Development of the Software Trigger and Vertex Locator for the LHCb Upgrade, and a Study of its Sensitivity to $B^0 \to K_1(1270)l^+l^-$ Decays

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor of Philosophy

by

Tom Williams Harrison

Supervisor: Dr David Hutchcroft

March 2022

Department of Physics, University of Liverpool

LHCb Collaboration, CERN

Abstract

Development of the Software Trigger and Vertex Locator for the LHCb Upgrade, and a Study of its Sensitivity to $B^0 \to K_1(1270)l^+l^-$ Decays

This thesis describes a set of work to further the development of the LHCb detector upgrade in three main ways.

For the LHCb trigger upgrade, run 2 simulated data is used to develop a tunable trigger component that can detect B meson daughters at a higher efficiency than the existing trigger, and a method for reducing the computation time of the trigger without loss of efficiency is outlined. The computational performance of various machine learning frameworks is also investigated.

For the VELO upgrade, software is developed to process binary data from the VeloPix readout for debugging and analysis, and is used in a test pulse analysis of the VeloPix ASICs and their associated systems, which found that the chips perform up to rated limits. The quality assurance testing procedure for VeloPix chips is also defined.

Finally, a sensitivity study is carried out into $B^0 \to K_1(1270)l^+l^-$ decay channels, which finds that for $50\,\mathrm{fb}^{-1}$ of LHCb upgrade data the resonant $\mu\mu$ decay is readily resolvable from the developed trigger line, while the resonant ee decay would be resolved under background rejection rates typically seen in similar analyses. The non-resonant decays may be resolvable with very high background rejection, but lack of detection would still provide constraints on beyond-standard-model phenomena.

Acknowledgements

I am hugely grateful to my supervisor David Hutchcroft, who for more than 4 years has never been too busy to answer my (hopefully decreasingly stupid) questions, whether that's at his door, over zoom, or in hundreds and hundreds of emails. He has been a font of knowledge above and beyond what's required of a supervisor, and has made learning about LHCb and its history fun, with facts like how proton bunches pull clouds of electrons from the inside of the LHC beam pipe, or how the LHCb magnet's field was measured by a cool-looking robot.

Credit for any chance of this thesis existing has to ultimately go to my dad, Paul Harrison, who inspired a love of science and physics that has stayed with me for as long as I can even remember.

Thanks to Carsten Welsch and Themis Bowcock, who gave me work in the Liverpool Semiconductor Detector Centre on the Muon g-2 experiment back when I wasn't sure what I was going to end up doing after undergrad. Barry King was a mentor to me at this time (and years before, when he ran the Nuffield scheme for 6th-formers), and also a friend, who would happily sit to talk about Kalman filters, straw tracker stereo angles, and how C++ tries too hard to be all things to all people, when Fortran does what you need. He is sadly no longer with us and is missed by all in the department. It was thanks to working on g-2 that I heard about an extra set of PhD places coming in, and it was thanks to Barry and the great people working in the clean rooms (Dave Sim, Kayleigh Thomson and Mike Wormald) that made me sure I wanted to keep working in the department.

Many thanks to Jeyan Thiyagalingam, who acted as second supervisor to me while he was still at Liverpool, and who gave me his expertise on machine learning.

My absolutely unreserved thanks go to all of the people working at CERN who have been so unbelievably generous with their time in answering questions and giving guidance about whatever I'm working on. While I was working the VELO at CERN, Karol Hennessy was like an unofficial supervisor, and continued to donate considerable time along with Kristof De Bruyn in helping me to take data when I got stuck. Victor Coco also spent time to introduce me to CERN and patiently teach me how to carry out the tile testing process. Many other people also gave me their time to teach me what they know, like Markus Frank and Kazu Akiba clearing up confusions about the VELO's transport and storage data formats.

I've also had phenomenal help and guidance on the trigger and its software by Alex Pearce, as well as Sascha Stahl and Conor Fitzpatrick, who were all incredibly friendly and approachable, even to total strangers asking for help. And this is without mentioning all of the other people on the VELO, DaVinci, and trigger channels on the LHCb Mattermost and mailing lists that have responded to my questions. It is only thanks to these people that I got this far, and only thanks to people like them around the world that anything works at all.

Thanks to Phil, Lauren, James, Joe, Kārlis, Ellis, and everyone else who made living in Meyrin and working at CERN such an incredibly fun time. My proudest achievement while there was hiking to the top of the 1720 m* Crêt de la Neige, which is down to Kārlis convincing me to come with him and then ruthlessly marching to the top with me behind. Likewise, thanks to everyone in the LIV.DAT CDT for being such a great, tight-knit group of friends inside and outside all of the actual training we did.

Thanks to Joe Williams and the others who helped proofread this thesis at the last minute, to whom credit should go for a surprising number of the closing parentheses you'll see.

*Can't escape the si units package even in the acknowledgements... And there's a footnote too.

Finally, to my wife and partner of over ten years, Anna, who has consistently uprooted her own life to accommodate mine, without even needing asking. She moved with me to Oxford for three years, working to help pay our way, before coming back up North, and later coming with me all the way to live in Switzerland. Over the past couple of months while I've been frantically writing my thesis she has made sure that I haven't had to lift a finger around the house*. That's while listening to four and a half years of "why isn't this software working".

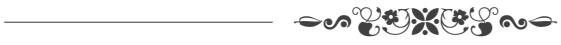
She's a five-sigma lady.

Slightly more abstractly, I'd also like to thank Alexandra Elbakyan, Aaron Swartz, Brewster Kahle, and Jason Scott, for protecting the life and liberty of human knowledge and culture.

Anyway, fun's over, here comes the thesis.

*it is now an absolute dump

Declaration of Authorship



This thesis is the result of my own work, except where a specific reference to the work of others is made. This thesis has not been submitted for any other qualification to this or any other university.

Tom Williams Harrison

To my dog Toby and cat Marnie, who even right now are finding new ways to distract me from work

Contents

List of Figures

List of Tables

Chapter ONE

Introduction

The underlying procedure of physics is to create a theoretical model that approximates the physical world, and test that theory in experiments. The application of a model to the world will have some error based on the situation it is applied to. For a sufficiently high accuracy, or physically extreme circumstances, a less abstracted, more fundamental model must be employed. At the non-cosmological scale, the bedrock of this hierarchy is the Standard Model.

The Standard Model is the name given to a set of quantum field theories that collectively describe the known fundamentals of the universe at the small scale. Although many aspects of the theory's implementation have been revised, the core tenets of the theory are now relatively old, having stood up to scrutiny for multiple decades. It has displayed phenomenal predictive power in many contexts, and the word "Standard" in its name is a testament to its ubiquity, having become the conventional lens for describing the universe at its most fundamental level.

In spite of this increasingly impressive tenure and record of predictability, there are theoretically irreconcilable problems in making the Standard Model play nicely with other observed aspects of the universe, and observed particle physics phenomena that hint at discrepancies between the model and reality. The hunt is still on for a "beyond Standard model" theory that is truly fundamental and

accurate.

The most versatile approach to measuring the properties of particles in a controlled setting has proven to be collider experiments. The higher the energy that particles are scattered off each other, the more deeply the internal structure is probed, providing insights about more fundamental components. From this principle, the field of high-energy physics has emerged. For over a century, scattering experiments of various forms have been carried out, at increasing energies and statistics. The latest, biggest iteration of this process is the Large Hadron Collider, or LHC, at CERN.

The LHCb detector is one of four main experiments at the LHC, and is unique among them in that it is designed to detect particles in the forward region. This capability gives the LHCb detector a view into large numbers of phenomena not afforded to other detector designs.

Chapter 2 of this thesis outlines the Standard Model as the theoretical underpinning of modern high-energy physics, and describes the most prominent tenets and currently recognised limitations of the theory. The physics of hard scattering hadronic collisions and heavy-flavour physics are described as they relate to the LHCb experiment at the LHC.

Chapter 3 describes the LHCb detector. An overview is given of each of the sub-detectors of the previous detector setup, and their dimensions, tolerances and measurement precision. The planned upgrade detector is then introduced, and the differences in design and resolutions of each component are compared to the previous detector. The LHCb computing infrastructure and data flow design are also described, and software and data formats relevant to other chapters are touched upon.

The LHCb trigger system and its potential design post-upgrade are discussed in detail in chapter 4. A machine learning-based multivariate approach to identifying B meson tracks in an event within the first stages of the software trigger is discussed. This is developed based on the existing simulated data and trigger system, with work to port the software to the upgrade software framework. A technique to reduce the computational footprint of events in the trigger without significant loss of efficiency is also developed. A glossary and additional profiling information are in appendix A, along with an overview given of various machine learning methods and their applications.

In chapter 5, the LHCb Vertex Locator upgrade is described in detail, including

its sensor design, frontend electronics, and readout system. The data formats of the readout system are described, and a software system to decode, reorder, translate, and analyse binary output frames from the VELO readout is developed. An analysis of the VeloPix ASIC chips and their readout system is performed via sending test pulse patterns to the VELO module and analysing the output data. The test procedure of VeloPix tiles for quality assurance purposes is also described. Appendix B contains a glossary of terms, data format information, and example software decoder code.

In chapter 6 a study is performed to estimate the upgrade LHCb detector's sensitivity to $B^0 \to K_1(1270)ll$ decays over its expected duration and conditions. The data, software and trigger configuration setup is described, and the resulting efficiencies and total yields of the signal and background are given. A method to enrich the background statistics is described, and for each decay mode, the signal significance under an appropriate mass window is estimated for $50\,\mathrm{fb}^{-1}$. More in-depth methodology is contained in appendix C.

Chapter TWO

Theoretical Description

This chapter will explore the theoretical framework of the Standard Model (also referred to as *SM*), which forms the basis of the currently accepted models of the universe at the small scale. The model has highly accurate predictive power over many phenomena, but there are also examples of where the theory comes into conflict, both with other existing theories and observed phenomena in high-energy physics. The chapter will also outline how hard proton-proton scattering at the LHC leads to hadronisation, and how the LHCb detector is designed to exploit the experimentally attractive features of b-quark hadrons. Finally, the Standard Model's significant suppression of flavour-changing neutral currents (FCNC) is described, and an overview is given of how these processes are most effectively used as a probe of new physics, which serves as a motivation of chapter 6.

2.1 | The Standard Model

The SM is the contemporary theory to describe the propagation and interaction of elementary particles, and is expressed in the form of quantum field theories over gauge symmetry groups. Each elementary particle is identified by a distinct set of properties such as mass, spin, and its various charges, which relate to the particle's coupling to the corresponding interactions.

2.1.1 | Taxonomy

The Standard Model consists of two families of elementary particle: *fermions* and *bosons*. Fermions possess half-integer values for their spin, and bosons possess integer values. Particles with whole-integer spin obey Bose-Einstein statistics, whereby multiple identical particles in an ensemble may occupy the same quantum state. Conversely, particles with half-integer spin obey Fermi-Dirac statistics and are forbidden from occupying an already-filled quantum state, in a phenomenon known as the Pauli exclusion principle [1, 2].

Fermions are divided into *quarks* and *leptons*. Quarks are particles that exhibit all three of the interactions described by the SM (electromagnetic, weak, and strong), with an electric charge of $-\frac{1}{3}$ e for down-type quarks and $+\frac{2}{3}$ e for up-type (with signs reversed for anti-quarks). Quarks are separated into three "generations", each with a positively and negatively charged quark, for a total of six. Leptons, like quarks, couple to the weak interaction, but do not couple to the strong interaction. There are three generations of lepton, each of which is made up of a lepton with electric charge and one without (its corresponding *neutrino*).

Of the elementary boson particles, there are three known types of *gauge bosons*, or force-mediating boson, which all have a spin value of 1. The photon, which mediates the electromagnetic interaction, is massless and has no electric charge, meaning it is not self-interacting. The W and Z bosons are responsible for the weak interaction (the W bosons carry charge as the W^+ and W^- bosons, whereas the Z boson is electrically neutral). Lastly, the gluons (of which there are eight types, see section 2.1.6) mediate the strong interaction, and are self-interacting due to their possession of colour charge. It is currently unknown if the force of gravity exists via a fundamental boson (the "graviton").

Finally, the Higgs boson is the only *scalar* (spin = 0) boson of the SM, and has no electric or colour charge (see section 2.1.11). Unlike the other bosons, the Higgs does not have an associated gauge symmetry, but rather is a product of the electroweak symmetry breaking in the Higgs mechanism [3–5]. For each type of charged particle, there is also a corresponding antiparticle with opposite values for each type of charge.

2.1.2 | Quantum Field Theories

The SM is based on a non-Abelian* gauge theory of gauge group $SU(3)_c \times SU(2)_L \times U(1)_Y$, where c refers to the colour charge of the strong interaction (section 2.1.6), and L and Y refer to the weak isospin and weak hypercharge of the electroweak interaction respectively (section 2.1.11).

The electromagnetic and strong interactions are modelled using quantum electrodynamics (QED) and quantum chromodynamics (QCD) respectively. As quantum field theories, both of these are based on the principle that a particle is a discrete mode of excitation in a particular quantum field, which evolves according to a Lagrangian density function that is invariant under a particular gauge transformation. This gauge symmetry gives rise, via Noether's theorem, to particular conserved quantities of the field.

To calculate the scattering matrix element for a given interaction, the n-particle Green's function is calculated perturbatively [6], which may be expanded according to graphical rules in the form of Feynman diagrams. This means that every possible set of virtual particle momenta is integrated over, and summed over for every possible internal diagram to an acceptable number of leading orders, based on the number of nodes in the diagram.

2.1.3 | Renormalisation

The process of integrating over Feynman diagrams becomes problematic when the integral *diverges*, such that the result becomes unbounded. There are two types of divergence, based on the energy regime:

Infrared (IR) divergence occurs due to a lack of a lower bound on the possible energies of massless particles (typically photons). The total number of massless particles in the interaction lacks an upper bound, meaning that higher-order terms may contribute more than lower-order terms, and the sum becomes non-perturbative.

In this soft photon case, a low-k cutoff may be added into the integral, and the limit taken as this cutoff approaches 0.

Secondly, *Ultraviolet (UV)* divergence occurs as a result of the fact that "virtual" particles inside an interaction in a Feynman diagram are not required to

*a non-Abelian gauge theory is one whose symmetry group does not commute

conform to the usual mass-energy relation* of $E^2 = m^2 + p^2$ (known as being "off-shell"). For the virtual particles comprising internal loops in Feynman diagrams the range of possible momenta is unbounded, causing a divergence when integrating over all momenta. Divergence of this kind occurs when, for a Feynman integral of the form:

$$\int \frac{d^d k}{(2\pi)^4} \frac{N(k)}{M(k)},$$

$d + d_N - d_M \geq 0$, where d_N and d_M are the highest powers of k in $N(k)$ and $M(k)$ respectively [7].

These effects are handled with a number of techniques known collectively as *renormalisation*, involving rescaling the couplings with a chosen constant such that the offending divergence is absorbed out of the useful quantity. In a common technique known as dimensional regularisation, the divergence in the integral is parameterised by modifying the dimension $d = 4 \rightarrow d = 4 - \epsilon$, and evaluating the integral for small ϵ. The field operator and renormalisation constants then pick up dependencies on parameters ϵ and μ [8]. The limit of $\epsilon \rightarrow 0$ is taken, but the *renormalisation scale* μ, related to the energy of the interaction, must be calculated as a free parameter, giving rise to the theoretical uncertainty of field theory models (although this uncertainty is reduced by calculating to higher orders).

The renormalisability of a theory is not guaranteed (see section 2.2) but it is known that all gauge theories with semi-simple Lie group symmetries† are renormalisable [10]. The SM (being a gauge theory with a symmetry group as the product of the simple groups SU(3) and SU(2), and the known-renormalisable U(1) group) has been successfully renormalised [11].

2.1.4 | Running Coupling

The dependence of a coupling constant α_R on the energy scale is given by the β function [12] in terms of the 4-momentum transfer Q^2:

$$\beta(\alpha_R) = Q^2 \frac{\partial \alpha_R}{\partial Q^2}$$

*In this text, all formulae will be written in natural units, such that $c, \hbar = 1$, and their symbols may be omitted from equations. In this system, for example, energy, mass and momentum all have the dimension of energy, conventionally with the units eV.

†A semi-simple Lie group is one whose Lie algebra is semi-simple. That is, the Lie algebra is the direct sum of some number of simple Lie algebras [9].

which may be approximated as a Taylor series in α_R. The energy scale Q^2 is taken to be the aforementioned renormalisation scale μ. This dependence can be measured experimentally via another observable. The comparison of the predicted and observed running couplings provides an important test of a field theory [13].

2.1.5 | Quantum Electrodynamics

Quantum Electrodynamics, or QED, is an Abelian gauge theory with group $U(1)$, describing how the photon interacts with fermions with an electric charge.

The propagator of the photon is a solution to Maxwell's equations, formulated before quantum phenomena were first observed. Similarly, the propagator for a charged spin-$\frac{1}{2}$ particle is the solution to the Dirac equation [14], which was itself devised as a Lorentz-invariant alternative to the Schrödinger equation in quantum mechanics, that might avoid the perceived shortcomings of the Klein-Gordon equation.

The Dirac equation was created as a response to the fact that, as an equation for a single quantum particle, the Klein-Gordon equation is unsuitable, as it allows negative-energy and negative-probability density solutions. (As it happens, the Klein-Gordon has solutions with a scalar (spin-0) field theory, and the Dirac equation has solutions for a spin-$\frac{1}{2}$ field.) For this reason, and to ensure Lorentz invariance, the equation was defined to be first-order in both ∂_t and ∇*.

The general form of the Dirac field equation:

$$i\partial_t\psi(t,\vec{x}) = (-i\vec{\alpha}\cdot\nabla + \beta m)\psi(t,\vec{x}),$$

together with the requirement for solutions to also obey the Klein-Gordon equation (in other words, enforcing the mass-energy relation $E^2 = m^2 + p^2$), means that α_i and β are constrained to take the form of 4×4 matrices. The anticommutation relation:

$$\{\gamma^\mu, \gamma^\nu\} = \gamma^\mu\gamma^\nu + \gamma^\nu\gamma^\mu = 2g^{\mu\nu}$$

imposed by the equations forms the Clifford algebra $Cl_{1,3}(\mathbb{R})$, represented by the Dirac gamma matrices $\gamma^{\{0,1,2,3\}}$, resulting in a covariant-form equation:

$$(i\partial\!\!\!/ - m)\psi(x) = 0$$

*∂_t is defined as the partial derivative with respect to time, $\frac{\partial}{\partial t}$; ∇ is the gradient operator.

where the Feynman notation $\not{\partial} \equiv \gamma^\mu \partial_\mu$* is used [6]. Solutions to $\psi(x)$ are plane waves with 4-component, complex-valued *bispinor* coefficients. Bispinors have components corresponding to the spin-up and spin-down states of the fermion and anti-fermion, with eigenvalues of $\pm\frac{1}{2}$ under the spin operator:

$$\hat{S}_z = \frac{1}{2}\begin{pmatrix} \sigma_z & 0 \\ 0 & \sigma_z \end{pmatrix}$$

2.1.6 | Quantum Chromodynamics

The field theory of Quantum Chromodynamics, or QCD, is a non-Abelian theory, invariant under gauge transformations of the group $SU(3)$. The quark fields consist of a triplet of fields, with 3 corresponding "colour" charges, labelled red, green and blue. Being a special unitary group, there are $N_c^2 - 1 = 8$ generators for the group†, which means there exist 8 gauge bosons to mediate the strong interaction, known as gluons, which form a basis of 8 linearly independent combinations of colour-anticolour charge pairs.

The QCD Lagrangian is as follows:

$$\mathcal{L}_{QCD} = -\frac{1}{4}F^{a\mu\nu}F^a_{\mu\nu} + \bar{\psi}_i(i\not{D}_{ij} - m\delta_{ij})\psi_j$$

with covariant derivative:

$$D^\mu_{ij} = \partial^\mu \delta_{ij} + ig_s t^a_{ij} A^{\alpha\mu}$$

and where:

$$F^a_{\mu\nu} = \partial_\mu A^\alpha_\nu - \partial_\nu A^\alpha_\mu + g_s f^{abc} A^b_\mu A^c_\nu$$

The t^a matrices generate the $SU(3)$ symmetry group and have the algebra $[t^a, t^b] = if^{abc}t^c$, where the "structure constants" f^{abc} are antisymmetric under index permutation [8]. It is conventional to represent the group as $t^a = \frac{1}{2}\gamma^a$, where γ^a are the Gell-Mann matrices [16].

*Throughout this chapter, the Einstein summation convention is used, whereby repetition of indices in a term implies summation over those indices

†In general, for a gauge group G, the matter fields form the defining representation of the group, whereas the gauge boson fields have generators that are the Lie algebra of the group (with dimension $N^2 - 1$ for $SU(N)$ groups), and transform under the adjoint representation of the group under global gauge transformations [15].

2.1.7 | Colour Confinement

The three colour charges are so named because a state made up of a combination of all three is essentially neutrally charged, in the same way that the equal combination of red, green and blue light is perceived by the human eye as "white", neutral light. As well as rgb triplet states (baryons), quarks may also exist in bound states as colour-anticolour pairs (mesons). Two quarks within the bound state of a hadron can be thought to experience a mutually attractive force via the exchange of colour-charged gluons.

Due to the self-interacting nature of gluons, rather than the field strength decreasing with the inverse square of the distance like in the electromagnetic field, the gluon field lines form bundles of flux, causing the potential to increase linearly with the separation distance. At a sufficiently large distance of separation, the potential between the pair exceeds double that of the mass energy of a quark-antiquark pair, allowing such a pair to be created from the vacuum between the original pair of quarks, and forming two new colour-neutral bound states. This phenomenon is known as colour confinement, and is the reason that only colour-neutral free particles are permitted, and all colour-charged matter consists of neutral bound states. The process of colour-neutral pair production may occur many times in the case of a quark produced at high energies, producing a "jet" of hadrons along the trajectory (see section 2.3.2). The phenomenon of confinement within bound hadronic states at low energy is also known as *asymptotic freedom*, in that the field asymptotically approaches a free field theory at higher energies [17].

The fact that the SM describes the strong interaction with an $SU(3)$ symmetry group, rather than a $U(3)$ group, is a statement that there is no colour singlet state of the gluon, represented as $\frac{1}{\sqrt{3}}(r\bar{r} + g\bar{g} + b\bar{b})$. This is due to experimental observation. If a "colourless", singlet-state gluon existed, then it would not be bound by colour confinement, and quarks would be able to interact over arbitrarily long ranges via this non-self-interacting, colour-neutral gluon, similarly to how the electrically-neutral photon mediates the electromagnetic interaction. Since we do not see long-range gluon interactions in this way, the singlet state of gluons is presumed not to exist. In the representation of 3×3 Hermitian matrices (the Lie Algebra of the $U(3)$ group), this is equivalent to removing one of the nine generators, proportional to the 3×3 identity matrix I_3, or in other words, imposing that the trace of the Hermitian matrices is 0. This is the definition of the special unitary group $SU(3)$.

2.1.8 | The CKM matrix

The quark sector of the SM consists of three generations, of different masses and "flavours" (weak eigenstates). These two properties have a small but extremely important distinction: the mass and weak eigenstates do not exactly mutually correspond. Each mass eigenstate is made up of a linear combination of weak eigenstates, and vice-versa. These relationships can be expressed as a 3×3 matrix, known as the Cabibbo-Kobayashi-Maskawa, or *CKM*, matrix [18, 19]:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{2.1}$$

Matrix element V_{ij} describes the relative probability amplitude for a quark in generation j to decay to the opposite-sign charged quark in generation i. Experimental observation by physics collaborations, including LHCb, have determined the values of the matrix to be:

$$\begin{pmatrix} 0.97401 & \pm 0.00011 & 0.22650 & \pm 0.00048 & 0.00361 & {}^{+0.00011}_{-0.00009} \\ 0.22636 & \pm 0.00048 & 0.9732022650 & \pm 0.00011 & 0.04053 & {}^{+0.00083}_{-0.00061} \\ 0.00854 & {}^{+0.00023}_{-0.00016} & 0.03978 & {}^{+0.00082}_{-0.00060} & 0.999172 & {}^{+0000024}_{-0.000035} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}$$

according to the latest figures from the Particle Data Group [20].

The CKM matrix has predominantly diagonal elements, meaning that flavour-changing charged current interactions are less probable (or are *Cabibbo-suppressed*). In particular, the element V_{tb} is particularly dominant in its column. Since the decay from a b quark to a t quark (corresponding to the entry V_{tb} is not possible due to the top quark being more massive than the b quark, any decays of a b quark are heavily suppressed (since the off-diagonal elements in the b column are very small).

The CKM matrix has an analogue in the neutrino sector known as the *PMNS matrix*, expressing the transformation between the mass and weak eigenstates of the three generations of neutrinos. The off-diagonal elements of the matrix, coupled with the very small mass difference between neutrino mass eigenstates, cause neutrinos to oscillate at a measurable frequency between weak eigenstates in flight, without decoherence.

As a unitary, 3×3 matrix, the CKM matrix can actually be described with 4 parameters. One such parameterisation is three mixing angles, $\theta_1, \theta_2, \theta_3$, and a

complex phase, δ, which are written in matrix form as:

$$V = \begin{pmatrix} c_1 & -s_1c_3 & -s_1s_3 \\ s_1c_2 & c_1c_2c_3 - s_2s_3e^{i\delta} & c_1c_2s_3 + s_2c_3e^{i\delta} \\ s_1s_2 & c_1s_2c_3 + c_2s_3e^{i\delta} & c_1s_2s_3 - c_2c_3e^{i\delta} \end{pmatrix}$$

where $c_i \equiv \cos(\theta_i)$, and $s_i \equiv \sin(\theta_i)$.

The CKM matrix is the successor to the 2×2 Cabibbo matrix, which was proposed in the form of a mixing angle θ_c between the weak and mass eigenstates of what are now known to be the first and second generations of quarks [21].

Another common parameterisation of the CKM matrix is the Wolfenstein parameterisation [22]. V_{tb} is taken to be equal to 1. The first parameter is $\lambda \equiv \sin(\theta_c) = V_{us} \approx 0.22$. The remaining matrix elements are then approximated via a polynomial expansion of λ. At order λ^2, the parameter $\pm A$ is introduced as a coefficient of V_{cb} and V_{ts}, and at λ^3, the parameters ρ and η are introduced to modify the complex elements V_{ub} and V_{td}, giving:

$$\begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & \lambda^3 A(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & \lambda^2 A \\ \lambda^3 A(1 - \rho - i\eta) & \lambda^2 A & 1 \end{pmatrix}$$

This parameterisation in λ^3 expresses the matrix in terms of a more well-known constant, with less constrained constants providing the corrective terms at higher order. All 4 parameters have values between 0 and 1.

The unitarity of the CKM matrix means that each set of 2 of the 3 different quark generations have the property $V_{ik}V_{jk}^* = 0$ [23] (using Einstein summation convention). This constraint corresponds to a set of 6 unitary triangles for different choices of i and j, of which a common choice is

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0$$

such that the sides have length 1, $\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}$, and $\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}$ [20].

Figure 2.1 shows the estimated shape of the unitary triangle with the confidence bands of various parameters. The peak of the triangle is at $(\bar{\rho}, \bar{\eta})$, where $\bar{\rho} = \rho(1 - \frac{1}{2}\lambda^2)$ and $\bar{\eta} = \eta(1 - \frac{1}{2}\lambda^2)$ are modified based on a λ^4 extension of the Wolfenstein parameterisation. The interior angles of this triangle are commonly labelled α, β, γ [23].

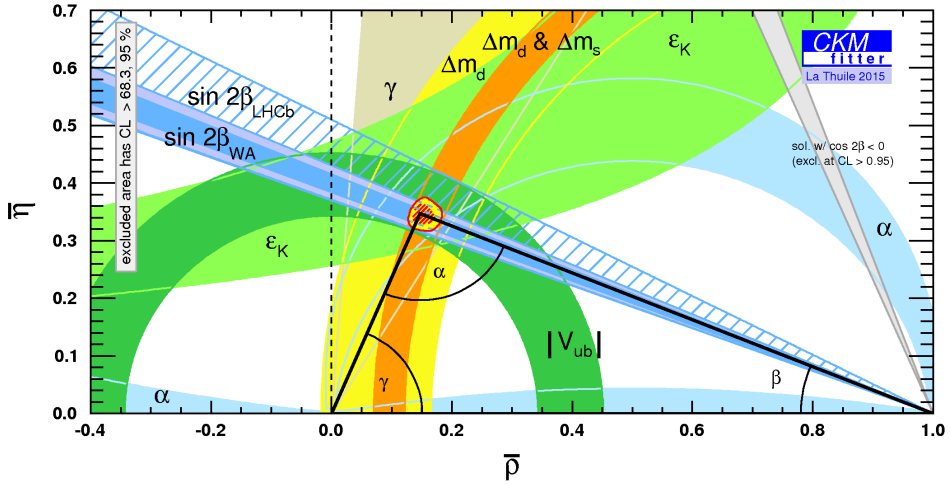


Fig. 2.1

Plot of the confidence intervals for the parameters of the CKM matrix, represented as the unitary triangle [24]. Taken from an LHCb collaboration presentation [25], from an analysis of $B^0 \to J/\psi K^0_S$ decays [26].

2.1.9 | CP violation

Throughout almost all of human history, simple observation and scientific experiment have shown there to be no qualitative difference between the behaviour of physical systems under the inversion of a single spatial coordinate, or *parity transformation*. Such parity conservation was generally regarded as a fundamental property of the universe, until a 1956 experiment found asymmetry in the angular distribution of the β-decay of Co^{60}, indicating a violation of parity conservation under the weak interaction [27]. The concept of parity as a universal symmetry was then amended to be a symmetry of the charge and parity combined, or CP-symmetry [28]. This was also observed to be violated in particular weak interactions, first indirectly, then later directly.

Indirect CP-violation was first observed in an experiment involving K^0_S and K^0_L decays, which have distinct decay modes with opposite CP eigenvalues of two and three pions respectively. With the K^0_L having a much longer decay lifetime than the K^0_S (hence the names "short" and "long"), a source of both kaons with a sufficiently large displacement should not produce two-pion decays, as almost all K^0_S particles should have already decayed. Such decays were observed [29], implying that K^0_S and K^0_L (which form a basis of $d\bar{s}, s\bar{d}$ pairs) have a non-zero mixing angle with their CP eigenstates K_1 and K_2, resulting in a

time-dependent oscillation between the weak eigenstates K_S and K_L. CP-violation has also been observed in other decays (although only involving the weak interaction) such as the oscillation of neutral B mesons, measured by experiments including LHCb [30].

Unlike the time-dependent nature of indirect CP-violation, direct CP violation is a time-integrated effect involving a given decay having a different branching fraction from its CP-inverse, such as that of $\mathcal{B}(B_S^0 \to K^-\pi^+)$ [31].

Each weak and mass eigenstate in equation 2.1 is invariant to its own complex phase. These 6 phases are together degenerate with the global phase, so in all there are $6 - 1 = 5$ parameters that are redundant in the CKM matrix. As the CKM matrix is a 3×3 unitary matrix, it contains $3^2 = 9$ real parameters. Subtracting the 5 parameters that can be removed by a complex phase rotation, this leaves 4 real parameters of the matrix. A real, 3×3 matrix can be rotated with 3 parameters by a change of basis while leaving the physical description the same. Since there are 4 remaining parameters, this means that the CKM has 3 mixing angles and necessarily includes 1 complex phase, that cannot invariantly be removed. This phase results in complex amplitudes between $W^\pm$ and quarks, which is the cause of CP violation in the quark sector [6]. This is also why the rate of CP violation is related to the parameter η in the Wolfenstein parameterisation. For 2 generations of quarks, there is a 2×2 matrix, whose extra parameters can be absorbed by complex rotations in a physically invariant way, leaving only a single (Cabibbo) mixing angle, and no ineradicable complex phase.

The CKM triangle also has area $\frac{J}{2}$, where $J = \Sigma_{m,n}\epsilon_{ikm}\epsilon_{jln}$, another metric for the scale of CP violation [32].

C and CP violation form one of the three Sakharov conditions* — necessary features of baryonic interactions to result in a baryon-asymmetric universe [33]. However, the measured CP-violating phases in the quark and lepton sectors are not sufficient to account for the amount of baryonic matter currently observed in the universe.

2.1.10 | $V - A$ Currents

Prompted by the necessity for a theory accommodating parity violation, the weak interaction, which was previously modelled by Fermi theory as a fully lo-

*The other two conditions are Baryon number violation and thermal inequilibrium.

calised multi-fermion interaction, was described as being mediated by a charged "current" that couples to the electromagnetic and weak interactions, now recognised as the $W^{\pm}$ boson [34]. The current in the Lagrangian is a combination of a vector current $V = \bar{\Psi}\gamma_\mu\Psi$ and an axial vector* $A = \bar{\Psi}\gamma_\mu\gamma_5\Psi$. A weak interaction with a current $V - A = \bar{\Psi}\gamma_\mu(1-\gamma_5)\Psi$ is maximally CP-violating [35]. Equivalently, the current is related to the left-handed chiral projection operator P_L by:

$$\gamma_\mu(1-\gamma_5)\Psi = 2\gamma_\mu P_L\Psi$$

sending amplitudes of right-handed spinors to 0, meaning that the weak interaction in the Standard Model only couples to left-handed particles (and right-handed antiparticles).

Assigning the left-handed fermions the form:

$$Q_L = \frac{1+\gamma_5}{2}\begin{pmatrix} u \\ d \end{pmatrix}, L_L = \frac{1+\gamma_5}{2}\begin{pmatrix} \nu \\ e \end{pmatrix}$$

(where Q and L represent quarks and leptons respectively), the current becomes:

$$J^A_\mu = \bar{\Psi}\gamma_\mu T^A\Psi$$

where T^A are the generators of an $SU(2)$ group with algebra:

$$[T^+, T^-] = [\frac{\sigma^+}{2}, \frac{\sigma^-}{2}] = i\frac{\sigma^3}{2} = iT_3$$

where $\sigma^{\pm}$ are linear combinations of the Pauli matrices:

$$\sigma^{\pm} \equiv \frac{\sigma_1 \pm \sigma_2}{\sqrt{2}}$$

[36].

Theoretical extensions of the Standard Model have been proposed that include $V + A$, or right-handed, charged currents, by extending the electroweak gauge group to the left-right symmetric $SU(2)_L \times SU(2)_R \times U(1)$ [37]. Such a gauge group would be parity-conserving above an energy threshold, and below which would undergo spontaneous symmetry breaking into the maximally CP-violating regime observed today [38].

*An axial vector or pseudovector is a quantity that transforms like a vector under rotation but transforms with the opposite sign under a parity transformation.

2.1.11 | Spontaneous Symmetry Breaking and the Higgs Mechanism

Spontaneous symmetry breaking is the only mechanism that can produce a renormalisable gauge theory with massive, non-Abelian vector bosons (the W and Z bosons). The weak interaction was known to be mediated by massive vector bosons, as its very short range would require a significant boson mass in the Yukawa potential. Prior to symmetry breaking, the electroweak interaction consists of a gauge symmetry group $SU(2) \times U(1)$, which generate three weak isospin W bosons, and one weak hypercharge B boson respectively, all of which are massless.

To spontaneously break the symmetry of this group requires an unstable potential. A doublet complex scalar field:

$$\Phi = \begin{pmatrix} \phi^0 \\ \phi^+ \end{pmatrix}$$

is introduced, with the potential:

$$V(\phi_1, \phi_2) = -\frac{\nu^2}{2}(\phi_1^2 + \phi_2^2) + \frac{\lambda}{4}(\phi_1^2 + \phi_2^2)^2$$

(written here as two real fields rather than one complex field). This is unstable on the point at $\phi_1, \phi_2 = 0$. A field in this potential will spontaneously collapse to a vacuum expectation value v, which may be in ϕ_1 without loss of generality:

$$\phi_1(x) = v + \chi(x)$$
$$\phi_2(x) = \theta(x)$$

Similarly, adding a "Mexican hat-shaped" potential to the Lagrangian of the Higgs:

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F_{\mu\nu} + (D_\mu\phi) * D_\mu\phi - [-\mu^2\phi * \phi + \lambda(\phi^*\phi)^2]$$

gives a scalar field of $A_\mu^{(\nu)} = 0, \phi^{(v)} = \frac{1}{\sqrt{2}}v$. Restating the Lagrangian in terms of perturbations in $B\mu$ and χ gives:

$$\mathcal{L}^{(2)} = -\frac{1}{4}B_{\mu\nu}^2 + \frac{e^2v^2}{2}B_\mu B_\mu + \frac{1}{2}(\partial_\mu\chi)^2 - \mu^2\chi_2$$

where $B_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu$.

The previous 4 massless vector bosons has been split into the massive $W^\pm$ and Z^0 vector bosons, which occupy the longitudinal polarisations in the Higgs

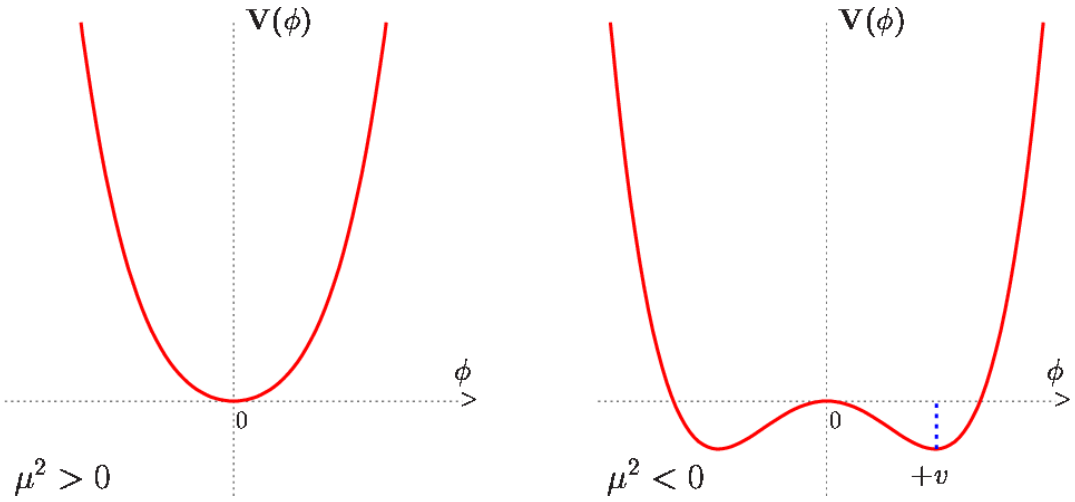


Fig. 2.2

Graph illustrating the transition between a stable extremum at the origin (left), and an unstable one, which will undergo spontaneous symmetry breaking (right) similar to a phase transition, and will settle on a new minimum at ϕ=v [39].

potential in the radial directions (up and down the sides of the hat), whereas the final boson (the photon) remains massless, polarised along the degenerate ring of the hat. The symmetry of $SU(2)_L \times U(1)_Y$ has been broken into simply $U(1)_{EM}$, leaving a single spin-1 Higgs field [3].

The vacuum expectation value (VEV) of the Higgs field via the Higgs mechanism is:

$$v = \frac{1}{\sqrt{\sqrt{2}G_F^0}} = 246.22GeV$$

where G_F^0 is the reduced Fermi constant. Direct detection of the Higgs boson was discovered by the ATLAS and CMS collaborations at the LHC in 2012 [40, 41].

2.2 | Limitations of the Standard Model

Many aspects of the Standard Model have proven enormously successful. Measurements of certain quantities in the natural world show extremely precise agreement with their SM predictions, such as the electromagnetic fine structure constant α, measured to a relative uncertainty of 8.1×10^{-11} [42].

However, there are current theoretical limitations with the Standard Model.

Most notably is that a unified theory of the SM with a theory of gravity has not been successful, as the theory of general relativity is non-renormalisable, due to having a coupling constant of negative dimension in mass: $8\pi G = M_P^{-2}$, where M_P is the Planck mass, and G is Newton's gravitational constant [43]. The negative dimension allows for the creation of an infinite number of Feynman diagrams with *superficial degree of divergence* [6]. This does not fundamentally rule out a unification of the two theories, but means that the standard procedures of renormalisation cannot be applied to resolve divergences.

There are also apparent experimental shortcomings to the Standard Model. For example, despite QED having predicted the anomalous magnetic dipole moment of the electron to incredible accuracy, research from the Muon g-2 project at the Brookhaven and Fermilab laboratories has revealed an apparent deviation of the anomalous magnetic dipole moment of the muon from the theoretical prediction, to a significance of 4.2σ [44]. LHCb data also shows a 3.1σ deviation in the value of the constant* R_K from the SM prediction [45] (these anomalies are not thought to be connected despite both relating to the muon).

Further, ways in which the Standard Model helps to explain certain phenomena are not sufficiently accounted for by the theory. The measured complex phase in the CKM matrix does not provide an adequate amount of CP violation in the quark sector to explain the observed baryon asymmetry in the universe. Another cosmological phenomenon unaccounted for by the SM is an explanation for the existence of dark matter and dark energy. There are multiple dark matter candidates that are beyond the standard model, such as Majorana neutrinos, but there has so far been no direct supporting evidence for any candidate.

2.3 | Physics of the LHC

2.3.1 | Scattering

The elastic scattering cross section of a point charge off a massive, finite-size charged particle is related to the square of the momentum transfer q^2 via:

$$\frac{d\sigma}{d\Omega} \propto |F(q^2)|^2$$

*The double ratio of the branching fractions of $B^+ \to K^+\mu^+\mu^-$ to $B^+ \to K^+e^+e^-$, between their non-resonant and resonant (via J/ψ) modes.

where the form factor F is the Fourier transform of the probed particle's charge distribution $F(q^2) = \int \rho(\vec{r}) e^{i\vec{q}\cdot\vec{r}} d^3\vec{r}$. Consequently, the "shape" of a particle may be observed by how its scattering cross section scales with the scattering energy level, with a independence of q^2 being indicative of a point charge (since $\mathcal{F}(1) = \delta(r)$). The high-energy scattering of electrons off protons gives a form factor indicative of interactions with point-like constituents of the proton, known as deep inelastic scattering [46], and the component *partons* are known to be quarks and gluons. At lower energies, an incident particle will probe only the *valence* partons, the quarks that identify a species of hadron. However, at higher energies the particle will interact with a parton "sea" of quarks and gluons, in ratios that depend on q^2. At such high energies, the momenta of the constituent partons are approximately collinear with the momentum of the whole hadron in the centre-of-mass frame, each carrying a fraction x of the whole momentum. The probability of finding a particular species of parton is dependent on the longitudinal momentum fraction x that the interacting parton carries. These functions $f(x)$ are the parton distribution functions [6], and change with q^2, shaping the physics of hadronic scattering experiments as colliders move to higher energies.

The high centre-of-mass energies at the LHC mean that collisions are typically inelastic, which is the source of all physics performed at all LHC experiments.

Hadronic collisions that produce physics of interest are ones in which a parton pair from two colliding hadrons (such as a quark-antiquark pair) impart a large amount of transverse momentum, in an interaction that is faster than the parton interactions inside the hadron. This hard scattering process allows the parton pair to escape their respective hadrons [6]. The rest of the interacting hadrons do so both between themselves and with the products of the hard scatter. They are non-perturbative, low p_T interactions, dubbed the *underlying event* [47].

2.3.2 | Hadronisation

Quark-antiquark pairs produced in proton-proton collisions undergo *hadronisation*, in which quarks combine to form hadronic bound states. The theoretical procedure by which this occurs is currently not fully understood, with phenomenological models describing the process.

Each oppositely charged quark produced in a pair production event in hard proton-proton scattering becomes a large shower of hadronic particles in a cone centred along the direction of travel, known as a *jet* [48].

A prominent model describing partonic hadronisation is the Lund string model [49]. As mentioned in section 2.1.7, the self-interaction of gluons as mediating particles of the strong interaction means that high-energy quarks travelling from the interaction point produce a linear potential along their path, in the shape of a string. A section of the "string" with potential difference more than double the rest mass of a given quark will "break" to produce a quark-antiquark pair.

The Lund model describes a $q\bar{q}$ pair as massless particles in $1+1$ dimensions, with the only non-vanishing part of the gluon field existing with a constant strength κ over the space between them. The Hamiltonian of the particles 1 and 2 is therefore:

$$H = T + V = |p_1| + |p_2| + \kappa|x_1 - x_2|$$

with the equation of motion [49]:

$$\frac{dp}{dt} = \pm\kappa$$

This equation of motion describes a relativistic massless string [50], in which the two quarks move like a "yo-yo", according to the spacetime diagram in figure 2.3.

Figure 2.4 demonstrates how a quark-antiquark pair $q_0\bar{q}_0$ produced with a sufficiently high centre-of-mass energy will travel apart, increasing the size of the potential between them, until at some point (x_1, t_1) a new pair $q_1\bar{q}_1$ is produced. The size of the new, split potential between q_0 and $\bar{q}_1$ is given by L_1. Given sufficiently high L_1, another pair $q_2\bar{q}_2$ is produced between q_0 and $\bar{q}_1$ at (x_2, t_2). In this case, the centre-of-mass energy of $\bar{q}_1$ and q_2 is low enough that they form a bound state as a hadron. This process repeats recursively until all quarks exist as bound hadronic states [49].

The kinematics of this process are constrained by the energy-momentum relations of the produced particles. In this example, the energy of the hadron $\bar{q}_1 q_2$ is $\kappa(x_2 - x_1)$, and the momentum is $\kappa(t_2 - t_1)$. Therefore, for a hadronic rest mass m, the relativistic energy-momentum relation gives:

$$(x_2 - x_1)^2 - (t_2 - t_2)^2 = \frac{m^2}{\kappa^2}$$

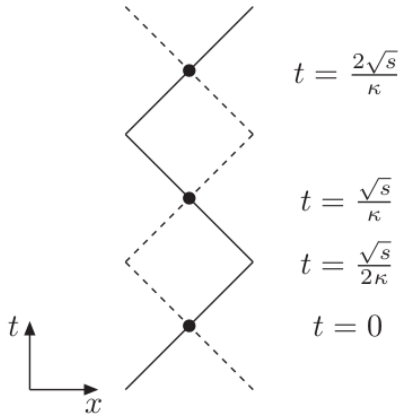


Fig. 2.3

Spacetime diagram of the motion of a $q\bar{q}$ pair, according to the simple Lund string model [51].

This constrains the point (x_2, t_2) to lie on a hyperbola in spacetime relative to (x_1, t_1), as traced by the line H_1 on the diagram.

It is worth nothing that the top has a decay width from the weak interaction of approximately $\Gamma_t = 2.0\,\text{GeV}$ [52]. Being larger than the QCD scale Λ_{QCD}, this means that the top quark is unique amongst quarks in that it effectively does not undergo hadronisation.

The Monte Carlo particle generation software Pythia [53, 54] employs the Lund string model for the calculation of fragmentation fractions [55]. Each hadron produced removes some fraction z of the total available momentum, which occurs until the remaining momentum in the light cone is not sufficient to produce further hadrons. The momentum fraction z follows the probabilistic distribution:

$$f(z) = \frac{(1-z)^a}{z} \exp\left[\frac{-bm_T^2}{z}\right]$$

known as the fragmentation function, where $m_T \equiv \sqrt{m^2 + p_T^2}$ is the "transverse mass" of the event, and a and b are tunable parameters with defaults of 0.68 and 0.98 respectively [55].

For heavy quark hadronisation, Pythia uses an additional factor on $f(z)$:

$$\left(\frac{1}{z}\right)^{r_Q \cdot b \cdot m_Q^2}$$

, where m_Q is the mass of the heavy quark, and r_Q is a factor based on the

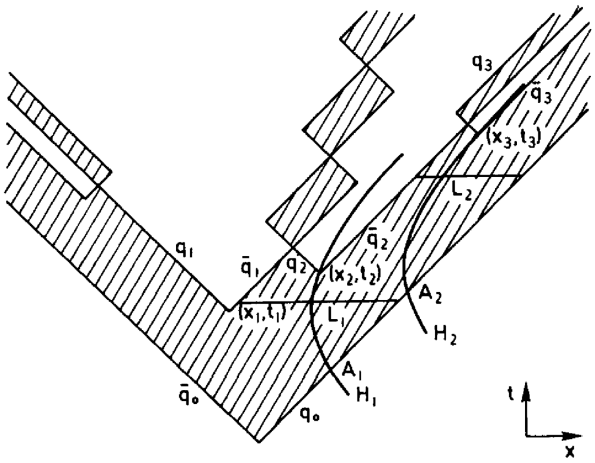


Fig. 2.4

Spacetime diagram illustrating a simple example of a hadronisation process from a $q\bar{q}$ pair produced at a single point in (x, t) [49].

quark flavour, with default of $r_c = 1.32$, $r_b = 0.855$, and $r_h = 1.0$, where h represents a hypothetical or custom quark. This is the *Bowler Modification* [56], which recognises that the straight-line "yo-yo" shape of bound hadronic motion in spacetime represents the massless limit. For heavier quarks, the true path is bent inwards from the rectangular asymptotic path, which requires the aforementioned correction factor due to the modified kinematics.

2.3.3 | Fragmentation Fractions

In a $b\bar{b}$ pair production in a proton-proton collision, the *fragmentation fraction* (or *production fraction*, or *hadronisation fraction*) f_X is the probability that the $\bar{b}$ quark binds to form a hadron of type X [57]. A $\bar{b}$ quark may bind with a u, d, s, or c quark to produce a meson, or may form a bound state as a baryon. The fragmentation fractions for these bound states are f_u, f_d, f_s, f_c, and f_{baryon} respectively. Naturally, $f_u + f_d + f_s + f_c + f_{\text{baryon}} = 1$.

2.4 | Heavy-Flavour Physics

Heavy flavour physics is typically defined as the physics of bound-state-forming high-mass fermions. In the quark sector, this means b and c quarks, as they are significantly more massive than the other light quarks (the t quark is too massive to hadronise, as its decay width is large enough such that it decays faster than the time scale of hadronisation, and so it is omitted from the defini-

tion). The aim of heavy-flavour physics experiments such as LHCb is often to perform precision measurements particle properties and standard model parameters, as well as indirect searches of exotic matter and beyond-standard-model processes.

2.4.1 | Desirable Properties of Heavy-Flavour Analysis

b quarks have a number of properties that are desirable in collider experiments. Their high mass allows a large number of decays to occur with a diverse phenomenology. A high mass generally implies a large decay width, however the Cabibbo suppression of the b quark means that it has a much longer lifetime, with all b bound meson states with light quarks having mean lifetimes of close to 1.5 ps. This results in a characteristic, detector-scale displacement of the B meson decay vertex from the primary interaction point, which is easy to identify and allows high-efficiency selection of B decays. The larger b lifetime also allows the oscillation of neutral B mesons to be measured, providing a probe of CP violation. The large mass of the b quark also means that $b\bar{b}$ pairs produced in the high-p_T hard scattering process remain in the forward region of higher pseudorapidity than lighter prompt-produced mesons. This enables their detection by forward-region detector designs like LHCb, which have more precise mass resolutions than low-η general purpose detectors [58]. Many B meson decay modes have other useful properties, such as $B \to DK$, which provides a measurement of the unitary triangle angle γ. It is dominated by the tree-level contribution, making it theoretically clean, and there are multiple observable final states that can be measured [59].

2.5 | Flavour-Changing Neutral Currents

After electroweak symmetry breaking, the mass states of the quarks are obtained by a change of basis according to the CKM matrix. As the gauge generators $T^{\pm}$ corresponding to the $W^{\pm}$ bosons (discussed in section 2.1.10) are off-diagonal, they pick up this mixing, corresponding to the phenomenon of flavour-changing charged currents. Conversely, the T_3 generator corresponding to the Z^0 boson is diagonal, meaning that neutral bosons do not affect flavour [36]. As such, flavour-changing neutral current (FCNC) processes are highly suppressed under the Standard Model. There are two leading contributions to FCNC in the standard model. One is a box diagram, whereby a quark changes

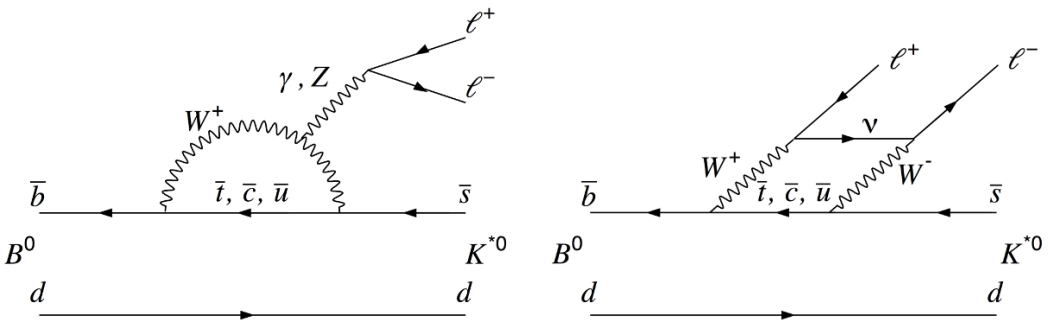


Fig. 2.5

Leading-order Feynman diagrams of flavour-changing neutral currents, displaying a electroweak penguin diagram [61] (left) and a box diagram [60] (right) [62].

flavour by interacting with a W^+ and W^- boson, which themselves exchange a neutrino in the interaction with a lepton-antilepton pair (the GIM mechanism [60]. The second is the electroweak penguin diagram [61], so named because of its (dubious) resemblance to a penguin, in which the quark changes flavour twice in a loop involving a $W^{\pm}$, and the temporary quark emits a boson that undergoes pair production of a fermion-antifermion pair.

These processes (displayed in figure 2.5 provide an excellent probe of new physics, as their strong suppression under the Standard Model means that any beyond-Standard-Model FCNC processes contributing to the same decay would be much more readily apparent. Assuming that a new particle (such as a flavour-changing neutral boson $Z\prime$) may not couple equally to all generations of leptons, studies of FCNC provide a test of lepton universality [62].

2.5.1 | Polarisation

The fact that the weak interaction is strictly left-handed in the Standard Model means that, in FCNC electroweak penguin diagrams in which the W boson emits a photon, the photon should have a left-handed polarisation. The polarisation of the photon leaves artefacts in the leptonic final state. This mean that the angular analysis of $b \to s\gamma$ decays can be used to probe for $V + A$ (right-handed) currents with beyond-Standard-Model origins such as the left-right symmetric model [63]. One method of angular analysis is the measurement of the forward-backward asymmetry [64]:

$$A_{\mathrm{FB}} \equiv \frac{N_F - N_B}{N_F + N_B}$$

where N_F is the number of events measured in the forwards direction, and N_B the number measured in the backwards direction

Measurement of the zero-crossing point of A_{FB} with respect to the q^2 of the vector current (found from its decay products) provides a very sensitive probe for such new physics.

2.5.2 | Parity Doubling

Analyses of the angular distribution and production rates of FCNC decays have been shown to be useful as a search for beyond-Standard-Model $V + A$ processes. These have included the decay of a B meson to a light vector meson* (such as K^*), and either a photon or lepton-antilepton pair. However $V + A$ amplitudes are contaminated by non-perturbative long-distance $V - A$ contributions that decrease the signal purity, leading to higher, difficult to control uncertainties. The equivalent decay to the *axial-vector*† (eg $K_1(1270)$) parity partner of the meson is special in that, by being close to a parity-degenerate final state, it introduces an almost equal but opposite-sign contribution to the long-distance $V - A$ contributions to the right-handed amplitudes. For this reason, measurements of right handed currents via the time-dependent rates of a parity pair of mesons may result in a much cleaner probe, and a tighter constraint on right-handed current theories [65].

*Vector mesons have $J^P = 1^-$

†Axial(-vector) mesons have $J^P = 1^+$

Chapter THREE

The LHCb Detector at the LHC

This chapter will outline the state of the LHCb detector as it has existed prior to second "long shutdown" phase of the LHC. It will then explore the design of the upgraded detector to be used in the upcoming third run of data taking. In particular, the design of the upgraded vertex locator (VeLo) subdetector is described in further detail in chapter 5.

3.1 Overview

The LHCb detector is a single-arm, forward-region spectrometer and general-purpose detector. The detector has a pseudorapidity* range of $1.6 < \eta < 4.9$ [58]. The LHCb has a general forward acceptance angle at which it is able to detect particles of 300 mrad horizontally and 250 mrad vertically, corresponding to $\eta = 1.6$, with $\eta > 4.9$ covering the hole left through the centre of the detector for the beam pipe.

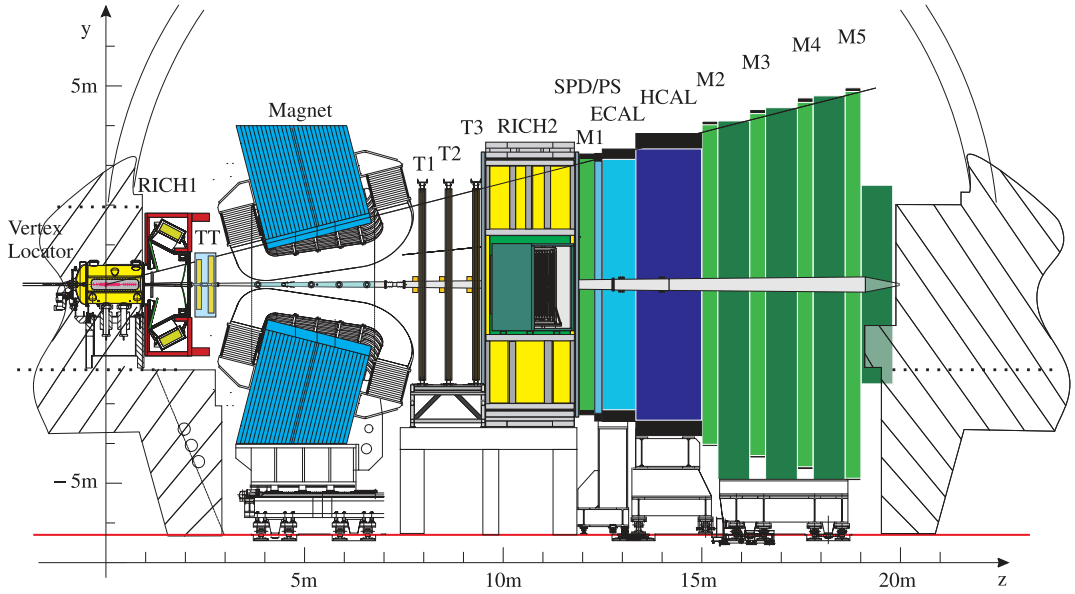


Fig. 3.1

Cross section of the LHCb detector as it existed for the first 2 runs of the LHC [66]. The M1 station was not part of the detector for some of the duration of run 2.

3.2 | LHCb Run 2 Detector

3.2.1 | VELO

The VELO, or *VErtex LOcator*, is the subdetector of LHCb that collision products travel through first after the interaction. It is designed to provide precise locations of a large number of hits, in order to accurately reconstruct particle tracks, and the locations of the primary and secondary vertices they originated from. Hits are detected over a series of 25 planar detector stations over the range $-17.5\,\text{cm} < z < 75\,\text{cm}$, each consisting of a set of silicon strips covering the xy plane orthogonal to the beam, ensuring that all tracks within the $1.6 < \eta < 4.9$ acceptance region produce at least 3 hits [67]. The geometry of the silicon strips is described in section 5.1.1.

3.2.2 | RICH

After exiting the VELO, particles pass through the first RICH (*Ring Imaging Cherenkov*) detector. A ring-imaging Cherenkov detector involves measuring

*defined as $\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$

the photons emitted as a ring by a charged particle as it passes through a dielectric medium, or "radiator" [68]. The angle of the cone of photons emitted is related to the speed of the particle in the medium by the formula:

$$\cos\theta = \frac{1}{\beta n}$$

where $\beta \equiv \frac{v}{c}$ and n is the refractive index of the medium. Given a known particle momentum from the LHCb detector's spectrometer capabilities (section 3.2.3), the particle's mass can be determined, aiding particle identification (PID), in particular the differentiation between charged π and K particles.

Both RICH subdetectors (figure 3.2) are comprised of a frustum of dielectric medium with a spherical mirror around the beamline, which along with a secondary, flat mirror, redirects light onto a plane of hybrid photon detectors (HPD), using reverse-biased silicon pixels. The HPDs of RICH 1 and RICH 2 are magnetically shielded to maximum field strength of $20 \times 10^{-14}\,\mathrm{T}$ and $20 \times 10^{-14}\,\mathrm{T}$ respectively [58, 69].

The RICH 1 subdetector uses a 5 cm-thick aerogel pane and C_4F_{10} gas as radiators, with a momentum range of $1 - 60\,\mathrm{GeV}$, and an acceptance of $\pm 300\,\mathrm{mrad}$ horizontal, $\pm 250\,\mathrm{mrad}$ vertical.

RICH 2 is downstream from the magnet. With a CF_4 gas radiator, it allows for PID in the range of $15 - 100\,\mathrm{GeV}$ at an acceptance angle of $\pm 120\,\mathrm{mrad}$ horizontal, $\pm 100\,\mathrm{mrad}$ vertical [58].

3.2.3 | Magnet

The LHCb dipole magnet provides the magnetic field that allows for the measurement of the momentum of charged particles passing through the detector. It takes the form of 2 saddle-shaped, non-superconducting Al-99.7 coils positioned symmetrically above and below the beamline, with a 1500 ton low-carbon steel yoke framed around the beamline [58, 70]. The magnet provides a vertical* magnetic field which can be used in either polarity (known as "MagUp" and "MagDown") to control bilateral systematic effects in the detector, which is particularly important in the measurement of CP asymmetry [58].

*Technically, the magnetic field is in $\pm y$, rather than strictly vertical, as the LHCb coordinate system matches the minor tilt of the beam pipe.

Fig. 3.2

2D schematics of the RICH 1 (left) and RICH 2 (right) subdetectors

Due to the vertical magnetic field, charged particles are deflected in the transverse direction of the horizontal plane, giving rise to the larger detector acceptance of 300 mrad horizontally, compared to 250 mrad vertically. The bending power of the magnet (integrated magnetic field strength for 10 m long tracks) is $\int Bdl = 4\,\mathrm{Tm}$, with a non-uniformity of $< \pm 5\%$. The magnet dissipates 4.2 MW of power with an excitation current of 2×1.3 MA [71], and so requires active water cooling.

In order to determine the bending power of the magnetic field, the magnetic field strength was robotically measured at many points in each orientation along the beamline via Hall probes (figure 3.4).

3.2.4 Tracking Stations

The tracking stations of the LHCb detector downstream (higher z) from the VELO are separated in z into two groups. Upstream (lower z) of the magnet is the 4-layered TT, or *Tracker Turicensis* (a backronym from the abbreviation, which originally stood for the now defunct name "Trigger Tracker" [72]). On the other side of the magnet are the three composite tracking stations $T1-T3$. The T stations are comprised of an inner silicon strip detector closer to the beam

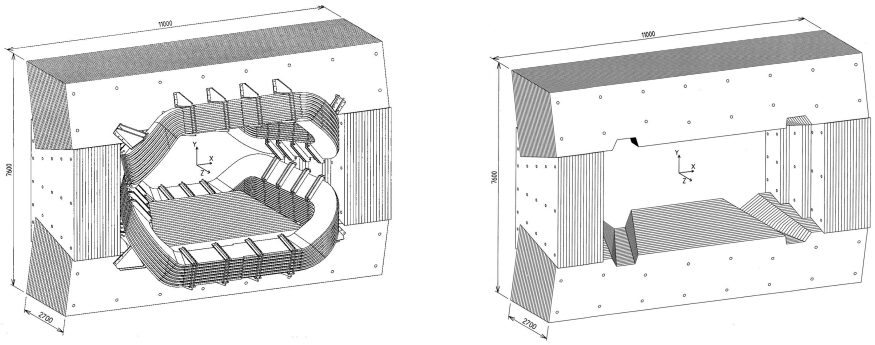


Fig. 3.3

The LHCb magnet (left) and the yoke only (right), both shown without shims [70]

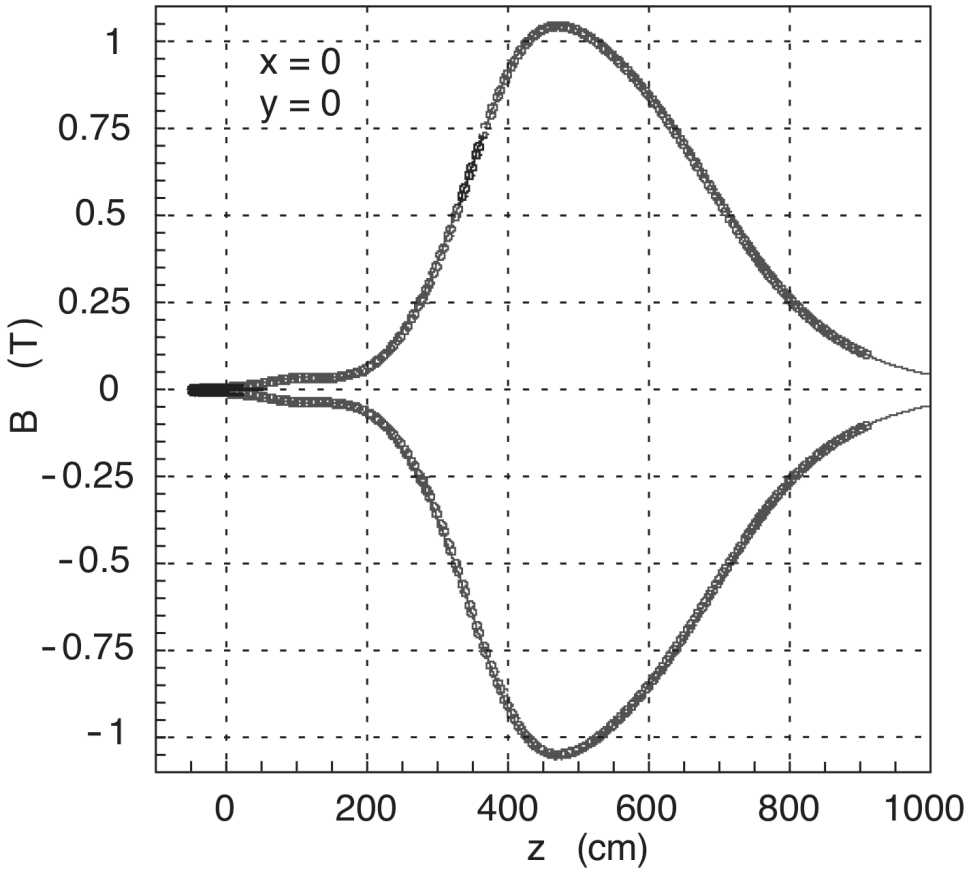


Fig. 3.4

The vertical component of the magnetic field B_y as a function of position along the beamline (z) [58]

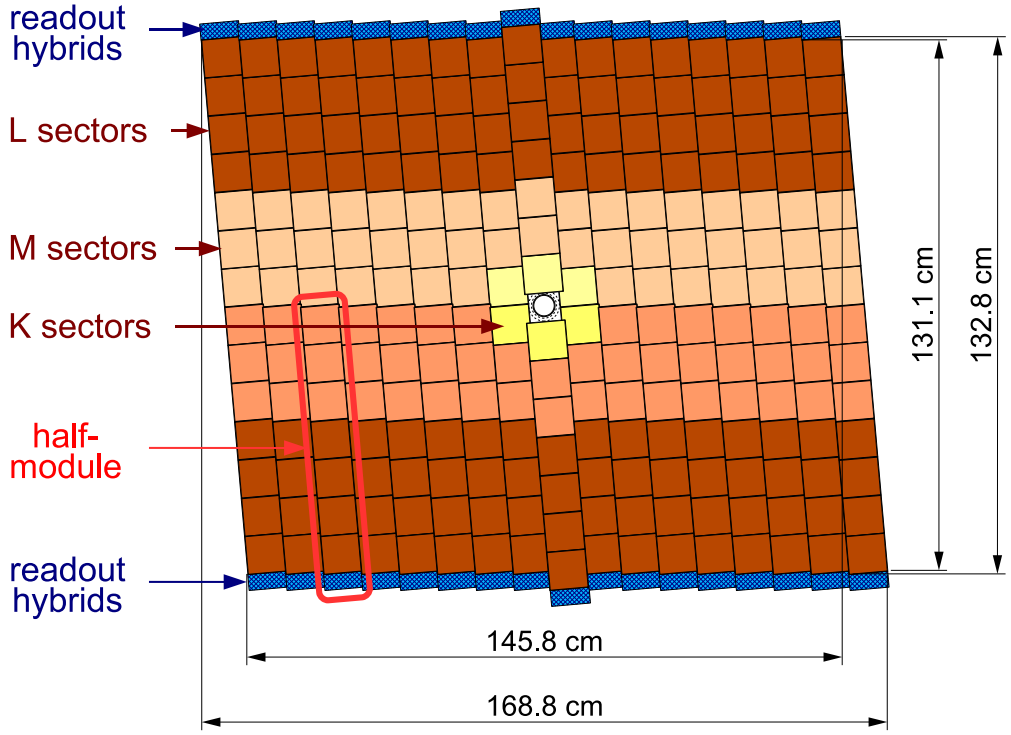


Fig. 3.5

A depiction of the third plane, in increasing z, of the TT tracker station [58]

line called *IT* (Inner Tracker), and a surrounding drift-time straw detector called *OT* (Outer Tracker). The TT and IT trackers are known in combination as Silicon Tracker (ST), as they use a common sensor of 200 μm-pitched silicon strips [58].

All ST stations, as well as the OT, use a set of 4 detector planes, with a $\pm 5°$ stereo angle in a x-u-v-x pattern, meaning that from upstream to downstream, the planes in each station have a rotation from vertical of $0°$, $-5°$, $5°$, and $0°$ (see figure 3.5). In the TT and IT, the strip pitch of 200 μm was chosen to meet the required spatial resolution of 50 μm.

Silicon Tracker

The TT covers the whole LHCb acceptance with dimensions of approximately 150 cm horizontally and 130 cm vertically, with deviation in the inner two planes caused by the stereo angles. Each plane (as shown in figure 3.5) is made up of 15 or 17 vertical sensor modules (one at $x = 0$, and 7 or 8 at each side for the first and last two planes respectively). Modules are interleaved in z by a

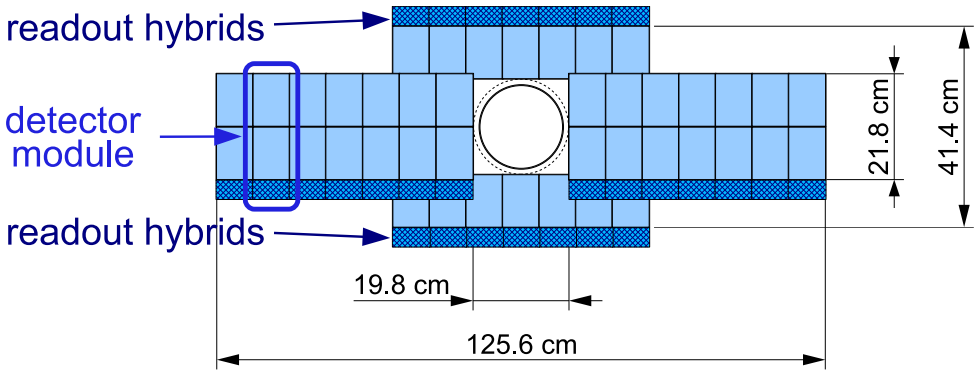


Fig. 3.6

A depiction of a non-stereo plane of an IT tracker station [58]

displacement of 1 cm and a slight overlap in x in order to cover acceptance gaps. Each module consists of 2 half-modules made of 7 silicon sensors and a readout hybrid at the end [58].

Each IT station is made up of four elements that form an overlapping cross shape around the beamline (figure 3.6) with a bounding width and height of 1256 mm and 414 mm respectively (excluding readouts).

Outer Tracker

The OT is an aluminium-mounted drift-time straw tracker with the same stereo angle properties as the silicon trackers. Straws with $\varnothing = 4.9\,\text{mm}$ and a $70\%\,\text{Ar} : 30\%\,CO_2$ gas give a drift time of $< 50\,\text{ns}$. Each module contains two equivalently staggered layers of straws to cover gaps in acceptance (figure 3.7). The distance of closest approach of a charged particle to the centre of each straw is determined from the difference in the drift time between the cascading electrons and ions depositing charge on the inner wire and outer wall respectively.

3.2.5 | Calorimeters

The calorimeters lie downstream of the second RICH detector and provide information about the position and energy of tracks. All calorimeters in the detector provide scintillating light to a photomultiplier tube (PMT) via a fibre. An electromagnetic calorimeter (ECAL) is followed by a hadronic calorimeter (HCAL). Both sets of calorimeters are segmented into sections with grids of equally spaced square cells of varying sizes. The ECAL is split into three grids

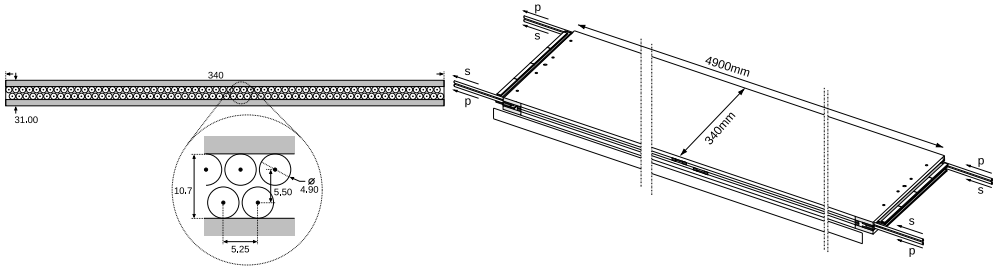


Fig. 3.7

A diagram of a OT module [58]

of increasing cell size (travelling outwards in r), whereas the HCAL is split into two grids (with a larger cell size and PMT gain than the ECAL due to a 30× lower scintillation yield).

The ECAL is preceded by a scintillator pad detector (SPD) and pre-shower (PS). These are separated by a 15 mm thick (2.5 X_0*) lead converter [58].

3.2.6 | Muon System

The muon system comprises of five stations, one between the RICH 2 station and calorimeters, and the others at the back stage of the detectors. Stations 2−5 are separated by iron blocks 80 cm thick to absorb and filter muons between stations, giving a minimum required energy of 6 GeV for a muon to leave the detector via the acceptance region. Stations $1-3$ are used to determine a muon's p_T to within a resolution of 20% for the L0 trigger (below), whereas the final two stations are predominantly used to identify muon tracks [58].

3.2.7 | Trigger and Readout

The LHCb trigger system provides a decision on whether to preserve the data from an event for offline analysis.

In run 2, the detector utilises an ASIC-based hardware trigger, called *L0* (Level 0) to perform a rudimentary decision based on relatively simple, unprocessed data from hardware readouts, such as occupation numbers in the calorimeters and muon system. This decision is made for every event, for a rate of 40 MHz. The L0 trigger reduces the rate to 1.1 MHz, which is then passed to the *HLT* (High-Level Trigger). This is a two-stage, software-based trigger, made up of

*X_0, or radiation length, is the mean length through a material to reduce a particle's energy by a factor of $\frac{1}{e}$, explained further in [73]

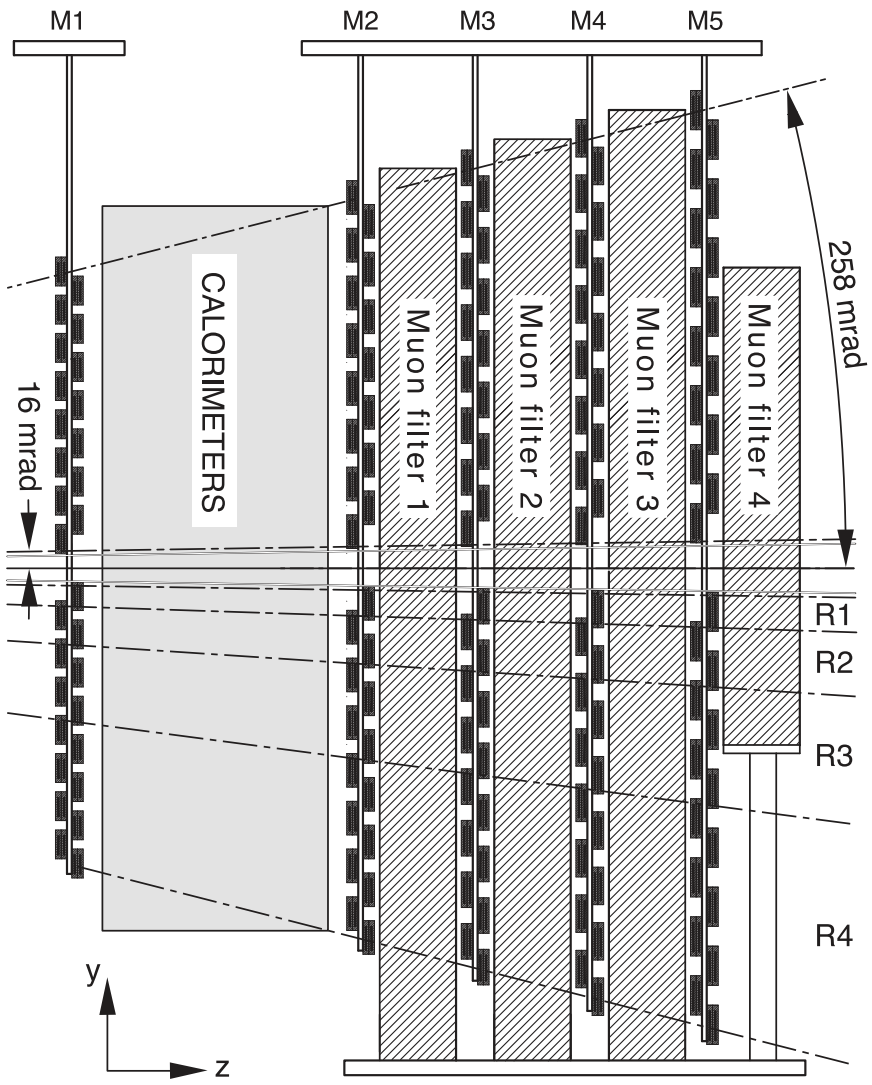


Fig. 3.8

View of the LHCb muon system and its inner and outer acceptance regions [58]

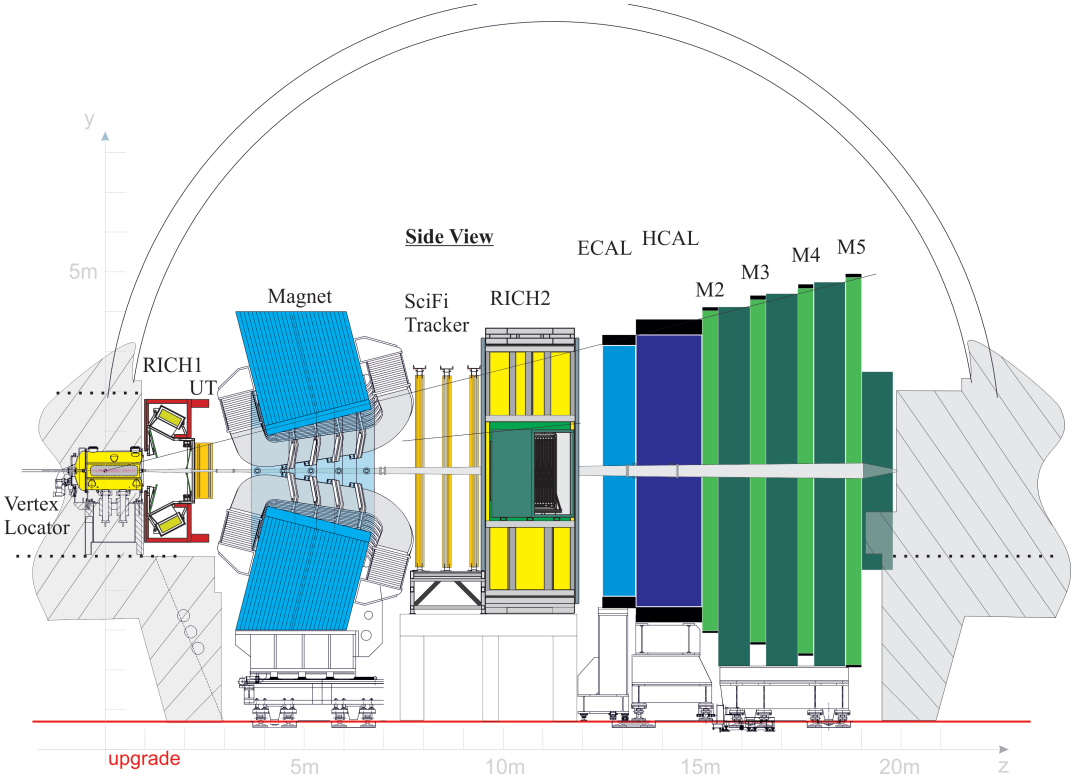


Fig. 3.9

Cross section of the upgrade LHCb detector for run 3 of the LHC [75]

HLT1, which partially reconstructs the data and performs a small number of selections. HLT2 carries out a larger range of inclusive and exclusive selections [74].

Chapter 4 discusses the run 2 trigger system and its upgrade for run 3 in further detail.

3.3 | LHCb Upgrade

In run 3, LHCb will operate with an increased interaction cross section corresponding to an instantaneous luminosity of $2 \times 10^{33}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [74], up from $2 \times 10^{32}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ in run 2 [58]. The upgraded detector is expected to take a total dataset of $50\,\mathrm{fb}^{-1}$ [76].

Due to the above luminosity increase, changing physics requirements, and technological advancements since the previous design, upgrades will be made across many components of the detector.

3.3.1 | Triggerless readout system

In addition to a higher luminosity, the LHCb upgrade forgoes a hardware trigger, and performs all decisions using a full-software trigger. This presents the challenge of increasing the incoming rate of events to the software trigger from 1.1 MHz to the full inelastic collision rate of 30 MHz [74].

3.3.2 | VELO

The upgrade of the VELO from silicon strip detectors to pixel detectors is discussed in chapter 5.

3.3.3 | RICH

While maintaining the same basic design, upgrades have been made to the RICH detectors to accommodate the increased luminosity during run 3. The run 2 HPDs with 1 MHz internal readout systems will be replaced by commercially available photomultiplier tubes with external readouts. The aerogel tile in RICH 1 will be removed, as simulations have shown that with a yield of 5.5 photons per track on average [58], it will not be useful for PID with the higher background resulting from the higher luminosity, and would only act as a decrease in effective efficiency due to multiple scattering [77]. Removing the tile was found to remove $0.035X_0$ [78].

3.3.4 | UT

The *UT*, or Upstream Tracker, replaces the TT as the tracking system upstream of the magnet, and an important detector element in the reconstruction and momentum resolution of many kinds of tracks, particularly those of long-lived particles that are likely to decay after leaving the VELO.

Like the TT, the UT contains four silicon strip detector planes, with the same x-u-v-x configuration and $\pm 5^\circ$ stereo angle. Again, the silicon strips are mounted in tiles on thin, vertical modules, however this time each vertical module is a single piece rather than being made up of two halves. Vertical modules, or *staves*, are 10 cm wide, 1.6 m long, and 3.5 mm thick. Each stave has UT hybrids mounted alternately on its front and back side. A UT hybrid consists of a 98.88 mm × 98.88 mm square silicon strip sensor and a set of 5 mm × 10 mm × 0.12 mm readout ASICs attached to a hybrid flex. The UT is 315 mm in width

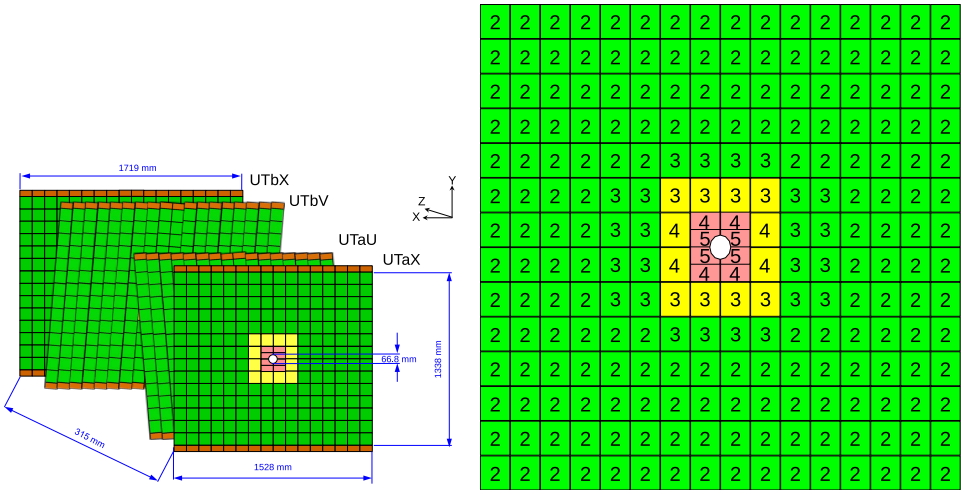


Fig. 3.10

Diagram of the four planes of the UT subdetector (left) and the number of elinks per ASIC (right). Detector planes are separated into hybrids of low (green), medium (yellow), and high (red) incidence [79].

between the front and back detector planes [79].

Figure 3.10 shows the layout of hybrids on a UT detector plane. Most hybrids (green) away from the beam line use 4 ASIC chips and 512 silicon strips with a pitch of 190 μm and length 97.28 mm. Hybrids closer to the beam (yellow) have half the strip pitch (85 μm, giving 1024 strips), and the same length. The closest hybrids to the beam (pink) have the halved pitch, as well as halved length, thus being split into two vertically and having 8 readout ASICs overall rather than 4. For this reason, all staves have 14 hybrids, except for the middle two, which have 16.

Each individual ASIC is also connected to 5 serialiser e-links. Between 2 and 5 of these are enabled, depending on the proximity to the beamline (figure 3.10) [79].

3.3.5 | SciFi Tracker

The *Sci-Fi Tracker*, or Scintillating Fibre Tracker (also referred to as simply fibre tracker or FT), is the successor to the T1, T2, and T3 tracking stations of the LHCb detector, and is designed to provide precise momentum resolution by measuring the bending angle of tracks after passing through the magnet. Like previously, SciFi consists of three stations with four detector planes each. However, now each SciFi station is a single scintillating fibre tracker, rather

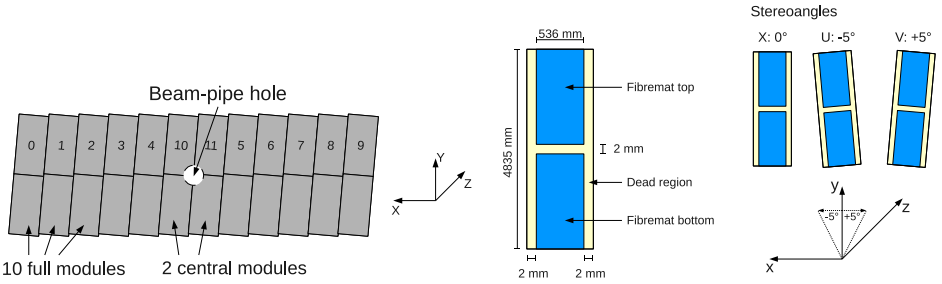


Fig. 3.11

View of the 12 modules that make up a single plane of a SciFi tracker station, at a $+5°$ stereo angle [79].

than the previous combination of a silicon strip detector (IT) for high η, and a drift-time straw tracker for low η. Scintillating fibres are fed into silicon photomultipliers (SiPMs) that can be read out at the full 40 MHz bunch crossing frequency. The physics requirements are for a 99% hit detection efficiency and a spatial hit resolution better than 100 μm in the bending plane [79].

Detector planes feature the stereo angle configuration of the previous TT stations, but consist of rows of 12 modules, the middle 2 of which have a hole cut for the beam to pass through (figure 3.11). Modules are 540 mm wide and 4835 mm tall, with a coverage by the fibre of 99.2% [79].

3.3.6 | Calorimeters

In order to perform full readout at 40 MHz, the readout and control electronics of the calorimeter system have been redesigned. The calorimeters themselves are largely unchanged, with the modification of a reduction in the PMT gain in order to protect the hardware from the higher luminosity over the length of run 3. Calorimeter cells are not expected to require replacing under the third long shutdown (LS3) after run 3 [79].

During run 2, the main purpose of the SPD/PS was as an input to the L0 trigger. Since the detector is now largely zero-suppressed and uses a full software trigger with a very scaled back hardware low-level trigger (LLT), these components have been deemed unnecessary and have been removed. The SPD was used in the trigger by rejecting busy events based on the SPD hit multiplicity. As this is now not an option, software trigger lines may use data from the multiplicity in the ECAL and HCAL [79].

3.3.7 | Muon System

As an integral part of particle identification, the muon system must work in tandem with the upgrade trigger to provide an acceptable identification efficiency at the increased luminosity, at the full 40 MHz bunch crossing rate. For this, the readout system of the muon detectors has been redesigned, and is in line with the front-end electronics of other LHCb subdetectors in using readout components based on the GBT (GigaBit Transceiver) technology. The full software trigger has lessened the requirement of the spatial resolution, and so the first muon station (prior to the calorimeters) is removed for the upgrade. The energy resolution is also improved via the removal of the SPD/PS [78].

3.4 | Computing

This section will briefly describe aspects of the LHCb data flow and software system that are used in other chapters.

3.4.1 | Run 3 LHCb Data Flow

The LHCb data flow is a modular pipeline in which, at each stage, events may be saved to disk in a specific file format before being processed by the next stage. This modularity means that certain data processing stages may be upgraded or swapped out without having to generate the input data from scratch. Naturally, this pipeline differs according to the type of data that is being processed; namely, if the data is simulated or real.

Simulated data are handled inside the LHCb simulation framework Gauss [80], and may be created simply via an event generator (EvtGen [81]), in which case only the 4 vectors of the decay particles are created. Events can be generated and forced to travel through a particular decay mode, or generated with a random sampling of the appropriate measured branching for all decays (known as unbiased, or minimum-bias). Generated events have the option of a set of basic generator-level cuts, such as ensuring all daughters of the event in question intersected with the angular LHCb acceptance region.

Alternatively, Monte Carlo events can involve the full simulation of all or some sections of the detector. Here, the particles are transported through the geometry of the detector with the Geant4 [82–84] package, which is much more computationally expensive than event generation. After the events have been

simulated, the simulated detector is digitised by BOOLE [85], meaning that the frontend electronics are simulated, and their outputs are stored in detector-specific "raw banks".

From here, both real and simulated detector outputs are run through the software trigger, in the MOORE framework. In run 3, the ALLEN [86] application will run inside Moore for the HLT1 stage (partial reconstruction and generic selections), allowing the entire HLT1 sequence to run in parallel on GPUs. The HLT2 stage (full reconstruction) is then run on CPU, and the output saved to disk for offline processing and analysis runs. In run 2, the HLT2 stage only performed a partial reconstruction, and the full reconstruction was performed by the offline *Brunel* package, however this role will be subsumed by HLT2, and be run fully online. Online processing happens in a dedicated on-site computing centre called the Event Filter Farm.

3.4.2 | Detector Description and Conditions

In order to simulate the LHCb detector, both in Monte Carlo particle simulation and digitisation and reconstruction of real events, the physical state of the detector must be described. The state of the detector is encoded separately to the logic of digitisation and reconstruction as various aspects change over the course of both commissioning and construction, and normal operations. The state of the LHCb detector is described by two sets of data: The physical detector itself is defined by the detector description, which encodes the materials and geometry of all components in each subdetector. This description is summarised as a tag in the git-based LHCb *Detector Description Database*, or DDDB, such as `dddb-20210617`.

For a given physical detector setup, there are still many aspects that will change over the course of its operation. These are known as the detector conditions, and summarised by a tag in the *Conditions Database*, or CONDDB, such as `sim-20210617-vc-mu100`. Examples of changeable detector conditions are the alignment of the movable detector components, and the polarity of the magnet.

Chapter FOUR

High-Level Trigger

4.0.1 The LHCb Trigger

In runs 1 and 2, the LHCb detector has used a trigger configuration consisting of a hierarchical combination of hardware and software triggers. All events are processed by a preliminary hardware trigger, referred to as the Level 0 trigger or L0, designed to rate limit the events passed to software. Events that pass L0 reach the first stage of the High-Level Trigger (HLT1), which performs a mostly inclusive, topological selection. Likewise, those that pass HLT1 are processed by the more exclusive, computationally extensive HLT2, looking for specific decays with a more thorough reconstruction. Events that pass the full trigger are then stored offline for full reconstruction analysis. This cascaded style of trigger helps to reduce overall computation for the full set of events.

The reduction in event rate by the run 2 trigger is as follows [87]:

$$\text{Bunch crossing rate: } 40\text{MHz} \xrightarrow{\text{L0}} \leq 1\text{MHz} \xrightarrow{\text{HLT}} \leq 12.5\text{kHz Offline storage}$$

The L0 trigger is a hardware trigger consisting of separate triggers on the calorimeters, the muon detectors, and the VELO pile-up*, whose outputs are

*mean pp interactions over visible events

computed by a decision unit (with a latency of 4μs between a pp interaction and its decision) and fed out through the front-end electronics. The L0 trigger readout supervisor controls the output rate via timing signals distributed through a control board to meet the maximum throughput of the front-end detector boards, and the LHCb Event Filter Farm (EFF) running the HLT. When an event passes the L0 decision unit (DU), the full detector data is read out of the front-end electronics, digitised, and transmitted to the event-builder, where it is aggregated and sent to the EFF to be processed by the HLT [58].

The HLT1 performs a partial reconstruction of an event as part of its decision sequence. A pattern recognition function creates tracks in the VELO from the set of hits. These tracks are then extrapolated to trackers further along the beam in the detector (TT, T1-T3), and a rudimentary χ^2 fit is performed. The hits in the trackers are then more accurately fitted to the successfully extrapolated tracks via a Kalman filter* [88, 89]. This fit produces another χ^2 associated with the extended tracks, and optimises the tracks and hits within constraints. The fitting process also assigns a "ghost probability" to each track (the likelihood that a track is a "ghost" given the consistency between the fitted track and the tracker hits). A ghost is an unrelated set of detector hits that spuriously appear to belong to a track, either by coincidence, or an artefact of the detector.

The decision of HLT1 is determined by the Boolean **OR** combination of a number of subroutines known as *trigger lines*, which are listed in full in appendix A.2. A given trigger configuration may activate different lines that are dependent on the type of decays and phenomena being searched for, particularly when running the trigger on Monte Carlo simulated events. The two trigger lines designed to capture generally "interesting" events are the multivariate lines HLT1TrackMVA and HLT1TwoTrackMVA, which select for the characteristics of B decays that the LHCb detector is suited to observe, and that motivate further analysis (see section 2.4.1. The single-track MVA performs a series of cuts on the track variables that amount to a Boolean expression (table A.3). The two-track MVA performs a set of preliminary selections on the set of tracks, and then performs vertex fit on each combination of 2 tracks. Selection cuts and a BDT on the variables of the combinations are then performed (table A.4). The two lines take as input the default set of charged particle candidates that

*Unlike the more comprehensive Kalman filter fit during full offline analysis, the HLT fit uses fewer passes and no outlier rejection.

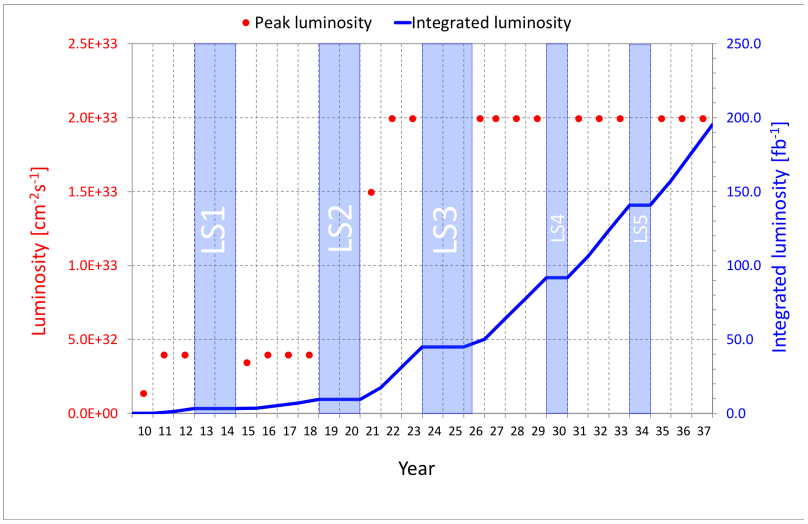


Fig. 4.1

Projected instantaneous and integrated luminosities of LHCb over the next runs of the LHC, from [91].

have been minimally reconstructed by the HLT1 and processed by the Kalman fit, with a loose global event cut (a cut based on the complexity of an event to optimise computing resources).

4.0.2 | LHCb Trigger Upgrade For Run 3

The management of the LHC incorporates multiple Long Shutdown phases coordinated across all collaborations (figure 4.1), during which time the beam is powered down and project upgrades can be made. The second such shutdown is began in 2018, and data-taking is expected to resume for Run 3 in 2022.

As LHCb has continued to take data, many of the more simple and common, "low-hanging fruit" particles and decay channels have been measured, and the project has turned its focus to measuring rarer, more specific decays. The original trigger TDR for LHCb [90] profiles software trigger efficiences for mostly 2- or 4-track decays such as $B_d^0 \to \pi^+\pi^-$ or $B_d^0 \to D_s^-(K^+K^-\pi^-)\pi^+$, whereas the trigger upgrade TDR [74] from 2014 lists full-trigger efficiencies for more exotic, low branching ratio decay channels, including penguins and semi-leptonic decays. As luminosity increases, computing constraints force trigger selections to be more exclusive.

The LHCb detector was initially designed to operate at an artificially low in-

stantaneous luminosity* of $2\times10^{32}\text{cm}^{-2}\text{s}^{-1}$, citing improved radiation resistance and simpler analysis from lower pile-up [58]. Due to the need for larger datasets for higher-precision measurements and the analysis of rarer decay channels, for Run 3 of the LHC, the LHCb detector is expected to operate at an instantaneous luminosity of $2 \times 10^{33}\,\text{cm}^{-2}\,\text{s}^{-1}$ [74], an increase of a factor of 5 over the previous Run 2 luminosity of $4\times10^{32}\,\text{cm}^{-2}\,\text{s}^{-1}$ [94], and corresponding to a pile-up of 7.6. This increase in luminosity means an increase in the average number of tracks per event that must be processed and selected in the first stages of the software trigger (with some trigger processes scaling super-linearly).

With the aim of maximising trigger efficiency, LHCb intends to move to a fully software-based trigger, as software allows for more complex trigger logic to enable higher efficiencies, and is flexible against changing future physics motivations and technological obsolescence [74]. This means that, rather than being exposed to the 1 MHz output of a hardware trigger, the first stage of the run 3 software trigger will ingest data at the full 30 MHz inelastic collision rate, as shown in figure 4.2. Current estimates for the computational capacity of the Event Filter Farm by the beginning of Run 3 allow a per-core maximum average computation time for a fully software-based trigger of around 13ms per event to process the visible bunch crossing rate of 30MHz [74]. (as opposed to the 20ms of the current HLT at the lower luminosity).

One particularly costly process in the early trigger stages in the Kalman fit that is applied to all tracks (except for a relatively minor cut in the global event cut, as well as a cut on soft tracks close to the beam line). This fit, which optimises the accuracy of a track's state variables at the interaction point (x, y, Tx, Ty, q/p) takes on average 0.54 ms/track on its own (Appendix A.11), too long to apply to the majority of tracks after the upgrade. Currently, the HLT1 MVA lines which perform an inclusive-b selection use these Kalman variables.

4.1 | Data Analysis Methods And Machine Learning

Machine learning describes a wide range of techniques to create an approximation of a model, without the model's parameters being determined manually. This is often done via an algorithm that seeks to minimise a cost function which

*compared to the ATLAS and CMS detectors which run at the LHC's maximum available luminosity of $10^{34}\text{cm}^{-2}\text{s}^{-1}$ [92, 93]

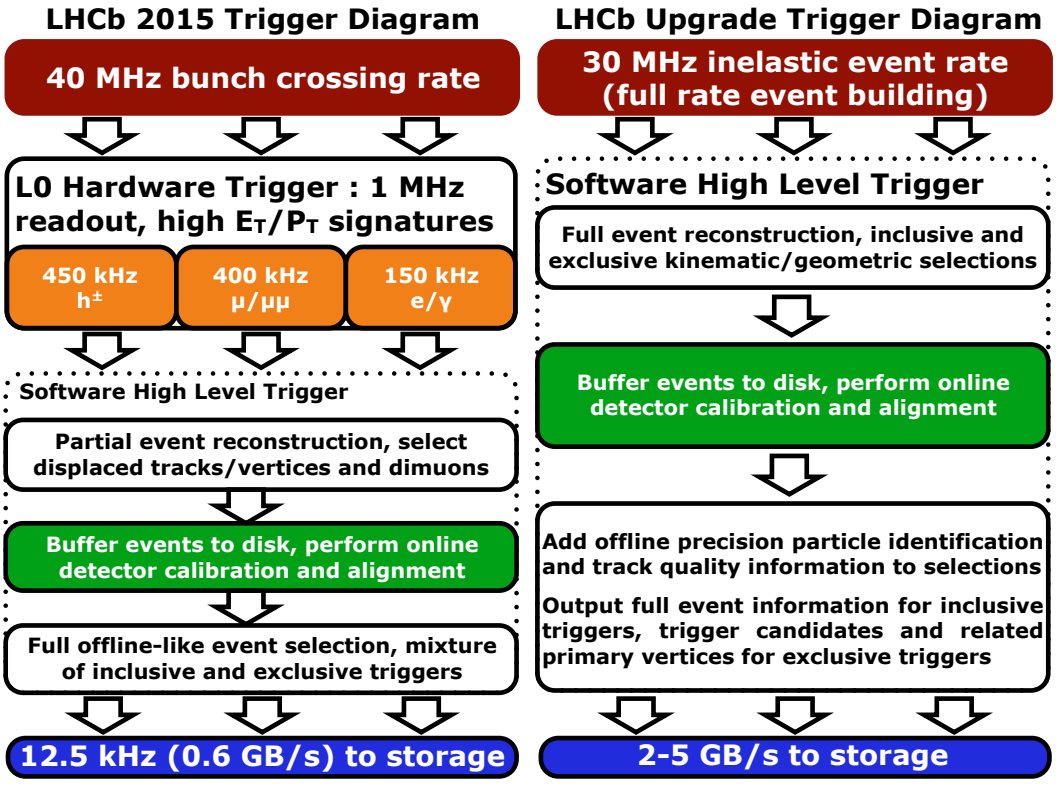


Fig. 4.2

Run 2 (left) and run 3 (right) data flow hierarchies of the LHCb trigger [95]

evaluates the deviation of the predictive model from empirical data. Different techniques perform differently with respect to computational cost, accuracy, generalisation, transparency and other factors. Equally important is the type and quality of the input data. Machine learning methods are categorised as *supervised* if the training data includes class labels or "truth" values that the model targets, or *unsupervised* if not. In this sense, a classifier of particle and track types that is tuned based on simulated Monte Carlo data is an example of a supervised machine learning model. An exploration of the various types of classification, regression, and machine learning methods can be found in appendix A.3.

4.2 | Analysis of a Neural Network as an Inclusive-b MVA

Unlike a decision tree or set of cuts, a neural network as a classifier outputs a value that, when normalised to the range $0 \rightarrow 1$, corresponds to the confidence of the classifier that the set of inputs belongs to a particular class (in this case, that a track or event contains a b particle). Provided there is a monotonic relationship between this output and the likelihood of a track containing a b particle, it may provide a natural parameter with which to tune the trigger acceptance rate at runtime.

4.3 | Software Used

4.3.1 | Gaudi framework

The software of LHCb is run inside the Gaudi framework [96] - a HEP framework designed to sequentially process HEP events. Monte Carlo simulation of particles in the detector is important for establishing "truth" data for classifiers and other analysis techniques to compare real data to. Particle simulation in the detector is managed by the program Gauss, which controls the Monte Carlo generation of particles (in Pythia), their decay (EvtGen), and their propagation through the space of the detector (Geant4). The HLT is run inside the Gaudi framework with the Moore application, which contains the relevant algorithms and options necessary to run the HLT lines at a specified TCK*.

*Trigger Configuration Key; a 32-bit ID of a particular configuration of L0 and HLT

As outlined in ref [97], in Gaudi, events are processed in an *event loop*, which contains a tree of *algorithms*, C++ functions that perform particular high-level operations on the data. Algorithms can have options that are set in *options*, python files that are evaluated by Gaudi at runtime. Gaudi also uses *tools*, shared functions that can be called by any algorithm. For lower-level controls often used by the framework itself, Gaudi has *services*, which handle tasks like writing to the disk.

With the decline in growth rate of sequential computing power, a higher importance is being placed on the ability to perform computation in parallel. HEP belongs to a class of problems referred to as "embarrassingly parallel", in that the entirety of the task involves little to no inherently sequential component, so the job can be split up into an arbitrary number of parallel subjobs (with the only limit being that each individual event is logically sequential). Ganga is a computing job management tool [98] that can split a job running on the Gaudi framework into a number of subjobs that run on the Worldwide LHC Computing Grid [99, 100].

When running the HLT on Moore, by default each trigger line is timed and recorded, with the aggregated times written at the end of the run. The Gaudi framework normalises its timing blocks based on the per-clock performance of a 2.8GHz Xeon CPU. The CPU running Gaudi will run a benchmark to find the performance ratio to this reference CPU.

4.3.2 | Machine Learning Library

Appendix A.6 provides an overview of some of the more commonly used various machine learning libraries. For the training of models, the PyTorch framework [101] was used. Section 4.10 explores the various options for deploying the model in production at the Event Filter Farm.

4.4 | Run 2 Datasets and Software

For training data, this work used an inclusive-B Monte Carlo dataset from 2016, of roughly equal parts Magup and Magdown, (details in table 4.1). Although this data was generated with at least one B particle in each event within $(400\,\text{mrad})^2$, only 70.7% of events contained at least one track from a B particle within the $\pm 250\,\text{mrad}$ detector non-bending acceptance region [58], with 11.3% of all reconstructable tracks having a b ancestor.

Type	Inclusive-B
Year	2016
DDDB	dddb-20150724
CONDDB	sim-20161124-2-vc-md100, sim-20161124-2-vc-mu100
Number of events	60 000 non-triggered
Number of tracks	900 000

Table 4.1

Details of Monte Carlo data used in this work. Inclusive-B means that the event generator forces a b quark in the event inside the wider 400 mrad acceptance region but permits any decay. For DDDB and CONDDB see section 3.4.2.

Events were processed in the trigger simulation framework Moore (v26r5). When evaluating the classification power of the existing classifiers (Hlt1TrackMVA and Hlt1TwoTrackMVA in the "tight" and "loose" configurations) copies of the trigger lines were made with and without the L0 emulation filter sequence applied. A custom trigger line was written to run the entire set of "forward" tracks in each event through the Kalman fit process, meaning that the neural networks, whether trained on the forward-track variables or the more precise Kalman-fit variables, were trained on the same set of tracks.

When assigning truth data to tracks, a track that contained a b quark in any particle in its ancestry would be deemed signal. Exceptions for this are tracks that contained a relatively long-lived unstable b-daughter particle* in its ancestry, which were deemed background under the criterion of containing a b quark. The reasoning for this is that such long-lived daughters would dominate the impact parameters† of these tracks, making the b-parent tracks indistinguishable from other parent particles, and as such they should not be considered in the efficiency.

Classifier training and evaluation was primarily performed outside of Gaudi and Moore. Per-track and per-event variables were written to separate CSV files via a custom monitoring trigger line, and read into a dataframe in Python

*$\{K_S^0, \Lambda^0, \Sigma^+, \Sigma^-, \Xi^0, \Xi^-\}$

†the impact parameter of a track is the minimum distance it would come from a given primary vertex if the straight line were extended arbitrarily far in each direction (ie, without starting at some origin vertex).

to train a PyTorch model.

The classification performance of all classifiers use metrics of signal efficiency ($\frac{\text{TP}}{\text{TP+FN}}$) and background rejection ($\frac{\text{TN}}{\text{TN+FP}}$)*, as these metrics are agnostic to the ratio of real signal to background in the dataset.

4.5 | Model Configuration

4.5.1 | Choice of neural network architecture

As previously mentioned, a neural network can produce a confidence value as its output, which can be cut on at any value to provide a decision, allowing the acceptance rate of the trigger to be smoothly tuned at runtime, without needing to retrain the model. A preliminary analysis of inclusive-B data found that, for the same input variables and background rejection rates, BDTs and neural networks achieved comparable maximum signal efficiency rates. Neural networks may be trivially trained and implemented on parallel hardware such as GPUs given the software available (see section 4.10).

When performing a decision on a single detector track, the fixed, low number of relevant input variables means that a feed-forward neural network is suitable to process the data.

4.5.2 | Normalisation

Neural networks, while technically universal approximators (see appendix A.4.1), have a limited range and precision of approximation based on the input data and number of hidden nodes. An input distribution spanning many orders of magnitude (as does, for example, a track's impact parameter and its associated χ^2) will not be of use to a neural network, as the magnitude of large values combined with the precision necessary for small values is too great. One obvious approach is to take the logarithm of all variables (though care must be taken that their distributions do not cross, or come arbitrarily close to, 0). Another choice is to create an interpolation transformation of each variable that transforms its distribution into a linear one, and then convert this into a Gaussian of a sensible mean and deviation. The distribution for this latter approach is dependent on the batch of data used to create the interpolation

*where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

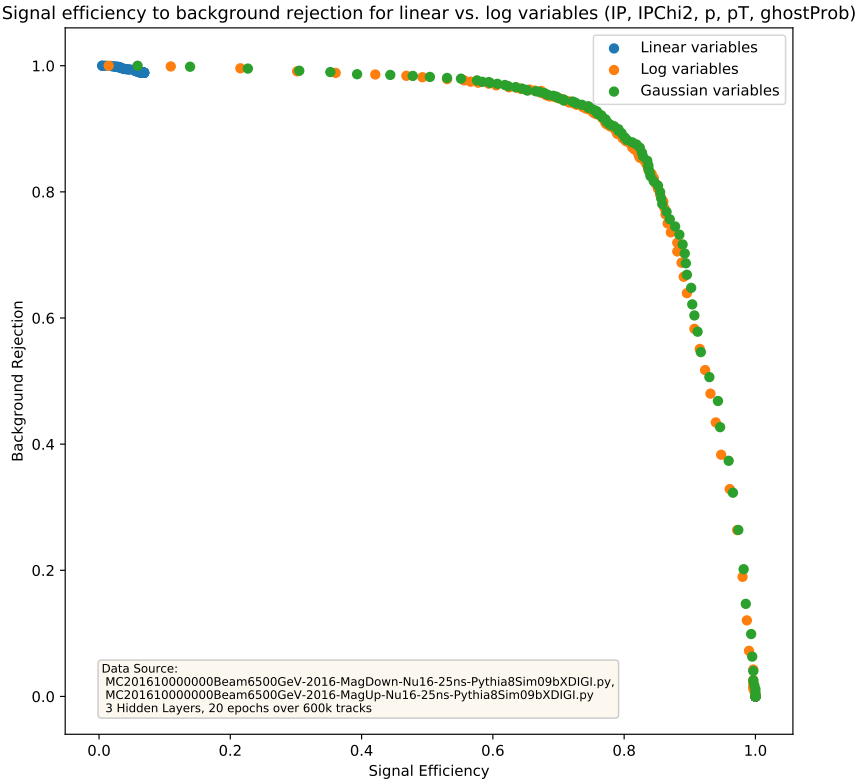


Fig. 4.3

Efficiency to background rejection plot for no normalisation, logarithmic normalisation and Gaussian normalisation.

function, which cannot be changed later without retraining the network from scratch. As shown in figure 4.3, using non-normalised variables for this problem will produce a completely useless classifier, with this particular network rejecting nearly all events, placing the whole curve in the top left, with almost total rejected background but no retained signal. Using logarithmic and Gaussian normalisations produced very similar results.

4.5.3 | Preparing Input Data For Training

Inspecting the distributions of input data by target class can give insights into the regions of variable space that contain the most discriminative information, and how a properly-trained classifier will likely treat an event based on the distribution of a single one of its variables. If the classes are particularly separated in a single variable, it may be obvious that only a single one-dimensional cut is sufficient.

The truth label for particle type is determined in the following way: HLTCSV-

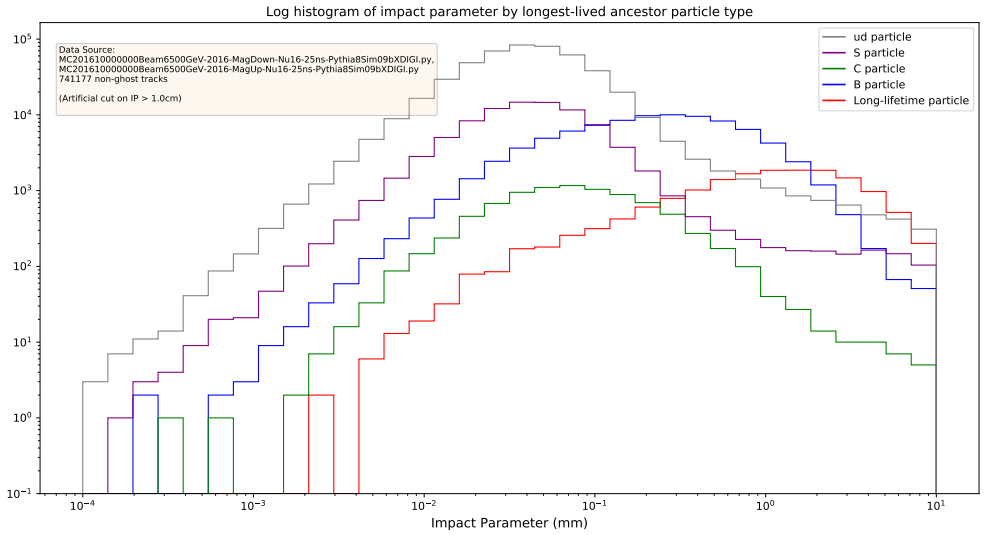


Fig. 4.4

Histogram of the number of reconstructed, Kalman-fitted charged tracks within 60 000 inclusive-B Monte Carlo events, with respect to their impact parameter after the fit. The tracks are separated by the longest-lived particle in their true simulated ancestry. These distributions have an artificial cut at 10mm imposed by the trigger. b-ancestor tracks are the largest contribution in the region around $0.5 - 2$mm.

Monitor.cpp ranks particle types in increasing order of mean lifetime,

$$\text{Rank } R_{particle} : R_{long} > R_b > R_c > R_s > R_{u,d}$$

where R_{long} is the rank of some (relatively) long-lived particle like K_s^0. For each track, the algorithm recursively finds the track's parent particle. The particle that represents a track is the highest-ranking particle type that appeared in the track's history back to the primary vertex. The reason for this ranking system is that a final-state particle that is directly reconstructed in the detector will have properties (such as impact parameter) resembling its longest-lived ancestor. This also means that b-containing ancestor particles (detection of which are the primary purpose of the trigger at LHCb) are represented in the classification by the trigger.

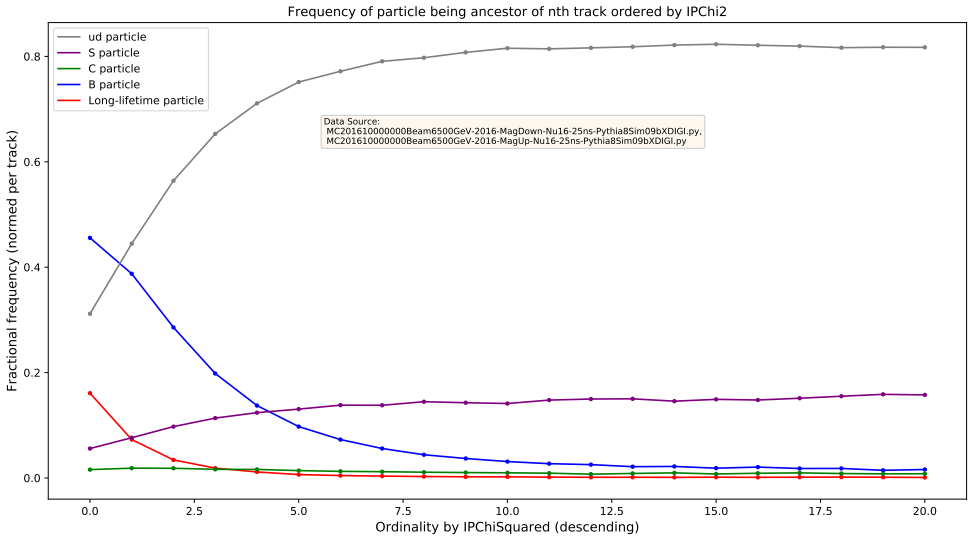


Fig. 4.5

Normalised frequency of the maximum ancestor rank (as defined in section 4.5.3) of the n^{th} track in the events, in descending order of χ^2_{IP} (the lowest χ^2 of the fit of the impact parameter to the set of primary vertices). In this Monte Carlo dataset, the highest χ^2_{IP} track in a given event is most likely to have a B particle as its longest-lived particle ancestor type.

4.5.4 | Imbalanced class data

When training a classifier to target multiple classes, it is important to consider the balance of these classes in the training data. When using real detector data, altering the training variables has little impact on the accuracy of the network. In this case fewer than 10% of events meet the criterion to count as a true positive, meaning that for a given training batch, roughly 90% of the contribution to the loss function would be through fail events. This has the effect of disproportionately biasing the network to return "fail" on event inputs (in fact, practically all events were classified this way in a proof of concept). However, due to the fact that a vast majority of events really are "fail" events, the network successfully classified 93% of events, despite obviously being completely useless as a trigger and identical in practice to "return 0;".

This highlights the importance of using an appropriate metric to quantify the utility of a classifier. The four fundamental quantities are, given a number of total events, true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Here the positives and negatives are synonymous

with signal and background respectively. These are often presented in a table known as a confusion matrix, and can be combined to give other useful values. Commonly used metrics in machine learning are:

$$\text{Precision} \equiv \frac{TP}{TP + FP}$$

$$\text{Recall} \equiv \frac{TP}{TP + FN}$$

which are often referred to as purity and efficiency respectively in HEP. Another common metric is

$$\text{Specificity} \equiv \frac{TN}{TN + FP}$$

also known as True Negative Rate, and, in HEP, Background Rejection.

Any classifier which acts as a cut on a regression model has a degree of freedom as to where the cut is taken, which can be visualised with a curve between two independent classifier performance quantities, such as the ROC curve (Receiver operating characteristic), a plot of true positive rate against false positive rate. A useful single metric is the area under this curve (AUC). Another commonly used curve in HEP is that of background rejection to signal efficiency. A straight line from $(0, 1)$ to $(1, 0)$ corresponds to a randomly guessing classifier. If the classifier performs better than random guessing, the curve will be above this line, incurring a trade-off between signal retention and background rejection that must be evaluated according to the needs of the project. An advantage of a classifier using this confidence-based cut (over, for instance, a combination of hard variable cuts determined by the output of a BDT) is that is provides a trivial but meaningful method to rate limit the bandwidth of pass events, which is crucial behaviour for a trigger.

There are two common methods for dealing with an imbalanced training dataset. The first is to dynamically filter input data so that the classifier trains on an even split of events by class. The second is to multiply the value of the loss function by a coefficient inversely proportional to each class' fractional representation in the dataset. The former approach was taken in this study, with training data being divided into two rolling buffers that were separately cycled through over the training epochs.

4.6 | Hyperparameter Optimisation

Combinations of network hyperparameters were explored for this work. As indicated in figure 4.6, there is a large increase in performance between 0 and 1 hidden layers, but less of an increase from 1 to 2 (roughly 1σ).

The performance of the network was also tested as a function of the number of nodes in its single hidden layer. For a normal feed-forward neural network such as this, the number of computations between two layers of sizes n, m grows as the $n \times m$. A width of > 5 did not increase classifier performance noticeably.

4.6.1 | Choice of Input Variables

A small neural network was trained to classify tracks with only a single input variable (figure 4.8):

	$\log_{10}$IP	$\log_{10}(\chi^2_{\text{IP}})$	$\log_{10} p$	$\log_{10} p_T$	ghostProb
Area Under Curve	0.835	0.858	0.511	0.614	0.563

Note that a random decision would give an estimated AUC of 0.5. The variable with the greatest individual discriminating power is χ^2_{IP}, with p being the weakest. The ghostProb variable (MVA likelihood that the track is a ghost track), while having a much lower total individual discriminating power than IP and χ^2_{IP}, discriminates a different domain of tracks, and thus is a valuable addition in a multivariate classifier, particularly for a high desired background rejection rate. Ghost tracks, being artefacts of the reconstruction rather than real tracks, are more likely to have arbitrary (and thus, larger) impact parameters than real tracks, and so have differing IP distributions. Figure 4.7 displays the correlation matrices of simulated data for the reconstructed variables used in this study. A correlation is found between the impact parameter and the ghost probability that is more significant than between other pairs of variables.

A Principal Component Analysis (PCA) was also applied to these variables in 5 000 tracks, with the largest two principal components contributing to 68.1% of total variance.

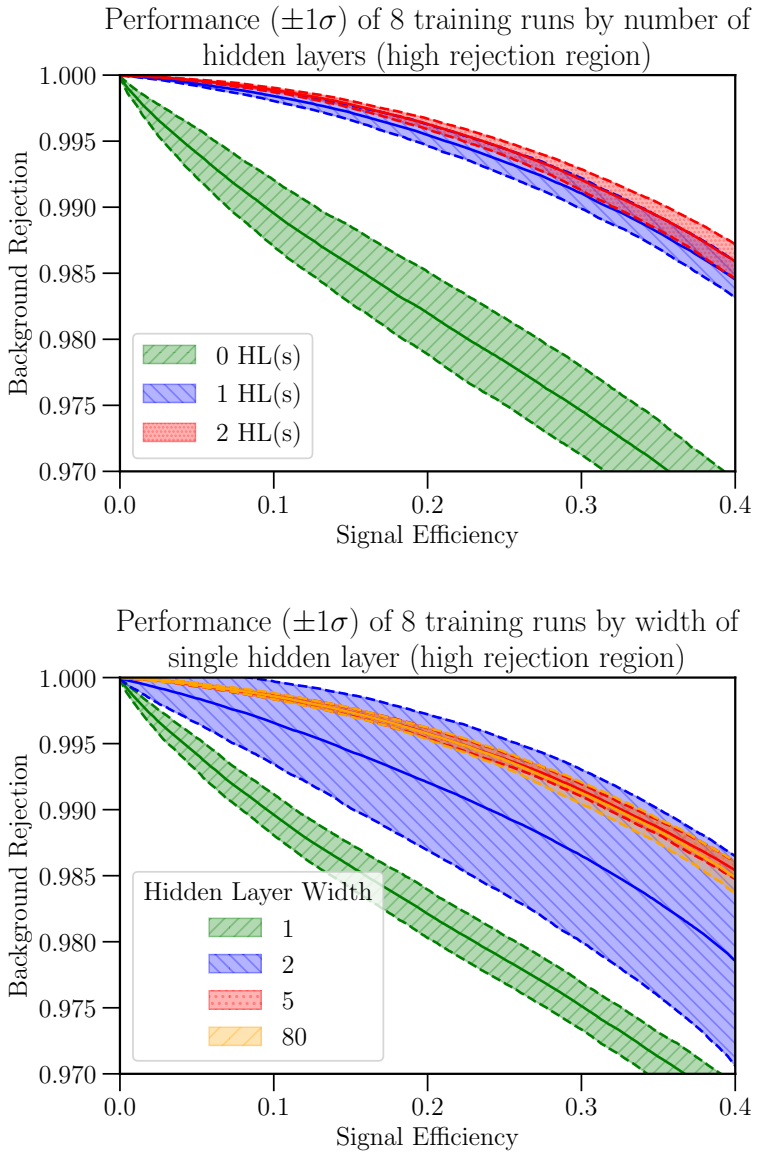


Fig. 4.6

Signal efficiency and background rejection defined in Appendix A.1. Solid lines denote the mean background rejection rate over 8 randomly seeded training runs, with the shaded area reaching above and below by the standard deviation of the set. The hidden layer width is the number of neurons contained in each hidden (intermediate) layer of the fully-connected neural network.

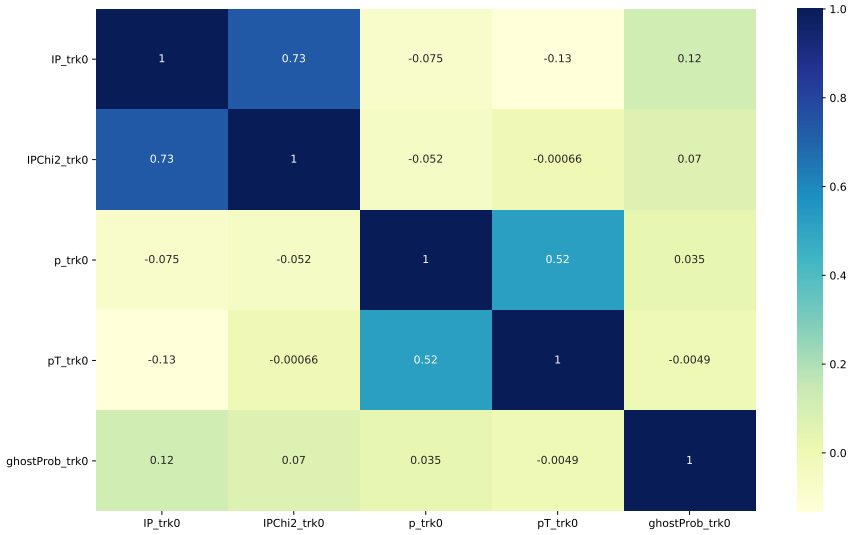


Fig. 4.7

A correlation matrix of the reconstructed momentum-based (p, p_T) and impact-parameter-based $(\mathrm{IP}, \chi^2_{\mathrm{IP}})$ variables, as well as the ghost probability, in a set of Monte Carlo simulated data (see table 4.1).

Principal Component	$\log_{10}$IP	$\log_{10}(\chi^2_{\mathrm{IP}})$	$\log_{10}$p	$\log_{10} p_T$	ghostProb	Variance Contribution
1	-0.689	-0.686	-0.079	0.011	-0.220	1.99 (39.8%)
2	-0.114	0.005	0.684	0.707	0.128	1.42 (28.3%)

4.6.2 | Low-p & p_T Filter

Tracks are harder for the NN to classify in the low p and p_T region due to having a very large number of background candidates and few identifying characteristics. A manual cut could be added here, as any b particles successfully classified here may not be of use in further analysis.

4.6.3 | Forward vs Fitted Variables

Here, *forward* variables are properties of a track that has been successfully reconstructed all the way from the VELO to the forward trackers in the detector. The *fitted* variables refer to values of these variables that have been refined to have a reduced error with the use of a Kalman filter. Figure 4.10 shows the normalised distributions of deviations between the fitted and unfitted versions

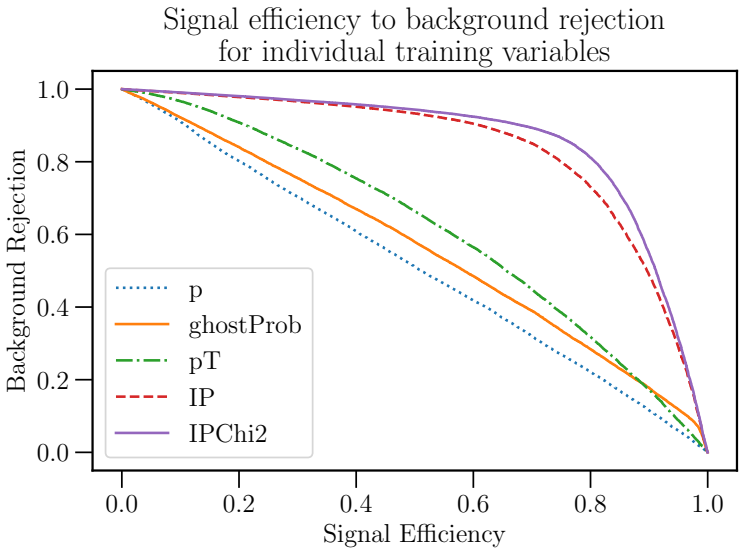


Fig. 4.8

A comparison of the classifying power of a single track variable. For convenience, these were implemented via a neural network trained on a single input, but all functions of a single variable of the same monotonicity (with range $0 \to 1$) are equivalent when generating curves based on a pass probability cut (for example, for the ghostProb variable, the network trivially learned to approximate some monotonically decreasing function, as a higher ghost probability should have a lower chance of passing the decision).

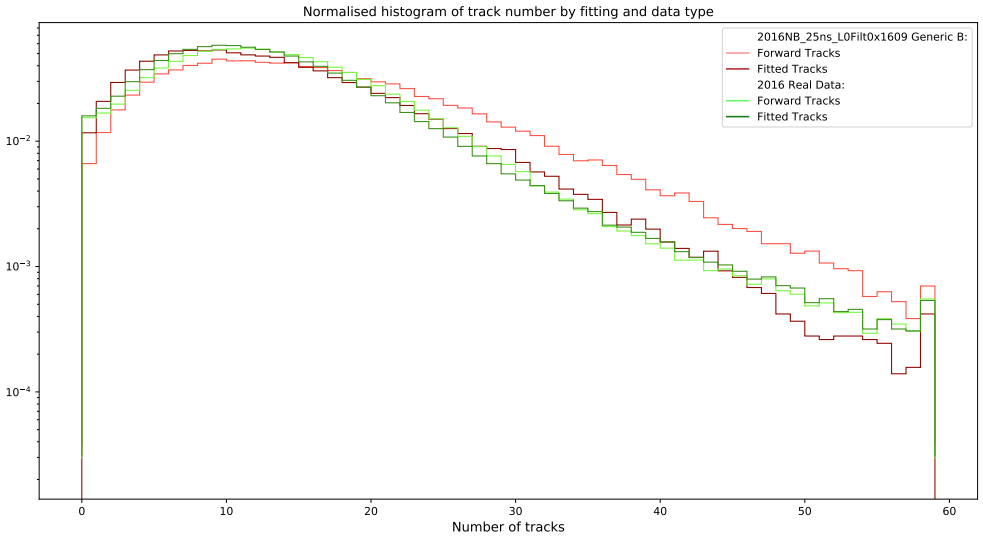


Fig. 4.9

Normalised distributions of the total number of tracks per events, separated by fitted vs forward tracks and real (green, dark green) vs Monte Carlo simulated (pink, red) data. The median number of tracks per event is between 10 and 15, with there being slightly more forward tracks than fitted, as expected.

of various input variables:

$$\frac{x_i^{forward} - x_i^{fitted}}{x_i^{fitted}}$$

separated by MC truth ancestor value. The ranges of these distributions were cut to remove the highest 10% of tracks by deviation, as there were very long tails. The p and p_T variables showed relatively small deviations before and after fitting, with 90% of tracks differing by less than 15%, whereas the IP and χ^2_{IP} showed very large deviation of up to nearly 7 orders of magnitude.

Figure 4.11 illustrates that using classification variables that have been refined with the HLT Kalman filter process improves background rejection rate for a given signal efficiency, with diminishing effects at higher targeted background rejection. Conversely, the addition of the ghost probability as an input variable provides a more consistent improvement to background rejection over the range of signal efficiencies, due to the orthogonality of the information it provides in addition to the other variables.

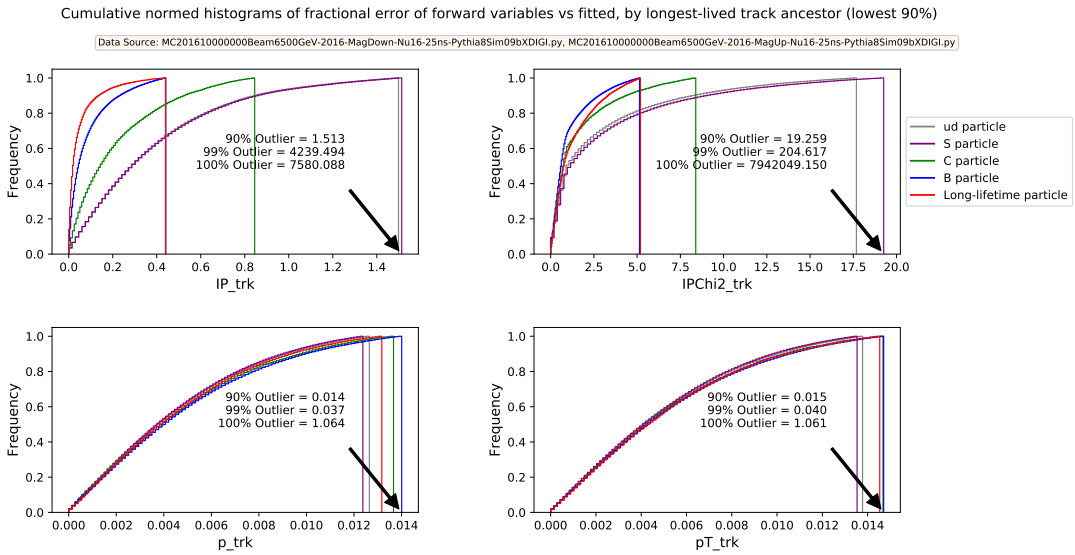


Fig. 4.10

Normalised histograms of the fractional error between fitted and unfitted forward-type track variables, separated by MC particle truth value. The x axis indicates the fractional deviation of the unfitted tracks from the fitted tracks, with the y axis indicating the fraction of tracks with error less than this value. Only the lowest 90% of the tracks by fractional error are displayed, as there exist very large outlier values (the 90, 99, and 100 percentile values are indicated by the arrows for each variable).

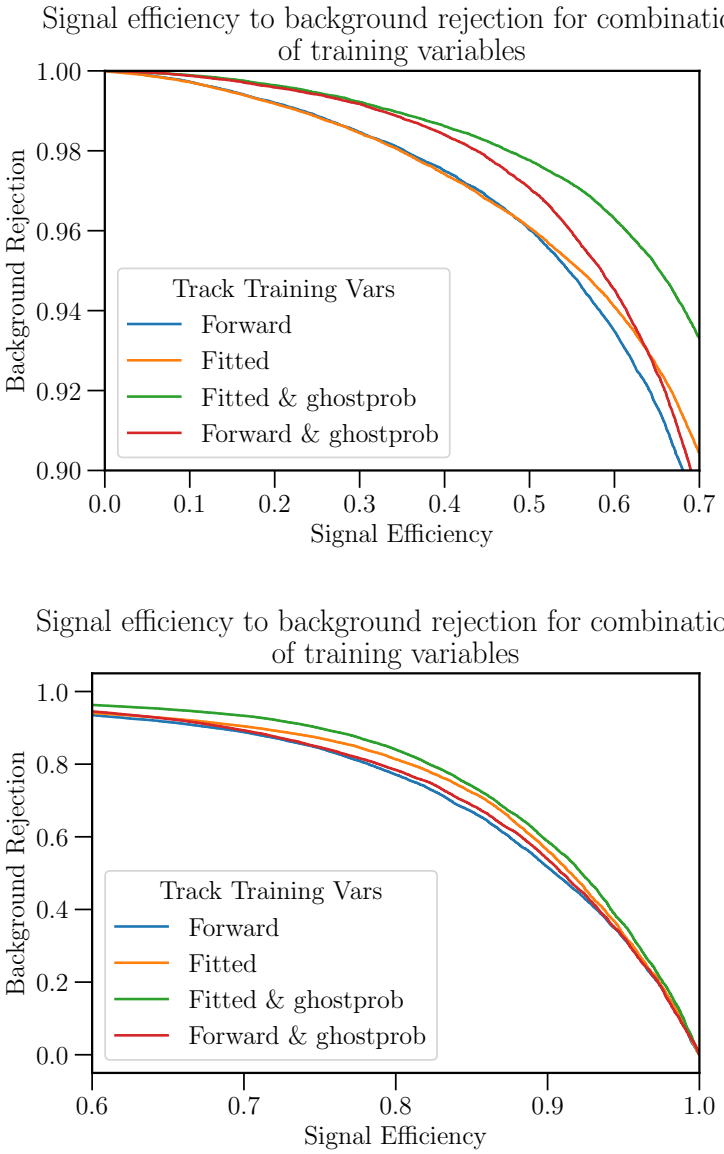


Fig. 4.11

The classification performance of several combinations of training variables in the high-background-rejection (top) and high-signal-efficiency (bottom) regions. "Fitted" tracks are forward-type tracks that have been processed by a Kalman filter to refine the kinematic variables used in training (IP, χ^2_{IP}, p, p_T).

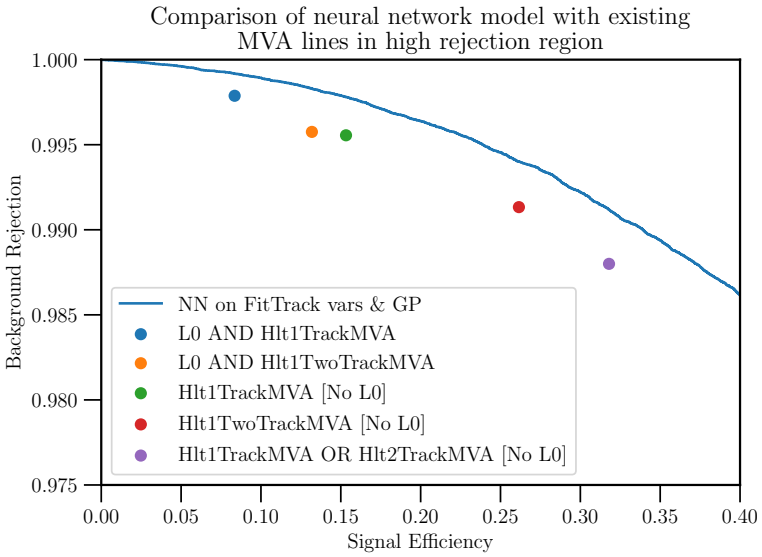


Fig. 4.12

A comparison of the performance of a neural network classifier (line) and the existing MVA lines (points) in evaluating single tracks. The neural net model is displayed as a curve, as it outputs a continuous value, to which any cut can be applied. "FitTrack vars & GP" refers to the variables listed in section 4.6.1, which have been refined via the Kalman fit procedure.

4.7 | Per-Track Accuracy

As shown in figure 4.12, the use of a small neural network gives an improvement in inclusive-b track classification over the existing MVA lines, giving greater background rejection rates for the same signal efficiency.

4.8 | Per-Event Inclusive-b Decision

This section investigates whether an inclusive decision for an entire event could be more accurately computed by taking into account information about all tracks in the event in a multivariate way. A sufficiently accurate decision that operates on all one-track MVA decisions in the event may obviate the need to run the combinatoric vertexing fits in some events, which would improve average trigger times. This would be especially advantageous given that the number of vertexing combinations grows roughly with the square of the event occupancy, which will increase significantly for run 3. Multiple models were considered for this, but more sophisticated options (e.g. recurrent neural net-

works, graph convolutional neural networks) were eliminated, due to the impracticality of such computation in the early stage of the trigger, and that the underlying data structure of the tracks (the decay tree) is not knowable at this stage.

4.8.1 | Inputs Pooling

The current inclusive-b implementation (a set of binary classifier evaluations) passes a whole event in HLT1 if a single constituent track (or track combination) passes the respective trigger line (in other words, the event-wide decision is the logical *OR* of all track decisions). For a probabilistic binary classifier such as a neural network, this is equivalent to selecting the highest pass probability in the set (appendix A.1) and comparing it with the "pass" output cut.

As well applying this method to the neural network track classifier, a method of track "pooling" was tested. In this method, all (single) tracks in the event were run through the neural network, producing a set of pass probabilities. This set of numbers was then fed into a number of *pooling*, or variadic, functions. The functions used were *sum, arithmetic mean, geometric mean, maximum, maximum/arithmetic mean, sample variance, and N*, where N is the size of the set. The set of outputs from these functions (which is fixed in size) was then used as input to another neural network, along with global event information (such as detector occupancy counts).

The method of using pooling functions was found to have a small increase in classification performance compared to the maximum pass probability cut method in the efficiency < 0.5 region. The addition of global event information to the network did not significantly alter classification performance.

4.9 | Two-Stage Selection To Significantly Reduce Number of Kalman Fits

The currently-existing Kalman fit algorithm applied to all forward tracks in an event has been found to take (11.0 ± 0.091) ms to process per event on average (appendix A.11). Scaling this linearly by the expected increase in ν^* of 1.6 to 7.6 from run 2 to run 3 gives an average cost of (52.2 ± 0.4) ms per track, which

*ν is the average number of pp interactions per bunch crossing

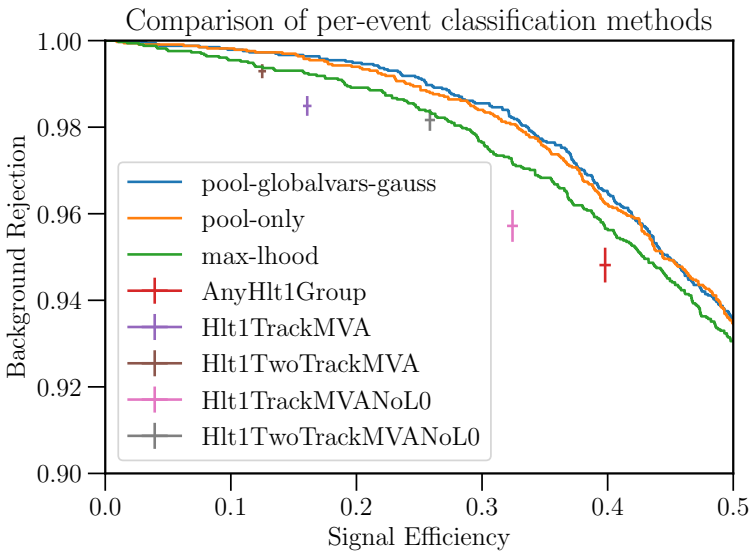


Fig. 4.13

A comparison between three methods of computing an event-wide inclusive b decision from the set of per-track inclusive b decisions. "pool-only" refers to the method of pooling the set of track pass probabilities as outlined in section 4.8.1. "pool-globalvars-gauss" refers to this method, with the addition of global detector occupancy data as input to the event-wide neural network (normalised to a Gaussian distribution). "max-lhood" refers to the method of doing a cut on the largest pass probability in the set. These three methods are compared to the various existing event-wide MVA decision.

is far beyond the allotted 13 ms for the entire HLT1 [74]. Here, a method for applying a selection before the track fit stage is discussed.

Instead of a selection applied from the decision of a single neural network with fitted track variables as input, a two-stage selection is constructed, consisting of two similar neural networks. The first-stage neural network accepts all tracks, with input being the forward track variables that have not been refined via the Kalman fit. This first stage performs a cut on this set of tracks based on their individual pass probabilities, rejecting a certain fraction of the set of tracks. This corresponds to a single point in the space of signal efficiency and background rejection. The Kalman fit is then applied to the remaining fraction, and the refined track variables are used as input to the second neural network. Varying the probability cut on the second-stage network as a free parameter produces an efficiency-rejection curve, with one end at (efficiency $= 0,$ rejection $= 1$), and the other end at the point produced by the first stage. This method is demonstrated in figure 4.14, with an arbitrary cut on the first-stage network as an example.

Despite this method involving a preliminary selection using lower-quality variables (non-Kalman fitted), it can still attain the same classification performance as a single-stage selection using only the higher-quality Kalman variables. From figure 4.14, any point on the higher curve (single selection on Kalman variables & ghostProbability) can be reached by first applying a slightly more lenient selection with the non-Kalman variables, and then performing a secondary selection on the reduced set of remaining tracks. This basically means that the vast majority of tracks can skip the Kalman fit for the inclusive-b selection without sacrificing classification performance (figure 4.15). Appendix A.8 describes the procedure to calculate this.

4.10 | Neural Networks In The Trigger Framework

Although platforms such as PyTorch and TensorFlow offer the ability to create more sophisticated machine learning models, this capability is not required for the feed-forward neural networks used in this work. A review of machine learning frameworks and their applicability to the HLT is outlined below:

- TensorFlow

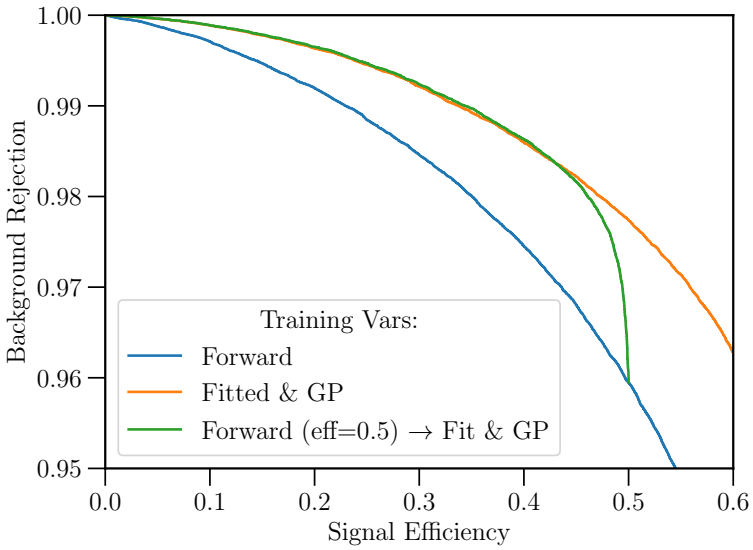


Fig. 4.14

Example of the performance of a two-stage classifier. For the green curve, the first-stage neural network trained on the forward-track variables processes the whole dataset, and a selection is performed based on a cut on its output response (in this case, an arbitrary example cut corresponding to an efficiency of 0.5 was used). This reduced data is then input to the second-stage neural network, and a second selection is applied. The curve reflects the combined performance of this two-stage classifier over the full dataset.

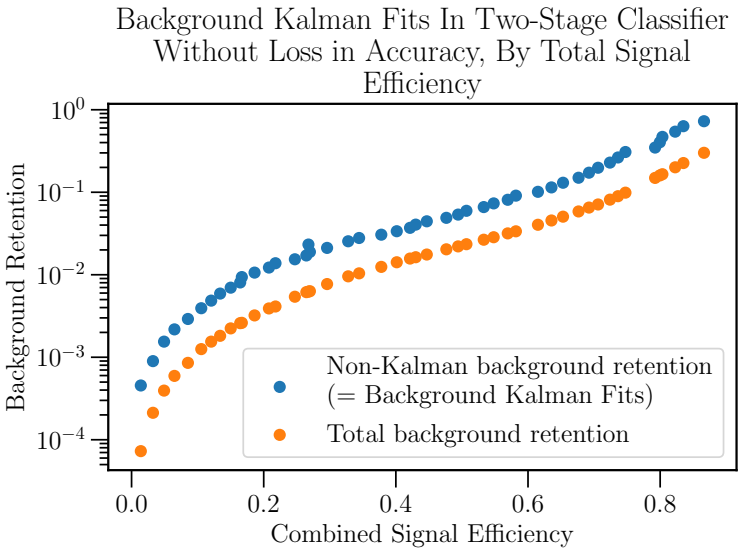


Fig. 4.15

The maximum number of background tracks that may be rejected without loss of classification performance. The higher curve (blue) shows the fraction of remaining background events out of the total that remain after the first-stage selection, as a function of the combined two-stage signal efficiency. This curve is equivalent to the fraction of tracks that the Kalman fit must be applied to before the second-stage selection. The bottom curve is the combined background rejection fraction after the second selection, as a function of the combined signal efficiency. For example, with a total efficiency of 0.2, it is only necessary to apply the fit to just over 1% of tracks, and the background fraction is further reduced by roughly a factor of 3. Values were calculated via the procedure in appendix A.8.

TensorFlow is one of the most common frameworks for machine learning. It has a versatile production options, including Python and C++, but requires the large TensorFlow environment, which may be difficult to embed in the Gaudi workflow. There are more minimal options for production, such as a C shared object library, however this is recommended by the developers as a means for creating bindings for other languages, rather than use as a library.

- PyTorch

  PyTorch is another industry mainstream platform that works with a dynamic, JIT-compiled computational model (as opposed to the statically compiled one of TensorFlow), which is well suited for research. Until recently, there were few options for high-performance deployment of PyTorch models, but as of December 2018, models can be interpreted and processed with a C++ library.*

- lwtnn (Lightweight Trained Neural Network)

  A neural network evaluation library based on C++ with minimal dependencies. Not as versatile as TensorFlow and PyTorch but capable of processing simple architectures (such as in this work).†

- TMVA

  Already well-integrated with the Gaudi framework. Has more limited versatility, but natively supports simple deep neural networks, and more recently supports the Keras (a frontend library to other machine learning platforms like TensorFlow).

This section presents the computational load of various machine learning libraries. The computation times are recorded for various batch sizes. All libraries evaluate the same feed-forward neural network, with the following shape (intended to be a definite upper bound to the limit of classifier performance found in section 4.6):

The results were determined by timing the network evaluations running as a standalone program on a computer with minimal background load from other processes. Results for the computation time of the models running on these standalone platforms may differ when compared to the frameworks running

*https://github.com/pytorch/pytorch/releases/tag/v1.0.0
†https://github.com/lwtnn/lwtnn

Input size	5
Hidden layer size	40
Number of hidden layers	2
Output size	2
Activation function	LeakyReLU

Table 4.2

Description of the feed-forward neural network used to profile the performance of machine learning platforms

inside Gaudi due to caching issues.

Timing unbatched network evaluations in compiled C++ (TensorFlow C-API and lwtnn) was performed by timing $nLoop \gg 100$ evaluations, and dividing the resulting time by $nLoop$, to minimise the effect of running the timing code itself. All platforms were timed 10 times, to produce a mean time with an uncertainty estimation. In the PyTorch platforms, the first trial took multiple σ more time than the others (presumably due to first-run configuration or caching issues), and so was not included in the calculation.

The PyTorch-Python and TensorFlow profiles were found using "machine A", described in appendix A.12. Due to CMake versioning issues, the PyTorch-C++ and lwtnn profiling was performed on another machine ("machine B"). An attempt was made to normalise the performance between the two machines. As lwtnn is a single-threaded CPU-based library, the timing of this entry has been normalised by the ratio of scores of the two CPUs in the PassMark benchmark*:

$$\frac{\text{score(FX-8320)}}{\text{score(i7-4790)}} = \frac{1397}{2282} = 0.61$$

The PyTorch C++ library scales the number of cores used as a function of batch size, so the value quoted for a batch size of 1 in the table has been scaled by this ratio, but the values in figure 4.16 have been left as-is.

*https://www.cpubenchmark.net/singleThread.html

Platform	Time	$\pm\sigma_{\bar{t}}$ (μs)	Machine Used
PyTorch (GPU)	210	± 24	A
PyTorch (CPU)	103	± 6.2	A
PyTorch C++ API (CPU, scaled)	41.8	± 0.10	B
Tensorflow C API	3.79	± 0.027	A
lwtnn (scaled)	1.43	± 0.0021	B

Table 4.3

Mean processing times and their standard errors in μs per NN evaluation, at a batch size of 1. The large fractional errors for the PyTorch framework on GPU is likely an artefact of the large overhead of the framework and the correlation with the behaviour of all other users on the system, due to unique resources that may need to be allocated.

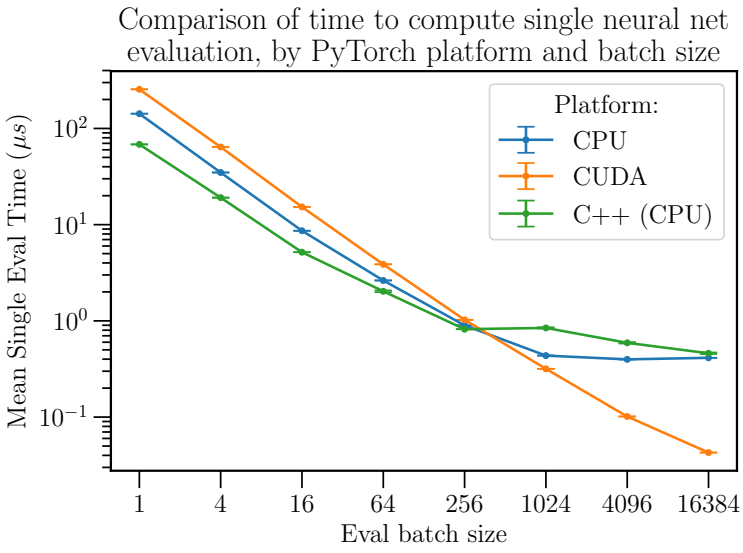


Fig. 4.16

PyTorch uses the parallel framework CUDA to process models on the GPU, allowing for evaluation time that scales inversely with batch size beyond that of CPU-based computation. Evaluating the model in these two modes in Python shows that for this machine ("A", see appendix A.12), the CPU is faster for batch sizes of up to 256.

4.11 | Future Upgrade Data Analysis

To coincide with the physical LHCb upgrade, the LHCb trigger and online system will operate inside a new software framework still in active development, including a functional event loop system based on the declarative resolution of data dependencies. As such, the Moore trigger framework and all trigger lines used are in the process of being rewritten. Work has been done to port the software developed to extract data for this study to the upgrade trigger framework in its current state of development. This may be used to perform a similar analysis to the one described in this chapter on upgrade simulated data.

Since this analysis was conducted, the Allen application has been developed. This application, running inside the Moore trigger framework, allows the HLT1 trigger sequence to be run in parallel on GPU hardware[86], including the track and vertex fitting. This means that the Kalman track fits, which was the most computationally time-consuming aspect of the HLT1 in run 2, may no longer require a reduction in processing time to viably run on all tracks in the event in run 3.

Chapter FIVE

LHCb VELO Upgrade, Testing and Quality Assurance

In this chapter, the details of the first major upgrade to the LHCb *VELO* (VErtex LOcator) subdetector component is described. A test pulse analysis of the LHCb VELO upgrade electronics is conducted, in order to verify operation in accordance with the specifications described in the Technical Design Report [102]. A software stack has been created to encode, decode and analyse streams of data from the VELO upgrade, which is also described.

A glossary of terms related to the VELO is displayed in appendix B.1.

5.1 VELO Upgrade

Throughout the operation of the LHCb experiment prior to run 3, the LHCb detector has used a silicon strip detector known as the VELO as the central component for precise vertex and track reconstruction around the collision point [67]. As part of the LHCb upgrade for Run 3, this strip detector will be replaced by an in-development silicon pixel detector.

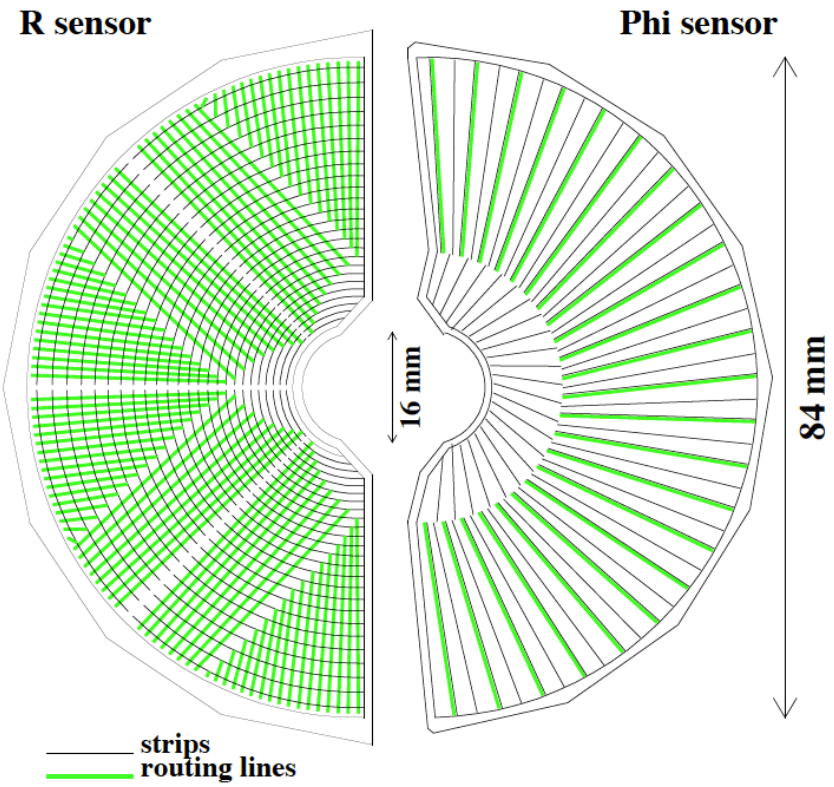


Fig. 5.1

Schematic view of the silicon sensor strips in R and ϕ for the original VELO detector [103].)

5.1.1 | Previous Strip Design

The previous VELO design consists of a set of modules, each of which being composed of two adjacent silicon strip detectors enclosing each side of the beam line. Each back-to-back pair of sensors has one with radial strips and one with azimuthal strips, such that a track passing through the sensors will have its position resolved in R and ϕ separately. Both the R and ϕ sensors are designed such that the strip pitch (distance between consecutive strips) is approximately $40\,\mu$m nearest the beam line, up to $100\,\mu$m furthest away (as shown in figure 5.1). In the R sensor, the azimuthal strips are spaced more closely for lower r. For the ϕ sensor, the constant angle of the radial strips means that the strip pitch increases linearly with r. The ϕ sensor is divided radially into two regions such that the angular density of the strips doubles from $r \geq 17.25\,$mm, and the outermost strip pitch is $97\,\mu$m [58].

5.1.2 | Pixel Detector

The upgraded detector uses sensors with an 256×256 array of $55 \times 55\,\mu\text{m}^2$ square pixels, giving the modules a constant precision in the plane orthogonal to the beam, that is slightly coarser than the previous design near to the beam, but finer further away.

In a pixel-based design, a hit's x and y position is unambiguously determined by which pixel of the sensor had charge deposited. In contrast, a strip detector must uniquely determine a point in the plane by combining information from two orthogonal sensors (the radial and azimuthal silicon strips). The pixel design therefore has the advantage of lower track ghost rates*, particularly at higher luminosities. The new, square geometric design of the upgraded VELO also means that the closest sensor material will be 5.1 mm from the beamline, rather than 8.2 mm for the old strip design [104].

5.2 | Geometry

The upgraded VELO is comprised of 26 tracking stations, each of which is made up of a pair of VELO modules that may be opened and closed around the beam line from each side. The two modules in each station are named "A side" ($+x$) and "C-side" ($-x$). A module contains 2 *hybrids* (PCB with serialiser ASIC), one on the front and one on the back. A hybrid fits 2 sensor tiles, each of which is a silicon sensor with a set of 3 bump-bonded VeloPix ASICs. There is a small gap in z between sensors lying on the front and back hybrids within a module. To ensure full coverage of large-angle tracks, there is consequently a 2-pixel ($110\,\mu\text{m}$) overlap between front and back sensors [102]. When closed, a pair of modules forms a roughly square acceptance region around the beam line. In all this gives a total of 26 stations, 52 modules, 104 hybrids, 208 tiles, and 624 VeloPix ASICs [102].

Modules are distributed along the beam line (figure 5.3) such that the maximum range of VeloPix sensors is $-277\,\text{mm} < z < 751\text{mm}$ (this includes a 12 mm difference in z between A-side and C-side modules in the same station).

Cooling the VeloPix and control ASICs (section 5.4) within the vacuum of the VELO is achieved by piping liquid CO_2 through microchannels in the silicon

*A *ghost* track is the name of a spurious reconstructed track that is an artefact of the detector and/or reconstruction process, rather than the actual path of a real particle.

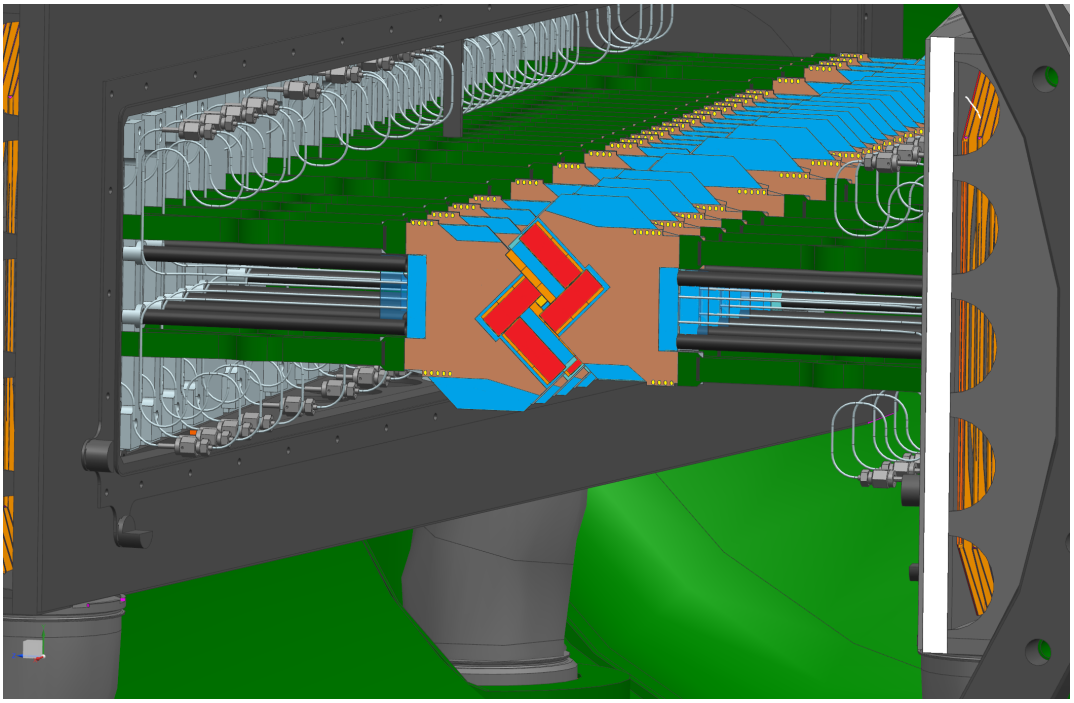


Fig. 5.2

3D rendering of the VELO upgrade stations, with left and right sides closed together. Sensors are shown in red at the centre. Areas not covered by sensors on one face of the module are covered by the sensor placements on the opposite face, with some overlap to accommodate tracks with high angles passing through the gap [105].

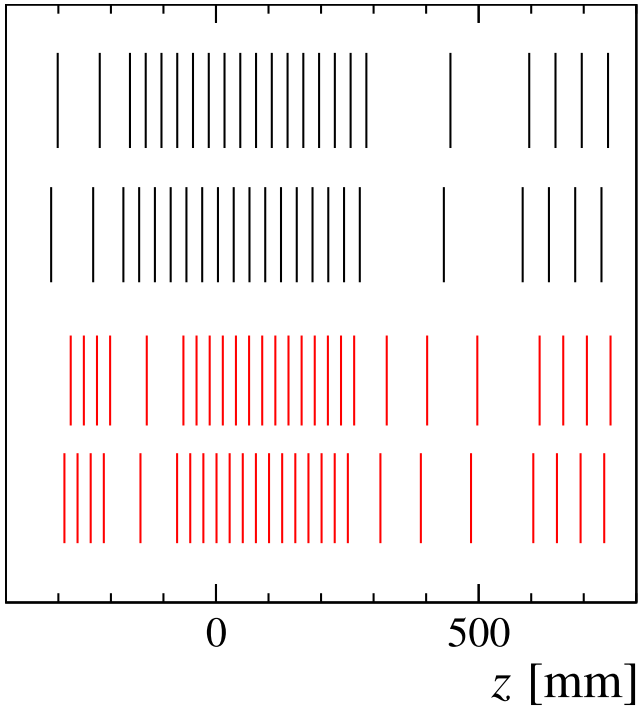


Fig. 5.3

Comparison of the z distributions of the VELO in run 2 (upper, black) and in the upgrade (lower, red) [102].

hybrid substrate, which evaporates as it passes through. The channels are made up of a series of wafers containing microscopic holes bonded to cover grooves etched into the substrate, which allow the liquid CO_2 through [106].

5.3 | VELO Frontend Electronics

The upgraded VELO system is based around the *VeloPix* ASIC [107], which governs the detection, arrangement, and readout of hits striking the sensors. VeloPix is based on the earlier design of the *Timepix3* ASIC [108], with attributes tuned to the physics use case of the VELO such as a lower internal timestamp resolution, higher pixel hit rate, and higher output bandwidth than the Timepix3 chip.

Each VeloPix ASIC has 4 readout serialisers (GWT or *Gigabit Wireline Transmitter* [109]). A GWT serialiser can transmit 128-bit dataframes at up to 5.12 Gb/s, and a variable number may be enabled depending on the estimated necessary bandwidth. The hottest ASICs closest to the beam line in a given module have all 4 serialisers enabled, whereas the ones furthest away have only

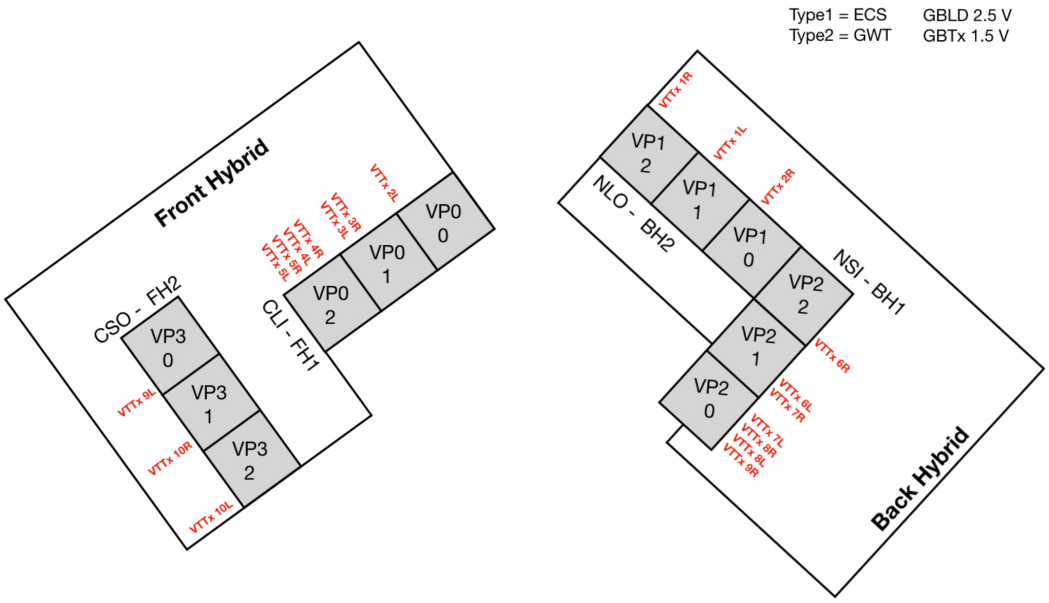


Fig. 5.4

Schematic of the relationship between VeloPix ASICs (grey squares) and their respective optical links (red text) enabled in normal operation.

a single serialiser enabled. As a result a single hybrid will use 10 GWT serialisers and thus 10 optical links. The exact relationship between ASICs and links is shown in figure 5.4. A greater number of links are needed for the ASICs closer to the beam as these will be exposed to a greater intensity of radiation during normal operation, and therefore require higher output bandwidth, as shown in figure 5.11.

5.3.1 | Pixel Electronics

Charged particles travelling through the VELO cause a current to flow through the high bias-voltage sensor pads of each pixel, and fed through a pre-amplifier.

The VeloPix chip, being a binary readout design, involves a conversion between the amplified analogue current from the sensor, and a discrete digital output defining whether a pixel has been struck by a particle. The analogue electronics in each pixel are naturally subject to noise of a particular mean and variance (estimated 130 e^- RMS [102] per pixel). Consequently, the chip must respond with a hit only when a particular voltage threshold is crossed in the pixel. This threshold is defined at the chip-level by a 14-bit value, with each pixel having

its own local unsigned 4-bit value. These two values are summed together and passed through a digital to analogue converter (DAC), and compared by a discriminator to the incoming voltage in each pixel to inform the pixel's binary response.

Figure 5.5 shows the readout design of the VeloPix ASIC. Pixels are grouped into readout blocks of 2×4 pixels called *superpixels* (this only affects the encoding of the data read out, rather than any physical aspect of the VeloPix such as resolution). The use of grouping pixels into superpixel groups works to reduce output bandwidth in all but the highest-occupancy conditions, as tracks tend to produce clusters of hits in the sensor (55% of tracks yield more than one pixel in simulation [102]). This means that if at least one pixel in a superpixel grouping is hit in an event, the timing and address information need only be written out once for the whole group of 8 pixels, at the expense of specifying whether each pixel in the superpixel was hit (8 bits). Also, since there are fewer superpixels than pixels, fewer bits are needed to specify the superpixel's address than for an individual pixel.

Superpixels are grouping of 2 horizontal $\times 4$ vertical pixels, meaning that a 256×256 pixel array contains 128×64 superpixels. The superpixels of each ASIC are arranged with their longer dimension pointing in the radial direction from the beam line. This is because particle tracks are more likely to form radial clusters in the VeloPix sensor. The superpixels in the VeloPix ASIC are arranged into column buses. When any pixels are hit, the 8-bit hitmap is combined with the timestamp and superpixel index information held locally within the superpixel into a 30-bit *superpixel packet*, or SPP, which is then stored in a buffer belonging to the superpixel on a first-in-first-out basis. If hits are received while both slots are full, the newest SPP is lost. The buffers of each superpixel together feed into the column bus that transports SPPs down to the EoC fabric by one superpixel per clock cycle (with clock cycles synchronised to bunch crossing events at 40 MHz), until the end node is reached in the end-of-column fabric ("EoC" in figure 5.5). From here, each side of the EoC fabric moves towards the centre column, where it is processed by the SPP router and serialiser. This means that a super pixel in row N would reach the end-of-column fabric N clock cycles (events) later. For this reason it is critical that the true hit time is reconstructed properly in order for the hit to be assigned to the correct event (section 5.5.6).

Access to the column bus node of each superpixel is competed for by the tailing

superpixel node and the local buffer, which is settled by a weight round-robin arbitration scheme [110, 111]. This entails using a local 6-bit counter in each superpixel to determine the favoured data source. A 6-bit counter has $2^6 = 64$ possible values, which is the same as the number of superpixel nodes in the column (256/4). The counter increments each clock cycle, wrapping back with $63 \rightarrow 0$. The superpixel's local buffer is favoured if the counter is 0, otherwise the previous node in the column is favoured. For a constant clock cycle, the counter in each superpixel is 1 lower for each superpixel down the column (with wrapping). This has the effect that there is at all times a single superpixel in the column that favours its local buffer, and this "special superpixel" constantly moves away from the end-of-column fabric (and wraps back) at a rate of one superpixel per clock, in the same way that an "electron hole" can be thought to move through a conductor.

The end-of-column fabric consists of 4 separate buses, each connected to a different set of EoC nodes modulo 4 (as in, a single lane will connect EoC nodes 0, 4, 8, etc, and another lane will connect 1, 5, 9, etc). The router is connected to 4 GWT-based serialiser links. There is no exclusive relationship between EoC buses and serial links; the router can send SPPs from any bus to any link. Unlike other components in the LHCb detector, the VeloPix ASICs serialise the frontend data to send to the DAQ system with the GWT (Gigabit Wireline Transmitter) system, rather than the GBT (GigaBit Transceiver, see section 5.4). This is due to the GWT's slightly higher maximum bandwidth and lower power consumption (which is a key concern for such vacuum-operated electronics).

5.4 | VELO Control and Data Processing

The full LHCb detector's move to a 40 MHz triggerless readout system has led to a new generation of its readout system being developed, which is also used by the VELO component. The *Timing and Fast Control* system (TFC) is responsible for the synchronisation and test pulse input for the VeloPix with clock-cycle (25 ns) precision. The Experiment Control System (ECS, also known as slow control) allows for the readout of the VeloPix registers. These protocols are implemented with three separate boards, *S-ODIN*, *TELL40* and *SOL40*, which have homogeneous hardware and differing firmware. The hardware designs are all based on *PCIe40*, a readout system designed for synchronous readout from the front end to data processing at the LHC bunch crossing frequency of 40 MHz

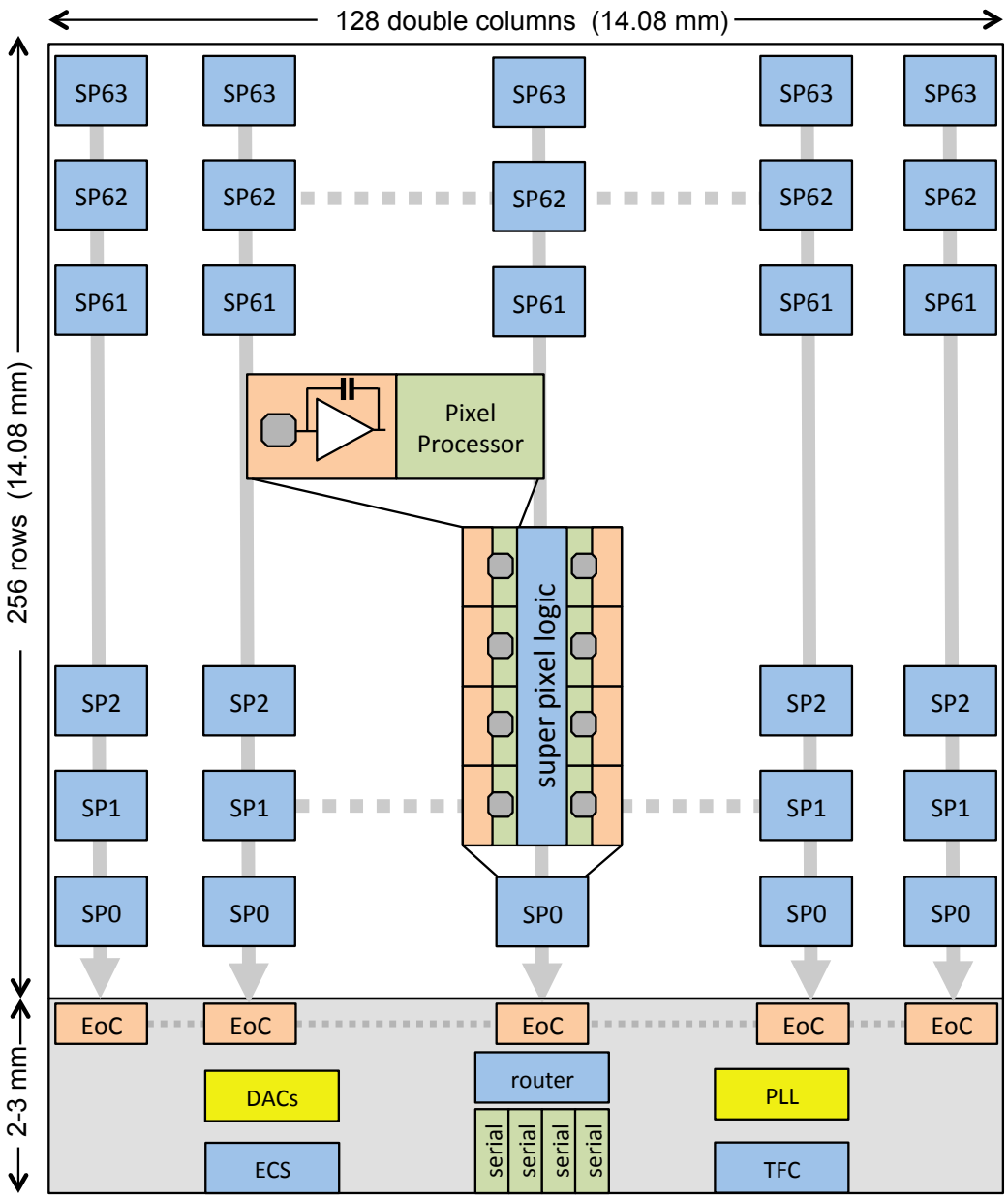


Fig. 5.5

Schematic of the superpixel data flow inside the VeloPix ASIC [102].

[112]. PCIe40 runs inside commercially available server hardware, interfacing via 3rd-generation PCIe with the maximum of 16 lanes, giving a maximum bandwidth between the board and server motherboard of 15.8 GB/s.

The readout supervisor, S-ODIN [113], acts as a centralised interface to distribute TFC signals and collect data to be sent to DAQ systems. The interface board, SOL40, is responsible for distributing TFC and ECS information to multiple frontends. Both fast and slow controls are transmitted over the same duplex link via the GBT (GigaBit Transceiver) technology (a radiation-hard chipset designed to transmit 120-bit dataframes at up to 4.8 Gb/s [114].

TFC and ECS commands may be sent manually via a graphical frontend interface that emulates the state of the Optical and Power Board and PCIe40 boards (below) as a finite state machine.

5.4.1 | Optical and Power Board

The Optical and Power Board, or OPB [115], sits between the front-end hybrids and the remote data-processing electronics such as the TELL40 and SOL40. A single OPB serves both front and back hybrids for a single VELO module. It has a direct electrical connection to the hybrids via flexi cables through vacuum feed-through tubes, and is responsible for providing power to hybrid components by performing DC-DC voltage conversion. The OPB is also responible for converting the electrical data flow from the frontend serial links into fibre-optic channels to send to the TELL40. This is driven by 16 *VTTx* (versatile twin transmitter) modules [116], for the combined 32 serial links on the front and back hybrids. Optical control signals in the form of TFC and ECS are also converted to electrical signals by 3 *VTRx* (versatile transceiver) modules [116]. Figure 5.6 describes the data and power flows mediated by the OPB.

5.4.2 | TELL40

The readout board, TELL40, is a device that recieves real-time data from the detector frontend electronics and organises it into a format that can be sent from the DAQ to the event filter farm over the network.

In the VELO, the TELL40 receives fixed-size dataframes from the VeloPix GWT serialisers, and processes the data as shown in figure 5.7. One TELL40 board contains 2 of these pipelines running in parallel, each of which can accommodate the data from 10 GWT links. In other words, one TELL40 board

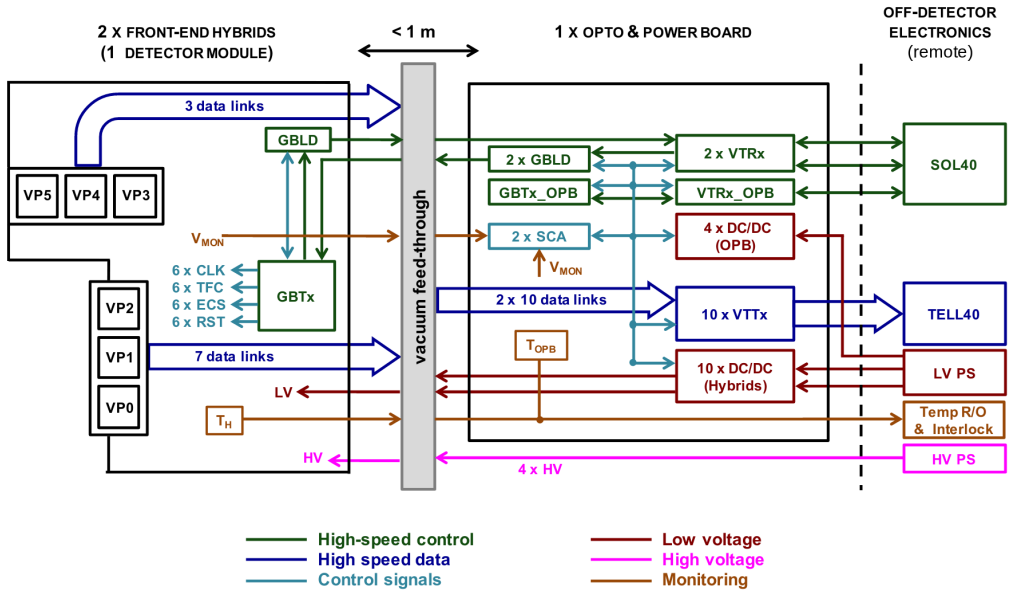


Fig. 5.6

Diagram of the power, data, and control flows of the VELO optical and power board (OPB). [115]

can process the data from 20 links, or one full VELO module.

The dataflow is made up of three distinct sections:

- Low-Level Interface (LLI)
- Data processing
- Event ID and MEP building

While the latter is shared across the LHCb framework, the first two are unique to the VELO.

When dataframes arrive at the TELL40, the LLI aligns them to the TELL40's 32-bit word size via a search for the `0xA` frame header. This alignment is verified through the use of a parity check with the frame's 4 parity bits. If this check passes, the 30-bit SPPs (*superpixel packets*, section 5.3) are descrambled, and the resulting frame is sent to the data processing block.

The data processing block first checks the SPPs for "special" frames. Since the VeloPix output is unchangeably zero-suppressed, an SPP with a fully-zero hitmap is not a valid representation of data, and can therefore be used to encode frames with a special state. In this case, the 30 bits of the SPP are

reinterpreted as:

- MSB $29-26$: 4-bit header encoding type of special SPP
- MSB $25-8$: Configurable 18 bits
- MSB $7-0$: Empty hitmap

With the possible special frames and their configurable formats listed in appendix B.3

If the SPP has a non-zero hitmap, indicating a normal data frame, the 9-bit BCID (bunch-crossing ID) is decoded from Gray code* to binary, and the individual SPPs are sent to the router, which orders the SPPs by bunch crossing number (BCID, see section 5.5.6).

The post-router section of the TELL40 then optionally performs additional computation on the SPPs, such as isolated cluster flagging (ICF), where a bit in the SPP is set depending on if it has a direct neighbour with a hit in the current event. The reordered SPPs are then formatted to a defined TELL40 format, and wrapped in the MEP LHCb transport layer before being sent to the event filter farm.

5.4.3 | VELO Bypass

In order to test the integrity of the readout pipeline, a method of extracting raw, unprocessed test pulse data from the VeloPix's GWT is desirable. For this, custom firmware is used in the TELL40 readout card, that bypasses the data processing block, including the router. As well as the data frames from the GWT serialisers, this firmware also adds debugging information to the output, such as a full 64-bit timestamp and metadata about the output stream, shown in section B.2.1. The 256-bit GWT bypass frame is thus twice the size of a regular GWT frame (which is simply an 8-bit header and the four 30-bit SPPs).

*A binary encoding system whereby incrementing the encoded value requires a single bit flip for all values, which has the potential to reduce energy consumption and error rates via reducing the required operations on the data in counting situations.

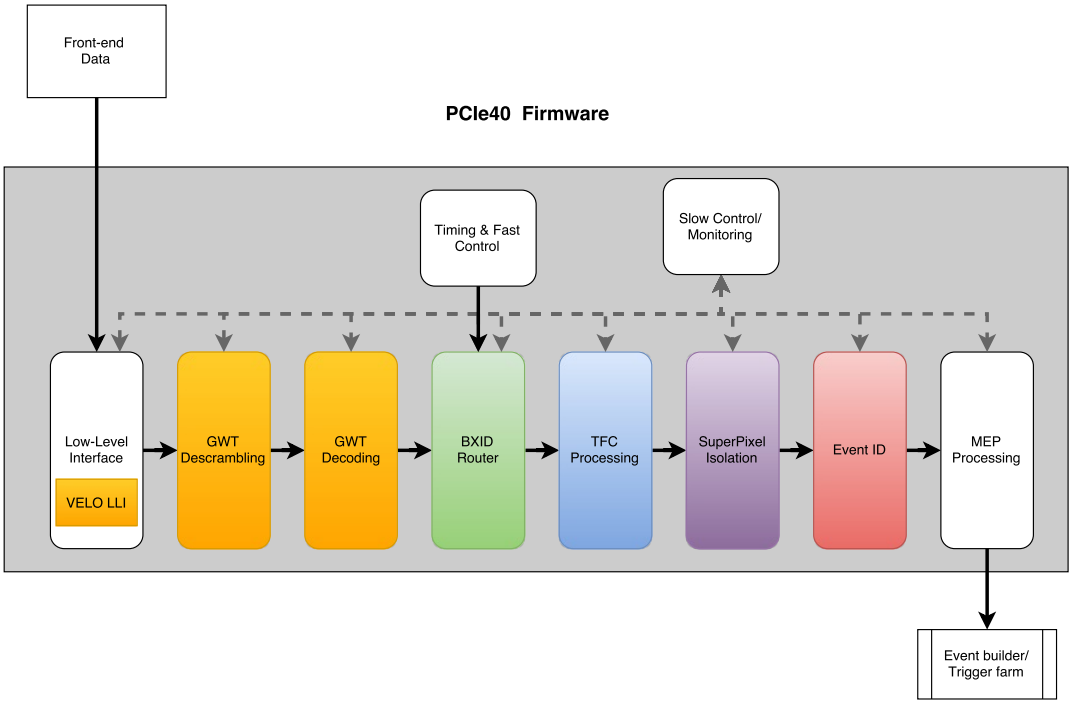


Fig. 5.7

Simplified flow diagram of the data processing pipeline of the production TELL40 firmware. Credit: Karol Hennessy

5.5 | VeloPix GWT Decoder Software

In order to analyse the behaviour of the VeloPix ASICs, the binary GWT bypass data must be decoded into a format that can be easily verified for integrity and fed into software for processing. For this, a software framework to manage encoding and decoding between formats was written (https://gitlab.cern.ch/LHCb-VeloPixSW/GWTBypass_SWDecoder).

5.5.1 | Software Architecture and Design

The software is written in standard C99 without extensions, to ensure maximum portability. Many programs that decode incoming byte streams to sequences of values do so by reinterpreting a byte sequence as a C struct with values of the same lengths, in the same order. However, the standard C ABI includes padding in memory between members of structs, in order to improve efficiency by aligning members to various processor word sizes. This extra memory means that the aforementioned technique requires use of the non-standard `__attribute__((__packed__))` struct qualifier. This approach also requires

further processing if the endianness of the incoming data does not match that of the host system. For this reason a simple, dedicated library was written to convert between arbitrary slices of bits in a byte buffer, and `uint64_t` types.

The software was also written with dependencies on only the C99 standard library, and minimal POSIX functionality.* It uses CMake as a build system, as this is the build system used by the LHCb software framework (Gaudi and Gaudi-based applications).

The structure of the software is made up of a stack of 4 independent layers, which reside in the source tree as 4 separate directories:

1. exe: Contains the code to actually create an executable, as well as parse command line arguments and set up the program with the options given.
2. gwtdecoder: Contains a set of modular objects to transform particular data structures (for example, take an output frame and decode it, or take out-of-order SPPs and output them ordered in time). The modules can be "pipelined" in different configurations (for example, a module that decodes the SPPs, and then insert the reordering module into the pipeline to output ordered SPPs), to allow for code re-use.
3. format: Contains the definitions of any input data formats, and some helper functions for them.
4. bitmanip: The basic code for decoding or encoding between a slice of bits in a byte buffer, and a `uint64_t`.

5.5.2 | Executable Wrapper

The source files in the `exe` subdirectory of `decoder` implement a simple executable wrapper to parse command-line options and run the resulting decoder setup. The rest of the software may be compiled as a standalone shared-object (`.so`) library to be run from within another program or framework.

5.5.3 | Module System

The execution of a decoding/encoding process in the software is handled by setting up a pipeline of modules at the beginning of runtime. The software defines a set of module files inside the `gwtdecoder` directory, each with its own

*POSIX-compliant `getopt`, and `dprintf`

purpose. For example, the file `process_spp_reorder.c` defines a module that takes in a stream of super-pixel packets, one at a time, and emits a stream of the same packets, but ordered by their reconstructed sensor timestamps.

Each module file exposes a single function with the same name as its file, which returns a pointer to the module's struct of type `decode_module_s`. The module struct is statically linked inside the module file (meaning that the module is a singleton, designed this way as a module should only need to be used once in a decoding pipeline). The functions inside the module file are also static, meaning they are privately encapsulated inside the module. The module's functionality is exposed to the wider program via three function pointers on its struct, which are defined at compile time:

- `.init`:
  Function signature: `static void (void);`
  Use: Optional. Run once, if not NULL, by `run_pipeline()`
- `.run`:
  Function signature: `static void (struct decode_module_s *input);`
  Use: Required. Run any number of times, by the parent module.
- `.deinit`:
  Function signature: `static void (void);`
  Use: Optional. Run once, if not NULL, by `run_pipeline()`

Modules are composed into a pipeline according to the supplied options by the function `process_opts()`. After the list is set up, a pointer to the root module is returned. The process `run_pipeline()` runs each module's `init` functions from the root module through each of its outputs, then runs the root module's `run` function once. The root module should run its child's `run` function itself, and so on until the final module writes out the data. A parent module passes a pointer to itself to the child module's `run` function.

There are a number of defined output types for modules GWT decoding modules to pass data to the child module. These are defined in `decode_module.h` as:

```
{ NONE_TYPE, LINE_TYPE, SPP_TYPE, TELL40_TYPE }
```

Each type defines an extra struct to extend the definition of the `decode_module_s` struct. For example, a module that emits SPP data (whether ordered or not) has `enum module_output_type output_type = SPP_TYPE`. This means that

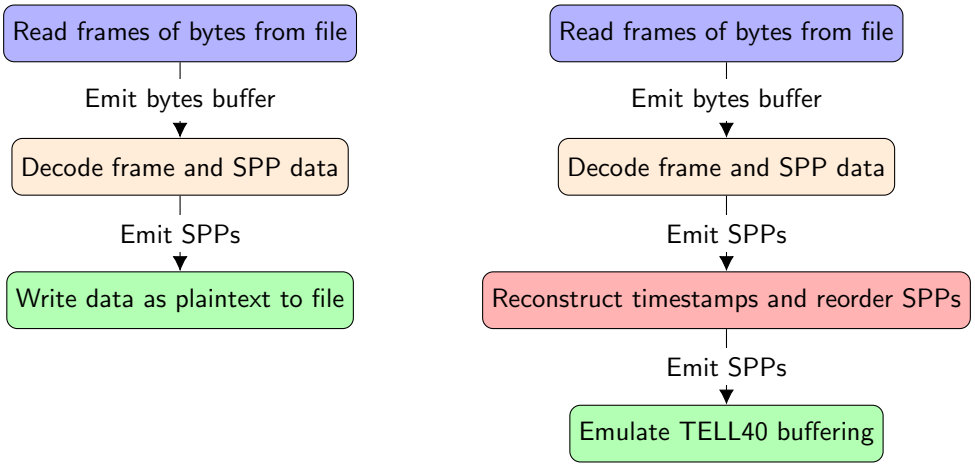


Fig. 5.8

Example decoding configurations. Left: A simple configuration to decode the GWT frame and write out as plain text. Right: Configuration to reorder SPPs and re-encode in TELL40 format.

the module's `decode_module_s` struct is actually a larger struct, `decode_module_spp_s`, which includes the regular `decode_module_s` data, and then the extra `decode_spp_s` data. A module that receives SPP data as input casts the input `decode_module_s` struct to the full `decode_module_spp_s` struct to access the extra data.

Figure 5.8 shows two examples of decoding pipelines that can be set up to process GWT data.

5.5.4 | Configurable Formats

As this software can decode arbitrary data formats, the GWT-related formats are specified in the `format` subdirectory. A data format is specified using a `.c` file and a `.h` file. The header file enumerates the different fields of bits inside the format's frame. The C file instantiates an array of structs that define the bit starting position, bit length, and endianness for each field in the enumeration. For example, for defining the fields of a normal SPP inside the GWT bypass frame:

decoding.h

```
1  struct pos_len_s {
2  int pos;
3  int len;
4  enum endian_e endianness;
```

```
};
```

decoding_config_spp.h

```
enum spp_field_e {
  FE_SPP_EoC_Addr,
  FE_SPP_SP_Addr,
  FE_SPP_BCID_9b,
  FE_SPP_Hitmap,
  FE_SPP_ALL_FIELDS,
};

extern const struct pos_len_s spp_fields_pos_len[
    FE_SPP_ALL_FIELDS];
```

decoding_config_spp.c

```
#include "decoding_config_spp.h"

const struct pos_len_s spp_fields_pos_len[FE_SPP_ALL_FIELDS] =
    {
  {0   , 7, BE},    //FE_SPP_EoC_Addr
  {7   , 6, BE},    //FE_SPP_SP_Addr,
  {13  , 9, BE},    //FE_SPP_BCID_9b,
  {22  , 8, BE},    //FE_SPP_Hitmap,
};
```

Defines the format for superpixel packets as shown in appendix B.2.

The natively-encoded values generated are stored as an array of `uint64_t`s, which is indexed by the same enums as the array of `pos_len_s` structs (for example, `mdv2_line_data[MDV2_Counter]` is a native `uint64_t`, whose corresponding encoded string has properties defined by `mdv2_fields_pos_len-[MDV2_Counter]`.

The information describing a set of fields in a data frame was chosen to be in the form of an array indexed by enums, rather than a struct with various named members. This is to enable the ability to decode an entire frame of data into native integers from a single function call (`decode_line`). This is possible because each data description enum should have a final member, eg `FE_SPP_ALL_FIELDS`. Since C enums by default start from 0 and are sequential, this final enum member is equal to the total number of data values to be decoded. The following snippet gives a placeholder example of a full set of values being decoded to native integers:

example.c

```
// unsigned char linebuf[]: contains frame of data
// uint64_t line_data[ALL_FIELDS]: to be populated
// struct pos_len_s fields_pos_len[ALL_FIELDS]: Describes data format
// ALL_FIELDS: Last member of data members enum; gives total number of values
decode_line(linebuf, line_data, fields_pos_len, ALL_FIELDS);
```

A buffer containing multiple bit fields may have fields of different endianness. However, this only makes sense if the buffers are byte-aligned, as two bit strings occupying the same byte will overlap if they have opposite endianness (as little-endian strings will occupy bytes starting from the least significant bit, whereas big-endian strings will start from the most significant bit).

5.5.5 | Bit Manipulation Library

The lowest stage of the software is a small library of functions to convert between bits inside a buffer of bytes and a host `uint64_t` number.

Details

The endianness of the bytes in the buffer is specified. The output `uint64_t` is always native (that is, it has the endianness of the host machine, likely to be little-endian).

The bit numbering scheme used to describe the bits inside the buffer is dependent on the endianness of the bytes. If the bytes are in big-endian order, the indices of the bit positions are in `MSB_0` format, ie the most significant bit is defined as 0. Likewise, for little-endian buffers, the indices denote `LSB_0` format, so 0 represents the least significant bit.

5.5.6 | Timestamp Reconstruction

When a particle hits a VeloPix pixel sensor, the pixel's superpixel logic unit records the bunch crossing in which the pixel was hit as a timestamp, before the data is sent down the column.

Due to the 25 ns period of the LHC, and the length of the beam ring, one 89.1 μs LHC orbit consists of 3564 bunch crossings. This means that all bunch crossings within a single orbit can be indexed within one 12-bit number ($2^{12} =$

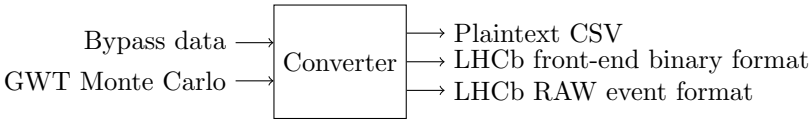


Fig. 5.9

Diagram of the possible input and output formats of the decoder software

$4096 > 3564$). In order to preserve output bandwidth, the VeloPix ASIC does not include a full 12-bit BCID with every outgoing superpixel packet. Instead, the number is truncated so that only the lower 9 bits are sent, corresponding to a window of 512 clock cycles or 12.8 ms. This means that 7 sets of 9-bit BCIDs are needed to index an orbit of 3564 crossings, with the final 20 units of the last 9-bit BCID left unused ($2^9 - (3564\%2^9) = 20$).

Given a 512-BCID window to identify SPPs, a full timestamp can be assigned by reconstructing a 512-BCID-sized "bucket" that each SPP occupies. For real data, this is possible by using the assumption that a given timestamp is not more than 512 BCIDs later than the one preceding it (in other words, that there are no windows of more than 512 events without any hits). Run3 LHCb will run at a high enough luminosity such that this is never likely to happen for even the coldest ASICs (given that the probability of receiving zero hits decreases exponentially for successive events). In this case, it is enough to keep track of the current "bucket", and increment this when the 9-bit BCID rolls over 0 again.

However, artificial test pulse data could involve arbitrarily large time gaps between hits. Fortunately, the bypass firmware records a full 64-bit timestamp every time it receives a new frame from the GWT. This can be used as a "ground truth" for an SPP's occupied bucket, and used to reconstruct the full timestamp. The 64-bit timestamp is only an approximation, though, as it differs from the true reconstructed timestamp by the amount of time between the hit being registered by the ASIC, and the SPP data being sent over the GWT. This depends on which ASIC on the sensor was hit (since the data takes time to travel down the column and along the EoC fabric), and also the recent rate of hits, as a large bandwidth will cause congestion along the buses.

See appendix B.4 for a simpflified example of timestamp reconstruction.

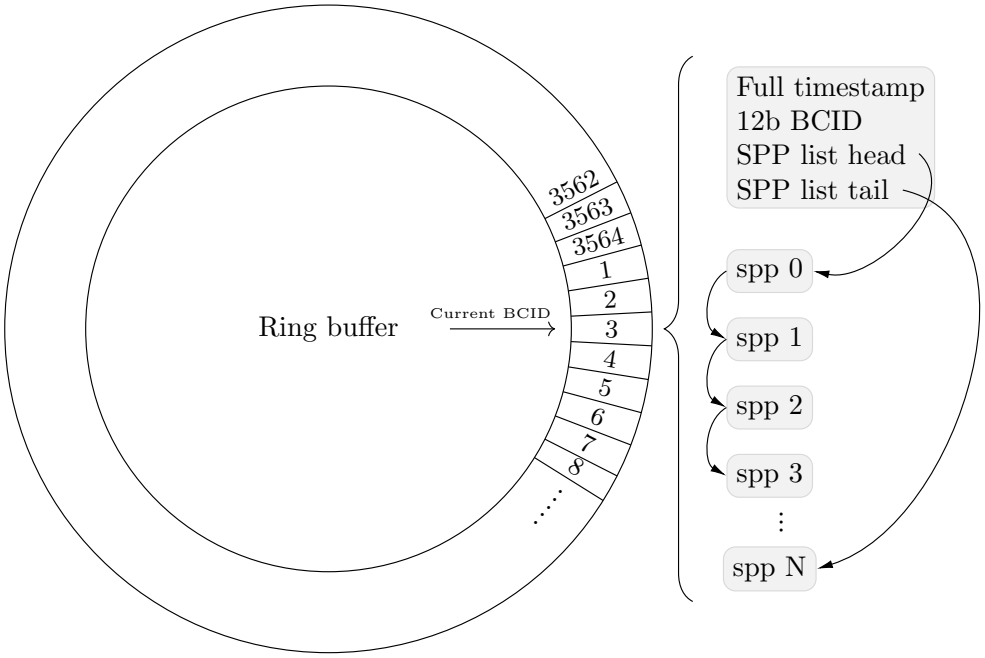


Fig. 5.10

Diagram of the data structure involved in SPP time reordering

5.5.7 Performance

5.5.8 Auxiliary Software Tools

As well as the software to decode VELO data between various formats, additional software was written for related tasks, such as test pulse analysis. The program `hitmap_convert` was created to convert between various plaintext representations of hits in a pixel matrix, and their in-memory counterparts.

Like the previously mentioned software, this consists of a shared-object library with a command-line interface wrapper over its functions. The program can convert a plaintext CSV representation of a pixel occupancy grid, and convert it into a 2D array of `int`s, and vice versa. The program can also convert hit information from an unordered list of coordinates to a grid of occupancies.

As well and reading from and writing to files, the decoder and hitmap conversion programs can also read and write data via the standard POSIX file descriptors `stdin` and `stdout`. This enables convenient pipelining of multiple tools, such as in this example:

example.sh

```
# Run decoder on /tmp/myinput.frg, then pass output hit list to
#hitmap_convert to produce an occupancy array:
```

```
./decoder < /tmp/myinput.frg | ./hitmap_convert > output_hitmap
    .csv
```

5.6 | Simulated Hit Distribution

Roughly one million unbiased Monte Carlo events were simulated inside the upgrade LHCb VELO (see table 5.1), and transported through the geometry of the VELO module closest in z to the beam crossing point (as this module will see the largest fluence of hits). The hits in the VeloPix sensors were then digitised and input to the simulated frontend logic (see section 5.3) before being output as GWT dataframes. The dataframes were then run through the software decoder to produce a chronological sequence of pixel hits, which were compiled into hitmaps. Figure 5.11 shows that the expected radial distribution of hits has been recovered from the simulation pipeline.

5.6.1 | Radial Fit

The two-dimensional output pixel distribution of the simulated events was normalised to sum to unity and fit to a simple power function Ar^{-k}. For the fit, the geometry of the chips is taken to be an array of 256×256 equally-sized square pixels. The "virtual" chips are arranged into rectangular "blocks" of 2×3 chips without overlap, and the four blocks are arranged without overlap into the shape shown in figure 5.11, such that each of the four innermost edges of the pattern are 5.1 mm from the beam line, which is taken as the origin $(0, 0)$. The centre of each pixel is used as the sampling point. This model of the VeloPix introduces some amount of error, as it ignores the slight overlap between chips in the transverse plane, the edge of elongated pixels on each chip, and the slight discrepancy in z between the front-facing and back-facing chips.

Figure 5.12 shows the results of the radial fit, which produces the formula $H = 1.104r^{-1.655}$, with a $\chi^2/\text{ndf} = 64.9$. One possible reason for the large χ^2 in this fit, aside from the slightly inaccurate arrangement of the sensors, is that the full radial range of the VELO module has been fitted over, whereas the square geometry of the module means that there is only a full radial acceptance within the range $7.2\,\text{mm} < r < 33.2\,\text{mm}$. When the fit is localised within this region, the minimised parameters are $H = 3.112r^{-1.670}$, with a somewhat reduced $\chi^2/\text{ndf} = 9.6$.

Type	MinBias
Gauss	v53r0
DDDB	upgrade/dddb-20171126
CONDDB	upgrade/sim-20171127-vc-md100
ν	7.6
Number of events	924 132 non-triggered

Table 5.1

Details of simulated Monte Carlo data used to approximate the expected distribution of hits in each VeloPix sensor

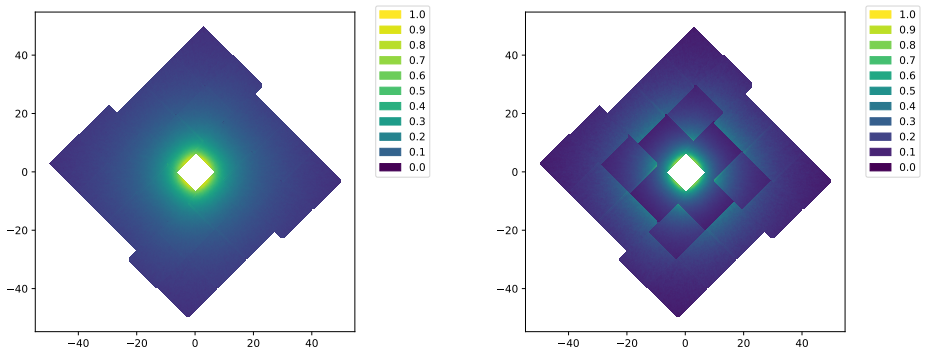


Fig. 5.11

Left: Heatmap of the number of hits recieved by each pixel of a VeloPix module within a Monte Carlo simulation dataset, as a proportion of the highest-occupancy pixel in the module. Right: The same plot, but with each pixel normalised by the number of output links available to its respective ASIC (see figure 5.4). The brightly-coloured seams along some edges of the ASICs are due to the column of wider pixels present at the edges.

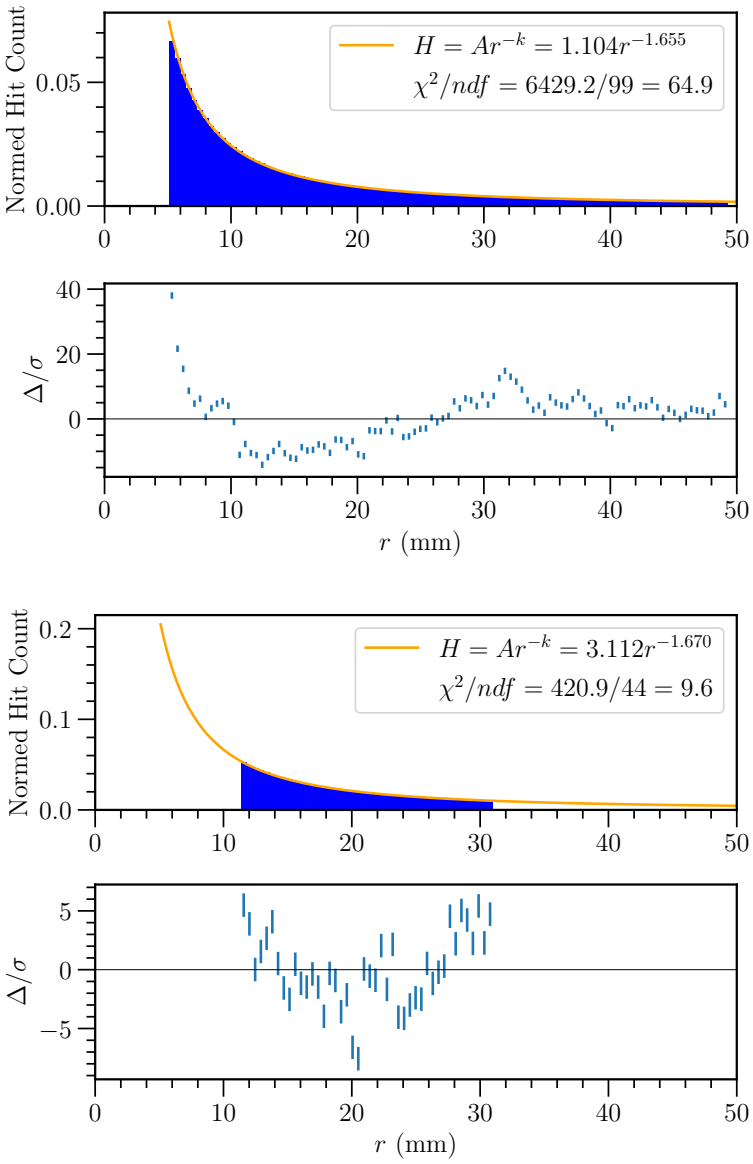


Fig. 5.12

Top subplots: Normalised histogram of the Monte Carlo VeloPix events (blue bars, see table 5.1) and the corresponding least-squares fit (orange). Bottom subplots: Pull chart of the discrepancy of the fit in units of standard deviations of each bin. In the top plot the radial fluence is fitted over the entire radial range, whereas in the bottom it is only fitted in the range of r that gives full radial acceptance (given the square shape of the sensors).

5.7 | Test Pulse Analysis

In order to handle the expected fluence of particles through the detector over run 3 (based on simulations of event occupancy and hit clustering), the upgrade VELO is designed in order to accommodate an output of 160×10^6 SPP/s for each active link on each VeloPix chip. This number is the product of the fact that the GWT serialiser links can transmit data frames at a rate of 40 MHz, with 4 SPPs in a dataframe. It is critical to test the physical VELO prior to production to ensure that the entire system performs as expected under high bandwidth, and does not succumb to congestion losses or other unexpected effects. A test pulse analysis was performed to probe the VeloPix system for these issues.

The analogue frontend of each pixel contains a switch controlled by a writeable bit, which when closed allows a test pulse signal to enter via the same input as the sensor pad. For a test pulse run, a 256×256 binary hit pattern is uploaded to the VeloPix, which sets this bit on each pixel. The pattern is intended to be processed by the ASIC at some desired frequency. This is achieved by sending a particular number of pulses with a specified number of clocks cycles "on" and "off" (typically 1 on and $n - 1$ off for a pulse cycle length of n clock cycles). This is by using a global input to the digital front end that determines whether to block the input front the analogue pixel electronics. The on/off test pulse signal is sent to the chip via TFC and the resulting response is then configured to be read out from the GWT bypass and saved to disk as binary data, which can then be decoded and reconstructed by the decoder software.

5.7.1 | Bandwidth saturation testing

An analysis was performed of the response of the VeloPix ASIC under several input patterns and rates. For multiple input patterns, it was found that when running under the maximum rated bandwidth of 160×10^6 SPP/s/link, there were no dropped or congested SPPs (all timestamps were correctly reconstructed given known input pattern). Increasing the hit pattern occupancy and/or pulse rate to exceed this bandwidth threshold meant that SPPs would become congested in the ASIC's buffers and eventually dropped.

Figure 5.13 illustrates how bandwidth congestion delays the middle columns initially, before propagating to the outer columns. Likewise, when using a fully-occupied top row of pixels as an input pattern, SPPs are lost due to

buffer congestion from the horizontal centre first, with the loss propagating outwards on both sides over time.

5.7.2 | Automatic Bandwidth Testing

A decoder software module and executable has been written to perform an analysis on test pulse data. Given an input hitmap file, and a binary GWT bypass output file, the module can reconstruct the hits whose data was written out through the GWT, and compare the output hits to the corresponding input pattern to detect dropped SPPs. The module can also determine the delay for each SPP between striking the sensor and being read out of the GWT link, to determine congestion levels in the ASIC. This test can be run for multiple input patterns and rates to form a comprehensive automatic bandwidth test, once the OPB and DAQ control software is able to be scriptable so that test pulses can be run programmatically. Such a test could be run on-site at the LHCb pit during technical stops to assess the functionality of production ASICs.

5.7.3 | MC analysis

The column-bus architecture and arbitration rules of the VeloPix chips means that their behaviour is dependent not simply on the number of input hits in each bunch crossing, but also the distribution of the hits (in time and over the area of the sensor). In an attempt to more accurately simulate the conditions that the ASICs will experience in the detector, a Monte Carlo sample of events was used to produce this distribution.

As the TFC test pulse mechanism is limited to a fixed hit pattern sent at regular intervals, the time and space distribution of the Monte Carlo data was converted to an input pattern by randomly sampling a histogram of the superpixels of the sensor over the full range of the MC dataset. This was done for all 12 of the chips that comprise a single VELO module.

The Monte Carlo dataset described in table 5.1 was used as the underlying distribution of hits in the VeloPix sensor from which the sample distributions were randomly selected. The technique settled on for producing the input hitmap from the MC pixel histogram is described in detail in appendix B.5. Three hitmaps were produced, all targeting the rated $4 \times 160 \times 10^6$ SPP/s of a chip with 4 serialiser links, but at 3 different levels of occupancy and pulse period. With 2 orbits (2×3564 clock cycles) of test pulse data, the lower-occupancy

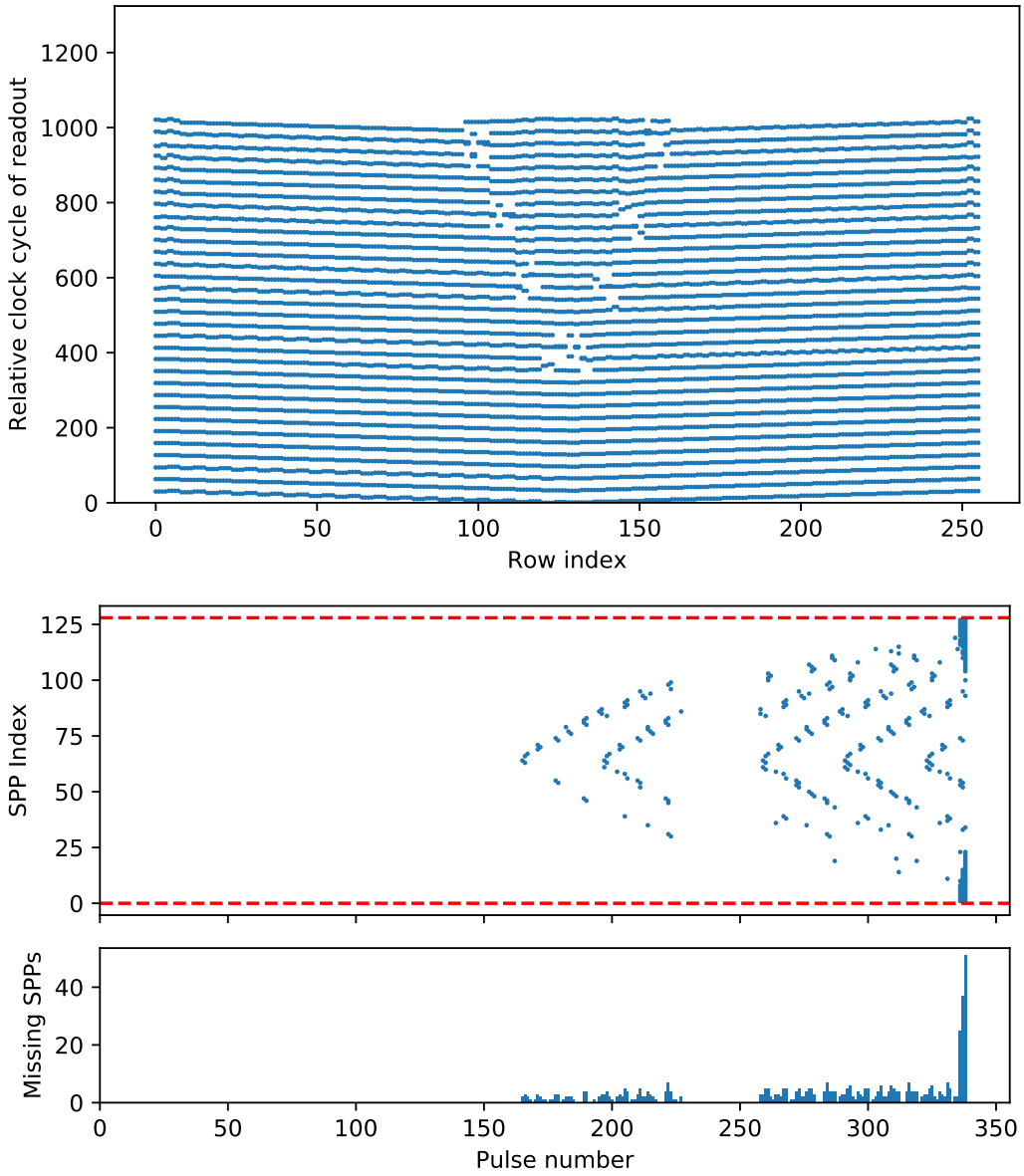


Fig. 5.13

Results of a test pulse run of the entire top row of pixels, above maximum rated bandwidth. (Top) The relative timestamps of hits as they leave the GWT. (Middle) The expanding triangular waves of lost hits shows the congestion of SPPs starting from the centre. (Bottom) The index of SPPs missing from the GWT output, in terms of the pulse they were sent as. The gap in lost hits from pulses $\sim 230 - 260$ is due to the behaviour of the TFC system. Pulses are sent at a fixed interval, except there is a delay of 31 clock cycles at the beginning of each new orbit, meaning that the pulse sent after 3564 BCIDs will have a longer cycle length, meaning the ASIC buffers partially clear before congestion increases again.

Advacam

- Receive ASICs from manufacturer
- Bump bonding
- Bump deposition inspection
- Bow measurements
- IV scan ($0 \rightarrow 300$V)

CERN

- Receive bump-bonded tiles from Advacam
- Powerup and register tests
- Current test and equalisation analysis
- Source scans (Sr90)
- IV scan ($0 \rightarrow 1000$V)
- Categorise tiles by quality rating

Fig. 5.14

Summary of the assembly and QA pipeline of the VeloPix triplet tiles

input hitmap displayed no loss of data, but the higher-occupancy patterns did lose hits, likely to be from congestion within the EoC fabric overflowing the FIFO buffers.

5.8 | VeloPix Tile Testing

In order to guarantee the working order of the VeloPix sensors and ASICs, a comprehensive series of tests are performed, some being preliminary quality assurance tests performed by the part suppliers, and other more stringent tests being performed at CERN.

The superpixel ASICs that each make up one third of a VeloPix triplet tile were manufactured by semiconductor vendor TSMC. The batches of ASICs that pass TSMC internal quality control were sent to semiconductor sensor company Advacam, where they were bump-bonded in threes to each tile of the sensor material. Advacam performed inspections of the bump deposition, and measurements of the bowing of the tile, as well as IV scans in the range of $0 \rightarrow 300$ V. The bump-bonded triplet tiles were then dispatched to CERN.

After arriving at CERN, the tiles were tested for correct operation on powerup, and checked for current leakage under a 140 V reverse-bias needle. An equalisation test was also run. The tiles were then exposed to a ^{90}Sr source to test the quality of the bump bonds for each pixel. Finally, the tiles were scanned under

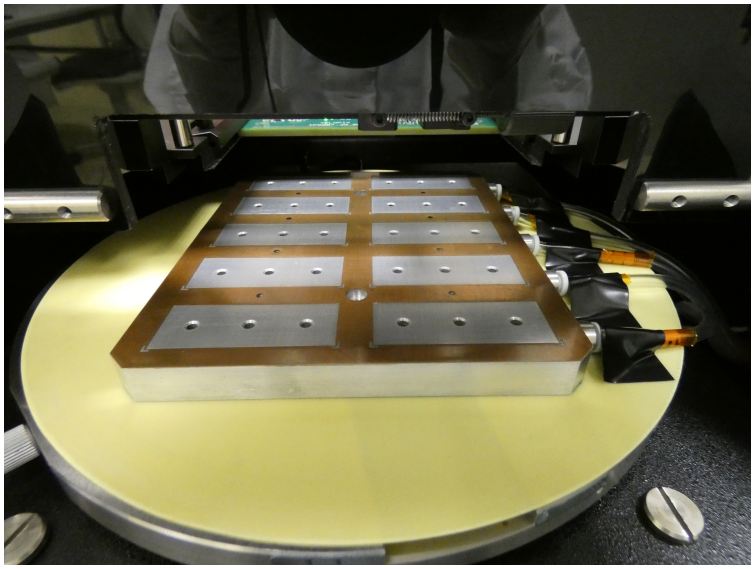


Fig. 5.15

The "jig" used to test VeloPix tiles for quality assurance.

a range of $0 \rightarrow 1000\,\mathrm{V}$ to produce IV curves. The set of information gathered from the suite of tests was then used to assign each tile a performance grade from A to F, with A and B being deemed production-ready, and suitable to be installed into VELO upgrade modules.

The procedure for the powerup and register test, equalisation analysis, and source scan is described in the following section.

5.8.1 | Test Procedure

The tiles were tested with the use of a probe station with a vacuum chuck that can be moved in a limited way via software. The tiles are run through the set of tests in batches of up to 10. Each batch is loaded onto a 2 tile by 5 tile jig which is itself held in place via vacuum to the probe station chuck (figure 5.15). The jig consists of an aluminium block with an overlaid copper stencil that has 10 tile-shaped rectangles etched out. Each tile inlay contains 3 vacuum holes, each lying under the centre of each ASIC chip. All tiles in a batch are then aligned to the same corner of their respective inlays, so as to reduce alignment discrepancy between successive rows and columns of tiles.

Hit data is read out from the VeloPix ASIC via a custom probe card connected to a SPIDR readout board (an existing readout system developed for the Timepix3 chip [117]), which itself connects via ethernet to a local PC

running Linux, which is used to control the probe station as well as receive data.

Before running the tests, the jig must be rotationally aligned, so that for each ASIC being tested, the row of probe card wires do not become misaligned from the conductive pads from one side of the ASIC to the other. This can be done semi-automatically by marking in software two stencil markers on the jig (one on each side) and instructing the probe station software to align these points. Next, all tiles on the jig must be checked for adequate horizontal and vertical alignment (within the width and height of a single ASIC pad. This is necessary because the probe station software does not allow for multiple locations on the chuck to be saved and restored. Rather, a single "home" point on the chuck is defined (the outer ASIC of one of the corner tiles), and the other tiles are reached through successive additions of a uniform height and width displacement. Likewise, the different ASICs of a single tile are reached via additions of a different, smaller width displacement.

To aid with alignment, each tile contains a small, cross-shaped reference marker that can be seen under the probe station microscope. For all tiles to be suitably aligned such that all probe card wires fall inside the bonding pads, a given reference point on the screen must fall inside the reference marker region of all tiles when moved by the various displacements. If not, then the tiles must be readjusted within the jig for a better alignment.

When all tiles are aligned to one another, the position of the first tile is then adjusted so that the probe card wires are aligned to the ASIC pads. The ASIC is moved to the approximate location under the probe card wires, and the chuck is gradually raised in increasingly smaller increments, while changing the focal position of the microscope between the pads and the wires. After aligning further, the chuck is raised in very small increments until the wires have left a visible contact mark on the pads. Next, a $-140\,\mathrm{V}$ bias needle is placed over the sensor probe region. The sensor region of the tile is brought into focus on the probe station, and the high-voltage needle is lowered until it comes into focus, and finally makes visible contact with the sensor material (the needle head will move laterally slightly as it is meets the sensor).

Powerup and Register Test

The short powerup and register ASIC test is then run, with the probe station software automatically iterating over each ASIC by raising and lowering the

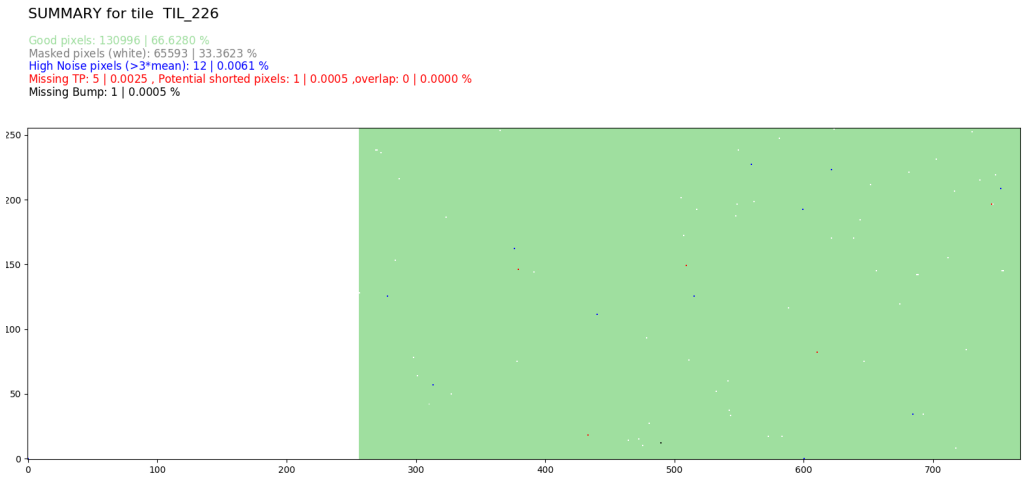


Fig. 5.16

An example of the results of a powerup and register test. This tile showed two healthy responses (middle, right) but one failed response (left).

chuck to bring a new chip into contact with the probe card wires and the probe needle. This test performs preliminary checks that the chip responds to being under voltage and the registers in the end-of-column fabric can be read out successfully. Chuck height can be increased to improve contact strength if tests show connection problems to the ASICs.

Figure 5.16 shows the result of the powerup and register test on a tile with a single faulty ASIC, which caused a lack of response from the left third of the tile, rendering the tile unusable. The responses from the middle and right ASICs together indicate 130996 "good" pixels of a possible 131072, a rate of 99.94%.

Equalisation Scan

After the powerup and current leakage test, an equalisation scan is performed. The completed scan will provide equalisation distribution data for each pixel of each ASIC.

As discussed in section 5.3.1, the binary pixel response to analogue line signal is dictated by a discriminator which compares the amplitude with a digital-to-analogue input from the combination of a 14-bit global and 4-bit local threshold. Since manufacturing tolerances and the unique positions inside the ASIC mean

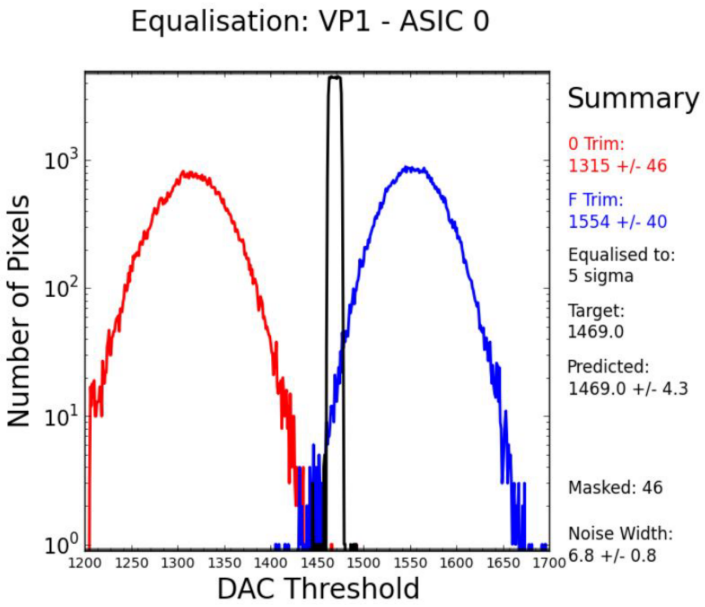


Fig. 5.17

An example of an equalisation process for all pixels in an ASIC chip [118]. The red histogram shows the distribution of the DAC threshold level for all pixels, with the pixels having a local DAC value of `00`. Likewise, the blue histogram shows the distribution for a local DAC value of `0F`. The black histogram shows the expected distribution when all local DAC trim values are set such that the variance of the distribution is minimised, with the threshold equalised to 1469 ± 4.3.

that pixels will have varying noise levels, the 4-bit per-pixel noise threshold is used to modify each pixel's threshold such that the response of all pixels to the same conditions are equalised. The equalisation scan works by measuring the responses of each pixel in parallel as the global threshold is varied. This produces a distribution of the number of hits seen as a function of the global threshold, which peaks when the threshold is equal to the mean noise voltage. This process is done twice, one with the local threshold at `00` and again with the local threshold at `0F`, to produce two distributions. The peaks of these two distributions are then compared as two datapoints of the effective noise level at the minimum and maximum values of the 4-bit threshold. These two values are then linearly interpolated, and the 4-bit thresholds of each pixel are set to equalise the effective noise level across all pixels in the chip.

Source Scan

When the equalisation analysis is completed, the batch of tiles undergo a source scan. For this, a ^{90}Sr beta source is placed over the probe region, and data is gathered of the particles striking the sensor while each ASIC is under high voltage. From this data, 2D histograms are produced for each ASIC, showing the total fluence received by each pixel in the ASIC. This aids in identifying incorrectly bump-bonded pixels in the case of low zero hit counts on the pixel level, as well as identifying wire bonding connection and other issues in the case of low counts over the whole ASIC or triplet tile.

Figure 5.18 shows the response of the same example tile from figure 5.16, this time from the source scan. Again, the faulty left ASIC gives no response, while the other two have a present distribution from the beta source. Note that the high occupancies on the edges of the two healthy responses are due to the extended edge pixels sizes, used to ensure full detector coverage across the overlap between sensors.

After all analysis steps are completed, the resulting data for each tile is uploaded to an online database system, along with the tiles' identification information and grade. From an analysis of a batch of 42 triplet tiles performed in June 2019 (including powerup and register test, equalisation, and source scan), 30 tiles were found to be grade A, 8 grade B, and 4 not production-worthy.

A silicon pixel redesign of the LHCb vertex locator has been described. A test pulse analysis of the VeloPix GWT bypass system has been performed, involving the development of a piece of software to decode, reorder, translate, and automatically analyse binary GWT and GWT bypass data frames. The test pulse analysis found that the practical bandwidth and congestion limits of the VeloPix ASICs and readout system are congruent with the limits specified in the technical design report [102]. The quality assurance process for the VeloPix sensor triplets has also been described.

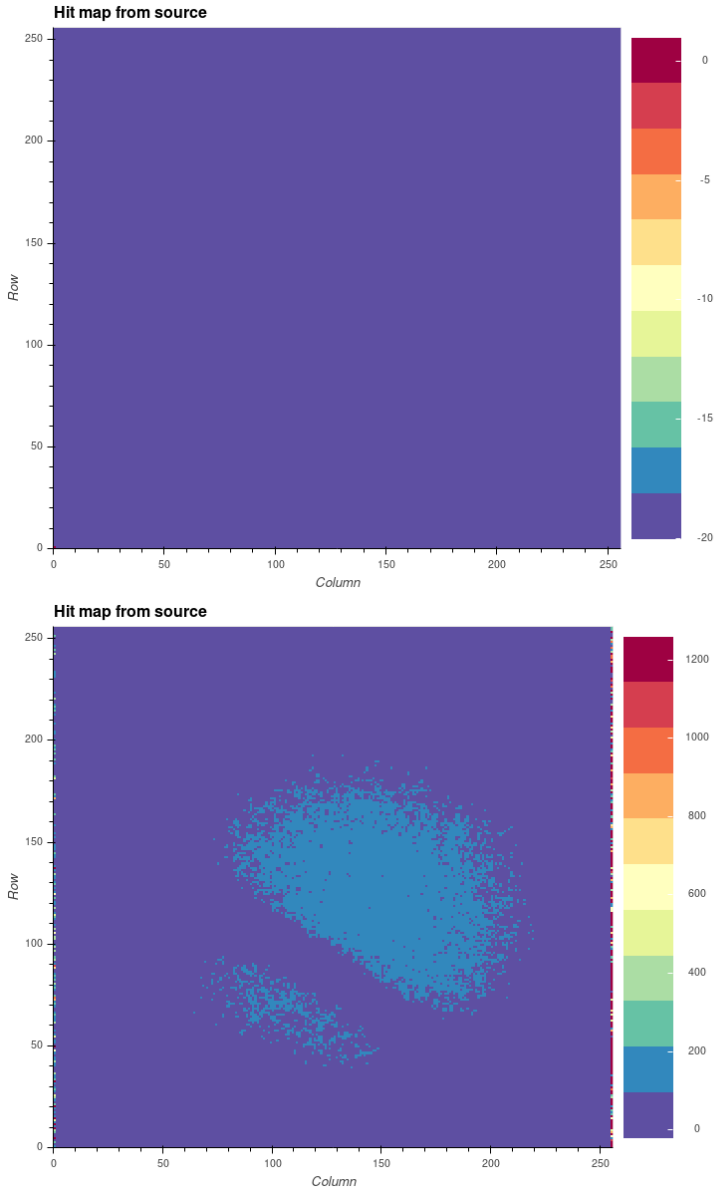


Fig. 5.18

An example of the results of a source scan on an ASIC-triplet tile. This tile showed two healthy responses (of which one is shown in the lower plot) but one failed response (left), shown by the zero occupancy across the whole ASIC. The apparent low occupancy of the healthy ASIC is an artefact of normalisation, due to the outer edge of extended pixels on each chip receiving proportionately more hits. The line-shaped region of lower occupancy across the middle of the ASIC is due to the shadow of the needle between the source and the sensor.

Chapter SIX

$B^0 \to K_1(1270)ll$ Sensitivity Study

In this chapter a sensitivity study is performed over the decay $B^0 \to K_1(1270)ll$

6.1 Decays Used

A set of 4 decays are assumed in this study, with 2 sets of 2 options. The lepton-antilepton pair may be either electrons or muons; and the lepton pair may be produced directly or via a J/ψ resonance. This gives the following decay modes:

$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)e^+e^-$ ..... "electron, non-resonant"

$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)\mu^+\mu^-$ ..... "muon, non-resonant"

$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to e^+e^-)$ ..... "electron, resonant"

$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$ ..... "muon, resonant"

6.2 Motivation

There are a number of reasons why $B^0 \to K_1(1270)ll$ decays are desirable to detect in run 3 of LHCb. Firstly, the branching fraction of $B^0 \to (K_1(1270) \to$

$J^P = 1^-$	$J^P = 1^+$	ΔM, MeV
$\rho(770)$	$a_1(1260)$	490
$\omega(782)$	$f_1(1285)$	503
$\phi(1020)$	$f_1(1420)$	400
$K^*(895)$	$K_1(1270)$	375

Table 6.1

Table of the mass difference of several meson parity doubles, with $J^P = 1^-$ and 1^+. See section 2.5.2.

$K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$ is not accurately known [20], and has the opportunity to reduce the uncertainty with greater statistics. The branching fraction of the non-resonant version, $B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)\mu^+\mu^-$, is not yet measured.

As discussed in section 2.5.2, a measurement of the K_1 branching fraction is experimentally significant, as the measurement of corresponding vector and axial-vector meson amplitudes significantly reduces the $V - A$ long-distance contributions to right-handed amplitudes and provides a much cleaner test of beyond-Standard-Model right-handed currents in the weak interaction [65]. Vector-axial meson pair candidates are listed in table 6.1, along with their mass difference. Meson pairs that are closer in mass are parity-degenerate to a better approximation, and thus may result in a more exact cancellation of the long-distance $V - A$ contributions to $V + A$ amplitudes. The $K^*, K_1(1270)$ pair have the smallest mass difference width of 375 MeV. There exists a mixing angle θ_{K_1} between $K_1(1270)$ and $K_1(1400)$, which is estimated to be less than 45° [119].

These four decay chains would allow for an accurate measurement of the $\mu\mu/ee$ ratio in the form of the double-ratio $R_{K_1(1270)}$:

$$\frac{\mathcal{B}(B^0 \to K_1(1270)\mu^+\mu^-)}{\mathcal{B}(B^0 \to K_1(1270)J/\psi(\to \mu^+\mu^-))} \Big/ \frac{\mathcal{B}(B^0 \to K_1(1270)e^+e^-)}{\mathcal{B}(B^0 \to K_1(1270)J/\psi(\to e^+e^-))}$$

Such a double ratio has already been calculated for the K^* (rather than K_1) as a test of Lepton Universality, and can significantly reduce systematic compared to the single ratio, as most aspects of the physics and reconstruction of the event are shared between the two decay modes [62]. Note that the resonant modes,

unlike the FCNC non-resonant modes, have tree-level contributions [120], and as such are not highly suppressed in the Standard Model.

6.3 | Data and Software

The sensitivity study was performed on a simulated dataset of Monte Carlo events in expected Run-3 detector conditions. Four sets were used, each of which were required to contain at least one of the respective four decays above in every event. Particles were decayed according to a relativistic Breit-Wigner line shape, as demonstrated in a fit to the mass distribution of $K_1(1270)$ in 2 000 generated events (figure C.1).

6.3.1 | Detector Conditions

The events in this study were simulated using expected Run 3 conditions. The upgrade detector was simulated with the DDDB tag dddb-20210617 (see section 3.4.2), and the particles were generated using a simulated beam energy of 13 TeV.

For each of the four decays in the study, 20, 000 signal events were fully simulated: 10, 000 with the CONDDB tag sim-20210617-vc-md100 (conditions file submitted on 2021-06-17, VELO closed, magnet polarity down) and 10, 000 with the tag sim-20210617-vc-mu100 (magnet polarity up).

To save simulation and analysis computation time, the events were generated with a generator-level cut of "Daughters in LHCb", meaning generated events were only simulated further if all the charged decay products were produced at an angle of $10\,\text{mrad} < \theta < 400\,\text{mrad}$ with respect to the beam axis, and all neutral projects were produced at $5\,\text{mrad} < \theta < 400\,\text{mrad}$.

The simulation framework is structured such that all subdetector components are simulated separately, and simulation of different combinations may be enabled in order to expend the minimum amount of computation time required to simulate the physics in the desired decay. This simulation included all components of the upgrade LHCb detector: VELO, Upstream Tracker (UT), Fibre Tracker (FT or SciFi), Rich1, Rich2, ECAL, HCAL, Muon, and Magnet (see chapter 3).

The events were simulated with a mean number of proton-proton interactions per bunch crossing of $\nu = 7.6$, in line with the expected run 3 detector condi-

tions (compared to $\nu = 1.6$ for run 2 simulations).

Spillover

In collider-based particle detectors that process the results of collision events at very short intervals, such as LHCb and other detectors at the LHC, one complicating factor for the correct reconstruction of events is *spillover*. This is phenomenon whereby interactions of particles with the detector in an event have not fully settled by the time of the next event, and so hits or tracks may be digitised and read out in a later event than the one where the particle passed through the detector. At the LHC (for both run 2 and run 3), the time between bunch crossings is 25 ns. A run-2 analysis of the Silicon Tracker subdetectors found that spillover contributed to a measurable increase in the overall rate of ghost tracks in the reconstruction [121].

For run 3, the upgraded silicon trackers, including the VELO, Upstream Tracker, and Fibre Tracker, have been designed with the goal of reducing the sensitivity to spillover, with sharper peaks in response to interacting charged particles. As such, the effects of spillover are expected to account for only a very small proportion of tracks. For example, the silicon pixel sensors in the upgrade VELO have been designed to have a time resolution reliably less than than the 25 ns bunch crossing period, and can be expected to have a negligible number of spillover tracks [102]. Because the inclusion of spillover increases overall simulation time by a large factor, the decision was made to simulate the events without spillover, in order to maximise statistics from the available computing time.

6.3.2 | LHCb software stack

Proton-proton collision generation was performed by *Pythia* [53, 54], and particles were decayed by *EvtGen* [81]. QED corrections and final-state radiation was performed by *PHOTOS* [122]. Particles were propagated through the simulated LHCb detector materials by *Geant4* [82–84]. This software was run as part of the LHCb simulation framework *Gauss* [80].

The simulated events were then digitised, preserving the MC truth PID and kinematic information, using the *Boole* [85] software framework. Events were filtered through the HLT1 and HLT2 using the software trigger *Moore*. More specific details about software versions are outlined in appendix C.1.

6.4 | Total Event Estimation

The total number of expected events N for the non-resonant muon decay in run 3 is given by:

$$N = L_{int} \times \sigma_b \times f_d \times \mathcal{B}(B^0 \to (K_1(1270) \to K^0_S \pi^+ \pi^-)\mu^+ \mu^-)$$

The integrated luminosity, L_{int}, of the LHCb experiment throughout run 3 is expected to be $50 fb^{-1}$.

At 13 TeV (the expected operating centre-of-mass energy of the LHC for run 3) the b-quark cross section σ_b is $144 \pm 1 \pm 21\,\mu$b [123].

6.4.1 | Fragmentation Fraction

The B_d fragmentation fraction f_d (see section 2.3.3) is the fraction of b quarks produced in the collision event that hadronise with a d quark to form a B^0 meson. The total b quark fragmentation comprises of the fragmentation fractions corresponding with the hadronisation of the b quark with u, d, c, and s quarks to produce B_u, B_d, B_c, and B_s mesons (f_u, f_d, f_c, and f_s respectively), as well to produce b baryons (f_{baryon}). That is, $f_u + f_d + f_c + f_s + f_{\text{baryon}} = 1$.

Studies of the ratios of b quark fragmentation fractions as $\sqrt{s} = 7$ TeV have shown that f_c is less than 1% [57], so as an approximation, f_c may be omitted from the above equation.

More recent studies at the LHCb experiment have measured fragmentation fraction ratios at energies of $\sqrt{s} = 13$ TeV [124]. Measurements of the fragmentation fractions f_s and $f_{\Lambda^0_b}$ (fragmentation fraction to form Λ^0_b baryons) were found by measuring the yield of $\bar{B}^0_s$ semileptonic decays and Λ^0_b semileptonic decays respectively. Both fractions were normalised to the combined yields of B^- (f_u) and $\bar{B}^0$ (f_d). Averaging over the hadronic transverse momentum range $4\,\text{GeV} < p_T(H_b) < 25\,\text{GeV}$ and the LHCb pseudorapidity range $2 < \eta < 5$ gives the values:

$$\frac{f_s}{f_u + f_d} = 0.122 \pm 0.006$$

$$\frac{f_{\Lambda^0_b}}{f_u + f_d} = 0.259 \pm 0.018$$

[124]

Assuming that, due to isospin symmetry, $f_u \approx f_d$, and given $f_{\text{baryon}} \approx f_{\Lambda^0_b}$, then

the above fraction equation becomes:

$$2f_d + f_s + f_{\Lambda_b^0} \approx 1$$

Substituting the above values of f_s and $f_{\Lambda_b^0}$ gives the approximation:

$$f_d \approx 0.36$$

at 13 TeV and averaged over the above range of p_T and η.

6.4.2 | Branching Fraction

Known branching fractions quoted in this section are cited from the Particle Data Group [20].

As the non-resonant branching fraction $\mathcal{B}(B^0 \to K_1(1270)\mu^+\mu^-)$ is not precisely known, it may be approximated in terms of the resonant decay and the K^{*0} parity partner decay, like so:

$$\begin{aligned}\mathcal{B}(B^0 \to K_1(1270)\mu^+\mu^-) &\approx \mathcal{B}(B^0 \to K^{*0}\mu^+\mu^-) \times \tfrac{\mathcal{B}(B^0 \to J/\psi K_1(1270))}{\mathcal{B}(B^0 \to J/\psi K^{*0})} \\ &= (9.4 \pm 0.5) \times 10^{-7} \times (1.3 \pm 0.5) \times 10^{-3} / ((1.27 \pm 0.05) \times 10^{-3}) \\ &= (9.6 \pm 3.7) \times 10^{-7}\end{aligned}$$

The branching fraction for the full decay is then given by:

$$\begin{aligned}&\mathcal{B}(B^0 \to (K_1(1270) \to K_S^0\pi^+\pi^-)\mu^+\mu^-) \\ &\quad = \mathcal{B}(B^0 \to K_1(1270)\mu^+\mu^-) \times \mathcal{B}(K_1(1270) \to K^{*0}\pi^+\pi^-) \\ &\quad = (9.6 \pm 3.7) \times 10^{-7} \times \tfrac{(42 \pm 6)}{2}\% = (2.0 \pm 0.8) \times 10^{-7}\end{aligned}$$

Where the 2 in the denominator accounts for the fact that the referenced value of 42% applies to the decay of $K_1(1270)$ to a pair of pions, which by symmetry is split equally between charged and uncharged pairs (only the charged pairs are searched for in this study).

The total number of expected events for run 3 may then be computed as:

$$\begin{aligned}N &= L_{int} \times \sigma_b \times f_d \times \mathcal{B}(B^0 \to (K_1(1270) \to K_S^0\pi^+\pi^-)\mu^+\mu^-) \\ &= 50 \times (10^{-15}\,\mathrm{b})^{-1} \times 144 \times 10^{-6}\,\mathrm{b} \times 0.36 \times (2.0 \pm 0.8) \times 10^{-7} \\ &= (5.2 \pm 2.0) \times 10^5\,\mathrm{evts} = (1.0 \pm 0.4) \times 10^4\,\mathrm{evts/fb^{-1}}\end{aligned}$$

Using the following known branching fractions:

Decay	$\mathcal{B}$	Estimated Run 3 Events
ee, non-resonant	$(2.2 \pm 1.0) \times 10^{-}7,$	$(5.7 \pm 2.6) \times 10^{5}$ $(1.1 \times 10^{4}/\,\mathrm{fb}^{-1})$
$\mu\mu$, non-resonant	$(2.0 \pm 0.8) \times 10^{-}7,$	$(5.2 \pm 2.0) \times 10^{5}$ $(1.0 \times 10^{4}/\,\mathrm{fb}^{-1})$
ee, resonant	$(1.6 \pm 0.7) \times 10^{-}5,$	$(4.2 \pm 1.8) \times 10^{7}$ $(8.5 \times 10^{5}/\,\mathrm{fb}^{-1})$
$\mu\mu$, resonant	$(1.6 \pm 0.7) \times 10^{-}5,$	$(4.2 \pm 1.8) \times 10^{7}$ $(8.4 \times 10^{5}/\,\mathrm{fb}^{-1})$

Table 6.2

Table of the derived branching fractions and estimated total run 3 events ($50\,\mathrm{fb}^{-1}$) of the four $B^0 \to K_1(1270)ll$ decays used in this study.

$$\mathcal{B}(B^0 \to K^{*0}e^+e^-) = 1.03^{+0.19}_{-0.17} \times 10^{-6}$$

$$\mathcal{B}(J/\psi \to e^+e^-) = (5.971 \pm 0.032)\%$$

$$\mathcal{B}(J/\psi \to \mu^+\mu^-) = (5.961 \pm 0.033)\%,$$

this derivation is repeated for the other three decays used in this study, to produce the data in table 6.2.

6.4.3 | K_1-Daughter Kinematics Estimation

As well as the requirement that decay daughter particles travel through the geometrical acceptance region of the LHCb detector, the kinematic variables of the final state particles also determines whether the event will be correctly reconstructed. Particles with an insufficient transverse momentum p_T will either not cross the detector's inner pseudorapidity threshold of $\eta = 5$, or will be subsumed within a large number of other soft particles, preventing correct reconstruction. For the latter reason, cuts on minimum p and p_T are often commonly imposed in various stages of the LHCb software trigger lines and analysis pipelines. As a course estimate of the reconstructibility, 5000 $B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$ events were simulated at the 4-vector level, and cuts of $p > 1000\,\mathrm{MeV}, p_T > 500\,\mathrm{MeV}$ were imposed on the pion daughters of the $K_1(1270)$ (a common pair of cuts for pion candidates in the LHCb trigger framework). The resulting yields are shown in table 6.3, giving a combined yield of 30.6%.

	$p > 1000\,\text{MeV}$	$p_T > 500\,\text{MeV}$	Both
π^+	95.8%	55.9%	55.9%
π^-	95.6%	55.5%	55.5%
π^+ and π^-	91.8%	30.6%	30.6%

Table 6.3

Results of 4-vector cuts of $p > 1000\,\text{MeV}, p_T > 500\,\text{MeV}$ on the pion K_1-daughters over 5000 $B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$ events.

6.5 | Estimation of Overall Reconstructibility

LHCb simulated particles may be classified into groups based on different criteria, designed to indicate whether or not the particle's track has the possibility of being reconstructed in non-simulation-aware software from the hits it left in different components of the detector. This is known as the track's *reconstructibility*. While other requirements (such as cuts on the particle's P or $\beta\gamma$) may be used in the criteria, here the default requirements have been used, which are solely geometrical. As such the number of reconstructible particles given may be thought of as a upper limit of the number of correctly reconstructed particle tracks, as it does not account for kinematics of the event or the effects of high occupancy on the pattern recognition algorithms.

The first criteria for reconstructibility is whether the particle travelled through the detector's acceptance region. Here, the acceptance is defined as whether the particle travelled through any of the basic volumetric regions corresponding to the upgrade VELO, UT, and FT subdetectors (historically these were the VELO, TT, and T1-T3 trackers). If not, the particle is classified as *OutsideAcceptance*. Particles passing this acceptance cut are then tested by a Monte Carlo selector which evaluates the validity of the simulation (such as whether the particle either has an MC parent particle or originated from a primary vertex) and performs optional kinematic selections (not used here). All particles used in this study passed the MC selector. Particles are then evaluated for their reconstructibility based on different criteria depending on whether they are charged. Since all particles in this decay are reconstructed from charged tracks, only the charged criteria will be outlined.

Each charged track is then separately evaluated for reconstructibility in each of the three aforementioned subdetectors. To be deemed reconstructible in a particle subdetector, a simulated track portion must have deposited a sufficient

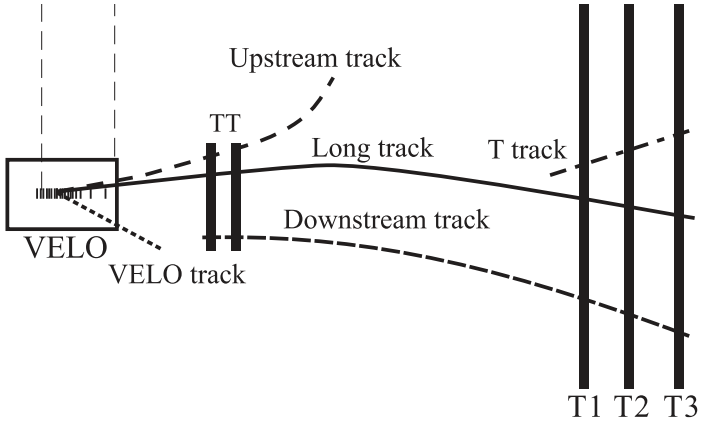


Fig. 6.1

A visualisation of the classification of tracks in the run 2 LHCb detector based on which subdetectors the track was reconstructed in [58]. The upgrade detector is analogous, but the TT is now UT, and T1-3 are now FT (see chapter 3).

number of hits. The overall track is then classified based on the following combinations (highest priority first):

Reconstructibility	Description
0 =OutsideAcceptance	Does not travel through any region
1 =NotReconstructible	Not reconstructible in any region
2 =ChargedLong	Reconstructible in VELO and FT (formerly T)
3 =ChargedDownstream	Reconstructible in UT (formerly TT) and FT
4 =ChargedUpstream	Reconstructible in VELO and UT
5 =ChargedTtrack	Reconstructible in FT
6 =ChargedVelo	Reconstructible in VELO

Similarly to this classification of simulated particles, reconstructed charged LHCb tracks are classified by the same combinations into *Long*, *Upstream*, *Downstream*, *VELO*, and *Ttrack*, based on which subdetectors reconstructed a track that was combined into the full track. Figure 6.1 shows this classification visually for the detector as it was in run 1 and run2. The only changes to this scheme for run 3 is the replacement of the TT with UT and the replacement of T1-T3 with SciFi.

Due to the relatively long lifetime of $\pi^{\pm}$ and $\mu^{\pm}$, as well as the relatively short lifetime of the B^0 and K_1, the dilepton B-daughters and the charged pion K_1 daughters all most commonly form *Long*-reconstructible tracks in the simulated detector. A significant minority are *Upstream* and *VELO*, and an very small

number are *OutsideAcceptance*, *NotReconstructible*, *Downstream* and *Ttrack*. Therefore, to reduce the background rate, K_1 particles are only reconstructed from pion candidates with long tracks.

The lifetime of 8.95×10^{-11} s of the K^0_S means that it is uncertain where in the detector, if at all, the $(B^0 \to K_1 \to)K^0_S$ will decay, and thus where the origin vertex of the charged daughter pions will be. Consequently, the reconstructibility categories are more evenly distributed (with the exception of *ChargedTtrack*). The reconstructibility categories of the 4 kinds of final-state particles in the decay are shown in figure 6.2.

From the directly-reconstructed final state particles ($\pi^+_{K^0_S}$, $\pi^-_{K^0_S}$, $\pi^-_{K_1}$, $\pi^+_{K_1}$, l^+, l^-), the indirect reconstructibilities of the parent particles in the decays were calculated, where a parent particle is reconstructible if all of its daughters are also reconstructible. That is, the Boolean reconstructibility of a parent is the logical AND of the Boolean reconstructibility of its children (therefore, a reconstructible B^0 in this case implies all particles in the decay are reconstructible). This was calculated twice, with respective reconstructibility requirements for the K^0_S-daughter pions to be *ChargedLong* and *ChargedDown*. All other final-state particles were required to be ChargedLong to be counted as reconstructible. The result, shown in figure 6.3, shows that the number of fully-reconstructible decays (the B^0 columns) are:

	Reconstructible events $(/20,000)$		
	LL	DD	LL OR DD
$B^0 \to K_1\mu\mu$	625	1183	1808 (9.0%)
$B^0 \to K_1(J/\psi \to \mu\mu)$	669	1055	1724 (8.6%)
$B^0 \to K_1 ee$	510	843	1353 (6.8%)
$B^0 \to K_1(J/\psi \to ee)$	595	905	1500 (7.5%)

where "LL OR DD" requires that the charged pion K^0_S daughters are required to both be long-long or down-down, but not long-down. The prevention of mixing reconstructibility categories greatly simplifies reconstruction efficiencies (see section 6.6.3) but does not remove many events, since the main factor differentiating long- vs down-reconstructible tracks is the location of the origin vertex, which is common to both pions.

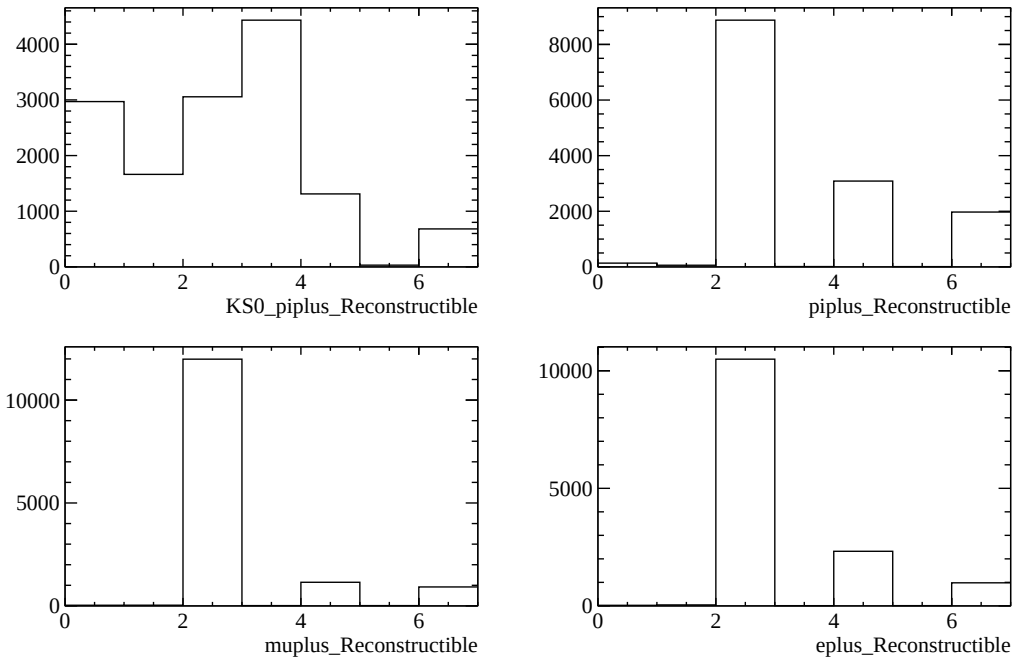


Fig. 6.2

Histograms showing the reconstructibility categories (as enumerated in table 6.5) of the $\pi^+_{K_1}$ (top-left), $\pi^+_{K^0_S}$ (top-right), e^+ (bottom-left) and μ^+ (bottom-right) particles. Each histogram is generated from $20,000$ $B^0 \to (K_1 \to K^0_S \pi^+ \pi^-)$ events, with a selection on $K^0_S \to \pi^+\pi^-$. Charge-conjugated categories were within statistical variation. Categories for e and μ are taken from non-resonant decays, but take a very similar distribution for resonant decays, with slightly more reconstructible as long tracks (see figure 6.3).

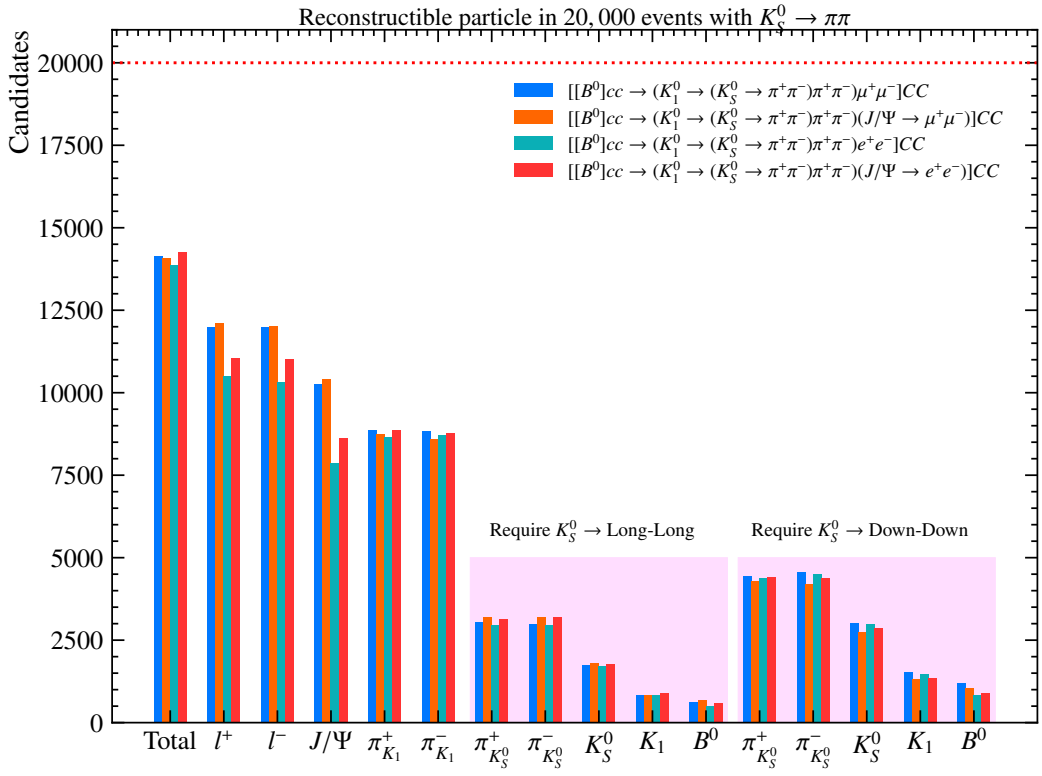


Fig. 6.3

The number of particles deemed reconstructible in their respective decay. Daughter pions of the K^0_S are deemed reconstructible if they can be reconstructed as "long" (left highlight) tracks, or "downstream" (right highlight) tracks. Parent particles are deemed reconstructible if all daughters are reconstructible. l^+ and l^- represent the counts of the dilepton pair in the respective decays. For the non-resonant decays, the J/ψ is intended to be a placeholder for a reconstructed displaced dilepton pair rather than a real J/ψ particle. The counts for "Total" are the total number of events where the simulated K^0_S decayed to $\pi^+\pi^-$. The events comprising the other counts are sampled from only these events.

6.6 | Software Trigger

6.6.1 | HLT1

The full HLT1 software trigger as it exists in Moore v53r3 (2021-12-02) was run on all events in each decay.

The Allen [86] software project running inside Moore, rather than Moore itself, is expected to be the canonical implementation of the HLT1 software trigger for run 3. However, due to the presence of occasional non-deterministic memory corruption bugs when running Allen on the simulated sets of events, the decision was made to recreate the study using Moore as the HLT1 in this study, to ensure data validity. For this reason, some HLT1 lines that have been implemented in Allen (and will likely be used during run 3) are not present in the version of the HLT1 used in this study. In particular, Allen has a trigger line dedicated to selecting $K^0_S \to \pi^+\pi^-$ decays, which may improve the overall HLT1 signal efficiency over the one quoted here.

6.6.2 | Global Event Cut

The global event cut, or GEC, is a cut in HLT1 trigger lines applied to events that have anomalously high detector occupancies, that are deemed too busy to retain. The reasoning for this is that overly busy events are more computationally expensive to reconstruct in the trigger, and due to combinatorics in combining tracks to produce vertices, this cost is super-linear. Particles in higher-occupancy events also have a higher probability of being incorrectly reconstructed, whether from unaffiliated detector hits being combined into a *ghost* track, or unrelated tracks being spuriously combined into a nonexistent parent particle, as well as many other categories of misreconstructed particles.

The GEC is applied as a simple cut on the combined number of clusters N produced in both the Fibre Tracker (FT) and Upstream Tracker (UT) by the respective clustering algorithms. For the current trigger configuration, the default cut is $N <= 9750$.

Since the trigger configuration regarding cuts on occupancy throughout all of run 3 is uncertain, the GEC has been disabled for the rest of this analysis. As shown in table C.8, for fully simulated signal events, about 66% were removed by the GEC. In this sense, this analysis is maximally optimistic, however since

busy events have a higher rate of incorrect reconstruction, this factor of signal efficiency increase is likely to represent an upper bound.

6.6.3 | HLT2

Cuts on variables of the various protoparticles in the HLT2 reconstruction were first applied very loosely and iteratively tightened after observing the distributions of reconstructed particles based on their status as true signal or misreconstructed background from other decays in the signal events. Being a three-stage decay, the cuts at each stage are relatively tight to suppress the background rate as much as possible, in order to prevent the number of misreconstructed mother particles becoming very large via successive combinatorics.

Each HLT2 line constructs a set of candidate decay trees. First the two pairs of charged pions (from the K_1 and K_s^0) and the dilepton pair are directly reconstructed as candidate tracks from the hits in the detector. The candidate particles are then successively combined until any mother B candidates remain. In both resonant and non-resonant decays, the dilepton is reconstructed as a "dummy" J/ψ protoparticle before being combined with the B mother, however in the non-resonant case this does not involve tight cuts on variables such as the true J/ψ mass.

Multiple versions of the HLT2 lines were produced, accommodating multiple orthogonal factors of the decay and reconstruction. Separate decay lines were made for decays to e^+e^- and $\mu^+\mu^-$, as well as resonant and non-resonant decays. The lines were also split based on whether the charged pion pair from the K_s^0 were reconstructed as *long-long* (LL) or *down-down* (DD) tracks (see figure 6.1), as it is beneficial in analyses to account for the trigger efficiencies separately for particles reconstructed in separate parts of the detector. When reconstructing electrons from hits in the electromagnetic calorimeters (ECAL), the candidates may be reconstructed with their associated bremsstrahlung radiation. Separate trigger lines were created to reconstruct electron-daughter decays with and without bremsstrahlung. Finally, for muon-daugher decays, "loose" versions of each HLT2 line were produced, with less stringent cuts at each stage of the reconstruction. In total, these trigger lines are listed in table 6.4.

For each stage of the reconstruction in a given HLT2 line, the existing sets of daughter candidates are selected (such as long-long-reconstructed π^+ and π^-

Hlt2_B2K1MuMu_LL_Res_Line	Hlt2_B2K1MuMu_LL_Loose_Res_Line
Hlt2_B2K1MuMu_DD_Res_Line	Hlt2_B2K1MuMu_DD_Loose_Res_Line
Hlt2_B2K1MuMu_LL_Nonres_Line	Hlt2_B2K1MuMu_LL_Loose_Nonres_Line
Hlt2_B2K1MuMu_DD_Nonres_Line	Hlt2_B2K1MuMu_DD_Loose_Nonres_Line
Hlt2_B2K1EE_nobrem_LL_Res_Line	Hlt2_B2K1EE_withbrem_LL_Res_Line
Hlt2_B2K1EE_nobrem_DD_Res_Line	Hlt2_B2K1EE_withbrem_DD_Res_Line
Hlt2_B2K1EE_nobrem_LL_Nonres_Line	Hlt2_B2K1EE_withbrem_LL_Nonres_Line
Hlt2_B2K1EE_nobrem_DD_Nonres_Line	Hlt2_B2K1EE_withbrem_DD_Nonres_Line

Table 6.4

List of the HLT2 lines used in this study. They differ on the basis of dilepton generation, resonance, K_s^0-daughter reconstruction (LL vs DD), cut stringency, and bremsstrahlung reconstruction.

candidates for the K_s^0), and may have a set of cuts applied to each daughter particle type. For all valid combinations of remaining daughter candidates, the 4-vectors are combined to form a mother 4-vector candidate, which is subject to a set of "combination cuts". Mother candidates passing these cuts have a more computationally expensive vertex fit applied to the daughters, to produce an associated end vertex for the mother candidate, and its associated χ^2. A fit is also done to determine the most likely primary vertex that (directly or indirectly) created the mother particle, based on which primary vertex resulted in the lowest χ^2 on the impact parameter. The fully reconstructed mother particle then has a further set of "mother cuts" applied. The full list of cuts for each trigger line is listed in appendix C.2.

It is expected that an MVA-based solution may improve signal efficiency, as many univariate cuts necessary to reduce background do not cleanly separate signal from background, particularly in the K_S^0 and K_1 stages.

6.7 | Signal Yield Estimation

6.7.1 | Efficiencies

Generator Efficiency

Table 6.5 shows the generation efficiencies of the decays for 20 000 generated events, with a generator-level acceptance cut of $10\,\text{mrad} < \theta < 400\,\text{mrad}$ ("Daughters in LHCb").

Decay	Generator Efficiency
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)\mu^+\mu^-$	$(15.27 \pm 0.14)\%$
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)e^+e^-$	$(15.16 \pm 0.14)\%$
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$	$(15.46 \pm 0.14)\%$
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to e^+e^-)$	$(15.24 \pm 0.14)\%$

Table 6.5

List of generator-level efficiencies for the four decay channels, for the cut "Daughters in LHCb".

Decay	Detector Efficiency
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)\mu^+\mu^-$	9.0%
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)e^+e^-$	6.8%
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to \mu^+\mu^-)$	8.6%
$B^0 \to (K_1(1270) \to K^0_S\pi^+\pi^-)(J/\psi \to e^+e^-)$	7.5%

Table 6.6

List of detector efficiencies for the four decay channels. These numbers are broken down in further detail in figure 6.3.

Detector Efficiency

The detector efficiency for each of the four decay modes is defined as the proportion of the 20,000 accepted generated events are deemed reconstructible at a more granular level (section 6.5). Since the decay product (or lack thereof) of the K^0_s was left undefined at the generator level, the detector efficiency essentially includes the $K^0_s \to \pi^+\pi^-$ branching fraction, as computed by Pythia, as well as the proportion of K^0_s that decay inside the detector. These efficiencies are listed in table 6.6, and are the sum of the number of reconstructible "long-long" and "down-down"reconstructed B^0 in figure 6.3.

HLT1 Efficiency

Tables 6.7 and 6.8 list the efficiencies of the HLT1 lines described in section 6.6.1. The efficiencies have the denominator of all decay products being deemed reconstructible, and are *Triggered On Signal* (TOS)* with respect to

*Triggered On Signal (TOS) means that the passed signal event may only be counted in the numerator of the efficiency if the existence of the true B^0 in the event is a necessary and sufficient condition for the trigger line to pass. Events for which this is not valid are called Triggered Independent of Signal (TIS).

HLT1 Line	Decay Channel	
	$\mu\mu$ Nonresonant	
	MagDown	MagUp
DiMuonHighMass	$(22.9 \pm 0.9)\%$	$(23.7 \pm 1.0)\%$
DiMuonLowMass	$(53.3 \pm 1.1)\%$	$(54.2 \pm 1.1)\%$
LowPtDiMuon	$(2.3 \pm 0.3)\%$	$(2.3 \pm 0.3)\%$
TrackMVA	$(36.5 \pm 1.1)\%$	$(36.6 \pm 1.1)\%$
TrackMuonMVA	$(32.6 \pm 1.1)\%$	$(33.1 \pm 1.1)\%$
TwoTrackMVA	$(50.6 \pm 1.1)\%$	$(52.1 \pm 1.1)\%$
	$\mu\mu$ Resonant	
	MagDown	MagUp
DiMuonHighMass	$(65.7 \pm 1.1)\%$	$(62.9 \pm 1.1)\%$
DiMuonLowMass	$(61.7 \pm 1.1)\%$	$(60.0 \pm 1.1)\%$
LowPtDiMuon	$(1.9 \pm 0.3)\%$	$(2.5 \pm 0.3)\%$
TrackMVA	$(39.4 \pm 1.1)\%$	$(39.4 \pm 1.1)\%$
TrackMuonMVA	$(38.0 \pm 1.1)\%$	$(37.7 \pm 1.1)\%$
TwoTrackMVA	$(55.0 \pm 1.1)\%$	$(54.8 \pm 1.1)\%$

Table 6.7

List of B^0-TOS HLT1 efficiencies for the two muon decay channels, with each figure representing 10 000 events.

the parent B^0. Most HLT1 lines have similar TOS efficiencies to the total efficiencies (meaning that most of the time, the trigger line fired for the correct reason). However, the Hlt1LowPtDiMuonLine has a TOS efficiency that is approximately a factor of 6 lower than its TIS+TOS efficiency for both muon decay modes. The overall HLT1 efficiencies are taken as the logical OR of a set of HLT1 lines. For the $\mu\mu$ decays, this is the full set of HLT1 lines used (Hlt1TrackMVALine, Hlt1TwoTrackMVALine, Hlt1TrackMuonMVALine, Hlt1DiMuonHighMassLine, Hlt1DiMuonLowMassLine, Hlt1LowPtDiMuonLine), whereas for ee decays, it is only Hlt1TrackMVALine and Hlt1TwoTrackMVALine.

HLT2 Efficiency

Tables 6.9 6.10 and list the efficiencies of the HLT2 lines described in section 6.6.3. The efficiencies have the denominator of all decay products being deemed reconstructible, as well as passing the set of HLT1 lines used for the respective HLT1 efficiencies. Events are TOS with respect to the parent B^0.

HLT1 Line	Decay Channel	
	ee Nonresonant	
	MagDown	MagUp
TrackMVALine	$(23.0 \pm 1.1)\%$	$(22.6 \pm 1.1)\%$
TwoTrackMVALine	$(37.9 \pm 1.2)\%$	$(36.2 \pm 1.3)\%$
	ee Resonant	
	MagDown	MagUp
TrackMVALine	$(23.1 \pm 1.0)\%$	$(22.8 \pm 1.0)\%$
TwoTrackMVALine	$(35.9 \pm 1.2)\%$	$(38.3 \pm 1.2)\%$

Table 6.8

List of B^0-TOS HLT1 efficiencies for the two electron decay channels, with each figure representing 10 000 events.

The TOS efficiency of the $B^0 \to K_1(1270)\mu^+\mu^-$ decay is shown in figure 6.4 as a function of the $\mu\mu$ mass, for the LL and DD reconstruction separately.

6.7.2 | Total Signal Yield

The signal yields from the combined HLT1 and non-loose HLT2 lines for all 20 000 simulated events (combined mag-down and mag-up) is shown in table 6.11. The total expected signal yield for run 3 is computed as:

$$n_{\text{Run 3 yield}} = N \times \epsilon_{\text{gen}} \times \frac{n_{\text{yield}}}{20\,000}$$

where N is the total number of expected signal events calculated in section 6.4, and ϵ_{gen} is the generator-level efficiency.

The B^0 signal mass peaks for the muon decays are presented in figure 6.5, with a crystal ball fit applied. The histogram frequencies have been scaled to be indicative of the signal yields given $50\,\text{fb}^{-1}$ of events, however the fractional uncertainties of the bins have not been altered. The distributions for the electron decay modes (both with and without reconstructed bremsstrahlung) have large radiative tails down to approximately 3 000 MeV, reducing the number of events reconstructed under the B^0 mass peak and making a fit impractical at the given level of signal events. The mass distributions for electrons reconstructed with bremsstrahlung also have a significant number of events larger than the B^0 mass, likely due to extra radiation in the calorimeters misattributed

HLT2 Line	Decay Channel	
	$\mu\mu$ Nonresonant	
	MagDown	MagUp
B2K1MuMu_LL	$(12.0 \pm 2.1)\%$	$(7.9 \pm 1.6)\%$
B2K1MuMu_LL_Loose	$(15.7 \pm 2.3)\%$	$(12.1 \pm 2.2)\%$
B2K1MuMu_DD	$(6.5 \pm 1.2)\%$	$(2.6 \pm 0.7)\%$
B2K1MuMu_DD_Loose	$(6.9 \pm 1.2)\%$	$(3.2 \pm 0.8)\%$
B2K1MuMu	$(8.7 \pm 1.1)\%$	$(4.7 \pm 0.8)\%$
B2K1MuMu_Loose	$(10.8 \pm 1.2)\%$	$(7.4 \pm 1.0)\%$
	$\mu\mu$ Resonant	
	MagDown	MagUp
B2K1MuMu_LL	$(11.4 \pm 2.0)\%$	$(10.6 \pm 1.8)\%$
B2K1MuMu_LL_Loose	$(13.8 \pm 2.2)\%$	$(13.4 \pm 2.0)\%$
B2K1MuMu_DD	$(4.1 \pm 1.0)\%$	$(4.9 \pm 1.0)\%$
B2K1MuMu_DD_Loose	$(4.9 \pm 1.1)\%$	$(6.9 \pm 1.2)\%$
B2K1MuMu	$(6.9 \pm 1.0)\%$	$(7.3 \pm 0.9)\%$
B2K1MuMu_Loose	$(10.0 \pm 1.2)\%$	$(10.6 \pm 1.1)\%$

Table 6.9

List of B^0-TOS HLT2 efficiencies for the two muon decay channels, with each figure representing 10 000 events. Efficiencies have the denominator of fully reconstructible signal events that have passed any of the following: Hlt1TrackMVALine, Hlt1TwoTrackMVALine, Hlt1TrackMuonMVALine, Hlt1DiMuonHighMassLine, Hlt1DiMuonLowMassLine, Hlt1LowPtDiMuonLine. Note that as well as being run on separate datasets, each line listed is a separate defined trigger line between the resonant and nonresonant columns, as defined in appendix C.2.

HLT2 Line	Decay Channel	
	ee Nonresonant	
	MagDown	MagUp
B2K1EE_nobrem_LL	$(6.2 \pm 2.3)\%$	$(5.0 \pm 1.9)\%$
B2K1EE_withbrem_LL	$(8.0 \pm 2.5)\%$	$(6.5 \pm 2.1)\%$
B2K1EE_nobrem_DD	$(3.8 \pm 1.2)\%$	$(5.8 \pm 1.7)\%$
B2K1EE_withbrem_DD	$(5.1 \pm 1.4)\%$	$(5.8 \pm 1.7)\%$
B2K1EE_nobrem	$(4.4 \pm 1.1)\%$	$(6.5 \pm 1.3)\%$
B2K1EE_withbrem	$(6.1 \pm 1.3)\%$	$(7.7 \pm 1.4)\%$
	ee Resonant	
	MagDown	MagUp
B2K1EE_nobrem_LL	$(9.0 \pm 1.4)\%$	$(9.4 \pm 2.6)\%$
B2K1EE_withbrem_LL	$(13.9 \pm 3.1)\%$	$(9.4 \pm 2.6)\%$
B2K1EE_nobrem_DD	$(2.6 \pm 1.0)\%$	$(1.6 \pm 0.8)\%$
B2K1EE_withbrem_DD	$(3.9 \pm 1.3)\%$	$(2.4 \pm 0.9)\%$
B2K1EE_nobrem	$(4.9 \pm 1.1)\%$	$(4.2 \pm 1.0)\%$
B2K1EE_withbrem	$(7.3 \pm 1.4)\%$	$(5.4 \pm 1.1)\%$

Table 6.10

List of B^0-TOS HLT2 efficiencies for the two electron decay channels, with each figure representing 10 000 events. Efficiencies have the denominator of fully reconstructible signal events that have passed any of the following: Hlt1TrackMVALine, Hlt1TwoTrackMVALine. Note that as well as being run on separate datasets, each line listed is a separate defined trigger line between the resonant and nonresonant columns, as defined in appendix C.2.

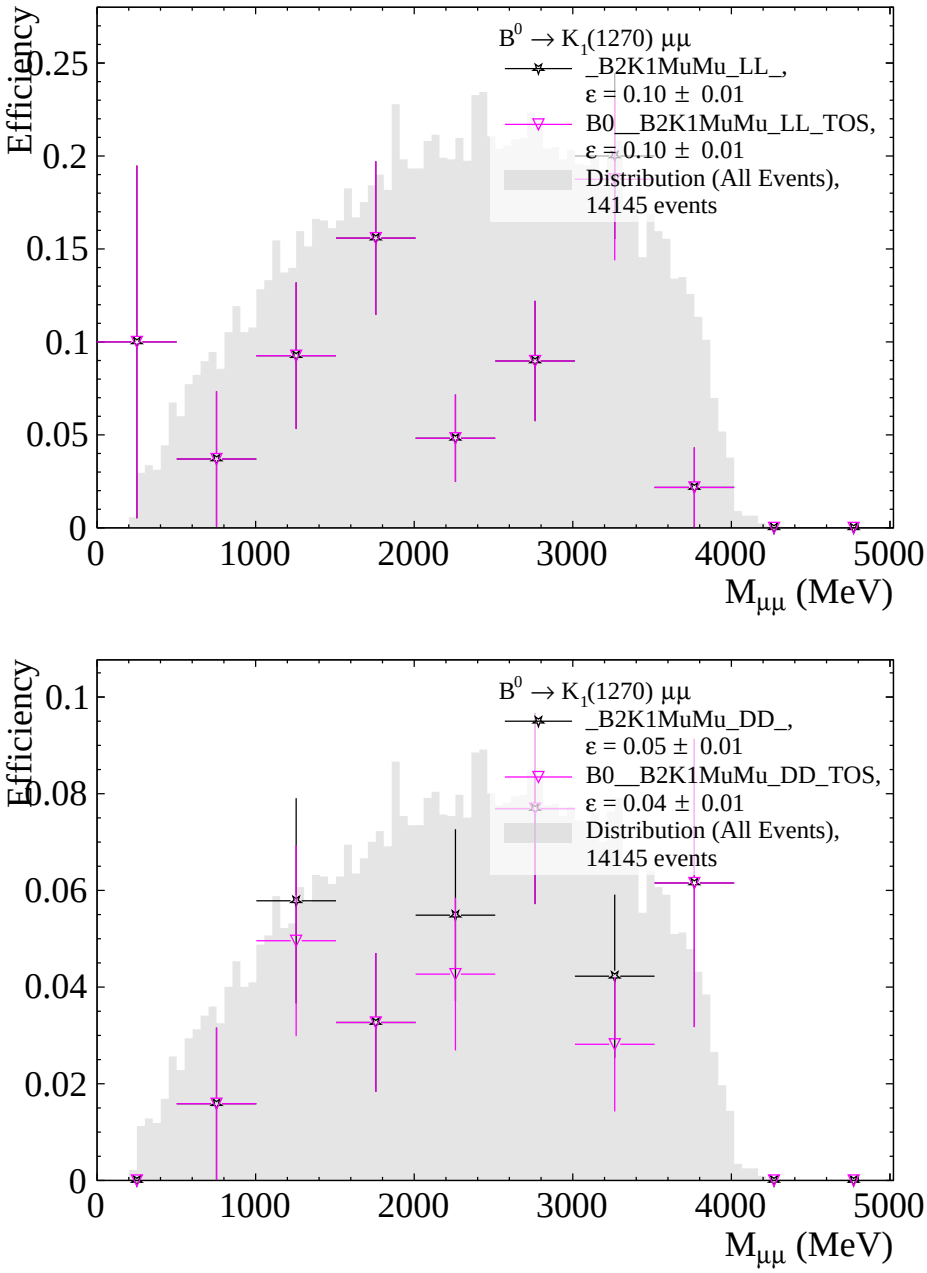


Fig. 6.4

Histograms of the efficiency of the non-loose HLT2 line for the non-resonant muon decay, as a function of the invariant dimuon mass, $M_{\mu\mu}$. The full distribution of the simulated events is shown as the grey distribution. The top and bottom plots show the efficiency for the reconstruction of the K_s^0 daughters as long-long and down-down charged pions respectively. The efficiencies have the denominator described in section 6.7.1. The pink and black markers denote the efficiency with and without the B^0-TOS requirement on the numerator respectively. The full agreement between these in the top plot is due to a combination of high TOS rate and low statistics.

Decay Channel	Yield (20 000 evts)			Yield (50 fb^{-1})
	LL	DD	Both	
$\mu\mu$, nonresonant	51	63	114	456 (9.1/ fb^{-1})
$\mu\mu$, resonant	81	56	137	44670 (893/ fb^{-1})
ee nonresonant (no brem)	31	42	73	318 (6.4/ fb^{-1})
ee nonresonant (with brem)	39	48	87	378 (7.6/ fb^{-1})
ee resonant (no brem)	36	14	50	16098 (321/ fb^{-1})
ee resonant (with brem)	43	26	69	22215 (444/ fb^{-1})

Table 6.11

List of the total yields from the HLT2 lines combined with their requisite HLT1 lines. The total run-3 yield combines signal events found via "LL" and "DD" K_s^0 candidates (see section 6.5. Statistics have been combined for both magnet polarities. For example, "ee nonresonant (no brem)" corresponds to the combined "MagDown" and "MagUp" HLT2 efficiencies in row "B2K1EE_nobrem" and column "ee Nonresonant" in table 6.10.

to the signal electron.

6.8 | Background Estimation

6.8.1 | Total Background Yield

The background yield is defined as the number of events passing the HLT1 and HLT2 trigger in a given dataset that do not contain the decay in questoin in full.

For this study, as pre-existing bank of 2.8 M HLT1-filtered unbiased events, corresponding to a total of 53 M events, were processed by the HLT2 lines, the results of which are display in table 6.12. The number of expected background events to pass the given trigger lines through run 3 is then:

$$N_{\text{bkg}} = L_{\text{int}} \sigma_{\text{inel}} \frac{n_{\text{sample}}}{5.3 \times 10^7}$$

where n_{sample} is the number of events passing the trigger line in the 53M-event sample, and $\sigma_{\text{inel}} = 75.4 \pm 3.0 \pm 4.5\,\text{mb}$ is the total inelastic proton-proton cross section at $\sqrt{s} = 13\,\text{TeV}$, extrapolated from the cross section in the LHCb phase space region of $p > 2\,\text{GeV}, 2 < \eta < 5$ [125]. These estimates are shown in table 6.13.

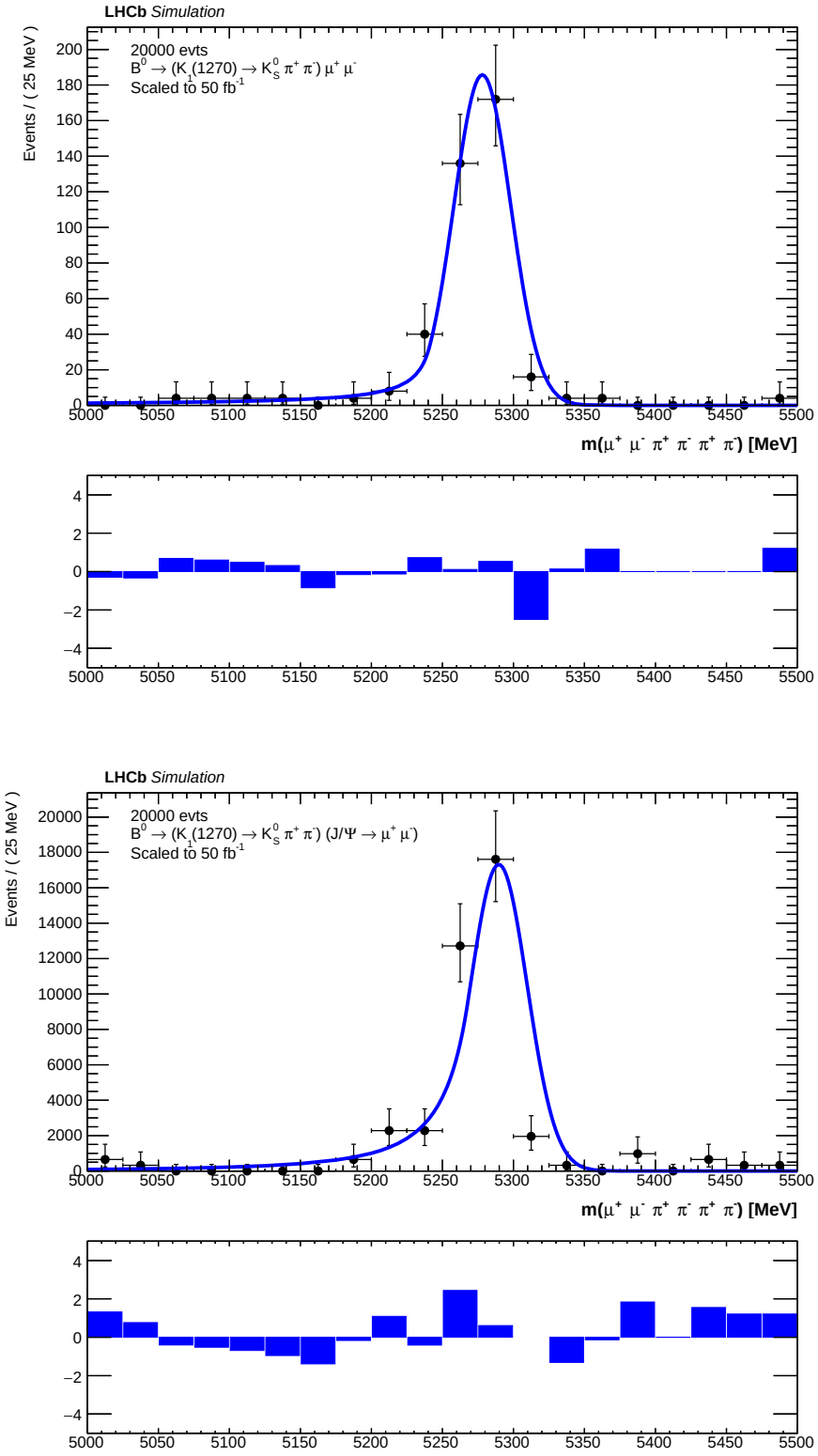


Fig. 6.5

Histograms of the enriched-ackground MC mother B^0 signal mass for the non-resonant (top) and resonant (bottom) muon decays. Both mass peaks have been fit to a crystal ball function and scaled to the expected signal statistics for $50\,\mathrm{fb}^{-1}$ of LHCb data.

HLT2 Line	Background yield (53 M evts)
`MuMu_LL_Nonres`	4
`MuMu_DD_Nonres`	11
`MuMu_LL_Res`	0
`MuMu_DD_Res`	0
`MuMu_LL_Loose_Nonres`	201
`MuMu_DD_Loose_Nonres`	316
`MuMu_LL_Loose_Res`	5
`MuMu_DD_Loose_Res`	29
`EE_nobrem_LL_Nonres`	5
`EE_nobrem_DD_Nonres`	14
`EE_withbrem_LL_Nonres`	3
`EE_withbrem_DD_Nonres`	7
`EE_nobrem_LL_Res`	5
`EE_nobrem_DD_Res`	16
`EE_withbrem_LL_Res`	3
`EE_withbrem_DD_Res`	9

Table 6.12

A list of the number of background events in a sample of 53 M unibased events passing both the HLT1 stage, and each HLT2 trigger line (see table 6.4).

	Total Background Yield	
Decay Mode	53 M evts	Run 3 estimate
$\mu\mu$, nonresonant	15	1.1×10^9 $(2.1 \times 10^7 / \text{fb}^{-1})$
$\mu\mu$, resonant	$0^\dagger$	7.1×10^7 $(1.4 \times 10^6 / \text{fb}^{-1})$
ee nonresonant (no brem)	19	1.4×10^9 $(2.7 \times 10^7 / \text{fb}^{-1})$
ee nonresonant (with brem)	10	7.1×10^8 $(1.4 \times 10^7 / \text{fb}^{-1})$
ee resonant (no brem)	21	1.5×10^9 $(3.0 \times 10^7 / \text{fb}^{-1})$
ee resonant (with brem)	12	8.5×10^8 $(1.7 \times 10^7 / \text{fb}^{-1})$

Table 6.13

A list of the total background yield of the four decay channels, in terms of the number passing in the 53M-event sample, and an extrapolation to run 3 based on the anticipated $50\,\text{fb}^{-1}$ LHCb integrated luminosity over the run. Yields for each mode are the summation of the non-loose "LL" and "DD" trigger lines (see table 6.12).

† There were no resonant $\mu\mu$ background events that passed the trigger lines in the sample, so the run-3 estimate is an effective "upper bound" assuming 1 event passed the trigger.

6.8.2 | Background Enrichment

The resulting background yield from 53M events is too low to form a distribution in M_{B^0} for the non-loose HLT2 lines. Given that a larger existing reserve of pre-computed unbiased Monte Carlo did not exist, and the extremely high computational expense of fully simulating large numbers of events, a method was developed to artificially enrich the amount of background events passing the HLT2, without requiring the simulation of more events. In this method, the HLT2 is configured to reconstruct the decay tree as before, up to the $K_1(1270)$ and dilepton candidates. From here, the trigger in its original configuration would pass the reconstructed K_1 and dilepton particles to a particle combiner, whereby all combinations are applied, and successive combination and vertexing cuts are tested on the resulting combinations. However, in this method, the K_1 and dilepton particles have the particle and vertex information of themselves and their reconstructed decay copied out of the event, and stored in respective buffers that persist into future events. After storing a sufficiently large number of particles, all collected K_1 and dilepton particles are retrieved, and the original set of B^0 combination and vertex cuts are applied to all combinations.

Since this method is used to estimate the distribution of the combinatorial background, the fact that almost all combinations contain a K_1 and dilepton particle from separate events does not affect the shape of the distribution per se. However, since the primary vertices of the two original events are not copied to the hybrid event, the cut on the resulting B^0's lowest impact parameter χ^2 to the primary vertices is not used, which increases the number of B^0 passing the trigger. As an approximation, it is assumed that this does not significantly change the shape of the distribution, and instead the enriched-background distribution is normalised to contain the same number of events per fb^{-1} between 3500 and 7000 MeV as the original, non-enriched background yields. The resulting distributions are shown in figure 6.6.

6.9 | Mass Peak Significance

The resonant $\mu\mu$ decay mode has the both the largest expected signal and lowest background yield (estimated to have an upper bound of $1.4 \times 10^6 / \text{fb}^{-1}$ given that no non-enriched background events passed the trigger. The signal and background were constrained to the B^0 mass peak of $5225\,\text{MeV} < M_{\mu\mu\pi\pi\pi\pi} < 5325\,\text{MeV}$. In this range, the absolute signal yield is 106 events (from a total of 137), corresponding to 3.5×10^4 events over $50\,\text{fb}^{-1}$. A decaying exponential was fit to the normalised enriched background in the range $7\,000\,\text{MeV} < M_B < 12\,000\,\text{MeV}$, extrapolating back, and integrating over the same window, gives a combinatorial background yield of 2.1×10^6 events over $50\,\text{fb}^{-1}$. This gives a signal significance of $\frac{s}{s+b} > 24.0$ (where s is the signal yield and b is the background yield), far past the accepted 5σ limit of observation.

This process is repeated for the other decay modes with the appropriate choice of mass window given the respective signal distributions, as summarised in table 6.14. The electron modes reconstructed without bremsstrahlung are omitted due to having a lower signal and higher background yield than those reconstructed with bremsstrahlung.

This study has reconstructed and selected events in the trigger via a set of orthogonal cuts on the kinematic and vertex fit variables, and standard lepton, pion, and ghost track identification MVA tools. A full analysis would likely contain an MVA specifically trained to isolate each decay channel's signal from its background, informed by knowledge of the constituent parts of the background. In a similar analysis to the one motivating this study, $B^0 \to K^0_s l^+ l^-$

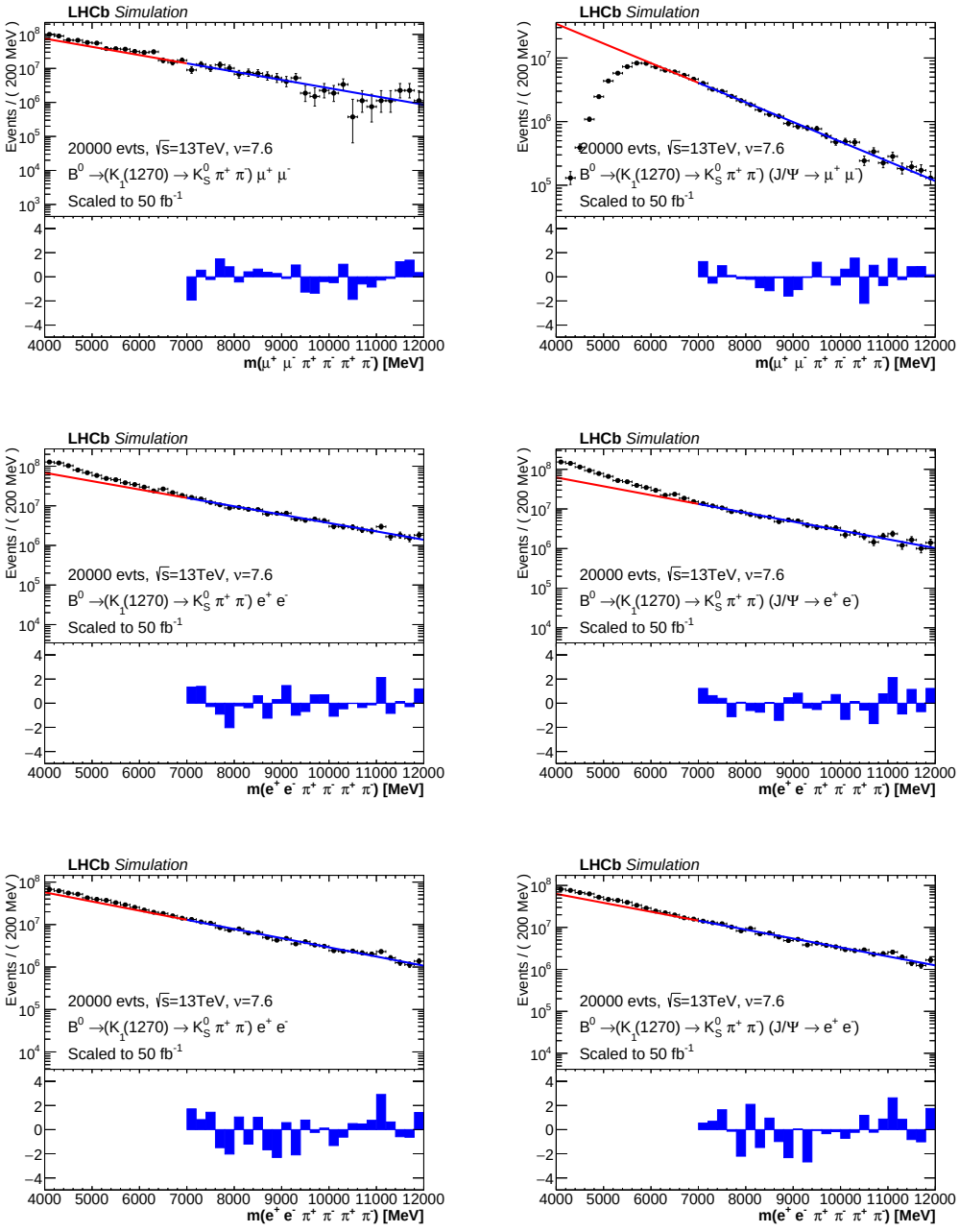


Fig. 6.6

Histograms of the mother B^0 signal mass for the non-resonant (left) and resonant (right) decay modes. The leptonic product is the dimuon (top), the dielectron reconstructed without bremsstrahlung (middle), and dielectron reconstructed with bremsstrahlung (bottom). The background rate is artificially enriched as outlined in section 6.8.2. Mass peaks have been fit to an exponential function in the region $7\,000\,\text{MeV} < M_{B^0} < 12\,000\,\text{MeV}$ and extrapolated back under the B^0 mass peak region. The distributions have been scaled to the expected signal statistics for $50\,\text{fb}^{-1}$ of LHCb data in the region $3\,500\,\text{MeV} < M_{B^0} < 7\,000\,\text{MeV}$.

Decay Mode	M_{B^0} Window	Estimated Yield ($50\,\text{fb}^{-1}$)	
		Signal	Background
$\mu\mu$, nonresonant	$5225\,\text{MeV} < M < 5325\,\text{MeV}$	3.6×10^2	3.0×10^7
$\mu\mu$, resonant	$5225\,\text{MeV} < M < 5325\,\text{MeV}$	3.5×10^4	2.1×10^6
ee nonresonant	$4750\,\text{MeV} < M < 5400\,\text{MeV}$	2.3×10^2	1.4×10^8
ee resonant	$4750\,\text{MeV} < M < 5400\,\text{MeV}$	1.4×10^4	1.6×10^8

Table 6.14

List of estimated run 3 signal and background yields under a given reconstructed B^0 mass window. Electrons are reconstructed with bremsstrahlung.

Decay Mode	Signal Significance	
	Original	Reduced Background
$\mu\mu$, nonresonant	0.067	2.1
$\mu\mu$, resonant	23.9	180
ee nonresonant	0.020	0.6
ee resonant	1.1	33.5

Table 6.15

List of signal significance figure of merit for 4 decay channels in the B^0 mass windows listed in table 6.14. The actual significance is listed, along with the significance under a hypothetical MVA that reduces the background rate by 1 000 without significantly reducing the signal efficiency. This is intended as a upper bound on what might reasonably be expected of an offline MVA tool.

decays are collected from run 1 and run 2 LHCb data in order to test lepton universality [126]. Here, MVAs to reject background efficiency act on run 2 LL and DD $\mu\mu$ and ee decays. These all result in a background rejection rate of approximately 3 orders of magnitude, with a < 1 order of magnitude reduction of signal. As an upper bound on the feasible background reduction of a hypothetical future analysis, a background yield reduction of 1 000 is applied to the four decay modes, with no effect on the signal yield. The resulting signal significances are listed in table 6.15.

These significances suggest that measurements of the J/ψ-resonant $B^0 \to K_1(1270)l^+l^-$ branching fractions, particular the $\mu\mu$ case, are within the resolution of the upgrade LHCb detector at $\sqrt{s} = 13\,\text{TeV}$, $\nu = 7.6$, given the

expected run 3 integrated luminosity of $L_{int} = 50\,\text{fb}^{-1}$. The resonant $\mu\mu$ decay should be resolvable simply from data output by orthogonal cuts in the aforementioned software trigger setup, whereas the ee decay would require a multivariate technique, either in the trigger itself or as an analysis tool, to reduce the background level by less than 3 orders of magnitude whilst keeping the signal approximately the same (slightly looser HLT2 cuts could perhaps be made to compensate for any loss of signal by an MVA). This is perhaps not likely to be feasible, but may not be impossible.

In the non-resonant cases, an MVA would need to suppress the background rate by significantly more than 3 orders of magnitude without loss of signal. With knowledge of the components of the background being severely limited by statistics, it is likely that this is not feasible for run 3 luminosity and detector conditions. As such, it may not be possible to measure the branching fraction double ratio R_{K_1} involving all 4 decay modes. However, the expected yields mean that such non-observation of these non-resonant modes in run 3 may still give upper limits on the branching fractions that provide constraints on weak right-handed current models (which predict non-resonant $B \to K_1 ll$ branching fractions significantly higher than the Standard Model).

6.10 | Further Work

An alternative approach to the lack of simulated run 3, unbiased events, rather than combinatorially enriching the background events, may be to run the trigger lines on already-existing pre-upgrade real data. Data from run 2 of LHCb is stored in "stripped" form, meaning decays of particular interest are reconstructed from the minimum-bias data and stored together. One example is the decay $[B^0 \to J/\psi(1S)(K^*(892) \to K^+\pi^-)]cc$. This decay contains a similar final state to the ones described in this study. The $K^*(892)$ has similar properties to the $K_1(1270)$, but its lower mass means that selecting for a Kaon around the $K_1(1270)$ mass window is likely to pick up combinatorial background. This effect, couple by the statistics of this decay and the large amount of existing data, provide a good way to search a very large effective amount of data to estimate the combinatorial background.

This approach would involve back-porting the trigger setup used in this study to the older LHCb software stack used for run 2. Also, the difference in detector resolutions and instantaneous luminosities between run 2 and run 3 are likely

to introduce systematics. Some factors may counteract others in affecting the total systematic effect on the predicted yield (for example, the higher event occupancy and more precise vertex and fibre tracker resolutions for run 3) but the effects are likely to be unpredictable, and would therefore only be useful as a crude estimate.

In addition to the combinatorial background, peaking backgrounds could be estimated by running the HLT on specific simulated decays, such as $B^0 \to K_s^0(J/\psi \to ll)$. With a higher simulated unbiased background sample, higher statistics in the background yield would allow such commonly occuring decays to be identified. These decays would then be simulated, or existing simulation data retrieved, and a peaking function fit to the distribution of the output yield. These decays would then be modelled separately from the combinatorial background, providing a more accurate background distribution fit.

The relatively low expected yields of the resonant decays means that a run 3 analysis may choose to also reconstruct the decays of $K_1(1270) \to K_0^*(1430)\pi$ and $K_1(1270) \to K^*(892)\pi$, which together represent roughly 40% of the total K_1 branching fraction [20]. The neutral pion provides additional reconstruction challenges, which is why these decay modes were not considered in this study.

Chapter SEVEN

Conclusion

The purpose of the LHCb upgrade is to enable the detection of particles at a much higher luminosity, in order to have improved statistics to detect particles and measure their parameters. Part of this upgrade is a redesign of the trigger system, which is moving to being software-only, meaning that it will run at the full 30 MHz inelastic collision rate rather than the reduced rate of 1 MHz from the readout of a hardware trigger. This higher rate, combined with the much higher event occupancy, means that significant changes are being made to the trigger system in order to keep the required computing resources within acceptable limits. This thesis present an investigation based on run 2 simulation of using various neural network configurations to identify tracks that originated from a B meson in the HLT1, and finds that a simple feed-forward neural network provides a tunable parameter that differentiates between B-daughter tracks with a higher efficiency than the existing one-track trigger line. This network configuration is ported to various machine learning frameworks and benchmarked at various batch sizes as part of an investigation into implementing machine learning methods inside the trigger software framework. A technique is also produced to greatly reduce the number of tracks that must be processed with a computationally expensive Kalman fit (the most time consuming segment of the run 2 trigger) without significant loss of signal efficiency. Finally, the software to take the required data for training is ported to the run 3 LHCb software stack, to facilitate a similar analysis on

run 3 data now that the software framework surrounding the trigger is in a more developed state, and the processing time requirements are more precisely known.

Another core feature of the LHCb upgrade is improved resolution, which is in part required to continue to reconstruct events with acceptable rates of error due to the increased occupancy. At the centre of this (literally) is the VELO upgrade, which involves a complete redesign called VeloPix based on silicon pixels rather than strips, and a binary, zero-suppressed readout system. Both in design and production, it is critical to ensure the correct operation of the VeloPix ASIC chips and their associated readout, control, and remote data processing systems. A crucial component of this is the VeloPix GWT bypass system, which allows test pulse data fed into the VeloPix to be read out, skipping much of the usual data processing pipeline. This thesis present the development of a fast, composable piece of software that can decode, reorder, translate, and automatically analyse binary GWT and GWT bypass data frames. This software is used in a test pulse analysis, in which various sensor hit patterns were fed into the VeloPix at different frequencies, and with different numbers of serialiser links enabled. This study found that the VELO is able to operate at the maximum bandwidth specified in its technical design report. The study also analysed the behaviour of the chip under above-rated loads, such as the pattern in which data is lost. Finally, the process of testing each VeloPix sensor triplet for quality assurance is described, including the stages conducted at CERN: the powerup and register test, the source scan and the equalisation procedure.

The increased luminosity of the LHCb upgrade is, by design, expected to unlock the potential discovery and refined measurements of variables from many different decays and phenomena, that were previously too rare to achieve sufficient statistics for. This thesis describes a study into the sensitivity of the upgrade LHCb detector to four $B^0 \to K_1(1270)l^+l^-$ decay channels, and outlines the theoretical reasons why measuring such branching fractions are attractive, including as parity-degenerate decays for suppressing long-distance vector-axial contributions to the search for beyond-standard-model right-handed current searches [65]. The estimated signal and background yields for the expected integrated luminosity of $50\,\mathrm{fb}^{-1}$ are given, and a method of enriching the combinatorial background yield is described. This is then used to calculate the expected signal and background yields under the B^0 mass peak, and a set of

signal significances are given for the signal and background, given the output of the developed trigger line. From this, it is concluded that the resonant decays $\mu\mu$ would be readily measured given a dedicated HLT2 line. The resonant ee decay may potentially be resolved, and more work would be required to investigate the contents of the background in this case, to find out how much it can be reduced. The non-resonant decays will likely not feasibly be resolved, with the electron mode in particular having a very low, smeared-out signal yield.

Appendices

Chapter ONE

Trigger

A.1 | Definition of Terms

The following is a list of terms defined as they apply in the context of this note:

- An *evaluation object* refers to a set of inputs (usually to a detector track or entire event, depending on context), as it is evaluated by some classifier.
- *Pass probability* is the evaluation output of a probabilistic binary classifier (one that involves a cut on a continuous output variable to determine predicted class). In this context it is treated like the probability that an evaluation object belongs to a particular class, given the input data.
- *MC Truth* is the true classification category of an evaluation object according to its Monte Carlo simulation.
- *True Positives*, True Positive Count, or TP, is the number of evaluation objects that *truly* fulfil the evaluation criterion (containing a b particle), and are also classified as positive by the classifier.
- *True Negatives*, True Negative Count, or TN, is the number of evaluation objects that *truly* fail the evaluation criterion (containing a b particle), and are also classified as negative by the classifier.
- *False Positives*, False Positive Count, or FP, is the number of evaluation objects that *truly* fail the evaluation criterion (containing a b particle),

but are incorrectly classified as positive by the classifier.

- *False Negatives*, False Negative Count, or FN, is the number of evaluation objects that *truly* fulfil the evaluation criterion (containing a b particle), but are incorrectly classified as negative by the classifier.

- *Signal efficiency*, or just efficiency, refers to the fraction of input objects containing a b particle that are correctly classified as such by the classifier. Formally:

$$\text{eff} = \frac{TP}{TP + FN}$$

- *Background rejection*, or just rejection, refers to the fraction of input objects not containing a b particle that are correctly classified as such by the classifier. Formally:

$$\text{rej} = \frac{TN}{TN + FP}$$

Signal efficiency and background rejection are particularly suitable metrics for this work as they are agnostic to the ratio between total signal and background.

A.2 | Definition of Variables and Trigger Lines

Table A.1 describes the variables used as cuts in the HLT1TrackMVA and HLT1TwoTrackMVA trigger lines. Table A.2 lists the full list trigger lines available in the run 2 configuration of the HLT1, as of 2017. Finally, table A.3 and A.4 detail the selections performed on tracks (and combinations of tracks) by the `Hlt1TrackMVA` and `Hlt1TwoTrackMVA` lines respectively.

Note on Corrected Mass

The "corrected mass" $M_{\text{corr}} = sqrt(M^2 + |p`_{T\text{missing}}|^2) + |p`_{T\text{missing}}|$, where $p`_{T\text{missing}}$ is the missing momentum transverse to the direction of flight of the B, which helps to account for possible missing decay daughters [127].

Variable	Description
M	Mass
M_{corr}	Corrected mass (section A.2).
p	Momentum
p_T	Transverse momentum
η	Pseudorapidity
χ^2_v/N_{DOF}	χ^2 of mother particle vertex fit, per degree of freedom
ghostProb	Probability of track being a ghost (artefact misclassified as a real track), evaluated by an MVA
$\chi^2_{\text{PV, min}}(\text{IP})$	Minimum χ^2 of primary vertex impact parameter fits
$\chi^2_{\text{trk}}/N_{\text{DOF}}$	χ^2 of track Kalman fit per degree of freedom
SUMTREE(x)	The sum of all values of x in the decay tree
NINTREE(x)	The number of particles fulfilling condition x in the decay tree
DIRA_{BPV}	Direction angle of the particle based on the best-fit primary vertex

Table A.1

A glossary of the variables used in the run 2 HLT1 MVA trigger lines.

Hlt1TrackMVA	Hlt1CalibTrackingPiPi
Hlt1TwoTrackMVA	Hlt1DiMuonNoIP
Hlt1TrackMVATight	Hlt1DiMuonNoIPSS
Hlt1TwoTrackMVATight	Hlt1DiProton
Hlt1TrackMuon	Hlt1DiProtonLowMult
Hlt1TrackMuonMVA	Hlt1IncPhi
Hlt1DiMuonHighMass	Hlt1L0Any
Hlt1DiMuonLowMass	Hlt1L0AnyNoSPD
Hlt1SingleMuonHighPT	Hlt1LowMultMaxVeloAndHerschel
Hlt1DiMuonNoL0	Hlt1LowMultMaxVeloCut
Hlt1B2GammaGamma	Hlt1LowMultMuon
Hlt1B2HH_LTUNB_KK	Hlt1LowMultPassThrough
Hlt1B2HH_LTUNB_KPi	Hlt1LowMultVeloAndHerschel_Hadrons
Hlt1B2HH_LTUNB_PiPi	Hlt1LowMultVeloAndHerschel_Leptons
Hlt1B2PhiGamma_LTUNB	Hlt1LowMultVeloCut_Hadrons
Hlt1B2PhiPhi_LTUNB	Hlt1LowMultVeloCut_Leptons
Hlt1BeamGasBeam1	Hlt1Lumi
Hlt1BeamGasBeam2	Hlt1MBNoBias
Hlt1BeamGasNoBeamBeam1	Hlt1MultiDiMuonNoIP
Hlt1BeamGasNoBeamBeam2	Hlt1MultiMuonNoL0
Hlt1Bottomonium2KstarKstar	Hlt1NoBiasNonBeamBeam
Hlt1Bottomonium2PhiPhi	Hlt1ODINTechnical
Hlt1CalibHighPTLowMultTrks	Hlt1SingleElectronNoIP
Hlt1CalibMuonAlignJpsi	Hlt1SingleMuonHighPTNoMUID
Hlt1CalibRICHMirrorRICH1	Hlt1SingleMuonNoIP
Hlt1CalibRICHMirrorRICH2	Hlt1Tell1Error
Hlt1CalibTrackingKK	Hlt1VeloClosingMicroBias
Hlt1CalibTrackingKPi	Hlt1ErrorEvent
Hlt1CalibTrackingKPiDetached	Hlt1Global

Table A.2

List of trigger lines available in run 2.

	Hlt1TrackMVA
Parameters	`MinPT` $= 1\,000\,\text{MeV}$
	`MaxPT` $= 25\,000\,\text{MeV}$
	`MinIPChi2` $= 7.4$
	`TrChi2` $= 2.5$
	`TrGP` $= 0.2$
	`Param1` $= 1.0$
	`Param2` $= 1.0$
	`Param3` $= 1.1$
Selections	$\chi^2_{\text{trk}}/N_{\text{DOF}} < TrChi2$
	$\text{ghostProb} < TrGP$
	$(((p_T > \text{MaxPT})\&(\chi^2_{\text{PV, min}}(\text{IP}) > \text{MinIPChi2}))\vert$ $(\text{MinPT} < p_T < \text{MaxPT})\&(\log(\chi^2_{\text{PV, min}}(\text{IP})) >$ $(\frac{\text{Param1}}{((p_T/GeV - \text{Param2})^2}+$ $\text{Param3} \times \frac{\text{MaxPT}-p_T}{\text{MaxPT}} + \log(\text{MinIPChi2})))))$

Table A.3

A description of the selections used by the run2 HLT1 track MVA, and their parameters. Tracks are filtered successively by each selection. "&" indicates the Boolean AND between expressions, and "|" indicates the Boolean OR[a].

[a]Operation that returns `True` if any of its inputs are true. Not to be confused with *Bullion Ore* (a metallurgical oxymoron) or *Bouillon Oar* (a short-lived seafaring tool that does make the prospect of capsizing slightly tastier).

	Hlt1TwoTrackTrackMVA
Parameters	`P` = 5000 MeV
	`PT` = 500 MeV
	`TrGP` = 999
	`TrChi2` = 2.5
	`IPChi2` = 4.
	`MinMCOR` = 1000 MeV
	`MaxMCOR` = $1e9$ MeV
	`MinETA` = 2
	`MaxETA` = 5
	`MinDirA` = 0
	`VOPT` = 2000 MeV
	`VxChi2` = 10
MVA Type	MatrixNet
MVA Variables	χ^2_v
	$\chi^2_{textBPV}$
	SUMTREE(p_T)
	NINTREE($\chi^2_{\text{PV, min}}$(IP) < 16)
Pre-combination Selections	($p_T > PT$)&
	($P > P$)&
	($\chi^2_{\text{trk}} < TrChi2$)&
	(ghostProb < $TrGP$)&
	($\chi^2_{\text{PV, min}}$(IP) > $IPChi2$)
Combination Selections	(χ^2_v/N_{DOF} < VxChi2)&
	(MinETA < η < MaxETA)&
	(MinMCOR < M_{corr} < MaxMCOR)&
	(DIRA_{BPV} > MinDirA)

Table A.4

A description of the selections used by the run 2 HLT1 two-track MVA, and their parameters. Tracks are filtered successively by each selection. "&" indicates the logical AND between expressions.

A.3 | Data Analysis Methods And Machine Learning

A.3.1 | Linear Classification and Regression

A classifier or regressor is defined as linear if its decision is based on some function of a linear combination of the set of inputs. Geometrically, the decision boundary of a binary linear classifier forms an $\mathbb{R}^{n-1}$ hyperplane in the $\mathbb{R}^n$ feature space [128].

An early development of linear classification was Linear Discriminant Analysis (LDA), based on Fisher's linear discriminant [129]. Another, Support Vector Machines (SVM), computes the maximum margin between two hyperplanes between linearly separable data [130]. Once a hyperplane is found, maximising its margin is a quadratic (convex) optimisation problem with a single minimum. Non-separable data may be computed with the introduction of a soft margin hyperplane [131].

A.3.2 | Dimensionality Reduction

Reducing the dimensionality of the input space is an important consideration in optimising machine learning models. Methods that can be used as a classifier, such as LDA, may also be used to reduce input dimensionality. Principal Component Analysis (PCA) is another such method [132]. Whereas LDA is a supervised method that maximises the distinction between output classes, PCA is an unsupervised method that maximises the total variance of the data. PCA can be used to find the linear combinations of variables with the largest variance, or to select the variables that each contribute the most to the total dataset variance. The relative merits and use cases of LDA and PCA are investigated in ref [133].

A.3.3 | Non-linear Models

Many methods exist for modelling non-linear distributions, including ways to introduce non-linearity into existing methods. Methods such as SVMs, PCA, and the perceptron may be altered to approximate non-linear models through the use of a kernel trick. This consists of performing a non-linear mapping on the input data from $\mathbb{R}^n$ into a higher dimensional space $\mathbb{R}^{m>n}$ (such that the

projection $\mathbb{R}^m \to \mathbb{R}^n$ leaves the data unchanged) prior to training the linear classifier [130]. The linearly separating hyperplane in the new, implicit space $\mathbb{R}^m$ is equivalent to a non-linear decision boundary in $\mathbb{R}^n$. In other words, the combination acts as a non-linear classifier.

A decision tree is a simple algorithm for computing a decision given some data. The tree is traversed from the root, with each node being its own decision based on a cut on one of the variables. The leaf that the procedure arrives at represents the final decision. Boosting is a technique that involves using an ensemble of classifiers to produce a more accurate one with mitigated overfitting, and was applied to decision trees in ref [134]. These Boosted Decision Trees (BDTs) are widely used in HEP, particularly to classify particles or events.

The multilayer perceptron (MLP) (commonly referred to as a feed-forward neural network) is an extension of the perceptron that allows for the non-linear approximation of functions. It consists of an input layer, an output layer, and an arbitrary number of "hidden" layers, each of which has a chosen activation function.

Deep neural networks allow, through the use of multiple layers of non-linear activation function, the modelling of non-linear functions. They also allow for the iterated abstraction of features, which is effectively shown in image-based convolutional neural networks.

A.3.4 | Gradient Descent and Backpropagation

The difference between a supervised network's output and the target truth can be quantified by some error function. This function can then be minimised via the first-order gradient descent algorithm. The method of backpropagation, popularised in [135], describes the procedure of propagating an output error function backwards through the layers of a deep neural network, so as to adjust the network's weights accordingly. The paper uses a neuron activation function of

$$y_j = \frac{1}{1 + e^{-x_j}}$$

where j denotes the index of a neuron within a layer, but notes that the only restriction on the function is that it has a bounded derivative.

The derivative of the error function is computed with respect to the weights

of each successive layer from the output to the input, and the weights are then adjusted by the negative of this gradient, up to a set coefficient (learning rate).

A.4 | More Advanced Machine Learning Techniques

Feed-forward ("vanilla") neural networks receive as input a fixed-length vector of features, whose elements represent the same variables each time, and output a fixed vector in return. This approach is simple, and allows for effective parallelisation during training and evaluation. However, there exist problems whose input is not effectively represented as a fixed-length vector. Common industrial examples of these types of input data are audio and video streams, and strings of text, from sentences to novels. These problems also most commonly involve returning a variable-length vector.

Recurrent neural networks (RNNs) are a model that involves defining a module of neurons that can feed its output back into the input, and contains a hidden state where information can be stored and iteratively processed over a time step (or a quantity analogous to time). The network adjusts its parameters by "unrolling" the RNN and performing the backpropagation algorithm as if it were a typical feed-forward neural network - a process known as *Backpropagation Through Time* (BPTT), outlined in [136].

Simpler forms of RNN are more prone to the problem of vanishing and exploding gradients. As first described in [137], and further explored in [138], this is where the exponential-like multiplication of the gradient over many unrolled layers causes it to converge to 0 or diverge to infinity, the result of which is that traditional RNNs have difficulty in "remembering" long-term correlations and dependencies. One solution to this problem is the *LSTM*, or Long Short-Term Memory, an internal unit for an RNN that performs better for dependencies over larger time scales. As the introductory paper [139] describes, the LSTM unit of an RNN contains gates that can direct the propagation of error in the network, allowing for faster training and superior performance in more complex problems.

Sequence to Sequence ("seq2seq" [140, 141]) is a method for transforming one variable-length sequence into another, possibly of different length and format.

Examples of this include describing a picture with a sentence, captioning a spoken sentence, and translating a text sentence from one language to another (the focus of refs [140] and [141]). This architecture primarily involves an encoder RNN (LSTM) that transforms an input sequence into a fixed-length vector, and a decoder RNN that transforms this vector into the output sequence. The hidden state of the encoder forms the fixed-length vector to be decoded.

Computing a decision for an event that can contain an arbitrary number of tracks is equivalent to transforming a set of elements to an output scalar (or vector for the case of classifying the likelihoods of multiple possible particles). This is not fully analogous to Sequence to Sequence, as the output vector is fixed-length. Also, although the input is variable-length, it is an unordered set rather than a sequence (the list of all tracks can be somewhat meaningfully ordered by variables such as the χ^2 of the impact parameter fit, however this is not the same as being logically sequential; the logical structure of a decay tree is that of a directed acyclic graph rather than a linear sequence). A method is presented in ref [142] for extending Sequence to Sequence to accommodate unordered input or output sets.

Convolutional neural networks (CNNs) [143] are neural networks based on a set of convolutional kernels that act on a set of data in one or more dimensions. A layer of the network consists of a set of filters of a certain size, that convolve over the data to produce a new sets of data. An example of a filter that may arise from training a CNN to analyse images is one that detects edges. Further, pooling layers will sub-sample the data with a particular stride length, such that the resulting datasets are smaller. An entire CNN typically contains many convolutional and pooling layers, until the resulting size of the output data is small enough to flatten and feed into a normal feed-forward neural network. As well as shrinking the final size of the data to be manageable within a fully-connected network, the result of the convolutions and pooling is that, with successive layers, large-scale, abstract features are extracted from the local, top-level data. As the network is trained, the values in the filters are updated according to a normal gradient descent algorithm.

Graph convolutional networks (GCNs) [144] are convolutional neural networks that operate on graph-based data. In the same way that regular CNN filters operate on the neighbouring data points in each dimension of the data, GCNs operate on the neighbouring nodes of each node in the graph.

During training in a neural network, the modification of parameters in some layer i (through standard optimisation methods) will alter the distribution of the layer's output, and thus the distribution of the input to layer $i+1$. This may complicate training, and require a longer, slower training period with a lower learning rate. Ref [145] refers to this effect as *internal covariate shift*, and proposes as a solution *batch normalisation*, a method of normalising layer inputs over each batch. The paper reports convergent learning with significantly faster choices of hyperparameters, and improved loss and accuracy when applied to existing models.

A.4.1 | Universality and Generality

It can be shown that a multilayer perceptron with only a single hidden layer is sufficient to approximate any continuous multivariate function to an arbitrary precision over an arbitrarily large range. This is the universal approximation theorem, proven for sigmoidal or "squashing" activation functions [146, 147] - that is,

$$\sigma(x) \to \begin{cases} 1 & \text{as } x \to +\infty \\ 0 & \text{as } x \to -\infty \end{cases} \tag{A.1}$$

and later proved for all "arbitrary bounded and non-constant activation functions" [148].

However, this theorem places no limits on the number of hidden layer nodes required to satisfy the desired precision and range.

The performance of any optimisation algorithm is theoretically limited in its generality, by what is known as the No Free Lunch Theorem, introduced in ref [149]. This seminal paper shows that, for any given class of search or optimisation problem, all algorithms to solve that problem will have equal performance when averaged over all possible problems in that class. This set of algorithms includes a random search. This consequence is similar to how the pigeonhole principle in mathematics proves that any lossless compression algorithm must increase file size for at least as many files as for which it decreases it.

This finding has significant implications for deep learning, which require an optimisation algorithm to minimise the cost function of a model. Ref [149] uses the search for a global maximum as an example of the equal performance

of different algorithms, showing that in the general case, gradient descent is just as effective as gradient ascent for finding the global maximum. There are caveats to the application of this theorem to cost minimisation algorithms: firstly, it is typically only necessary to obtain a sufficiently good solution, not the optimal one; secondly, the problem space will not be uniformly probable, with certain configurations more likely to appear than other, more pathological cases, which makes one algorithm more performant in practice.

A.5 | HEP-Based Research Into Machine Learning

A.5.1 | Bonsai Boosted Decision Trees

An algorithm coined *Bonsai Boosted Decision Tree* [150] has been used in the main LHCb topological trigger for almost all of the data collected by the detector. The algorithm was designed to solve the limitations of a BDT in the context of a software trigger, and is so named because it "permits the grower of the tree to control and shape its growth". It involves discretising the input variables of a BDT, and transforming the reduced-precision tree into a 1-dimensional lookup table to optimise for speed.

A.5.2 | Data Planing

When training a neural network, it is desirable to quantify how much a particular variable contributes to the discriminating power of the network. This discriminating power could be inferred by the network training directly on that variable, or training on a combination of dependent variables (as a non-linear classifier could infer the discriminating power of the cylindrical radius r by training on the cartesian plane coordinates x and y). *Data planing*, as described in ref [151]* is the procedure of weighting a set of events with respect to a particular variable, such that the intrinsic discriminating power of that variable in the dataset disappears. The network can be re-trained on this data and the area under the ROC curve (see section 4.5.4) compared between evaluations of the two networks (with a higher value indicating better performance). This allows the network to present some intuition about what potential high-

*This paper was also the subject of a talk of the same name at 2nd IML Machine Learning Workshop, CERN, April 10, 2018.

level attributes of the system convey the most information in telling apart two classes of data. (Ref [151] states the previous example of cylindrical and cartesian coordinates).

A.5.3 | HEPDrone

In a typical high level trigger software framework such as that of LHCb, there are multiple trigger lines that process events, of which an increasing number are likely to be forms of multivariate classifiers such as feed-forward neural networks. Ref [152] explores the features of a drone network. This is a shallow, wide neural network that is trained to emulate the output of a more complex, deeper, pre-trained neural network. The drone network will have a lower or equal performance to the original network, but will be more highly paralellisable by its architecture. The paper explores the situation of many different neural networks from multiple machine learning libraries all running simultaneously inside one software trigger framework, and suggests a potential solution of a single large drone network with multiple outputs, that takes in the inputs of, and emulates the outputs of, multiple neural networks in parallel.

A.6 | Review of machine learning libraries

The space of machine learning frameworks is relatively young, with a lack of consolidated concensus and a number of competing packages - some of which are still being actively developed.

Despite many differences in these pieces of software, they are all designed to perform a similar function - that is, to define a graph of (differentiable) computational operations that are then executed on a set of input data. The advantage of this more declarative style of programming (rather than the imperative style of code that operates directly on data) is that the framework can take advantage of any existing parallel computation resources (such as Graphics Processing Units or GPUs) to accelerate execution over a large enough scale of data. The parallel framework most commonly called by these packages is Nvidia's proprietary CUDA (Compute Unified Device Architecture), although there is slowly growing support for the equivalent open standard OpenCL.

The language Python (particularly version 3) is the most widely supported and used language with which to call most machine learning libraries, although other language bindings do exist in some cases.

A.6.1 | TensorFlow

The most widely-used package is Google's TensorFlow [153]. This package statically compiles the computational graph after its definition, meaning that the definition and run stages are totally distinct, and new operations cannot be added once the "TensorFlow session" is being run. Perhaps because of this paradigm, TensorFlow machine learning models are supported to be run "in production" on a wide variety of platforms, both hardware and software. However, it also means that testing, debugging, or logging different models is more challenging, and a greater amount of custom functionality has to be executed using the set of supported TensorFlow operations, with less ad-hoc Python code.

TensorFlow has multiple APIs with different levels of abstraction, depending on the complexity and customisation necessary.

A.6.2 | Keras

Keras [154] is not a full framework in itself, but rather a wrapper of the API of other frameworks, (including TensorFlow, Theano*, and Microsoft Cognitive Toolkit), which abstracts away some lower level implementation details.

A.6.3 | PyTorch

PyTorch is a slightly newer package for machine learning [101] based on a previous library, Torch (which used the Lua scripting language), with its largest contributor being Facebook. The creation of the computational graph is done with a dynamic JIT ("Just In Time") compiler, meaning that the contents of variables can be inspected and modified at runtime. The original code for creating neural networks for event data analysis was written using TensorFlow, but for this reason a port using PyTorch was written, as it made model debugging and creating different network architectures simpler and faster.

In this time, TensorFlow has developed an "eager execution" mode to its software [155]. This mode evaluates operations immediately in a similar way to PyTorch, bringing the same kinds of potential advantages. Conversely, Pytorch has been updated to version 1.0 with greater versatility to use in production [156].

*Theano is a machine learning library for Python that is no longer maintained.

A.6.4 | TMVA

TMVA (Toolkit for Multivariate Data Analysis) is a machine learning toolkit embedded in the data analysis framework ROOT [157]. ROOT is very widely used in HEP, and is interoperable with other tools. The ROOT file format is the basis for storing and exploring many types of data in HEP, including particle detector events.

TMVA contains methods for many traditional classifiers and regressors, and has recently implemented methods for parallelised deep learning of different architectures, such as recurrent and convolutional deep neural networks. As well as the C++ API, TMVA has interfaces for the Python and R languages.

A.7 | Calculation of Efficiency-Rejection Arrays

Signal efficiency – background rejection curves can be calculated by varying the common parameter *pass probability*. Placing the binary classification boundary of pass *vs* fail on a particular value of pass probability (in other words, doing a *cut*) will yield particular values of signal efficiency and background rejection. A simple but naïve algorithm for computing this involves calculating the overall efficiency and background rejection over the entire output set for many different cuts on the output probability. This is an $O(N \times E)$ operation, where $N =$ the total number of evaluation objects, and $E =$ the desired number of value pairs for efficiency and rejection.

A more computationally efficient algorithm (described in [158] to compute true positive rate and false positive rate for a ROC curve, but altered in this work to calculate efficiency and background rejection) takes advantage of the monotonicity of the efficiency and background rejection with respect to the pass probability cut value. The pairs of pass probabilities and truth labels are sorted by the probability value in descending order. Instead of manually producing a set of values for the probability cut and computing the efficiency and rejection for the whole set each time, each value of the probability in the list is used as the cut value one after the other, and the monotonicity of the list means that only one label needs to be inspected to recompute the efficiency and rejection each time. This means that a set of N efficiency-rejection pairs can be computed in $O(N)$ time.

This method was employed to calculate the efficiency-rejection curves of the

two-stage neural network classifier. Such a setup has two manual degrees of freedom: the pass probability cut threshold for each stage. Since the incoming data to the second stage is mutated depending on the first-stage probability threshold, the first-stage threshold must be fixed while the second-stage threshold is varied. These two degrees of freedom amount to a large number of efficiency-rejection curves. The first-stage threshold determines the lower-right point on each curve, and varying the second-stage threshold traces out one individual curve. A single curve was calculated in the following way:

- An array is constructed with three elements in each row: The first-stage neural network output probability, the second stage probability, and the binary class label indicating whether the track actually passed the condition (originating from a b particle).
- The array's rows are sorted by the first-stage probability p_1 in descending order.
- Some threshold p_{cut} is used as the fixed cut on the first-stage output probability. The bisection of the array corresponding to $p_1 \geq p_{cut}$ is selected, and the other section is discarded.
- The remaining rows are then sorted in descending order of the second-stage probability p_2, and the second-stage probability and class label columns are processed according to the previously outlined procedure to produce arrays of signal efficiency and background rejection (for this particular value of p_1).

A.8 | Calculation of Values For Maximum Kalman-Free Background Rejection

As demonstrated in figure A.1, for any value of background rejection rate from the first-stage classifier (point A), there is a minimum total background rejection that attains the same signal efficiency as the better classifier on its own (point C). Phrased backwards, for any signal efficiency and background rejection attained by the more expensive classifier on its own, there is a maximum number of background and signal tracks that can be rejected by a non-Kalman NN beforehand without any loss of classifier performance.

This total "Kalmanless rejection" fraction is given for many values of the

combined signal efficiency in figure 4.15, and was computed in the following way:

- Find the first-stage pass probability threshold cut that produces the required efficiency (point A in figure A.1). This is done by sorting the probabilities and generating the efficiency values one by one with the efficient method from appendix A.7. When the required efficiency value eff_A is crossed, the array is bisected by the current row in the array and the higher set of probabilities are used. The difference in background rejection between the upper and lower curves is noted as $\text{rej}_{diff} = \text{rej}_B - \text{rej}_A$.
- As before, the total two-stage efficiency and background rejection values are calculated. This time, each value of the background rejection is compared to the corresponding value of the sole expensive classifier selection for the closest value of signal efficiency. When the difference in background rejection is less than some fraction of the original difference:

$$\text{rej}_{diff}^{current} \leq \epsilon \times \text{rej}_{diff}$$
$$\epsilon \ll 1$$

the value of the combined two-stage background rejection is deemed to be approximately equal to that of the sole expensive classifier*. This corresponds to point C on the figure. The combined two-stage signal efficiency is then the value corresponding to this background rejection.

A.9 | Calculation of Error Bars For Efficiency and Rejection of Existing Trigger MVAs

The existing trigger MVAs perform a binary decision over the set of tracks or events, and return a set of binary values, which when computed with the corresponding truth labels, gives a single value for signal efficiency and background rejection. Error bars were estimated by calculating the Binomial Proportion Confidence Intervals for each metric using a Clopper-Pearson interval. Treating the efficiency and rejection rates of sets of events as binomially distributed is valid, but is not strictly accurate for sets of tracks, as all tracks from each primary vertex are mutually dependent, subject to the physics of the particular event (e.g. a jet of many particles that all pass the trigger line).

*A value of $\epsilon = 0.001$ was used in this work.

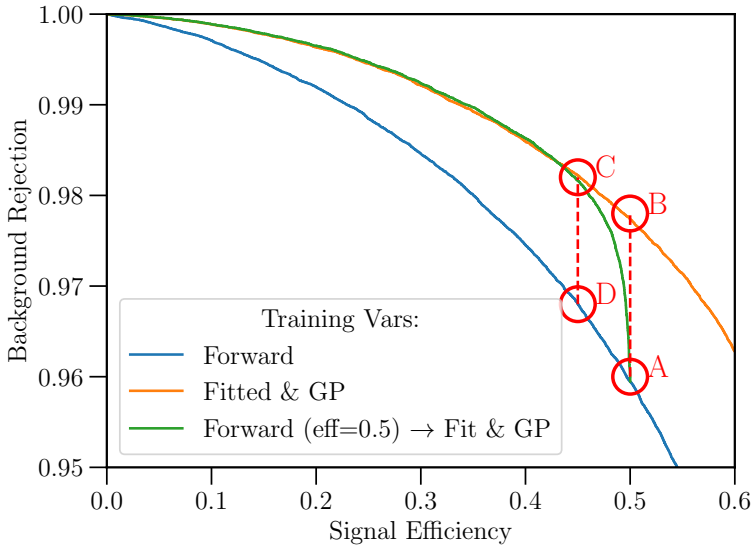


Fig. A.1

Minimum total background rejection with two-stage classifier. See appendix A.8 for details

A.10 | Activation Function

For this work, a variety of neuron activation functions were tested in the neural network classifier, including ReLU, tanh, ELU, and LeakyReLU. Of those tested, no single function stood out as the best. However, training runs using the simple ReLU function:

$$f(x) = \begin{cases} x & x \geq 0 \\ 0 & x \leq 0 \end{cases}$$

were prone to large discontinuities in the ROC curve, caused by many different inputs giving the same output value from the network. This is due to "dead neurons", whereby neurons can get "stuck" with a gradient of 0 in the negative x region, with no way to respond to the updating loss function.

This problem is solved with the use of the equally simple "LeakyReLU" function:

$$f(x) = \begin{cases} x & x \geq 0 \\ \epsilon x & x \leq 0 \end{cases}$$

$$\epsilon \ll 1$$

Algorithm	TrackHLT1FitHLT1Seq	ForwardHLT1FitterAlg
User time (ms)	11.823	10.976
Clock time (ms)	11.842	10.988
min (ms)	0.071	0.015
max (ms)	93.0	89.4
σ (ms)	8.79	8.3
N	8386	8386
$\frac{\sigma}{\sqrt{N}}$ (ms)	0.096	0.091
total (s)	99.309	92.154

Table A.5

Summary of the computation time required by the HLT1 Kalman track fit for run 2 conditions, on unbiased simulated data.

As multiple activation functions were found to produce equal classification performance, the LeakyReLU function was chosen for subsequent tests due to its relative simplicity, and minimal computational complexity.

A.11 | Kalman Fit Computation Time

The computation time of the Kalman track fit was determined here using a run in Brunel* of the first 10 000 events from the unbiased run 2 simulated "2016magup" dataset. The software was run on a server with a 3.07GHz Intel Core i7 950 CPU, and 15GB of RAM, but all times are renormalised to the reference server running a 2.8GHz Xeon. The `ForwardHLT1FitterAlg` algorithm performs the Kalman fit to the tracks in the event, and is a subroutine of `TrackHLT1FitHLT1Seq`. The extra processing time of the latter is due to overhead including initialising the state of the VELO and copying VELO tracks.

The sample benchmarked involved 169432 tracks processed by the Kalman fit in 8386 events. The results are shown in table A.5.

The 92.154 seconds of total time in ForwardHLT1FitterAlg gives a per-track time of

$$(11.0 \pm 0.091)\,\text{ms/evt} \times \frac{8386\,\text{evt}}{169432\,\text{trk}} = (0.544 \pm 0.0045)\,\text{ms/trk}$$

*a run 2 offline reconstruction software framework

A.12 | Details of computers used for profiling

	Machine A	Machine B
CPU	3.60GHz Intel i7-4790	AMD FX-8320
Single-thread passmark score	2282	1397
GPU	Nvidia GeForce GTX 980 Ti	n/a
RAM	16 GB	16 GB

Chapter TWO

VeloPix

B.1 Glossary

Table B.1 contains a glossary of VELO-related terms.

B.2 VELO Data Formats

B.2.1 GWT Bypass

The GWT bypass data frames are described diagrammatically in figures B.1 and B.2, which have the following properties:

Endianness	Big-endian
Bit-numbering scheme	*LSB_0*
→ Byte order	Left to right

with the byte order implied by the other two properties. The GWT bypass frame data structure is also displayed in table B.2, with table B.3 showing the internal structure of an SPP (super-pixel packet).

B.3 SPP Special Frame Format

When the hitmap of an SPP is fully zero, that particular GWT frame is a "special frame". There are 5 types of special frames, with IDs listed in table

GWT:	Gigabit Wireline Transmitter
GBT, GBTx:	GigaBit Transceiver
TELL40:	Readout board
SOL40:	Control distribution
SODIN:	Readout supervisor board
MiniDAQ:	Data Acquisition board
VELO:	Vertex Locator
VeloPix (VP):	ASIC design of VELO upgrade based on pixels rather than Silicon Strips
LSB:	Least significant bit
MSB:	Most significant bit
Big-Endian:	Byte order starting with most significant byte
Little-Endian:	Byte order starting with least significant byte
LSB_0:	Bit numbering scheme with the LSB having index 0
MSB_0:	Bit numbering scheme with the MSB having index 0
VeloPix Pixel:	Square, indivisible region of VeloPix sensor
Super Pixel:	2×4 pixels, acted on by a single ASIC logic unit
Super Pixel Packet: .	Data packet containing information regarding all 8 pixels in a given super pixel
BCID:	Bunch Crossing ID, the integer index of a bunch crossing event assigned to a VELO hit in the VeloPix ASIC
ASIC:	Application-Specific Integrated Circuit
FPGA:	Field-Programmable Gate Array

Table B.1

Glossary of terms related to the VELO, VeloPix sensors, and data processing boards

Field	Position		Size	Decoder enum name
	LSB_0	*MSB_0*		
64b Counter	255-192	0-63	64	`MDV2_Counter`
BCID	191-180	64-75	12	`MDV2_BCID`
Data length	179-160	76-95	20	`MDV2_Data_length`
Start of run	159	96	1	`MDV2_Start_of_run`
Trigger	158	97	1	`MDV2_Trigger`
Data flow	157	98	1	`MDV2_Data_flow`
Sync	156	99	1	`MDV2_Sync`
Info	155-124	100-131	32	`MDV2_Info`
Link number	123-120	132-135	4	`MDV2_Link_number`
SPP3	119-90	136-165	30	`MDV2_SPP3`
SPP2	89-60	166-195	30	`MDV2_SPP2`
SPP1	59-30	196-225	30	`MDV2_SPP1`
SPP0	29-0	226-255	30	`MDV2_SPP0`

Table B.2

Data format of GWT bypass frame. All positions and sizes are in bits. See table B.3 for the internal data format of GWT bypass SPPs

Field	Position		Size	Decoder enum name
	LSB_0	*MSB_0*		
End-of-Column Address	29-23	0-6	7	`FE_SPP_EoC_Addr`
SuperPixel Address	22-17	7-12	6	`FE_SPP_SP_Addr`
BCID 9b	16-8	13-21	9	`FE_SPP_BCID_9b`
Hitmap	7-0	22-29	8	`FE_SPP_Hitmap`

Table B.3

Internal data format of a front-end SPP, as output by the GWT and GWT bypass. All positions and sizes are in bits.

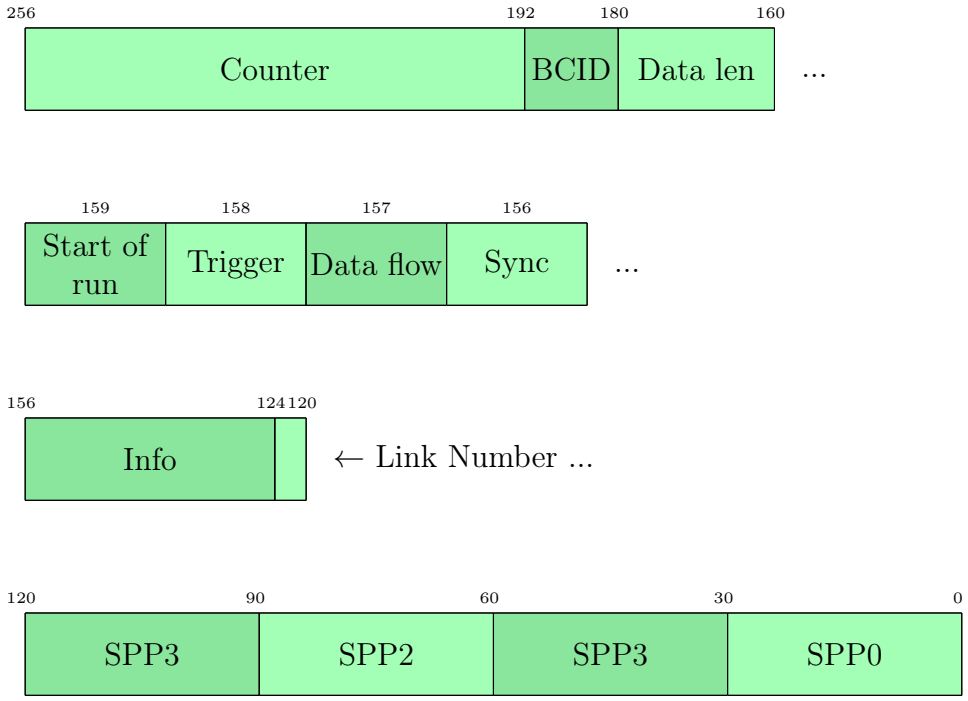


Fig. B.1

Diagram of the GWT bypass data format, split over multiple lines for readability. Bit numbers over field dividing lines denote the higher bit of the two. See figure B.2 for the internal data format of GWT bypass SPPs.

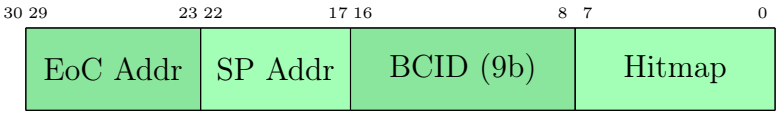


Fig. B.2

Diagram of the GWT bypass SPP data format.

0x0	BXID
0x1	TFC Sync
0x2	Chip ID
0x3	TFC Align Mode
0x4	Invalid (IDLE)

Table B.4

IDs of the 5 types of special frames in the GWT output.

0x0: BXID

Bit (LSB)	17 – 12	11 – 0
Data	0	BXID

0x1: TFC Sync

Bit (LSB)	17 – 0
Data	0x2BABE

0x2: Chip ID

Bit (LSB)	17 – 15	14 – 0
Data	0	Chip ID

0x3: TFC Align Mode

Bit (LSB)	17 – 6	5 – 0
Data	0	TFC Bits

Table B.5

The data formats of the an SPP in any of the 4 valid special frame types.

B.4, which are stored as the value of the 4-bit header in the GWT frame. Each type of frame has a configurable format described in table B.5.

B.4 | Test Pulse Timestamp Reconstruction

Below is a simplified (invalid data checks removed) listing for how to reconstruct the full timestamp from the 64-bit bypass counter and the 9-bit VeloPix truncated BCID:

get_time.c

```
uint64_t get_time(uint64_t link_counter, int spp_bcid_9b) {

  /* Subtract the calibration zero-time from the 64-bit counter
     */
  int64_t link_counter_time = link_counter - link_counter_zero;

  /* Calculate the global time from the 9b SPP BCID and
   * link counter. Only correct if an SPP is read out
   * of the link < 512 counts after hitting the SP.
   * (May not be correct at high rates).
   * In other words, assume that the full SPP time
```

```
   * is inside the 512 clock cycles prior to the
   * 64b link readout counter. This can be uniquely
   * determined by the 9b truncated SPP BCID (as 2^9=512)
   */

  int64_t link_orbit = link_counter_time / N_PER_ORBIT;
  int64_t link_bucket = link_counter_time % N_PER_ORBIT;
  int64_t spp_orbit = link_orbit;
  int bcid_offset;

  if (spp_bcid_9b > link_bucket) {
    /* If the 9b BCID is larger than the readout counter
     *  in that whole orbit, the readout must be in the next
     *  orbit relative to the spp_bcid_9b. We should be in the
     *  final (smaller) 9b bin of the previous orbit.
     */
    spp_orbit--;
    bcid_offset = N_PER_ORBIT / BCID_9B_MAX;
    int orbit_final_9b_size = N_PER_ORBIT % BCID_9B_MAX;
  } else {

    bcid_offset = (link_bucket - spp_bcid_9b) / BCID_9B_MAX;
  }
  int bcid_12 = bcid_offset*BCID_9B_MAX + spp_bcid_9b;
  uint64_t tfull_velopix = (uint64_t)(spp_orbit*N_PER_ORBIT +
   bcid_12);

  return tfull_velopix;

};
```

B.5 | Test Pulse Input Hitmap Creation

The technique for producing the test pulse input hitmaps from MC data with a more realistic distribution is as follows:

- Convert the MC hit data into a 256×256 hitmap, normalised by the number of events in the MC sample.
- Quantise the pixels in the hitmap into $i \times j$-superpixel bins.
- Choose a target maximum bin occupancy and produce a multiplier m based on the ratio between the target maximum occupancy and the highest occupancy bin in the quantised hitmap.

- Normalise the quantised hitmap so that the highest-occupancy bin matches this target.
- For each bin with an expectation occupancy of x pixels per event, randomly sample each pixel in the bin with a binomial probability of x, to produce a 256×256 binary output hitmap.
- Quantise the output hitmap into 2×4-pixel superpixels, and sum to get the total number N of active SPs in hitmap.
- Use a pulse input period of $T' = \text{ceil}(T) = \text{ceil}(N/4l)$ clock cycles, where l is the number of serialiser links that will be active during the test (each of which can output 4 SPPs per clock cycle).

For a chip rated to produce an output of $b = 4l$ SPPs per clock cycle, this process produces a hitmap and cycle period that (given a desired maximum bin occupancy) result in an expected output bandwidth of:

$$b' = \frac{N}{T'} \equiv \frac{N}{T+\epsilon}, \quad \epsilon = 1 - \frac{N\%(4l)}{4l}, \quad \frac{T}{T+1} < \frac{b'}{b} \leq 1$$

This process attempts to approximate the distribution of hits on the sensor with the rated SPP output, without requiring an overly involved simulation of how hits cluster into superpixels.

Bins of size $i = 2j$ were chosen to produce square regions on the chip (as the superpixels have size 2×4). With 64 superpixels to a chip vertically, for the bins to fit the chip, size j is constrained to be $2^n, n \in \{1, 2, 3, 4, 5, 6\}$

The larger-bin histogram is set up with a bin size such that no bin has an expected occupancy close to or below 1 over the given number of clock cycles. However, the bins should also be kept reasonably small to maintain the sample the spatial hit distribution as finely as possible. As the TFC input only supports sending a superpixel once on the given clock cycle, multiple hits in the same superpixel over the given pulse period is not possible, and so the period length should strike a balance between low statistics for cold regions, and unrealistic levels of congestion for hot regions.

With these constraints in mind, 3 sets of parameters were chosen to test the performance of the hottest chip (VP0 2 in figure 5.4) with all 4 links enabled at targeted maximum bin occupancies of 1%, 4%, and 10%:

The test pulse input hitmaps produced by these 3 sets of parameters are shown in figure B.3.

Bin Size Pixels	Target occupancy (%)	$T \to T'$ (clock cycles)	b' (SPP/clock)
16×16	1	$11.3 \to 12$	15.08
16×16	4	$47.2 \to 48$	15.73
16×16	10	$109.8 \to 110$	15.97

Table B.6

List of 3 sets of parameters for producing test pulse input hitmaps via the process described in section B.5.

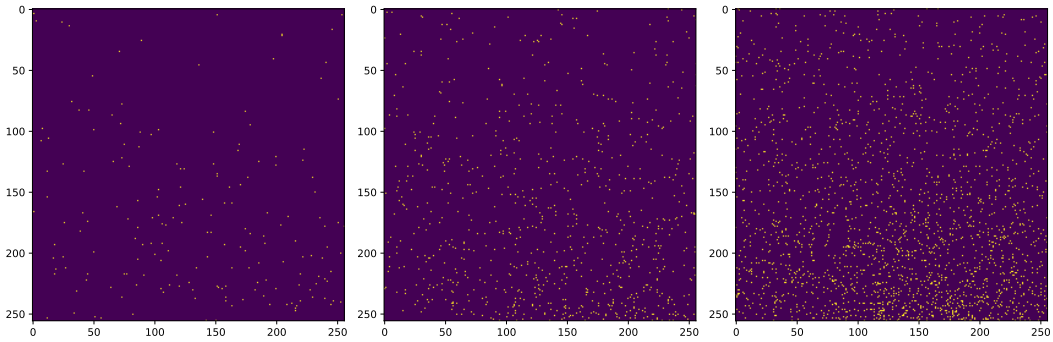


Fig. B.3

Test pulse input hitmaps produced by the first (left), second (middle) and third (right) rows of parameters in table B.6, to be tested with pulse periods of 12, 48, and 110 clock cycles respectively.

Chapter THREE

$B^0 \to K_1(1270)ll$ Sensitivity Study

C.1 | Simulation Software Versions and Data

Table C.1 outlines the various software versions and detector conditions used in the sensitivity study.

C.2 | List of Cuts For All HLT2 Lines

The tables below list all of the cuts applied to the input particles, 4-vector combinations, and vertex-fit (mother) particles in various versions of the HLT2 trigger lines used in this study. Table C.2 provides a glossary of the variable names.

C.3 | Particle Decay Properties

Particles in the decay channels were decayed by EvtGen with a relativistic Breit-Wigner lineshape, according to the respecitve known particle decay widths, as shown in figure C.1.

Pythia	8.244
PHOTOS	3.56
Geant4	v106r2p4
Gauss	v55r2
Boole	v43r0
Moore	v53r3
DDDB	upgrade/dddb-20210617
CONDDB	upgrade/sim-20210617-vc-md100, upgrade/sim-20210617-vc-mu100
ν	7.6
Events	20 000 per decay mode

Table C.1

Details of simulated Monte Carlo events used in sensitivity study

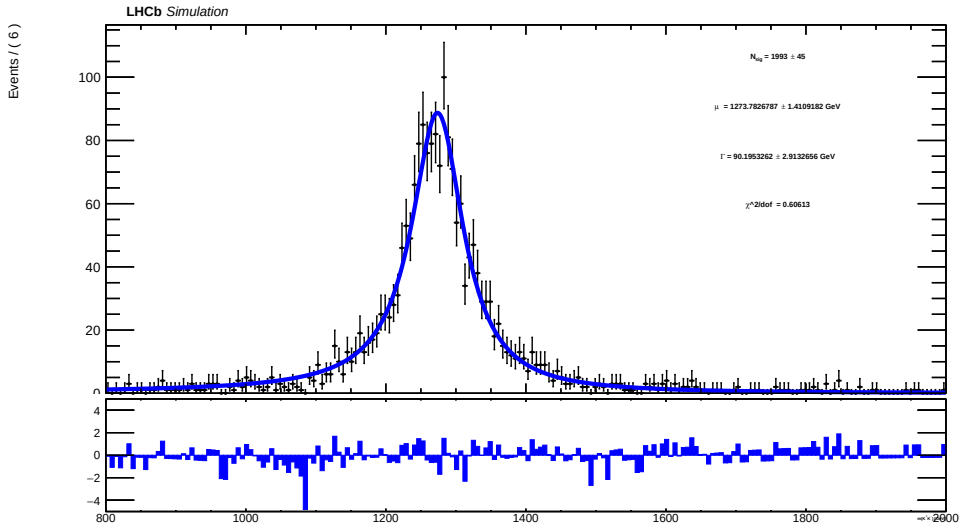


Fig. C.1

A fit of 2 000 simulated $K_1(1270)$ masses to a relativistic Breit-Wigner shape.

Variable	Description
M	Mass
M_{corr}	Corrected mass
p	Momentum
p_T	Transverse momentum
$\delta_M(K_S^0)$	Absolute difference between reconstructed mass and documented particle mass
χ^2_v/N_{DOF}	χ^2 of mother particle vertex fit, per degree of freedom
$\text{PROBNN}_{\text{ghost}}$	Probability of track being a ghost, evaluated by PROBNN
PROBNN_{π}	Probability of track being a pion, evaluated by PROBNN
PROBNN_{e}	Probability of track being an electron, evaluated by PROBNN
minPID_{μ}	Output of LHCb particle ID framework based on the particle's ID as a muon
$\chi^2_{\text{PV, min}}(\text{IP})$	Minimum χ^2 of primary vertex impact parameter fits
τ_{BPV}	Reconstructed lifetime of mother particle based on best primary vertex fit
δz_{BPV}	z displacement of mother based on best primary vertex fit
DLS_{BPV}	Decay length significance of mother based on best primary vertex fit
$\text{MAXTREE}(P, x)$	The maximum value of x for all particles of ID P in the decay tree

Table C.2

A glossary of the variables used in HLT2 cuts in this sensitivity study.

K_s^0 (LL)	
Inputs	$\pi^+\pi^-$ (LL)
π^+, π^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 36$
	$\text{PROBNN}_{\text{ghost}} < 0.4$
	$\text{PROBNN}_{\pi} > 0.5$
	$p_T > 0$
Comb	$\delta_M(K_S^0) < 50\,\text{MeV}$
Mother	$\delta_M(K_S^0) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 30$
	$\tau_{\text{BPV}} > 2.0\,\text{ps}$

K_s^0 (DD)	
Inputs	$\pi^+\pi^-$ (DD)
π^+, π^-	$\text{PROBNN}_{\text{ghost}} < 0.4$
	$\text{PROBNN}_{\pi} > 0.5$
	$p > 3000\,\text{MeV}$
	$p_T > 175\,\text{MeV}$
Comb	$\delta_M(K_S^0) < 80\,\text{MeV}$
Mother	$\delta_M(K_S^0) < 64\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 30$
	$\delta z_{\text{BPV}} > 400\,\text{mm}$

Loose K_s^0 (LL)	
Inputs	$\pi^+\pi^-$ (LL)
π^+, π^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 36$
	$\text{PROBNN}_{\text{ghost}} < 0.8$
	$\text{PROBNN}_{\pi} > 0.2$
	$p_T > 0$
Comb	$\delta_M(K_S^0) < 50\,\text{MeV}$
Mother	$\delta_M(K_S^0) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 30$
	$\tau_{\text{BPV}} > 2.0\,\text{ps}$

Loose K_s^0 (DD)	
Inputs	$\pi^+\pi^-$ (DD)
π^+, π^-	$\text{PROBNN}_{\text{ghost}} < 0.8$
	$\text{PROBNN}_{\pi} > 0.2$
	$p > 3000\,\text{MeV}$
	$p_T > 175\,\text{MeV}$
Comb	$\delta_M(K_S^0) < 80\,\text{MeV}$
Mother	$\delta_M(K_S^0) < 64\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 30$
	$\delta z_{\text{BPV}} > 400\,\text{mm}$

Table C.3

List of the daughter, combination, and mother cuts for the various K_s^0 candidates used in the HLT2 lines.

$K_1(1270)$ (LL or DD)	
Inputs	$\pi^+\pi^-$ (LL), K_s^0 (LL or DD)
π^+, π^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 8$
	$\text{PROBNN}_{\text{ghost}} < 0.4$
	$\text{PROBNN}_{\pi} > 0.5$
	$p_T > 200\,\text{MeV}$
K_s^0	$480\,\text{MeV} < M < 510\,\text{MeV}$
	$p_T > 500\,\text{MeV}$
Comb	$\delta_M(K_1(1270)) < 550\,\text{MeV}$
Mother	$p_T > 2000\,\text{MeV}$
	$\delta_M(K_S^0) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 10$

Loose $K_1(1270)$ (LL or DD)	
Inputs	$\pi^+\pi^-$ (LL), Loose K_s^0 (LL or DD)
π^+, π^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 6$
	$\text{PROBNN}_{\text{ghost}} < 0.8$
	$\text{PROBNN}_{\pi} > 0.2$
	$p_T > 150\,\text{MeV}$
K_s^0	$480\,\text{MeV} < M < 510\,\text{MeV}$
	$p_T > 400\,\text{MeV}$
Comb	$\delta_M(K_1(1270)) < 550\,\text{MeV}$
Mother	$p_T > 15000\,\text{MeV}$
	$\delta_M(K_S^0) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 20$

Table C.4

List of the daughter, combination, and mother cuts for the various $K_1(1270)$ candidates used in the HLT2 lines.

Resonant Dimuon ($J/\psi(1S)$)	
Inputs	$\mu^+\mu^-$
μ^+, μ^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 0$
	$\text{PROBNN}_{\text{ghost}} < 0.4$
	$\text{minPID}_{\mu} > -5$
	$p_T > 300\,\text{MeV}$
Comb	$\delta_M(J/\psi(1S)) < 120\,\text{MeV}$
Mother	$\chi^2_v/N_{\text{DOF}} < 25$
	$\text{DLS}_{\text{BPV}} < 3$

Non-resonant Dimuon	
Inputs	$\mu^+\mu^-$
μ^+, μ^-	$\chi^2_{\text{PV, min}}(\text{IP}) > 25$
	$\text{PROBNN}_{\text{ghost}} < 0.4$
	$\text{minPID}_{\mu} > -5$
	$p_T > 300\,\text{MeV}$
Comb	–
Mother	$\chi^2_v/N_{\text{DOF}} < 25$
	$\text{DLS}_{\text{BPV}} < 9$

Table C.5

List of the daughter, combination, and mother cuts for the various dimuon candidates used in the HLT2 lines.

Resonant ee ($J/\psi(1S)$), No Brem	
Inputs	e^+e^-
e^+, e^-	$\text{PROBNN}_e > 0.75$
	$p_T > 200\,\text{MeV}$
Comb	–
Mother	$10\,\text{MeV} < M < 3200\,\text{MeV}$

Non-resonant ee, No Brem	
Inputs	e^+e^-
e^+, e^-	$\text{PROBNN}_e > 0.75$
	$p_T > 200\,\text{MeV}$
Comb	–
Mother	$50\,\text{MeV} < M < 5000\,\text{MeV}$

Resonant ee ($J/\psi(1S)$), With Brem	
Inputs	e^+e^-
e^+, e^-	$\text{PROBNN}_e > 0.75$
	$p_T > 200\,\text{MeV}$
Comb	–
Mother	$M > 10\,\text{MeV}$
	$M < (M_{J/\psi} + 120\,\text{MeV})$

Non-resonant ee, With Brem	
Inputs	e^+e^-
e^+, e^-	$\text{PROBNN}_e > 0.75$
	$p_T > 200\,\text{MeV}$
Comb	–
Mother	$50\,\text{MeV} < M < 5000\,\text{MeV}$

Table C.6

List of the daughter, combination, and mother cuts for the various dielectron candidates used in the HLT2 lines.

B^0 (LL or DD, ee or $\mu\mu$)	
Inputs	$K_1(1270)$ (LL or DD)
	l^+l^- (ee or $\mu\mu$),
Comb	$3000\,\text{MeV} < M < 7000\,\text{MeV}$
Mother	$\chi^2_{\text{PV, min}}(\text{IP}) < 50$
	$3500\,\text{MeV} < M_{\text{corr}} < 7000\,\text{MeV}$
	$\text{MAXTREE}(\mu^-, \text{PROBNN}_{\text{ghost}}) < 0.8$
	$\text{MAXTREE}(\pi^+, \text{PROBNN}_{\text{ghost}}) < 0.8$
	$\delta_M(K^0_S) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 20$

Loose B^0 (LL or DD, ee or $\mu\mu$)	
Inputs	$K_1(1270)$ (LL or DD)
	l^+l^- (ee or $\mu\mu$),
Comb	$3000\,\text{MeV} < M < 8000\,\text{MeV}$
Mother	$\chi^2_{\text{PV, min}}(\text{IP}) < 200$
	$3500\,\text{MeV} < M_{\text{corr}} < 7500\,\text{MeV}$
	$\text{MAXTREE}(\mu^-, \text{PROBNN}_{\text{ghost}}) < 0.9$
	$\text{MAXTREE}(\pi^+, \text{PROBNN}_{\text{ghost}}) < 0.9$
	$\delta_M(K^0_S) < 35\,\text{MeV}$
	$\chi^2_v/N_{\text{DOF}} < 50$

Table C.7

List of the daughter, combination, and mother cuts for the various B^0 candidates used in the HLT2 lines.

Decay Type	Passthrough Rate
$\mu\mu$, non-resonant	66.4%
$\mu\mu$, resonant	65.5%
ee, non-resonant	65.6%
ee, resonant	66.0%

Table C.8

A list of the HLT1 global event cut (GEC) passthrough rates for each of the four decays.

Chapter FOUR

Auxiliary Work

Listed below are auxiliary duties that I undertook during my time of study, which were partially or tangentially related to subjects of this thesis, and are referenced in passing from other chapters.

D.1 | Industrial Placement

As part of the terms of the studentship, 6 months were spent on an industrial placement at a machine learning startup company.

The placement involved co-developing an interactive machine learning application, designed as separable services to be deployed at scale on cloud infrastructure. Software was written to automatically extract, categorise, link, and preprocess textual information from arbitrarily structured technical documents, which was used to train a natural language model that provides automated insight into new documents. A variety of language models were probed, including k-means clustering, word embedding, TFIDF (term frequency, inverse-document frequency) and Siamese neural networks.

D.2 | CERN Student Mentoring

Whilst working at CERN during 2019, part of the time was committed to mentoring and supervising a CERN summer school student, who was tasked with a project on the LHCb VELO upgrade.

The project involved debugging aspects of the VELO readout system to ensure correctness. The software decoder described in section 5.5 was used to analyse the sync frames coming from the readout, in terms of their number and timing. Work was also done to augment the noise analysis code in the ASIC pixel equalisation process.

Bibliography

[1] W. Pauli, "Über den zusammenhang des abschlusses der elektronengruppen im atom mit der komplexstruktur der spektren", Zeitschrift für Physik **31**, 765–783 (1925).

[2] *On the connexion between the completion of electron groups in an atom with the complex structure of spectra*, (2010) http://www.fisicafundamental.net/relicario/doc/Pauli_1925.pdf.

[3] P. W. Higgs, "Broken Symmetries and the Masses of Gauge Bosons", Phys. Rev. Lett. **13**, edited by J. C. Taylor, 508–509 (1964) 10.1103/PhysRevLett.13.508.

[4] F. Englert and R. Brout, "Broken Symmetry and the Mass of Gauge Vector Mesons", Phys. Rev. Lett. **13**, edited by J. C. Taylor, 321–323 (1964) 10.1103/PhysRevLett.13.321.

[5] G. S. Guralnik et al., "Global Conservation Laws and Massless Particles", Phys. Rev. Lett. **13**, edited by J. C. Taylor, 585–587 (1964) 10.1103/PhysRevLett.13.585.

[6] M. Peskin, *An introduction to quantum field theory* (1995), https://doi.org/10.1201/9780429503559.

[7] J. Bjorken and S. Drell, *Relativistic quantum mechanics*, International series in pure and applied physics (McGraw-Hill, 1964).

[8] A. Banfi et al., *Lecture Notes for the 2017 School for Experimental High Energy Physics Students* (2017).

[9] A. W. Knapp and A. Knapp, *Lie groups beyond an introduction*, Vol. 140 (Springer, 1996).

[10] C. Becchi et al., "Renormalization of gauge theories", Ann. Phys. (N.Y.); (United States), 10.1016/0003-4916(76)90156-1 (1976) 10.1016/0003-4916(76)90156-1.

[11] G. 't Hooft and M. J. G. Veltman, "Regularization and Renormalization of Gauge Fields", Nucl. Phys. B **44**, 189–213 (1972) 10.1016/0550-3213(72)90279-9.

[12] K. Symanzik, "Small distance behavior in field theory and power counting", Commun. Math. Phys. **18**, 227–246 (1970) 10.1007/BF01649434.

[13] A. Deur et al., "The QCD Running Coupling", Nucl. Phys. **90**, 1 (2016) 10.1016/j.ppnp.2016.04.003.

[14] P. A. M. Dirac, "The quantum theory of the electron", Proc. Roy. Soc. Lond. A **117**, 610–624 (1928) 10.1098/rspa.1928.0023.

[15] H. Dreiner et al., *Supersymmetry* (2004).

[16] M. Gell-Mann, "Symmetries of baryons and mesons", Phys. Rev. **125**, 1067–1084 (1962) 10.1103/PhysRev.125.1067.

[17] D. J. Gross and F. Wilczek, "Ultraviolet Behavior of Nonabelian Gauge Theories", Phys. Rev. Lett. **30**, edited by J. C. Taylor, 1343–1346 (1973) 10.1103/PhysRevLett.30.1343.

[18] M. Kobayashi, "Nobel lecture: cp violation and flavor mixing", Reviews of Modern Physics **81**, 1019 (2009).

[19] T. Maskawa, "Nobel lecture: what does cp violation tell us?", Reviews of Modern Physics **81**, 1027 (2009).

[20] P. A. Zyla et al. (Particle Data Group), "Review of Particle Physics", PTEP **2020**, 083C01 (2020) 10.1093/ptep/ptaa104.

[21] N. Cabibbo, "Unitary Symmetry and Leptonic Decays", Phys. Rev. Lett. **10**, 531–533 (1963) 10.1103/PhysRevLett.10.531.

[22] L. Wolfenstein, "Parametrization of the Kobayashi-Maskawa Matrix", Phys. Rev. Lett. **51**, 1945 (1983) 10.1103/PhysRevLett.51.1945.

[23] A. J. Buras, "Unitarity triangle: 2002 and beyond", in 14th Rencontres de Blois on Matter - Anti-matter Asymmetry (Oct. 2002).

[24] J. Charles et al., "Current status of the Standard Model CKM fit and constraints on $\Delta F = 2$ New Physics", Phys. Rev. D **91**, 073007 (2015) 10.1103/PhysRevD.91.073007.

[25] by LHCb collaboration, "Matter-antimatter trigonometry with LHCb", (2015).

[26] R. Aaij et al. (LHCb), "Measurement of CP violation in $B^0 \to J/\psi K^0_S$ decays", Phys. Rev. Lett. **115**, 031601 (2015) 10.1103/PhysRevLett.115.031601.

[27] C. S. Wu et al., "Experimental test of parity conservation in beta decay", Phys. Rev. **105**, 1413–1415 (1957) 10.1103/PhysRev.105.1413.

[28] C. S. Wu et al., "Experimental test of parity conservation in beta decay", Phys. Rev. **105**, 1413–1415 (1957) 10.1103/PhysRev.105.1413.

[29] J. H. Christenson et al., "Evidence for the 2π Decay of the K^0_2 Meson", Phys. Rev. Lett. **13**, 138–140 (1964) 10.1103/PhysRevLett.13.138.

[30] R. Aaij et al. (LHCb), "Measurement of the flavour-specific CP-violating asymmetry a^s_{sl} in B^0_s decays", Phys. Lett. B **728**, 607–615 (2014) 10.1016/j.physletb.2013.12.030.

[31] R. Aaij et al. (LHCb), "First observation of CP violation in the decays of B^0_s mesons", Phys. Rev. Lett. **110**, 221601 (2013) 10.1103/PhysRevLett.110.221601.

[32] C. Jarlskog, "Commutator of the quark mass matrices in the standard electroweak model and a measure of maximal CP nonconservation", Phys. Rev. Lett. **55**, 1039–1042 (1985) 10.1103/PhysRevLett.55.1039.

[33] A. D. Sakharov, "Violation of CP Invariance, C asymmetry, and baryon asymmetry of the universe", Pisma Zh. Eksp. Teor. Fiz. **5**, [Usp. Fiz. Nauk161,no.5,61(1991)], 32–35 (1967) 10.1070/PU1991v034n05ABEH002497.

[34] R. P. Feynman and M. Gell-Mann, "Theory of the fermi interaction", Phys. Rev. **109**, 193–198 (1958) 10.1103/PhysRev.109.193.

[35] L. Maiani, "Cp violation in purely lefthanded weak interactions", Physics Letters B **62**, 183–186 (1976) https://doi.org/10.1016/0370-2693(76)90500-1.

[36] R. Barbieri, *Ten Lectures on the ElectroWeak Interactions* (Scuola Normale Superiore, 2007).

[37] R. N. Mohapatra and J. C. Pati, "Left-Right Gauge Symmetry and an Isoconjugate Model of CP Violation", Phys. Rev. D **11**, 566–571 (1975) 10.1103/PhysRevD.11.566.

[38] G. Senjanovic and R. N. Mohapatra, "Exact Left-Right Symmetry and Spontaneous Violation of Parity", Phys. Rev. D **12**, 1502 (1975) 10.1103/PhysRevD.12.1502.

[39] A. Djouadi, "The anatomy of electro-weak symmetry breaking. i: the higgs boson in the standard model", Physics Reports **457**, 1–216 (2005).

[40] G. Aad et al. (ATLAS), "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC", Phys. Lett. B **716**, 1–29 (2012) 10.1016/j.physletb.2012.08.020.

[41] S. Chatrchyan et al. (CMS), "Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC", Phys. Lett. B **716**, 30–61 (2012) 10.1016/j.physletb.2012.08.021.

[42] L. Morel et al., "Determination of the fine-structure constant with an accuracy of 81 parts per trillion", Nature **588**, 61–65 (2020) 10.1038/s41586-020-2964-7.

[43] D. F. Litim, "Renormalisation group and the Planck scale", Phil. Trans. Roy. Soc. Lond. A **369**, 2759–2778 (2011) 10.1098/rsta.2011.0103.

[44] B. Abi et al. (Muon g-2), "Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm", Phys. Rev. Lett. **126**, 141801 (2021) 10.1103/PhysRevLett.126.141801.

[45] R. Aaij et al. (LHCb), "Test of lepton universality in beauty-quark decays", Nature Phys. **18**, 277–282 (2022) 10.1038/s41567-021-01478-8.

[46] M. Breidenbach et al., "Observed behavior of highly inelastic electron-proton scattering", Phys. Rev. Lett. **23**, 935–939 (1969) 10.1103/PhysRevLett.23.935.

[47] T. Affolder et al. (CDF), "Charged Jet Evolution and the Underlying Event in $p\bar{p}$ Collisions at 1.8 TeV", Phys. Rev. D **65**, 092002 (2002) 10.1103/PhysRevD.65.092002.

[48] J. D. Osborn (LHCb), "Jet hadronization at LHCb", PoS **High-pT2019**, 013 (2020) 10.22323/1.355.0013.

[49] B. Andersson et al., "Parton fragmentation and string dynamics", Physics Reports **97**, 31–145 (1983) https://doi.org/10.1016/0370-1573(83)90080-7.

[50] X. Artru and G. Mennessier, "String model and multiproduction", Nucl. Phys. B **70**, 93–115 (1974) 10.1016/0550-3213(74)90360-5.

[51] C. Bierlich, *Lund yo-yo model diagram*, https://particle.wiki/wiki/File:Lund-yoyo.png.

[52] Y. Ilchenko, "Measurements of the top quark mass and decay width with the D0 detector", in Meeting of the APS Division of Particles and Fields (Nov. 2011).

[53] T. Sjöstrand et al., "Pythia 6.4 physics and manual", Journal of High Energy Physics **2006**, 026–026 (2006) 10.1088/1126-6708/2006/05/026.

[54] T. Sjöstrand et al., "An introduction to pythia 8.2", Computer physics communications **191**, 159–177 (2015).

[55] *Pythia fragmentation documentation*, http://atlas.physics.arizona.edu/~shupe/Pythia8/pythia8223/share/Pythia8/htmldoc/Fragmentation.html.

[56] M. G. Bowler, "e+ e- Production of Heavy Quarks in the String Model", Z. Phys. C **11**, 169 (1981) 10.1007/BF01574001.

[57] G. Aad et al. (ATLAS), "Determination of the ratio of b-quark fragmentation fractions f_s/f_d in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector", Phys. Rev. Lett. **115**, 262001 (2015) 10.1103/PhysRevLett.115.262001.

[58] LHCb Collaboration, "The LHCb Detector at the LHC", JINST **3**, S08005 (2008) https://doi.org/10.1088/1748-0221/3/08/S08005.

[59] R. Aaij et al. (LHCb), "Measurement of CP observables in $B^\pm \to D^{(*)}K^\pm$ and $B^\pm \to D^{(*)}\pi^\pm$ decays using two-body D final states", JHEP **04**, 081 (2021) 10.1007/JHEP04(2021)081.

[60] S. L. Glashow et al., "Weak Interactions with Lepton-Hadron Symmetry", Phys. Rev. D **2**, 1285–1292 (1970) 10.1103/PhysRevD.2.1285.

[61] J. Ellis et al., "The phenomenology of the next left-handed quarks", Nuclear Physics B **131**, 285–307 (1977) https://doi.org/10.1016/0550-3213(77)90374-1.

[62] R. Aaij et al. (LHCb), "Test of lepton universality with $B^0 \to K^{*0}\ell^+\ell^-$ decays", JHEP **08**, 055 (2017) 10.1007/JHEP08(2017)055.

[63] M. Nicol (LHCb), "Photon polarisation in b->s gamma using B->K*e+e- at LHCb", EPJ Web Conf. **28**, edited by G. Bernardi et al., 12028 (2012) 10.1051/epjconf/20122812028.

[64] R. Aaij et al. (LHCb), "Measurement of the forward-backward asymmetry in $Z/\gamma^* \to \mu^+\mu^-$ decays and determination of the effective weak mixing angle", JHEP **11**, 190 (2015) 10.1007/JHEP11(2015)190.

[65] J. Gratrex and R. Zwicky, "Parity doubling as a tool for right-handed current searches", Journal of High Energy Physics **2018**, 10.1007/jhep08(2018)178 (2018) 10.1007/jhep08(2018)178.

[66] LHCb Collaboration, *LHCb reoptimized detector design and performance: Technical Design Report*, Technical Design Report LHCb (CERN, Geneva, 2003).

[67] LHCb Collaboration (LHCb Collaboration), *LHCb VELO (VErtex LOcator): Technical Design Report*, Technical Design Report LHCb (CERN, Geneva, 2001).

[68] J. Seguinot and T. Ypsilantis, "Photo-ionization and cherenkov ring imaging", Nuclear Instruments and Methods **142**, 377–391 (1977) https://doi.org/10.1016/0029-554X(77)90671-1.

[69] M. Alemi et al., "First operation of a hybrid photon detector prototype with electrostatic cross-focussing and integrated silicon pixel readout", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **449**, 48–59 (2000) https://doi.org/10.1016/S0168-9002(99)01448-5.

[70] LHCb Collaboration (LHCb Collaboration), *LHCb magnet: Technical Design Report*, Technical design report. LHCb (CERN, Geneva, 2000).

[71] LHCb Collaboration, *Lhcb magnet web page*, https://lhcb.github.io/starterkit-lessons/first-analysis-steps/.

[72] M. Needham and D. Volyanskyy, *Updated geometry description for the LHCb Trigger Tracker*, tech. rep. (CERN, Geneva, June 2006).

[73] M. Gupta, *Calculation of radiation length in materials*, tech. rep. (CERN, Geneva, July 2010).

[74] *LHCb Trigger and Online Upgrade Technical Design Report*, tech. rep. CERN-LHCC-2014-016. LHCB-TDR-016 (May 2014).

[75] *Diagram of the upgrade lhcb detector*, https://twiki.cern.ch/twiki/pub/LHCb/LHCbUpgrade/upgrade-detector.jpg.

[76] P. Collins (LHCb), "The LHCb Upgrade", in 9th Conference on Flavor Physics and CP Violation (Aug. 2011).

[77] *Letter of Intent for the LHCb Upgrade*, tech. rep. (CERN, Geneva, Mar. 2011).

[78] LHCb Collaboration, *LHCb PID Upgrade Technical Design Report*, tech. rep. (Nov. 2013).

[79] LHCb Collaboration, *LHCb Tracker Upgrade Technical Design Report*, tech. rep. (Feb. 2014).

[80] I. Belyaev et al. (LHCb), "Handling of the generation of primary events in Gauss, the LHCb simulation framework", in 2010 IEEE Nuclear Science Symposium, Medical Imaging Conference, and 17th Room Temperature Semiconductor Detectors Workshop (2010), pp. 1155–1161, 10.1109/NSSMIC.2010.5873949.

[81] D. J. Lange, "The EvtGen particle decay simulation package", Nucl. Instrum. Meth. A **462**, edited by S. Erhan et al., 152–155 (2001) 10.1016/S0168-9002(01)00089-4.

[82] S. Agostinelli et al., "Geant4—a simulation toolkit", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506**, 250–303 (2003) https://doi.org/10.1016/S0168-9002(03)01368-8.

[83] J. Allison et al., "Geant4 developments and applications", IEEE Transactions on Nuclear Science **53**, 270–278 (2006) 10.1109/TNS.2006.869826.

[84] J. Allison et al., "Recent developments in geant4", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **835**, 186–225 (2016) https://doi.org/10.1016/j.nima.2016.06.125.

[85] G. Corti et al., "Software for the LHCb experiment", IEEE Trans. Nucl. Sci. **53**, 1323–1328 (2006).

[86] "Performance of the GPU HLT1 (Allen)", (2020).

[87] R. Aaij et al. (LHCb), "Precision luminosity measurements at LHCb", JINST **9**, P12005 (2014) 10.1088/1748-0221/9/12/P12005.

[88] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Journal of Basic Engineering **82**, 35–45 (1960) 10.1115/1.3662552.

[89] R. Hierk et al., "Performance of the LHCb 00 track fitting software", (2000).

[90] LHCb Collaboration, *LHCb trigger system: Technical Design Report*, Technical Design Report LHCb, revised version number 1 submitted on 2003-09-24 12:12:22 (CERN, Geneva, 2003).

[91] *Lhcb long-term luminosity schedule*, https://lhc-commissioning.web.cern.ch/lhc-commissioning/schedule/LHC-long-term.htm.

[92] A. Airapetian et al., "ATLAS: Detector and physics performance technical design report. Volume 1", (1999).

[93] L. Evans and P. Bryant, "Lhc machine", JINST **3**, S08001 (2008) https://doi.org/10.1088/1748-0221/3/08/S08001.

[94] R. Alemany-Fernandez et al., "The LHCB Online Luminosity Control and Monitoring", 3 p (2013).

[95] *Trigger schemes*, http://lhcb.web.cern.ch/lhcb/speakersbureau/html/TriggerScheme.html.

[96] G. B. et al, "Gaudi — a software architecture and framework for building hep data processing applications", Computer Physics Communications **140**, 45–55 (2001) 10.1016/s0010-4655(01)00254-5.

[97] LHCb Collaboration, *Lhcb starter kit*, (2017) https://lhcb.github.io/starterkit-lessons/first-analysis-steps/.

[98] A. Maier, "Ganga — a job management and optimising tool", Journal of Physics: Conference Series **119** (2008).

[99] C. Council, "Proposal for building the lhc computing environment at cern", CERN/2379/Rev (2001).

[100] LHC Computing Grid, *LHC computing Grid: Technical Design Report. Version 1.06 (20 Jun 2005)*, Technical Design Report LCG (CERN, Geneva, 2005).

[101] A. Paszke et al., "Automatic differentiation in pytorch", (2017).

[102] LHCb Collaboration, *LHCb VELO Upgrade Technical Design Report*, tech. rep. CERN-LHCC-2013-021. LHCB-TDR-013 (Nov. 2013).

[103] R. Aaij et al., "Performance of the LHCb Vertex Locator", JINST **9**, P09007 (2014) 10.1088/1748-0221/9/09/P09007.

[104] E. Buchanan, "The LHCb Vertex Locator (VELO) Pixel Detector Upgrade", JINST **12**, C01013. 10 (2017) 10.1088/1748-0221/12/01/C01013.

[105] *Lhcb velo conference plots*, https://lbtwiki.cern.ch/bin/view/VELO/VELOConferencePlots.

[106] A. Mapelli et al., "Micro-channel cooling for high-energy physics particle detectors and electronics", in 13th intersociety conference on thermal and thermomechanical phenomena in electronic systems (2012), pp. 677–683, 10.1109/ITHERM.2012.6231493.

[107] T. Poikela et al., "VeloPix: the pixel ASIC for the LHCb upgrade", JINST **10**, C01057 (2015) 10.1088/1748-0221/10/01/C01057.

[108] T. Poikela et al., "Timepix3: a 65k channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout", Journal of Instrumentation **9**, C05013–C05013 (2014) 10.1088/1748-0221/9/05/c05013.

[109] V. Gromov et al., "Development of a low power 5.12 Gbps data serializer and wireline transmitter circuit for the VeloPix chip", JINST **10**, C01054 (2015) 10.1088/1748-0221/10/01/C01054.

[110] T. S. Poikela, "Readout Architecture for Hybrid Pixel Readout Chips", Presented 15 Jun 2015 (Apr. 2015).

[111] M. Katevenis et al., "Weighted round-robin cell multiplexing in a general-purpose atm switch chip", IEEE Journal on Selected Areas in Communications **9**, 1265–1279 (1991) 10.1109/49.105173.

[112] M. Bellato et al., "A PCIe Gen3 based readout for the LHCb upgrade", J. Phys.: Conf. Ser. **513**, 012023. 6 p (2013) 10.1088/1742-6596/513/1/012023.

[113] F. Alessio and R. Jacobsson, *System-level Specifications of the Timing and Fast Control system for the LHCb Upgrade*, tech. rep. (CERN, Geneva, Feb. 2014).

[114] P. Moreira et al., "The GBT Project", 10.5170/CERN-2009-006.342 (2009) 10.5170/CERN-2009-006.342.

[115] L. Eklund et al., *The VELO optical and power board*, tech. rep. (CERN, Geneva, Dec. 2021), 10.17181/CERN.TIOS.BTPB.

[116] C. Soós et al., "System-level testing of the Versatile Link components", JINST **8**, C12044 (2013) 10.1088/1748-0221/8/12/C12044.

[117] J. Visser et al., "SPIDR: a read-out system for Medipix3 & Timepix3", JINST **10**, C12028 (2015) 10.1088/1748-0221/10/12/C12028.

[118] A. Giraudo, Internal presentation, 2019.

[119] H.-Y. Cheng, "Revisiting Axial-Vector Meson Mixing", Phys. Lett. B **707**, 116–120 (2012) 10.1016/j.physletb.2011.12.013.

[120] R. Aaij et al. (LHCb), "Measurement of the polarization amplitudes in $B^0 \to J/\psi K^*(892)^0$ decays", Phys. Rev. D **88**, 052002 (2013) 10.1103/PhysRevD.88.052002.

[121] V. Battista et al., *A study of spillover clusters and ghost tracks in the Silicon Tracker with 25 ns bunch spacing*, tech. rep. (CERN, Geneva, Feb. 2016).

[122] P. Golonka and Z. Was, "Photos monte carlo: a precision tool for qed correctionsin z and w decays", The European Physical Journal C **45**, 97–107 (2006) 10.1140/epjc/s2005-02396-4.

[123] R. Aaij et al. (LHCb), "Measurement of the b-quark production cross-section in 7 and 13 TeV pp collisions", Phys. Rev. Lett. **118**, [Erratum: Phys.Rev.Lett. 119, 169901 (2017)], 052002 (2017) 10.1103/PhysRevLett.118.052002.

[124] R. Aaij et al. (LHCb), "Measurement of b hadron fractions in 13 TeV pp collisions", Phys. Rev. D **100**, 031102 (2019) 10.1103/PhysRevD.100.031102.

[125] R. Aaij et al. (LHCb), "Measurement of the inelastic pp cross-section at a centre-of-mass energy of 13 TeV", JHEP **06**, 100 (2018) 10.1007/JHEP06(2018)100.

[126] R. Aaij et al. (LHCb), "Tests of lepton universality using $B^0 \to K^0_S \ell^+ \ell^-$ and $B^+ \to K^{*+} \ell^+ \ell^-$ decays", (2021).

[127] K. Abe et al. (SLD), "A Measurement of R(b) using a vertex mass tag", Phys. Rev. Lett. **80**, 660–665 (1998) 10.1103/PhysRevLett.80.660.

[128] G. Lebanon and J. Lafferty, "Hyperplane margin classifiers on the multinomial manifold", Proceedings of the Twenty-first International Conference on Machine Learning, 66 (2004).

[129] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Annals of Eugenics **7**, 179–188 (1936) https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.

[130] B. E. Boser et al., "A training algorithm for optimal margin classifiers", in Proceedings of the fifth annual workshop on computational learning theory (ACM, 1992), pp. 144–152.

[131] C. Cortes and V. Vapnik, "Support-vector networks", Machine learning **20**, 273–297 (1995).

[132] K. Pearson, "On lines and planes of closest fit to systems of points in space", The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 559–572 (1901) 10.1080/14786440109462720.

[133] A. M. Martinez and A. C. Kak, "Pca versus lda", IEEE Transactions on Pattern Analysis and Machine Intelligence **23**, https://doi.org/10.1109/34.908974 (2001) https://doi.org/10.1109/34.908974.

[134] H. Drucker and C. Cortes, "Boosting decision trees", in Advances in neural information processing systems (1996), pp. 479–485.

[135] D. E. Rumelhart et al., "Learning representations by back-propagating errors", nature **323**, 533 (1986).

[136] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model", Neural networks **1**, 339–356 (1988).

[137] Y. Bengio et al., "Learning long-term dependencies with gradient descent is difficult", IEEE transactions on neural networks **5**, 157–166 (1994).

[138] R. Pascanu et al., "On the difficulty of training recurrent neural networks", in International conference on machine learning (2013), pp. 1310–1318.

[139] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation **9**, 1735–1780 (1997).

[140] K. Cho et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation", arXiv preprint arXiv:1406.1078 (2014).

[141] I. Sutskever et al., "Sequence to sequence learning with neural networks", in Advances in neural information processing systems (2014), pp. 3104–3112.

[142] O. Vinyals et al., "Order matters: sequence to sequence for sets", eprint arXiv:1511.06391 (2015).

[143] Y. LeCun, Y. Bengio, et al., "Convolutional networks for images, speech, and time series", The handbook of brain theory and neural networks **3361**, 1995 (1995).

[144] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2017.

[145] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift", eprint arXiv:1502.03167 (2015).

[146] G. Cybenko, "Approximation by superpositions of a sigmoidal function", Mathematics of control, signals and systems **2**, 303–314 (1989).

[147] K. Hornik et al., "Multilayer feedforward networks are universal approximators", Neural networks **2**, 359–366 (1989).

[148] K. Hornik, "Approximation capabilities of multilayer feedforward networks", Neural networks **4**, 251–257 (1991).

[149] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", IEEE transactions on evolutionary computation **1**, 67–82 (1997).

[150] V. V. Gligorov and M. Williams, "Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree", Journal of Instrumentation **8**, P02013 (2013) https://doi.org/10.1088/1748-0221/8/02/P02013.

[151] S. Chang et al., "What is the machine learning?", Physical Review D **97**, 056009 (2018) https://doi.org/10.1103/PhysRevD.97.056009.

[152] S. Benson and K. Gizdov, "Hepdrone: a toolkit for the mass application of machine learning in high energy physics", eprint arXiv:1712.09114, https://doi.org/10.1088/1748-0221/8/02/P02013 (2017) https://doi.org/10.1088/1748-0221/8/02/P02013.

[153] Martín Abadi et al., *TensorFlow: large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015.

[154] F. Chollet et al., *Keras*, https://keras.io, 2015.

[155] Google AI, *Eager execution: an imperative, define-by-run interface to tensorflow*, (2017) https://ai.googleblog.com/2017/10/eager-execution-imperative-define-by.html (visited on 10/31/2017).

[156] The PyTorch Team, *The road to 1.0: production ready pytorch*, (2018) https://pytorch.org/2018/05/02/road-to-1.0.html (visited on 05/02/2018).

[157] A. Hoecker et al., "TMVA: Toolkit for Multivariate Data Analysis", PoS **ACAT**, 040 (2007).

[158] T. Fawcett, *ROC Graphs: Notes and Practical Considerations for Researchers*, tech. rep. (2004).