is trapped}
13:          trapLocation[ z ]  $\leftarrow \text{lookupTrapLocation}(\text{trappingTime}, \text{isHole}, \text{depth}_i)$ 
14:          trapLocation[ x ]  $\mathrel{+}= \text{lorentzDrift}(\text{depth}_i, \text{trapLocation}[z], \text{isHole})$ 
15:          trapLocation  $\mathrel{+}= \text{diffusion}(\text{rand})$ 
16:          for ( $x\_index = -1$  to  $1$ ) do {nearest neighbour loop}
17:            xPixel := trapLocation[x] + ( $x\_index \cdot x\_pixelPitch$ )
18:            for ( $y\_index = -1$  to  $1$ ) do {nearest neighbour loop}
19:              yPixel := trapLocation[y] + ( $y\_index \cdot y\_pixelPitch$ )
20:               $\phi_f \leftarrow \text{lookupWeightingPotential}(\text{xPixel}, \text{yPixel}, \text{trapLocation}[z])$ 
21:              inducedCharge := energyDeposited  $\cdot (3.6 \text{ eV})^{-1} \cdot (\phi_f - \phi_i)$ 
22:            end for
23:          end for
24:        end if
25:      end for
26:    else { ! doRadDamage }
27:      do original signal formation code in Algorithm 1
28:    end if
29:  end for
30: end for

```

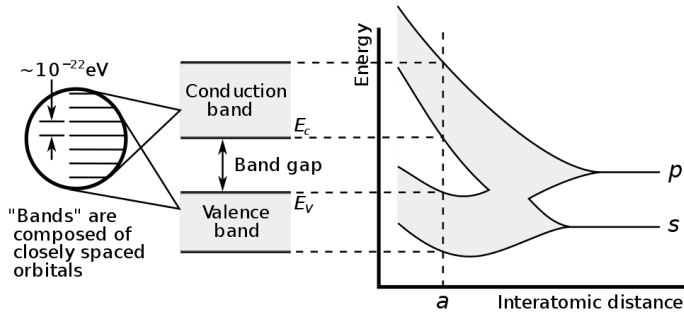


Figure A.1: As interatomic distances are reduced, degenerate energy levels in the lattice split into closely spaced bands [155].

A.2 A brief overview of semiconductor physics

This part of the appendix is intended as a concise summary of the relevant semiconductor and pn-junction properties necessary to understand Chapter 2, which may already be familiar to readers and so is not included in the main text. The summary starts with a description of semiconductor properties in section A.2.1, followed by a description of the formation of a pn-junction in section A.2.3, concluding with an overview of the relevant properties of a diode in section A.2.4.

A.2.1 Semiconductor properties

Silicon is a semiconductor, a material that is neither as conductive as a metal nor as resistive as an insulator but that can be manipulated to behave more like either. Electrical conductivity arises from electrons being able to jump easily from atom to atom in a lattice. When atoms form a lattice, the degenerate energy levels in each atom split due to the requirement of non-overlapping wavefunctions for fermions. These closely spaced split levels form a band of near-continuous energy states, see Figure A.1. The valence band is occupied by the highest energy electrons in the atom, which participate in chemical bonds. The conduction band is the energy at which charge carriers are dissociated from atoms and can move freely within the lattice. In conductors, the valence band contains only weakly bound electrons and so is also the conduction band. In non-conductors the valence and conduction bands have a separation energy that corresponds to the energy needed to break a lattice bond, this is known as the band gap. The band gap is essentially a range of forbidden energy states that electrons cannot occupy.

Insulators have a large band gap separating their valence and conduction bands, which means only large amounts of energy will liberate valence electrons from their stable bonds. Semiconductors have a smaller band gap than an insulator, on the order of eV, so if energy is introduced to the lattice, electrons in the valence band will be able to occupy the conducting band and move freely. Likewise, the space left behind when an electron leaves the valence band, a hole, can move between atoms in the lattice. The hole is a lack of electron and so can effectively be considered a positive-charge carrier.

The mean number of electrons occupying the conduction band in a semiconductor, for a given temperature, can be calculated using Fermi-Dirac statistics. This is also known as the density of free charge carriers or the charge carrier concentration and is proportional to the conductivity

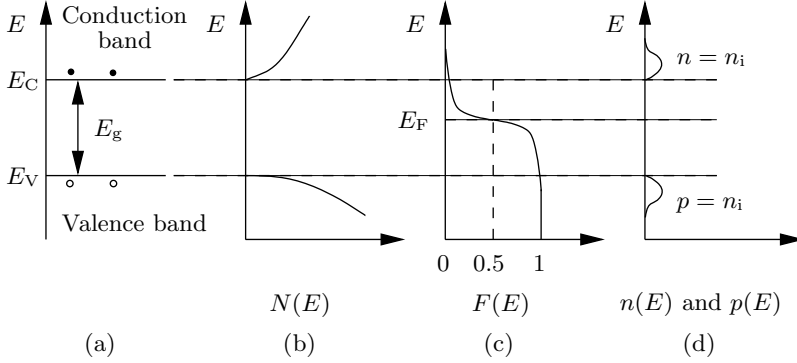


Figure A.2: Properties of an intrinsic semiconductor above 0 K, (a) band structure showing thermally liberated charge carriers, (b) density of states, (c) Fermi function, (d) the electron and hole concentrations [72].

of the semiconductor. The density of free charge carriers in the lattice is given by:

$$n_i = \int_{E_C}^{\infty} N(E) F(E) dE \quad (\text{A.1})$$

where E_C is the lowest energy, or bottom, of the conduction band, $N(E)$ is the density of states in the conduction band, and $F(E)$ is the Fermi-Dirac function. The Fermi-Dirac function describes the probability of the state at energy E being occupied for a given temperature, T :

$$F(E) = \frac{1}{1 + e^{(E-E_F)/kT}} \approx e^{-(E-E_F)/kT} \quad (\text{A.2})$$

where k is Boltzmann's constant, T is the temperature, and E_F is the Fermi level's energy or chemical potential. Another way of viewing the formula is to say that E_F is the energy, for a given temperature, at which there is a 50% chance that an electron is occupying it. In reality, E_F usually lies in the band gap and the Fermi-Dirac distribution is used to calculate the occupancy of the conduction band.

In silicon it can be shown that $(E_C - E_F) > 10 kT$ holds till ~ 650 K, so the approximation on the right hand side of equation A.2 holds. For an equal carrier concentration of electrons and holes in a semiconductor, an expression can be found for the intrinsic Fermi level. It can be shown that the intrinsic Fermi level lies halfway between the conduction and valence bands at room temperature. From the density of states:

$$N(E)dE = 4\pi \left(\frac{2m_q^*}{h^2} \right)^{2/3} E^{1/2} dE \quad (\text{A.3})$$

where m_q^* is the effective mass of the charge carrier and h is Planck's constant. Substituting and evaluating equations A.3 and A.2 in equation A.1 gives the concentration, n_{q_i} of free charge carriers:

$$n_{q_i} = 2 \left(\frac{2\pi m_q^* kT}{h^2} \right)^{3/2} e^{-\delta E/kT} \quad (\text{A.4})$$

where $\delta E = \begin{cases} E_C - E_F, & \text{for electrons} \\ E_F - E_V, & \text{for holes} \end{cases}$

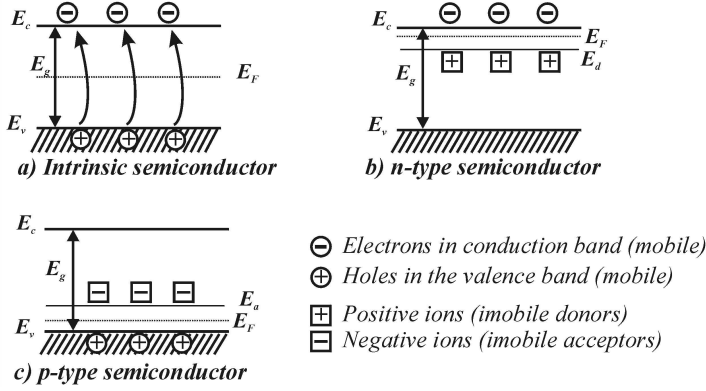


Figure A.3: The effect of dopants on the Fermi level in semiconductors [156].

where E_v is the energy at the top of the valence band. If the lattice is electrically neutral then the number of free electrons will equal the number of free holes, so from equation A.4, the Fermi level is:

$$E_{Fi} = \frac{E_c - E_v}{2} + \frac{3kT}{4} \ln \left(\frac{m_h^*}{m_e^*} \right) \quad (\text{A.5})$$

At 300 K, the second term in equation A.5 is three orders of magnitude smaller than the first term and thus the Fermi level is essentially halfway between the valence and conduction bands, see Figure A.2. Equation A.5 shows the Fermi level getting closer to the conducting band with increasing temperature. Another way of interpreting this is that the number of free charge carriers increases with temperature. The intrinsic charge carrier concentration scales as:

$$\begin{aligned} n_i^2 &= n_e \cdot n_h \\ &= 4 \left(\frac{2\pi kT}{h^2} \sqrt{m_e^* m_h^*} \right)^3 e^{-E_g/kT} \end{aligned} \quad (\text{A.6})$$

where n_i is the intrinsic carrier concentration and E_g is the band-gap. In silicon the band-gap is 1.12 eV and at 300 K the effective mass of the electron and hole are 1.09 and 1.15 times the electron rest mass, respectively. Thus at 300 K the intrinsic carrier concentration is $\sim 1 \times 10^{10} \text{ cm}^{-3}$.

A.2.2 Doping

Intrinsic semiconductors have an equal number of holes and electrons and are made of one element. However, it is possible to change the relative carrier concentration such that either holes or electrons are the dominant charge carriers. This process is called doping.

Doping replaces a number of atoms in the lattice with elements that have either one less or one more electron in their outer shell. In the case of silicon this means that atoms in the lattice will be switched with either group 3 or group 5 atoms, usually phosphorus or boron. If a group 3 atom is used there will be one less valence electron available to participate in bonding, effectively creating an additional hole. The dopant atom is thus known as an electron acceptor. If a group 5 atom is used, there will be an extra electron present in the valence band not used for bonding and the dopant atom is known as an electron donor. Thus adding either a donor or acceptor dopant to the lattice will change the carrier concentration such that $n_e \neq n_h$. A typical dopant concentration, for example in doped silicon for use in high energy physics, is between $10^{12} - 10^{19}$ [71]. From equation A.6, this means that there are 10^{2-9} more of

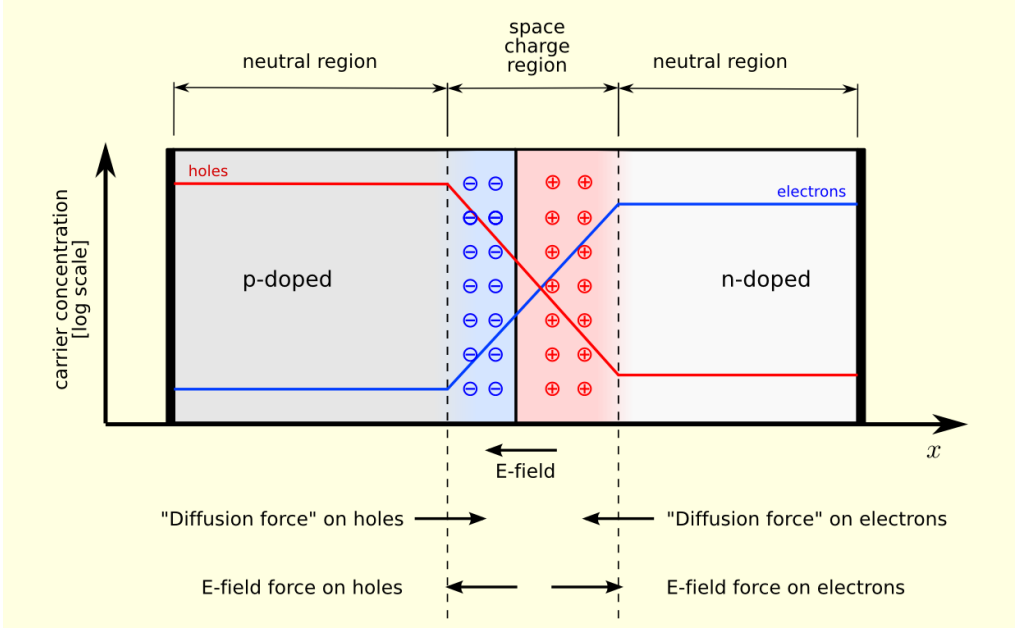


Figure A.4: Cartoon of a pn-junction's band diagram in thermal equilibrium. The blue and red lines represent the free electron and hole concentrations, respectively [157].

the dominant charge carriers and 10^{2-9} less of the minority charge carriers. Donors effectively introduce an energy level close to the conduction band which, from equation A.4, increases the Fermi level, reducing the number of holes acting as charge carriers. A material doped with a donor is known as n-type. This can also be thought of as there being more free electrons available to recombine with holes. Conversely, acceptors introduce an energy level close to the valence band, decreasing the Fermi level and reducing the number of free electrons, as can be seen in Fig. A.3. A material doped with acceptors is known as p-type.

A.2.3 The pn-junction

A pn-junction is the metallurgical transition in a material between a p-type semiconductor and an n-type semiconductor, see Fig. A.4, and is the basis of a diode. In thermal equilibrium a pn-junction has a net electrical current of zero but charge continuously flows across the junction boundary. The net current can be broken down into two equal and opposite components: drift and diffusion.

The n-type material has an excess of free electrons compared to an intrinsic semiconductor and the p-type material has an excess of holes. During the junction formation, free charge carriers start off in a region of higher concentration and, through random thermal motion, will diffuse across the junction. This diffusion process is characterised by the Einstein relation:

$$\vec{J}_{\text{diff}} = D_q \vec{\nabla} n_q \quad (\text{A.7a})$$

$$D_q = \frac{kT}{e_q} \mu_q \quad (\text{A.7b})$$

where J_{diff} is the diffusion current, D is the diffusion constant defined in equation A.7b, ∇n_q is the gradient of the concentration of the charge carrier, q , e_q is the electric charge, and μ_q is

the mobility of the charge carrier.

As diffusion proceeds across the junction, a space-charge is built up around it. This occurs because the free charge carriers diffuse away from their parent atom and leave behind ions in the lattice that are fixed in space. As the majority charge carrier in an n-type material is an electron, the space-charge region in the n-type material will be positively charged, and in the same way the p-type space-charge region will be negatively charged.

As many free charge carriers diffuse across the material boundary, a space-charge region will build up around the pn-junction. This results in an intrinsic potential difference across the junction, see Fig. A.4. The potential difference caused by thermal diffusion is known as the ‘built-in’ potential, V_{bi} , and is characterised by:

$$V_{bi} = \frac{kT}{e_q} \ln \left(\frac{N_D N_A}{n_i^2} \right) \quad (\text{A.8})$$

where e is the fundamental electric charge, N_D is the n-type (donor) dopant concentration, N_A is the p-type (acceptor) concentration, and n_i is the intrinsic carrier concentration. This formula assumes that all dopant atoms have been ionised by diffusion. If this is not the case N_D and N_A are replaced with their respective free charge carrier concentrations. The potential difference across the junction will cause minority free charge-carriers in the material to traverse the pn-junction. For example, the built-in potential gradient goes from high to low from the n-type material towards the p-type. This means that electrons placed in that field will travel from the p-type material towards the n-type material, in the opposite direction to the diffusion current. This drift current is given by:

$$\vec{J}_{\text{drift}} = e_q n_q \vec{v}_q = e_q n_q \mu_q \vec{\nabla} V_{bi} \quad (\text{A.9})$$

where e is signed charge, n is the free charge carrier concentration, v is velocity, μ is the mobility, and q denotes that these values are different for holes and electrons.

With no external fields applied, the junction is in a thermal equilibrium, where the diffusion current is equal to the drift current. The space-charge region has less free charge carriers than the surrounding lattice, thus this region is said to be ‘depleted’ of free charge carriers and is often called the depletion region.

Applying an external bias

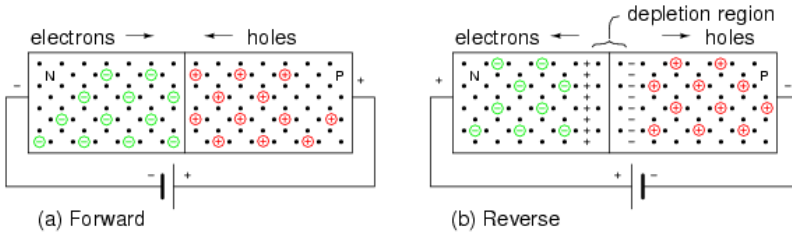


Figure A.5: Applying a bias voltage across the pn-junction either causes it to behave like a conductor, forward bias, or increases the depletion region, reverse bias [158].

The diffusion current in thermal equilibrium is limited by the built-in potential which is caused by the build-up of space-charge around the pn-junction. This diffusion current can be increased or suppressed by applying an external bias voltage to the junction. If a bias voltage is applied such that the p-type material is at a higher potential than the n-type material, an

electric field will be created which points in the opposite direction to the built-in electric field. Increasing the value of the bias voltage will cause the built-in voltage to be cancelled out and allow free charge carriers to flow freely across the junction under diffusion and in the drift of the applied field. This setup is known as forward biasing, see Fig. A.5a. Forward biasing effectively gives electrons and holes enough energy that they can traverse the pn-junction and recombine with holes, creating a continuous current. Applying a forward bias changes the electrical properties of a diode, causing it to behave like a conductor and removing the depleted space-charge region.

If instead, the bias voltage is applied such that the n-type material is at a higher potential than the p-type material, the pn-junction is said to be reverse biased. This causes an electric field which adds to the electric field from the built-in potential, further suppressing diffusion across the boundary. Under reverse bias, the excess free charge carriers in each material move away from the pn-junction, increasing size of the space-charge region and further depleting it of charge carriers, see Fig. A.5b.

When the bias voltage reaches a critical value called the depletion voltage, V_{dep} , the depletion region extends across the entire depth of the material. At full depletion the number of free charge carriers in the bulk is minimised. This is equivalent to suppressing the diffusion current, or thermal noise, of the intrinsic junction. When the junction is initially reverse biased, any minority carriers on either side will flow across the junction causing a current spike. However, after this initial flow of current, the junction effectively acts like an insulator. Current only flows across the diode if an external source of charge carriers is injected into the lattice under reverse bias.

The width, w , of the depleted space-charge region is related to the applied bias voltage by equation A.10, which derives from solving Poisson's equation for the junction:

$$w = \sqrt{\frac{2\epsilon}{e_q}(V_{\text{bi}} - V_{\text{bias}}) \left(\frac{N_A + N_D}{N_A N_D} \right)} \quad (\text{A.10})$$

where ϵ is the material permittivity, V_{bi} is the built-in potential at thermal equilibrium, and V_{bias} is the applied bias voltage. This relationship simplifies further for the case when one carrier concentration exceeds the other by orders of magnitude. For example, in the case of $N_A \gg N_D$, the depletion region extends further into the n-type material than the p-type material for a given bias voltage. The width of the depletion region is then given by:

$$\begin{aligned} w &= \sqrt{\frac{2\epsilon}{e_q \cdot N_D}(V_{\text{bi}} - V_{\text{bias}})} \\ &= \sqrt{2\epsilon \cdot \mu_e \cdot \rho_d(V_{\text{bi}} - V_{\text{bias}})} \end{aligned} \quad (\text{A.11})$$

where μ_e is the electron mobility, and ρ_d is the resistivity of the n-type material.

A.2.4 Diode properties

Finding the depletion voltage *a priori*, in devices constructed from pn-junctions, such as the diode, is not always practical as it relies on knowing the junction properties to a good accuracy. In real devices the full depletion voltage can be found by measuring the capacitance, C , of the junction:

$$C = \frac{\epsilon}{d} = \sqrt{\frac{\epsilon \cdot e_q \cdot N_D}{2}} \cdot \frac{1}{(V_{\text{bi}} - V_{\text{bias}})} \quad (\text{A.12})$$

As the space-charge region grows, more non-conducting material is effectively placed between conducting (undepleted) regions, and the capacitance decreases. When full depletion is reached

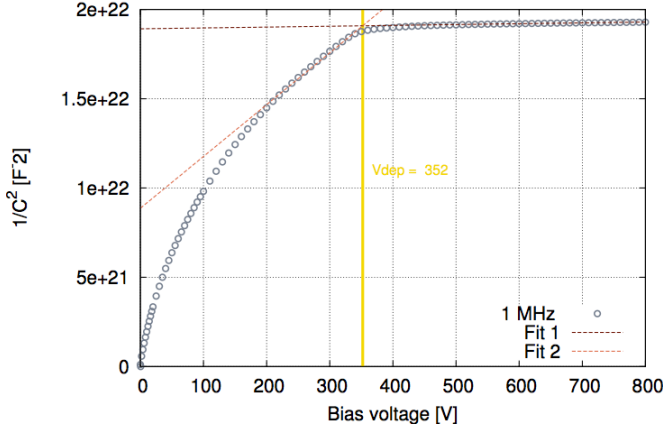


Figure A.6: Example CV characteristic of a silicon diode, measured using a high-impedance LCR meter with a frequency probe of 1 MHz. V_{dep} of the diode is found by fitting the knee in the curve, sensor is reverse biased.

the capacitance of the diode saturates. As such, a CV characteristic of a diode reveals the full depletion voltage, an example of this is shown in Fig. A.6.

Over-depleting, or reverse-biasing the diode at a voltage greater than V_{dep} , initially increases the charge carrier velocity but eventually that too reaches a saturation value, v_s . The mobility thus decreases with increasing electric field strength:

$$\mu = \frac{v_s}{E_c (1 + (E/E_c)^\beta)^{1/\beta}} \quad (\text{A.13})$$

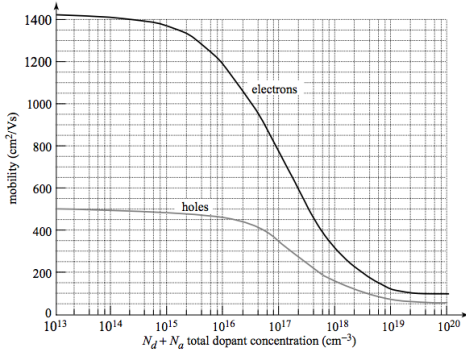
where v_s is the saturation velocity, or maximum velocity with which a charge carrier can move in the bulk, E is the electric field magnitude at a specific point in the bulk, and β and E_c are fit parameters of the model [102]. As the velocity saturates and the electric field increases, the mobility drops off, which is shown in Fig. A.7b. This behaviour is intrinsically linked to the mean free path the charge carrier takes before it collides with a lattice atom. As the electric field strength across the pn-junction increases, so too does the velocity of the charge carrier, but the mean free path stays the same. Thus, eventually lattice collisions become the limiting factor on charge carrier velocity and the mobility decreases. This phenomenon is temperature dependent, decreasing the temperature increases the mean free path, and with it the saturation velocity.

Similarly, increasing the effective carrier concentration also decreases the mobility of the charge carrier, as collisions with the ionised bulk following depletion disrupt the passage of the charge carriers. This is characterised by the phenomenological relation:

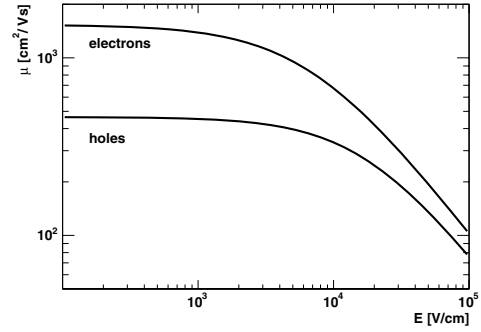
$$\mu = \frac{\mu_{\text{max}} - \mu_{\text{min}}}{1 + (N/N_{\text{ref}})^\alpha} + \mu_{\text{min}} \quad (\text{A.14})$$

where N is the dominant carrier concentration and μ_{max} , μ_{min} , N_{ref} , and α are parameters found by fitting to experimental data [160]. Thus keeping the carrier concentration low by using a high-purity material with a low dopant concentration for the bulk of the diode will keep the mobility of charge carriers high.

The bias voltage cannot be increased indefinitely. After a certain value of V_{bias} , the junction potential will break down due to electrons being accelerated to such high energies they end up liberating other electrons from the lattice. These liberated charge carriers also have



(a) The change in mobility of charge carriers as a function of the dopant concentration at 300 K with no external bias applied [159].



(b) The mobility of charge carriers at 300 K as a function of Electric field strength, which is calculated with equation A.13[72].

Figure A.7: The effect of doping concentration and electric field on charge carrier mobility.

significant energy and liberate others creating an avalanche of charge carriers flowing across the junction: a runaway, or ‘breakdown’, current. Breakdown itself is not destructive but if significant current flows then the heat dissipated by the charge carriers can deform the lattice permanently and destroy the junction [69].

Chapter 2 continues from here by explaining how the unique properties of a silicon diode can be used as a sensing medium in pixel detectors.

A.3 List of variable name changes

For clarity, the variable names in the new digitizer code discussed in this thesis have been changed from the original source code. The table below provides a look-up to convert between the variable names of the legacy code and the variable names used in more recent additions to the PixelDigitization package, for example the radiation damage tools.

Old names	New names	Description
pos	startPosition	Particle entry point into detector volume, supplied from Geant4 (vector)
cs	endPosition	Particle exit point from detector volume, supplied from Geant4 (vector)
xEta / xPhi / xDep	eta_0 / phi_0 / depth_0	Individual components of startPosition
xEtaf / xPhif / xDepf	eta_f / phi_f / depth_f	Individual components of endPosition
xEta1 / xPhi1 / xDep1	eta_i / phi_i / depth_i	Components of position at step i in simulation
xEtaD / xPhiD	eta_drifted / phi_drifted	Components of position after drift has been applied
cEta / cPhi / cDep	totalChangeEta / totalChangePhi / totalChangeDepth	Components of the total path length
length	pathLength	
nsteps	scaledNumberOfSteps	Number of steps used in simulation loop, scaled by the path length.
ncharges	scaledNumberOfCharges	Number of charge chunks used in simulation loop, scaled by the path length.
e1	stepSizeEnergy	Amount of energy deposited by the particle in each step. This effectively determines how many charge carriers each chunk of charge represents.
spess	dist_electrode	Distance between charge carrier and electrode.
istep	i	Outer loop iterator.
i	j	Inner loop iterator.

Table A.1: Summary of variable name changes in PixelDigitization subChargesTool classes.

Appendix B

TrueNorth

The appendices here supplement Chapter 3, starting with the complete TrueNorth neuron description in section B.1, only a subset of which was described in the main text. Then, section B.2 describes an algorithm to decide the neuron weights in a parallel cross-bar.

B.1 Complete neuron description

The full neuron equation in TrueNorth contains many parameters that were not used in the final implementation of the Kalman filter. They are included here for completeness. Equation B.1 describes the complete neuron update equation, including the concepts of both leak and stochasticity that were not described in the main text. The symbols in equation B.1 are described in table B.1.

$$\begin{aligned}
 V_j(t) &= V_j(t-1) + \sum_{i=1}^{256} A_i(t) \times w_{i,j} \times \text{sgn} \left(s_j^{G_i} \right) \times \text{weight} \\
 \text{weight} &= \begin{cases} |s_j^{G_i}|, & \text{if } b_j^{G_i} = 0 \\ 1, & \text{if } \left(b_j^{G_i} = 1 \right) \wedge \left(s_j^{G_i} \geq \rho_j^{G_i} \right) \\ 0, & \text{otherwise} \end{cases} \tag{B.1a}
 \end{aligned}$$

Leak update:

$$\begin{aligned}
 V_j(t) &+= \Omega_j \times \text{leak} \\
 \Omega_j &= \begin{cases} \text{sgn} \left(\lambda_j \right), & \text{if } \varepsilon_j = 0 \\ \text{sgn} \left(\lambda_j \times V_j(t) \right), & \text{if } \varepsilon_j = 1 \end{cases} \\
 \text{leak} &= \begin{cases} \lambda_j, & \text{if } c_j = 0 \\ 1, & \text{if } \left(c_j = 1 \right) \wedge \left(\lambda_j > \rho_j^\lambda \right) \\ 0, & \text{otherwise} \end{cases} \tag{B.1b}
 \end{aligned}$$

The potential, V_j is capped at a ceiling and floor defined as:

$$\begin{aligned} ceiling &= \begin{cases} \alpha_j + M_j, & \text{if } \gamma_j = 2 \\ 393216, & \text{otherwise} \end{cases} \\ floor &= \begin{cases} -\beta_j - M_j, & \text{if } \gamma_j = 2 \\ -393216, & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.1c})$$

The neuron will cross the positive threshold and fire if:

$$V_j > (\alpha_j + \eta_j) \quad (\text{B.1d})$$

The neuron will cross the negative threshold if:

$$V_j(t) < \begin{cases} -\beta_j, & \text{if } \kappa_j = 1 \\ -\beta_j - \eta_j, & \text{if } \kappa_j = 0 \end{cases} \quad (\text{B.1e})$$

If V_j crosses the positive threshold it will be reset according to:

$$V_j(t) = \begin{cases} Vrst_j, & \text{if } \gamma_j = 0 \\ V_j(t) - \alpha_j - \eta_j, & \text{if } \gamma_j = 1 \\ V_j(t), & \text{if } \gamma_j = 2 \end{cases} \quad (\text{B.1f})$$

If V_j crosses the negative threshold it will be reset according to:

$$V_j(t) = \begin{cases} -Vrst_j, & \text{if } \kappa_j = 0, \gamma_j = 0 \\ V_j(t) + \beta_j + \eta_j, & \text{if } \kappa_j = 0, \gamma_j = 1 \\ V_j(t), & \text{if } \kappa_j = 0, \gamma_j = 2 \\ -\beta_j, & \text{if } \kappa_j = 1 \end{cases} \quad (\text{B.1g})$$

where sgn denotes the signum function.

scope	symbol	range	description
Basic neuron update	i, j	$[1, 256]$	Axon and neuron indices, respectively
	$V_j(t)$	$[-393216, 393216]$	Neuron potential at tick t
	$A_i(t)$	$[0, 1]$	Presence of spike in axon i at tick t
	$w_{i,j}$	$[0, 1]$	Presence of synapse connecting axon i and neuron j
	G_i	$[1, 4]$	Axon label
	$s_j^{G_i}$	$[0, 2^8 - 1]$	The weight assigned spikes incident on neuron j from axons with label G_i
Stochastic weights	$b_j^{G_i}$	$[0, 1]$	Selects whether to use a static neuron weight or a stochastic weight
	$\rho_j^{G_i}$	$[0, 1]$	Output of a pseudorandom number generator to decide whether neuron will spike in stochastic mode
Basic leak update	Ω_j	$[-1, 1]$	Controls the behaviour and direction of the leak
	λ_j	$[0, 2^8 - 1]$	Leak weight
	ε_j	$[0, 1]$	Changes the sign of the leak depending on the sign of the neuron potential
Stochastic leak	c_j	$[0, 1]$	Selects whether to use a static leak or a stochastic leak
	ρ_j^λ	$[0, 255]$	Output of a pseudorandom number generator to decide whether neuron will leak in stochastic mode
Threshold and reset	α_j	$[0, 2^{18} - 1]$	Positive threshold. If V_j exceeds this the neuron will fire
	β_j	$[0, 2^{18} - 1]$	Negative threshold. If V_j drops below this it will not fire
	M_j	$[0, 2^{18} - 1]$	Acts as a mask for the reset stochasticity and also extends the ceiling for the special reset case where $\gamma_j = 2$
	γ_j	$[0, 2]$	Membrane potential reset mode
	κ_j	$[0, 1]$	Negative V_j threshold saturation or normal reset
	η_j	$[0, 2^{18} - 1]$	Masked stochastic threshold. This is a bitwise AND of $\rho_j^{TM} \& M_j$
	$Vrst_j$	$[0, 2^{15} - 1]$	Reset voltage in special reset mode

Table B.1: Complete neuron parameter list.

B.2 Neuron parameter search

The TrueNorth implementation of a Kalman filter involves finding an accurate representation of weights, for example the elements in the Kalman gain matrix, in the form of the neuron weight and threshold. The general procedure for doing this is finding a rational representation of the matrix element of the form:

$$W = \frac{s_j^{G_i}}{\alpha_j} \quad (\text{B.2})$$

Whilst $s_j^{G_i}$ is only an 8 bit signed register, its range can be extending by duplicating the associated axon input q times. For example, to extend the range of weights from $[-255, 255]$ to $[-1275, 1275]$, $q = 5$. So if it was required that $s_j^{G_i} = 1,200$, four axons could be assigned a weight affiliated with their label of 255 and one axon would be assigned the weight 180 as $4(255) + 180 = 1200$. In the case of the serial Kalman filter implementation, an entire core can be dedicated to optimising only the weight representation, so there are 256 total axons that can be used to increase the range of the neuron weight register. In this case the rational representation search consists of using the Matlab inbuilt continued fraction esitimator *rat*¹. This is an iterative process the precision of the number that should be represented as an argument. For example, consider a weight, $W = 0.9182991$, if the precision is set to $1e - 3$:

$$\text{rat}(W, 1 \times 10^{-3}) = \frac{45}{49} = 0.918(37)$$

Since $45 < 255$, $q = 1$ so no additional axons are needed. If, instead, we want W to 4.s.f.:

$$\text{rat}(W, 1 \times 10^{-4}) = \frac{281}{306} = 0.9183(00)$$

Since $281 < (255 \times 2)$, $q = 2$ so one additional axon is needed. If, 7.s.f. are needed:

$$\text{rat}(W, 1 \times 10^{-4}) = \frac{1,922}{2,093} = 0.9182990(92)$$

As $1922 < (255 \times 8)$, $q = 8$ so seven additional axons are needed.

However, in the parallel implementation there are not 256 axons available to increase the range of the neuron weight and using the full range of the denominator in the ration representation of W , the threshold α_j , is not possible (see section 3.4.4) because of the reset scheme used and the number of axon-neuron pairs dedicated to parallelism. There is also the additional constraint that because each neuron can only store up to four weights, the threshold, or denominator, must be a multiple of 255. In this case an adapted rational representation search is used.

The aim is to maximise the precision to which W is represented for a given blocksize, b . Although it is possible to use the desired precision to set the maximum speedup through parallelism attainable, in practice all corelets in the parallel implementation must have the same blocksize. So it is better to pick b based on either latency constraints or the minimum acceptable precision for weight representation and then choose $p, q, s_j^{G_i}, \alpha_j$.

In the parallel implementation, the minimum stepsize (and hence resolution) attainable is based on the value of the denominator. The larger the denominator, the smaller the stepsize for weight

¹The *rat* function man page: <https://www.mathworks.com/help/matlab/ref/rat.html>.

representation, and the more precisely W can be represented. However, if more of the available axon-neuron pairs not taken up in parallelism are used to extend the threshold, this number is denoted by p , then less are available to extend the range of the neuron weight q . Using equation 3.24, repeated here for clarity: a look-up table of p and q values for a given blocksize, b , is created. This is the maximum possible value of both p and q as a pair, so a smaller value of either could always be used:

$$W \leq \frac{s_{max}}{\alpha} \quad (\text{B.3})$$

where

$$s_{max} = 255 \cdot q \quad (\text{B.4})$$

With this in mind, the search goes as follows:

1. A look-up table of p and q values for a given blocksize, b , is created.
2. Since alpha must be a multiple of 255:

$$\alpha = 255 \cdot p \quad (\text{B.5})$$

Then:

$$s = \text{round}(255 \cdot p \cdot W) \quad (\text{B.6})$$

As α gives the precision, we want to maximise p .

3. However, there is a relationship between s_{max} and s :

$$\begin{aligned} s_{max} &\geq s \\ s_{max} &\geq 255 \cdot p \cdot W \\ \frac{255 \cdot q}{255 \cdot p} &\geq W \\ \therefore \frac{q}{p} &\geq W \end{aligned} \quad (\text{B.7})$$

So p is maximised up to the point where this relationship is no longer true.

4. With p and q assigned, s and α are calculated according to equations B.6 and B.5.

So the optimal weight representation available for a given b is found by maximising p , whilst still obeying the inequality B.7.

Appendix C

Additional information for DV plus jets

This appendix contains supplementary material for the ATLAS analysis described in Chapter 4. Section C.1 shows subsets of the simulated signal grid, section C.2 shows data rates from the ATLAS detector from the HLT stream down to the specific DRAW filter used by the analysis. Section C.3 shows additional material for the study of the efficiency of the DRAW filters and section C.4 shows additional diagnostic plots for the reconstructed DV mass.

C.1 DV plus jets simulated signal grid

The cascade gluino decay used as a baseline signal model for the displaced vertex analysis described in this thesis was simulated over a range of neutrino and gluino masses, see Figure C.1, and four neutrino lifetimes: 0.01, 0.1, 1, and 10 ns (not shown in plot). The black crosses in the plot show the entire grid of mass parameters but only those with blue markers are simulated for initial studies with the aim of selecting additional points from the grid in regions of interest. Figure C.2 shows the subset of grid points used for the DRAW filter studies.

C.2 Data stream rates

In this section some additional material showing the data rates at various points in the ATLAS experiment data chain are presented. Data streams leaving the HLT are shown in Figure C.3 for a typical 12 hour run in October 2018. The rate is dominated by trigger-level analyses whilst the Main Physics stream, from which most data analyses in the ATLAS experiment are derived, is shown in yellow. Whilst the Main Physics stream has a smaller rate than the trigger-level analyses, it dominates the bandwidth of the trigger, as was discussed in Chapter 1. The data rates for the same run, sorted by trigger and focusing this time on the Main Physics stream, are shown in Figure C.4. An event can pass more than one of the trigger categories shown so the overall stream rate is shown by the dashed line. The analysis discussed in Chapter 4 uses data from the Jet stream.

The DRAW_RPVLL data stream is filtered from the physics_main stream at a lower rate, shown in Figure C.5 for a different 12-hour run, from September 2018. Within that the rate of the filter used for this analysis is shown in Figure C.6.

Figure C.7 shows the same filter's acceptance rate vs $\langle\mu\rangle$ for one 2017 run before and after introducing the high p_T flags described in section 4.3.1.

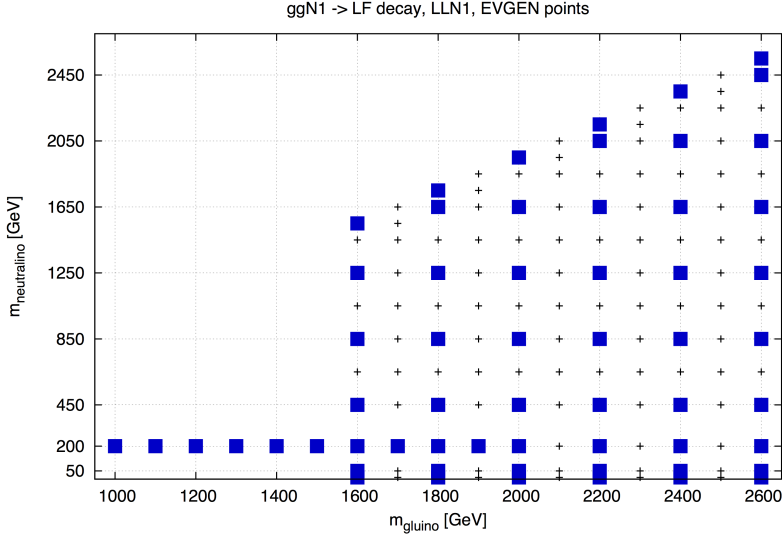


Figure C.1: A plot showing the simulated neutralino and gluino masses for the model depicted in (a). In addition to what is shown in this plot, each point extends into a third dimension of neutralino lifetimes: $\{0.01, 0.1, 1, 10\}$ ns. The plot shows a sparse (blue, square markers) and dense (black crosses and blue squares) grid of mass points. The sparse grid is used for an initial evaluation and any regions of interest can then be investigated in more detail with points from the denser grid.

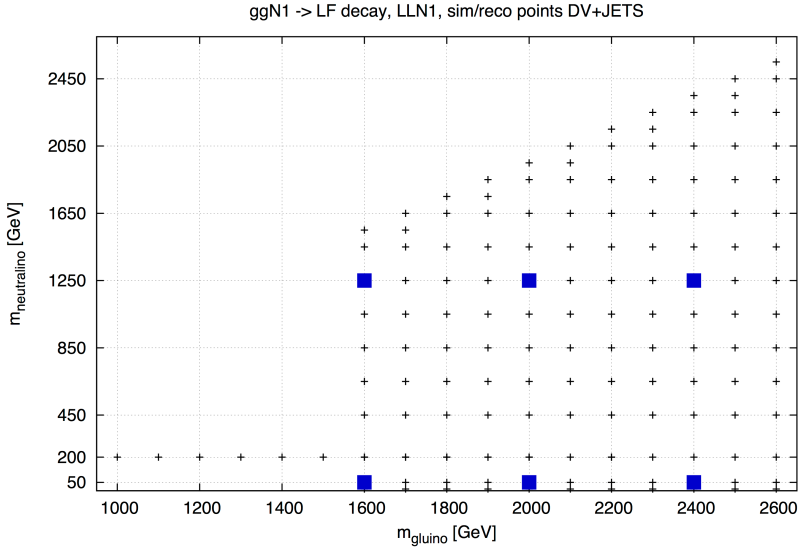


Figure C.2: The subset of the signal parameter grid used for this study (blue squares) overlaying the full signal grid (black crosses). All four neutralino lifetimes $\{0.01, 0.1, 1, 10\}$ ns are also studied but not shown here.

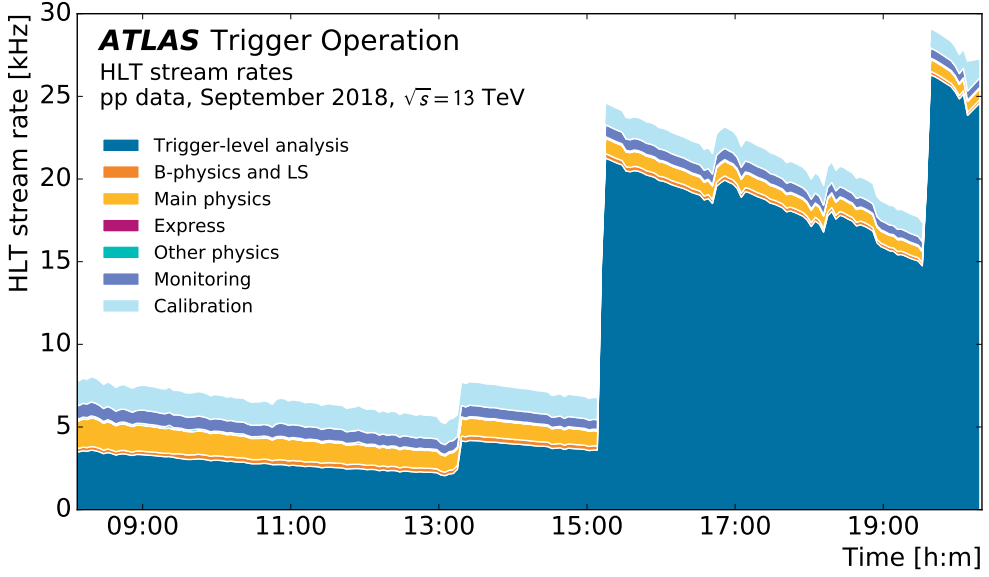


Figure C.3: The rate of data leaving the HLT for one 12-hour, 2018 data run of the ATLAS experiment [37].

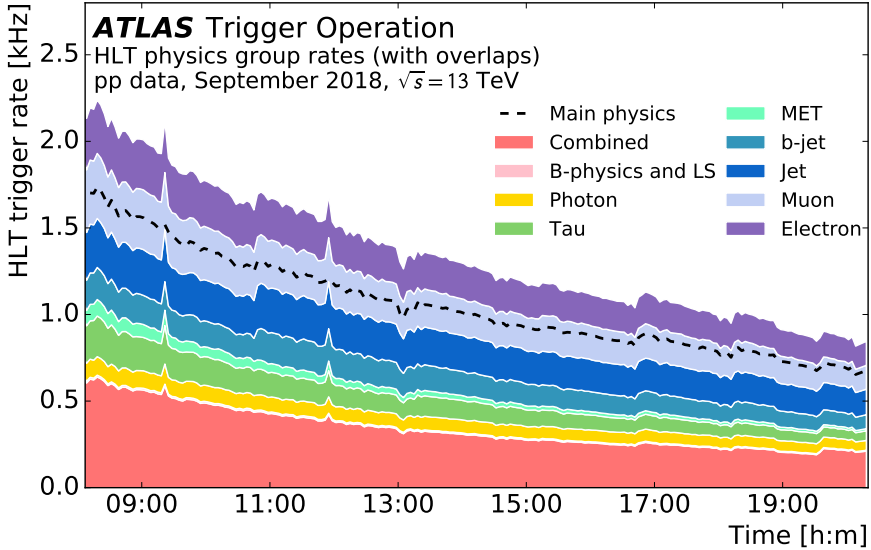


Figure C.4: The data rate in the Main physics stream for one 12-hour, 2018 data run of the ATLAS experiment [37].

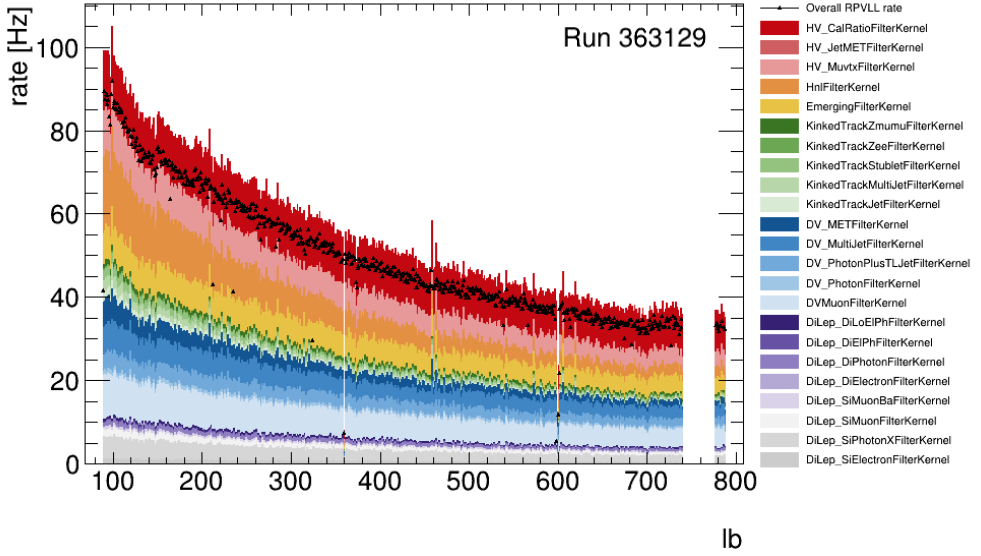


Figure C.5: The data rate of the RPVLL DRAW stream for one 12-hour, 2018 data run of the ATLAS experiment, courtesy of Karri Folan Di Petrillo.

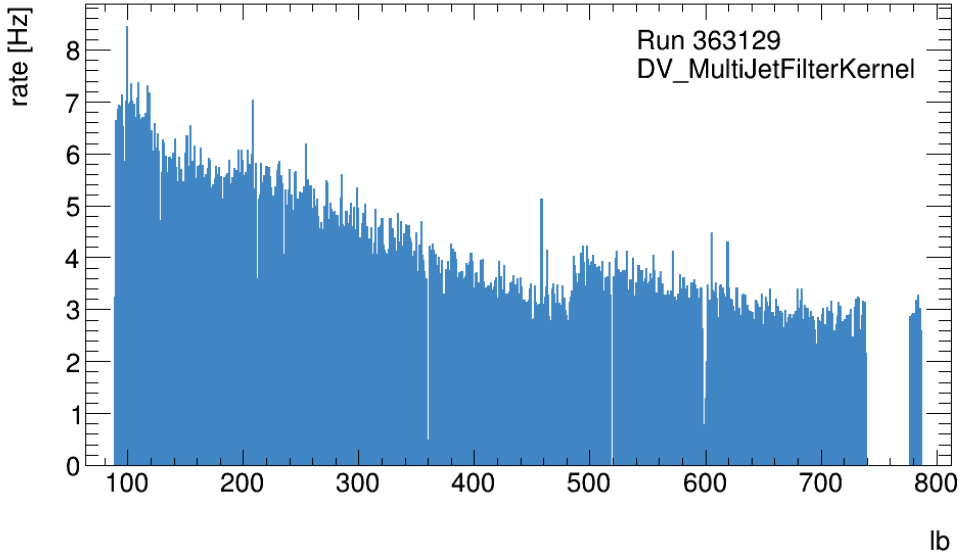


Figure C.6: The data rate of the DVMultiJetFinalFilter stream for one 12-hour, 2018 data run of the ATLAS experiment, courtesy of Karri Folan Di Petrillo.

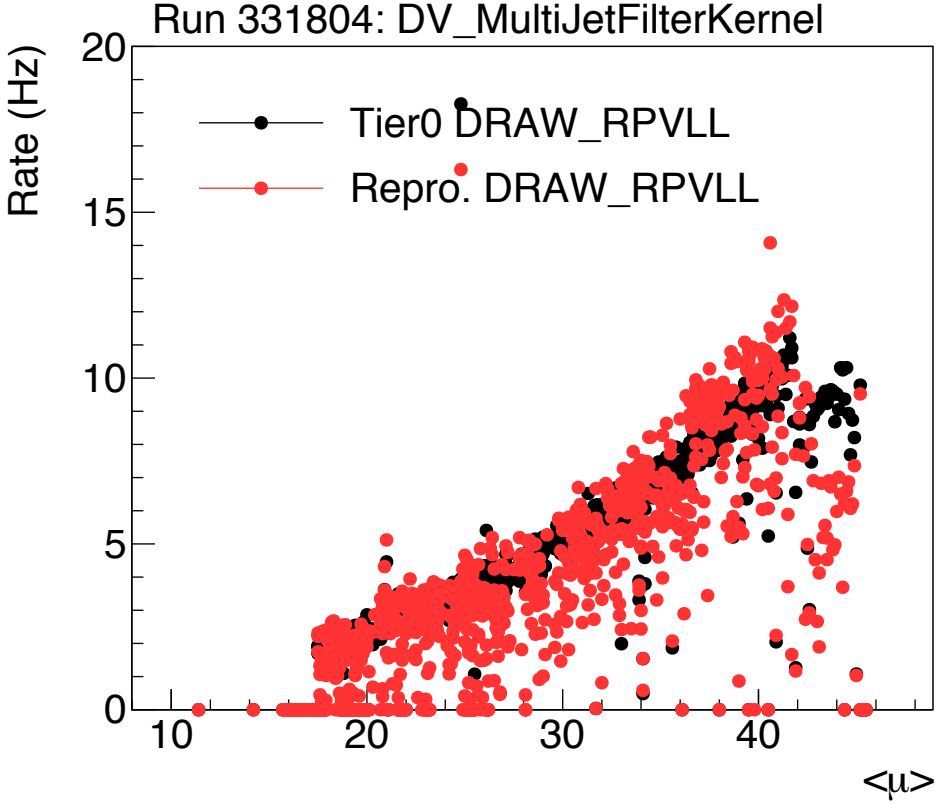


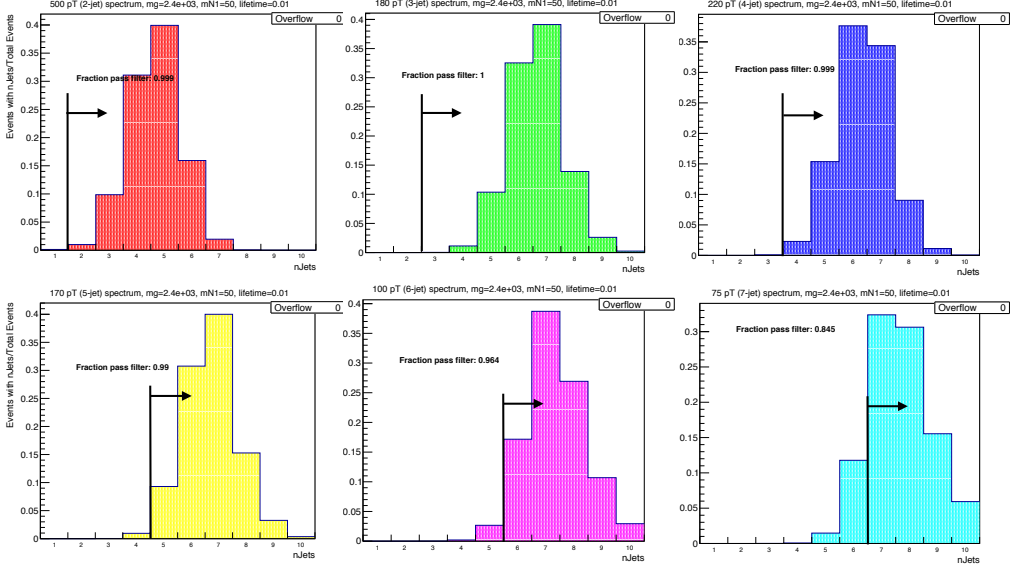
Figure C.7: The DRAW filter acceptance rate vs. average interactions per bunch crossing, $\langle\mu\rangle$ for one 2017 run. The black markers show the data rate for the DVMultiJetFinalTracklessFilter and the red marker for the DVMultiJetFinalFilter, showing an increase by no more than 1%. It should be noted that the increase in double-trackless p_T threshold described later in this section was not implemented in the DVMultiJetFinalFilter at the time this plot was made. Plot courtesy of Karri Folan Di Petrillo.

$\langle\mu\rangle$ range	Single -trackless	25 GeV dbl jet p_T threshold		50 GeV dbl jet p_T threshold	
		DVMultiJetFinal -TracklessFilter	Double -trackless	DVMultiJetFinal -TracklessFilter	Double -trackless
Low $\langle\mu\rangle$: [10–30]	0.626	0.71	0.49	0.629	0.216
Med $\langle\mu\rangle$: [30–50]	0.631	0.775	0.603	0.639	0.234
High $\langle\mu\rangle$: [50–70]	0.625	0.832	0.733	0.627	0.212
$\Delta\langle\mu\rangle = \text{Med} - \text{low}$	-	4.5%	11.3%	1%	2%
$\Delta\langle\mu\rangle = \text{High} - \text{med}$	-	7.7%	13%	1%	2%
$\Delta\langle\mu\rangle = \text{High} - \text{low}$	-	12.2%	24.3%	$\mathcal{O}(1\%)$	$\mathcal{O}(1\%)$

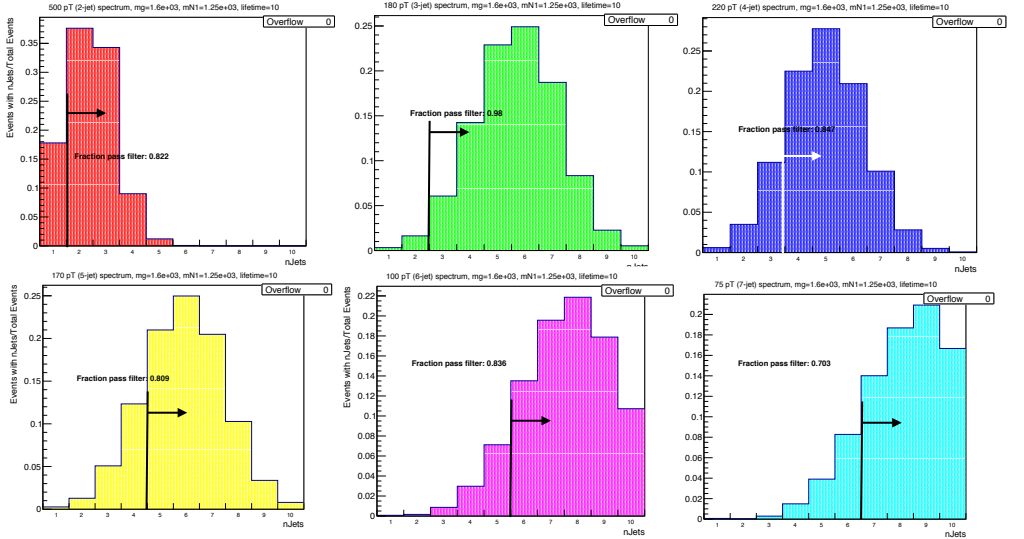
Table C.1: A table demonstrating the pile-up dependency of the double-trackless jet filter, leading to the pile-up dependency of the DVMultiJetFinalTracklessFilter. The first three rows show the acceptance rate of flags that comprise the DVMultiJetFinalTracklessFilter across three ranges of the average interactions per bunch-crossing, $\langle\mu\rangle$. The last three rows show the percentage difference when comparing each flag’s acceptance rate between these ranges. The double-trackless and DVMultiJetFinalTracklessFilter rates are additionally sorted into a set of two columns, the first for a double-trackless p_T threshold of 25 GeV and the second for a double-trackless p_T threshold of 50 GeV.

C.3 DRAW studies

This section shows supplementary information for the DRAW filter studies described in the main text. Table C.1 demonstrates the pile-up dependency of the double trackless filter until the jet p_T threshold was doubled from 25 GeV to 50 GeV. Figure C.8, shows the multiplicity of jets at each p_T threshold in the high p_T jet DRAW flags that were added to the filter in 2017. The figure shows the number of jets in an event with each flag p_T cut as well as the threshold applied (shown as a black or white arrow, pointing in the direction of acceptance) for two different signals. The first is a high mass gluino (2.4 TeV) decaying to a light neutralino (50 GeV) with a short lifetime (0.01 ns), that will likely decay within the fiducial region resulting in an event with a high jet multiplicity with high p_T jets from the large mass difference between the sparticles and the boost of the neutralino. Figure C.8a shows the jet multiplicity spectra for this signal. As the flags are OR-ed together, at least 99.9% of the events in this signal will pass the high p_T part of the filter; the least effective of the flags has a 75 GeV threshold for 7-jets with a pass rate of 84.5%. The second signal point, shown in Figure C.8b, is a lower mass gluino (1.6 TeV) decaying to a heavier neutralino (1250 GeV) with a longer lifetime (10 ns), that may not decay within the fiducial region resulting in an event with a smaller jet multiplicity. In this case the 4-jet flag is the most efficient at 84.7% and the least efficient flag still accepts $\sim 70\%$ of the events.



(a) Jet multiplicity per event for a 2.4 TeV gluino decaying to a 50 GeV neutralino with a 0.01 ns lifetime.



(b) Jet multiplicity per event for a 1.6 TeV gluino decaying to a 1.25 TeV neutralino with a 10 ns lifetime.

Figure C.8: The number of jets per event at each p_T threshold used in the high p_T filter for signal points. The threshold applied in the filter is indicated with an arrow, pointing in the direction of acceptance.

C.4 Displaced vertex reconstruction

This section shows additional material relating to the DV reconstruction and track attachment studies in Chapter 4.3.5. Figure C.9 shows plots relating to track reconstruction and attachment. Figure C.10 contains plots comparing the track-refit accuracy and track attachment purity. Figure C.11 contains additional diagnostic plots exploring the properties of attached and unattached tracks. The first two figures show two different simulated signals for each type of plot: a 2.4 TeV gluino decaying, in the first case, to a 50 GeV neutralino with a lifetime of 0.01 ns, and in the second case to a 1.25 TeV neutralino with a 1 ns lifetime. The plots in the last figure were made using the latter of the simulated signals.

Figures C.9(a) and (b) show how the truth track recovery fraction changes with the reconstructed DV's position in the transverse plane, R . The truth track recovery fraction is defined as the number of truth-matched tracks that are reconstructed and is normalized by the number of charged LLP decay products with $p_T > 1$ GeV. In comparing the 50 GeV and 1.25 TeV neutralinos, it can be seen that the reconstruction efficiency is consistently higher for the lower mass neutralino. For example, for a DV reconstructed at 50 mm from the IP, the mean truth track recovery is between 95–100%, whilst for the heavier neutralino it is between 75–80%.

Figures (c) and (d) show the recovered mass fraction, which is the invariant mass of all truth-matched reconstructed tracks divided by the invariant mass of the charged LLP decay products with $p_T > 1$ GeV, so it shows the effect of the track attachment in plots (a) and (b) on the mass of the DV. Comparing the two plots shows that the mass fraction is a little smaller for the heavier neutralino than the lighter at low R , but both retain a relatively high fraction of mass despite their lost tracks. This efficiency quickly decreases for the heavier neutralino and by $R = 100$ mm the mean mass fraction has dropped to around 60% whereas the lighter neutralino still has a mean of 90%.

Figures (e) and (f) plot the recovered attached fraction against R . This is defined as the number of tracks attached to the DV normalized again to the number of charged LLP decay products with $p_T > 1$ GeV. Comparing (e) and (f) shows a higher recovered attached fraction for the 50 GeV neutralino, for example at around 50 mm, it has a value of 80–85% whereas the heavier neutralino's attachment fraction is around 15% lower.

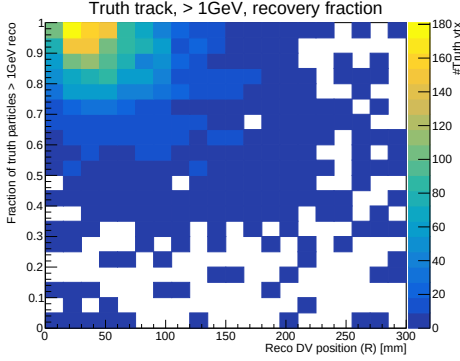
Figure C.10(a) and (b) show the track parameter accuracy and its effect on the DV mass. The plot is made by calculating the invariant mass of the DV from track-fit kinematics, i.e. the reconstructed DV mass, and normalizing it to the truth-matched particle invariant mass. This gives a sense of how faithfully the track fit recovers the underlying particle kinematics, but not the underlying visible LLP mass. Both the lighter and heavier neutralinos see a large spread in values, although it is interesting that the track mass accuracy is more often greater than one (implying the DV mass is over-estimated from fitted particle kinematics) for the lighter neutralino than the heavier. The heavier neutralino tends to have a lower track mass accuracy, which decreases with R , implying that even with efficient reconstruction and attachment, the track parameter fit underestimates the DV mass. All of these factors contribute to the invariant mass of the DV's, shown in Figure 4.21.

Figure C.10(c) and (d) shows the recovered track attachment with increasing LLP decay-product multiplicity. The x -axis shows the number of charged LLP decay products with $p_T > 1$ GeV and the y -axis is the number of truth-matched tracks in the reconstructed DV that was truth matched to the LLP decay in the y -axis. Both the reco DV and LLP are matched for each entry in the plot. This plot explores the combined reconstruction-attachment efficiency as a function of how many particles the LLP decays to. For the 50 GeV neutralino in figure (c), the efficiency appears to be linear with decay product multiplicity but in figure (d) the efficiency appears to be affected by the increasing number of decay products and there is an

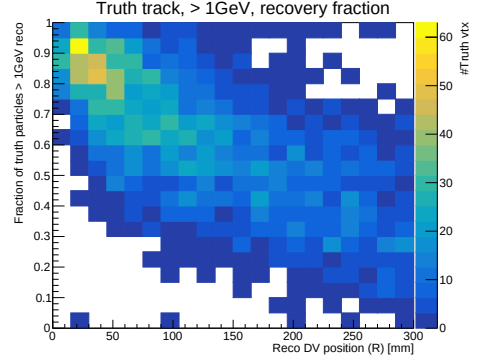
overall reduced efficiency. This implies that the number of decay products also affects the DV reconstruction efficiency, not just the mass and boost of the decaying system.

Figure C.10(e) and (f) show the track composition of the DV in a stacked histogram vs. R . These plots explore the purity of the DV track attachment and show what fraction of each DV's tracks are: truth-matched to an LLP decay product, pile-up tracks or fake (ghost) tracks, or tracks from a different simulated process, i.e. an accidental crossing that is neither fake nor from PU. For both neutralinos the track purity is very high, exceeding 95% for all but the most displaced vertices.

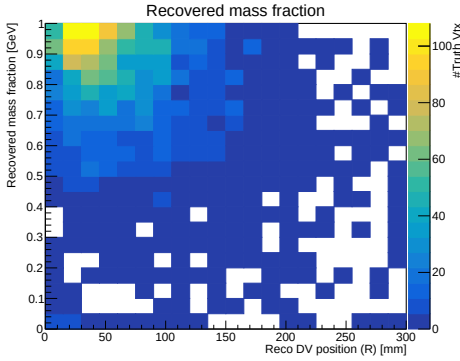
Figure C.11 show additional studies to identify the source of the loss of track attachment in the high-mass neutralino signal sample. For all three plots the solid, black line shows the distribution of the variable for attached tracks, the red, solid line for not-attached tracks, and the dashed red line for not attached tracks with an impact parameter (either transverse or longitudinal) greater than 3 mm. The tracks are truth matched to LLP decay products and for the tracks attached to a DV it is ensured that the reconstructed DV position is within 1 mm of the true LLP decay position. All plots were based on simulated events in which 2.4 TeV gluinos decay to a 1.25 TeV neutralino with a lifetime of 1 ns. Figures (a) and (b) show the p_T and η values of the tracks and in both cases neither the shape nor range of the distribution for the unattached tracks differ to the attached tracks indicating that these values are not the reason for them not being attached. Figures (c) and (d) show derived track variables, $\Delta\phi$ and L , as shown in the cartoon in figure (e), where $\Delta\phi$ is the opening angle the track makes with the vector connecting the DV and PV, and L is the transverse separation from the PV perigee to the DV. Again, neither of these variables show deviations when comparing the attached and unattached distributions implying that the angle and length with respect to the DV is also not the reason for not attaching a track.



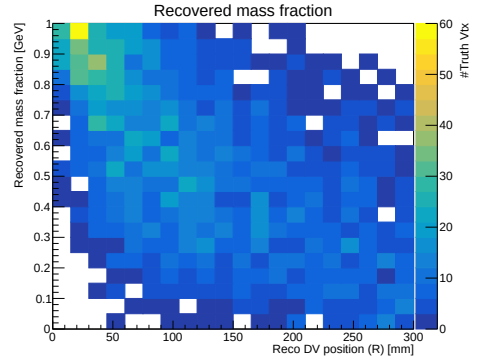
(a) Truth track recovery, 50 GeV neutralino with 0.01 ns lifetime.



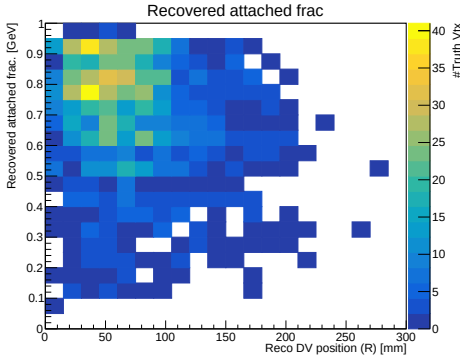
(b) Truth track recovery, 1250 GeV neutralino with 1 ns lifetime.



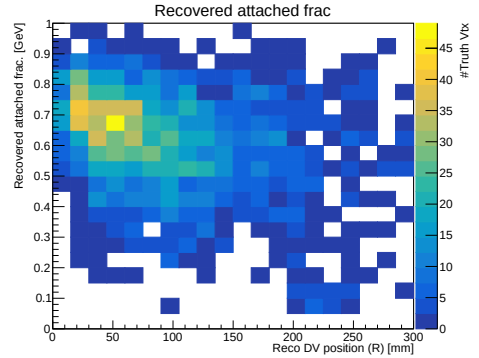
(c) Recovered mass fraction, 50 GeV neutralino with 0.01 ns lifetime.



(d) Recovered mass fraction, 1250 GeV neutralino with 1 ns lifetime.

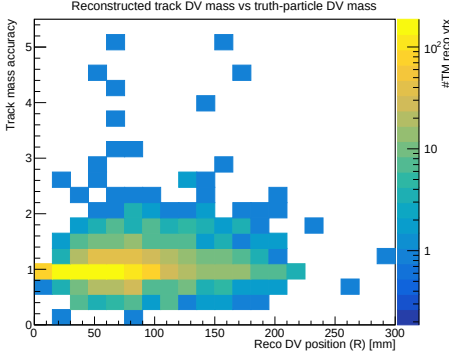


(e) Reconstructed attached fraction, 50 GeV neutralino with 0.01 ns lifetime.

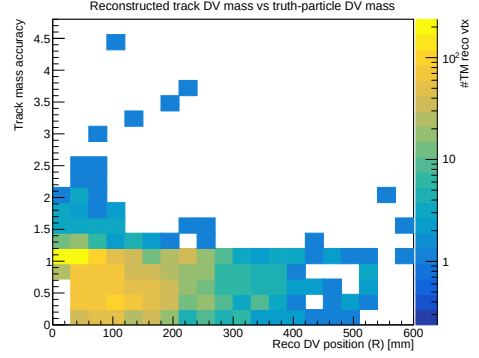


(f) Reconstructed attached fraction, 50 GeV neutralino with 0.01 ns lifetime.

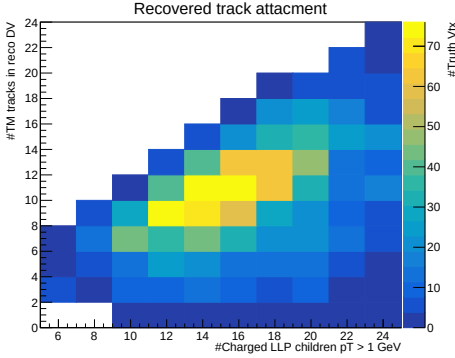
Figure C.9: Plots relating to track reconstruction and attachment for two different simulated neutralino masses and lifetimes, see text for details.



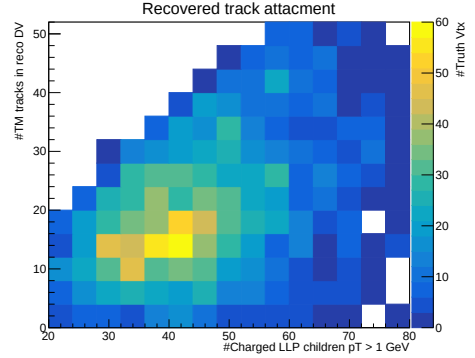
(a) Track parameter accuracy, 50 GeV neutralino with 0.01 ns lifetime.



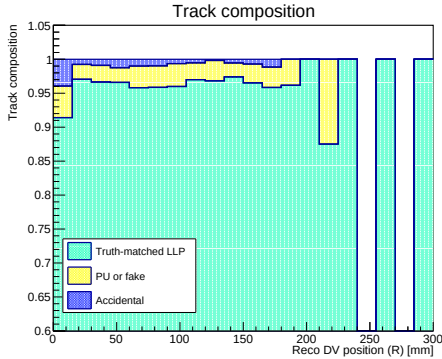
(b) Track parameter accuracy, 1250 GeV neutralino with 1 ns lifetime.



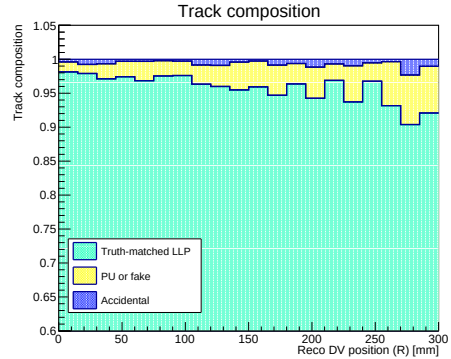
(c) Recovered track attachment vs. LLP decay product multiplicity, 50 GeV neutralino with 0.01 ns lifetime.



(d) Recovered track attachment vs. LLP decay product multiplicity, 1250 GeV neutralino with 1 ns lifetime.

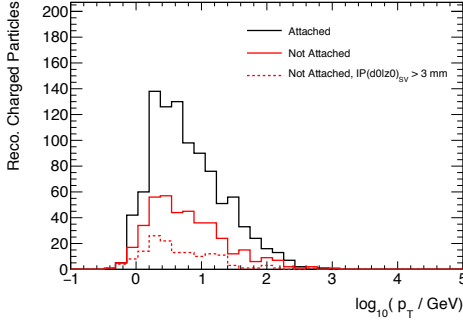


(e) Breakdown of DV track composition, 50 GeV neutralino with 0.01 ns lifetime.

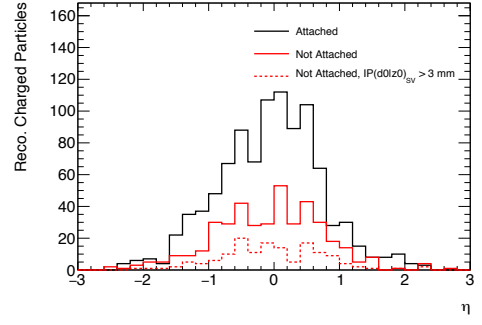


(f) Breakdown of DV track composition, 1250 GeV neutralino with 1 ns lifetime.

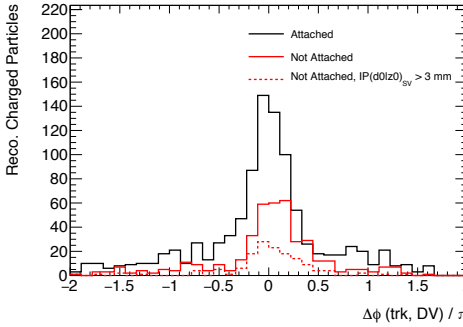
Figure C.10: Plots exploring the effect of the track fit on the reconstructed DV mass, on how the track attachment fraction changes with LLP decay product multiplicity, and on the purity of the tracks in the DV.



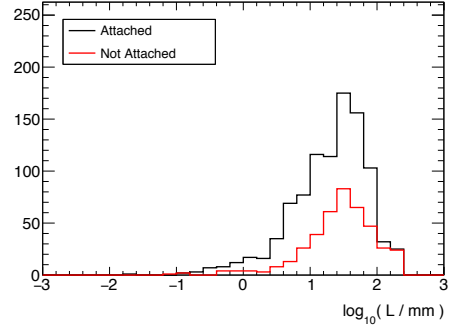
(a) The $\log_{10}(p_T)$ distribution of tracks, measured in GeV.



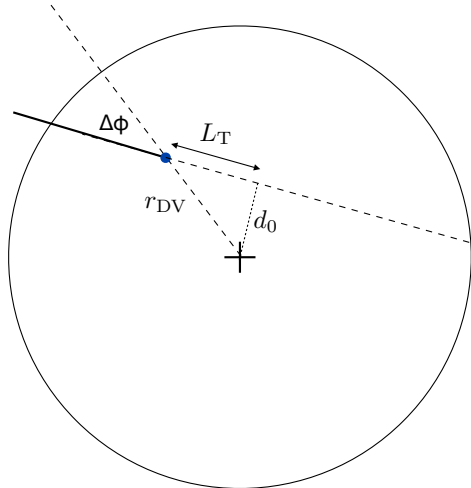
(b) The η , or pseudorapidity, distribution of tracks.



(c) The track opening angle, $\Delta\phi$, with respect to the vector connecting the DV and PV.



(d) The transverse separation, L , between the distance of closest approach and the DV position.



(e) A cartoon illustrating the PV (cross), the DV position (blue circle), the vector connecting PV and DV (large-dashed line), the angle between the track and PV-DV vector ($\Delta\phi$) and the transverse separation, L .

Figure C.11: Plots of attached (solid, black lines) and unattached (solid, red lines) tracks as well as unattached tracks with anomalously large impact parameters where either d_0 or $z_0 > 3$ mm (dash, red lines). 194

Bibliography

- [1] David Galbraith. “Standard model of the Standard Model”. (Image). URL: <http://davidgalbraith.org/portfolio/ux-standard-model-of-the-standard-model/>.
- [2] G. Aad et al. “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”. In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2012.08.020>. URL: <http://www.sciencedirect.com/science/article/pii/S037026931200857X>.
- [3] Serguei Chatrchyan et al. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”. In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].
- [4] T. Aaltonen et al. “Evidence for a Narrow Near-Threshold Structure in the $J/\psi\phi$ Mass Spectrum in $B^+ \rightarrow J/\psi\phi K^+$ Decays”. In: *Phys. Rev. Lett.* 102 (2009), p. 242002. DOI: 10.1103/PhysRevLett.102.242002. arXiv: 0903.2229 [hep-ex].
- [5] Roel Aaij et al. “Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \rightarrow J/\psi K^- p$ Decays”. In: *Phys. Rev. Lett.* 115 (2015), p. 072001. DOI: 10.1103/PhysRevLett.115.072001. arXiv: 1507.03414 [hep-ex].
- [6] Y. Fukuda et al. “Evidence for oscillation of atmospheric neutrinos”. In: *Phys. Rev. Lett.* 81 (1998), pp. 1562–1567. DOI: 10.1103/PhysRevLett.81.1562. arXiv: hep-ex/9807003 [hep-ex].
- [7] *Standard Model Summary Plots Spring 2019*. Tech. rep. ATL-PHYS-PUB-2019-010. Geneva: CERN, 2019. URL: <http://cds.cern.ch/record/2668559>.
- [8] M. Tanabashi et al. “Review of Particle Physics”. In: *Phys. Rev. D* 98.3 (2018), p. 030001. DOI: 10.1103/PhysRevD.98.030001.
- [9] Lyndon Evans and Philip Bryant. “LHC Machine”. In: *JINST* 3 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.
- [10] ATLAS Collaboration public plots. “Luminosity Public Results Run2”. 2019. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2>.
- [11] Mark Thomson. *Modern particle physics*. New York: Cambridge University Press, 2013. ISBN: 9781107034266. URL: <http://www-spines.fnal.gov/spines/find/books/www?cl=QC793.2.T46::2013>.
- [12] A. Airapetian et al. “ATLAS: Detector and physics performance technical design report. Volume 1”. In: (1999).

- [13] J.J. Goodson. “Search for Supersymmetry in States with Large Missing Transverse Momentum and Three Leptons including a Z-Boson”. Presented 17 Apr 2012. PhD thesis. Stony Brook University, 2012.
- [14] Thomas Lenzi. “Development and Study of Different Muon Track Reconstruction Algorithms for the Level-1 Trigger for the CMS Muon Upgrade with GEM Detectors”. MA thesis. U. Brussels (main), 2013. arXiv: 1306.0858 [physics.ins-det]. URL: <https://inspirehep.net/record/1236817/files/arXiv:1306.0858.pdf>.
- [15] G. Aad et al. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.
- [16] Bernard Aubert et al. “Construction, assembly and tests of the ATLAS electromagnetic barrel calorimeter”. In: *Nucl. Instrum. Meth.* A558 (2006), pp. 388–418. DOI: 10.1016/j.nima.2005.11.212.
- [17] Antonio Salvucci. “Measurement of muon momentum resolution of the ATLAS detector”. In: *EPJ Web Conf.* 28 (2012), p. 12039. DOI: 10.1051/epjconf/20122812039. arXiv: 1201.4704 [physics.ins-det].
- [18] M. Aleksa et al. “Measurement of the ATLAS solenoid magnetic field”. In: *JINST* 3 (2008), P04003. DOI: 10.1088/1748-0221/3/04/P04003.
- [19] Joao Pequenaio. Computer generated image of the ATLAS inner detector. 2008.
- [20] M. Backhaus. “The upgraded Pixel Detector of the ATLAS Experiment for Run 2 at the Large Hadron Collider”. In: *Nucl. Instrum. Meth.* A831 (2016), pp. 65–70. DOI: 10.1016/j.nima.2016.05.018.
- [21] *Vertex Reconstruction Performance of the ATLAS Detector at $\sqrt{s} = 13$ TeV*. Tech. rep. ATL-PHYS-PUB-2015-026. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/2037717>.
- [22] Daniel Adam Dobos, Claus Gossling, and Reiner Klingenberg. “Commissioning Perspectives for the ATLAS Pixel Detector”. Presented on 1 Sep 2007. PhD thesis. Dortmund: Dortmund U., 2007. URL: <https://cds.cern.ch/record/1092108>.
- [23] Georges Aad et al. “Operation and performance of the ATLAS semiconductor tracker”. In: *JINST* 9 (2014), P08009. DOI: 10.1088/1748-0221/9/08/P08009. arXiv: 1404.7473 [hep-ex].
- [24] H. Pernegger. “The Pixel Detector of the ATLAS experiment for LHC Run-2”. In: *JINST* 10.06 (2015), p. C06012. DOI: 10.1088/1748-0221/10/06/C06012.
- [25] E. Hines. “Performance of Particle Identification with the ATLAS Transition Radiation Tracker”. In: *Particles and fields. Proceedings, Meeting of the Division of the American Physical Society, DPF 2011, Providence, USA, August 9-13, 2011*. 2011. arXiv: 1109.5925 [physics.ins-det]. URL: <https://inspirehep.net/record/929696/files/arXiv:1109.5925.pdf>.
- [26] Fabio Sauli. *Instrumentation in High Energy Physics*. Vol. 9. Advanced Series on Directions in High Energy Physics. <https://doi.org/10.1142/1356>: World Scientific, 1992.
- [27] P. Strizenec. “Performance of the ATLAS Liquid Argon Calorimeter after three years of LHC operation and plans for a future upgrade”. In: *JINST* 9 (2014), p. C09007. DOI: 10.1088/1748-0221/9/09/C09007.

- [28] Alexander Solodkov. “Performance of the ATLAS Tile Calorimeter”. In: *Proceedings, 3rd Large Hadron Collider Physics Conference (LHCP 2015): St. Petersburg, Russia, August 31-September 5, 2015*. CERN. Geneva: CERN, 2016, pp. 683–688.
- [29] A. Artamonov et al. “The ATLAS forward calorimeters”. In: *JINST* 3 (2008), P02010. DOI: 10.1088/1748-0221/3/02/P02010.
- [30] G. Aad et al. “Commissioning of the ATLAS Muon Spectrometer with Cosmic Rays”. In: *Eur. Phys. J. C* 70 (2010), pp. 875–916. DOI: 10.1140/epjc/s10052-010-1415-2. arXiv: 1006.4384 [physics.ins-det].
- [31] S. Aefsky et al. “The optical alignment system of the ATLAS muon spectrometer end-caps”. In: *JINST* 3 (2008), P11005. DOI: 10.1088/1748-0221/3/11/P11005.
- [32] Theodoros Argyropoulos et al. “Cathode strip chambers in ATLAS: Installation, commissioning and in situ performance”. In: *IEEE Trans. Nucl. Sci.* 56 (2009), pp. 1568–1574. DOI: 10.1109/TNS.2009.2020861.
- [33] Giordano Cattani and the RPC group. “The Resistive Plate Chambers of the ATLAS experiment: performance studies”. In: *Journal of Physics: Conference Series* 280.1 (2011), p. 012001. URL: <http://stacks.iop.org/1742-6596/280/i=1/a=012001>.
- [34] ATLAS Approved Plots: DAQ/HLT. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ApprovedPlotsDAQ>.
- [35] M. zur Nedden. “The LHC Run 2 ATLAS trigger system: design, performance and plans”. In: *JINST* 12.03 (2017), p. C03024. DOI: 10.1088/1748-0221/12/03/C03024.
- [36] Theodor Christian Herwig. “ATLAS jet trigger performance in 2016 data”. In: *PoS ICHEP2016* (2016), p. 854. DOI: 10.22323/1.282.0854.
- [37] ATLAS Collaboration public plots. “Trigger Operation public results”. 2019. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/TriggerOperationPublicResults>. Trigger_rates_and_bandwidth_for.
- [38] M. Lamanna. “The LHC computing grid project at CERN”. In: *Nucl. Instrum. Meth.* A534 (2004), pp. 1–6. DOI: 10.1016/j.nima.2004.07.049.
- [39] “Athena The ATLAS Common Framework”. Developer Guide. URL: <http://www.lnf.infn.it/~gatti/MANUALI/AthenaDeveloperGuide-8.0.0-draft.pdf>.
- [40] E Lund et al. “Track parameter propagation through the application of a new adaptive Runge-Kutta-Nystrom method in the ATLAS experiment”. In: *JINST* 4.ATL-SOFT-PUB-2009-001. ATL-COM-SOFT-2008-006 (2008). Approved for publication by the ATLAS Publication committee, submitted to JINST., P04001. 13 p. URL: <https://cds.cern.ch/record/1113528>.
- [41] The ATLAS collaboration. “A neural network clustering algorithm for the ATLAS silicon pixel detector”. In: *Journal of Instrumentation* 9.09 (2014), P09009. URL: <http://stacks.iop.org/1748-0221/9/i=09/a=P09009>.
- [42] Andreas Salzburger. “Hadron Collider Physics Summer School Tracking Lecture”. (Image) Presented 15th August 2014. 2014. URL: <https://indico.fnal.gov/event/8769/session/12/#20140814>.

- [43] Jovan Mitrevski. “Preparing ATLAS reconstruction software for LHC’s Run 2”. In: *Journal of Physics: Conference Series* 664.7 (2015), p. 072034. URL: <http://stacks.iop.org/1742-6596/664/i=7/a=072034>.
- [44] R. Frühwirth. “Application of Kalman filtering to track and vertex fitting”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 262.2 (1987), pp. 444–450. ISSN: 0168-9002. DOI: [http://dx.doi.org/10.1016/0168-9002\(87\)90887-4](http://dx.doi.org/10.1016/0168-9002(87)90887-4). URL: <http://www.sciencedirect.com/science/article/pii/0168900287908874>.
- [45] P. V. C. Hough. “Machine Analysis Of Bubble Chamber Pictures”. In: *Proceedings, 2nd International Conference on High-Energy Accelerators and Instrumentation, HEACC 1959: CERN, Geneva, Switzerland, September 14-19, 1959*. Vol. C590914. 1959, pp. 554–558. URL: http://inspirehep.net/record/919922/files/HEACC59_598-602.pdf.
- [46] M. Aaboud et al. “Performance of the ATLAS Track Reconstruction Algorithms in Dense Environments in LHC Run 2”. In: *Eur. Phys. J. C* 77.10 (2017), p. 673. DOI: 10.1140/epjc/s10052-017-5225-7. arXiv: 1704.07983 [hep-ex].
- [47] Georges Aad et al. “Charged-particle distributions in $\sqrt{s} = 13$ TeV pp interactions measured with the ATLAS detector at the LHC”. In: *Phys. Lett. B* 758 (2016), pp. 67–88. DOI: 10.1016/j.physletb.2016.04.050. arXiv: 1602.01633 [hep-ex].
- [48] *An imaging algorithm for vertex reconstruction for ATLAS Run-2*. Tech. rep. ATL-PHYS-PUB-2015-008. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/2008700>.
- [49] R. Frühwirth, W. Waltenberger, and P. Vanlaer. “Adaptive vertex fitting”. In: *J. Phys. G* 34 (2007), N343. DOI: 10.1088/0954-3899/34/12/N01.
- [50] Federico Meloni. “Primary vertex reconstruction with the ATLAS detector”. In: *JINST* 11.12 (2016), p. C12060. DOI: 10.1088/1748-0221/11/12/C12060.
- [51] *Performance of the reconstruction of large impact parameter tracks in the ATLAS inner detector*. Tech. rep. ATL-PHYS-PUB-2017-014. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2275635>.
- [52] *Performance of vertex reconstruction algorithms for detection of new long-lived particle decays within the ATLAS inner detector*. Tech. rep. ATL-PHYS-PUB-2019-013. Geneva: CERN, 2019. URL: <https://cds.cern.ch/record/2669425>.
- [53] W Lampl et al. *Calorimeter Clustering Algorithms: Description and Performance*. Tech. rep. ATL-LARG-PUB-2008-002. ATL-COM-LARG-2008-003. Geneva: CERN, 2008. URL: <https://cds.cern.ch/record/1099735>.
- [54] The ATLAS collaboration. *Electron efficiency measurements with the ATLAS detector using the 2015 LHC proton-proton collision data*. Tech. rep. ATLAS-CONF-2016-024. 2016.
- [55] Morad Aaboud et al. “Measurement of the photon identification efficiencies with the ATLAS detector using LHC Run 2 data collected in 2015 and 2016”. In: *Submitted to: Eur. Phys. J.* (2018). arXiv: 1810.05087 [hep-ex].
- [56] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “The anti- k_t jet clustering algorithm”. In: *JHEP* 04 (2008), p. 063. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189 [hep-ph].

- [57] Claudio Santoni. “The reconstruction of jets, missing E T and boosted heavy particles with ATLAS in Run 2”. In: *EPJ Web of Conferences* 126 (Jan. 2016), p. 04043. DOI: 10.1051/epjconf/201612604043.
- [58] Georges Aad et al. “Performance of pile-up mitigation techniques for jets in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector”. In: *Eur. Phys. J. C* 76.11 (2016), p. 581. DOI: 10.1140/epjc/s10052-016-4395-z. arXiv: 1510.03823 [hep-ex].
- [59] Steven Schramm. “ATLAS Jet Reconstruction, Calibration, and Tagging of Lorentz-boosted Objects”. In: *EPJ Web Conf.* 182 (2018), p. 02113. DOI: 10.1051/epjconf/201818202113.
- [60] Morad Aaboud et al. “Measurements of b-jet tagging efficiency with the ATLAS detector using $t\bar{t}$ events at $\sqrt{s} = 13$ TeV”. In: *JHEP* 08 (2018), p. 089. DOI: 10.1007/JHEP08(2018)089. arXiv: 1805.01845 [hep-ex].
- [61] Jochem Snuerink. “The ATLAS Muon Spectrometer: Commissioning and Tracking”. PhD thesis. Twente U., Enschede, 2009.
- [62] Georges Aad et al. “Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ TeV”. In: *Eur. Phys. J. C* 76.5 (2016), p. 292. DOI: 10.1140/epjc/s10052-016-4120-y. arXiv: 1603.05598 [hep-ex].
- [63] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. Tech. rep. CERN-LHCC-2017-021. ATLAS-TDR-030. Geneva: CERN, 2017. URL: <http://cds.cern.ch/record/2285585>.
- [64] Noemi Calace. *ATLAS ITk Layout Design and Optimisation*. Tech. rep. 2016. URL: <https://cds.cern.ch/record/2234847>.
- [65] Mauricio Garcia-Sciveres. *RD53A Integrated Circuit Specifications*. Tech. rep. CERN-RD53-PUB-15-001. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/2113263>.
- [66] ATLAS Collaboration public plots. “Computing and software public plots”. Twiki page. 2019. URL: <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults>.
- [67] I Bird et al. *Update of the Computing Models of the WLCG and the LHC Experiments*. Tech. rep. CERN-LHCC-2014-014. LCG-TDR-002. 2014. URL: <https://cds.cern.ch/record/1695401>.
- [68] Giuseppe Cerati et al. “Kalman filter tracking on parallel architectures”. In: *22nd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2016) San Francisco, CA, October 14-16, 2016*. 2017. arXiv: 1702.06359. URL: <https://inspirehep.net/record/1514547/files/arXiv:1702.06359.pdf>.
- [69] A. S. Grove. *Physics and technology of semiconductor devices*. John Wiley & Sons, 1967. ISBN: 9780471329983.
- [70] Warwick University. *(Image) Semiconductors: The Ideal Nonrectifying Barrier*. URL: <https://www2.warwick.ac.uk/fac/sci/physics/current/postgraduate/regs/mpags/ex5/devices/hetrojunction/ohmic/> (visited on 2011).
- [71] N. Dinu. “Instrumentation on silicon detectors: from properties characterization to applications”. Habilitation à diriger des recherches. Université Paris Sud - Paris XI, Oct. 2013. URL: <https://tel.archives-ouvertes.fr/tel-00872318>.

- [72] Leonardo Rossi et al. *Pixel Detectors*. Ed. by Alexander Chao et al. Particle Acceleration and Detection. Springer, 2006. ISBN: 9783540283324.
- [73] R.H. Richter et al. “Strip detector design for ATLAS and HERA-B using two-dimensional device simulation”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 377.2 (1996), pp. 412–421. ISSN: 0168-9002. DOI: [http://dx.doi.org/10.1016/0168-9002\(96\)00257-4](http://dx.doi.org/10.1016/0168-9002(96)00257-4). URL: <http://www.sciencedirect.com/science/article/pii/S0168900296002574>.
- [74] G. Pellegrini et al. “Technology development of p-type microstrip detectors with radiation hard p-spray isolation”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 566.2 (2006), pp. 360–365. ISSN: 0168-9002. DOI: <http://dx.doi.org/10.1016/j.nima.2006.07.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900206012587>.
- [75] Frank Hartmann. “Evolution of Silicon Sensor Technology in Particle Physics”. In: *Springer Tracts Mod. Phys.* 231 (2009), pp. 1–204.
- [76] T. Rohe et al. “Sensor design for the ATLAS-pixel detector”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 409.1–3 (1998), pp. 224–228. ISSN: 0168-9002. DOI: [http://dx.doi.org/10.1016/S0168-9002\(97\)01266-7](http://dx.doi.org/10.1016/S0168-9002(97)01266-7). URL: <http://www.sciencedirect.com/science/article/pii/S0168900297012667>.
- [77] A. La Rosa et al. “Novel silicon n-in-p pixel sensors for the future ATLAS upgrades”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 718 (2013). Proceedings of the 12th Pisa Meeting on Advanced Detectors La Biodola, Isola d’Elba, Italy, May 20 – 26, 2012, pp. 329–330. ISSN: 0168-9002. DOI: <http://dx.doi.org/10.1016/j.nima.2012.10.091>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900212012624>.
- [78] Mauricio Garcia-Sciveres. “Results of FE65-P2 Pixel Readout Test Chip for High Luminosity LHC Upgrades”. In: *PoS ICHEP2016.CERN-RD53-PROC-16-001* (2016), p. 272. 8. URL: <https://cds.cern.ch/record/2231609>.
- [79] I. Peric et al. “The FEI3 readout chip for the ATLAS pixel detector”. In: *Nucl. Instrum. Meth.* A565 (2006), pp. 178–187. DOI: 10.1016/j.nima.2006.05.032.
- [80] Hans Bichsel. “A method to improve tracking and particle identification in TPCs and silicon detectors”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 562.1 (2006), pp. 154–197. ISSN: 0168-9002. DOI: <http://doi.org/10.1016/j.nima.2006.03.009>.
- [81] Hans Bichsel. “Straggling in thin silicon detectors”. In: *Rev. Mod. Phys.* 60 (3 1988), pp. 663–699. DOI: 10.1103/RevModPhys.60.663. URL: <https://link.aps.org/doi/10.1103/RevModPhys.60.663>.
- [82] Richard Talman. “On the statistics of particle identification using ionization”. In: *Nuclear Instruments and Methods* 159.1 (1979), pp. 189–211. ISSN: 0029-554X. DOI: [http://dx.doi.org/10.1016/0029-554X\(79\)90348-3](http://dx.doi.org/10.1016/0029-554X(79)90348-3). URL: <http://www.sciencedirect.com/science/article/pii/0029554X79903483>.

- [83] “X-ray data booklet”. In: ed. by Albert Thompson and Douglas Vaughan. Lawrence Berkeley National Lab., 2009. Chap. 1. URL: <http://cxro.lbl.gov/x-ray-data-booklet>.
- [84] Hans Bichsel. “Straggling of Heavy Charged Particles: Comparison of Born Hydrogenic-Wave-Function Approximation with Free-Electron Approximation”. In: *Phys. Rev. B* 1 (7 1970), pp. 2854–2862. DOI: 10.1103/PhysRevB.1.2854. URL: <https://link.aps.org/doi/10.1103/PhysRevB.1.2854>.
- [85] George Ausman and Flynn Mclean. “Electron-hole Pair-creation energy in SiO₂”. In: *Applied Physics Letters* 26.173 (1975). DOI: doi.org/10.1063/1.88104.
- [86] Claude Leroy and Pier-Giorgio Rancoita. “Particle interaction and displacement damage in silicon devices operated in radiation environments”. In: *Reports on Progress in Physics* 70.4 (2007), p. 493. URL: <http://stacks.iop.org/0034-4885/70/i=4/a=R01>.
- [87] Michael Moll. “Radiation damage in silicon particle detectors: Microscopic defects and macroscopic properties”. PhD thesis. Hamburg U., 1999. URL: <http://www-library.desy.de/cgi-bin/showprep.pl?desy-thesis99-040>.
- [88] Z. Li and H. W. Kraner. “Studies of the dependence on oxidation thermal processes of effects on the electrical properties of silicon detectors by fast neutron radiation”. In: *Conference Record of the 1991 IEEE Nuclear Science Symposium and Medical Imaging Conference*. 1991, 239–245 vol.1. DOI: 10.1109/NSSMIC.1991.258884.
- [89] Michael Moll. “Development of Radiation Hard Sensors for very High Luminosity Colliders”. Slides. 2002. URL: <http://www.phys.hawaii.edu/vertex2002/vprivate/Moll.pdf>.
- [90] Magdalena Ciurea. “Defects in silicon: From bulk crystals to nanostructures”. In: *Romanian Reports in Physics* 60.3 (2008), pp. 735–478.
- [91] C. A. Londos et al. “Vacancy-oxygen defects in silicon: the impact of isovalent doping”. In: *Journal of Materials Science: Materials in Electronics* 25.6 (2014), pp. 2395–2410. ISSN: 1573-482X. DOI: 10.1007/s10854-014-1947-6. URL: <http://dx.doi.org/10.1007/s10854-014-1947-6>.
- [92] J. Weingarten. “ATLAS IBL sensor qualification”. In: *JINST* 7 (2012), p. C01039. DOI: 10.1088/1748-0221/7/01/C01039.
- [93] Stefano Terzo et al. “Thin n-in-p planar pixel sensors and active edge sensors for the ATLAS upgrade at HL-LHC”. In: *JINST* 9.12 (2014), p. C12029. DOI: 10.1088/1748-0221/9/12/C12029. arXiv: 1409.8579 [physics.ins-det].
- [94] V. Chiochia et al. “A double junction model of irradiated silicon pixel sensors for {LHC}”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 568.1 (2006). New Developments in Radiation Detectors Proceedings of the 10th European Symposium on Semiconductor Detectors 10th European Symposium on Semiconductor Detectors, pp. 51–55. ISSN: 0168-9002. DOI: <http://doi.org/10.1016/j.nima.2006.05.199>. URL: <http://www.sciencedirect.com/science/article/pii/S0168900206010850>.
- [95] B Henrich and R Kaufmann. “Lorentz-angle in irradiated silicon”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 477.1–3 (2002). 5th Int. Conf. on Position-Sensitive Detectors, pp. 304–307. ISSN: 0168-9002. DOI: <https://doi.org/10.1016/S0168->

- 9002(01) 01865–4. URL: <http://www.sciencedirect.com/science/article/pii/S0168900201018654>.
- [96] ATLAS Collaboration. *Modeling Radiation Damage Effects for Pixel Sensors in the ATLAS Detector*. Tech. rep. CERN-EP-2019-061. Geneva: CERN, 2019.
- [97] Tommaso Lari. *Lorentz Angle variation with electric field for ATLAS silicon detectors*. Tech. rep. ATL-INDET-2001-004. INFN, 2001. URL: <http://cds.cern.ch/record/684187/files/indet-2001-004.pdf>.
- [98] M. Moll, E. Fretwurst, and G. Lindström. “Investigation on the improved radiation hardness of silicon detectors with high oxygen concentration”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 439.2–3 (2000), pp. 282–292. ISSN: 0168-9002. DOI: [https://doi.org/10.1016/S0168-9002\(99\)00842-6](https://doi.org/10.1016/S0168-9002(99)00842-6). URL: <http://www.sciencedirect.com/science/article/pii/S0168900299008426>.
- [99] G. Barrand et al. “GAUDI - A software architecture and framework for building HEP data processing applications”. In: *Comput. Phys. Commun.* 140 (2001), pp. 45–55. DOI: 10.1016/S0010-4655(01)00254-5.
- [100] Zachary Marshall and the Atlas Collaboration. “Simulation of Pile-up in the ATLAS Experiment”. In: *Journal of Physics: Conference Series* 513.2 (2014), p. 022024. URL: <http://stacks.iop.org/1742-6596/513/i=2/a=022024>.
- [101] Tadej Novak. *New techniques for pile-up simulation in ATLAS*. Tech. rep. ATL-SOFT-PROC-2018-043. Geneva: CERN, 2018. URL: <https://cds.cern.ch/record/2649298>.
- [102] C. Jacoboni et al. “A review of some charge transport properties of silicon”. In: *Solid-State Electronics* 20.2 (1977), pp. 77–89. ISSN: 0038-1101. DOI: [http://dx.doi.org/10.1016/0038-1101\(77\)90054-5](http://dx.doi.org/10.1016/0038-1101(77)90054-5). URL: <http://www.sciencedirect.com/science/article/pii/0038110177900545>.
- [103] Qi Zeng and Dong Su. “Search of New Physics with Boosted Higgs Boson in Hadronic Final States with ATLAS Detector”. Presented 01 May 2017. 2017. URL: <https://cds.cern.ch/record/2281075>.
- [104] Koen Walters. “Allpix squared user manual”. User guide. URL: <https://project-allpix-squared.web.cern.ch/project-allpix-squared/usermanual/allpix-manual.pdf>.
- [105] Helmuth Spieler. *Semiconductor Detector Systems*. Vol. v.12. Semiconductor Science and Technology. Oxford: Oxford University Press, 2005. ISBN: 9780198527848. URL: <http://www.oup.co.uk/isbn/0-19-852784-5>.
- [106] Simon Ramo. “Currents induced by electron motion”. In: *Proc. Ire.* 27 (1939), pp. 584–585. DOI: 10.1109/JRPROC.1939.228757.
- [107] W. Shockley. “Currents to Conductors Induced by a Moving Point Charge”. In: *Journal of Applied Physics* 9.635 (1938).
- [108] James Jeans. “The mathematical theory of electricity and magnetism”. In: 2nd ed. Cambridge University Press, 1908. Chap. 7, p. 160.

- [109] Helmut Spieler. *Semiconductor detector systems*. Semiconductor Science and Technology. Oxford: Oxford Univ. Press, 2005. URL: <https://cds.cern.ch/record/1010490>.
- [110] Bartosz Paprocki and Janusz Szczepanski. "Efficiency of neural transmission as a function of synaptic noise, threshold, and source characteristics". In: *Biosystems* 105.1 (2011), pp. 62–72. ISSN: 0303-2647. DOI: <https://doi.org/10.1016/j.biosystems.2011.03.005>. URL: <http://www.sciencedirect.com/science/article/pii/S030326471100058X>.
- [111] Dmitri B. Chklovskii, Thomas Schikorski, and Charles F. Stevens. "Wiring Optimization in Cortical Circuits". In: *Neuron* 34.3 (2002), pp. 341–347. ISSN: 0896-6273. DOI: [https://doi.org/10.1016/S0896-6273\(02\)00679-7](https://doi.org/10.1016/S0896-6273(02)00679-7). URL: <http://www.sciencedirect.com/science/article/pii/S0896627302006797>.
- [112] Larry Squire, ed. *Fundamental Neuroscience*. 4th ed. Academic Press, 2012.
- [113] Khan Academy. "Overview of neuron structure and function". (Images).
- [114] A. L. Hodgkin and A. F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of Physiology* 117.4 (1952), pp. 500–544. ISSN: 1469-7793. DOI: 10.1113/jphysiol.1952.sp004764. URL: <http://dx.doi.org/10.1113/jphysiol.1952.sp004764>.
- [115] Peter Diehl and Matthew Cook. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity". In: *Frontiers in Computational Neuroscience* 9 (2015), p. 99. ISSN: 1662-5188. DOI: 10.3389/fncom.2015.00099. URL: <http://journal.frontiersin.org/article/10.3389/fncom.2015.00099>.
- [116] Todd Hylton. "Systems of Neuromorphic Adaptive Plastic Scalable Electronics". (Slides). URL: <https://aeromotores.files.wordpress.com/2011/03/darpa-synapse-program.ppt>.
- [117] Steven K. Esser et al. "Convolutional networks for fast, energy-efficient neuromorphic computing". In: *Proceedings of the National Academy of Sciences* 113.41 (2016), pp. 11441–11446. DOI: 10.1073/pnas.1604850113. eprint: <http://www.pnas.org/content/113/41/11441.full.pdf>. URL: <http://www.pnas.org/content/113/41/11441.abstract>.
- [118] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME—Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.
- [119] Jeroen Schouwenberg. "Kalman Filter Trained Look-Up Tables for the ATLAS Level-1 Topological Missing Transverse Energy Trigger". MA thesis. Radboud University, 2014.
- [120] O. Welch and G. Bishop. *An introduction to the Kalman filter*. Tech. rep. TR 95-041. UNC-CH, 1995. URL: <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>.
- [121] Simon Hayik. *Kalman Filtering and Neural Networks*. Wiley, 2001. ISBN: 9780471369981.
- [122] Chris Eliasmith. *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological systems*. A Bradford Book, 2004. ISBN: 9780262050715.

- [123] A. Andreopoulos et al. “A low-power neurosynaptic implementation of Local Binary Patterns for texture analysis”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016, pp. 4308–4316. DOI: 10.1109/IJCNN.2016.7727762.
- [124] Lawrence Lee et al. “Collider Searches for Long-Lived Particles Beyond the Standard Model”. In: *Prog. Part. Nucl. Phys.* 106 (2019), pp. 210–255. DOI: 10.1016/j.pnpnp.2019.02.006. arXiv: 1810.12602 [hep-ph].
- [125] David Curtin et al. “Long-Lived Particles at the Energy Frontier: The MATHUSLA Physics Case”. In: (2018). arXiv: 1806.07396 [hep-ph].
- [126] Morad Aaboud et al. “Search for heavy charged long-lived particles in proton-proton collisions at $\sqrt{s} = 13$ TeV using an ionisation measurement with the ATLAS detector”. In: *Phys. Lett. B* 788 (2019), pp. 96–116. DOI: 10.1016/j.physletb.2018.10.055. arXiv: 1808.04095 [hep-ex].
- [127] Morad Aaboud et al. “Search for heavy long-lived multicharged particles in proton-proton collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector”. In: *Phys. Rev. D* 99.5 (2019), p. 052003. DOI: 10.1103/PhysRevD.99.052003. arXiv: 1812.03673 [hep-ex].
- [128] Morad Aaboud et al. “Search for heavy charged long-lived particles in the ATLAS detector in 31.6 fb^{-1} of proton-proton collision data at $\sqrt{s} = 13$ TeV”. In: (2019). arXiv: 1902.01636 [hep-ex].
- [129] Georges Aad et al. “Search for magnetic monopoles and stable particles with high electric charges in 8 TeV pp collisions with the ATLAS detector”. In: *Phys. Rev. D* 93.5 (2016), p. 052009. DOI: 10.1103/PhysRevD.93.052009. arXiv: 1509.08059 [hep-ex].
- [130] Georges Aad et al. “Search for long-lived stopped R-hadrons decaying out-of-time with pp collisions using the ATLAS detector”. In: *Phys. Rev. D* 88.11 (2013), p. 112003. DOI: 10.1103/PhysRevD.88.112003. arXiv: 1310.6584 [hep-ex].
- [131] Georges Aad et al. “Search for nonpointing and delayed photons in the diphoton and missing transverse momentum final state in 8 TeV pp collisions at the LHC using the ATLAS detector”. In: *Phys. Rev. D* 90.11 (2014), p. 112005. DOI: 10.1103/PhysRevD.90.112005. arXiv: 1409.5542 [hep-ex].
- [132] Simon Knapen et al. “Tracking down Quirks at the Large Hadron Collider”. In: *Phys. Rev. D* 96.11 (2017), p. 115015. DOI: 10.1103/PhysRevD.96.115015. arXiv: 1708.02243 [hep-ph].
- [133] Morad Aaboud et al. “Search for long-lived charginos based on a disappearing-track signature in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *JHEP* 06 (2018), p. 022. DOI: 10.1007/JHEP06(2018)022. arXiv: 1712.02118 [hep-ex].
- [134] Morad Aaboud et al. “Search for long-lived particles in final states with displaced dimuon vertices in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Rev. D* 99.1 (2019), p. 012001. DOI: 10.1103/PhysRevD.99.012001. arXiv: 1808.03057 [hep-ex].
- [135] Morad Aaboud et al. “Search for long-lived particles produced in pp collisions at $\sqrt{s} = 13$ TeV that decay into displaced hadronic jets in the ATLAS muon spectrometer”. In: *Phys. Rev. D* 99.5 (2019), p. 052005. DOI: 10.1103/PhysRevD.99.052005. arXiv: 1811.07370 [hep-ex].

- [136] *SUSY March 2019 Summary Plot Update*. Tech. rep. ATL-PHYS-PUB-2019-012. Geneva: CERN, 2019. URL: <http://cds.cern.ch/record/2669016>.
- [137] Georges Aad et al. “Search for long-lived, heavy particles in final states with a muon and multi-track displaced vertex in proton-proton collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector”. In: *Phys. Lett. B* 719 (2013), pp. 280–298. DOI: 10.1016/j.physletb.2013.01.042. arXiv: 1210.7451 [hep-ex].
- [138] ATLAS Collaboration. “Search for massive, long-lived particles using multitrack displaced vertices or displaced lepton pairs in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector”. In: *Phys. Rev. D* 92 (7 2015), p. 072004. DOI: 10.1103/PhysRevD.92.072004. URL: <https://link.aps.org/doi/10.1103/PhysRevD.92.072004>.
- [139] Morad Aaboud et al. “Search for long-lived, massive particles in events with displaced vertices and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector”. In: *Phys. Rev. D* 97.5 (2018), p. 052012. DOI: 10.1103/PhysRevD.97.052012. arXiv: 1710.04901 [hep-ex].
- [140] *Search for long-lived, massive particles in events with a displaced vertex and a displaced muon in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector*. Tech. rep. ATLAS-CONF-2019-006. Geneva: CERN, 2019. URL: <https://cds.cern.ch/record/2668377>.
- [141] Christoph Borschensky et al. “Squark and gluino production cross sections in pp collisions at $\sqrt{s} = 13, 14, 33$ and 100 TeV”. In: *Eur. Phys. J. C* 74.12 (2014), p. 3174. DOI: 10.1140/epjc/s10052-014-3174-y. arXiv: 1407.5066 [hep-ph].
- [142] Christoph Borschensky et al. *SUSY Cross Sections for 13, 14, 33 and 100 TeV pp Collisions*. 2019. URL: <https://twiki.cern.ch/twiki/bin/view/LHCPhysics/SUSYCrossSections> (visited on 2019).
- [143] *Reinterpretation of searches for supersymmetry in models with variable R -parity-violating coupling strength and long-lived R -hadrons*. Tech. rep. ATLAS-CONF-2018-003. Geneva: CERN, 2018. URL: <https://cds.cern.ch/record/2308391>.
- [144] Fabio Maltoni and Tim Stelzer. “MadEvent: Automatic event generation with MadGraph”. In: *JHEP* 02 (2003), p. 027. DOI: 10.1088/1126-6708/2003/02/027. arXiv: hep-ph/0208156 [hep-ph].
- [145] J. Alwall et al. “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations”. In: *JHEP* 07 (2014), p. 079. DOI: 10.1007/JHEP07(2014)079. arXiv: 1405.0301 [hep-ph].
- [146] *ATLAS Run 1 Pythia8 tunes*. Tech. rep. ATL-PHYS-PUB-2014-021. Geneva: CERN, 2014. URL: <https://cds.cern.ch/record/1966419>.
- [147] Richard D. Ball et al. “Parton distributions with LHC data”. In: *Nucl. Phys. B* 867 (2013), pp. 244–289. DOI: 10.1016/j.nuclphysb.2012.10.003. arXiv: 1207.1303 [hep-ph].
- [148] Morad Aaboud et al. “Search for R -parity-violating supersymmetric particles in multi-jet final states produced in p - p collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC”. In: *Phys. Lett. B* 785 (2018), pp. 136–158. DOI: 10.1016/j.physletb.2018.08.021. arXiv: 1804.03568 [hep-ex].

- [149] *2015 start-up trigger menu and initial performance assessment of the ATLAS trigger using Run-2 data*. Tech. rep. ATL-DAQ-PUB-2016-001. Geneva: CERN, 2016. URL: <https://cds.cern.ch/record/2136007>.
- [150] *Trigger Menu in 2016*. Tech. rep. ATL-DAQ-PUB-2017-001. Geneva: CERN, 2017. URL: <https://cds.cern.ch/record/2242069>.
- [151] *Trigger Menu in 2017*. Tech. rep. ATL-DAQ-PUB-2018-002. Geneva: CERN, 2018. URL: <https://cds.cern.ch/record/2625986>.
- [152] J Adelman et al. “ATLAS offline data quality monitoring”. In: *Journal of Physics: Conference Series* 219.4 (2010), p. 042018. DOI: 10.1088/1742-6596/219/4/042018. URL: <https://doi.org/10.1088%2F1742-6596%2F219%2F4%2F042018>.
- [153] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. “On the use of charged-track information to subtract neutral pileup”. In: *Phys. Rev. D* 92.arXiv:1404.7353. CERN-PH-TH-2014-052 (2014). Comments: 15 pages + appendix, 10 figures, 014003. 15 p. URL: <https://cds.cern.ch/record/1698963>.
- [154] *Selection of jets produced in 13TeV proton-proton collisions with the ATLAS detector*. Tech. rep. ATLAS-CONF-2015-029. Geneva: CERN, 2015. URL: <https://cds.cern.ch/record/2037702>.
- [155] Chetvorno. “Solid-state electronics band structure”. (Image). 2017. URL: https://commons.wikimedia.org/wiki/File:Solid_state_electronic_band_structure.svg.
- [156] “Basic notions about solids”. (Image). URL: <http://www.porous-35.com/electrochemistry-semiconductors-3.html>.
- [157] TheNoise at English Wikipedia. “Pn-junction in thermal equilibrium”. (Image). URL: <https://commons.wikimedia.org/wiki/File:Pn-junction-equilibrium.png> (visited on 2007).
- [158] Tony Kuphaldt. “Lessons In Electric Circuits”. In: vol. III. 2001. Chap. 2. URL: https://www.ibiblio.org/kuphaldt/electricCircuits/Semi/SEMI_2.html.
- [159] Charles Sodini. “Microelectronic devices and circuits”. In: 2007. Chap. 3 - Semiconductor Physics II. URL: <http://web.mit.edu/6.012/>.
- [160] D. M. Caughey and R. E. Thomas. “Carrier mobilities in silicon empirically related to doping and field”. In: *Proceedings of the IEEE* 55.12 (1967), pp. 2192–2193. ISSN: 0018-9219. DOI: 10.1109/PROC.1967.6123.

Acknowledgements

I am grateful for the support of the Lawrence Berkeley National Laboratory ATLAS group, who have hosted me for five fantastic years and to Stockholm University for hosting this joint programme and for being so flexible given the unique conditions under which this degree was conducted.

I am incredibly fortunate to have been supervised by Sam Silverstein, Sara Strandberg, and Maurice Garcia-Sciveres, each of which have dedicated an incredible amount of time and effort both to the intricate details of my work and to shaping my experience on this degree. I hope you know that the opportunities you've provided me with will echo throughout the course of my life. Thank you so much for giving me the tools to truly start my career.

Simone Pagan Griso, Zach Marshall, and Simon Viel helped me through extensive debugging sessions when I was getting to grips with Athena during my authorship task. Thank you for all the time and genuine care you put into responding to my questions and following up our discussions.

Many thanks to Paolo Calafiura for giving me the freedom to explore TrueNorth and for supporting me throughout the project. Thanks also to Ben Nachman for helping me get on board with radiation damage, for being so quick to follow-up with my Allpix queries, for being so enthusiastic to follow-through with my suggestions. Many thanks to Vakho Tsulaia and Charles Leggett for lending me your time to talk about COOL and the future of computing.

Starting data analysis later than many students was very daunting and I am so fortunate to have been left in the capable hands of Laura Jeanty and Simone Pagan Griso. Thank you so much for being patient with my ignorance (and my pedantic slowness) and for letting me contribute to the extent that you did. I'm looking forward to working again with both of you going forward! Thanks also to my analysis colleagues at CERN, in particular Christian Ohm, Larry Lee, and Hide Oide for answering all of my questions and being so generous with your time.

Thank you to Christophe Clement for finding me remote shifts that worked on the West Coast! I had a great time working with the data quality team in 2018, in particular thanks to Pekka Sinervo, Francesco Spano, and Emma Kuwertz for answering all of my questions and integrating me into the team. I look forward to working with you all again in the future! The time I spent in Stockholm itself was brief but was made all the richer for sharing an office with Lennart and Christian, thank you for being so generous with your space! And thank you to my fellow instrumentation students Dirk, Markus, and Eduardo for keeping me company.

Working at LBNL has made me appreciate the benefits of working in a large collaboration. Every person here has taught me so much, not only about how to do particle physics but also about how to be a scientist in a community. Thank you to Ian Hinchliffe and Marjorie Shapiro for treating me like I was one of the group and helping me through all of the logistics of being a visiting student.

Thanks to Aleksandra Dimitrievska for always having the best snacks and the warmest

advice. Thanks to Peilian Lu for being supportive when I was sad and showing me how to eat a potato in ways I never imagined. Thank you to Laura Jeanty and Karol Krizka for being my plant buddies. Thank you to Lina Galtieri for spoiling me with all of your incredible tales about the glory-days of particle physics and lending me all those books about Alvarez and Fermi - I'm still waiting to read your memoir! Thank you Rhonda Witharm for letting me bond my own chip, all those years ago, and for the incredible stories of your pre-LBL life - I'm also waiting on your memoir!

Thank you to my fellow grad students for putting together the greatest softball team the world has ever know. Whilst the ATLAS Soft Scatterers may have only won one match in their debut season (by forfeit) we sure made up for it in team spirit! Thank you to Jennet Dickinson for being the best cephalopod a friend could imagine. Thanks to Emily Duffield for all the chats and banter. Thanks to Cesar Gonzalez Rentaria for being such a great office buddy and for making me laugh every day. Thanks to Greg Ottino for reminding me to take a break once in a while and for sharing it with me. Thanks to Patrick McCormack and Neha Santpur for being my buddies in pheno. You're both such fierce and fantastic people and I can't wait to work with you again! And of course thanks to the ultimate beluga and my Stockholm buddy, Veronica Wallangen.

Thanks to my family for their ceaseless support.

And to Timon, thanks for always being there. You ground me and shake me up every single day and I can't imagine my life without you.