



AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Field of Science: Natural Sciences

Scientific discipline: Physical Sciences

DOCTORAL THESIS

Design and implementation of the monitoring and analysis software platform for upgraded LHCb VELO detector and search for a new fitting method based on computational intelligence approach for $B \rightarrow Dh$ decays.

Author: Paweł Kopciewicz

Supervisor: prof. dr hab. inż. Tomasz Szumlak

Completed in: AGH University of Science and Technology, Department of Particle Interactions and Detection Techniques, Krakow, 2022

Declaration of the Author of this dissertation

Aware of legal responsibility for making untrue statements I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

signature

date

Declaration of the Supervisor of this dissertation

This dissertation is ready to be reviewed.

signature of thesis Supervisor

date

Contents

Abstract (English version)	5
Abstract (Polish version)	6
Acknowledgments	7
Contribution	8
Acronyms	11
1 Introduction	13
2 Computational Intelligence	15
2.1 Classification	15
2.2 Neural and deep models	18
2.3 Intelligent optimization	21
2.4 Generative Adversarial Networks	22
2.5 Learning by way of optimization	24
3 The LHCb experiment	26
3.1 The LHCb spectrometer	27
3.2 The Upgrade I	32
3.3 Upgrade I for Vertex Locator	35
4 Software for Upgrade I VELO	48
4.1 The LHCb ecosystem	48
4.2 The decoding	51
4.3 The VeloPix calibration	56
4.4 Monitoring	62
4.5 The signal scans	65
4.6 Data emulation	66

5 Pixel detector studies	68
5.1 Time-over-Threshold analysis	68
5.2 Test Pulse scans	81
5.3 Analog optimization	85
6 Machine Learning for LHCb	93
6.1 Recent Applications	93
6.2 Proposed concepts	97
6.3 Mass spectrum reconstruction of $B^0 \rightarrow Dh$	106
7 Footprint and prospects	117
Bibliography	118

Abstract (English version)

The document brings together two related research objectives in the context of applications in High Energy Physics. The first topic concerns the Vertex Locator (VELO), a vertex detector at the LHCb experiment at CERN. Since 2018, the LHCb experiment has been in the phase of a major upgrade - Upgrade I, where the technology of the selected detectors in the LHCb spectrometer is adjusted to the increased instantaneous luminosity. The spectrometer will have to handle about five times as many concurrent collisions as before and operate at an increased readout frequency from 1.1 MHz to 40 MHz. The upgraded VELO takes advantage of the VeloPix ASIC - a pixel readout chip, where each pixel features its own electronic readout channel. The ASICs of the highest occupancy will be only 5.1 mm off the axis of the LHC beams. The dissertation presents the calibration of VELO, as well as the decoders, monitoring, and data simulation incorporated into the software platform Vetra. It further proposes and discusses a few applications of the software for certain studies of pixel detectors in general.

The second part is a search for novel tools based on Machine Learning (ML) in particle decays and mass spectrum reconstruction. Smart algorithms have had many successes in recent years, such as image generation or intricate decision-making. Although the number of emerging techniques is remarkable, their application in other than native fields often lags behind the new inventions. An overview of modern methods and their applications is outlined in the thesis. In particular, the use of Deep Neural and Generative Adversarial Networks (DNN, GAN) in the problem of generating physics data samples is demonstrated. The potential applications of deep models for mass spectrum fitting is shown, as well as heuristic optimization methods used in the process. Such methods can show higher efficiency in estimation than conventional, like maximum likelihood. Since imperfect measurements made by complex detectors affect the reconstruction process, the ML-based approach could bring new solutions to the problem and complement the statistical methods used in modelling so far.

Abstract (Polish version)

Praca łączy dwa tematy badawcze powiązane z sobą w kontekście zastosowań w Fizyce Wysokich Energii. Pierwszy temat dotyczy detektora wierzchołka VELO (Vertex Locator) w eksperymencie LHCb w CERN. Od 2018 roku eksperyment LHCb przechodzi modernizację, podczas której spektrometr LHCb dostosowywany jest do zwiększonej świetlności LHC. Spektrometr ma obsługiwać około 5-krotnie więcej zderzeń cząstek w tym samym czasie oraz pracować ze zwiększoną częstotliwością odczytu z 1.1 MHz do 40 MHz. Zmodernizowany detektor wierzchołka wykorzysta w detekcji cząstek naładowanych pikselowy czip VeloPix ASIC, w którym każdy piksel posiada własny, elektroniczny układ odczytu. Matryce o najwyższej okupancji znajdują się jedynie 5.1 mm od osi wiązek LHC. W pracy przedstawiona jest kalibracja detektora wierzchołka, a także monitoring i symulacja danych, które zostały wbudowane w platformę Vetra. Przedstawione są algorytmy skanowania matryc VeloPix, a także kilka ogólnych zastosowań platformy do prac nad detektorami pikselowymi.

W drugiej części praca poszukuje nowatorskich narzędzi opartych o techniki uczenia maszynowego (Machine Learning) w rekonstrukcji rozpadów cząstek i widma masy. Algorytmy sztucznej inteligencji odniosły w ostatnich latach wiele sukcesów, takich jak generowanie obrazów lub podejmowanie złożonych decyzji. Zastosowanie takich technik w innych dziedzinach nauki cechuje się jednak pewną bezwładnością. Problem ten przedstawiony jest w niniejszej rozprawie, gdzie dokonany jest przegląd najnowszych metod uczenia maszynowego oraz ich potencjalne zastosowanie. Praca pokazuje użycie głębokich sieci neuronowych (Deep Neural Network) i generatywnych sieci wrogich (Generative Adversarial Network) do problemu generowania danych fizycznych. Pokazane jest potencjalne użycie głębokich modeli do problemu dopasowania widma masy, a także wykorzystanie heurystycznych metod optymalizacji. Algorytmy oparte o metody inteligentne potrafią wykazać większą sprawność w estymacji niż metody konwencjonalne, np. największej wiarygodności. Ponieważ w rekonstrukcji rozpadów dużą rolę odgrywa niedoskonałość pomiarowa detektorów, podejście oparte o techniki uczenia maszynowego mogłoby wspomóc metody statystyczne obecnie używane przy modelowaniu.

Acknowledgments

Many thanks to the Supervisor of this dissertation Tomasz Szumlak for the overall mentoring of the work.

Exchanging views and ideas in the Krakow group of LHCb - Tomasz, Wojtek, Maciej, Agnieszka and Bartek - was helpful in the research presented in this dissertation.

Much obliged to the Vertex Locator detector group at CERN, in particular to Kazu, Victor and Karol.

The work was supported by the National Science Centre grant no. 2020/37/N/ST2/04008.

Contribution

The Author contributed to the testing of the vertex detector readout chip, the VeloPix ASIC (2015-2018), assisted at the SPS testbeams (2015-2018), and **designed the calibration system and software platform for Vertex Locator data analysis** (2019-2022). The Author further developed methods for a few pixel detector studies discussed in this work. The thesis covers only part of the Author's contribution to the Vertex Locator (which began in 2015), and essentially focuses on Upgrade I. The Author and the Supervisor Tomasz Szumlak invented a few Machine Learning concepts proposed in this document.

The Author was a principal investigator of the project funded by the National Science Centre (NCN) in Poland:

- *Automatic procedure to determine the optimal working point of the upgraded LHCb pixel vertex detector and development of a novel technique for reconstructing beauty meson mass spectra*, 2020/37/N/ST2/04008, **PRELUDIUM 19**, the amount awarded: 69 840 PLN, start date: 2021-01-19, completion date: 2022-12-18.

Talks and posters presented by the Author at the international conferences:

- P. Kopciewicz, M. Kacprzak, *A modern machine learning approach to statistical estimation for particle mass modeling*, XIV International Conference on Beauty, Charm and Hyperon Hadrons, Krakow, Poland, 5-11 June 2022, poster
- P. Kopciewicz, V. Morksyi, *Exploration of the evolution-based footprint on the Generative Adversarial Networks*, SSCI21: IEEE Symposium Series on Computational Intelligence, Orlando, Florida, United States, 5-7 December 2021, talk
- P. Kopciewicz, *The LHCb Vertex Locator Upgrade*, iWoRiD: 22nd International Workshop on Radiation Imaging Detectors, Ghent, Belgium, 27 July - 1 June 2021, talk

- P. Kopciewicz, *The upgrade of LHCb VELO*, IPRD19: 15th Topical Seminar on Innovative Particle and Radiation Detectors, Siena, Italy, 14-17 October 2019, talk
- P. Kopciewicz, *The upgrade II of LHCb VELO*, IPRD19: 15th Topical Seminar on Innovative Particle and Radiation Detectors, Siena, Italy, 14-17 October 2019, poster
- P. Kopciewicz, *Investigations on the radiation damage of the LHCb VELO: a full review*, VCI19: 15th Vienna Conference on Instrumentation, Vienna, Austria, 18-22 February 2019, poster
- P. Kopciewicz, *Software platform for the monitoring and calibration of the upgraded LHCb VELO*, XXV Cracow Epiphany Conference on Advances in Heavy Ion Physics, Krakow, Poland, 8 - 11 January 2018, talk

Apart from the LHCb collaboration publications, to which the Author contributed only indirectly, and the Vertex Locator detector group publications, with some direct contributions, the Author published the following publications/proceedings as the **first or corresponding author**:

- [1] P. Kopciewicz, K. Akiba, T. Szumlak et al., *Simulation and Optimization Studies of the LHCb Beetle Readout ASIC and Machine Learning Approach for Pulse Shape Reconstruction*, Sensors, Vol. 21, 6075, 2021
- [2] P. Kopciewicz, S. Maccolini, T. Szumlak, *The LHCb vertex locator upgrade - the detector calibration overview*, Journal of Instrumentation, Vol. 17, C01046, 2022
- [3] P. Kopciewicz, T. Szumlak, M. Majewski et al., *The upgrade I of LHCb VELO - towards an intelligent monitoring platform*, Journal of Instrumentation, Vol. 15, C06009, 2020
- [4] P. Kopciewicz, T. Szumlak, W. Krupa et al., *Software Platform For The Monitoring And Calibration Of The LHCb Upgrade I Silicon Detectors*, Acta Physica Polonica B, Vol. 50, 2019
- [5] P. Kopciewicz, V. Morskyi, *Exploration of the evolution-based footprint on the Generative Adversarial Networks*, IEEE Symposium Series on Computational Intelligence, 2022

- [6] P. Kopciewicz, S. Lukasik, *Exploiting flower constancy in flower pollination algorithm: improved biotic flower pollination algorithm and its experimental evaluation*, Neural Computing and Applications, Vol. 32, pp. 11999-12010, 2020
- [7] (submitted) P. Kopciewicz, M. Kacprzak, T. Szumlak, *A Modern Machine Learning Approach for B Meson Decay Generative Modeling*, Acta Physica Polonica B, 2022

The Author was a **peer reviewer in Journal of Instrumentation** (JINST).

Acronyms

ADAM - Adaptive Moment Estimation	FPA - Flower Pollination Algorithm
ADC - Analog to Digital Converter	FPGA - Field Programmable Gate Array
AI - Artificial Intelligence	GA - Genetic Algorithm
AIDA - Abstract Interfaces for Data Analysis	GAN - Generative Adversarial Network
ANN - Artificial Neural Network	GBT - Gigabit Bidirectional Transmitter
ASIC - Application-Specific Integrated Circuit	GD - Gradient Descent
BCE - Binary Cross-Entropy	GL - Gauss-Landau
BDT - Boosted Decision Tree	GPS - Global Positioning System
BE - Big Endian	GPU - Graphics Processing Unit
BEGAN - Boundary Equilibrium GAN	GWT - Gigabyte Wireline Transmitter
BFPA - Biotic Flower Pollination Algorithm	FRG - Fragment (Data Type)
BGAN - Boundary-Seeking GAN	HCAL - Hadronic Calorimeter
BX ID - Bunch Crossing Identification	HEP - High-Energy Physics
CNN - Convolutional Neural Network	HL-LHC - High-Luminosity LHC
CP - Charge Parity	HLT - High-Level Trigger
CPU - Central Processing Unit	ICF - Isolated Cluster Flag
CSV - Comma-Separated Values	Idisc - Discriminator Current
CUDA - Compute Unified Device Architecture	Ikrum - Krummenacher Current
DAC - Digital to Analog Converter	Ipre - Preamplifier Current
DCGAN - Deep Convolutional GAN	kNN - k-Nearest Neighbors
DNN - Deep Neural Network	L0 Trigger - Level-0 Trigger
DAQ - Data Acquisition System	LE - Little Endian
DE - Differential Evolution	LF - Low Frequency
DIM - Distributed Information Management	LFSR - Linear Feedback Shift Register
DNS - Domain Name System	LHC - Large Hadron Collider
DT - Decision Tree	LHCb - Large Hadron Collider beauty
DUT - Device Under Test	LS1 - Long Shutdown 1
ECAL - Electromagnetic Calorimeter	MC - Monte Carlo
ECS - Experimental Control System	MDF - Mast Data File
FIFO - First In, First Out	ML - Machine Learning

MLE - Maximum Likelihood Estimation	SPIDR - Speedy Pixel Detector Readout
MPV - Most Probable Value	SPS - Super Proton Synchrotron
MSE - Mean-Square Error	SQL - Structured Query Language
OPB - Optical-and-Power Board	SVM - Support Vector Machine
PID - Particle Identification	TCK - Trigger Configuration Key
R&D - Research and Development	TES - Transient Event Store
RESTful - Representational State Transfer	TFC - Timing and Fast Control
RF - Radio-Frequency	TMVA - Toolkit for Multivariate Data Analysis
RICH - Ring Imaging Cherenkov	ToA - Time-of-Arrival
RNN - Recurrent Neural Network	ToT - Time-over-Threshold
SciFi - Scintillating Fibre	TP - Test Pulse
SGD - Stochastic Gradient Descent	TT - Tracker Turicensis
SHADE - Success History-based Adaptive DE	UT - Upstream Tracker
SN - Signal-to-Noise	WGAN - Wasserstein GAN
SODIN - Supervisor ODIN	VELO - Vertex Locator
SP - Super-Pixel	

Chapter 1

Introduction

Investigating the CP (Charge-Parity) violation is one of the leading research directions in modern High Energy Physics (HEP). Due to the difficulty of observing such phenomena directly, the only chance for verifying current predictions is to recreate the conditions occurring in the Young Universe using particle accelerators providing high-energy parton collisions and frontier detection solutions. CERN (European Center for Nuclear Research) hosts a program dedicated entirely to the research of the processes of CP violation under the name LHCb (Large Hadron Collider beauty). The experiment started in 2010, and during its operation at LHC, it collected physics data over periods known as Runs 1 and 2 until December 2018. Since then, the experiment has been in a major upgrade phase (dubbed Upgrade I), in which the technology of the selected detectors in the LHCb spectrometer is changed to adjust them to the increased luminosity, pile-up and to the improved readout. The dissertation is mainly devoted to the Vertex Locator (the vertex detector in the LHCb spectrometer) and focuses on the software for Upgrade I, presenting the calibration and monitoring platform *Vetra* that was designed and implemented by the Author of this thesis. It further discusses a few selected comprehend studies on pixel detectors that take advantage of the platform, which are clustering, Time-over-Threshold studies, and chip analog optimization. As the complementary research objective, the thesis first presents a wide range of Computational Intelligence and Machine Learning (ML) methods and gives ideas and concepts for their potential applications in physics analysis and mass spectrum reconstruction, on the example of $B^0 \rightarrow D_s^- \pi^+$ decay. A very detailed survey of the latest ML applications in HEP precedes the proposed application concepts.

The thesis is organized as follows. Chapter 2 makes a survey of selected Machine Learning methods, including Generative Adversarial Networks, and outlines their state-of-the-art. Chapter 3 is a comprehensive description of the LHCb experiment, Upgrade

I in general, and the Vertex Locator. Chapter 4 presents the software for the upgraded Vertex Locator; it begins from a technical description of the LHCb software environment and further proceeds to the software that was created by the Author, accountable for data processing, decoding, pixel equalization, calibration and detector monitoring. Chapter 5 presents a few studies on the pixel detectors; clustering, Time-over-Threshold, and analog optimization. Chapter 6 surveys the recent applications of Machine Learning in HEP and LHCb, explores potential fields that could take advantage of such methods, and finally proposes a few concepts based on the latest from Machine Learning, providing preliminary studies on the example of B^0 meson decay. Chapter 7 is a brief summary where the footprint of the dissertation is discussed.

Chapter 2

Computational Intelligence

Computational Intelligence, a sub-field of Artificial Intelligence (AI), is one of the most popular and burgeoning fields of Computer Science, as it gains far-reaching practical applications in real life [8]. ML methods find a diverse adoption in a high spectrum of science, as they consistently prove to be more efficient than standard mathematical approaches, such as those based on statistical inference used for modelling so far. The recent growth of the methods is so extensive that the whole field was arbitrarily divided into many research areas, and the number of research directions grows every year. Though the majority of the studies focus on putting small improvements to the existing models, new methods are discovered as the whole field is getting more attention. This Chapter describes the most significant branches of ML, which are classification, intelligent optimization, and neural networks, and gives a generic overview of their state-of-the-art at the moment the thesis is being written (Jun-Sep 2022). The methods are selected with a scope of their usage in HEP, particularly the ones that are referred to later in Chapter 6, where the applications of ML in the LHCb experiment are discussed in detail. This Chapter is divided as follows; Section 2.1 gives a hands-on experience regarding classification, Section 2.3 outlines optimization algorithms including metaheuristics, Section 2.2 explains neural and deep models, Section 2.4 continues so with Generative Adversarial Networks as an example of modern generative models, and finally, Section 2.5 makes a survey on deep model learning via optimization algorithms.

2.1 Classification

Classification is the process of recognition and categorization, e.g. an object into a class. This common data analysis problem can be solved using various approaches, where those

performing best are based on ML methods [9]. A method, arguably, lying between ML and a purely probabilistic approach is Naive Bayes [10], where the objects are classified using Bayesian cognition [11, 12]. Classification belongs to a group of Supervised Learning (classes should be given a priori [13]). The most common classifiers are Decision Tree (DT) [14], k-Nearest Neighbors algorithm (kNN) (1951, Fix et al. [15]), Fisher's Linear Discriminant (1936, Fisher [16]) and Linear Discriminant Analysis (LDA) [17] with further improvements [18, 19], Support-Vector Machine (SVM) (1995, Cortes et al. [20]) and a range of hybrids [21, 22]. A neural network can be used as classifier as well (see Section 2.2).

DT is a method of classification and regression [23] widely used in Data Science. The logic behind the approach is to support the decision-making process by considering likelihoods, risks and potential rewards. DTs are built of nodes and branches (rules and options), which follow the logical reasoning explicitly [24]. Each node applies a rule according to which the local decision is made. Rules do not necessarily have to be related to the object, which plays an advantage over the other classification methods. Conditions employed in decision-making can be binomial or polynomial but always discrete, which differs DT from Fuzzy Logic [25]. The first decision is made in the root node, and the process continues climbing downwards to the bottom of the tree. The training includes choosing the right rule at the root and the hierarchy of rules for the burst nodes. A tree can grow and add nodes until a growth-stop condition is satisfied, e.g. a limit for the total depth of the tree or a certain level of prediction accuracy, which is measured by uncertainty or entropy indicators [26, 27]. To measure the classification impurity, a Gini index [28] (or Gini impurity) is used commonly, which indicates the probability of a feature classified incorrectly when selected randomly. Trees can process the problem in a brute-force way when trained deep enough. They can support any decision-making process, e.g. a minimax strategy [29], provided the consecutive layers of branches are the moves of opposite agents. The numerical complexity of the tree is a function of the depth and the number of nodes in each branch. Regardless of whether DTs are used for classification or regression, they train fast compared to models based on neural nets and are usually chosen when the training dataset is too small to build a competent deep model. They are vulnerable to overfitting [30] (especially when appended nodes face different function complexity) and generally feature high variance - low bias attributes¹ [31–33]. To decrease the tree complexity, a branch-pruning method can be applied, such that it gets rid of the branches that are less significant

¹A variance-bias trade-off is a common classification dilemma. High bias oversimplifies the classification but is more predictable for data the model has not seen before (low variance), and vice versa.

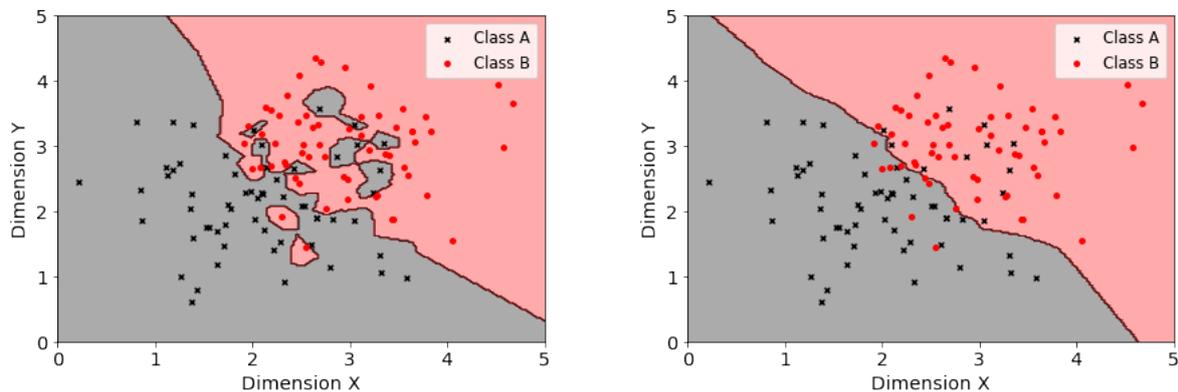


Figure 2.1: kNN classifier for $k = 1$ (left) and $k = 100$ (right). With higher values of k the model tends to the higher bias.

in the decision-making process [34]. Many pruning methods are available, e.g. reduced-error pruning [35], cost-complexity pruning [36] or alpha-beta pruning [37], which are used depending on the problem [38, 39]. Common metrics to improve DT performance are bootstrapping [40] - a resampling method with replacement (which means an element can appear multiple times in a sample) used for hypothesis testing on the pulled samples from approximating distribution [41, 42] (which can be used during learning), bagging - a bootstrap aggregation [43] to create weak learners via bootstrapping [44], and other ensemble methods [45]. More advanced models use a sequential combination of many relatively weak and inaccurate predictors (usually decision stumps), which is dubbed boosting or gradient-boosting if the training relies on gradient minimization [46–48]. Most popular gradient-boosted models [49] are XGBoost [50], CatBoost [51] for categorization, and AdaBoost with adaptive boosting [52, 53]. DTs show a strong presence in HEP field [54], which is further described in Chapter 6.

The other well-performing classification algorithm is k-Nearest Neighbors (kNN) [55]. When classifying an object, the algorithm searches for the k closest neighbors around the data point [56] and postulates that it belongs to the same class as the majority of the neighbors, or to which the highest number of neighbors were classified if the task is a non-binary classification [57]. Although the idea might look quite simple for it to be used in large datasets, it performs very well even in multivariate problems. The value of k may vary and can be adjusted by a grid search or other optimizing algorithm [58] for the highest possible efficiency. There are several kinds of distance definitions one can use in kNN, for instance, binomial scaled, Minkowski [59], or ordinary Euclidean. kNN is relatively

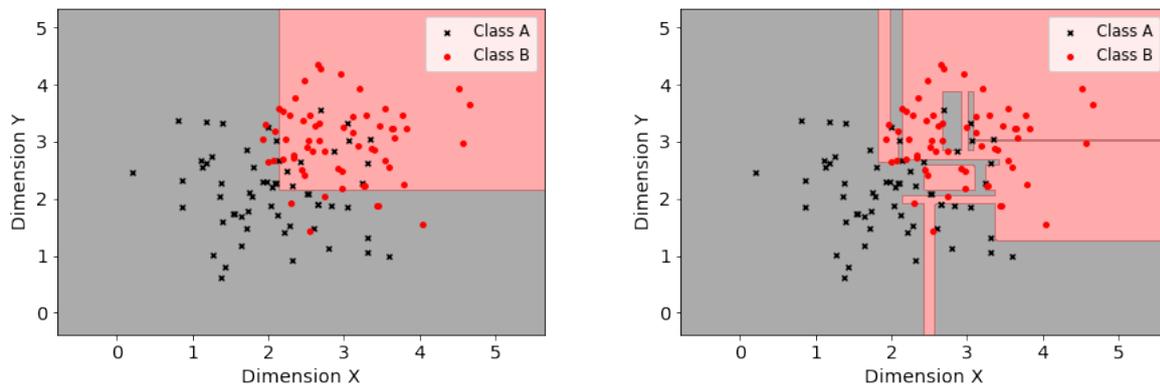


Figure 2.2: A Decision Tree of depth $d = 2$ (left) and $d = 10$ (right).

simple to use and has found multiple applications due to its good performance [60]. A kNN classifier is illustrated in Fig. 2.1 for the random dataset, where the difference in k parameter is highlighted. A low value of k ($k = 1$) leads to low bias and high variance of the model, while its high value ($k = 100$) to high bias and low variance [61]. For comparison, a DT classifier for the same dataset is shown in Fig. 2.2, with a small depth of the tree ($d = 2$) and a big depth ($d = 10$). Note that no objects from the training dataset are misclassified when DT is overtrained. The Python code in the form of Jupyter notebook (web application for sharing computational Python documents [62]) to create the plots mentioned above can be found in github repository of the Author [63] (sklearn library was used to model the DT [64]).

2.2 Neural and deep models

Neural networks [65] are computer systems for data processing inspired by the nature of a neural system and are used for classification, regression, approximation and prediction. The processing mechanism originates from a nerve cell neuron [66], a cell that functions as a binary classifier [67]. Its computational relative, perceptron [68], activates when the real-valued input passes a certain threshold of the activation function and if so, yields the real value of this function. The shape of the activation function is not predetermined and can vary depending on which one is more relevant to the considered problem [69]. The Heaviside step function is the simplest possible but allows the neuron to only distinguish between two linearly separated classes. The most common activation functions are ReLu (Rectified Linear Unit) and LeakyReLu (Leaky Rectified Linear Unit) [70], or others of this

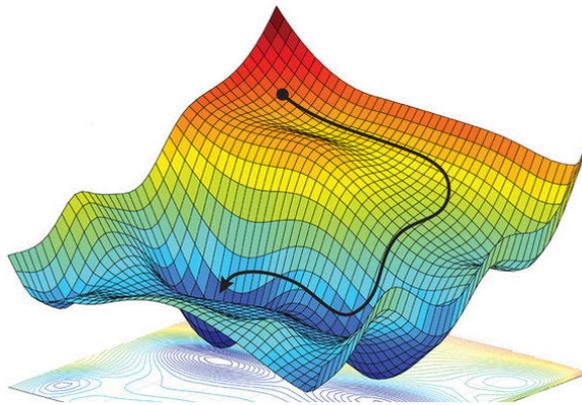


Figure 2.3: A gradient descent visualized in a loss function landscape. [77]

type [71]. Those allow the networks for more complex class separation, although a single non-linearity allows only one flex in the class separating line. Another commonly used is sigmoid (S-curve), which is non-linear throughout and allows more complex separation between two classes [72]. A disadvantage of this function is its vanishing derivative when approaching infinity and, thus, the vanishing difference between higher input real values. This can be detrimental for the class separation when the perceptron training relies on gradient calculations and generally requires scaling of the input data.

The standard training of the perceptron (or the whole network of neurons) evaluates the model on the loss function [73] and computes the gradient of this evaluation as a function of the parameters contributing to its value inside the model in a procedure called backpropagation (which computes the error propagation). The gradient naturally points to a direction of the highest ascent of the loss function, therefore to decrease the loss, a fraction of the gradient is subtracted from each parameter in the model. The fraction is defined by a learning rate [74]. The learning rate is adjustable and can change during the learning process [75, 76]. The procedure results in a Gradient Descent (GD) of the loss, where the model is fitted better to the training dataset in each gradient subtraction step. Fig. 2.3 illustrates the GD of a simple loss function landscape of two parameters. There is a variety of possible loss functions that can be used [78–80], but the most popular are Mean-Square Error (MSE) for approximation and Binary Cross-Entropy (BCE) [81] for binary classification. The loss function can reflect the policy when the neural network is trained via reinforcement² [82]. When the loss function has regions of local flatness, its gradient can vanish [83, 84], which in turn can prevent the network from learning properly.

²Reinforcement learning can be applied when no training data is available, but the model can interact with the environment. The model trained via reinforcement learns to accomplish a task through repeated trial. The interrelation between actions and received rewards is then defined by a policy.

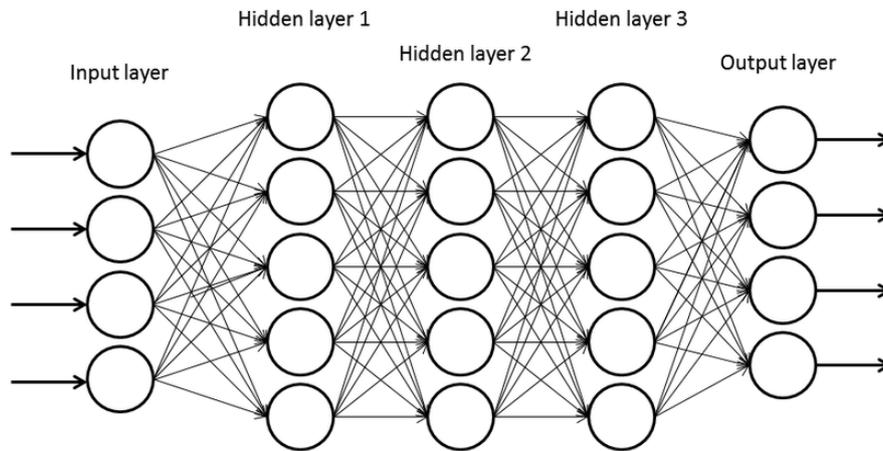


Figure 2.4: A neural network with 3 hidden layers. [89]

The opposite effect is called exploding gradient [85, 86] and occurs when the learning rate is too large or when the loss function is sharply corrugated. There are plenty of solutions how to investigate and prevent negative gradient effects, such as [87, 88] to mention but a few.

An exemplary scheme of a neural network with a few inner layers (hidden layers) is illustrated in Fig. 2.4. The term deep is used for a neural network with at least one hidden layer inside. A mathematical explanation of the latent space (a vector space of hidden layer parameters that are not directly interpretable) properties during the training and their impact on the neural network efficiency are not forthright and became the entire field research [90].

The network architecture [91, 92], the type of connection between neurons [93], the training method [94–97] and gradient propagation [98, 99] can be adjusted depending on the application. Among the popular and modern models, the most noteworthy are Convolutional Neural Networks (CNN) [100], commonly used for computer vision, and Recurrent Neural Networks (RNN) [101], for text generation and speech recognition. The latter got recently vested with gated architectures, e.g. Hochreiter et al. LSTM (Long Short Term Memory) [102] and Cho et al. GRU (Gated Recurrent Unit) [103], with many successes in text generation. The most popular algorithms to train the network are ordinary Gradient Descent (GD) [104], Stochastic Gradient Descent (SGD) [105] when training on batches³ and ADAM, improved SGD with gradient weighting [106].

³When the dataset is large, evaluation of the loss function gets CPU-costly. The loss function can be instead evaluated on random batches from the dataset, hence stochastic.

2.3 Intelligent optimization

Optimization is a mathematical problem of finding a suite of parameters that minimize a given cost function or fitness function [107]. Optimization tasks of higher dimensionality usually cannot be solved by GD-based methods, as they are often highly multimodal and the algorithm gets stuck in a local minimum immediately [108]. Such problems can hardly be solved using brute force or greedy strategy, as it would require high, often unacceptable computational cost [109]. Intelligent optimization deals with such objectives using methods that are much less costly than a greedy approach.

The most popular are Genetic Algorithms (GA) [110], which refer to the natural selection of species. A whole concept relies on this analogy and consists of selection, cross-over and mutation inspired by nature. Selection resolves premature convergence in the population [111], cross-over tends to approach the objective function [112], while mutation is used to prevent the algorithm from being trapped in local minima [113], although might possibly be more impactful at times [114]. Various cross-over [115, 116], selection [117] and mutation approaches [118–120] can be found in literature. Genetic Algorithms are quite old compared to other ML methods (they were developed in 1960s and 1970s by Holland [121]), and over the years of their existence, many algorithm variants were explored [122]. In particular, Genetic Algorithms formed a ground for nature-inspired metaheuristics⁴ [124]. Those were intensely studied in recent years [125] proving good performance [126, 127], although sometimes criticized for using metaphors instead of mathematical explanations (especially Sorensen [128] and [129–131]). Among the popular metaheuristics are, for instance, the swarm optimization [132], the firefly algorithm [133], the flower pollination algorithm [134] (FPA), the simulated annealing [135], or the ant colony optimization [136]. The methods were refined multiple times, e.g. Lukasik and myself proposed the idea of cutting the local randomizing factors in favor of organized population movement in the flower pollination algorithm [6].

Differential Evolutionary (DE) algorithms (1970, Storn et al. [137]) were founded on GA basis [138] and have been actively developed in many directions [139–142]. They demonstrate excellent performance at a variety of problems of multi-objective optimization [143, 144]. Various mutation concepts were applied to DE models [145], making the mutation role more impactful on the exploration process [146, 147]. One of the best variant of DE, which won a few IEEE CEC competitions (the most popular optimization algorithms competition) [148] is SHADE (2013, Tanabe et al. [149]), a self-adaptive DE, further

⁴Metaheuristic is a paradigm aimed to find guidelines for a solution to the optimization problem [123].

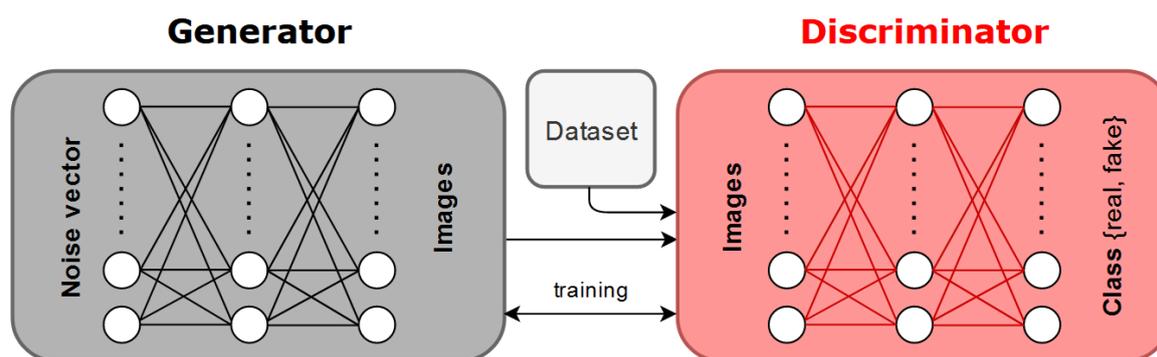


Figure 2.5: The generic scheme of GAN.

improved to LSHADE [150], available in github [151].

2.4 Generative Adversarial Networks

Generative Adversarial Networks - GANs (2014, Goodfellow et al. [152]) are lately considered one of the most compelling generative models. They have found multiple applications, from computer vision [153], image processing [154], image background extraction [155], semantic segmentation [156], feature generation, natural language processing (NLP) [157], image-to-image translation [158], and multiple others [159].

GAN compounds of two models, one generative and one discriminative. These are usually deep neural networks of corresponding architecture. The models are trained against each other, such that the generator tries to fool the discriminator to make it unable to tell whether the data come from the generator or from the existing dataset. At the same time, the discriminator trains as a classifier to predict the origin of each data sample. A generic scheme of GAN is sketched in Fig. 2.5. Expressed mathematically, the generator G and the discriminator D follow a minimax game to find the optimal path to minimize their adversarial loss. In the original model [152], the loss function for G was formulated as (2.1):

$$L_{gen} = \min[\log(D(x)) + \log(1 - D(G(n)))], \quad (2.1)$$

where x stands for the real data, and $G(n)$ for the generated data on the noise vector n . This formula uses fixed discriminated prediction to express the G loss. D follows the minimax strategy when calculating the gradient, which was expressed by Goodfellow as



Figure 2.6: Fake samples generated by BigGAN (Brock et al. [178]).

(2.2):

$$\min_G \max_D V(D, G) = E(X_{p_{data}(x)}[\log(D(x))]) + E(Y_{p_y(y)}[\log(1 - D(G(y)))]), \quad (2.2)$$

where $D(X)$ is the estimate of the probability that a true data sample would be found to be true, $D(G(y))$ is the estimate of the probability that a false sample would be deemed false, EX and EY are expected values over the whole dataset, and $V(D, G)$ is a combined loss function in which D and G strive to find the equilibrium point.

Since 2014, many improvements have been proposed to the vanilla model, at times completely revising its architecture or capabilities. In 2017 Arjovsky et al. [160] proposed an innovative way to compute the adversarial loss, commonly known as Wasserstein loss [161], based on Wasserstein metric⁵ [162], introducing WGAN, which turned out to be more efficient in a number of cases. Since a recast of the loss function can revise the latent space to behave differently, many alternate models introduce gradient manipulation, e.g. gradient penalty [163]. Some research ideas resulted in models changing the GAN paradigm, such as Zhu et al. CycleGAN for image-to-image translation [164], Radford et al. DCGAN (Deep Convolutional GAN) combining CNN architecture with GAN [165], or Hjelm et al. BGAN (Boundary-seeking GAN) [166], which computes generator weights via policy gradient [167]. In 2017 Berthelot et al. proposed BEGAN (Boundary Equilibrium GAN) [168] with loss equilibrium enforcement paired with the Wasserstein distance. Isola et al. (2016) [169] released famous pix2pix software for image-to-image translation (online version demo in [170]) using Conditional GAN (2014, Mirza and Osindero [171]). A summary of the recent advances and ideas of GAN models can be found in [172, 173].

Many improvements rely on concepts that were discovered in other ML fields, such

⁵In short, the Wasserstein distance is the cost of turning one distribution into another (number of moves times Euclidean distance) in the Optimal Transport Problem. When applied to GAN, it measures the distance between the distributions of real and generated datasets.

as self-attention [174] (arguably one of the best discoveries of 2017 in ML), resulting in self-attention GAN by Zhang and Goodfellow et al. [175] (Goodfellow was also the author of the original model). Because the latent space attributes lack unambiguous mathematical reasoning, it finds a reflection on the research directions [176]. The same relates to the image synthesis process [177]. Fig. 2.6 shows the example images generated by a customized GAN. As a matter of fact, most of the recent advances in the field were achieved through customized models, e.g. generation of the human faces from textual description, Sabae et al. StyleGAN2 in this specific case [179].

2.5 Learning by way of optimization

Optionally one could look at the deep model training as an optimization problem. The fitness function is nearly impossible to define, but if existed, it might have an indefinite number of local minima, trapping and restricting GD from moving in any direction [180]. In fact, Morse and Stanley [181] suggested that GD in deep neural networks can significantly suffer from such restrictions. On the other hand, adding an escape mechanism for the gradient to GD is usually costly in terms of CPU/GPU usage and misuses the learning process [182].

Acquiring heuristic training to the neural networks was investigated quite early [183] as an alternative to backpropagation [184, 185]. Since that time, many hybrid concepts have been proposed [186] and used for real-life applications, e.g. prediction in Time Series [187, 188]. The conclusion was that even though the heuristic training is unlikely to overcome backpropagation [189], the hybrids can be surprisingly effective [190, 191], or train faster than policy gradient methods in reinforcement learning [192, 193]. Applying any GA or DE methods to the deep models is known as neuroevolution [194, 195]. Models of this kind were successful at designing, or more in context, evolving the model architecture [196–198], classifiers [199, 200] and CNNs [201, 202]. Martin et al. [203] devised EvoDeep with lateral hyper-parameters to train the deep model, while Sun et al. [204] optimized the weights using Neural Architecture Search.

In 2019, Wang et al. proposed Evolutionary GAN [205], where the architecture of the generator evolves to counteract the discriminator habits, with further major improvements to the model [206, 207]. The GAN model oriented to multi-objective optimization (optimization of more than one objective function [208]) brought promising results [209]. He et al. [210] reversed that approach and used GAN in multi-objective optimization. Morskyi

and myself proposed GAN with an evolutionary footprint with GPU-efficient mutations [5], which help the model to learn more smoothly and achieve the results quicker in terms of optimization complexity, although not quicker in terms of CPU operations. There are plenty of similar enhancements one could find in the generic survey [211], as the field has been tremendously growing over the past few years.

Chapter 3

The LHCb experiment

The LHCb (Large Hadron Collider beauty) is one of the leading experiments operating at LHC, a ring-shaped accelerator situated near Geneva. The main goal of the experiment is to investigate a combined CP symmetry violation in a sector of beauty and charm meson and baryon decays [212] and to study rare decays of those towards discovering New Physics phenomena. For the eight years of the experiment (2010-2018), LHC Run 1 and 2 [213], the LHCb collaboration [214] led to a number of discoveries, e.g. the observation of a complex pentaquark state [215] or the confirmation of theoretically predicted CP symmetry breaking in the selected decays of charm mesons [216]. In December 2018 Run 2 finished, and the LHCb experiment moved into a long-time modernization phase (Long Shutdown 1, LS1) [217] before Run 3, planned firstly to begin in 2020, later postponed to 2022 owing to the Covid-19 pandemic. This Upgrade I adjusts various components of the LHCb spectrometer to the increased instantaneous luminosity, with a physical goal of new possibilities to discover signs of New Physics (Beyond the Standard Model Physics [218]), with plans going as far ahead as to the Upgrade II [219] planned for around 2030 for High-Luminosity LHC (HL-LHC) era [220, 221].

The LHC is designed to provide proton-proton collisions of the centre-of-mass energy of 14 TeV [222]. The accelerated particles move in opposite directions in two separate LHC rings. The beams are crossed in a few interaction points providing collisions to the complex detectors of LHC experiments. The LHC flagship experiments are ATLAS [223], ALICE [224], CMS [225] and LHCb. Their approximate location on the LHC ring is depicted in Fig. 3.1. While ALICE is almost entirely devoted to ion-ion collisions, the research programs of ATLAS and CMS are more comprehensive and used to search for Beyond the Standard Model Physics [227]. Among the smaller LHC experiments, LHCf [228], TOTEM [229], MoEDAL-MAPP [230], and FASER [231] can be mentioned. A summary

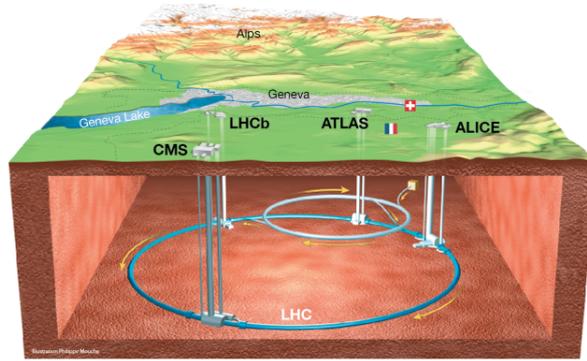


Figure 3.1: The experimental site facility with four foremost experiments marked. [226]

of the research goals of all experiments mentioned above can be found in [232].

3.1 The LHCb spectrometer

The spectrometer of LHCb has a one-arm geometry and constitutes of many complex detectors [233]. Its geometry is justified by the fact that the pairs of b quark and antiquark, while produced, always go to the same cone, as both are boosted in the same direction by the momentum of the interacting partons. The LHCb acceptance coverage in pseudorapidity enfolds the region of $1.8 < \eta < 4.9$ (pseudorapidity is defined as $\eta = -\ln[\tan(\theta/2)]$, and θ is the polar angle of the track with respect to the beam).

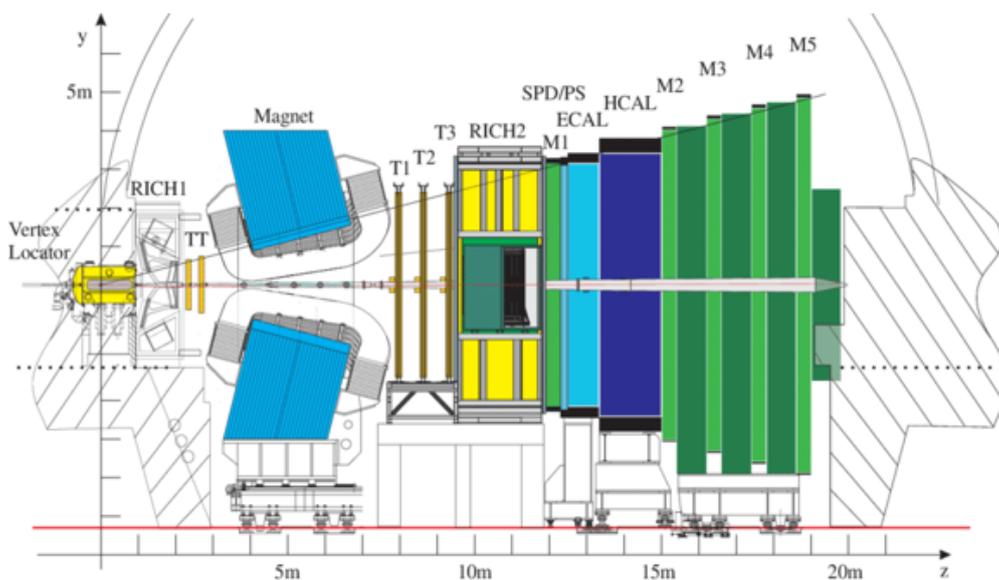


Figure 3.2: The LHCb spectrometer geometry. [233]

The LHCb detectors with their relative position are illustrated in Fig. 3.2. A single dipole magnet, referred to as LHCb Magnet [234] (shown in Fig. 3.3) is used to measure the momentum of the charged particles. A magnetic field of the device bends particle trajectories, allowing the transverse momentum to be estimated by measuring the sagitta of the reconstructed track. The momentum estimation is possible only for tracks that were registered in tracking detectors in front of and behind the magnet. The magnet does not interact with photons nor particles with no charge.



Figure 3.3: LHCb Magnet (left) [235] and the spectrometer profile view [236].

Until LS1, the spectrometer used to have three tracking detectors, which were Vertex Locator (VELO), Tracker Turicensis (TT), and three Tracking Stations (T1, T2, T3), which were further categorized as Inner Tracker [237] and Outer Tracker [238]. The Inner Tracker and Tracker Turicensis are often referred to as LHCb Silicon Tracker [239, 240]. Vertex Locator (Fig. 3.4) closely surrounds the beam interaction region and measures tracks of charged particles [242]. Even though the distance between the beams and the closest stations of VELO is very small (around 8.1 mm, which is decreased to 5.1 mm after LS1), the particles that do not decay until reaching the detector material are considered long-lived [243] (e.g. kaons or pions). Since there is no possibility of observing tracks of B mesons (or other short-lived particles), they have to be measured indirectly. A pattern of tracks in the vertex detector is used to reconstruct a vertex of the decay (the secondary vertex) and a vertex of the collision (the primary vertex). When the products of a decay are picked and measured completely, it is possible to reconstruct a full decay tree. The process of reconstruction is handled by the triggers and the offline reconstruction software. VELO is assembled of two halves (which are shown in Fig. 3.5) that are able to move apart for the survey and when the beam stabilizes, as the particles from the beam halo can damage the sensitive parts of the detector.

There are 21 stations equipped with 2 modules, each distributed along the beam (consult

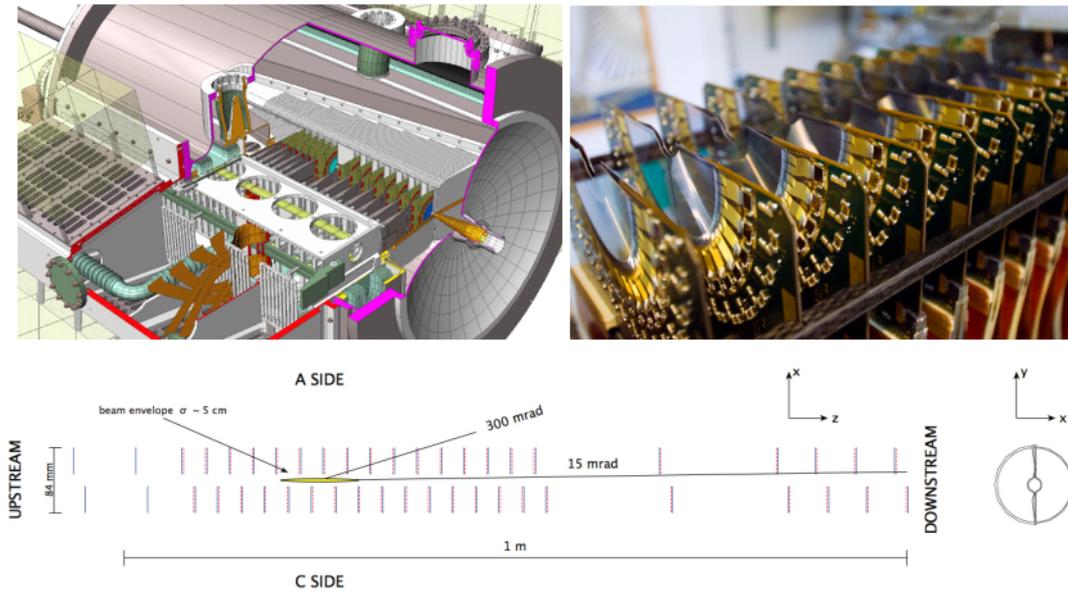


Figure 3.4: The scheme and geometry of the Vertex Locator. [241]

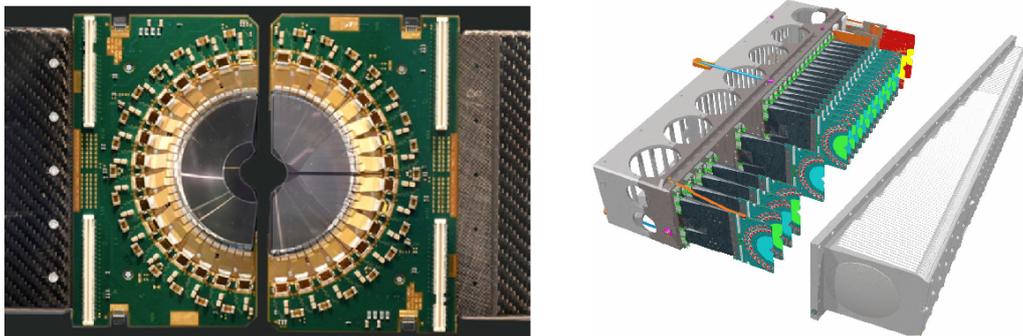


Figure 3.5: The R , Φ modules in closed position (left) and one side of the Vertex Locator (right). [244]

Fig. 3.4). Additional 2 pile-up stations in the backward direction to the spectrometer are installed to support the event triggering. Each module features 2 n-on-n type silicon microstrip sensors assembled on either side [245]. A sensor of R -type disposes the strips perpendicularly to the centre and measures the radial coordinate of the travelling charged particle in the polar system. Complementary, Φ sensor measures the polar angle by having the strips laid out radially off the centre. The minimum pitch of $37 \mu\text{m}$ is achieved in the innermost region of the sensors. The geometry of the modules is visualized in Fig. 3.4. A readout chip for VELO and Silicon Tracker is the Beetle ASIC [246, 247], which provides both analog and digital readout [248]. The analog readout is used in VELO, while the digital in Silicon Tracker only [249, 250].

RICH (Ring Imaging Cherenkov) [251] is a system of two Cherenkov detectors, RICH1 and RICH2, separated by LHCb Magnet, standing a part of charged hadronic particles identification system through measuring the Cherenkov radiation of charged particles crossing their gas radiator [252]. When a charged particle passes the entrance of the system (the radiators), and the velocity of that particle exceeds the speed of light in the medium, Cherenkov photons are emitted. The emission angle is a function of the particle velocity and the refractive index of the material, which is used for identification (see Fig. 3.6). RICH1 (in front of the magnet) covers the momentum range of 2 - 60 GeV/c, while RICH2 the range of 15 - 100 GeV/c. The LHCb Cherenkov detectors provide unique particle identification capabilities compared to other LHC detectors.

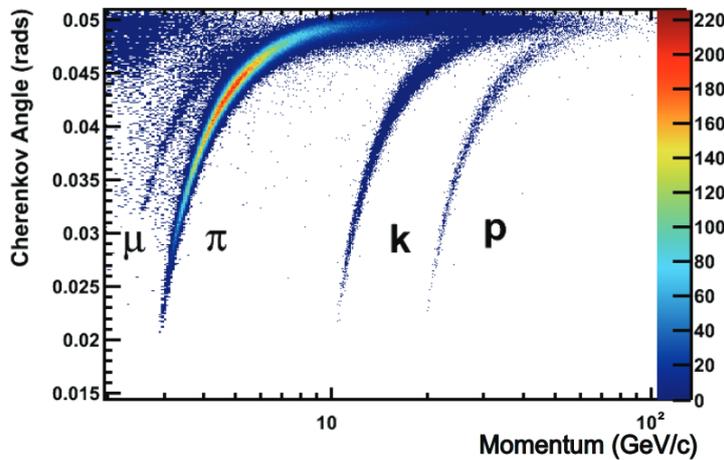


Figure 3.6: Reconstructed Cherenkov angle in RICH1. [253]

The LHCb calorimeter system [254] is composed of a scintillating pad detector (SPD), a preshower (PS), an electromagnetic calorimeter (ECAL), and a hadronic calorimeter (HCAL) [255]. The main purpose of the system is to identify hadrons, photons and electrons (SPD and PS are used to tag charged particles), and to measure their energy. HCAL provides the trigger on the presence of hadrons with high transverse momentum (a signature of B meson decays) and takes account of around 70% fast trigger decisions [256]. ECAL (shown in Fig. 3.7) precisely reconstructs the kinematical parameters of neutral particles [258].

The furthest downstream with respect to the interaction point lies a system of Muon Stations [259] (M1 - M5 in Fig. 3.2), which targets the muon detection when they travel through the gaseous proportional chambers [260]. The muon system plays an essential role in providing fast information to the triggers. It is capable of performing stand-alone tracking, assuming that all particles detected came from the interaction point, which in



Figure 3.7: The LHCb electromagnetic calorimeter ECAL. [257]

turn allows for a coarse estimation of their momentum.

The trigger system is essentially divided into two parts, the hardware part named L0 (Level-0) Trigger and the software HLT (High-Level Trigger) [261]. The system decides if the physics event is interesting or not and whether to save it or to discard. The L0 trigger makes a decision taking into account information from calorimeters, Muon System and two VELO pile-up stations (see [242]). The L0 trigger operates at 40 MHz rate, the nominal LHC frequency, and yields at the maximum rate of 1.1 MHz. The reduction is performed on the information provided mostly by calorimeters and Muon System. Once the event is accepted, the other detectors are read out at that specific bunch crossing number. The TELL1 readout processing boards [262] send the data to Event Filter Farm, which is a CPU farm. The architecture of this system is explained in Fig. 3.8, which differentiates Run 1 and 2 taking cognizance of the changes that were made between those periods due to the experience and computational memory improvements [264]. The first stage of the software trigger HLT1 aims to reduce the incoming data rate such that the second stage, HLT2, has enough time to fully reconstruct the selected events. Run 1 trigger featured a buffer which could store data from L0 to put the decision off in time to the moment when HLT is idle. This software architecture changed in 2015 [265, 266]. A buffer between L0 and HLT1 was removed as HLT1 improved efficiency via far more superior software, and HLT1 and HLT2 no longer work at the same time (HLT2 became asynchronous with respect to the beam crossing). Instead, the data brought off the HLT1 are saved to a disk and processed by HLT2 with all CPUs at the time when the detector takes no data. The trigger yields the reconstructed tracks with the identification of the particles assigned where possible. The process of further analysis holds a generic name Offline Analysis

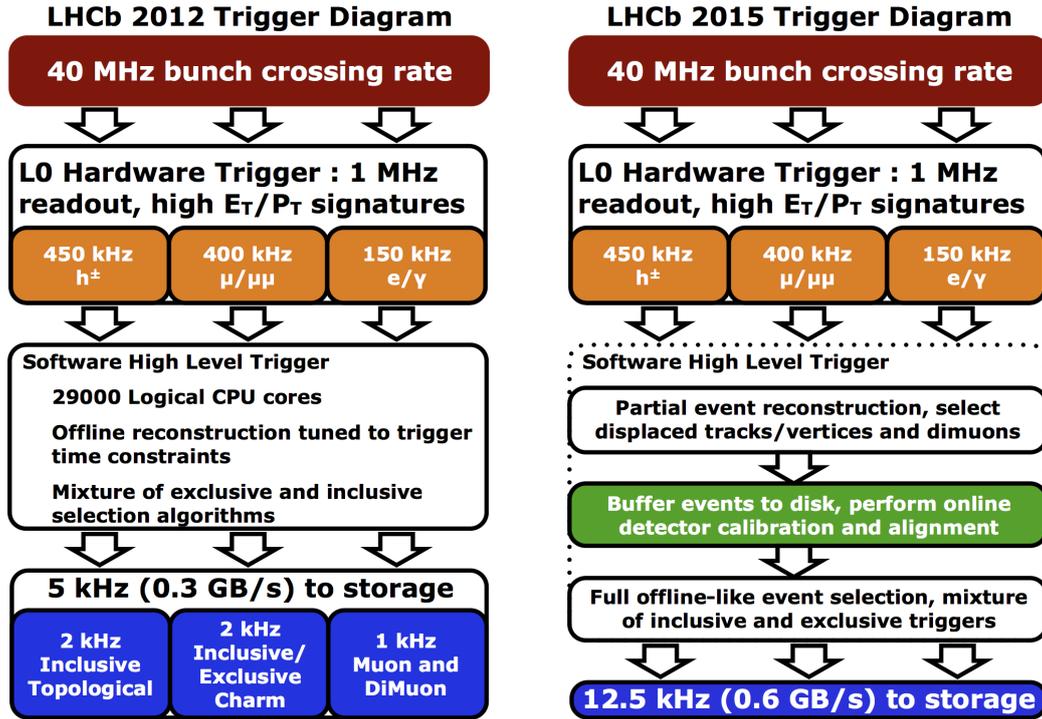


Figure 3.8: LHCb trigger architecture for Run 1 (left) and 2 (right). [263]

and includes e.g. sprucing and stripping, using pre-selection algorithms based mostly on topological and kinematical cuts. The software environment of LHCb is described later in Chapter 4.

3.2 The Upgrade I

The modernization period of LS1 is known as Upgrade I. The goal of the upgrade is to adjust the detector to the full 40 MHz readout (from 1.1 MHz trigger used before). In order to increase the signal yield for hadronic channels, which until 2018 lacked in sample size to match the theory uncertainties, the trigger system is rearranged and the hardware trigger removed completely [267, 268], as outlined in Fig. 3.9. The new triggerless¹ readout will operate at five times higher instantaneous luminosity and deal with particle fluence exceeding $10^{15} n_{eq} \text{cm}^{-2}$ (n_{eq} - neutron equivalent). The higher granularity of tracking detectors enhances their planar resolution. The number of pile-up events (concurrent collisions) will reach an average number of 7.6 [269] per bunch crossing. It is required to replace the obsolete readout systems of many LHCb components, in particular, TELL1

¹This property of the entire upgrade trigger system is driven by the new vertex detector that features triggerless pixel sensors.

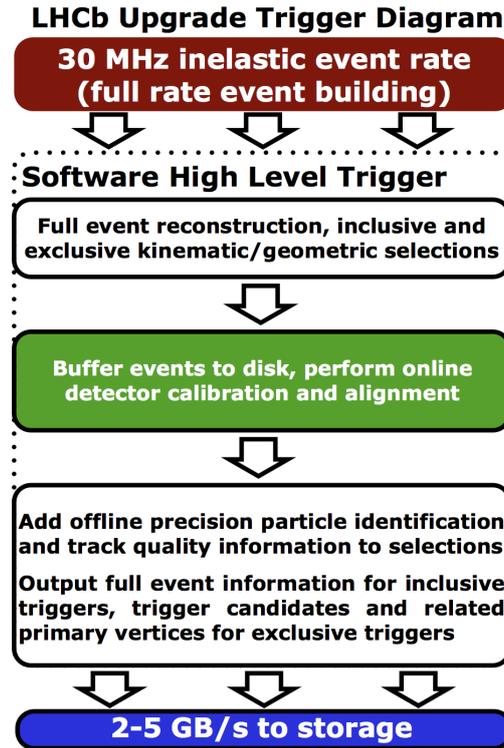


Figure 3.9: The Upgrade I LHCb trigger architecture. [263]

boards to TELL40 [270, 271], which will operate at 40 MHz frequency (a specification of new TELL40 boards is given in [272]).

The Upgrade I tracking system is replaced in a whole in terms of the Run 1 and 2 hardware. VELO undergoes a massive upgrade of the technology (strips to pixels), readout and electronics, which adjusts the detector to cope with the 40 MHz readout. The Upgrade I for VELO is explained in a separate Section 3.3. The Silicon Tracker and Outer Tracker are removed in favor of the Upstream Tracker (UT) [273] and Scintillating Fibre Tracker (SciFi) [274], which are entirely new LHCb sub-detectors. Placed behind VELO, at the exact same spot as TT before (shown in Fig. 3.10), UT consists of four measuring stations with silicon micro-strip sensors. The stations are laid out in the same manner as the removed Tracker Turicensis [237].

Two upstream planes of UT are split into 16 staves, and two downstream ones into 18 staves. Each stave is tiled with silicon wafers of about $10\text{ cm} \times 10\text{ cm}$ square shape. A pitch of the sensors varies from $190\ \mu\text{m}$ to $90\ \mu\text{m}$ in the outer and inner region, respectively [4]. The schematic view of the planes is shown in Fig. 3.11. The electronic module to read out UT strips is SALT (Silicon ASIC for LHCb Tracking) [277], which further connects to TELL40 boards.

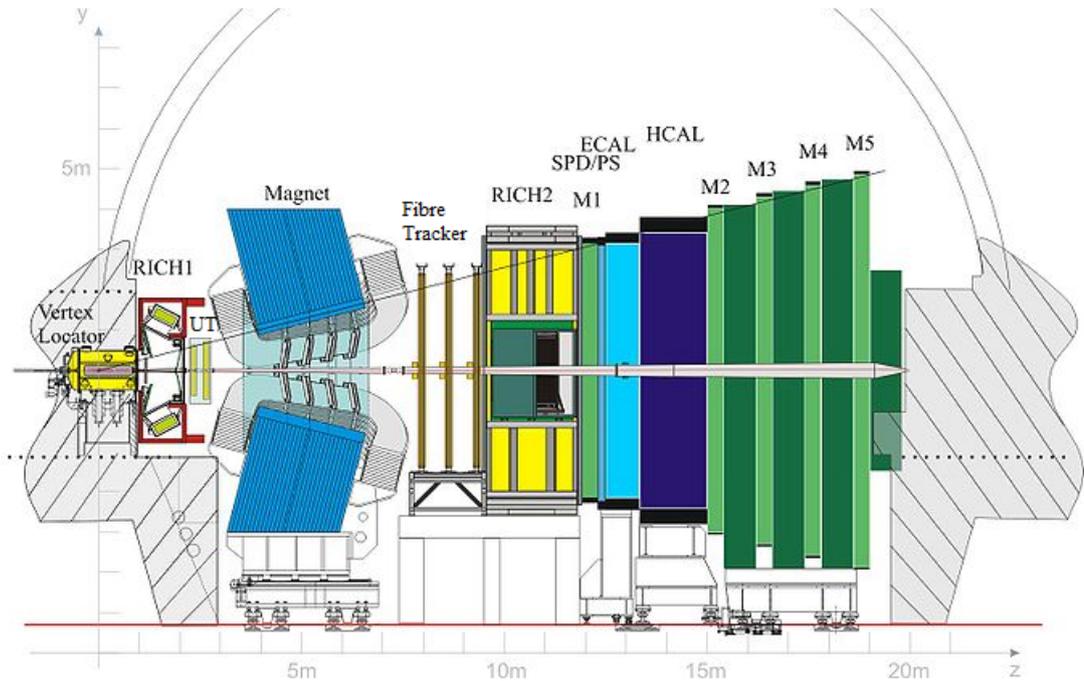


Figure 3.10: The Upgrade I LHCb spectrometer geometry. [275]

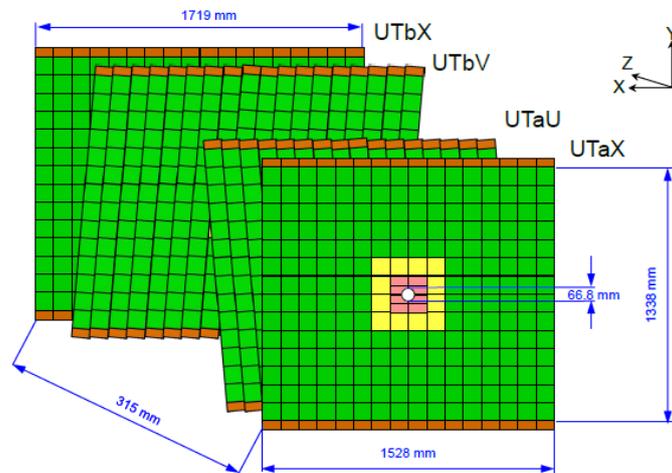


Figure 3.11: The Upstream Tracker planes geometry, illustrated downstream. [276]

The SciFi Tracker replaces the Tracking Stations. The tracker comprehends 12 stereo-layers divided into 12 modules, where each module is further divided into 8 fibre mats with 6 stacked layers of $250\ \mu\text{m}$ scintillating fibre [278]. Each plane will be mounted on a C-frame that is visualized in Fig. 3.12 once fully assembled. The scintillation light appearing in fibres is detected and measured by Silicon Photomultiplier (SiPM) [280] with accompanying readout chip Pacific ASIC [281].

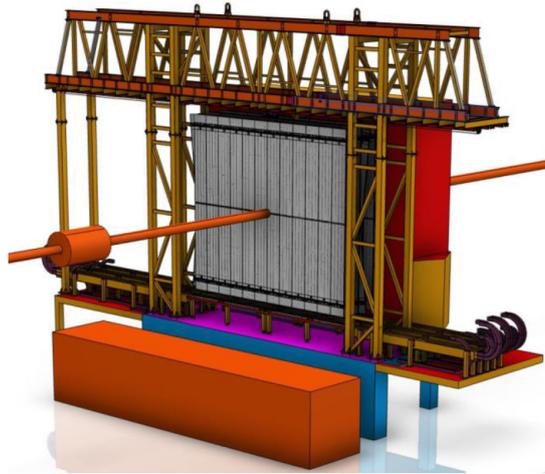


Figure 3.12: SciFi C-frame visualized when fully assembled. [279]

RICH detectors take advantage of the LS1 preparation time for the installation of the new photodetectors and electronics, improvements of the mechanics and other minor changes [282]. Calorimeters and Muon System confront mostly minor adjustments [283,284].

3.3 Upgrade I for Vertex Locator

The original LHCb micro-strip vertex detector will be replaced with a pixel detector able to operate at luminosities of around $2 \times 10^{33} \text{ cm}^{-2}\text{s}^{-1}$. Although the role of VELO in the spectrometer remains similar, so does the general profile of the detector (see Fig. 3.13), the technology behind the detection mechanism is changed significantly [285]. The new geometry of VELO is shown in Fig. 3.14, where the R and Φ modules are replaced by pixel

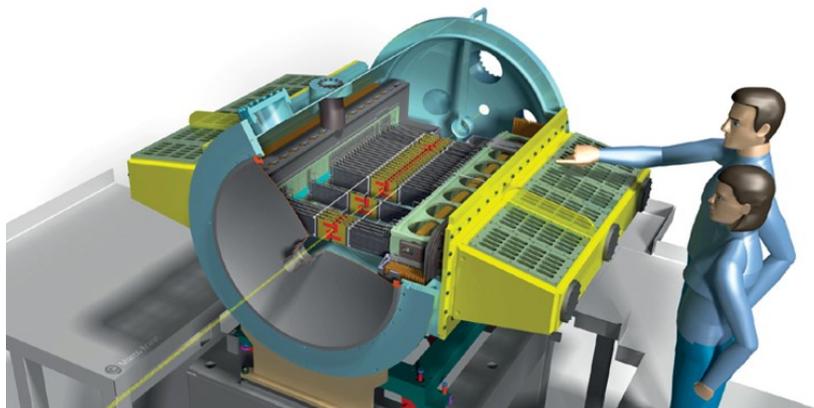


Figure 3.13: Upgrade I Vertex Locator visualization. [285]

matrices. Besides the enhanced planar resolution, Upgrade I VELO raises the nominal full-detector readout frequency from 1.1 Mhz to 40 MHz. The innermost sensors will be located as close as 5.1 mm to the LHC beams. This change, combined with the increased

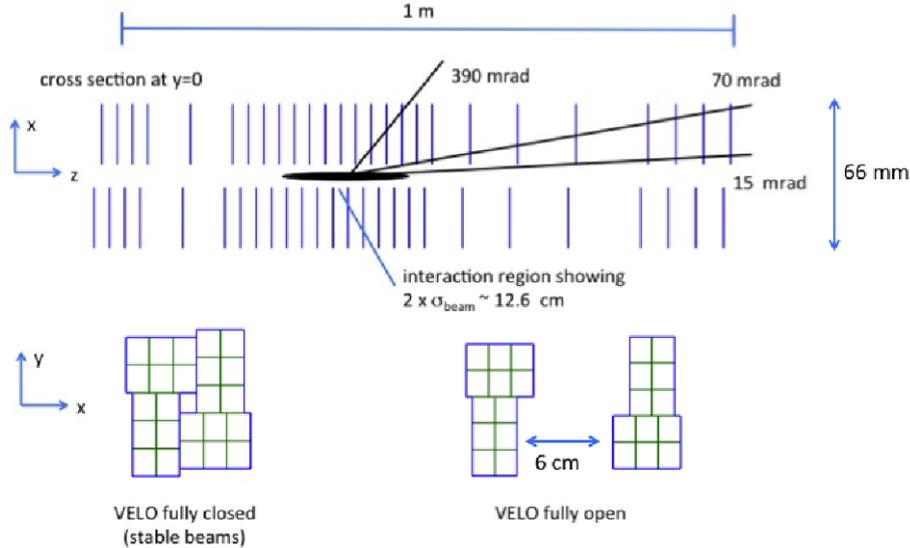


Figure 3.14: Upgrade I Vertex Locator geometry. [285]

luminosity, gives a radical rise in the occupancy, which will reach 900 Mhits/s for the hottest ASICs [286]. When expressed in the data rate units, the ASIC would produce 15 Gbit/s of raw data. The upper limit on the radiation dose foreseen for the upgraded VELO is 400 MRad. Fig. 3.15 illustrates how many pile-up events the new detector has to cope with in a function of luminosity and collision rate. A contribution of more than 6 pile-up events makes it difficult for the experiment to tackle and would not be possible with the old triggers.

An increase in instantaneous luminosity is also associated with intensifying the efforts to counteract radiation damage in the detector. The expected fluence of hadronic radiation in upgraded VELO in just one year would match the total fluence of its predecessor. Radiation damage was a significant factor already during Run 2 [287–289], especially at the final stage of the detector operation. Rapid adverse effects in VELO sensors led to a dangerous situation where the leakage currents would exceed the thermal run-away threshold [290, 291] and potentially make the sensors not operational any longer [1]. The fluence studies for new VELO supported by GEANT4 [292] and FLUKA [293] estimated the radiation to be highly nonuniform, as depicted in Fig. 3.16. The innermost sensors will suffer from irradiation much more than the sensors in the periphery region. The expected radiation damage for the upgraded vertex detector is presented in Fig. 3.16, as a function

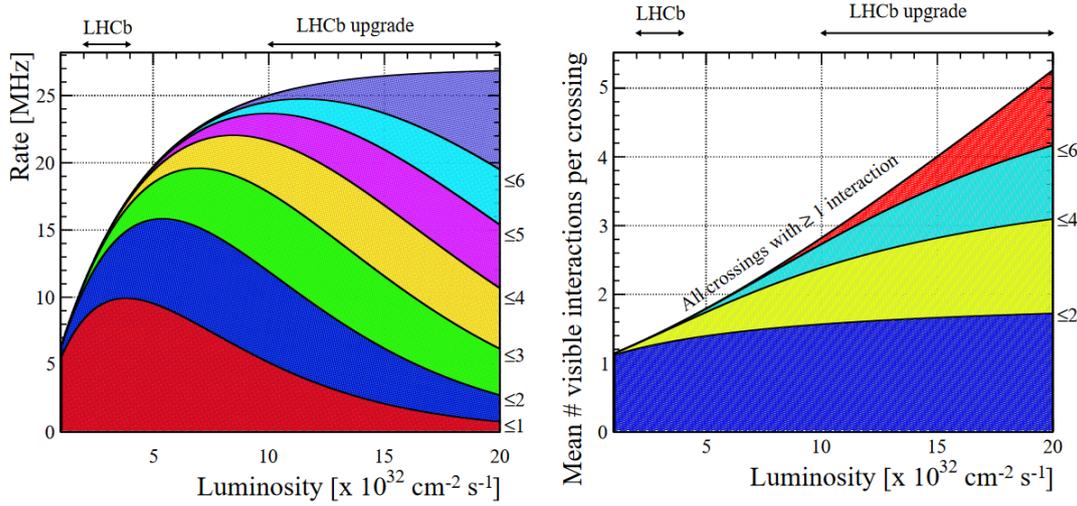


Figure 3.15: Interaction rate and average number of pile-up events (left) and average number of interactions in a function of luminosity (right). [285]

of the distance to the z-axis (consult Fig. 3.10), an axis along the LHC beam direction.

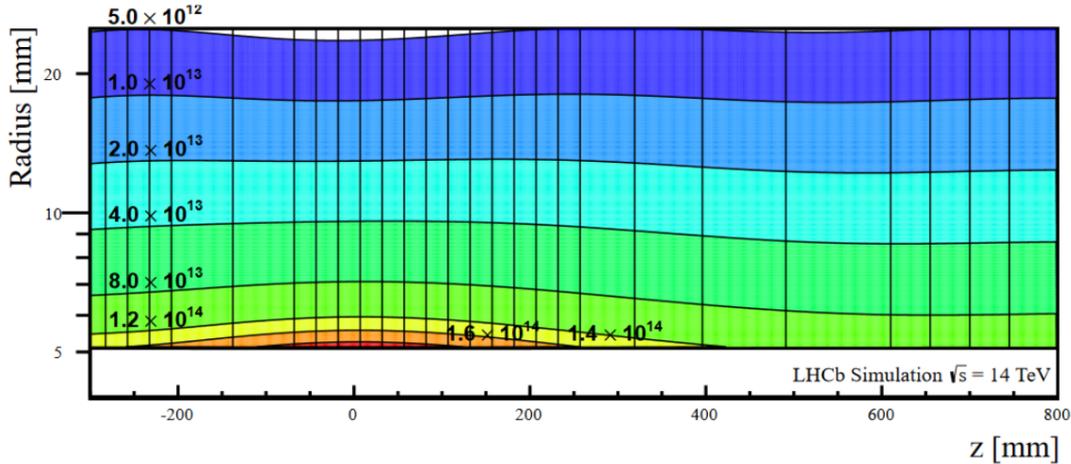


Figure 3.16: Anticipated integrated radiation dose per 1 fb^{-1} expressed in $\text{MeV } n_{eq}/\text{cm}^2$. The radius is the distance of the sensor element to the beam. [285]

Another element changing in the upgraded VELO is a Radio-Frequency Foil (RF Foil) [294]. The RF Foil separates the primary LHC vacuum from the VELO secondary vacuum, leading away the beam image currents and the weak-field induced by the beam [295]. It also shields the assemblies from electromagnetic radiation. The foil is shaped to fit the modules, as presented in Fig. 3.17. The foil for Upgrade I is only $500 \mu\text{m}$ thick to limit the material for the particles to travel through before appearing in the detector. The foil was milled from a single aluminium block at Nikhef [296] and further chemically etched

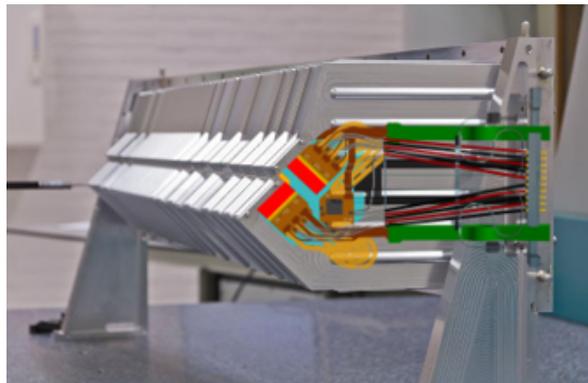


Figure 3.17: Visualization of the RF foil with VELO modules fitting inside.

to reach $250\ \mu\text{m}$ thickness in the most central region. The foil was shifted to CERN and installed in the detector with a video publicly available at [297]. The milling process and the foil after installation are presented in Fig. 3.18.

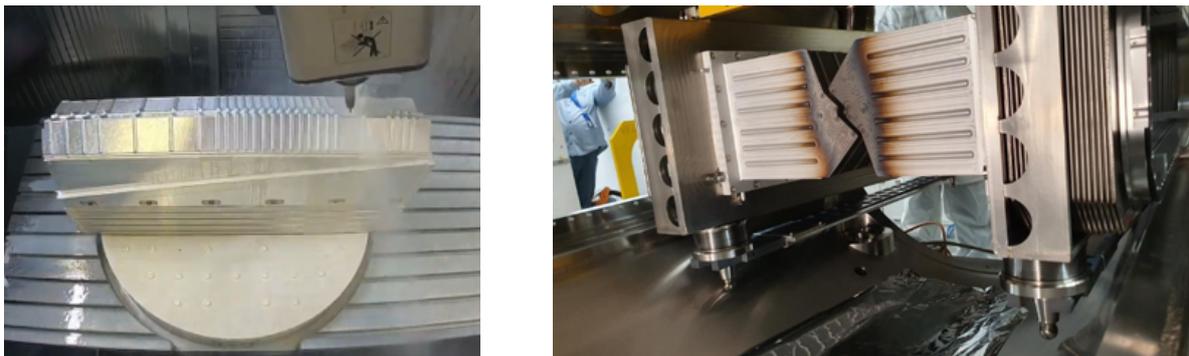


Figure 3.18: Foil at the milling (left) and once installed in the detector (right).

The cooling mechanism for VELO modules is embedded into the panel substrates for the sensors in the form of microchannels [298]. This solution was found to combine strong cooling capabilities using two-phase CO_2 with a very low material budget. The CO_2 from the cooling plant goes through the inlets and circulates in $120\ \mu\text{m} \times 200\ \mu\text{m}$ microchannels (the substrate is $500\ \mu\text{m}$ thick) and provides cooling to sensor chips and electronics [299]. The microchannels are routed inside the substrate in such a way that the temperature gradients of the whole panel are possibly the lowest. The cooling mechanism concept and panel substrate are shown in Fig. 3.19.

The readout chip for upgraded VELO is the VeloPix ASIC [300]. Its design is based on the widely known MediPix chip family [301] and is a direct successor of the TimePix [302] and the TimePix3 [303] chips. The VeloPix is built in a 130 nm CMOS (Complementary Metal-Oxide Semiconductor) technology and features the trigger-less zero-suppressed

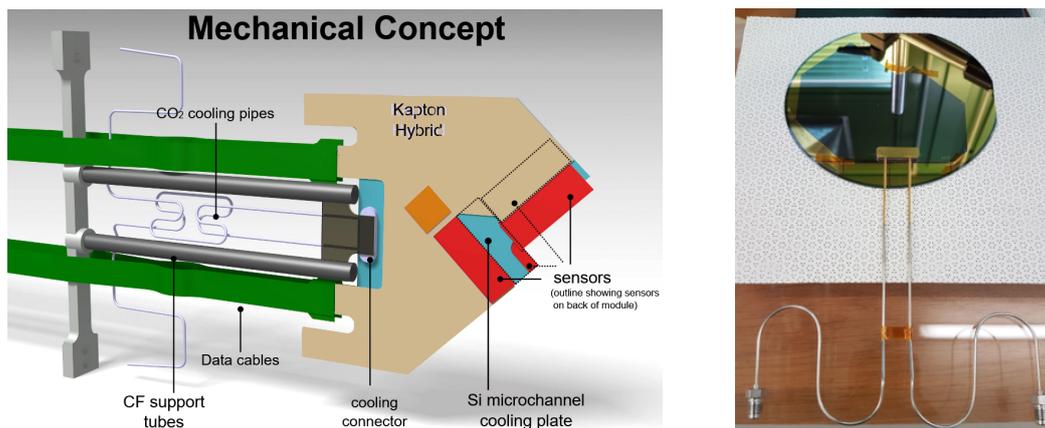


Figure 3.19: The cooling mechanism concept (left) [276] and a panel substrate during production (right).

binary readout. While only the photon counting mode was available in Medipix, a Time-over-Threshold (ToT) mode was added and embedded into the Velopix ASIC. One sensor combines three pixel matrices, each consisting of 256×256 square pixels of $55 \mu\text{m}$ pitch. Each pixel is handled by its individual readout channel, which gives over 41 million channels in total for the entire detector, around 200 times more than in the previous version of the detector. Single ASIC can transmit the data with a rate of 5.13 Gbit/s [304] through the optical fibres of the GWT (Gigabyte Wireline Transmitter) [305], a data transmission protocol being a simplified version of Gigabit Bidirectional Transmitter (GBT) [306, 307], an optical link designed for the LHC detector upgrades [308]. The total data stream from the whole detector can reach a transcendent volume of 12.1 Tb/s. The data rate from the ASICs is highly inhomogeneous in a function of the ASIC distance to the beam, which is pictured in Fig. 3.20.

A module provides mechanical support for 4 VeloPix sensors (also called triplets), which gives 12 squared matrices, 256×256 pixels each, in total. The module is mounted on the aluminium foot and comprises panel substrates, front-end hybrids, GBTx hybrid [310] (a chip for processing control signals) and ASICs. A visualization of a single VeloPix triplet at the final stage of development in Fig. 3.21, while a fully assembled module is shown in Fig. 3.22 with the components explained. More details on the hybrids and electronics in upgraded VELO can be found in [311]. The nomenclature of the ASICs is $\text{Mod}x\text{VP}yz$, where x, y, z are the module number, the tile (sensor) number and the ASIC number, respectively.

The VeloPix chip groups the pixels into logical structures Super-Pixels (SP) in the form of 2×4 pixel rectangles. Once a pixel in SP is hit, the SP word is created with a map

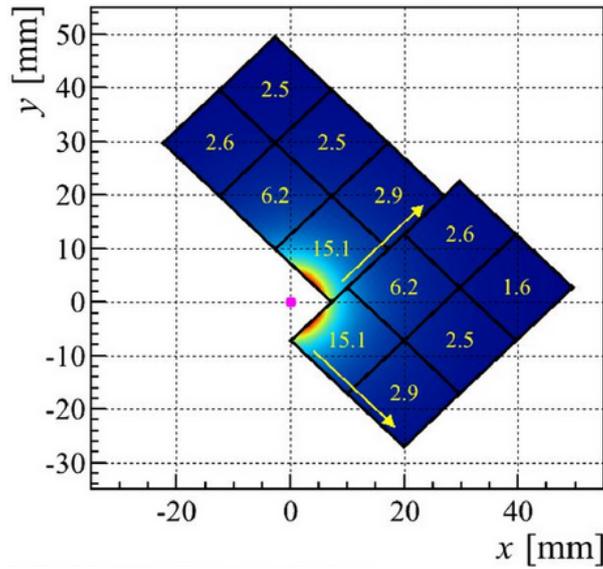


Figure 3.20: Simulated data rate in Gbit/s across 12 tiles in a single module. [309]



Figure 3.21: The VeloPix ASIC at the final stage of development.

of hits within SP, an address of the SP, and a 10-bit Time-of-Arrival (ToA). The SP word is then driven across the column along the other SPs to the End of Column (EoC) by the readout bus. The SP packet arrives at the data fabric nodes and later to the router, where the data are forwarded to the GWT links and Low-Frequency (LF) serializer and eventually brought off the chip. The complete readout logic is presented in Fig. 3.23, and explained in detail in [312].

The pixel front-end is presented in Fig. 3.24. The analog readout consists of a charge-sensitive preamplifier with a 3 fF feedback capacitance and the leakage current compensation component. A 4 fF capacitor is connected to the preamplifier to allow the usage of Test Pulse (TP).

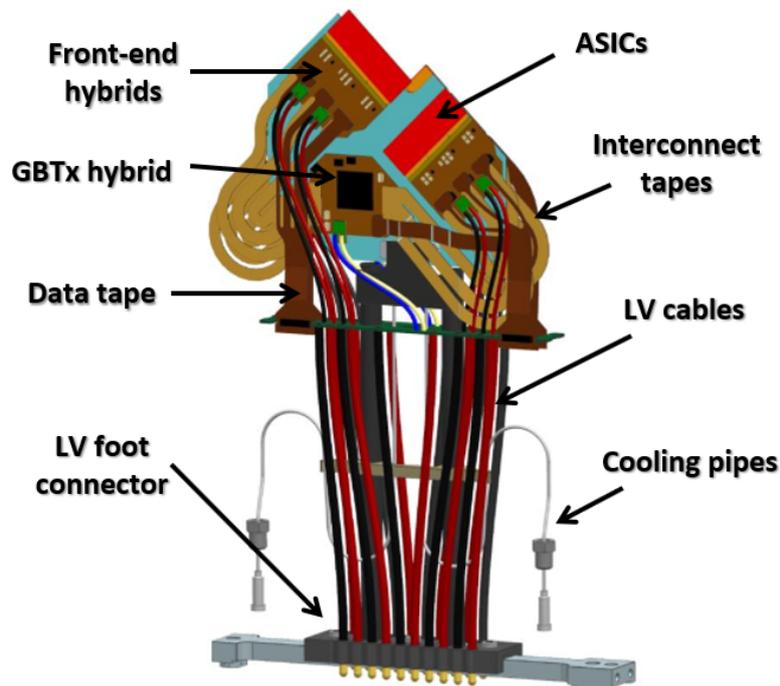


Figure 3.22: A fully assembled Upgrade I VELO module.

The preamplifier is constructed with the Krummenacher feedback architecture [313] for electron collection and was optimized to work with 50 fF load capacitance. The output of the preamplifier is linked to the 40 MHz discriminator that compares the given global threshold value (a signal threshold) to the analog signal from the preamplifier. The discriminator takes into account a 4-bit Trim DAC configurable of the threshold shift. The Trim DAC is used for calibration purposes to account for variations in individual pixel properties, where it equalizes the effective signal threshold across the matrix to the closest possible value. For instance, if a trim for the given pixel was set to be 1 ('0001'), its value of around 10 mV is added to the signal in the discriminator. The process is explained further in Section 4.3. The digitized signal goes through a masking gate. The masking gate is used later in calibration to exclude bad or damaged pixels, as one noisy pixel can fill up to a fourth of the maximal bandwidth of the SP column readout bus. The digital front-end features a ToT threshold component used to discard the signals that do not last enough time expressed in ToT units (i.e. electronic clock units). A 6-bit ToT processor of LFSR (Linear Feedback Shift Register [314]) allows to measure ToT of the signal up to 63 clock counts, that is 1575 ns for a 40 MHz clock. If a valid event is sent to the SP readout, it creates a bit word using the FIFO method (first in, first out) with ToA information added. ToA is expressed in a 9-bit BX ID unit (BX ID - bunch crossing identification).

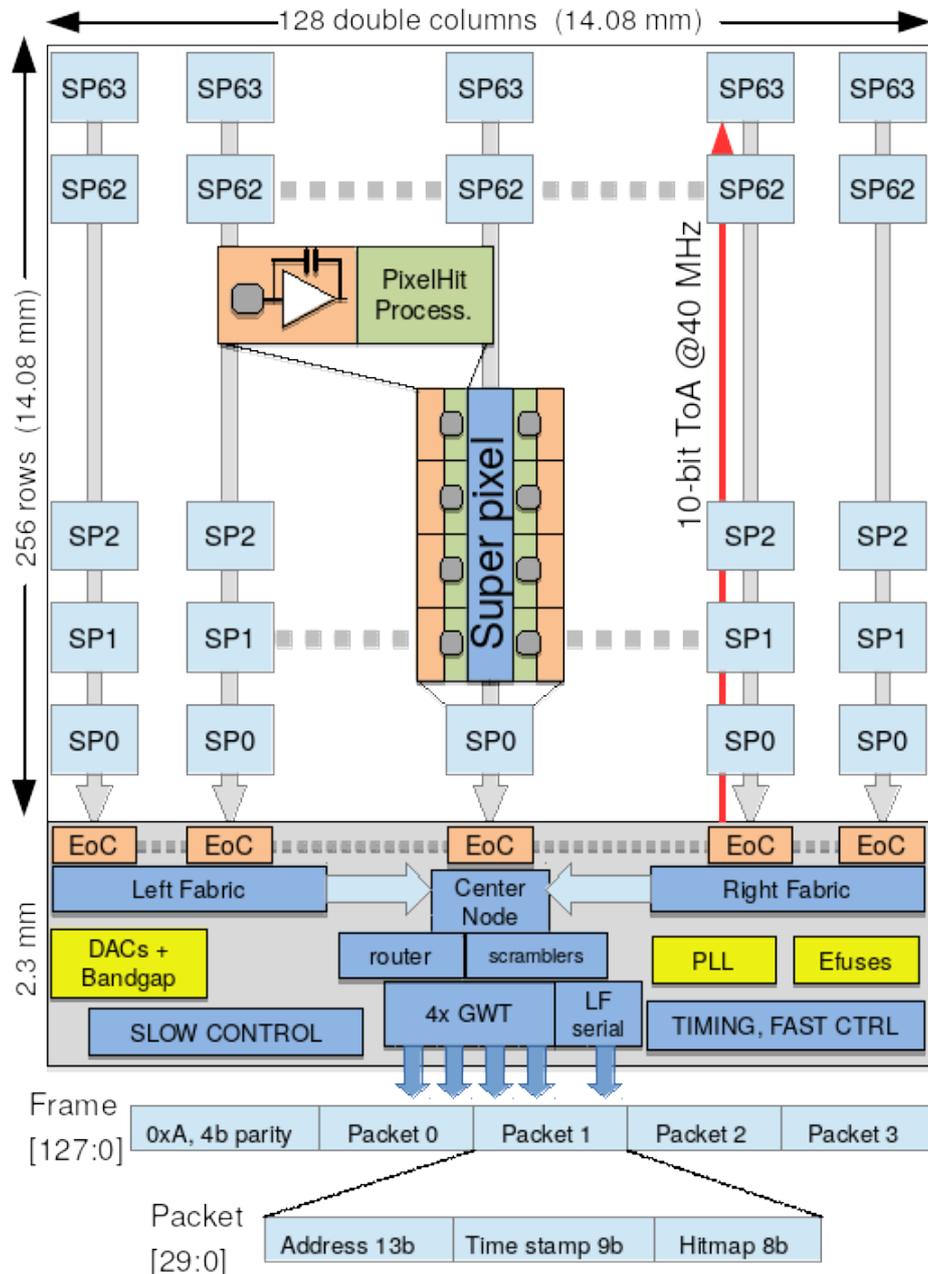


Figure 3.23: The Velopix Super-Pixel (SP) logic readout after front-end stage. [312]

The whole front-end readout described above is presented in Fig. 3.24. Bias voltages in the front-end are generated by the analog periphery to match the user-defined DAC (Digital to Analog) settings.

The DAQ (Data Acquisition) system is an intermediate element between the VeloPix readout and CPU clusters. GWT links transmit the data to the Optical-and-Power Board (or Opto-and-Power Board, OPB) [315], a board that controls signal transducers and bias voltages for the ASICs. TELL40 boards later condition the signal before sending it

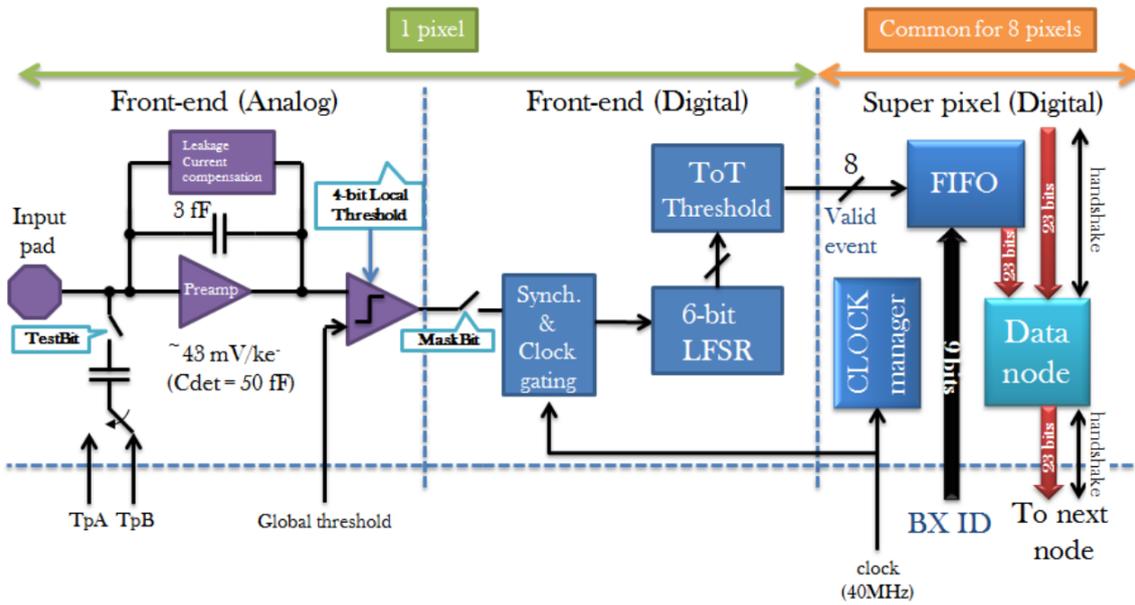


Figure 3.24: A scheme of a single pixel front-end readout channel. [312]

to the CPU farm, adding timing information in data packets named Supervisor ODIN (SODIN) [316]. More details on this mechanism can be found in [317] and in [318] in a wider context, as the system is common for all LHCb detectors.

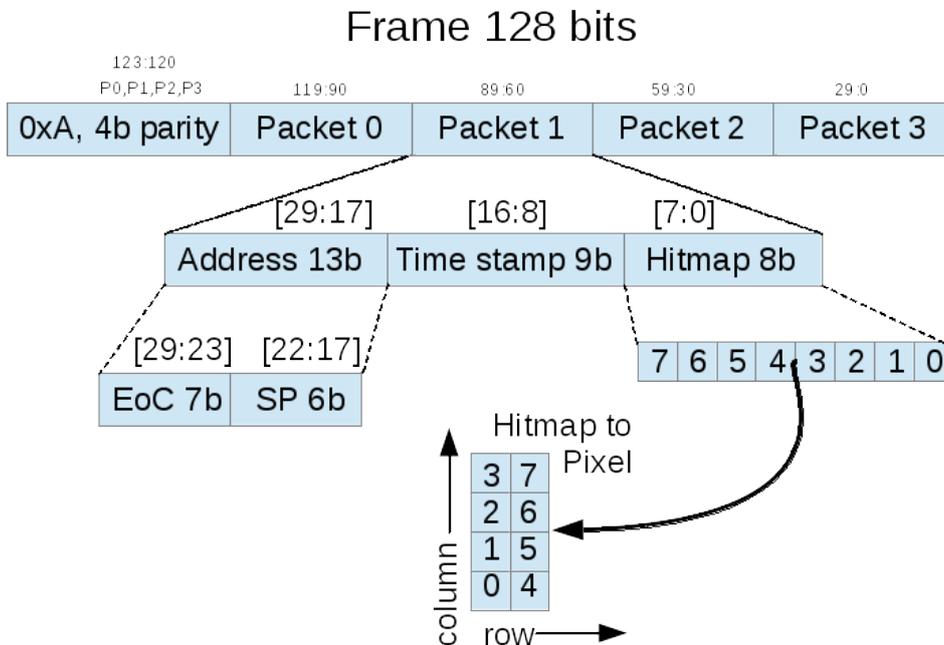


Figure 3.25: The GWT data frame with an SP word explained.

The GWT protocol encapsulates the hits coming off four different SPs into one data frame driven further out of the chip. If more than four SPs in a column register a hit, the

signals queued further in the readout bus are discarded, and the information about the hits is lost. Although this limit seems sufficient for standard data taking, it makes any pixel scanning procedure hard to design. The format of the GWT data frame is presented in Fig. 3.25. Four different SP packets follow the main header, and each packet consists of a 13-bit SP address (an ASIC ID, column and row coordinates), a 9-bit timestamp of ToA (BX ID) and an SP word, which is an 8-bit hitmap encoding the lighted pixels in the SP.

The ECS (Experimental Control System) communication protocol can be used for testing and scanning the VeloPix matrices. This 6-bit analog readout registers either the number of hits for each pixel or ToT of the last registered hit in the given shutter. The advantage of ECS is that all pixels can be read out at the same time, which is time-efficient for any pixel scanning procedure. However, the application of ECS is restrained due to its susceptibility to noise, especially that associated with opening the shutter. The majority of control scans and part of the calibration process make use of ECS to shorten the scanning time, leaving the primary GWT protocol mainly for physical data taking purposes [3].

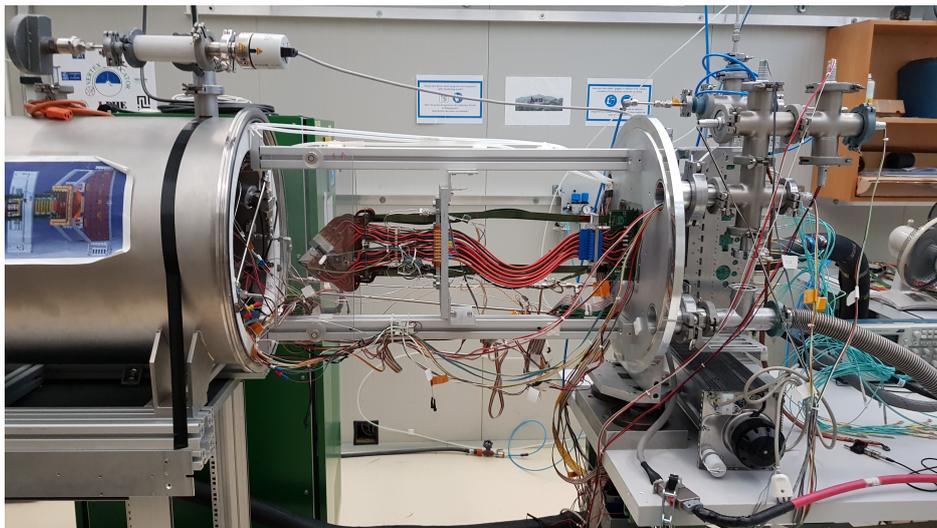


Figure 3.26: The slice test module setup at the VELO lab at CERN.

For the commissioning of the modules and the readout, the development setup was built at CERN for the slice tests, which included a full assembled module resembling the final VELO environment. The module was installed in a vacuum tank with CO₂ cooling connected to the model substrate. The setup incorporated full readout electronics, a cooling plant, high-voltage and the voltage interlock system. Nearby, a smaller setup was built for testing the software and OPBs. The setup was made up of 4 VeloPix triplets enveloped by aluminium shield and therefore was named Aluminium Sandwich (see Fig. 3.27).

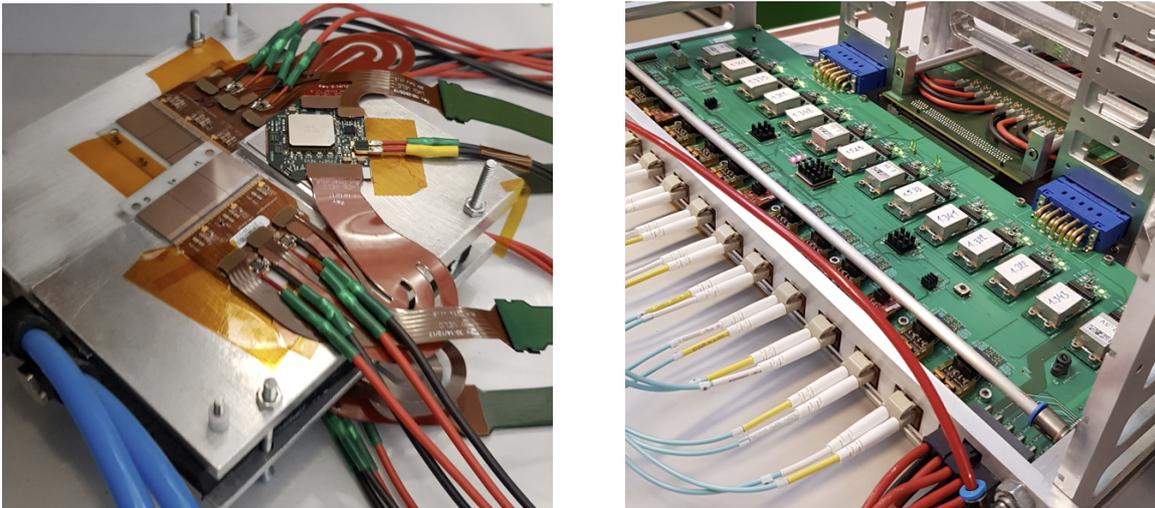


Figure 3.27: Aluminium Sandwich (left) and its readout OPB (right).

A few testbeam series (2015-2018) at the SPS (Super Proton Synchrotron) testbeam at CERN Preveessin gave the opportunity to test the irradiated sensors under the beam. The studies took advantage of TimePix3 Telescope [319, 320] to measure a range of sensor properties and performance. The telescope was built mainly for high-resolution tracking [321] and R&D (Research and Development) of vertex detectors [322]. It combines 8 planes with sensors bump-bonded to TimePix3 ASICs. The planes can move and adjust the pitch with respect to each other. A Device Under Test (DUT) is mounted in the middle of the planes. Many studies are available with this setup [323]. The project scheme of the telescope is shown in Fig. 3.28. After a new pixel detector readout chip TimePix4 [324] was released in 2021, featuring an excellent timing resolution of sub-200 ps [325], the timing resolution of the telescope can be further improved.

Before the slice tests and the aluminium sandwich mentioned above were arranged, a developing stage setup for the early versions of the VeloPix was assembled at CERN and a few cooperating facilities, including Krakow. The setup used a SPIDR system to program and read out the VeloPix. The SPIDR (Speedy Pixel Detector Readout) [326] is a readout system for Timepix family ASICs developed by the researchers from Nikhef [327], with its functionality further enhanced on the Velopix ASIC, which was possible due to its general-purpose readout mechanism. The SPIDR supports 1 and 10 Gbit Ethernet, which drives the ECS readout. The data can be passed through the FPGA to the computer with the drivers and firmware installed. SPIDR uses the Xilinx VC707 evaluation board [328] as FPGA (Field Programmable Gate Array). The setup is suitable for studies on ASIC readout and the software, delivering a ground for stress tests of the scanning procedures.

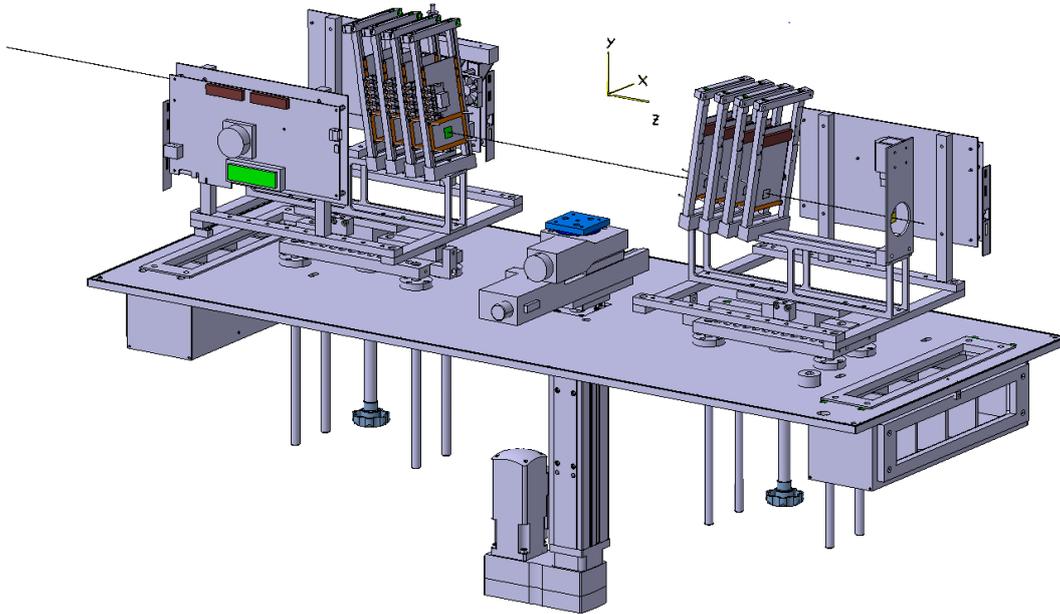


Figure 3.28: A scheme of TimePix3 Telescope at SPS at CERN. [319]

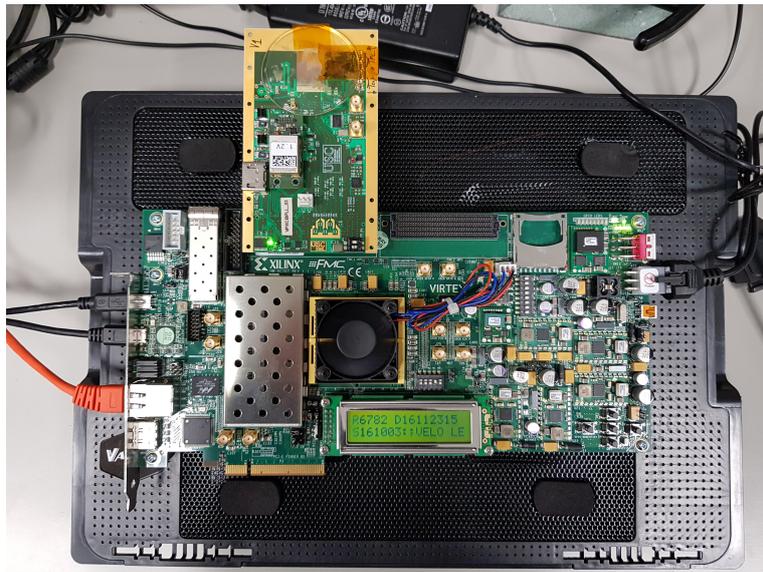


Figure 3.29: The VeloPix and SPIDR setup in Krakow.

The fully assembled setup in Krakow is shown in Fig. 3.29. It is worth noting that the Krakow test setup was completed and operated by the Author. It was used in development studies of the software components of the detector daily operation software.

The final modules of VELO were constructed at Nikhef and the University of Liverpool mostly during the Covid-19 pandemic (2019-2021), mounted on frames of A and C side

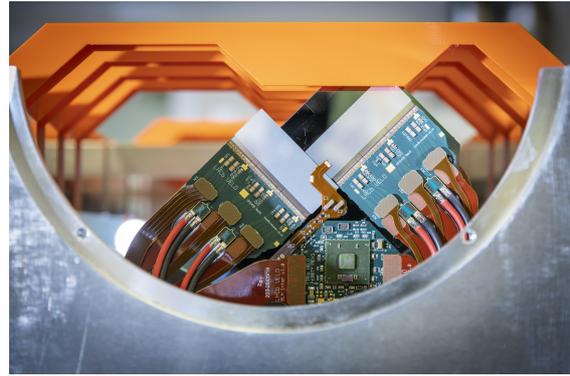
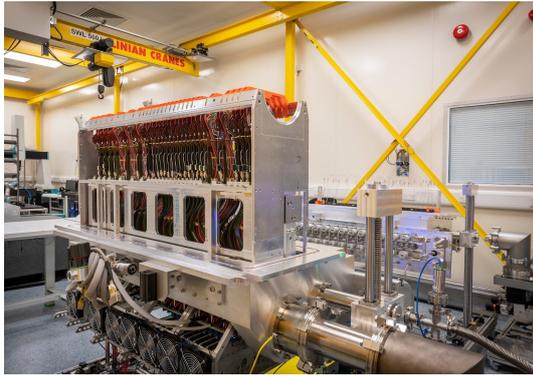


Figure 3.30: The Upgrade I VELO C-side assembled.

(two VELO halves), shifted to CERN and installed in LHCb cavern, on 2nd March 2022 and 11th May 2022, for C and A side respectively (see Fig. 3.30 for C-side).

Chapter 4

Software for Upgrade I VELO

The operation software is a critical element of the detector system and takes part in almost all aspects of its functionality. It is responsible for handling the chip readout, decoding and analyzing the data, building the events, and finally calibrating and monitoring the detector. The following Chapter describes the operation software, putting a focus on the calibration and monitoring, which are parts of the Upgrade I commissioning to serve in LHCb for the next Runs. The Chapter is organized as follows. The LHCb software environment and *Vetra*, a platform for detector calibration, are explained in Section 4.1. The decoders are discussed in Section 4.2. The detector calibration is explained in Section 4.3 and covers both hardware and software aspects of the process. The detector monitoring is discussed in Section 4.4, the signal scans in 4.5 and finally, the data emulation, in Section 4.6. The *Vetra* platform and all its components presented below were designed, implemented and tested both in the lab and in the operational environment by the Author of this thesis. The text in this Section is, in some places, very technical and specific to the LHCb vertex detector. It is not only a report from the research activities but also a source of information for the experts that will operate the detector throughout Runs 3 and 4. It was written during the commissioning period of the upgraded detector.

4.1 The LHCb ecosystem

A major architecture built within LHCb to organize software of all kind is Gaudi [329] application server, a C++ ecosystem with Python elements acting as pipeline glue¹, for high-level settings. Gaudi is geared to work with big data and has many built-in

¹Python as a glue language is a jargon notion meaning that it can be used to bring different libraries, written in different languages, to work together.

functionalities that allow fast and optimized processing of such. The processing elements pertain to the LHCb hardware mechanism and organize data into events. The event is a basic structure containing instantiated detector objects that participated in the event detection or processing. A single event is then a collection of objects (produced particles) and active detector elements that respond within the time window of a single bunch crossing. The *EventSelector* service reads the data of the Python level such that the raw data banks can be further extracted by the C++ preprocessor. Gaudi encompasses multiple services for both physics analysis and detector studies and stands a part of LHCb Software Continuous Integration System [330]. A configuration file defines sequences of the algorithms, while the options file specifies the low-level options. *LHCbConfigurableUser* defines high-level settings and serves as a handbook for documentation. Gaudi allows specifying the *LHCbApp* (a class defining basic information about the data to analyze) conditions, *EventSelector* options, output configuration and optional histogram persistency. A Gaudi kernel (a commanding unit) is created when a user runs the application, and the algorithms are later executed in the C++ framework. The classes mentioned above are documented in detail in [331].

Among quite a few Gaudi applications in LHCb, the most recognized are; Brunel for decay reconstruction [332] (becoming obsolete in incoming Run 3 due to the removal of hardware trigger), Boole for digitization [333], DaVinci for physics analysis [334], Moore for HLT application [335, 336], and Gauss for physics event generation and detector simulation [337, 338]. The extended description of how those applications interact with each other can be found in [339].

All elements of the detector software created as a part of this thesis were deployed in one large Gaudi application, a software platform Vetra [4]. Vetra originates from the application of the exact same name from Run 2. Vetra in the past was a data processing project which emulated the readout electronics and TELL1 trigger processing boards [340], and was used for the detector simulation [341]. The original Vetra was able to simulate the entire readout logic, including the Beetles. Related front-end project Lovell [342] served machine learning analysis for pedestal² trending using Vetra as a back-end. The whole software was instantiated on LHCb-nightlies [343], a platform for automatic system building and testing. At the end of Run 2, Dendek founded an individual branch of the testbeam application Tb-UT to handle Upstream Tracker tests [344]. The project was further enhanced by Rachwal et al. by encompassing the other LHCb tracking detectors

²Pedestal in the previous Vertex Locator was a baseline of the front-end buffer in each channel.

and making it a universal tool for the testbeam studies TbGaudi [345].

Vetra in Upgrade I is intended to combine the software system for Vertex Locator and some functionalities of the Upstream Tracker monitoring software [4, 346]. In fact, these parts were developed independently and are only maintained within the same name without actual merging of the systems (the Author is responsible for the Vertex Locator part). Since the design of the project and the infrastructure was rearranged, Vetra for Vertex Locator was built from scratch (the new project only retained its name, all of the components were designed anew). The platform is used for decoding, calibration, data quality monitoring, pixel performance studies, analog readout optimization and emulation of the calibrated detector. The platform is able to run on user-defined recipes, which build a sequence from the block elements. This segmented structure makes Vetra flexible for processing any type of analysis on the data stream from the detector. Vetra for Upstream Tracker [347] was developed by Dendek and Krupa et al. and includes the RAW data decoding and the readout electronics monitoring. Vetra is available on GitLab [348] as the LHCb official software.

A specific code architecture Gaudi::Functional [349] is used, which is a thread-safe framework with optimized transient file processing (transient files are kept in the memory over the whole application execution). The platform is adapted to C++11 standard and multithread operation. Gaudi::Functional features class inheritance from the templates with a specific relationship to the transient data; there are several algorithms that differ depending on input and output connections with the transient objects. Algorithms with no inputs inherit from Producer base class, algorithms with one input and one output from Transformer, with multiple outputs from MultiTransformer, and algorithms with no output from Consumer.

Vetra is equipped with ML tools for fast and time-efficient fitting, data validation and developing trend analysis. The platform is built on the lb-stack-setup, the LHCb software stack [350], allowing it to install and import algorithms from the other applications distributed in LHCb ecosystem. A stack is created during the installation process, which mounts the software dependencies as specified by the user. Some of the essential LHCb dependencies and kernel files are installed irregardless as they are needed for the proper functioning of Gaudi. Vetra shares software dependencies with Lbcom [351], Gauss, Boole and Moore Gaudi projects.

A connection with the other LHCb software elements can be established straightforward. Vetra closely cooperates with STORCK (Data Storage and Tracking) system, an online

SQL (Structured Query Language) database service for detector-related data [352, 353]. Vetra connection with Siemens WinCC Open Architecture (WinCC OA) [354], used for managing the communication with the detector [355], is discussed further in this Section. The output of Vetra is used to build and upload calibration recipes for the detector, which are critical elements for daily operation. In short, the recipe is a binary file guiding the WinCC OA in deploying the calibration and contains all calibration and hardware settings necessary for the detector to run. It defines the condition of the detector and has a vital impact on the data quality. Vetra can connect to the software for the local commissioning monitoring Titania [353] and online monitoring Monet [356]. Vetra also takes part in VELO closing procedure [357] and possibly in clustering and vertexing in the future, as those functioning in Moore consume unnecessary much CPU time [358].

4.2 The decoding

Four different decoders are available in Vetra to use and customize. *VPDecoderDIM* decodes a DIM (Distributed Information Management [359]) format. DIM is a communication system for distributed environments [360]. Its integration with the LHCb data streaming can be found in [361] and to PVSS (a system of Supervisory Control and Data Acquisition [362]) in [363]. DIM is used to communicate between data acquisition machines and the detector, and further between the WinCC OA system and the DIM servers running in the LHCb network³. *VPDecoderFRG* decodes the FRG (FRG - Fragment) data frames. The FRG frames are raw data of GWT (Gigabyte Wireline Transmitter) bypass [364], a communication path between the SP (Super-Pixel) readout and the CPU farm, which bypasses the GWT links. *VPDecoderMDF* decodes the MDF binary raw banks (MDF - Mast Data File). MDF is a special LHCb format for event-structured data in a Gaudi sense (see example in [365]), where the term event corresponds to one physics event produced in a single bunch crossing. MDF is the default format used for data processing in trigger during regular data taking. *VPDecoderCSV* reads the standard compressed CSV files.

A decoder should stand as the first element in a sequence unless the data are already decoded. A bunch of predefined sequences were implemented in Vetra to provide template solutions. Each predefined sequence has its own options file. Unfortunately unavailable outside the CERN network, a technical details on running the template decoding sequences can be found in Vetra TWiki internal documentation [366].

³The LHCb network, or the online network, is a network for software, data acquisition systems and monitoring of all spectrometer components.

Table 4.1: A DIM header in the commissioning format, with assumed a threshold scan type for the config header.

Subheader	Total [b]	Quantity	Size [b]
Significator	2	0xCBCB	2
Data	6	Module	1
		Tile	1
		ASIC	1
		ASIC unique	2
		Trim	1
Generic	30	Type	1
		Date	8
		Operator	13
		e-log entry	4
		Runs	4
Config	40	No. of step	4
		Step	4
		Shutter length	4
		Min threshold	4
		Max threshold	4
		Reserved	20

The DIM decoder processes the stream of the DIM writer, a communication system which listens to the individual detector modules, and which is placed externally to the WinCC OA environment. It catches and processes the data coming from the ECS (Experimental Control System) detector readout through Slow Control (Slow Control is implemented in GBT chipset [367]). The DIM writer produces a 6-byte format of LFSR (see Section 3.3), which requires a six-to-eight bitwise mapping to decode. *VPDecoderDIM* is a Gaudi Producer, and alike to the other decoders, it is suited to open the Vetra sequence. It produces frames of 256x256 numbers corresponding to the 256x256 pixel map of the VeloPix ASIC. When a sequence initializes, *VPHeader* object is instantiated to read out the 78-byte binary header with data and scan information added earlier in the DIM writer. The format of the data stream was designed by the Author to meet the system requirements, and the header of this data stream is presented in Tab. 4.1 (or [368]). The header is constructed in the WinCC VeloPix library and prefixed to the data stream that is later sent to the DIM writer. The header consists of a significator 0xCB0xCB and combines three specific parts. The data header combines basic information about the chip, following the nomenclature convention in [369]. The generic header describes the purpose of the data stream, e.g. the type of the scan. The config header is scan-specific and contains the scan details. The DIM decoder uses *VPHeader* to read the scan specification and to convert the binary header to the human-readable version. This version is later prefixed to

the decoded data when saved as CSV format. The specification read out from the header is intended to guide Vetra when analyzing the data, although it is possible to redefine the specification in the options file.

VPDecoderMDF processes the MDF stream coming either from the GWT or the simulation software (MDF is the standard output of the High-Level Trigger). Because of the format complexity, *VPDecoderMDF* is the most advanced decoder in Vetra. *EventSelector* builds *RawEvents*, which are structure objects of the detector events (detector banks), extracting the data from MDF. A support for MDF readout is provided by IOHelper (Input/Output Helper), a Gaudi internal service⁴. Vetra searches for the VELO-type banks once *RawEvent* is stripped, and together with the SODIN bank, they are instantiated in TES (Transient Event Store, a transient data memory space in Gaudi - see [370]) in the form of *VPDigits* objects. Objects in TES are accessible in any further sequence elements. The structures saved to TES for each *RawEvent* are a 48-bytes header (the event-specific header), a SODIN bank and up to 208 VELO banks, corresponding to 208 sensors in the detector, stacked bitwise on top of each other.

Each raw event is tagged by a bunch crossing identification of the trigger (BX ID) and granted an event identification number (EventID). The bank is either a TELL40 or SODIN entity, with the source marked by a 16-bit ID (SourceID). The SourceID is used to differentiate which part of the detector the data fragment comes from. The raw event header is a 48-bytes block with the basic information about the event and its content, such as the checksum of the size for the sanity check (used to check whether the banks are fully extracted from MDF). The ODIN fragment is a 48-byte structure with Timing and Fast Control (TFC) [371] properties of the triggered event. The bank contains a few important parameters for the calibration. The bank is a stack of 5-byte header and ten 4-byte unsigned integers, giving a total of 48 bytes. The first three integers are the run number, the event number, and the step number, where the latter is used when scanning the sensor to express the step of the scan. The integer of the step number shares two bits for the error type used for SODIN diagnostics. Further bytes are GPS time (Global Positioning System time), TCK (Trigger Configuration Key), partition ID, a word for custom properties and the BX ID that counts from 0 to 3563, to wrap around the full LHC orbit [372]. The eighth integer is the orbit ID, that resets its value to 0 each time a new run is started. The last two words represent the EventID. More details of the SODIN format and mechanism for Run 3 (the specification was changed in Upgrade I) can be

⁴IOHelper handles the optional file compression if the MDF data are large.

found in [373].

The data banks resemble the structure of the ODIN but feature their own detector words. The VELO bank is built of the header specifying a sensor identification and 4-byte SP words, each corresponding to the SP that registered a hit in the BX ID specified in the ODIN fragment. Following this logic, if three SPs in a sensor encounter hits, three SP words are added to the bank. This mechanism is the same during the detector operation, where the SP words make up the physics data that are sent out by the detector. If a sensor encounters no hit, a bank is not created. The ODIN fragment is not skipped unless there are no other banks tagged with the same BX ID.

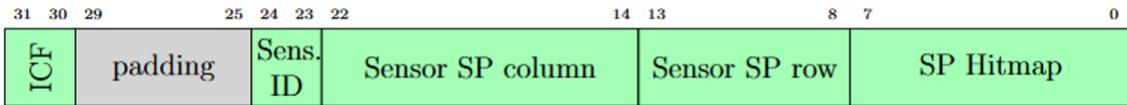


Figure 4.1: The Super-Pixel word in the VELO bank data fragment.

A structure of the SP word is shown in Fig. 4.1. The first eight LSBs (Least Significant Bits) represent the SP hitmap, defining specific pixels that were activated within the SP. The SP row number varies from 0 to 63 (along the longer edge of the SP) and SP column from 0 to 383 (shorter edge of the SP). The 9-bit SP column counts the columns per tile and not per ASIC (each tile bonds three readout chips). Sensor ID is a two-bit number (virtually 0b00, 0b01 or 0b10). The ICF (Isolated Cluster Flag) indicates whether the SP packet is isolated (the packet is isolated if neither from the closest SP neighborhood is activated). This flag was intended to support the clustering in CPU, but in the current version of the system it is not used in the process. Because a VELO bank is created if at least one pixel is activated in a certain event, its minimal size is bounded to 12 bytes (8 from the bank header and 4 from a single SP word). The exemplary structure of the simplest VELO bank with one SP word added is shown in 4.2.

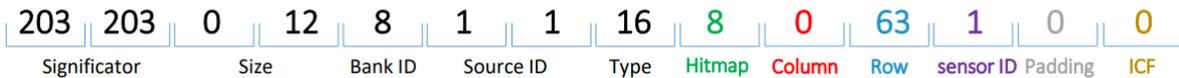


Figure 4.2: The SP word in decimal form (after order segmentation from Fig. 4.3).

Once the decoder parsed the events, the ODINs are separated and sorted according to the timing specification. The decoder iterates over the banks and builds objects that represent individual pixels (*VPDigits*). They are forwarded to the subsequent algorithm

that the decoder was associated with in the configuration file (or in a template sequence). With a decoder as the opening piece, any sequence can be created on demand, in a transparent and flexible manner, depending on what kind of analysis (or monitoring) is to be carried.

Ob00000000**0** **Ob**10000000**0** **Ob**00111111**1** **Ob**00001000**0**
ZOb 00001000 00111111 10000000 00000000
 (**H**itmap, **r**ow, **c**olumn, **s**ensor ID, **p**adding, **I**CF)

Figure 4.3: The endianness of the SP word, originated in the firmware LE (top) translated to BE (bottom), marked by Z (for BE unsigned integer).

A stand-alone reader and decoder for MDF is available in Vetra repository [348]. The binary events are written in Little Endian (LE) order, so each consecutive byte within the 4-byte unsigned integer is inverted. For compatibility reasons Vetra changes the endianness to Big Endian (BE) order instead, and so does the stand-alone decoder. Because some entities are shared between the bytes, the decoding needs to replace the bits (see Fig. 4.3).

The MDF decoder, if used in Vetra, forms a sequence as shown in Fig. 4.4. The high-level configuration (communicated with the options file) defines the sequence and runs the Gaudi services. Transient Event Store handles the raw data as instructed by *EventSelector* (it does not load a whole event stack at once to smoothly process large MDFs).

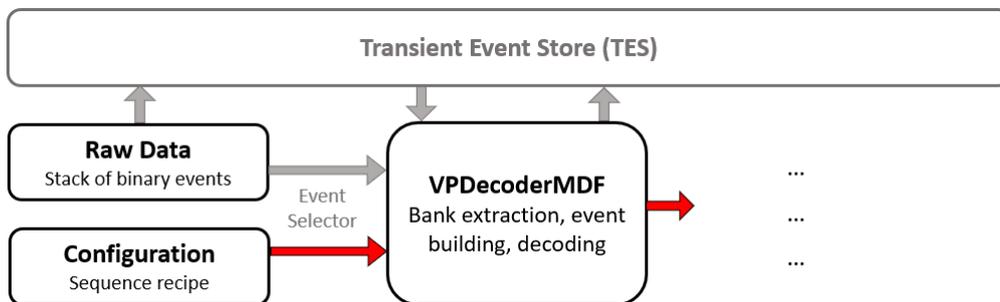


Figure 4.4: A sequence template for any analysis requiring MDF decoding.

The FRG format is a raw data format of GWT frames, where four SPs are written in a 128-byte word. This format of GWT data (see Section 3.3, Fig. 3.25) is used for special analysis of pixel sensor performance and firmware testing [374]. To each 128 bytes of SP,

another 128-byte word is added with information about the hit counters (hence the format is also called GWT Counters). The structure is somewhat similar to the VELO bank in Gaudi raw data format, but neither ICF, nor padding and SourceID are saved into the data banks. The specification of GWT counters is documented thoroughly in [375]. A sequence element in Vetra that decodes GWT counters data is *VPDecoderFRG*.

Because the Vetra internal decoders are a vital part of the LHCb ecosystem (note that ODINs, MDFs and *RawEvent* are common entities in software for the entire LHCb spectrometer), the decoders can be used externally to the platform. A sequence builder is accessible from the options files, with no need to change the configuration settings. *VPDecoderMDF* uses *VPChannelID* (the entity to represent the pixel readout channel, originated from the LHCb project [376]) to keep the channel representation compatible with other software projects distributed in LHCb. A block structure of Vetra sequencer makes it possible to import a specific algorithm outside Vetra and to link it in a foreign environment. Moreover, Vetra can be linked by any Gaudi project when asked the compiler to do so. The configuration settings define the internal architecture of the processing mechanism (e.g. manipulation of *EventSelector*), which guides the Gaudi application manager and, if needed, can be adjusted to the specific usage. The template sequence are hard-coded in the configuration file and can be called from the options file. Most of the sequences anticipated to be used frequently were predefined by the Author, such that the mechanism of running the software remains relatively simple to the average user.

4.3 The VeloPix calibration

The most critical task standing ahead of the high-level software is the detector calibration. The reason of the VeloPix calibration is that properties of the electronics readout channels are not entirely uniform. The pedestal, which is the centre of the noise fluctuations, varies for every channel. The height of the pedestal is expressed in Threshold DAC units, a front-end voltage DAC. In brand-new ASICs, it usually varies from 1000 to 1300 DAC. The pedestal spread and the centre value get higher for irradiated ASICs, in general. The preamplifier is sampled by the discriminator, which compares the sample to the signal threshold, a global setting of the ASIC. The effective signal threshold varies in each channel and is proportional to the range between the pedestal and the global signal threshold. In order to equalize the effective thresholds across the ASIC, the equalization procedure has to be applied. The procedure aims to adjust Trim DAC in each channel individually, such

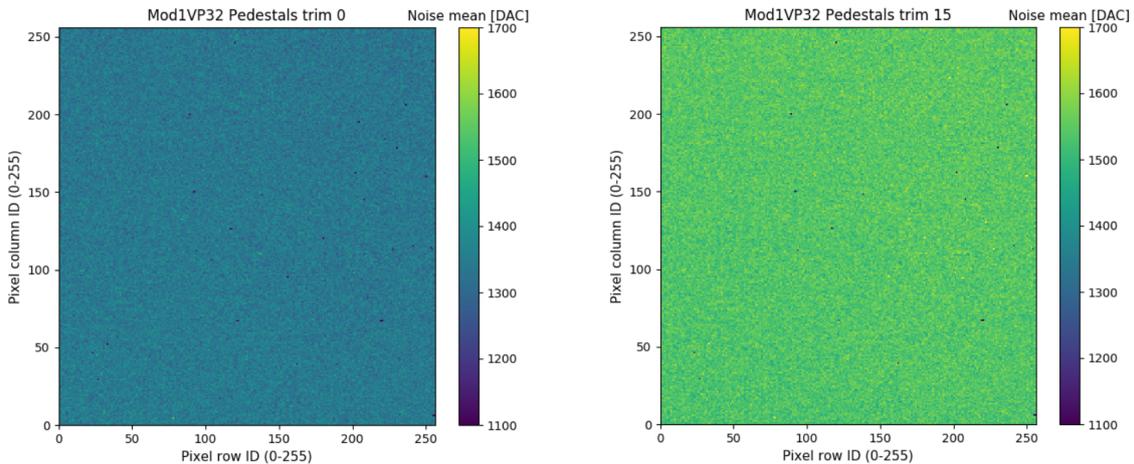


Figure 4.5: The pedestal map for minimal (left) and maximal (right) Trim DAC.

that the pedestal non-uniformity is compensated at the discriminator level. The procedure consists of a few steps. First, it estimates the pedestal values by shifting the global signal threshold and probing the noise. When the discriminator accepts the event, the global threshold is in the range of the noise fluctuations around the pedestal. Because the noise fluctuation is approximately normally distributed, the pedestal for each pixel is taken as the mean of that distribution. Next, the procedure applies the same threshold scan but with the trims set to their maximal values. The second scan is necessary as one can not predict the exact scale factor between the Trim DAC and the Threshold DAC in the channel. On average, 1 Trim DAC corresponds to 30 Threshold DAC, but this value varies significantly. After the scans are taken, the value in the middle of both distributions is chosen as the equalization aim for the pedestals, and Trim DACs are computed for the pedestals to match the aimed value. A deviation of the equalized distribution is correlated with a resolution of Trim DAC, which corresponds, on average, to the amplitude of a signal arising when the charge of 375 electrons is injected into the channel. When pixels are equalized, a global signal threshold is found in such a way that it matches the amplitude of the signal arising from 1000 electrons. The whole procedure can be visualized in the form of three pedestal distributions; first from the initial threshold scan, second from the threshold scan with maximal trims applied, and third from the estimated equalized pedestals. The exemplary pedestal maps taken at the slice test (Mod1VP32 ASIC) are plotted in Fig. 4.5, and the scan plot produced in Vetra with the equalized pedestals and Threshold DACs converted to voltage in Fig. 4.6.

A threshold scan is taken with a step of 5 Threshold DAC, which is enough to

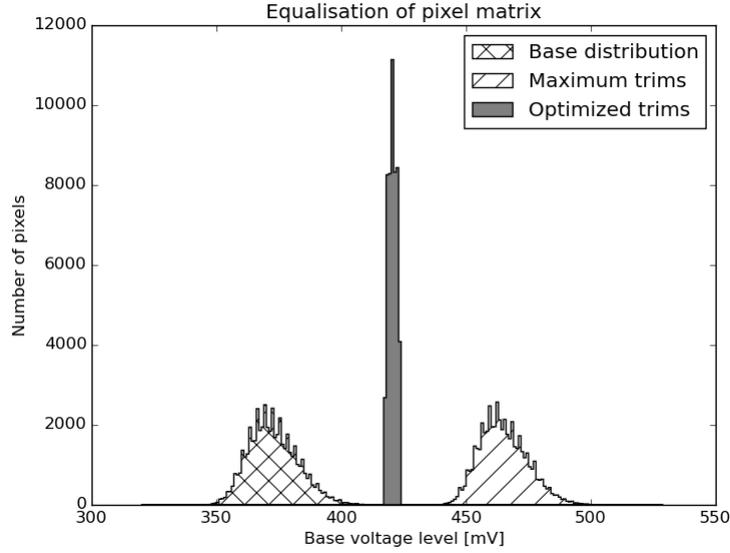


Figure 4.6: An outlook of pedestal equalization in Vetra for all pixels in a single ASIC.

reconstruct the standard deviation of the noise fluctuations. The major contributors to the noise are the shot noise and the thermal noise, and the latter is a non-linear function of temperature [377]. Because the noise rate is proportional to the irradiation, it is one of the metrics to assess the condition of the readout and to roughly estimate the leakage current in the analog front-end. When a noise rate is high, a channel is masked in the masking gate (a default condition >9 Threshold DAC). Furthermore, a channel is masked if the equalization procedure fails or the equalized pedestal does not fall in a given margin around the aimed value (15 Threshold DAC, half of the average trim resolution). Finally, a mask is applied if the Trim DAC to Threshold DAC scale factor is unusually high or low (>40 or <20 Threshold DAC). Those masking categories are A, B, C and D, respectively.

The threshold scan in Vetra is processed by *VPThresholdScanECS* sequence for DIM, CSV and FRG raw data formats, and *VPThresholdScanGWT* for MDF data format. *VPThresholdScanECS* instance switches the default *EventSelector* event service off and handles the data in the form of arrays of unsigned 8-bit type. The sequence initializes the decoder, executes the scan analysis and saves the output to the files of CSV-type. The CSV files feature a readable version of the header. The data are optionally saved in ROOT format using AIDA (Abstract Interfaces for Data Analysis) interface [378] and the histogram persistency in Gaudi. *VPThresholdScanGWT* uses *EventSelector* to build and inspect the events. The ODIN fragments are read out before the actual analysis takes place. When the bank iteration loop is over, Vetra analyzes the data and produces noise rate

and pedestal maps. The format of those outputs is consistent with *VPThresholdScanECS* sequence. When only a certain area of the ASIC was scanned, Vetra automatically adjusts the analysis target. Multiple threshold scans for multiple ASICs can be processed at the same time. In particular, it is possible to run a single instance of Vetra for the equalization procedure of the entire detector, i.e. 52 modules. However, for the time-efficiency, it is recommended to split the modules in groups and run multiple instances of the Vetra application. A fragment of an exemplary Vetra log file for the equalization procedure is shown in Fig. 4.7. The objects *VPASICTrimScan* from the log are the threshold scan data banks stored in TES.

```

VPEmu::VPEquali... INFO Checking for two files in VPASICTrimScan bank...
VPEmu::VPEquali... INFO Setup details:
VPEmu::VPEquali... INFO Trim 0 mean distribution: 1341 +/- 31
VPEmu::VPEquali... INFO Trim F mean distribution: 1539 +/- 29
VPEmu::VPEquali... INFO Equalisation Target: 1440
VPEmu::VPEquali... INFO Achieved: 1440 +/- 3.6
VPEmu::VPEquali... INFO Masked Pixels: 95
EventLoopMgr      SUCCESS Terminating event processing loop due to scheduled stop
ApplicationMgr    INFO Application Manager Stopped successfully
EventLoopMgr      INFO Histograms converted successfully according to request.
ToolSvc           INFO Removing all tools created by ToolSvc
ApplicationMgr    INFO Application Manager Finalized successfully
ApplicationMgr    INFO Application Manager Terminated successfully with a

```

Figure 4.7: The ending fragment of the log file of the equalization procedure performed in Vetra.

Because two threshold scans are needed for the equalization (with minimal and maximal trims), the equalization sequence accepts the input data if two scans are provided. The sequence for equalization is either *VPEqualizationECS* or *VPEqualizationGWT*, depending on the given type of the raw data format. The equalization algorithm takes the analyzed threshold scans and computes the optimal trimming, the masks and further estimates the pedestals. The equalization sequence scheme in Vetra is depicted in Fig. 4.8.

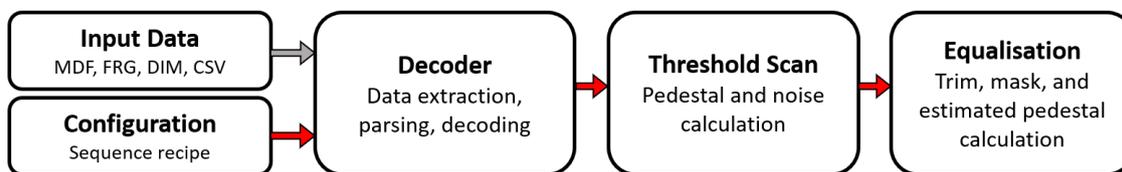


Figure 4.8: A sequence structure of the equalization procedure in Vetra.

The actual calibration process in the detector is multi-folded. A tool for direct communication with the detector is WinCC OA, which handles the DAQ system, the hybrids and the OPBs, and maintains the whole readout of the electronics modules through Slow Control. A command to scan the thresholds in the ASIC is sent via the WinCC equalization

panel. It allows for configuring the scan parameters, such as the Threshold DAC step and range, and also for selecting the sensors for the scan. The panel is presented in Fig. 4.9 and was redesigned multiple times by the detector group and with the assistance of the Author. The WinCC navigation panel in a development version, for the miniDAQ [379] and the front-end configuration is shown in Fig. 4.10. The standard calibration is taken using the ECS readout. Once the scan is completed, WinCC constructs a header and asks the DIM writer to build the banks. WinCC runs on a dedicated DAQ machine which shares a disk space with the experiment online network. When the files are saved, their paths and the metadata are pinned to WinCC datapoints (the WinCC-specific object types), so they can be further used in the operational panels.

VeloPix ECS Equalisation

Settings

Scan Range

	0-Trim	F-Trim	Control
Minimum	1100	1350	1300
Maximum	1550	1800	1500
Step Size	5	5	5

Shutter

Clock Cycles: 800
Multiplier: 0
Shutter Time: 20 us

BXID

Off
 On
 1DEB

Number of masks: 1 (must be a perfect square)

Save File: Save Config

Actions for Complete Module

Configure New Settings | Default Actions | Run | Run Orig | Create Plots | Remove All Files

Status:

Storck

User ID: Default
ff6effaea839b2058f199f3

Workspace: 41bd-9e3a-699ebf1ce72a

Server: 137.138.31.83:8000

Status:

Push to Storck

Figure 4.9: The developing WinCC equalization panel.

The datapoints are integrated with the DIM server (see Section 4.1), which allows them to transfer the content within the online network (the LHCb control room network) and optionally to the outside networks, but this solution is temporarily not used in the system. A DIM server (written in Python in a whole [380]) operates on the online network and listens to the DNS (Domain Name System) port of the machine which runs the WinCC project, receiving the datapoints content every time it is changed in WinCC.

When the server receives the raw data locations (paths), it finds the files on a disk and calls STORCK client (database client) to upload the files. The uploading is managed by the bidirectional RESTful (Representational State Transfer) request. A mechanism of the connection and database infrastructure is explained in [381] and in PhD dissertation of Majewski [382].

When STORCK receives the raw calibration data, it flags the data instances as

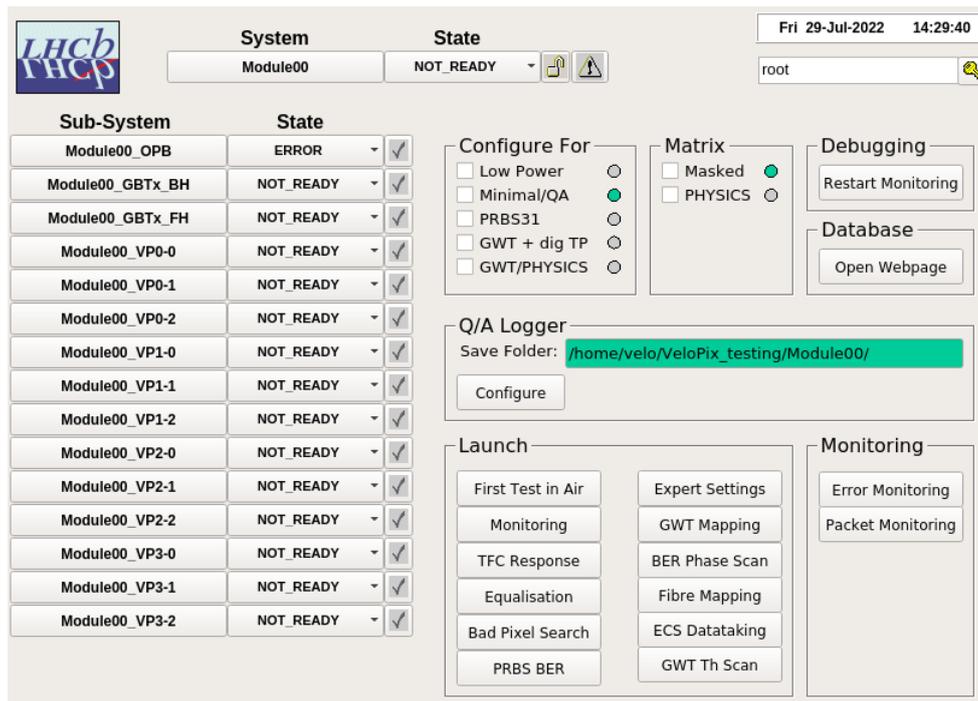


Figure 4.10: The developing WinCC navigation panel.

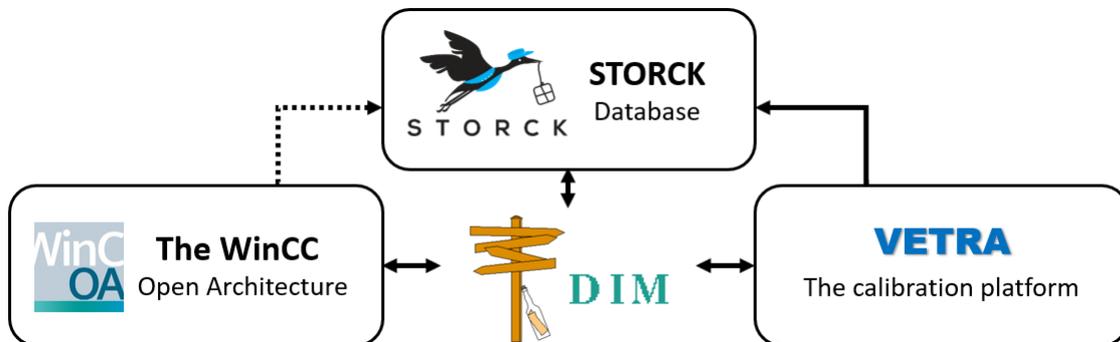


Figure 4.11: The data flow in the calibration software. The dotted line marks the connection available in the development version only.

unprocessed and awaits a call to download. A download request is made either for the unprocessed data or for the data taken at a given time (time of the beginning of the calibration procedure acts as the unique key for each calibration dataset). The request is issued by the same DIM server, which, after downloading is finished, runs Vetra on the online cluster (CPU cluster of the online network). Optionally the request can be sent from Vetra if the pipeline changes in the future. In a development version, a direct upload request to STORCK and a wrapped system command to run Vetra are available in the equalization panel. The disadvantage of this solution is that it could easily overload the

limited CPU resources of the DAQ machines.

After Vetra finishes the analysis, the output data (e.g. the calibration constants) are sent to STORCK and associated with the original raw data as its child data objects. The request can be made either from the DIM server or from a sort of master manager, an entity that can communicate directly with the STORCK client. The data flow between the entities is illustrated in Fig. 4.11, noting that the final decision on the pipeline structure will be made before the software commissioning in the second half of 2022. When STORCK receives the processed data from Vetra, it removes the unprocessed flag from the raw files. To apply the calibration, a recipe creator requests the calibration data of a specific timestamp and further creates a recipe binary file for the detector. The recipe contains the datapoints instruction to recreate the maps of Trim DACs and masks for each ASIC. Eventually, the recipe creator sends the command to the WinCC OA project through DIM and asks the WinCC instance to apply the recipe.

4.4 Monitoring

The calibration monitoring in Vetra includes plotting, pixel quality overview, data validity checklist and a developing version of the trending tools. Monitoring scripts are mostly Python-based and should be considered external to Gaudi and VELO-specific. Vetra supports the plotting of pedestals, Trim DACs, masks and noise rates, which come in the form of 256x256 pixel maps. The exemplary plots of noise rate for two ASICs, one brand-new and one damaged, are shown in Fig. 4.12. The noise rate is considered normal in the pixels marked as blue. Pixels appearing with no noise encountered issues in the scanning process. For instance, a failure in the readout of two neighboring columns in the damaged ASIC is clearly visible. The Trim DAC distribution for the same ASICs is shown in Fig. 4.13. Each pixel with no noise rate measured is granted a Trim DAC value of 0.

It is possible to save the reconstructed hit distribution for a pixel or a group of pixels when analyzing raw data from a threshold scan. This functionality is by default switched off in calibration sequences, as the threshold scan only measures the normally distributed noise, which is fully described by a mean and a standard deviation. However, when a scan involves the signal, such plots are more valuable and can give an insight into the processed signal, as outlined in Section 5.2.

The versatility of the algorithms in Vetra allows for building user-defined sequences, in particular occasional scans, performed once during the detector commissioning, also called

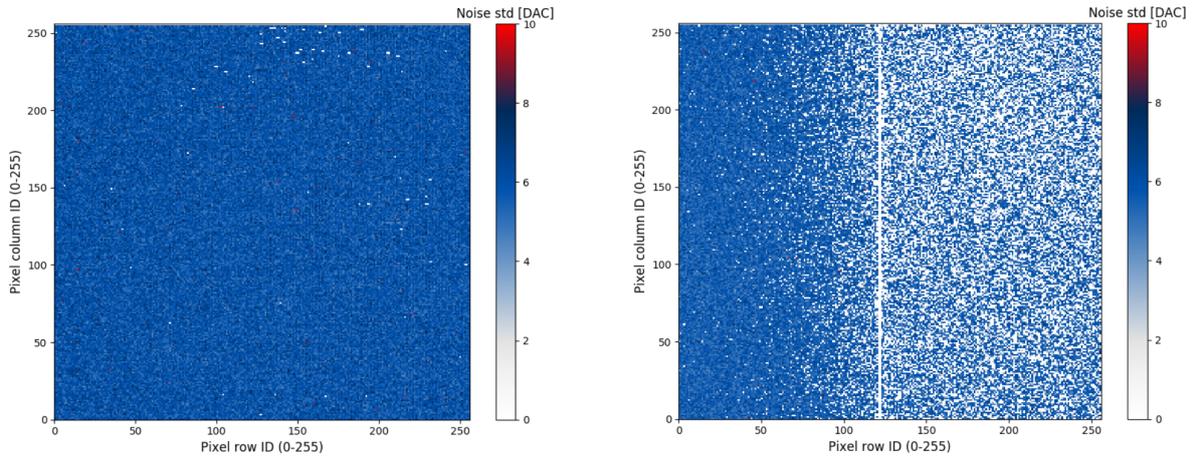


Figure 4.12: Noise rate for a brand-new (left) and a damaged ASIC (right).

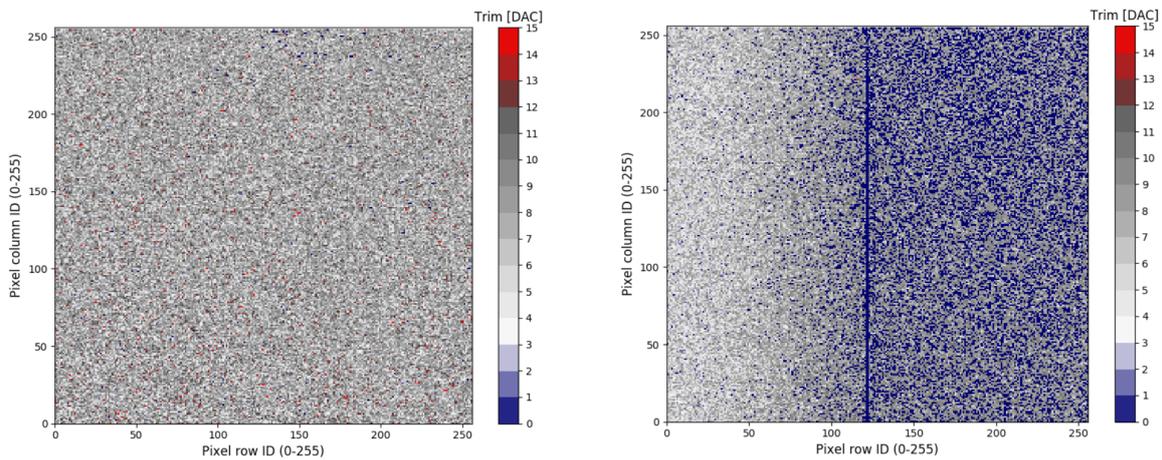


Figure 4.13: Trim DACs equalizing a brand-new (left) and a damaged ASIC (right).

special runs. An exemplary special run is a temperature scan of the calibrated detector (this scan is performed for selected modules once per year). It was studied whether the temperature of the sensors affects the column readouts in the ASIC, and several successive calibrations were taken with various intensities of the CO₂ cooling. The scan showed some irregularities in the noise rate distribution as a function of the column number, as shown in Fig. 4.14. Another interesting effect discovered in this analysis was that the noise rate gets higher when the temperature of the sensor drops down and no trims are applied. Fig. 4.14 visualizes this effect for two ASICs, Mod1VP02 and Mod1VP32, from the tested module in the slice test setup. For comparison, the noise rate maps for minimal and maximal Trim DAC values are shown in Fig. 4.15 for Mod1VP32. The higher values of the noise width are clearly visible in the map corresponding to Trim DAC of 0. This observation changed

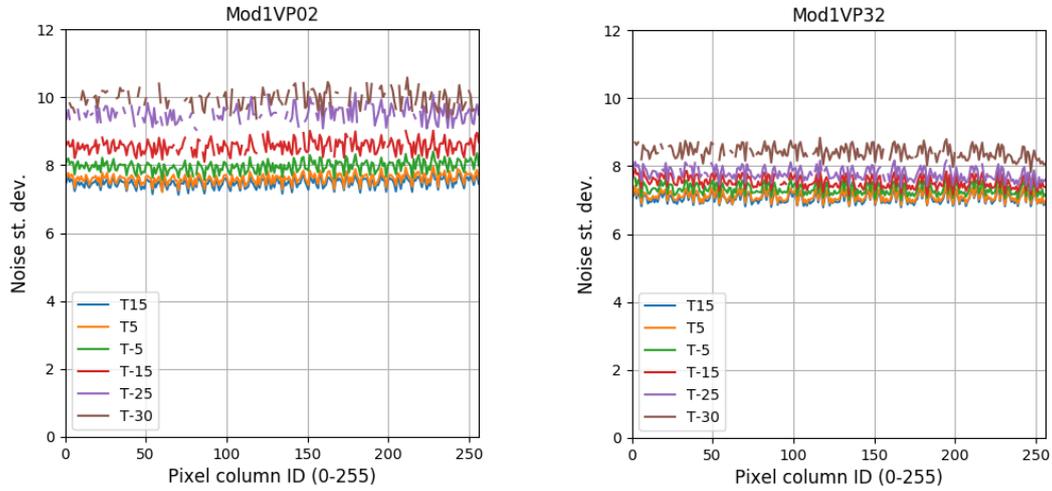


Figure 4.14: The average noise rate as a function of column number for various temperatures, for Mod1VP02 (left) and Mod1VP32 (right) in the slice test.

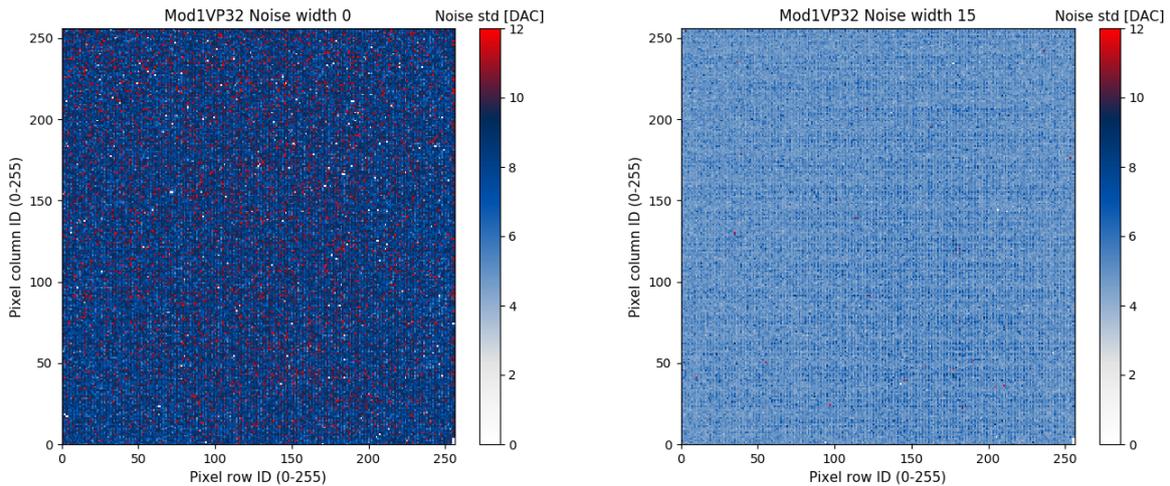


Figure 4.15: Noise rate measured for the Trim DAC of 0 (left) and 15 (right), for Mod1VP32 in the slice test.

a view of how noise measurement algorithms should work, and the conclusion was to not measure the noise when no trimming is used. Complete analysis scripts and monitoring for this scan were added to Vetra and can be used to trend any possible changes as a function of sensor's temperature and particle fluence.

A trend monitor was developed to analyze the calibration data taken on different dates. The trending tool can benefit the decision-making on whether the calibration has to be repeated and a new recipe file created. A noise rate is a function of the leakage current, and the outlook of its history is the easiest way to monitor the changes in sensors and

front-end electronics. The trending monitor must trend each pixel individually, which is reasoned by highly nonuniform fluence.

Vetra can host monitoring components for other parts of the detector system, e.g. clustering or closing procedure. Monitoring tasks in WinCC panels are troublesome to tackle due to the CPU curb of the DAQ machines and should only include the monitoring of the electrical settings and temperature controls. While Vetra is responsible for the raw data monitoring, the monitoring of physics data and the vertex reconstruction is handled by Monet [383].

4.5 The signal scans

Threshold scans in the calibration procedure measure a mere noise to find channel pedestals. For the study of signal processing, a threshold scan can be used to measure the amplitude of Test Pulses (TP). In this way, one can mimic the signal processing throughout the readout chain⁵. TP is configured by two DAC settings *VTP_COARSE* and *VTP_FINE*, which are the DACs to setup the voltage at the input capacitor (see Fig. 3.24). The capacitor injects a charge to the channel that is a product of the voltage step and the 4 nF capacitance. The relationship between the input charge and the signal amplitude is driven by the gain of the preamplifier and is studied further in Section 5.2, as an exemplary use-case for the software implemented in Vetra.

To analyze a threshold scan taken with TPs, Vetra uses a similar sequence to the equalization procedure, where *VPEqualization* is replaced by *VPTestPulseScanECS*, or *VPTestPulseScanGWT* sequence, depending on the raw data type. One should note that a 6-bit processor of the ECS readout limits the maximal number of TPs seen in a single shutter period to 63. To distinguish the noise region from the signal, it is recommended to send less number of TPs, as the noise saturates the readout even in a very short shutter period. There are no such limitations when using the GWT readout.

Studying the signal processing highly benefits from the ToT information available in the ECS readout mode. The readout provides ToT for the last TP seen in the pixel with a 6-bit resolution. When more TPs are sent while the shutter is open, only the last one is saved and sent off the chip. The readout arranges the ToT data in the same format as the counting mode. There are multiple applications which take advantage of ToT readout, e.g. CCE (Charge Collection Efficiency) studies [384, 385], a pulse shape reconstruction

⁵The TPs are injected via a specialized module of the readout chip, so it cannot properly emulate the response of the silicon sensor. However, it is enough to perform many vital tests of the chip.

and the insight of the energy spectrum. Some of the applications are broadly discussed in Section 5.1, e.g. studies from 2018 SPS testbeam.

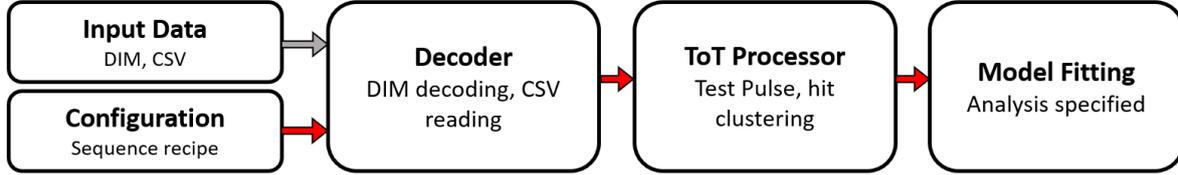


Figure 4.16: A block structure of ToT processor sequence in Vetra.

It is possible to analyze a threshold scan taken with the ToT mode. The processing sequence to handle this specific readout in Vetra is *VPThresholdScanToT*, which replaces the standard *VPThresholdScan*. The sequence configuration can be adjusted to take into account the scan setup, that is either a constant amplitude of TP or a spectrum if the scan was taken using a radioactive source. The energy spectrum reconstruction is not trivial and is rigorously discussed in Section 5.1. Vetra manages the ToT sequences and applies sequence elements if further analysis is required, such as hit clustering. A default block structure for a ToT sequence is presented in Fig. 4.16.

4.6 Data emulation

Vetra offers a wide range of sequences to emulate different functionalities of the readout electronics. The reason for adding the emulation segment to Vetra was that testing the data system on the emulated readout is sometimes essential for software and firmware development. For instance, the segment was used to emulate the GWT stream from the entire detector when only a single developing module was available for laboratory purposes. The stream was used to probe the database bottlenecks when drafting the system pipeline in the online network. The emulation also supported the development of the calibration algorithms in Vetra.

It is possible to emulate both ECS and GWT readouts, although the latter was equipped with much more advanced options. It can simulate the pixel readout mechanism and the SP packet builder. It can further imitate the column readout and creation of the data banks, including the ODIN fragments. The emulator is flexible enough to simulate complex readout effects, such as discarding the SP packets when exceeding the fibre’s bandwidth. A threshold scan is emulated on top of the emulator engine, which cooperates with the ODIN encoding manager, *VPEncoder*. Produced files are of MDF type. The emulator engine is

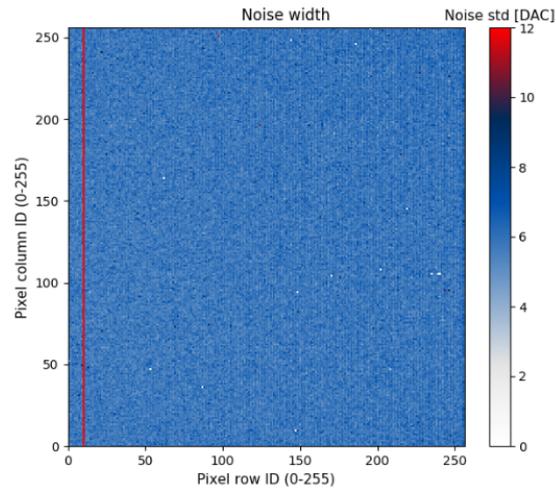


Figure 4.17: Simulated noise rate from the GWT readout.

used by algorithms that can imitate any scan of the ASIC. In particular, this scan can be a threshold scan, and therefore, the equalization scan, so the emulation of the calibration is possible. A threshold scan and equalization are managed by *VPThresholdScanEmulator* and produce the calibration data consistent with the ones which come from the actual firmware. A block structure of Vetra allows constructing a sequence composed of emulation blocks and analysis blocks, such that the emulated data are analyzed yet in the same *EventSelector* loop. An exemplary noise rate measured in a simulated threshold scan is shown in Fig. 4.17. The possibility of task merging is helpful when designing the software infrastructure but was mainly intended for the convenience of the user.

Chapter 5

Pixel detector studies

This Chapter presents the selected analyses that were possible to perform with the software described in Chapter 4. Although the most substantial outcome of the software platform is the detector calibration, a full range of applications is available in Vetra, and some of them are explained in this Chapter. These include clustering described in Section 5.1, pixel gain study in Section 5.2, and chip analog optimization in Section 5.3. All discussed analyses were proposed and implemented by the Author of the thesis.

5.1 Time-over-Threshold analysis

This Section presents the analysis regarding the charge collection mechanism in silicon pixel prototype sensors for Upgrade I Vertex Locator. The theory behind such detectors was broadly documented in multiple sources, e.g. [318,386] and on the direct example of MediPix [387]. When a charged particle travels through a sensor, the generated charge sensed by the pixel analog readout derives from Shockley-Ramo theorem [388] and distributes according to the Gauss-Landau (GL) model, a convolution of the detector response function and the Landau curve. The latter describes the energy deposition in semiconductor by incident particles, provided the thickness of the bulk is small enough [389], which boils down to the fact that the probability of particle absorption is negligible. The Landau distribution does not have an exact analytical formula but an indefinite integral and has to be computed using numerical methods. As the detector response function is usually assumed to be a Gaussian, to express the Gauss distribution convoluted with Landau, an indefinite integral can be written (5.1):

$$GL(x) = (G \otimes L)(x) = \int_{-\infty}^{\infty} G(x')L(x - x')dx' \quad (5.1)$$

where $G(x)$ and $L(x)$ are (5.2)(5.3) (c - scale parameter) respectively:

$$G(x) = A_1 \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (5.2)$$

$$L(x) = A_2 \int_0^{\infty} \exp(-t) \cos\left(\frac{tx}{c} + 2\frac{t}{\pi} \ln\left(\frac{t}{c}\right)\right) dt \quad (5.3)$$

The Landau-Gauss convolution is then defined by (5.4):

$$(G \otimes L)(x) = A \int_{-\infty}^{\infty} \int_0^{\infty} \exp(-t) \cos\left(\frac{t(x - x')}{c} + 2\frac{t}{\pi} \ln\left(\frac{t}{c}\right)\right) \exp\left(-\frac{(x' - \mu)^2}{2\sigma^2}\right) dt dx' \quad (5.4)$$

The multivariate integral in (5.4) hinders the way of getting the real value of the expression and makes it inconvenient in terms of operational complexity. A high-resolution map of the inner integral (5.4) in function of c and x was modelled and accommodated in a special math library in Vetra. When the GL model is used, the engine manager loads the library and takes advantage of the GL mapping, computing only the outer integral in (5.4) using numerical methods, which shortens the execution time by about a factor of 10.

Because a gain of the preamplifier in relation to injected charge is linear, the ToT of hits approaches a GL distribution in a sensor exposed to a particle beam of fixed energy. Observing a pure GL is hardly possible due to readout imperfections. The ToT processor in the chip's front-end lacks the event buffer and pipeline of hit latency and is restrained by a 6-bit LFSR resolution. The resolution saturates the processor when ToT of the signal is higher than 63 counts. When a few hits appear in sample intervals in a single shutter period, only the last hit is saved. Another issue is associated with a discrete mechanism of detection. If an incident particle travels through the sensor, no guarantee is given that the charge the particle induces is visible at one pixel only. When a hit occurs near the border region, it can induce enough charge for the adjacent pixel to light up. If near the corner, it can possibly activate all adjacent pixels. To find the total deposited charge, activated pixels have to be associated into a cluster, and ToT collected and added up. A hit usually creates a 1-pixel or 2-pixel cluster. A cluster of 3-pixel or 4-pixel size is likely to come from the corner hits. Clusters bigger than those can appear either when the deposited energy was high or when smaller clusters are coupled together. Fig. 5.1 visualizes the track intercept (intra-pixel position) within a pixel on the example of TimePix chip, differentiating the

size of the clusters. If the incident particle is a muon, it can travel in a not straight way and leave a track of multiple pixels depositing its total energy [390]. When a charge of less than $1000 e^-$ is spilt over to the adjacent pixel, the signal fails to cross the signal threshold and the signal is effectively suppressed. This loss of charge is troublesome to model and is reflected on ToT distribution, usually recognized as a deficiency of events in the Landau tail.

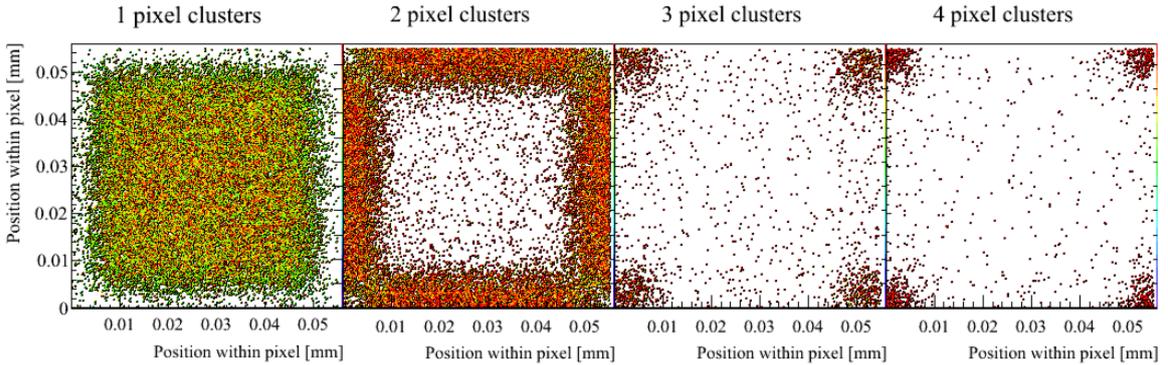


Figure 5.1: Intercept (intra-pixel position) of track positions within a pixel illustrating a hit distribution for different cluster sizes for TimePix chip. [321]

Because no ToA can be measured by the ECS ToT readout, there is no way to pair the clusters with the BX IDs within a single shutter period. In particular, if a hit conjoined a pixel with its neighbor in a time interval, there is no prospect to tell whether the cluster comes from two different events or is actually one event spread over. Furthermore, a new hit can reinstate a ToT in one of the pixels of an already existing cluster. In general, such overridden clusters make an appearance in a function of hit occupancy. In turn, the occupancy increases as a function of the shutter time and beam luminosity. It also depends on the height of the signal threshold, which is shown in Fig. 5.2, where a relatively low (1500 DAC) next to a very high (2500 DAC) signal threshold is used. An exemplary view of the hits from a single shutter is shown in Fig. 5.3. To lower the occupancy and increase the cluster finding efficiency, one could lower the shutter time, but the disadvantage of this is that opening and closing the shutter generates a noise which dominates the signal in the region of shutter boundaries and effectively lowers the signal-to-noise ratio if the shutter period is shortened.

Note that clustering of ToT events is not the same clustering that is done in trigger software, Moore, for the track reconstruction from the default VELO GWT readout. The ToT features a simple readout mechanism at the cost of deficiencies described above. It

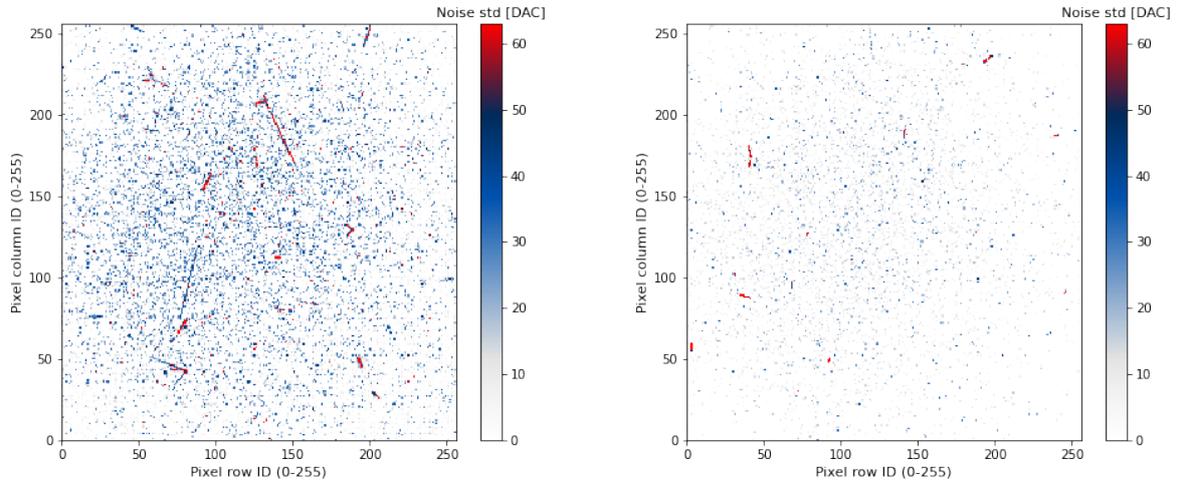


Figure 5.2: Time-over-Threshold frame for the signal threshold at 1500 DAC (left) and 2500 DAC (right).

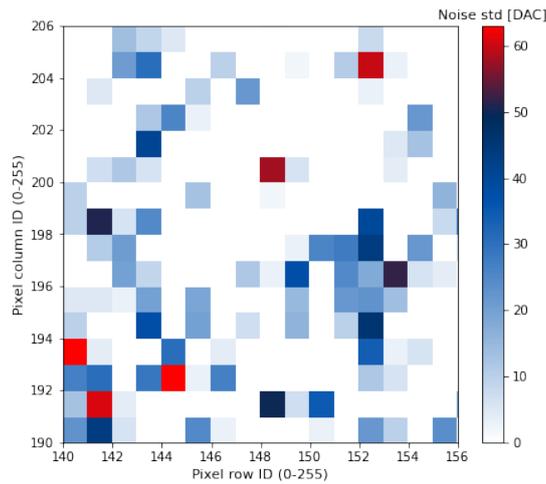


Figure 5.3: A local central area of VeloPix matrix taken from Fig. 5.2.

is mainly used for chip development, calibration studies, radiation damage studies and testbeam purposes. The most generic approach for clustering is to outline the activated pixels that are adjacent to each other, either orthogonally and diagonally or orthogonally only. Visualization of the concept is shown in Fig. 5.4. One challenge is to design the outlining process in a CPU-efficient way. The brute-force algorithm requires a lot of calculations, especially if they are based on recursion, as every time a pixel is added to the cluster, its neighbors have to be checked as well. Moreover, the brute-force algorithm fails when the hit occupancy is high and the crowded objects form large, non-separable structures. A solution to address this problem in VeloPix chips proposed by Akiba and

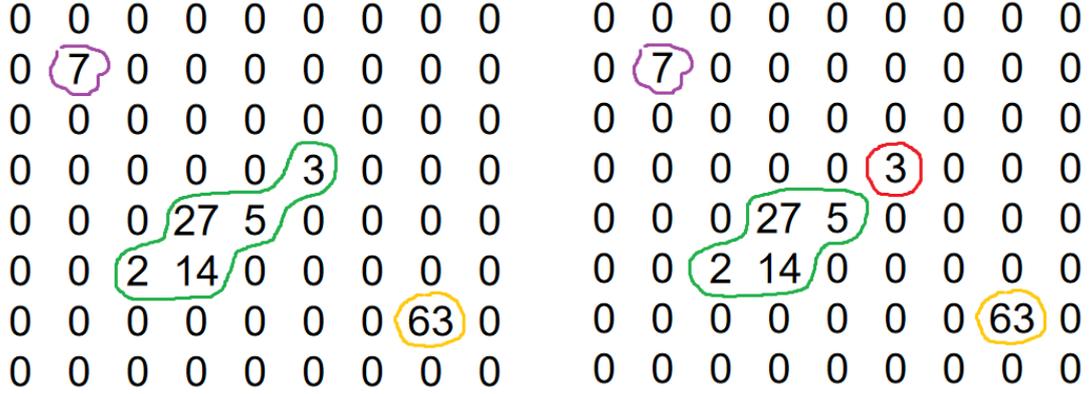


Figure 5.4: Two different approaches for the clusterization algorithm.

myself was to consider to apply ML for organized clustering. Clustering problem has a long history in the field [391] with a range of approaches with state-of-the-art given in [392–394]. When the clusters are overlapped, a deblending method has to be used [395, 396], which builds the objects and estimates their amplitude. An algorithm used for clustering the celestial maps (an example in Fig. 5.5 and [397]) was tested at 2018 SPS testbeam, but could not recreate the GL distribution after reblending. Because no translation function between the obtained distribution of ToT and the actual GL distribution was known, and no large enough data sample was taken to demand highly efficient cluster computing, the idea was abandoned. However, a conclusion was that testbeam experiments could take advantage of that method when the ASIC gets higher granularity in the future.

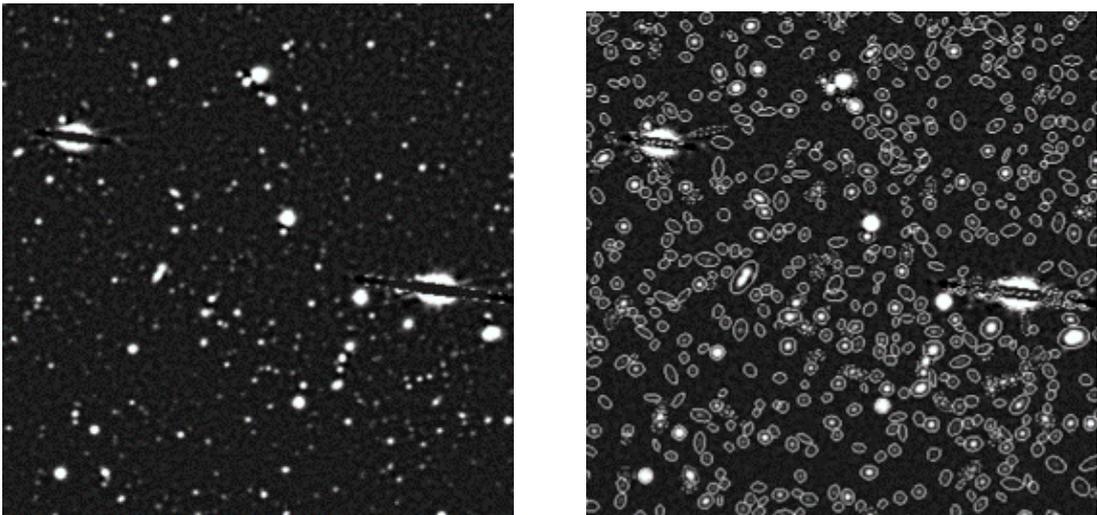


Figure 5.5: Clustering the objects on the astronomical map. [397]

The chosen way of clustering determines further analysis. On the one side, the diagonally

built clusters increase the odds that two clusters would be incorporated into one outline. On the other side, the orthogonal criterion of neighborhood can get the cluster split apart. Therefore, the reconstructed distribution of ToT is either underestimated or overestimated. Both effects can be appropriately addressed and compensated in the analysis, however, dealing with overestimated distribution is much simpler when it comes to algorithmic computing. If a big cluster is split apart and classified as a bunch of smaller clusters, the number of fake clusters can dominate the whole distribution and burden the further modelling with a high variance. This, in turn, would lead to higher uncertainties of the physics parameters. On the contrary, when some clusters are inaccurately merged with the diagonal condition, the distribution can be simply modelled by two convoluted GL distributions. Hence, it was decided that the approach of diagonal and orthogonal outlining would be used in the analysis.

As mentioned above, a GL convoluted with itself can represent two clusters that were incorporated into one structure. The expression 5.5 defines the convolution:

$$(GL \otimes GL)(x) = \int_{-\infty}^{\infty} GL(x')GL(x - x')dx' \quad (5.5)$$

where $GL(x)$ is given in (5.4).

This enhancement to the model is naturally needed only for clusters of larger size than 1 pixel. For clusters with a size of 2 pixels, the relationship between a number of correctly reconstructed clusters described by the standard $GL(x)$ and a number of two 1-pixel clusters combined represented by $(GL \otimes GL)(x)$ depends on hit occupancy. A small contribution of three combined 1-pixel clusters can appear in a set of 3-pixel clusters. In this case, a triple convolution can be added to the model 5.6:

$$(GL \otimes GL \otimes GL)(x) = (GL \otimes \int_{-\infty}^{\infty} GL(x')GL(x - x')dx')(x) \quad (5.6)$$

Following this logic, a quadruple GL convolution would represent four merged 1-pixel clusters. Nevertheless, the contribution of such events is usually very low and can be neglected, in particular that the consecutive peaking appears further and further in a receding tail of the primary GL.

A property of the analog front-end for a discharge of preamplifier is a Krummenacher current I_{krum} . I_{krum} is defined by I_{krum} DAC, a voltage DAC multiplexed from the analog periphery and shared between two channels. In ToT readout, the Hit & Shutter Processor samples the signal at the edge of the discriminated clock and measures ToT of

the signal. In fact, the value of ToT is a function of the Ikrum and the input charge. Ikrum defines a slope of the falling edge of the signal pulse, and increased Ikrum shortens the falling time. Because Ikrum can compensate the leakage currents expected in irradiated ASICs, it is beneficial to know the exact relationship between Ikrum and signal total length, which was the reason for implementing the study in Vetra. Higher Ikrum reduces the time for a channel to return to baseline after processing a signal, shortening the time after which the next signal is seen as separate (so-called dead time). A downside of higher Ikrum is that when a signal is of low amplitude, it is possible to have it arise and fall between the consecutive clocks, not being sampled at all. Moreover, Ikrum slightly contributes to the dissipated power in ASIC and, as a result, increases the temperature of the front-end hybrid. So the final selection of the Ikrum is complex and may have a major impact on the detector efficiency and data quality.

A processing sequence developed for the ToT frames analysis in Vetra is *VPToTAnalyser*, which uses *VPClustering* for clustering and *VPModelFitter* for the model fitting. *VPClustering* is a simple outlining script that combines the pixels into composite objects with clustering criteria given in a function argument. *VPModelFitter* is more advanced and employs a metaheuristic for the model fitting. The modelling of multiple convolutions is tough for the usual optimization algorithms as each new convolution adds a computing time and new parameters to the model. In GL, those are the amplitude, the scale parameter, the width, and the mean in (5.4) (note that some of the parameters are shared between GL and its convolutions). The fitting algorithm was actually designed and implemented by the Author of this thesis [6] and relies on local convergence factors in nature-inspired flower pollination. Although there is no way to assure a global minimum of the fit function is found, the convergence proved to be stable and to approach the true distribution very well, even though having to deal with imperfect GL distributions. Given all the material needed for the analysis, Vetra can be used in the testbeam environment, and the exemplary analysis of 2018 SPS testbeam is presented below.

During the testbeam, 100 frames of ToT for a few different Ikrums were collected with the VeloPix ASIC as DUT in TimePix3 Telescope. Each frame is processed by the clustering algorithm in Vetra, which provides a number of clusters for a total distribution. Clusters from successive frames are added and saved into files with respect to the clustering approach and the size of the clusters. Another file is created with fit parameters of the model that was specified in the options file for Vetra. Finally, an image is created with a graph of the fit visualization.

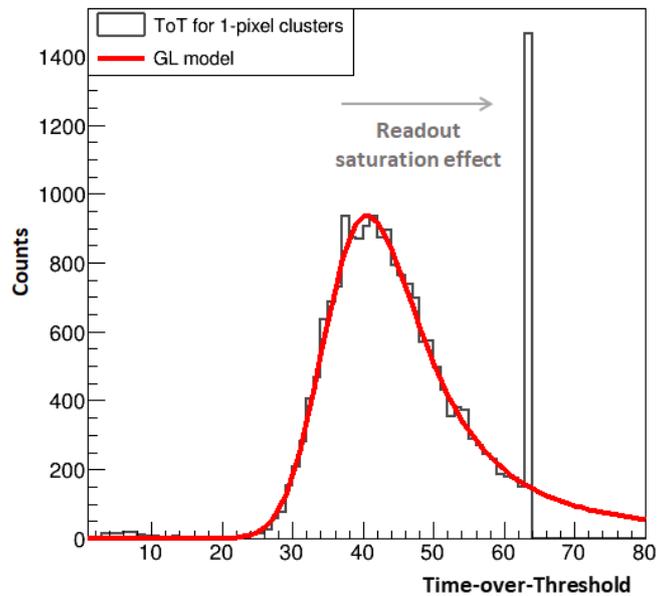


Figure 5.6: A distribution of ToT for 16 Ikrom DAC and clusters of the size of 1 pixel that suffered from 6-bit readout saturation.

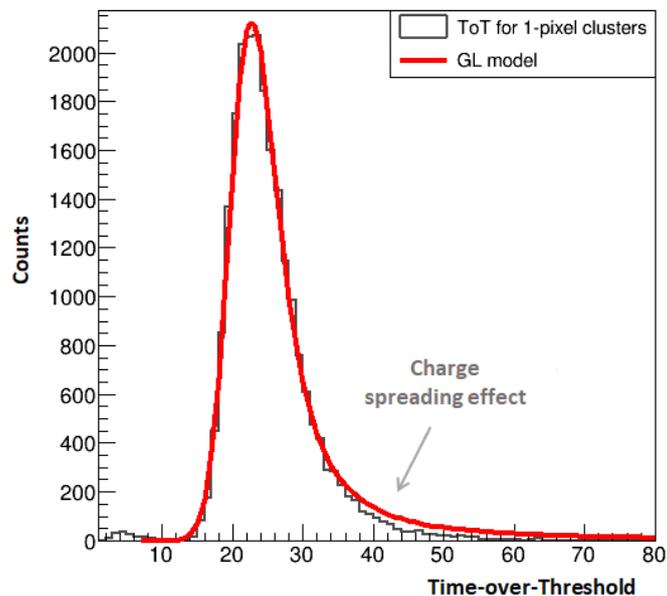


Figure 5.7: A distribution of ToT for 32 Ikrom DAC and clusters of the size of 1 pixel that suffered charge spreading.

To present the readout-specific properties reflected in ToT distribution, the curve for 1-pixel clusters for 16 Ikrom DAC is presented in Fig. 5.6. The peak comes from the saturated events of ToT higher than 63 counts. One way to find a fit is to manually limit the fitting range, although the fit from the graph was actually obtained after evaluating

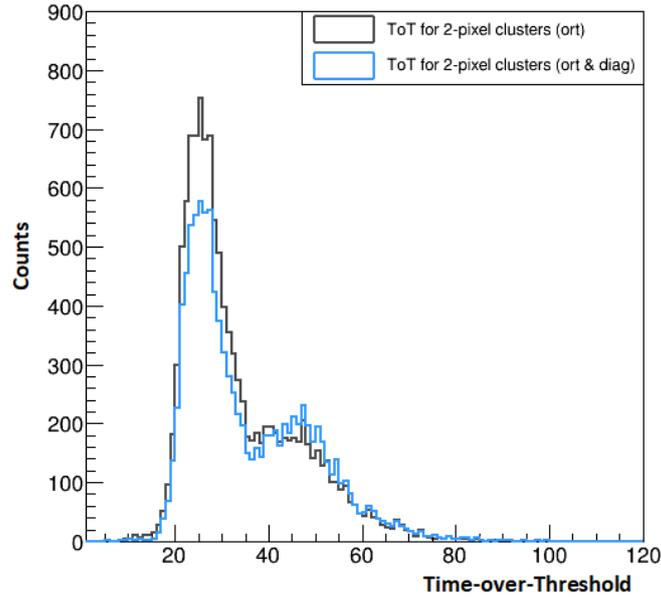


Figure 5.8: A distribution of ToT for 32 Ikum DAC and 2-pixel clusters.

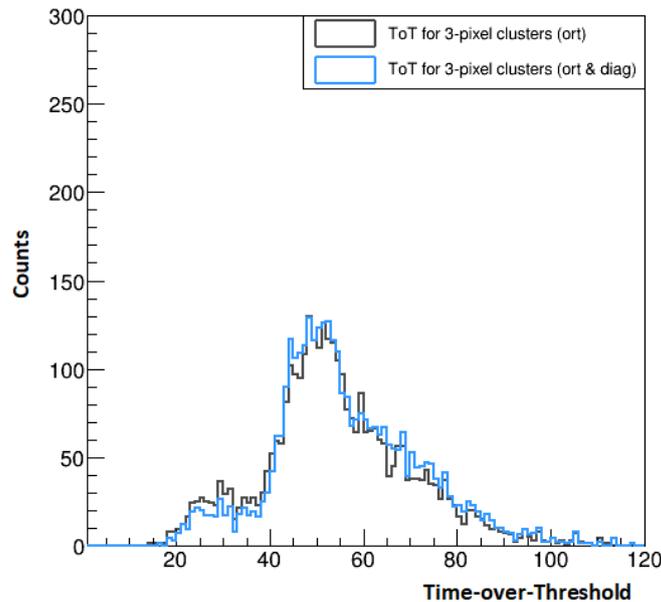


Figure 5.9: A distribution of ToT for 32 Ikum DAC and 3-pixel clusters.

the fit function on every histogram bin. Even though a high MSE from 18% of the points is biasing the fit model, the fit approaches the distribution, which is a solid stress test proving the algorithm robust. A similar distribution but for 32 Ikum DAC is shown in Fig. 5.7. ToT distributed around lower values and the saturation peak disappeared because of the increased Ikum. One way to describe the shift of the distribution is the MPV (Most Probable Value), which is the value of ToT at the peak. The MPV has no representation

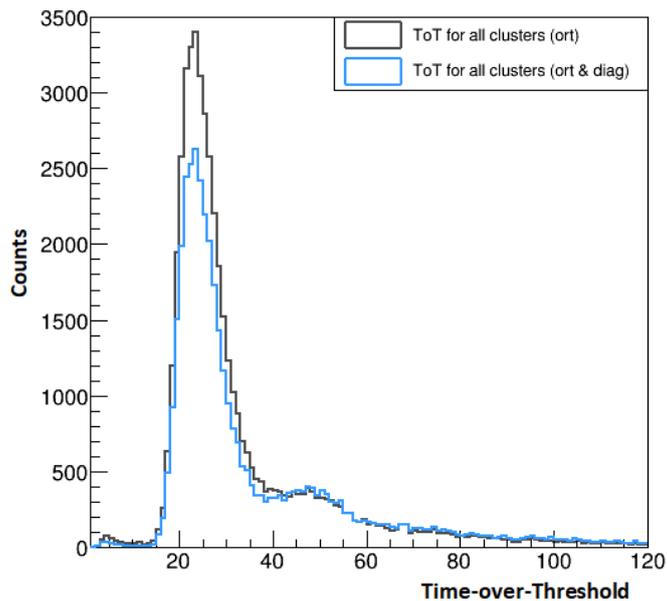


Figure 5.10: A distribution of ToT for 32 Ikum DAC and all clusters.

in the GL formula but can be easily computed¹. The distribution for 1-pixel clusters for 32 Ikum DAC exposes the underestimated Landau tail, which is the artefact after charge sharing, where the hits of high deposited energy are more likely to spread residual charge to adjacent pixels (the spread is not higher than 1000 e^- per neighbor, as it would trigger the adjacent readout and create 2-pixel clusters). Because of the low impact the effect was no further modelled, but after the relationship between the hit interception and residual charge sharing is known, it could be added as a scale factor varying in function of ToT.

A difference in clustering approach can be seen in Fig. 5.8 for 2-pixel clusters, in Fig. 5.9 for 3-pixel clusters, and in Fig. 5.10 for all clusters. In general, the orthogonal-only approach results in higher participation of low ToT events and, consequently, lower participation of high ToT events. As inferred at the beginning of this Section, both distributions should either overestimate or underestimate the deposited energy, depending on the region of ToT. In Fig. 5.8, a visible peak is shaped around doubled MPV of the primary GL, which is a coupled cluster peak of GL convoluted with itself. The selected approach for clustering accepts diagonal connections in favor of simplified overall modelling. A blue-line distribution in Fig. 5.10 can be modelled using a consecutive GL convolutions (5.4), (5.5) and (5.6). The fitted model is presented in Fig. 5.11 for 32 Ikum DAC, and in Fig. 5.12 for 64 Ikum DAC. A bulge around the doubled MPV is recognized as the ToT distribution

¹The MPV can be computed from a derivative of (5.4) or more simply via inspecting the reconstructed fit, which was done in this case.

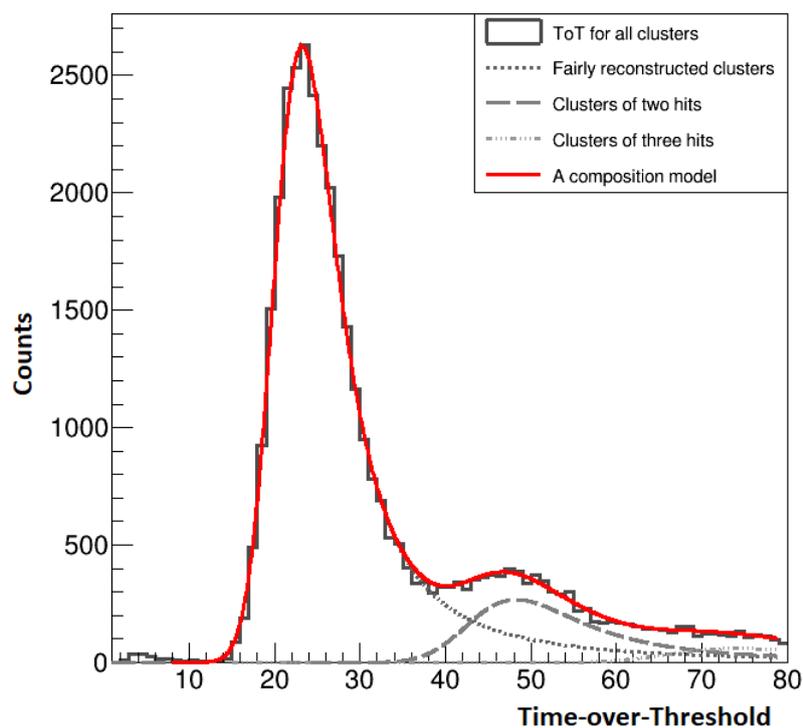


Figure 5.11: A composition model for clusters of any size, for 32 Ikrum DAC.

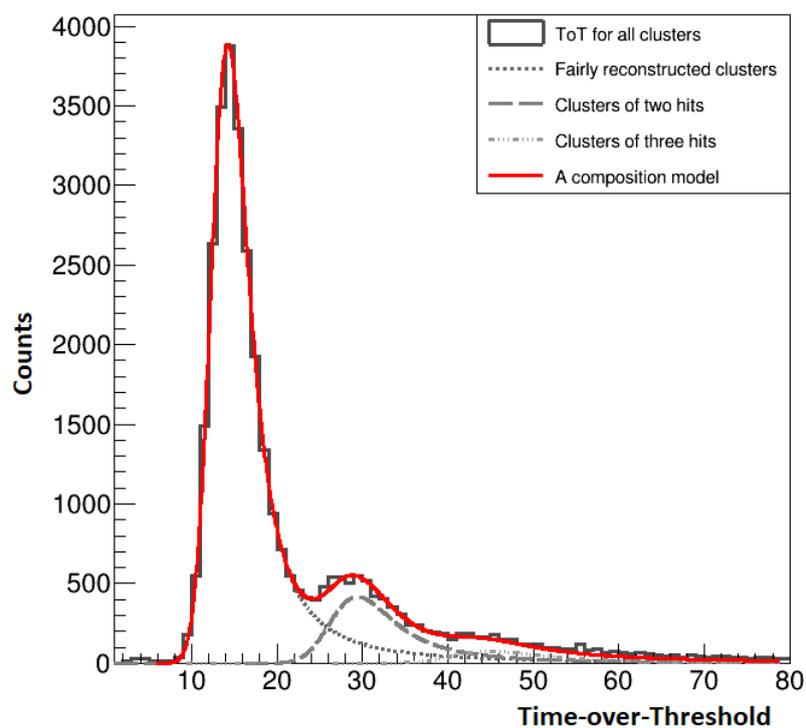


Figure 5.12: A composition model for clusters of any size, for 64 Ikrum DAC.

of clusters combining two hits. The amplitude of clusters composed of three hits is much lower. Further components are omitted due to their residual contribution. The relative ratio of amplitudes will vary as a function of hit occupancy.

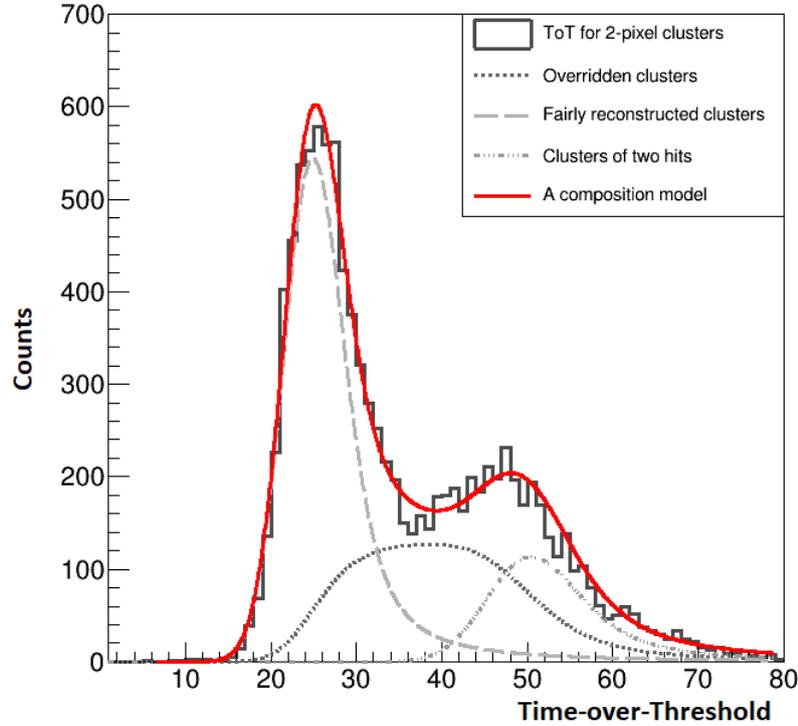


Figure 5.13: A composition model for 2-pixel clusters, for 32 Ikum DAC.

As mentioned earlier, a new hit can override a pixel in an already existing cluster. The incidence of this effect depends strongly on the hit occupancy. In the mathematical model proposed by Akiba, Coco, and myself, the amplitude of such a cluster is a total of a random variable from GL distribution and a random variable from the same distribution reduced by the extinct fraction of amplitude. The exact model depends on the cluster size. For 2-pixel clusters, it is a convolution of GL and another GL convoluted with continuous uniform distribution and can be expressed by the following formula (5.7):

$$(GL \otimes GLC)(x) = \int_{-\infty}^{\infty} GL(x')GLC(x - x')dx' \quad (5.7)$$

where $GLC(x)$ is given by:

$$GLC(x = yz) = \int_{-\infty}^{\infty} GL(z)C(x/z)\frac{1}{|z|}dz \quad (5.8)$$

and $C(y = x/z)$ is a continuous uniform distribution in $(0, 1)$.

Table 5.1: All reconstructed GL (for fairly reconstructed clusters only) parameters from 2018 testbeam in Vetra. Parameters nomenclature corresponds to (5.4).

Ikrum DAC	Model	MPV	MSE	c	A	μ	σ
16	1 cl.	39.7	572.7	5.33	132.9	36.01	3.93
32	2 cl. diag.	24.89	222.7	1.29	299.7	24.59	2.98
32	all diag.	23.83	1033	2.79	679.1	22.13	2.20
32	1 cl.	22.30	1435	2.25	589.2	21.98	2.28
64	all diag.	14.20	510.1	1.38	1693	14.41	1.48
64	1 cl.	14.72	910.3	0.9	1705	14.31	1.67
96	1 cl.	10.84	2081	0.72	3155	11.07	1.40
128	1 cl.	9.31	2070	0.62	4271	9.28	1.15
160	1 cl.	8.32	1375	0.49	5118	8.22	1.04
192	1 cl.	7.62	2088	0.42	7441	7.45	0.93

Now, a complete model for 2-pixel clusters can be recreated, consisting of one GL (5.4), GL first convolution (5.5), and the GL first convolution corrected for overridden clusters (5.7). A visualized model for the testbeam data is presented in Fig. 5.13. Because the number of 2-pixel clusters is much lower than 1-pixel clusters, the amplitude fluctuations and signal-to-noise ratio are higher, causing MSE of the fit to be higher as well. The ratio of (5.4) and (5.7) evinces the participation of overridden clusters in 2-pixel clusters pool and is mathematically related to the average hit occupancy².

Tab. 5.1 presents the fit parameters for all models that were possible to build with the approach described above. These include 1-pixel cluster distribution for a few different Ikrums that were scanned during the testbeam. Fit parameters for the distribution of all clusters with the diagonal condition for Ikrum DAC 32 and 64 is presented as well. It is technically possible to fit the model for higher Ikrums, although the narrow distribution in such a complex model happens to lead to an undesirable high MSE. The fit for Ikrum DAC of 16 is not possible to get due to saturation, similar to Fig. 5.6. On top of that, the fit parameters for 2-pixel cluster distribution from Fig. 5.13 are included. The relationship between Ikrum DAC and ToT is inversely proportional and can be modelled by a rational function $a/x + b$, where b is the discharge rate independent from the DAC setting [387]. The relationship is presented in a separate graph in Fig. 5.14 using MPV from Tab. 5.1, and the resulting function is $y = 567.4/x + 5.249$.

The MPV of GL under the SPS beam was earlier measured to correspond to 16.6ke^- channel input charge. The mean of the three reconstructed MPVs for 32 Ikrum DAC from Tab. 5.1 is 23.68. A discharge time of the 3 fC feedback capacitance (see Fig. 3.24) is found as $25 * 23.68 = 592$ ns. The slope of the pulse shape is then $16.6 \text{ ke}^- / 592 \text{ ns} = 28.04 \text{ e}^- / \text{ns}$

²The profile of hit occupancy at the SPS testbeam is not uniform and has a circular symmetry [345].

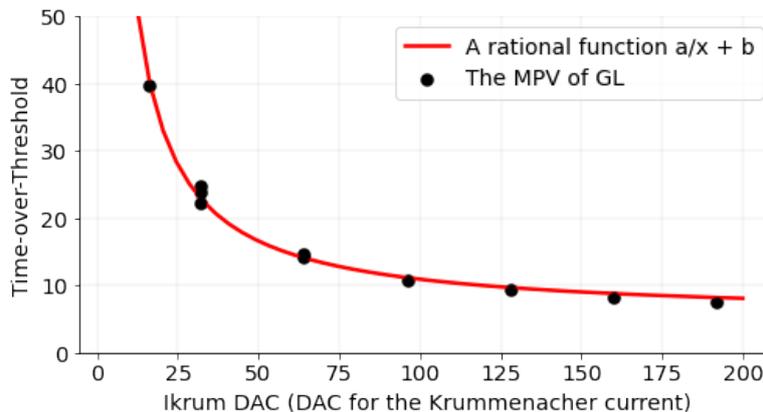


Figure 5.14: A measured MPV in function of Ikrum DAC.

for 32 Ikrum DAC. Once the slope of the pulse is found, it is possible to nearly estimate the pulse shape, assuming the rising time is linear and less than 25 ns, and ToT of the pulse at the signal threshold level is measured.

5.2 Test Pulse scans

Discussed in Section 4.5 Test Pulse (TP) scan is used to estimate the gain of the analog preamplifier, and to reconstruct ToT of the signal as a function of the input charge. The analysis in both cases is applied to each channel individually. In order to measure the amplitude of TP, it is scanned around the optimized pedestal by shifting the signal threshold. When the readout is saturated, threshold DAC is recognized as lying in the noise region. Outside that region, the number of injected TP is seen until the threshold shift reaches the TP amplitude, after which TP cannot cross the signal threshold and no TPs are seen anymore. The noise fluctuations slightly affect the amplitude of TP seen in the readout, usually around 5-10 threshold DAC. Therefore the amplitude is found at the level where 50% of injected TP failed to reach the signal threshold. Because TP is sent by switching the *VTP_COARSE* and *VTP_FINE* (see Section 4.5) at the input capacitor, TP has alternately positive and negative polarity. Those are usually symmetric in respect to the pedestal with little fluctuations, although there were some evidence that TP can be asymmetric when utmost trimming is used. Exemplary histograms of counts as a function of shifted signal threshold are presented in Fig. 5.15. The noise is normally distributed and capped to 63 the number of events (due to 6-bit ECS readout saturation). Because it is not possible to get enough bins on the noise region edges to appropriately

model the normal distribution, it is skipped in the fitting. TP is seen as a sigmoid function (S-curve function) of three parameters, the amplitude (which is fixed to a number of TPs), the steepness, and the shift. Fitting a function of effectively two parameters is relatively simple but requires a fast algorithm, as it is to be done in each channel individually.

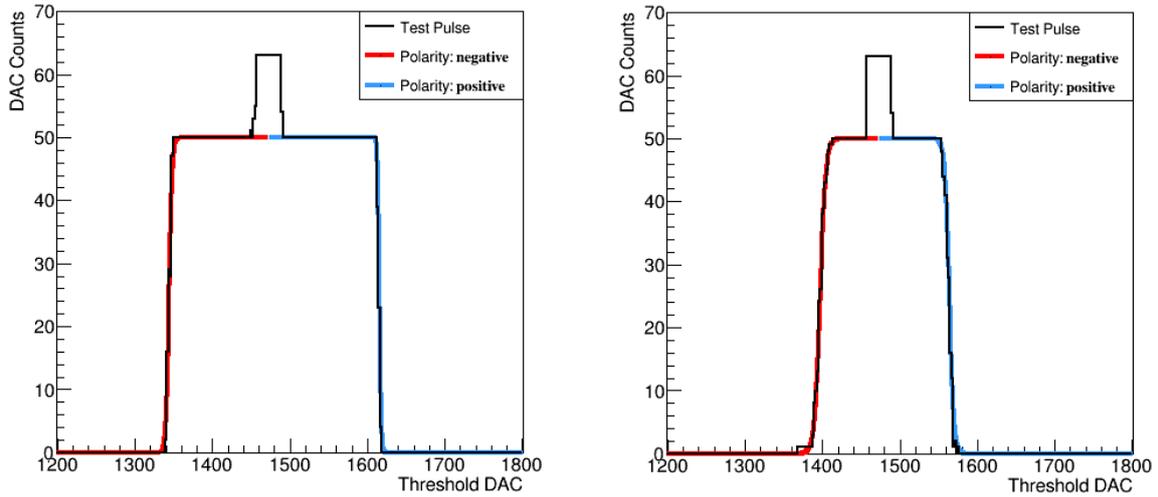


Figure 5.15: Exemplary TPs of different amplitude seen in two readout channels.

When a few TPs of different heights are studied, it is possible to estimate the amplitude as a function of input charge. In order to assure that no asymmetry biases the calculations, the amplitude of the pulse is averaged from positive and negative polarity. The input charge of TP can be calculated as explained in Section 4.5. The processing sequences in

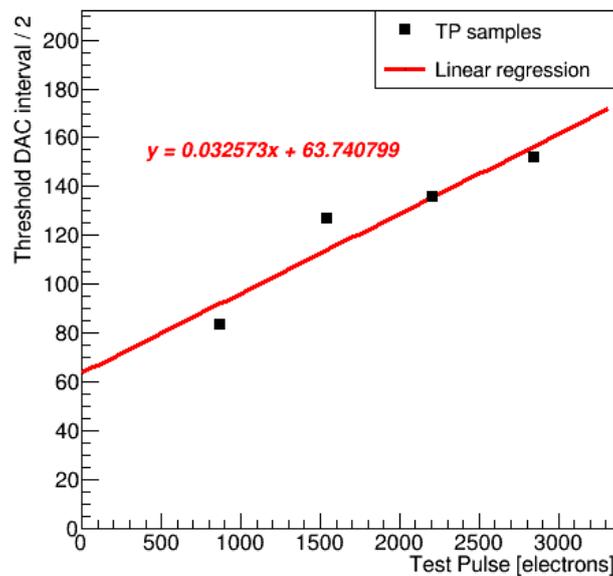


Figure 5.16: Amplitude of TP in a function of injected charge.

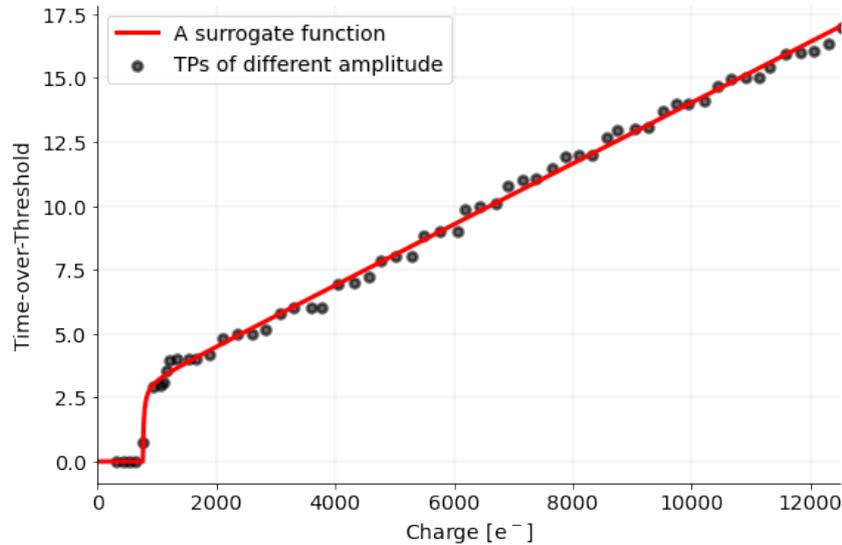


Figure 5.17: A surrogate function fit for the TP scan for 32 Ikum DAC.

Vetra for the gain analysis are *VPGain* and *VPGainAnalyser*. The latter produces data that can be further monitored by *VPGainMonitor*. A template sequence for the full gain measurement procedure is encapsulated in *Vetra*, and the option to select only a limited number of pixels is available. The exemplary output for four different TPs is shown in Fig. 5.16, where two of the points correspond to TPs from Fig. 5.15. A slope of the linear model fitted to those points represents the differential gain of the pixel, in this case 0.0326 DAC/e^- . The offset in the function is a result of non-linear behavior in the noise region, where the differential gain is higher.

A bunch of studies can take advantage of the gain reconstruction procedure. The gain is necessary for energy deposition studies, as it acts as a transformation function between the charge and signal amplitude. The gain can serve as a channel property to monitor in the function of irradiation. It is also helpful when equalization is done using TPs³.

While the above TP reconstruction relies on the counting mode of the readout, the studies can be conducted using the Time-over-Threshold mode as well. The procedure is different because the readout saves only the last TP seen in the shutter. To operate on 50 TPs as before, it is necessary to open the shutter (which corresponds to triggering the readout chain) 50 times. On the other hand, it is enough to scan the amplitude in a function of input charge for only one signal threshold in ToT readout, as the width of the signal, at the optimized pedestal, corresponds to its amplitude. The counting mode could

³Although the equalization procedure operates on noise, it is possible to equalize the pedestals operating on TPs with the fixed amplitude.

measure the whole range of Threshold DAC regardless of whether the noise participated in measured events. Setting the signal threshold at the noise region is no longer possible when switching to the ToT mode, as the yield of 1 ToT count in the noise region proximity does not allow to distinguish whether it was caused by the signal or the noise that passed the threshold at the edge of the clock. Therefore the signal threshold should be higher than a pedestal by at least $5\sigma_{noise}$ (5 times the noise standard deviation). Because the width of the signal is likely lower at that level than at the pedestal, the measured ToT is always underestimated in respect to the real signal width. This effect, in addition to the gain non-linearity near the noise region, inflects the linear function of the input charge, like it is shown in Fig. 5.17. The presented data were taken in the lab for one readout channel and 32 Ikum DAC. The resulting curve, called the surrogate model, representing the relation between the collected charge and ToT counts, is common for pixel detectors operating with ToT and was originally studied by Jakubek [398], and further applied in MediPix and TimePix detector studies [399, 400]. The surrogate model is given in (5.9), where Q is the charge in electrons:

$$ToT = p_0 + p_1Q - \frac{C}{Q - t} \quad (5.9)$$

The meaning of the surrogate parameters is scarcely discussed in the literature, but the studies in Vetra helped to draw some conclusions on their relationship with the signal and readout setup. The parameters p_0 and p_1 define the linear relationship between the charge and ToT high above the signal threshold region. Because a single unit of ToT corresponds to 25 ns (a single clock cycle), one can use p_1 to calculate the discharge current, which is an alternative method to that described before in Section 5.1. The parameters C and t are strongly related to the height of the signal threshold. The linear relationship correctly suggests that the width of the signal should gradually approach 0, but a signal of low ToT at its pedestal (usually 1 or 2) fails to pass the signal threshold and thus disappears when discriminated. While t roughly estimates how much charge is needed for the signal to pass the threshold, C is associated with the shape of the pulse at its peak. At the same time, C and t try to model the gain non-linearity near the noise region. This non-linearity can be observed on the same Fig. 5.17, where it manifests in the noticeable positive offset of the linear model (p_0 parameter). A concurrent modelling objective for t is the reason why it can only roughly estimate the charge needed for the signal to reach the signal threshold. When the offset parameter, p_0 , is scanned in the function of the Krummenacher current, it strongly decreases in the region of lower Ikum (<32 Ikum DAC) to reach a plateau

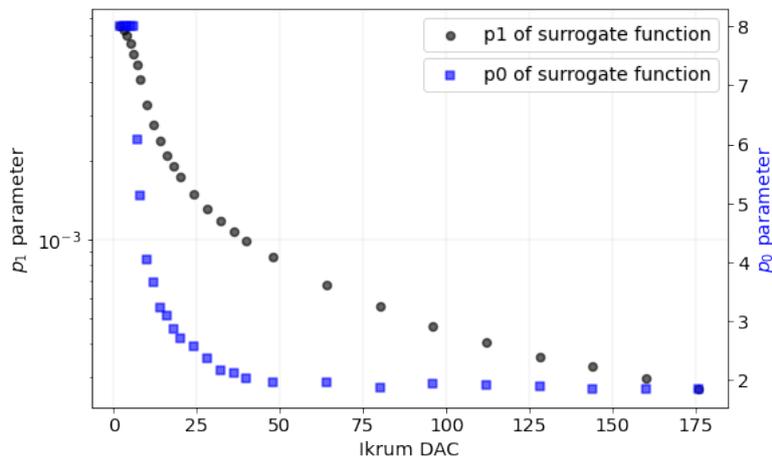


Figure 5.18: Surrogate parameters p_0 and p_1 in a function of Ikrum DAC.

for higher Ikrum (>50 Ikrum DAC), which is shown in Fig. 5.18. This is because the low Ikrum increases the ToT of the pulse but does not elevate low-charge pulses to pass the signal threshold, which escalates the gain imbalance. When p_0 reaches the plateau, its value represents the non-linearity of the analog front-end, which is independent from the Ikrum. The same effect is seen when studying C parameter, which similar to p_0 seems fixed for higher Ikrums.

Because the surrogate parameters provide a lot of useful information on the state of the readout channels, they can be used for studies of irradiated ASICs. Majewski [382] provides an insight into how the parameters can change in the function of irradiation. In particular, p_0 and p_1 can be used in an innovative way to monitor the gain and the discharge current over the detector lifespan. It is highly recommended to glean the results of such scans together with calibration-related data, as long as enough time is available to run quite a time-consuming scanning procedure.

5.3 Analog optimization

Yet another application for Vetra is analog chip working point optimization. The aim of that study is to keep the readout chip configured for the highest performance possible. The analog part of the chip uses 12 biases from individual analog biasing lines from analog periphery. The DAC configuration of those defines the working point of the analog front-end, and essentially the whole analog stage of signal processing. The default configuration was compromised between many criteria and is considered optimized in the standard setup. However, there are some circumstances under which the default settings should be

adjusted. In a scenario of increased leakage currents in irradiated ASICs, lowering the chip temperature is expedient in maintaining the signal-to-noise ratio. This can be pursued by reducing the dissipated power of the front-end electronics. The studies in Vetra proved that reductions in DAC settings can provide significant temperature drop while sustaining the readout performance, which is discussed further in this Section. Another scenario assumes that high irradiation damage of the on-chip electronics changes some of its properties, and therefore its initial working point might not be optimal anymore. In this case, the study aims to reoptimize the DAC configuration of the ASIC and has been, by far, the most complex procedure incorporated into Vetra.

The idea originated from the pulse shape optimization in the old Vertex Locator studied by the Author et al. [1]. It was one of the attempts to understand the rapid irradiation effects in the last months of detector utilization. The studies concerned a refined reduction of biases in the chip to lower the power consumption and its temperature, while maintaining the signal gain and signal-to-noise ratio.

A suite of a few new settings was picked to deliberately lower the most power-consuming DACs, and was arranged partly based on the electronics recommendation and previous observations of trends in DAC scans, and partly arbitrarily⁴. The new settings were evaluated using the attributes of the pulse shapes. The pulse was the analog signal buffered from the analog pipeline in the Beetle chip and could be recreated as the Beetles featured the fully analog readout. Because the interval between consecutive triggers was 25 ns, the solution used to reconstruct the pulse was to delay the electronic readout with respect to either incident particle or Test Pulse, which meant to delay the clock of the TELL1 boards and the clock of the Beetles by the same amount and direction. After consecutive delays of 1 ns, and reading out 6 consecutive triggers (that is 150 ns), the pulse shape could be recreated. A few shapes from the studies are shown in Fig. 5.19, expressed in signal-to-ratio units. The detailed reconstruction procedure and the corresponding DAC settings can be found in [1].

Settings from the suite were later tested by the Author using the simulation of the Beetle signal processing and assisted in the evaluation of the pulse quality. The pulse shape was incorporated in the form of a deep neural network. The ratio of spill-over events⁵ was estimated to decrease in some of the pulses. After investigation, a configuration which traded the small decrease of signal-to-noise ratio (<2%) for the lowered power consumption

⁴The actual studies firstly involved 10 different configurations picked mostly arbitrarily, and a few weeks later, 10 other configurations inferred from the first measurements and previous DAC scans.

⁵A spill-over is the pulse remainder in the next pipeline position, read out by consecutive trigger.

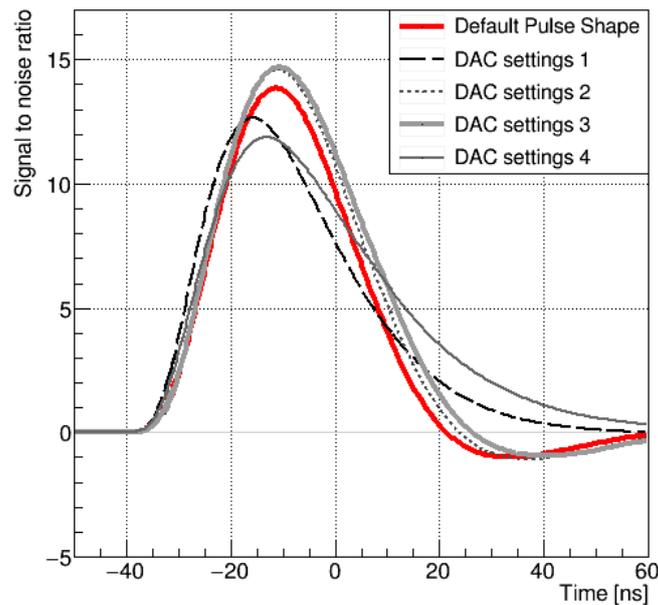


Figure 5.19: DAC optimization of the pulse shape for Vertex Locator before the upgrade.

(3%) and lowered spill-over events ratio (30%) was chosen as the new recommended. Following the encouraging results from the simulation, it was decided to test the best setting in the real detector in November 2018. It turned out that updated settings reduce the chip temperature by over 1°C , even though they were not optimized and were the result of exploring only a few trends in the multivariate analysis. Because the experiment was a month close to the long shutdown, and the new configuration required time-consuming tuning, the configuration was not used in the end, but the promising outcome from the studies convicted that electronics reconfiguration can help to counteract the radiation damage in the Upgrade I detector.

To apply similar studies in the upgraded detector, the procedure has to be adjusted to the fully binary readout of the VeloPix. The direct reconstruction of the pulse is no longer possible, and the methods from Section 5.1 and 5.2 are not precise enough to differentiate subtle variety in the pulse shape. Instead, the settings are evaluated in terms of the analog front-end attributes that are easier to measure, i.e. the signal-to-noise ratio, the power consumption, the signal gain, and the timewalk⁶. The signal-to-noise ratio is calculated as the amplitude of the signal divided by the standard deviation of the noise. The gain is measured as described in Section 5.2. The power consumption of the ASICs can be monitored via Slow Control. To study the timewalk, two TPs of low and high amplitude

⁶Timewalk appears when the latency between hit and registration by the discriminator varies in the function of signal amplitude [401].

Table 5.2: Selected DACs for optimization procedure.

Biasing line	Name	Description	Default	Default [DAC]
V_{in_casc}	Vcasc	Cascode voltage for the input transistor	600mV	199
V_{bias_preamp}	Ipre	Bias voltage for the input current source	830nA (x3)	164
$V_{bias_preampcasc}$	Vprecasc	Bias voltage for the cascode of the preamp current source	400mV	103
V_{fbk}	Vfbk	Feedback voltage of the preamplifier	500mV	143
V_{bias_disc}	Idisc	Current for discriminator bias	1.2 μ A	128
V_{bias_ikrum}	Ikrum	Bias current for Ikrum	25 nA	32

(1.1 ke⁻ and 10.1 ke⁻) are sent to two different pixels, and their Time-of-Arrival (ToA) is read out via Slow Control (it was read out by Low-Frequency when using SPIDR in the preliminary optimization studies). When TPs are delayed by consecutive phases, it is possible to find the region in which ToA is different for low and high TPs, and this way to measure the timewalk. Vetra was equipped with algorithms processing the scans needed for analog optimization, and a combination of those can be used to optimize the analog front-end of the VeloPix ASIC.

From 12 biases of the analog front-end, 6 were selected as relevant for the analog optimization procedure. They are listed in Tab. 5.2 and include biases for the preamplifier, a bias for the feedback NMOS transistor (feedback resistance), a bias for the Ikrum and a bias for three stages of the discriminator. V_{bias_ikrum} , which is a bias for Ikrum, brings a heavy impact on signal shaping and was investigated in detail already in Section 5.1 and 5.1. Small changes of its value do not contribute much to the power consumption of the ASIC, and its value will be most likely reserved for purposes other than analog optimization. On the other hand, the preamplifier bias current Ipre is the most power-consuming bias in the analog front-end. The biases chosen as irrelevant are TP steps fine and course, the signal threshold, and DAC internal biases.

The studies used SPIDR for collecting the data and Vetra for the data analysis. To examine the Ipre relationship with the front-end performance, four TPs corresponding to 1100 e⁻, 2100 e⁻, 3100 e⁻ and 4100 e⁻ charges, respectively, were injected into the readout channels in a pattern of 1 per 16x16 pixels, and the analysis averaged the results from every tested channel. The scan included 15 different values of Ipre varying from 5 to 210 DAC units. At each point, a threshold scan was conducted for each of four TP amplitudes, which also aimed to find the pedestals and the noise standard deviation. The result of such a scan

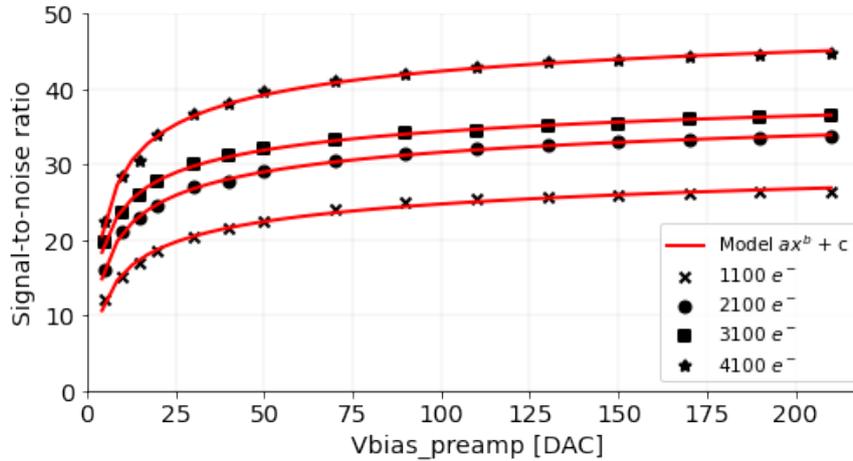


Figure 5.20: Signal-to-noise ratio for different TP amplitudes in a function of V_{bias_preamp} .

is shown in Fig. 5.20, where the TP amplitudes are given in the signal-to-noise ratio. The curves are smooth and were found to be satisfactorily modelled by the arbitrarily selected power function $ax^b + c$. With increased I_{pre} , the SN (signal-to-noise) ratio increases as well, but with a lower rate for higher values of I_{pre} . This is, in fact, derived from the decreasing value of noise standard deviation, and not the increased amplitude of the signal. The amplitude of the signal reaches a plateau for I_{pre} higher than 125 DAC units, which gives a wide safety margin for lowering this bias in the optimization procedure. The pedestal linearly grows in the tested range of I_{pre} , from on average 1450 to 1468 threshold DAC units. The gain is independent of the I_{pre} except for the values of >170 DAC units, where it grows very slightly. The power consumption of V_{bias_preamp} is high, and a change from 5 to 210 DAC units increases the analog power consumption by around 100%, and the total power consumption (including digital DACs) by around 40%. Because this power consumption drop is remarkable, a temperature scan was conducted for the same range of V_{bias_preamp} to measure its relationship with the temperature of the ASIC. During the scan, the ASIC was taking data continuously for 22 hours, and the setting of I_{pre} of 210 DAC units gradually decreased every next two hours, until it reached 5 DAC units at the end of the scan. This scan with I_{pre} decreasing in time was performed to not overestimate the temperature difference from the fraction of the heat aggregating in time in the busy ASIC. During the scan, the temperature of the chip, the chipset board PCB (the intermediate element between the ASIC and FPGA), and FPGA were registered. The results are shown in Fig. 5.21. The temperature of the chipset board and FPGA slightly drop in the function of I_{pre} , but in fact, they slightly increase in time, most likely due to

the temperature change in the laboratory. The temperature of the ASIC decreased in time in function of lowering I_{pre} , and the total drop was around 1.5°C . A timewalk scan in the function of I_{pre} showed that the probability of the timewalk slightly increases when I_{pre} is lowered, and it appears in 20% of the phases for 100 DAC (a phase of the sampling point can be changed via Slow Control), which means there is a 20% chance of observing the timewalk when receiving the signals of high amplitude interval for I_{pre} of 100 DAC. The results show that lowering the I_{pre} can be valuable for power consumption optimization, and the parameter can be safely lowered from the default value of 164 DAC to around 120 DAC, trading around 0.4°C drop for a slightly decreased SN ratio and slightly increased timewalk probability.

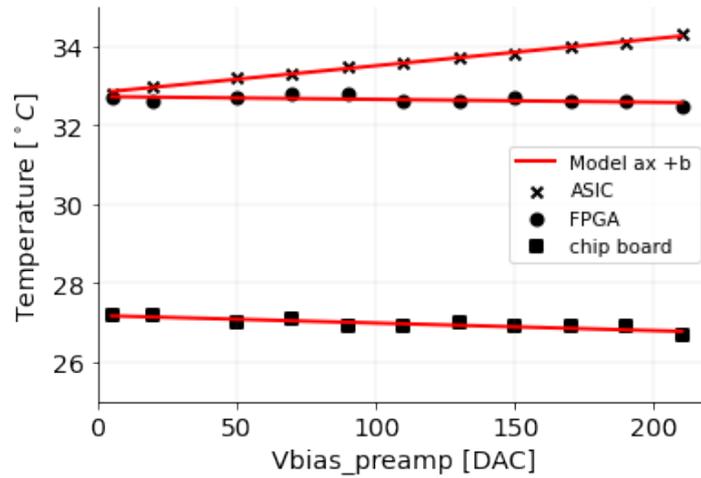


Figure 5.21: Temperature as a function of V_{bias_preamp} , spanned for 12 hours with V_{bias_preamp} decreasing in time.

The next bias to study was V_{bias_disc} , which defines the bias current for the discriminator. The total power consumed by this DAC is around 3.5 times less than V_{bias_preamp} . The scan was done in a similar setup to the previously mentioned and the results of TPs expressed in SN ratio units in the function of I_{disc} are presented in Fig. 5.22. The curves were not consistent for TPs of different amplitudes and no function was found to model the curves. The gain of the signal significantly decreases when the default I_{disc} value is lowered. Moreover, from the tested range of I_{disc} DAC (50 to 150), the pedestals increased from 1250 to 1470 on average, which is a very high ascend. There seems to be no safety margin for lowering I_{disc} without the big trade-offs, and therefore it is not recommended to change it until necessary.

A similar scan applied to V_{in_casc} , which biases the input transistor in the preamplifier,

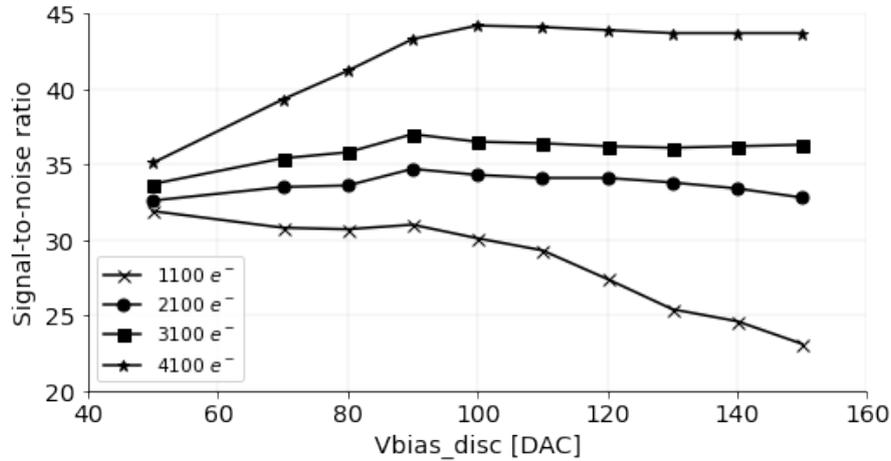


Figure 5.22: Signal-to-noise ratio for different TP amplitudes in a function of $Vbias_disc$.

shows that potential reduction of the DAC is possible as its nominal value of 199 is far from the point when the SN ratio reaches a plateau, but the bias turned out to not contribute significantly to the power consumption of the ASIC. The gain, noise standard deviation and the timewalk are uniform at the whole SN ratio plateau region.

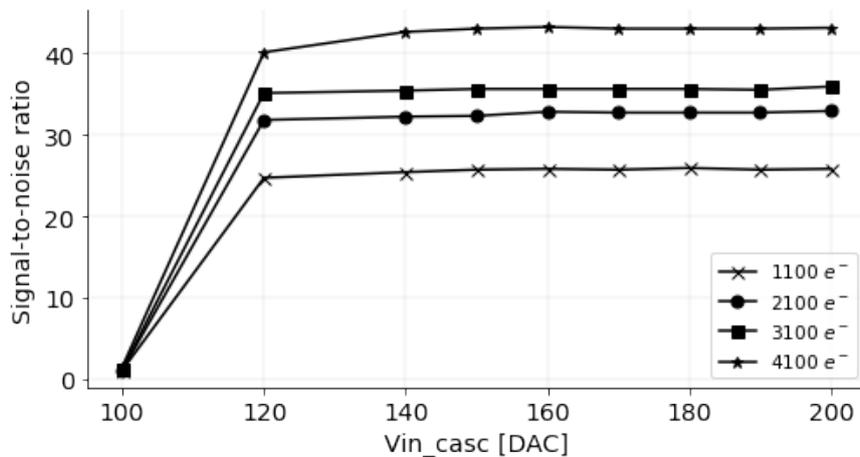


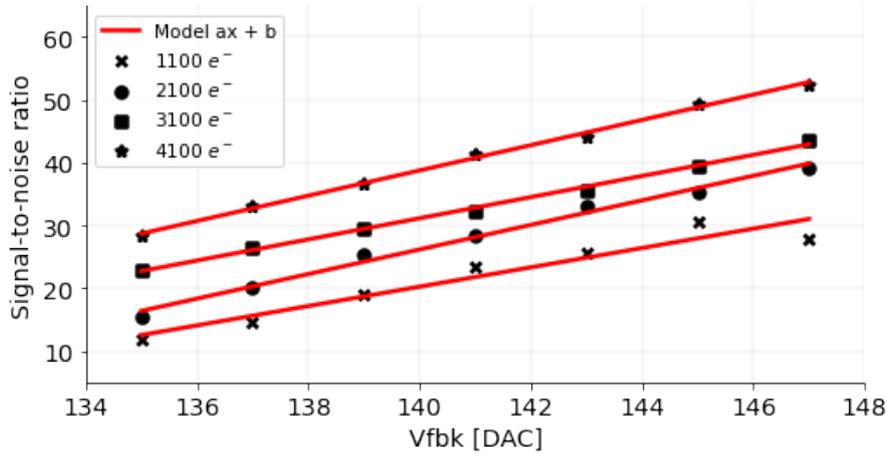
Figure 5.23: Signal-to-noise ratio for different TP amplitudes in the function of Vin_casc .

Scanning the $Vfbk$, which is the bias for the front-end feedback transistor, is only possible in close range of its default value (143 DAC), as the SN ratio drastically changes even for small changes of the bias, as can be seen in Fig. 5.24. Modifying the bias in this narrow range does not contribute to the power consumption significantly. Instead, no safety margin is given and similar to I_{disc} , it is not recommended to change this parameter until necessary or until one needs to study the signal behavior as a function of the feedback resistance. The SN ratio can be modelled by a linear function, at least in the region

Table 5.3: Fit parameters for DAC scans.

Name	Model	TP [e^-]	a	b	c
S/N ($Vbias_preamp$)	$ax^b + c$	1100	-39.035	0.2181	39.041
		2100	-43.646	0.2641	44.536
		3100	-41.585	0.2657	46.586
		4100	-55.072	0.3106	55.533
S/N ($Vfbk$)	$ax + b$	1100	1.541	-195.49	-
		2100	1.957	-247.81	-
		3100	1.684	-204.59	-
		4100	2.012	-242.98	-

between 135 and 147 Vfbk DAC. The parameters of the fits from Fig. 5.24 and 5.20 are grouped in Tab. 5.3.

**Figure 5.24:** Signal-to-noise ratio for different TP amplitudes in the function of $Vfbk$.

The studies show that the only effectual bias for power consumption optimization, which has the acceptable trade-off when lowered and enough safety margin for the SN ratio is $Vbias_preamp$. When the thesis is being written, Vertex Locator is not yet operating, and it was not yet decided whether the value of this bias will be lowered from the beginning, or kept default for some time of the detector operation. Nevertheless, it gives a potential 0.4°C temperature reduction of the front-end hybrids and is noted as a solution to lower the operational temperature of the ASICs when needed.

Chapter 6

Machine Learning for LHCb

Although ML shows a solid and rapidly growing appearance in HEP, its applicability is usually bounded to individual cases, sometimes only in the low-profile form or know-how. Moreover, until a reliable ML method finds popularity in applications, it might become obsolete in favor of other methods from the latest research. This is not entirely true in some of the cases, as the gradient-boosted trees such as XGBoost are adopted massively in their most refined form. Nevertheless, the usage of ML in HEP usually lags behind the actual research in the field.

This Chapter aims to examine whether some of the newest approaches of ML have the potential to far support statistical methods used today in particle modelling. The models such as Generative Adversarial Networks (GANs), even though proved excellent performance in many applications (see Section 2.4), are barely recognized in physics decay analysis. The concepts and ideas proposed in this Chapter are the inventions of the Author and the Supervisor, unless specified or referenced otherwise, and are mostly discussed in Section 6.2. Applications of the proposed approach are presented in Section 6.3, and both are preceded by the introduction and a survey of recent concepts in Section 6.1, with emphasis on those appearing in LHCb.

6.1 Recent Applications

Detector physics and physics decay analysis can be conceptualized as two focal areas for ML applications. The particle analysis typically concerns the multivariate classification problem of selecting events for the candidates for the decay under study. Hence the methods are usually well-validated classifiers. The most popular are BDTs (Boosted Decision Trees), in which the rules learned by the algorithm are intuitive and can be

reasoned by physics criteria (e.g. in the form of kinematical cuts). The experiments pursue the latest technologies for the detectors to possibly maximize their performance, therefore the methods used in the detector physics tend to be more exclusive.

This interrelation can be seen in the literature. The applications of ML in ATLAS and LHCb are well documented [402], and show a potential to grow [403]. There are a few usages of ML in jet physics analysis [404–406], e.g. in ATLAS [407], or CMS [408–410], discussed timely at ML4Jets (Machine Learning for Jets) conference, which has so far held three editions [411], latest in 2021. In 2014, Kaggle (a site of rich community in applicable ML) hosted a competition to analyze the Higgs data [412] with an immense response of 1784 teams, eventually won by XGBoost, which promoted the method (as a follow-up, Kaggle further hosted a competition on high efficient tracking [413], also part of IEEE WCCI competition [414]). Noteworthy previews and explanations of the recent methods used in LHC experiments can be found in MLHEP (Machine Learning for High Energy Physics School) materials [415].

The space for computational intelligence in detector physics is particle identification and track reconstruction, or more generally, pattern recognition [416]. The major challenge for the models derived from computer vision is that the tracks of the particles are usually submanifold structures (which means they are of lower dimensionality to the medium) of high sparsity, while in computer vision the pixels are generally uniformly dense. Some models tried to address this issue, e.g. Submanifold Sparse Convolution Network [417, 418], or others [419]. Some methods were devoted explicitly to LHC applications [420–422], or collider physics in general [423]. In LHCb, deep models were used in long track reconstruction¹ and for online event rejection [424]. For the potential use of ML to speed up detector simulation, some ideas were discussed recently [425].

Another concern is that when the data instances or the features are correlated, a deep model is more likely to overfit or create redundancies, which further roughen the loss function landscape. One solution is to decorrelate the data (there is always a linear transformation that turns the n -dimensional space into an uncorrelated base), which should be done prior to the deep processing. Because the tracking data are often correlated in a part-known fashion, one of the emergent solutions is to analyze the data as graphs [426] using Graph Neural Networks, which help to extract and draw entangled information from the provided dataset [427, 428].

Generative Adversarial Networks are rarely used due to their relatively young age

¹Long track, or downstream track, is a track seen in the whole spectrometer, from the vertex detector to muon chambers.

(formulated in 2014), their complexity and layered structure, and the low interpretability of the adversarial training. It is the Author's view that these models are incomparably more difficult to arrange an effective training of, and sometimes a full investigation of the gradient effects gets literally impossible, as generation from noise vectors makes them, contrary to regular neural nets, stochastic in nature. In addition, the evaluation of the generative model is tricky to define, and the adversarial loss evaluates the mutual performance of the generator and discriminator, while the actual selection power only implicitly. One can use metrics such as prediction accuracy or precision and recall [429], where precision is the discriminator accuracy on the samples marked as true, and recall is the ratio of true samples marked as true to all true samples. Another metric is ROC (Receiver Operating Characteristic) [430, 431], a relationship of the recall and the ratio of false samples tagged true to all samples predicted incorrectly for all possible decision thresholds (though used rather in Autoencoders² than GANs).

Generative models were used in the simulation of RICH detectors to reproduce the particle type probability as a function of track parameters and detector occupancy [433]. Custom GANs were used for simulation of the calorimeters [434, 435] and modelling the cascades of electromagnetic and hadronic interactions in the absorbing material in ATLAS [436]. Models based on graphs were applied to electromagnetic shower generation [437] and in high-granularity calorimeters [438] in general. Wasserstein GAN was applied to fast calorimeter simulation in terms of planned HL-LHC upgrades [439].

The LHCb Particle Identification (PID) system [440–442], and online track reconstruction [443] take advantage of a deep model named ProbNN [444], which yields a probability value corresponding to a given particle hypothesis. A concept of simulating the whole LHCb PID system is illustrated in Fig. 6.1. Because different reconstruction algorithms are present in each detector, the actual PID is combined in ANNPID [445] (ANN - Artificial Neural Network). It takes input from tracking, RICH, the calorimeters and Muon System [446]. Classifier algorithms such as AdaBoost, MatrixNet [447], and neural nets were used to optimize the LHCb topological trigger, a trigger which selects vertices in compatibility with those expected from the decay of a b-hadron [448] (LHCb captures decay topologies after triggering on partially reconstructed b-hadron decays [268]).

Another potential ML usage worth mentioning is Data Acquisition. DAQ systems require extremely low, sub-microsecond latency that would benefit from having deep

²Autoencoder [432] is an artificial neural network aimed to learn a representation (an encoder) of the data in order to reduce their dimensionality while assuming the data tend to be locally correlated (e.g. a pixel in the image is usually highly correlated with adjacent pixels).

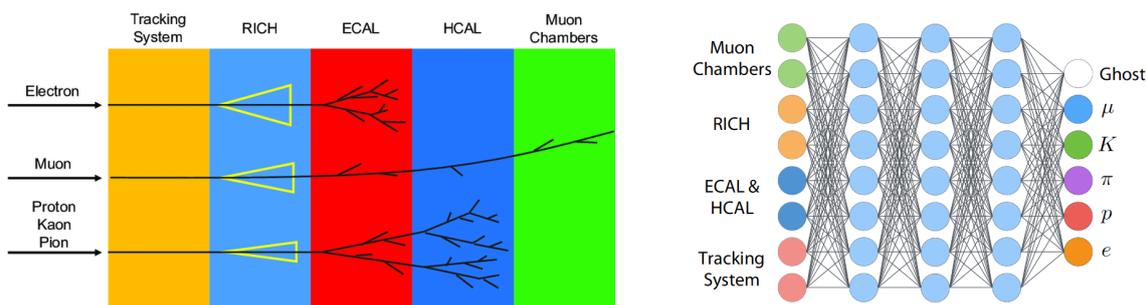


Figure 6.1: The LHCb Particle Identification system overview (left) and proposed deep model application in [441] (right).

systems embedded into the hardware. Duarte et al. substantiated that the use of ML-based triggers in low-latency FPGAs is feasible and proposed a few use-case applications [449].

The most established environments for ML development [450] are PyTorch (released in Sep 2016, Facebook research [451]) and Keras (Mar 2015, Google Brain [452]), with Tensorflow back-end (a library for high-speed graph processing [453]), both implemented in Python language. While Keras is more robust and trains the models using Tensorflow machine code [454] (it also features tools delivered by Tensorflow [455, 456]) or Theano [457] (a library for numerical computations with an ability to run on GPU), PyTorch is more flexible and easier at debugging, purposed for research (especially PyTorch Lightning [458]), optimized for array processing and strongly accelerates on GPU [459], in particular with CUDA (Compute Unified Device Architecture), a computing framework developed by NVIDIA, where the tasks can be parallelized across many GPUs asynchronously to optimize compute-intensive processing. PyTorch uses tensors as general data abstraction and relocates the array computing to GPU [460].

A fairly advanced tool for ML development is TMVA (Toolkit for Multivariate Data Analysis) [461, 462]. It originated as a stand-on application [463], moved in 2013 to ROOT³, where it was embedded as a platform for machine learning [470, 471]. ROOT provides basic supervised learning methods, for instance, the multi-layer perceptron, which is a feedforward neural network⁴. ROOT provides maximum likelihood fitting using MINUIT [472] as an optimizer, and RooFit [473], a library built on ROOT for modelling the distributions of events in physics analysis, where the modelling is done by unbinned maximum likelihood methods [474]. TMVA enhances the methods available in ROOT and provides, among others, gradient-boosted Decision Trees, SVM, kNN, function discriminant

³ROOT [464] is an open-source C++-like programming framework used widely in HEP [465–469].

⁴A term feedforward is used when the stream of information is oriented in only one direction, i.e. the connections between neurons do not form cycles.

analysis (FDA, [475]) and kernel density estimation (KDE, [476]). TMVA allows data preprocessing, such as decorrelation or decomposition. TMVA is compatible with Python and its data objects. It is possible to use TMVA in the Python interpreter directly using PyROOT [477]. PyROOT is also used to import and read files of format native to ROOT in Python.

6.2 Proposed concepts

This Section discusses some theoretical concepts of ML application for physics analysis and mass spectrum reconstruction. The first presents deep modelling as a novel fitting method. The objective of the fitting is to find a suite of parameters for the model that describes the given dataset best. Although it is possible to find such parameters by purely statistical methods, like the method of moments or maximum likelihood, the estimation of the moments (or model parameters in case of maximum likelihood) usually does not go hand in hand with optimizing the fit, i.e. minimizing the MSE between the fit and the dataset. Using a fitting algorithm, based either on gradient descent or metaheuristic, minimizes a loss function, which can be the MSE function, and leads to the best possible fit. Such an approach results, however, in two bothersome aspects of the approximating model. The first is that there is never a warranty that the algorithm converges into the global extremum of the loss function, gets trapped in a local one, or perhaps does not converge at all⁵ The second concern is that the dataset is very rarely a pure distribution but often with the admixture of miscellaneous detector effects that weight the distribution in a hard-to-determine, sometimes highly composite, fashion. If the fitting model does not take into account any admixtures, it relies on a crude assumption that analyzing the dataset with oversimplified model will still provide the best approximation of the actual parameters. It is only when the distribution is no longer visually resembling the model that there is an obvious need to change it, usually to an empirical one (such as the Crystal Ball function [478] used for multi-purpose modelling).

To study how different fitting techniques approach to the distribution, a single Gaussian of fixed parameters (amplitude, μ , σ) was sampled in random intervals as in Fig. 6.2. Three different methods are used to fit the Gaussian model to the data points; the method of moments (estimating μ by the mean and σ by the standard deviation estimator), a metaheuristic fitting algorithm (more precisely, DE algorithm, see Section 2.3) and a simple

⁵When the loss function has regions of local flatness or learning rate is not low enough, the GD algorithm is unlikely to converge, which manifests in different results each time the algorithm is run.

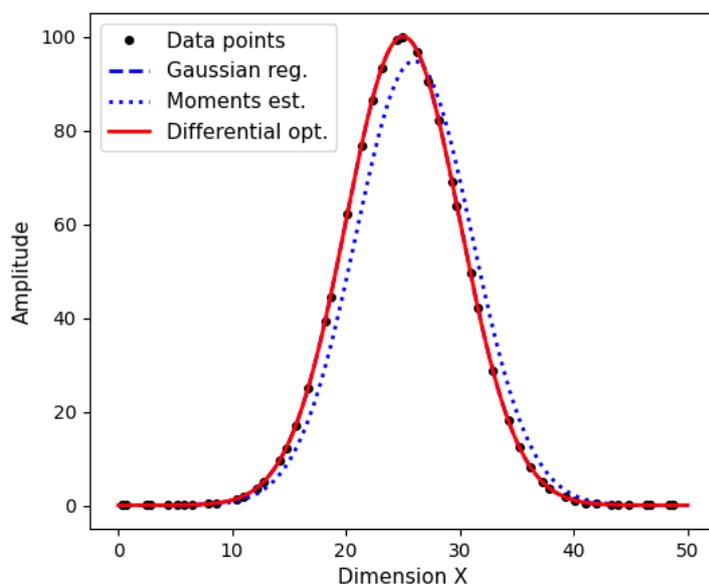


Figure 6.2: Gaussian fit to the ideal Gauss distribution computed by three different approaches.

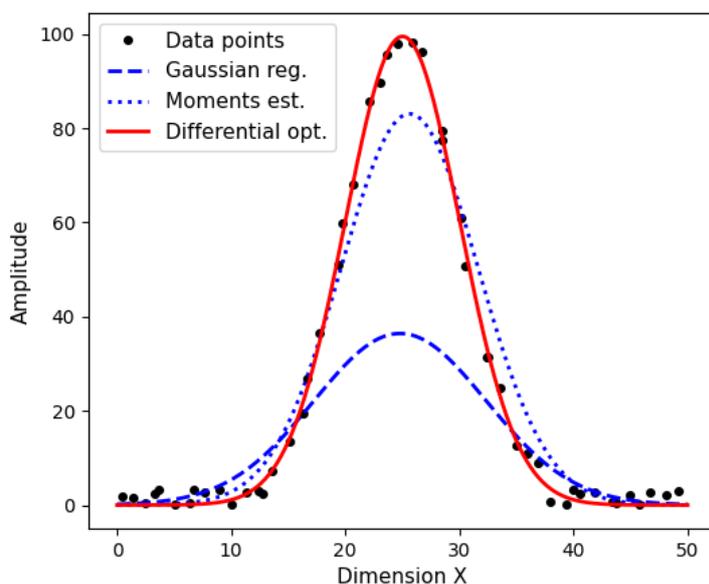


Figure 6.3: Gaussian fit to deformed Gauss distribution computed by three different approaches.

regression. The regression is the least squares regression that can be explicitly derived using the natural logarithm of Gaussian (following the conception in [479]). The regression provides exact model parameters. The DE-based fitting algorithm quickly converges in the right parameters set. The method of moments only roughly estimates the model parameters, which can be seen in Tab. 6.1.

Table 6.1: Gaussian amplitude A , μ and σ obtained by three approaches.

Model	Data	μ_{true}	μ	σ_{true}	σ	A_{true}	A
Regression	pure	25	25	5	5	100	100
	with noise	25	24.799	5	7.5587	100	36.429
Moment Est.	pure	25	25.966	5	5.1035	100	94.777
	with noise	25	25.630	5	6.0064	100	83.061
Optimization	pure	25	25	5	5	100	100
	with noise	25	25.006	5	5.0028	100	99.893

The same fits were repeated for a dataset sampled from the same distribution with a relatively small insertion of normally distributed noise (of $\sigma_{noise} = 5$, whereas the Gaussian amplitude is fixed to $A = 100$). The fits act quite differently; the main reason why the regression fails this much (Fig. 6.3) is that the natural logarithm used in the method can quickly approach large negative values when the height of some samples is reduced to values close to 0 (this could be, however, avoided by adding additional conditions). The method of moments performs progressively worse at minimizing the loss, even though the modelled μ in this particular case is closer to its true value than in the previous case. The fitting algorithm converges into a stable set of parameters that slightly differ from those that were given to the random number generator. Because the DE-based algorithm is one of the most advanced in the field and was proved to successfully optimize hybrid functions of more than 100 parameters [480] (i.e. more than 100 degrees of freedom), it can be assumed with high confidence that the set of parameters found is actually optimal. Following this observation, the garbled distribution is no longer able to provide the original parameters it has seen during creation; instead, the ratio of true and reconstructed σ was $5.6 * 10^{-4}$. On the other hand, 5% noise contribution is followed by only 0.056% shift of the approximated parameter. Postulating a composite model that could recreate the parameters with higher precision is hardly possible, as the noise is purely stochastic and symmetric, and such a model, if succeeded in this particular case, would be inaccurate for another batch of data. However, when the noise contribution added to Gaussian would not be symmetric, a model that could describe the distribution with higher precision should exist.

The dataset can be modelled by a neural network, which approximates the shape of the distribution without any constraints. It is expected for the deep model to give much lower MSE (the exact value depends on the network complexity) than a Gaussian fit. However, there is no facile way to extract or estimate distribution parameters, as the neural network model is essentially a black-box model. A peak can correspond to the expected value, but

there is no statistical measure to approximate the uncertainty of such estimation. Even though fitting the shape via deep models is often very useful, it is only a superior way of building models like Crystal Ball. Such a network is unquestionably a flexible and versatile tool, but with no ability to calculate the parameters of the original distribution.

The solution proposed in the studies was to use a deep network approach for parameter estimation based solely on the shape of the distribution curve. A model input layer taking N points from each sample shape makes the model very complex, as the loss function turns to N -dimensional. If the model is able to learn the relationship between the shape and the parameters of the distribution, it could potentially support statistical methods used in estimation. This approach is not entirely new and was originally conceptualized by Bishop [481], who used the neural network to estimate the width and location of the peak (i.e. mean) of the Gaussian-like spectral lines. The applied model was a multi-layer perceptron, which was trained to estimate the parameters of one spectral line at a time. Bishop articulated that such an approach does not require a big initial setup or guesswork, thereby can be scaled up with the CPU speed or even engineered into hardware.

In continuation of Bishop's research, the idea was to set up a deep model to simultaneously evaluate two Gaussian shapes and to ascertain whether the model could cope with the overlaps, as well as higher fluctuations of the curves, as the studies reported by Bishop did not delve into this aspect.

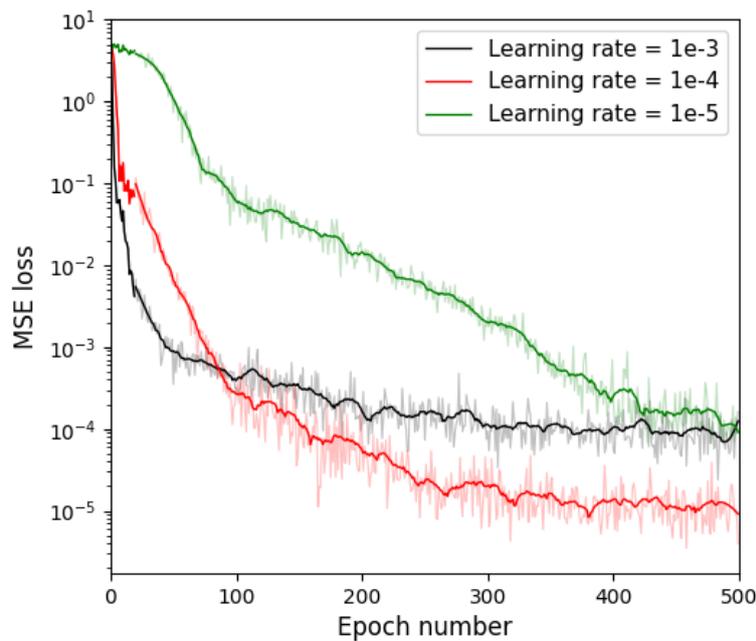


Figure 6.4: The loss function while descending the gradient in the shape learning model.

Because of its flexibility and GPU acceleration, PyTorch was used for deep modelling from this point onward. The code of part of the study discussed below is available in the Author’s github in [482].

The chosen network is a deep feedforward network with $60 \times 30 \times 30 \times 6$ architecture. The input layer has 60 nodes as the function domain is split into 60 bins (covering $C \in [0, 20)$ region), and the output layer gives 6 parameter estimates, which are A , μ , and σ for each Gaussian. Training was run on 5000 generated samples, each consisting of two possibly overlapping functions with parameters varying in the following ranges, $2 < \mu < 6$, $0.1 < \sigma < 2$ for the first Gaussian, and $6 < \mu < 10$, $0.1 < \sigma < 2$ for the second Gaussian, both with fixed amplitudes (3rd and 6th outputs of the model were idle). The number of epochs⁶ was fixed at 500. To fine-tune the learning rate, it was scanned with three proposed values, for which the loss across GD was plotted as a function of epochs, which is shown in Fig. 6.4. Since the loss function is likely to have many complex features (note that the input vector combines 60 data points, hence the loss function is a 60-dimensional complicated function), the loss fluctuates in a logarithmic scale and was smoothed by Savitzky-Golay filter [483]. The lowest learning rate (marked green) makes only small leaps in the loss function overall landscape⁷ and, arguably, is too tiny to swiftly escape when attracted by local minima. On the other hand, a high learning rate is unable to converge but moves in circles around the extremum. The learning rate of $\alpha = 10^{-4}$ is the quickest to minimize the loss and was therefore chosen as optimal.

The results of shape learning are illustrated in Fig. 6.5, where the residuals of predicted μ and σ are shown in the form of histograms using the testing dataset (the testing and training datasets were generated separately). The distribution of $(\mu_{true} - \mu_{predicted})$ residual in principle falls in the $(-3, 3) * 10^{-2}$ region, likewise for $(\sigma_{true} - \sigma_{predicted})$ in $(-0.7, 0.5) * 10^{-2}$ region. The residual distribution for σ is asymmetrically skewed toward negative values, indicating that the prediction somewhat overestimated the real parameters. This is most likely an effect of peak overlapping, which always seems to stretch the size of the distribution. Noteworthy is the fact that very low residuals are reached; the averaged error was $< 0.5\%$, even though the training and testing datasets contained even tightly overlapped peaks. A serious disadvantage of this method is the absence of an explicit way to determine estimation uncertainty. While the fitting algorithms based on the maximum likelihood approach can estimate the uncertainty (e.g. Cramér–Rao bound on

⁶The epoch is a full loop of the training over the batches or individual data samples.

⁷The notion of landscape is an ML-specific jargon description of the loss function behaviour. It originates from the attempts to create comprehensive visualization tools for representing the loss functions.

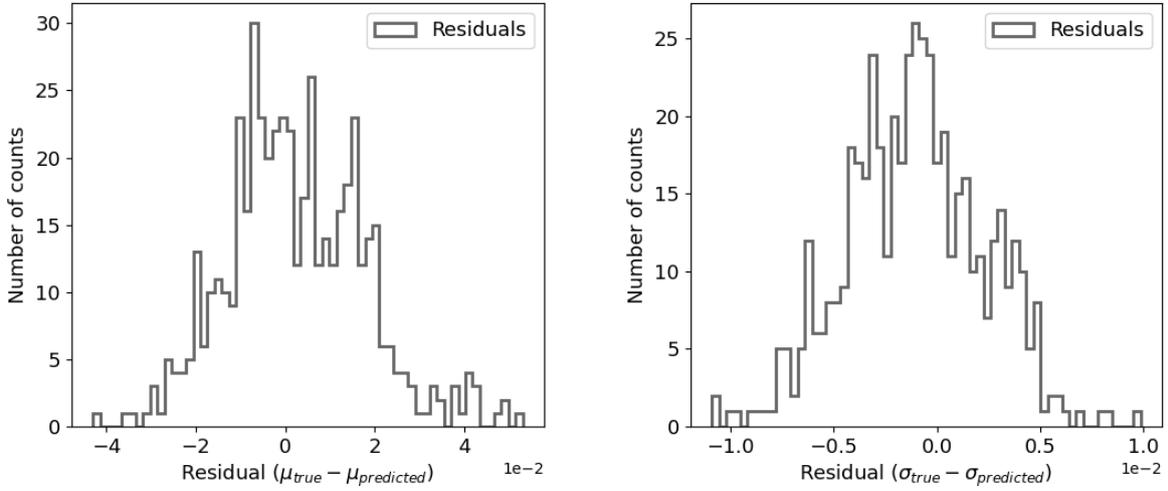


Figure 6.5: Residuals of ground truth and prediction for μ (left) and σ (right).

the variance [484]), the neural network, even if performing better, cannot. The question of whether it is advantageous to seek the best possible physics measurements in the absence of uncertainty is broad and perhaps beyond the scope of this dissertation. A remedy to this dilemma could be to progress the state of understanding of the deep network’s latent space attributes, though as long as it remains enigmatic in mathematical reasoning, that is unlikely to happen. Some of the Author’s ideas were to lay out a deep model that could, in turn, estimate uncertainties based on statistical samplings, such as bootstrap, or to look for certain trends in the latent space by mining and studying small pools of neurons to search for a relationship between error propagation and physics uncertainty, and this will most likely be the line of further research on this matter.

Having a trained model for statistical estimation, the follow-up idea was to investigate whether deep modelling could be used in shape synthesis in such a way that the model could tell what fundamental components the distribution could be decomposed into, and so from what fundamental components it could originate. In accordance with algebra, any function can be modelled as an infinite series of orthogonal polynomials (computer models for such fittings are called Mixture Models [485]) or even Gaussians (Gaussian Mixture Models, GMM [486]), where the probability density function is modelled by using multiple Gaussian kernels [487], e.g. by RooKeysPdf in RooFit [488]. As a matter of fact, the possible representations can also be infinitely many, but by imposing certain constraints on the modelling, such as the limited number of fundamental components, a number of expected components, their approximate shape or expected bounds on certain parameters (e.g. amplitudes), a deep neural network could judge from the very shape

of the distribution which composition suite is more likely. The model, if learned on the custom-defined shapes, could estimate the contribution of each fundamental component and, using the approach from previous paragraphs, estimate the parameters upon which the distribution was defined. This problem refers directly to the optimal shape adjustment in physics analysis, where one looks for the possible components in the physics data distributions, scrupulously modelling the kinematical cuts and known detector effects on MC (Monte Carlo) data. Because of the complexity of the decays, multiple cuts and constraints, the contributions from misclassified decays are often not defined by analytic formulas but by the shape that is proposed to fit the observed distribution the best. The shape of mass distribution in non-trivial decays quickly gets complex after the consecutive event selection criteria are applied. The analysis of such data is very complicated and may involve a large number of nuisance parameters. Creating a sufficient background model requires much effort and is usually a blend of several components. In standard methods commonly used in such analysis, a large number of free parameters decreases the chance that a fitting algorithm would converge to the optimal point. The multidimensional parameter space may have many local minima, which in turn may correspond to a different set of physics parameters, even though all of them provide a similar result in terms of the minimization of MSE. When only a tiny data sample is available (e.g. in rare decays), it furthermore increases the estimation instability. Eventually, results may not be reliable enough to make a statistical enrichment to the exploration of the physics process.

The network recognizing the fundamental components was configured to have $60 \times 60 \times 4 * N_{out} \times N_{out}$ layer architecture, where N_{out} is the number of all possible compositions that can be built of fundamental components (the choice came from a simple grid scan which found the configuration achieving the best results). Activation functions were chosen to be LeakyReLU (see Chapter 2.2), and the input layer to extra scale the data to $(-1, 1)$ (so-called batch normalization). To make the output vector comply with the probability interpretation of the density function, a softmax layer was appended as the last layer⁸. The training is supervised by one-hot encoded labels of possible softmax outcomes, and the performance is evaluated as the softmax accuracy (the outcome with the highest probability is picked as the class guessed by the network).

The study included two fundamental components, a decreasing exponential function (describing the combinatorial background) and Gaussian. It was assumed that 7 different composite functions can be defined using these basic components, which are up to three

⁸Softmax [489] is a function transforming a vector or array into a discrete PDF (Probability Density Function).

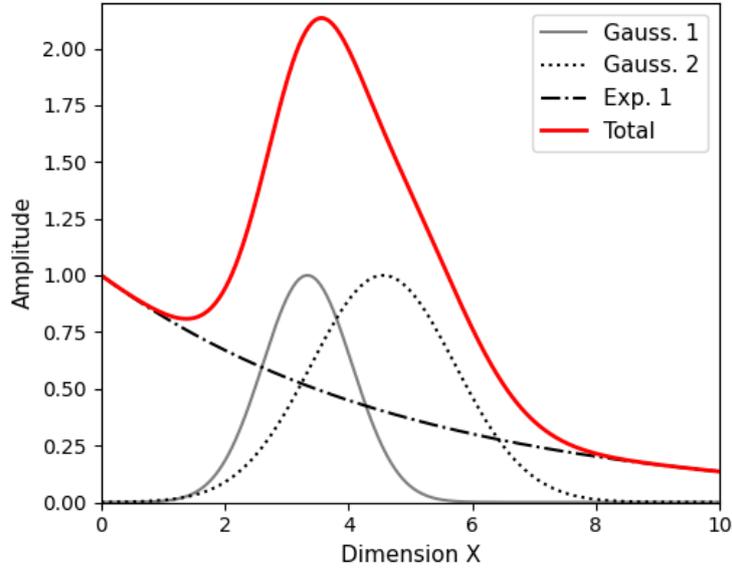


Figure 6.6: A sample function composition used for training the deep model aimed to recognize fundamental components.

Gaussians with the optional presence of the exponential function. The domain $C \in [0, 10)$ was split into 60 bins, and 2000 samples of each possible composition were generated. Gaussians were generated without individual boundaries on parameters like before, and instead, the parameters always varied in $2 < \mu < 8$ and $0.1 < \sigma < 2$ (this means some samples might have got heavily superimposed Gaussians). An exemplary sample, being a composition of two Gaussians and one exponential function used for model training, is shown in Fig. 6.6. Note that the peaks are not distinct, but the shape with asymmetric edges of the combined peak can still guide the network toward the right classification.

After training, another 2000 samples from each possible combination were generated to test the prediction accuracy. The results are presented in Table 6.2. The model is surprisingly effective, even though it has to deal with overlapped functions. A single Gaussian was classified most of the time correctly, but some of the samples (1.3%) were chosen by the network as two Gaussians, and vice versa. This is the effect of confusing the model by including tightly superimposed functions in the training data. The model was later trained on many more possible compositions, and its accuracy dropped to approximately 80% when distinguishing between seven and eight overlapped Gaussians.

When studying the shape of a mass spectrum distribution, one could use generative modelling to reconstruct the shape from a suite of given parameters that have physics interpretation (e.g. invariant mass). Since GANs (Generative Adversarial Networks) work stochastically on noise vectors, they can be brought to the fore in generating MC samples.

Table 6.2: Shape classification accuracy (top row are the classes).

Model	G	2G	3G	3GE	2GE	GE	E
G	0.987	0.013	0	0	0	0	0
2G	0.022	0.951	0.027	0	0	0	0
3G	0.001	0.060	0.938	<0.001	0	0	0
3GE	0	0	0.002	0.943	0.054	0.001	0
2GE	0	0	0	0.041	0.955	0.004	0
GE	0	0	0	0	0.002	0.998	0
E	0	0	0	0	0	0	1

It is possible to incorporate the least squares function as the loss function in GANs [490] (instead of binary cross-entropy), similarly to how the fitting algorithms would optimize the loss function, minimizing the Pearson χ^2 divergence (therefore it can also be used in mass spectrum modelling). Eventually, Generator and Discriminator can become two independent systems serving different purposes. The Generator can act as an alternative to MC generators or to confidence interval calculation methods like jack-knife [491] or bootstrap. It could also support or even replace the BDT approach (Boosted Decision Tree) as a superior technique. After the physics selection, the resulting mass distribution of the signal decay may still be considered contaminated with various background events, which is, in turn, a place where the event discriminator could be adopted.

**Figure 6.7:** The image-to-image translation of the Sierra Nevada using CycleGAN [164].

Advanced GANs dedicated to image-to-image translation may find even more intriguing applications. The detector apparatus and the detection deficiencies always leave a footprint on collected physics data, which is further laboriously modelled using detector simulation studies and MC data. If the image-to-image translation model was to be applied to this task, it could learn how the detector interacts with physics data, and how to further appropriately convert the data by extracting the detector impact functions. Image-to-

image translation models, such as CycleGAN (see Section 2.4), were successful in a variety of tasks, converting images of Google Maps to satellite maps, animals to other animals (e.g. horses to zebras), summer to winter sceneries (see example in Fig. 6.7), or even artworks to impressions of another artistic fashion or realistic photos. CycleGAN is able to learn complex relations between pixels in high-resolution images that are challenging to quantify in a mathematical way. The model attempts to find a translation function by finding a latent representation of the image and then renders some layers of this representation, leaving the rest intact. In this regard, if the first deep layer is responsible for learning a general relationship between the physics parameters, the detector effects could be modelled by the latent representation of subsequent hidden layers. It is true that perceptual assessment of generated images machine learning papers present cannot be used in physics analysis, and the application of such models in this field demands more reliable and trustworthy methods in a statistical sense. Nonetheless, this matter deserves continuing research, and a keen interest in the latest in machine learning can firmly push forward the progress of analysis methods used in modelling so far.

6.3 Mass spectrum reconstruction of $B^0 \rightarrow Dh$

The studies presented in this Section are mostly preliminary and aim to explore possible strategies for reconstructing the mass spectrum by improving methods generally used in physics analysis today. Some of the proposed concepts from the previous Section are applied to MC data. The methods are evaluated on $B^0 \rightarrow Dh$ decay ($B^0 \rightarrow D_s^- \pi^+$).

The generative modelling study commenced with the analysis of $B^+ \rightarrow K^+ K^- K^+$ and $B^- \rightarrow K^- K^+ K^-$ decays (the idea for this channel came from LHCb Open Data in 2020) which was studied by Kacprzak, Szumlak, and myself [492]. A few variants of GAN were used to learn the generation of MC data from RapidSim (an engine for fast MC data generation of heavy-quark hadron decays [493], adjusted for LHCb geometry). They were the original GAN, BGAN (Boundary-Seeking GAN) [166], DualGAN (2017, Yi et al., using unsupervised learning for data multiplication and image-to-image translation [494]), and WGAN (GAN with Wasserstein loss [160] and convolutional layers [165]). The model's implementation was inspired by a collection of PyTorch code of GAN variants in [495] written by Lindernoren. At a glance, neither of the variants showed close capabilities to even a simple MC generator, and the implication was that adversarial systems are not well-suited for modelling complex mathematical statements, but rather tend to learn

general features over small patches of the data. Because many researchers have raised the matter of using GANs in physics analysis in the past two years, the study was reexamined recently to capture the shortcomings of generative systems and to come up with some thoughts and ideas for possible adoption based on the Author’s experience.

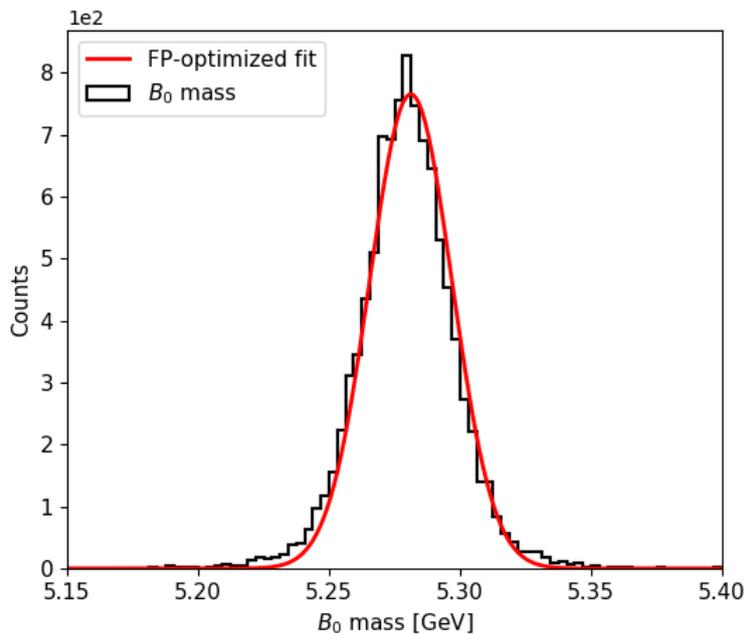


Figure 6.8: A reconstructed mass of B^0 from $B^0 \rightarrow D_s^- \pi^+$ with FP-optimized fit, where possibly the lowest MSE is reached by the algorithm.

To study the mass spectrum reconstruction of B^0 in $B^0 \rightarrow D_s^- \pi^+$, a dataset of 10^4 samples was generated in RapidSim, simulating the LHCb simplified geometry. The mass of B^0 and the energy of the decay can be fully reconstructed from D_s^- and π^+ kinematical variables (momentum). The resulting mass spectrum is then a function of six parameters and gets a Gaussian shape. The actual B^0 mass is 5.27963 ± 0.0002 GeV according to 2022 PDG [496]. To study the reconstruction efficiency, ten different methods were chosen and used for the mass spectrum fit. The model requires only three parameters (A , μ , σ) and is evaluated on 90 bins of the distribution histogram. The MLE (Maximum Likelihood Estimation), in this case equivalent to the method of moments, was chosen as a benchmark for other methods. The Stochastic Gradient Descent (SGD) was taken as a naive optimization algorithm (greedy strategy). Genetic Algorithm (GA) and Differential Evolution (DE) were used in the standard setup (mutation as bit flipping, 5% per bit in 32-bit floating representation). A metaheuristic BFPA (Biotic Flower Pollination Algorithm) by Lukasik and myself (see Section 2.3) with a code in the Author’s github repository in [497] was used with the hyperparameters earlier optimized in the BFPA paper [6].

Table 6.3: Model parameters of B_0 mass spectrum fit for a suite of reconstruction methods.

Estimation method	A	μ	σ	CPU time* [s]
MLE (Max. likelihood)	771.6	5.2792	0.0161	<0.1
Original FPA, Yang [134]	765.8	5.2803	0.0157	11.5
BFPA, Lukasik and myself [6]	765.8	5.2803	0.0157	7.2
Stochastic GD	763	5.2826	0.0160	2.1
GA	765.2	5.2815	0.0155	11.3
DE	765.8	5.2803	0.0157	12
LSHADE [150]	765.8	5.2803	0.0157	5.8
DNN	759.2	5.2759	0.0166	<0.1**
DNN, Bishop's approach [481]	780.1	5.2807	0.0162	0.1**
CNN	783.4	5.2731	0.0168	<0.1**

* Intel(R) Core(TM) i5-3470 3.20 GHz, no multi-threading

** Once trained (GPU powered)

To compare BFPA with the original FPA by Yang [134], it was used as well. The best non-hybrid DE-based optimization algorithm LSHADE [150] (Section 2.3) was used. Last three methods are deep models, a common approximating DNN (Deep Neural Network), DNN in Bishop understanding for the shape learning, and finally, a CNN model, triple convoluted (three convolution layers reducing the dimensionality of the whole patch in steps $90 \rightarrow 30 \rightarrow 15 \rightarrow 3$ before the first deep layer⁹), with the latent space shape corresponding to DNNs (architecture details are given in the previous Section). The mass spectrum shape and an exemplary fit obtained by FPA are shown in Fig. 6.8.

Fit parameters A , μ , σ were obtained from all models in order to compare their performance. Each model excluding ordinary DNN yields the parameters explicitly. The parameters in DNN are computed using MLE of the approximated shape, which means the mean of the learned shape is the estimated Gaussian μ . The reconstructed parameters are grouped in Table 6.3, given together with the CPU time of the algorithm execution. The execution was stopped while the algorithm could not reach a better solution anymore or converged in one certain point (which means the loss reached a plateau region). MLE gives a very close to correct result of 5.2792 GeV (0.0004 GeV deviation) and is actually the most accurate from the suite. FPA, BFPA, DE and LSHADE reached a solution of 5.2803 GeV, which is most likely the global minimum of the MSE loss function, with a little bit more deviation to MLE and approaching the B^0 mass from the upper side. While Bishop's DNN reached 5.2807 GeV, the other methods are less accurate. MLE is the quickest to evaluate the mass with almost no CPU time cost. LSHADE required less

⁹The idea for reducing 90-dimensional input to 3-dimensional was that the distribution can be actually expressed by only 3 parameters, as it was assumed to be Gaussian.

number of evaluation than other optimization methods, and its processing time is the lowest. All three tested deep models do not require much time to estimate the parameters once fully trained, but performed worse than MLE and optimization algorithms. It is worth noting, however, that the networks operated on a complex 90-dimensional space of the input vector. An interesting observation is that the standard deviation estimator σ is a little bit lower for the optimized fit (0.0157 GeV for FPA, BFPA, DE and LSHADE) than MLE provides (0.0161 GeV). On the other hand, this might suggest that when the LHCb geometry simulation is included, the Gaussian model is not perfect in modelling the data and underestimates the distribution tails like in Fig. 6.8. Even though MLE was derived for the Gaussian model, in this specific case it is equivalent to the method of moments, which is less accurate, but always valid. If the Gaussian model is no longer correct, MLE estimator turns into a still correct method of moments estimator. This advantage is not present in models based on optimization algorithms. Neural networks can always tune to the given dataset, and even minimize MSE to zero by overtraining when latent space is big enough to reproduce all the data features including even the noise fluctuations. This lowered MSE does not seem to lead, however, to improved parameters, at least compared to MLE. Because of the lack of theory for parameter uncertainty in deep models, the number of nodes in the network cannot be freely increased due to the risk of overtraining. If there was a connection between approximated model and the parameters resulting from MLE, and, in particular, their uncertainties, one could pick the deep network with a such number of nodes, that the MSE and uncertainty would be minimized at the same time. A perspective of discovering such a relationship is lucrative, as it could revolutionize the statistical calculations, poorly supported by ML methods by far. Moreover, there are some clues that such a relationship exists, as the neural network, regardless of its size, is always a continuous function $f(x)$, and thus the transition function $g(x)$ that transforms $f(x)$ into the considered analytical model (a Gaussian in this specific case) should always exist. If the expression of μ was transformed by the inverse transformation $g^{-1}(x)$, it would yield its latent space representation in the deep model. In a similar way, the uncertainty of μ could be obtained, or eventually computed by measuring the error propagation between the latent space representation of the given parameter. This is certainly a matter worth further research and will be pursued by the Author in the future.

To study the generative system's capability to recreate physics events, the same data sample of $B^0 \rightarrow D_s^- \pi^+$ was used in the learning process of GAN. Because GAN requires a sophisticated hyper-parameter configuration for the training to keep Generator G and

Table 6.4: Optimized hyper-parameters of GAN.

Hyper-parameter	Settings
Latent noise vector	10
G architecture	10 x 300 x 600 x 6
G activation functions	2 x LeakyReLU $\alpha = 0.2$, 1 x Tanh
G outputs	B^0 , π^+ momentum
Optimizer	ADAM, $\beta_1 = 0.5$
D architecture	6 x 600 x 300 x 2
D activation functions	3 x LeakyReLU $\alpha = 0.2$
Learning rate	$1.6 * 10^{-4}$
Batch size	500
Optimizer steps	$N/500$ ($N=10^4$) * epoch
Max. epoch	500

Discriminator D at the constant learning pace¹⁰, the setup was tested multiple times such that the optimal learning phase can be achieved. The hyper-parameters of the model used in the study after optimization are listed in Tab. 6.4. The dimensionality of the latent noise vector is chosen to be 6, as it guarantees for the network to learn the physics functions only with a linear transformation of the uncorrelated base, as there can be some correlations in 6 variables of momentum (the lower or higher number of random noise dimensionality would no longer keep a linear transformation).

The architectures of G and D are similar, where G returns 6 variables of momentum, while D the class of the sample. The activation functions in G are two LeakyReLU (the negative slope coefficient $\alpha = 0.2$, positive slope 0.5). A hyperbolic tangent as the last intermediate activation function is applied to carry the non-linearity physics function of momentum generation, while the aim for the inmost LeakyReLU functions is to guide the network to recreate the linear transformation between variables. This assumption for the roles should be satisfied when the loss function is actually minimized in the learning process. Batch normalization is applied to keep the real variables close to non-linearity points in LeakyReLU functions and to prevent a vanishing gradient in the hyperbolic tangent layer. ADAM was used as optimizer ($\beta_1 = 0.5$, an internal ADAM property, the exponential decay rate in gradient weighting). The batch size was chosen to be 500, and because the total of epochs was 500 as well, the number of gradient descent steps (thus the number of evaluations) was 10^4 .

The learning rate α was carefully scanned by changing its values between $5 * 10^{-5}$ and $5 * 10^{-3}$, where the best possible value was found to be $\alpha = 1.6 * 10^{-4}$. The adversarial

¹⁰When D trains faster than G, the G's optimizer makes bigger and more generic steps to minimize the increasing loss (it turns into exploration phase), while D wastes time for a local search (exploitation phase) for the untrained G, thus the performance spike of D is only elusive.

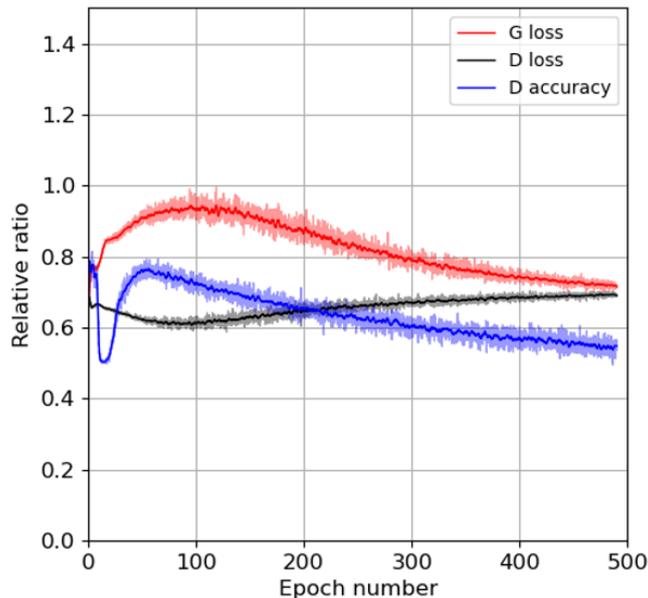


Figure 6.9: Generator loss, Discriminator loss, and Discriminator prediction accuracy for the learning rate $\alpha = 1.6 * 10^{-4}$ during the training of GAN.

loss for G and D, together with D prediction accuracy (accuracy is defined as the number of correct guesses to all guesses) for the learning rate of $\alpha = 1.6 * 10^{-4}$ is shown in Fig. 6.9. With this learning rate, a perfect training scenario is achieved; prediction accuracy approaches 0.5 on the upper side as the training continues, which means G is steadily improving in generating samples that are indistinguishable from the real ones, challenging D incrementally. It is important to stop the training before the prediction accuracy gets close to 0.5, as G starts to notice that D is yielding random guesses, and the feedback mechanism to compute adversarial loss for G is no longer profitable. Note that sheer prediction accuracy is not enough to tell whether the model is trained properly, as long as the accuracy is averaged on real and fake samples. The loss of D and its accuracy is always in the counter-trend. To ensure G and D train at virtually the same pace, their adversarial losses should both approach 0.75 in the longer term. Sustaining such a textbook learning process in physics reconstruction is a remarkable accomplishment, as GANs hardly achieve this condition even in image synthesis, a field native to these models.

For an outlook of selected other configurations of the hyper-parameter space, see Fig. 6.10. For a relatively high learning rate $\alpha_1 = 8 * 10^{-4}$ and the interior architecture 300×600 in both G and D (more precisely, it is always reversed in D; 600×300 in this case), the training is highly unstable, and G struggles to provide samples credible enough. When the same learning rate is maintained but the number of nodes in the inner layers of the

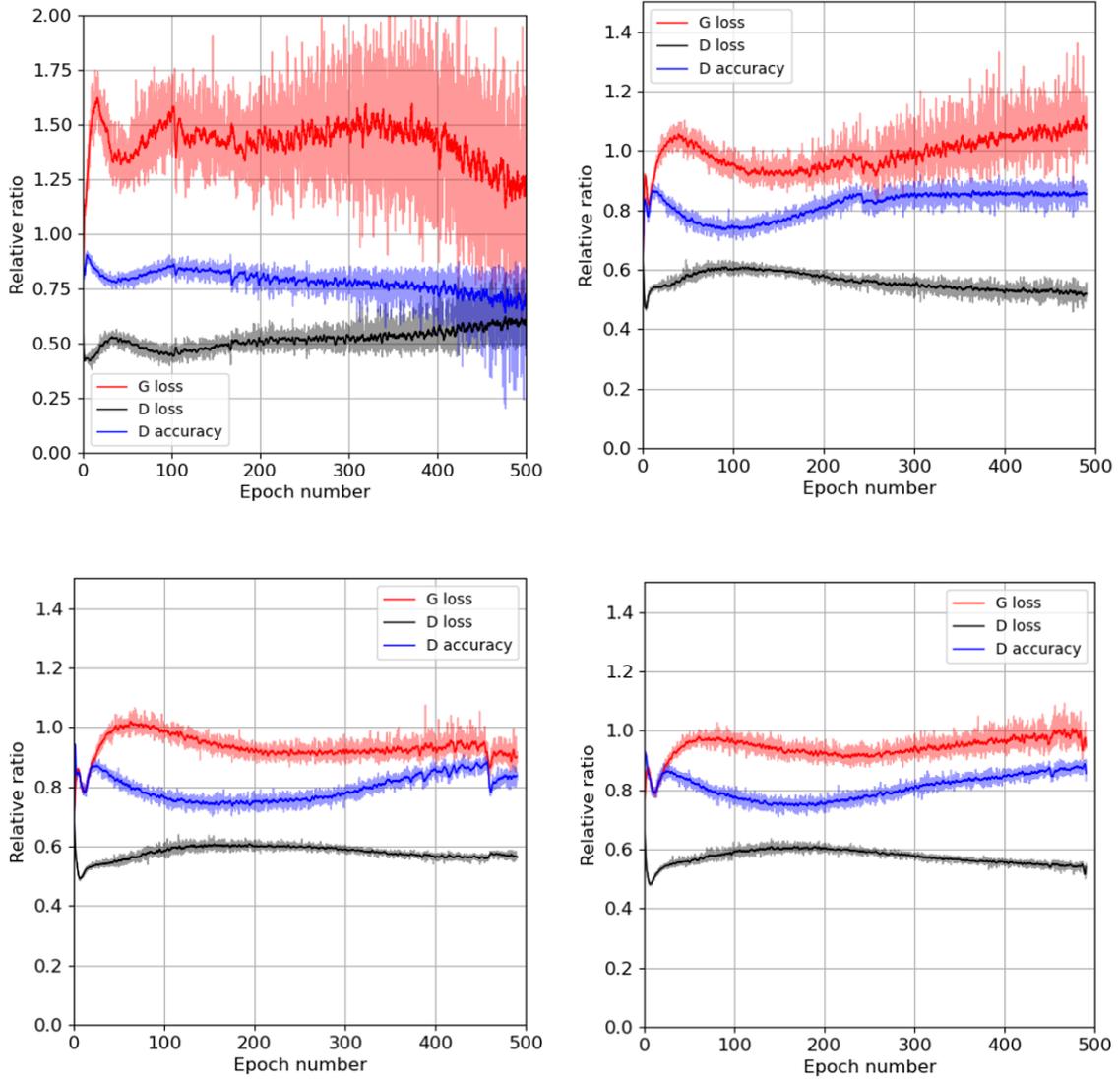


Figure 6.10: Examples of scans taken for hyper-parameter space optimization. Learning rate $\alpha_1 = 8 * 10^{-4}$ and the inner architecture 300×600 (upper left, note that the range of Y axis is increased compared to others), $\alpha_1 = 8 * 10^{-4}$ and 400×800 (upper right), $\alpha_2 = 4 * 10^{-4}$ and 300×600 (bottom left), and $\alpha_2 = 4 * 10^{-4}$ and 400×800 (bottom right). Savitzky-Golay filter is used to smooth the tidal curves.

networks is increased to 400×800 , both G and D will start to experience high adversarial losses. For lower training rate $\alpha_2 = 4 * 10^{-4}$ but still higher than optimal (the optimal was $\alpha = 1.6 * 10^{-4}$), the losses are not getting decreased as much as in the optimal scenario in previous Fig. 6.9. The conclusion is that the learning rate should always be matched to the specific architecture of the system, and thus be the last element to optimize. This substantial role in the learning rate makes it more significant to oversee compared to

regular neural networks, as its suboptimal values can even make GAN not train at all.

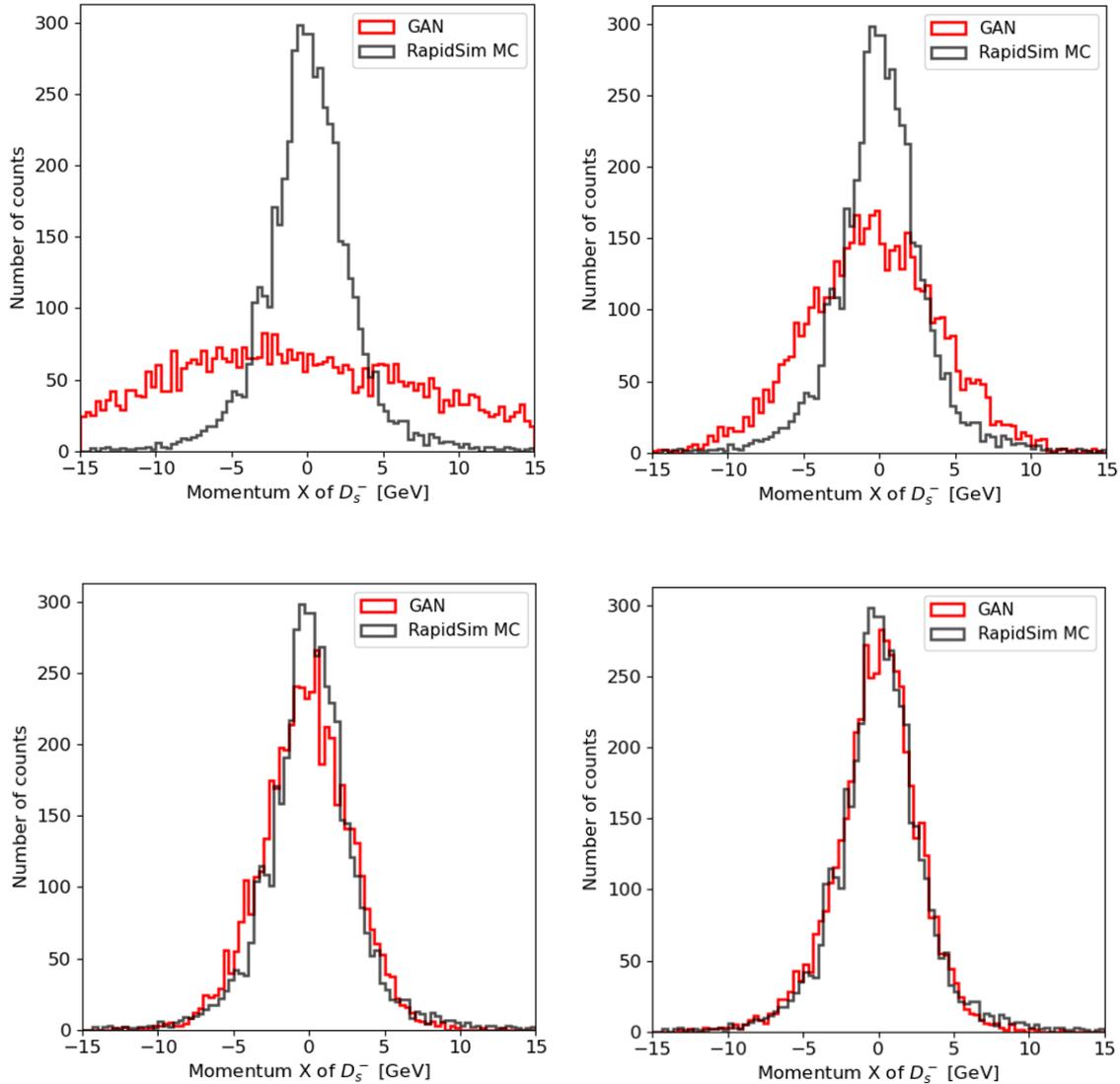


Figure 6.11: The model sliced at 100, 200, 300 and 500 epochs (fully trained).

A stepwise glimpse of how the model learns recreating the distribution in the latent space is shown in Fig. 6.11, where the generated distributions of X momentum of D_s^- are illustrated for training slices at 100, 200, 300 and 500 epoch number. The model consistently minimizes the adversarial loss (note that the MSE is no longer used) and would have continued to improve the reconstructed distribution had it not been stopped due to the risk of converging accuracy and unreliable feedback from the discriminator. Nevertheless, the model severely struggles in recreating the mathematical dependencies between subsequent real values, and cannot straightly recreate the mass distribution that

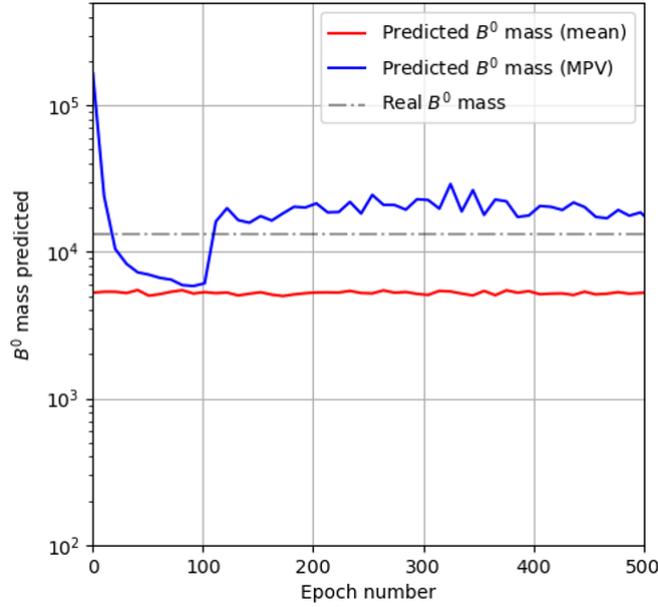


Figure 6.12: A reconstructed mass of B^0 computed as the mean and the MPV of the spectrum obtained from the generated momenta distributions.

is a function of the momentum of D_s^- and a few other variables, which can be seen in Fig. 6.12. In other words, the model is able to learn the distribution shape, but cannot properly recreate the hidden correlation ties between the input real values. This can be either caused by the lack of MSE minimization or the high complexity of the mathematical correlation. A clue that a wrong choice of the loss function can be the reason might be seen when the binary cross-entropy of discriminated samples is replaced by the Wasserstein distance, which significantly improves the model in learning the correlations. Though the model is not able to recreate the mass anyhow, it begins to nicely recreate the total energy of the products, which is given by a similar but less complex function (the actual invariant mass can be expressed as $m = \sqrt{(E^2 - p^2)}$, where E is the total energy). The calculated total energy of the decay products from the generated distributions of momenta is visualized in Fig. 6.13. This nice improvement shows that GAN (in this case, GAN with Wasserstein loss) can be successful at the task if heavily adjusted. A lot of other ideas to deal with this problem were invented and investigated by the Author and independently by the other members of the Krakow LHCb group, partly discussed in [7].

One of the solutions proposed by Szumlak was to equip the loss function with a component that reflects the mathematical expressions used to compute the invariant mass. The solution proposed by the Author was to add input nodes to the generator which would take the computed invariant mass from the real values given to the other nodes at the same

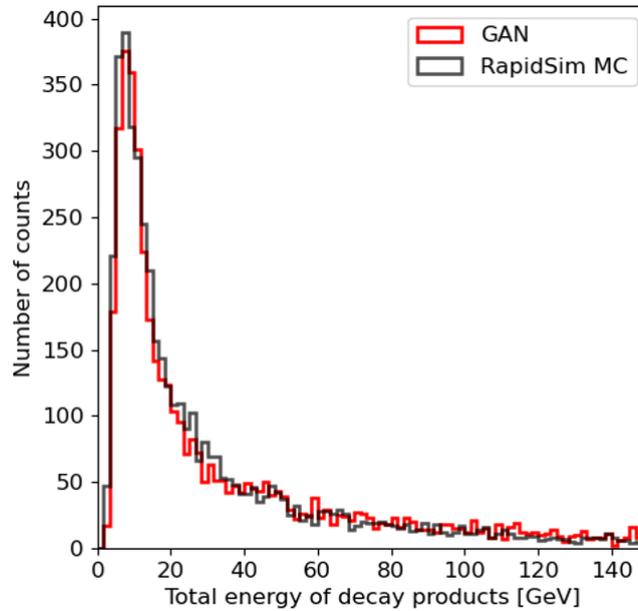


Figure 6.13: The reconstructed total energy of $B^0 \rightarrow D_s^- \pi^+$ products obtained from the generated momenta distributions.

sample. This way GAN gains a support expression that helps in recreating the correlations between the generated kinematical variables. An example of the generated samples of invariant mass of B^0 is shown in Fig. 6.14, for two different stages of the training, after 250 and 500 epochs, respectively. The fits to the generated distributions are made using the Gaussian model and were optimized by BFPA (similar to Table 6.3).

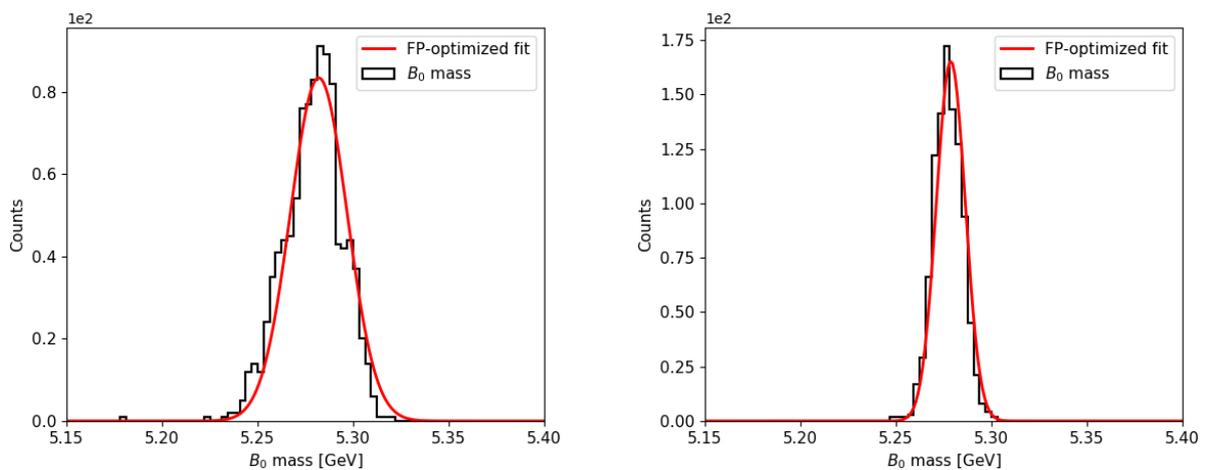


Figure 6.14: A reconstructed mass of B^0 obtained from the generated momenta distributions by the GAN with Wasserstein loss customized input nodes, after 250 epochs (left) and 500 epochs (right).

Table 6.5: Model parameters fitted to the reconstructed B^0 mass spectrum by various GAN variants.

Generative model	Est. model	A	μ	σ
GAN	MLE	168.1	5.31	0.0081
	BFPA	165.7	5.315	0.0084
GAN + Wasserstein	MLE	165.1	5.279	0.0075
	BFPA	165.3	5.278	0.0076
WGAN (DCGAN + Wasserstein)	MLE	165.3	5.272	0.0079
	BFPA	164.9	5.268	0.0081
BGAN	MLE	170.4	5.362	0.0083
	BFPA	170.5	5.361	0.0082
GAN, Morskyi and myself [5]	MLE	168.3	5.31	0.008
	BFPA	166.1	5.309	0.0082
DualGAN	MLE	169.7	5.279	0.066
	BFPA	170.1	5.279	0.065

In the final study, a few modern GAN concepts were tested in the same task as above to check their relative performance. Those are the original GAN, GAN with Wasserstein loss, GAN with convolutional generator and the Wasserstein loss (WGAN in Arjovsky’s approach [160], see Section 2.4), BGAN, GAN with the improvements proposed by Morskyi and myself [5] (code and studies available in github of Morskyi [498]), and DualGAN. To estimate the reconstruction precision, BFPA fitting was applied in a similar way as in Fig. 6.14, and the MLE approach was used independently. The results of such a comparison are presented in Table 6.5. The estimated parameters are different depending on the chosen modelling and the variant of GAN. The real invariant mass of B^0 , $m = 5.27963$ GeV, was approached the best by GAN with Wasserstein loss and DualGAN. While the success of the latter model is not surprising, as DualGAN is a model dedicated to data multiplication (it learns to modify the data rather than generate it from scratch), the high precision of GAN with Wasserstein loss is intriguing. Worth noting is the fact that WGAN shows weaker generation capabilities, which is most likely caused by the convolutional layers in the generator. Together with the fact that GAN with Wasserstein loss significantly improved the generation capabilities presented in the previous part of this Section, it makes the model very promising for potential application in the future. In particular, it is possible that changing and testing different forms of the loss function can further improve the GAN performance and generation capabilities in physics analysis, and is surely a direction for further studies on this matter.

Chapter 7

Footprint and prospects

The software platform for Upgrade I Vertex Locator presented and discussed in this thesis is used on daily basis at LHCb to calibrate and monitor the vertex detector, which currently is at the final developing stage before the commissioning. The platform will serve as a calibration, monitoring and pixel detector studies tool during LHC Runs 3 and 4. The software was written by the Author of this thesis, and the platform was connected and embedded into the LHCb software and hardware ecosystem, as described in Chapter 4. The platform allows for analysing the data for studies that were hardly possible before, introducing calibration data emulation, data quality monitoring, smart optimization of the clustering models, and many other functionalities.

A wide range of Machine Learning methods and modern concepts were discussed in the thesis, beginning from the state-of-the-art in Chapter 2, to practical applications given in Chapter 6. Many ideas for using intelligent systems in physics analysis were proposed and examined, articulating the advantages that this approach could bring, but also potential issues that have to be addressed in order to successfully employ ML-based methods in the aforementioned field. Although structures like Generative Adversarial Networks, invented in 2014, were a revolution in Machine Learning, their use in High Energy Physics is still very scarce. The proposed ideas, although mostly preliminary, fill a large gap that has shaped in recent years between the rapid pursuit of new artificial intelligence concepts in computing technology and their use in other fields of science.

The Author wishes that both the software platform for Vertex Locator and some discussed concepts for physics analysis will provide a tiny indirect contribution to the research to make further breakthroughs in discovering the universe.

Bibliography

- [1] P. Kopciwicz et al., *Simulation and Optimization Studies of the LHCb Beetle Readout ASIC and Machine Learning Approach for Pulse Shape Reconstruction*, Sensors 2021, Vol. 21, pp. 60-75
- [2] P. Kopciwicz et al., *The LHCb vertex locator upgrade - the detector calibration overview*, Journal of Instrumentation, Vol. 17, 2022, C01046
- [3] P. Kopciwicz et al., *The upgrade I of LHCb VELO - towards an intelligent monitoring platform*, Journal of Instrumentation, Vol. 15, 2020, C06009
- [4] P. Kopciwicz et al., *Software Platform for the Monitoring and Calibration of the LHCb Upgrade I Silicon Detectors*, Acta Physica Polonica B, Vol. 50, 2019
- [5] P. Kopciwicz, V. Morskyi, *Exploration of the evolution-based footprint on the Generative Adversarial Networks*, In proceedings of: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), 2022
- [6] P. Kopciwicz, S. Lukasik, *Exploiting flower constancy in flower pollination algorithm: improved biotic flower pollination algorithm and its experimental evaluation*, Neural Computing and Applications, Vol. 32, 2020, pp. 11999-12010
- [7] P. Kopciwicz et al., *A modern machine learning approach to statistical estimation for B meson mass modeling*, Acta Physica Polonica B, 2022 (submitted)
- [8] I. Sarker, *Machine Learning: Algorithms, Real-World Applications and Research Directions*, SN Computer Science, Vol. 2, 2001
- [9] S. B. Kotsiantis, *Supervised Machine Learning: A Review of Classification Techniques*, Informatica, Vol. 31, 2007, pp. 249-268
- [10] X. Daniela et al., *Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages*, International Journal of Computer Science Issues, Vol. 4, 2009
- [11] N. Pandiangan et al., *Implementation of Decision Tree and Naive Bayes Classification Method for Predicting Study Period*, Journal of Physics Conference Series, Vol. 1569, Is. 2, 2020, 022022
- [12] M. Colombo, *Bayesian Cognitive Science, Unification, and Explanation*, The British Journal for the Philosophy of Science, Vol. 68, 2017, pp. 415-484
- [13] Q. Liu, *Supervised Learning*, Encyclopedia of the Sciences of Learning, 2012, pp. 192-211
- [14] L. Rokach, O. Maimon, *Decision Trees, Data Mining and Knowledge Discovery Handbook*, Springer, Boston, MA, pp. 165-192
- [15] E. Fix et al., *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*, Report, USAF School of Aviation Medicine, Randolph Field, Texas, 1951
- [16] R. Fisher, *The Use of Multiple Measurements in Taxonomic Problems*, Annals of Eugenics, Vol. 7, 1936, pp. 179-188
- [17] A. Tharwat, *Linear discriminant analysis: A detailed tutorial*, AI Communications, Vol. 30, 2017, pp. 169-190

- [18] C. Chatterjee et al., *On self-organizing algorithms and networks for class-separability features*, IEEE Transactions on Neural Networks, Vol. 8, 1997, pp. 663-678
- [19] G. Demir et al., *Online Local Learning Algorithms for Linear Discriminant Analysis*, Pattern Recognition Letters, Vol. 26, 2005, pp. 421-431
- [20] Cortes et al., *Support-vector networks*, Machine Learning, Vol. 20, 1995, pp. 273-297
- [21] L. Wang et al., *Combining decision tree and Naive Bayes for classification*, Knowledge-Based Systems, Vol. 19, Is. 7, 2006, pp. 551-515
- [22] S. Mazinani, K. Fathi, *Combining KNN and Decision Tree Algorithms to Improve Intrusion Detection System Performance*, International Journal of Machine Learning and Computing, Vol. 5, Is. 6, 2015, pp. 476-479
- [23] C. Tan, Steinbach, *Data Mining Classification : Basic Concepts, Decision Trees, and Model Evaluation*, 2004
- [24] M. Azad et al., *Decision Rules Derived from Optimal Decision Trees with Hypotheses*, Entropy 2021, Vol. 23, 2021
- [25] J. F. Baldwin, D. Xie, *Simple Fuzzy Logic Rules Based on Fuzzy Decision Tree for Classification and Prediction Problem*, Intelligent Information Processing II, 2004, pp. 175-184
- [26] A. Subasi, *Chapter 2 - Data preprocessing*, Practical Machine Learning for Data Analysis Using Python, 2020, pp. 27-89
- [27] H. QingHua et al., *Information entropy for ordinal classification*, Science China Information Series, Vol. 53, 2010, pp. 1188-1200
- [28] F. Farris, *The Gini Index and Measures of Inequality*, The American Mathematical Monthly, Vol. 117, 2010, pp. 851-864
- [29] A. Elnaggar, M. Aziem, H. El-Deeb, *A Comparative Study of Game Tree Searching Methods*, International Journal of Advanced Computer Science and Applications, 2014
- [30] M. Bramer, *Avoiding Overfitting of Decision Trees*, Principles of Data Mining, pp. 121-136
- [31] S. German et al., *Neural Networks and the Bias/Variance Dilemma*, Neural Computation, Vol. 4, Is. 1, 1992, pp. 1-58
- [32] S. Doroudi, *The Bias-Variance Tradeoff: How Data Science Can Inform Educational Debates*, AERA Open, 2020
- [33] P. Mehta et al., *A high-bias, low-variance introduction to Machine Learning for physicists*, Physics Reports, Vol. 810, 2019, pp. 1-124
- [34] H. Almuallim, *An efficient algorithm for optimal pruning of decision trees*, Artificial Intelligence, Vol. 83, 1996, pp. 347-362
- [35] E. Frank et al., *Reduced-Error Pruning With Significance Tests*, Working Paper Series, 1999
- [36] B. Kiran et al., *Cost-Complexity Pruning of Random Forests*, Lecture Notes in Computer Science, Vol. 10255, 2017
- [37] G. Baudet, *An analysis of the full alpha-beta pruning algorithm*, 10th Annual ACM Symposium on Theory of Computing, 1978
- [38] F. Esposito et al., *A comparative analysis of methods for pruning decision trees*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, 1997, pp. 476-491
- [39] D. Patil, *Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy*, International Journal of Computer Applications, Vol. 11, 2010
- [40] B. Efron et al., *An Introduction to the Bootstrap*, Boca Raton, FL: Chapman & Hall/CRC, 1993
- [41] A. Zoubir et al., *Bootstrap Methods and Applications*, IEEE Signal Processing Magazine, Vol. 24, 2007, pp. 10-19

-
- [42] N. Adnan, *Improving the random forest algorithm by randomly varying the size of the bootstrap samples*, IEEE IRI 2014
- [43] T. Lee et al., *Bootstrap Aggregating and Random Forest*, Macroeconomic Forecasting in the Era of Big Data, 2019, pp. 389-429
- [44] S. Kotsiantis et al., *Bagging Model Trees for Classification Problems*, 10th PCI, 2005
- [45] P. Buhlmann, *Bagging, Boosting and Ensemble Methods*, Handbook of Computational Statistics, 2011, pp. 985-1022
- [46] H. Drucker et al., *Boosting Decision Trees*, 1996
- [47] S. Fafalios et al., *Gradient Boosting Trees*, Gnosis Data Analysis PC, 2020
- [48] L. Zhao et al., *Decision Tree Application to Classification Problems with Boosting Algorithm*, Electronics, MDPI, 2021
- [49] Y. Coadou, *Chapter 2: Boosted Decision Trees*, Artificial Intelligence for High Energy Physics, 2022, pp. 9-58
- [50] R. Santhanam et al., *Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets*, NCRISECT 2017
- [51] J. T. Hancock, T. Khoshgoftaar, *CatBoost for big data: an interdisciplinary review*, Journal of Big Data, Vol. 7, 2020
- [52] R. Schapire, *Explaining AdaBoost*, Empirical Inference, 2013
- [53] N. Figueroa, *Advanced Machine Learning Practical 3 : Classification (SVM, RVM & AdaBoost)*, 2017
- [54] Y. Coadou, *Boosted Decision Trees and Applications*, EPJ Web of Conferences, Vol. 55, 2013
- [55] A. Ashari et al., *Performance Comparison between Naive Bayes, Decision Tree and k-Nearest Neighbor (...)*, IJACSA, Vol. 4, Is. 11, 2013
- [56] G. Guo et al., *KNN Model-Based Approach in Classification*, Lecture Notes in Computer Science, Vol. 2888, Springer, Heidelberg, 2004
- [57] L. Wang, *Research and Implementation of Machine Learning Classifier Based on KNN*, Materials Science and Engineering, Vol. 677, Is. 5, 2019
- [58] S. Theodoridis, K. Koutroumbas, *Chapter 14 - Clustering Algorithms III: Schemes Based on Function Optimization*, Pattern Recognition, 2009, pp. 701-763
- [59] C. Hennig, *Minkowski distances and standardisation for clustering and classification of high dimensional data*, available online at <https://doi.org/10.48550/arXiv.1911.13272>, 2019
- [60] K. Taunk, *A Brief Review of Nearest Neighbor Algorithm for Learning and Classification*, ICCS 2019
- [61] L. Yu et al., *A Bias-Variance-Complexity Trade-Off Framework for Complex System Modeling*, ICCSA 2006
- [62] *Jupyter Notebook application documentation*, available online at <https://jupyter.org/>
- [63] P. Kopciwicz, *Github repository of kNN and DT comparison*, available at https://github.com/PawelKopciwicz/kNN_DT_comparison
- [64] *Sklearn Decision Tree Classifier documentation*, available online at <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
- [65] B. Krose, P. van der Smagt, *An introduction to neural networks*, Journal of Computer Science, Vol. 48, 1993
- [66] W. McCulloch, W. Pitts, *A Logical Calculus of Ideas Immanent in Nervous Activity*, Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp. 115-133
- [67] M. Minsky et al., *Perceptrons: An Introduction to Computational Geometry*, The MIT Press, 2017 (1968)

- [68] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain*, Psychological Review, Vol. 65, 1958, pp. 386-408
- [69] J. Feng, S. Lu, *Performance Analysis of Various Activation Functions in Artificial Neural Networks*, Journal of Physics Conference Series, 2019
- [70] T. Szandala, *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*, Bio-inspired Neurocomputing, 2021
- [71] A. Maniatopoulos, N. Mitianoudis, *Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function*, Information, Vol. 12, 2021
- [72] A. M. Pretorius et al., *ReLU and sigmoidal activation functions*, FAIR 2019
- [73] K. Janocha, W. Czarnecki, *On Loss Functions for Deep Neural Network in Classification*, Schedae Informaticae, Vol. 25, 2016, pp. 49-59
- [74] D. Wilson et al., *The need for small learning rates on large problems*, IJCNN 2002
- [75] A. Senior et al., *An empirical study of learning rates in deep neural networks for speech recognition*, IEEE ICASSS 2013
- [76] K. Nakamura, *Learning-Rate Annealing Methods for Deep Neural Networks*, Electronics, Vol. 10, 2021
- [77] A. Amini et al., *Spatial Uncertainty Sampling for End to End Control*, Workshop on Bayesian Deep Learning, NIPS 2017
- [78] Q. Wang et al., *A Comprehensive Survey of Loss Functions in Machine Learning*, Annals of Data Science, Vol. 9, 2022
- [79] Y. Tian et al., *Recent advances on loss functions in deep learning for computer vision*, Neurocomputing, Vol. 497, 2022, pp. 129-158
- [80] S. Jadon, *A survey of loss functions for semantic segmentation*, document available at <https://arxiv.org/abs/2006.14822>
- [81] A. Ruby et al., *Binary cross entropy with deep learning technique for Image classification*, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 4, 2020
- [82] S. Bose, M. Huber, *Training Neural Network with Policy Gradient*, IJCNN 2017
- [83] S. Hochreiter, *The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions*, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, Vol. 6, 1998, pp. 107-116
- [84] H. Tan, K. Lim, *Vanishing Gradient Mitigation with Deep Learning Neural Network Optimization*, ICSCC 2019
- [85] G. Philipp et al., *The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions*, document available online at <https://arxiv.org/abs/1712.05577>, 2017
- [86] R. Pascanu et al., *Understanding the exploding gradient problem*, available at <https://arxiv.org/abs/1211.5063v1>, 2012
- [87] Y. Hu et al., *Overcoming the vanishing gradient problem in plain recurrent networks*, available at <https://arxiv.org/abs/1801.06105>, 2018
- [88] Z. Hu et al., *Handling Vanishing Gradient Problem Using Artificial Derivative*, IEEE Access, Vol. 9, 2021, pp. 22371 - 22377
- [89] L. Miralles et al., *A methodology based on Deep Learning for advert value calculation in CPM, CPC and CPA networks*, Soft Computing, Vol. 21, 2017, pp. 651-665
- [90] M. Uzair, N. Jamil, *Effects of Hidden Layers on the Efficiency of Neural Networks*, INMIC 2020
- [91] K. Ahamed, S. Akthar, *A Study on Neural Network Architectures*, Computer Engineering and Intelligent Systems, Vol. 7, No. 9, 2016
- [92] K. Khalil et al., *An Efficient Approach for Neural Network Architecture*, ICECS 2019

-
- [93] F. Fan et al., *A New Type of Neurons for Machine Learning*, International Journal for Numerical Methods in Biomedical Engineering, Vol 34. Is. 2, 2017
- [94] J. Barzilai, *Chapter 15 - On neural-network training algorithms*, Human-Machine Shared Context, 2020, pp. 307-313
- [95] P. O. Glauner, *Comparison of Training Methods for Deep Neural Networks*, available at <https://arxiv.org/pdf/1504.06825.pdf>
- [96] A. Muhammad et al., *Survey on Training Neural Networks*, International Journal of Software Engineering and Knowledge Engineering, Vol. 5, Is. 3, 2015, pp. 169-173
- [97] H. Larochelle et al., *Exploring Strategies for Training Deep Neural Networks*, Journal of Machine Learning Research, Vol. 1, 2009, pp. 1-40
- [98] A. Farizawani, *A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches*, Journal of Physics: Conference Series, Vol. 1529, 2020, 022040
- [99] S. Yamani et al., *Optimizing back-propagation gradient for classification by an artificial neural network*, American Journal of Physics and Applications, Vol. 2, Is. 4, 2014, pp. 88-94
- [100] K. O'Shea, R. Nash, *An Introduction to Convolutional Neural Networks*, available at <https://arxiv.org/abs/1511.08458?context=cs>
- [101] R. Schmidt, *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*, available at <https://arxiv.org/abs/1912.05911>
- [102] S. Hochreiter et al., *Long Short-Term Memory*, Neural Computation, Vol. 9, 1997, pp. 1735 - 1780
- [103] K. Cho et al., *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*, available at <https://arxiv.org/abs/1409.1259>, 2014
- [104] S. Du et al., *Gradient Descent Finds Global Minima of Deep Neural Network*, Proceedings of Machine Learning, Vol. 97, 2019, pp. 1675-1685
- [105] A. Ashok, A. Garlapati, *Stochastic Gradient Descent for Deep Learning*, available at <https://www.andrew.cmu.edu/user/anubhava/bettersgd.pdf>
- [106] D. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization*, available at <https://arxiv.org/abs/1412.6980>
- [107] M. Khachay et al., *Mathematical Optimization Theory and Operations Research (...)*, Machine Learning and Big Data Processing, 2019
- [108] K. Blackmore et al., *Local Minima and Attractors at Infinity for Gradient Descent Learning Algorithms*, Journal of Mathematical Systems, Vol. 6, 1970
- [109] S. Sen, A. Kumar, *4 - Optimization I: Brute Force and Greedy Strategy*, Design and Analysis of Algorithms, 2019
- [110] M. Melanie, *An Introduction to Genetic Algorithms*, A Bradford Book, the MIT Press, 1999
- [111] A. Shukla et al., *Comparative Review of Selection Techniques in Genetic Algorithm*, IEEE ICFT-CAKM 2015
- [112] S. Podvalny et al., *The Crossover Operator of a Genetic Algorithm as Applied to the Task of a Production Planning*, Procedia Computer Science, Vol. 150, 2019, pp. 603-608
- [113] A. Sharma, M. Sinha, *Influence of crossover and mutation on the behavior of Genetic algorithms in Mobile Ad-hoc Networks*, ICCSGD 2014
- [114] I. De Falco et al., *Mutation-based genetic algorithm: Performance evaluation*, Applied Soft computing, Vol. 1, 2002, pp. 285-299
- [115] P. Kora et al., *Crossover Operators in Genetic Algorithms: A Review*, International Journal of Computer Applications, Vol. 162, 2017

- [116] V. Kumar et al., *A Study of Crossover Operators (...) New Sinusoidal Motion Crossover Operator*, International Journal of Computational Intelligence Research, Vol. 13, 2017, pp. 1717-1733
- [117] D. Goldberg et al., *A Comparative Analysis of Selection Schemes Used in Genetic Algorithms*, Foundations of Genetic Algorithms, Vol. 1, 1991, pp. 69-93
- [118] S. Katoch et al., *A review on genetic algorithm: past, present and future*, Multimedia Tools and Applications, Vol. 80, 2020, pp. 8091-8126
- [119] K. Opara, J. Arabas, *Comparison of mutation strategies in Differential Evolution - A probabilistic perspective*, Swarm and Evolutionary Computation, 2018, pp. 53-69
- [120] R. Mallipeddi et al., *Differential evolution algorithm with ensemble of parameters and mutation strategies*, Applied Soft Computing, Vol. 11, Is. 2, 2011, pp. 1679-1696
- [121] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975
- [122] Z. Wang, A. Sobey, *A comparative review between Genetic Algorithm use in composite optimisation and the state-of-the-art in evolutionary computation*, Composite Structures, Vol. 233, 2020
- [123] C. Blum et al., *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*, ACM Computing Surveys, Vol. 35, 2001, pp. 268-308
- [124] X. Yang, *Nature-Inspired Optimization Algorithms*, 2020
- [125] X. Yang, *Nature-Inspired Algorithms and Applied Optimization*, 2018
- [126] A. Halim et al., *Performance assessment of the metaheuristic optimization algorithms: an exhaustive review*, Artificial Intelligence Review, Vol. 54, 2021, pp. 2323-2409
- [127] W. Wong et al., *A Review on Metaheuristic Algorithms: Recent Trends, Benchmarking and Applications*, ICSCC 2019
- [128] K. Sorensen, *Metaheuristics—the metaphor exposed*, International Transactions In Operational Research, 2013
- [129] X. Yang, *Metaheuristic Optimization: Algorithm Analysis and Open Problems*, Lecture Notes in Computer Science, Vol. 6630, 2011
- [130] D. Molina et al., *Comprehensive Taxonomies of Nature- and Bio-inspired Optimization (...)*, Cognitive Computation, Vol. 12, 2020, pp. 897-939
- [131] F. Peres et al., *Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development*, Applied Sciences, Vol. 11, 2021
- [132] M. El-Shorbagy et al., *Particle Swarm Optimization from Theory to Applications*, International Journal of Rough Sets and Data Analysis, Vol. 5, 2018
- [133] S. Lukasik, S. Zak, *Firefly Algorithm for Continuous Constrained Optimization Tasks*, ICCCI 2009
- [134] X. Yang, *Chapter 11 - Flower Pollination Algorithm*, Nature-Inspired Optimization Algorithms, 2014, pp. 155-173
- [135] D. Henderson et al., *The Theory and Practice of Simulated Annealing*, Handbook of Metaheuristics, 2006, pp. 287-319
- [136] M. Dorigo et al., *Ant colony optimization*, IEEE Computational Intelligence Magazine, Vol. 1, 2006
- [137] R. Storn, K. Price, *Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, Journal of Global Optimization, Vol. 11, 1997, pp. 341 - 359
- [138] R. Storn, *On the usage of differential evolution for function optimization*, Proceedings of North American Fuzzy Information Processing, 1996
- [139] Bilal et al., *Differential Evolution: A review of more than two decades of research*, Engineering Applications of Artificial Intelligence, Vol. 90, 2020
- [140] S. Hassan. et al., *Multi-variant differential evolution algorithm for feature selection*, Scientific Reports, Vol. 90, 2020

-
- [141] M. A. Bakar et al., *A Customized Differential Evolutionary Algorithm for Bounded Constrained Optimization Problems*, Complexity, 2021
- [142] X. Zhong, P. Cheng, *An Improved Differential Evolution Algorithm Based on Dual-Strategy*, Mathematical Problems in Engineering, 2020
- [143] V. Kachitvichyanukul, *Comparison of Three Evolutionary Algorithms: GA, PSO, and DE*, Industrial Engineering and Management System, Vol. 11, 2012, pp. 215-223
- [144] U. Mlakar et al., *Multi-Objective Differential Evolution for feature selection in Facial Expression Recognition systems*, Expert Systems with Applications, Vol. 89, 2017, pp. 129-137
- [145] X. Zhong et al., *Ranking-based hierarchical random mutation in differential evolution*, PLOS ONE, Vol. 16, 2021
- [146] T. Wang et al., *Adaptive Dynamic Disturbance Strategy for Differential Evolution Algorithm*, MDPI Applied Science, 2020
- [147] C. Lu et al., *Enhanced Differential Evolution Based on Adaptive Mutation and Wrapper Local Search Strategies for Global Optimization Problems*, JART, Vol. 12, 2014, pp. 1131-1143
- [148] U. Skvorc et al., *CEC Real-Parameter Optimization Competitions: Progress from 2013 to 2018*, IEEE CEC 2019, pp. 3126-3133
- [149] R. Tanabe et al., *Evaluating the performance of SHADE on CEC 2013 benchmark problems*, IEEE CEC 2013, pp. 1952-1959
- [150] R. Tanabe et al., *Improving the search performance of SHADE using linear population size reduction*, IEEE CEC 2014, pp. 1658-1665
- [151] *LSHADE for IEEE WCCI 2018 optimization competition*, available at <https://github.com/dmolina/shadeils>
- [152] I. Goodfellow et al., *Generative adversarial nets*, Advances in Neural Information Processing Systems, 2014, pp. 2672-2680
- [153] Z. Wang et al., *Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy*, ACM Computing Surveys, Vol. 54, 2021
- [154] M. Gupta et al., *Chapter 1 - Super-resolution-based GAN for image processing: Recent advances and future trends*, Generative Adversarial Networks for Image-to-Image Translation, 2021, pp. 1-15
- [155] M. Bakay et al., *BSCGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks*, IEEE ICIP 2018
- [156] N. Souly et al., *Semi Supervised Semantic Segmentation Using Generative Adversarial Network*, IEEE ICCV, 2017, pp. 5689-5697
- [157] G. de Rosa, J. Papa, *A survey on text generation using generative adversarial networks*, Pattern Recognition, Vol. 119, 2021, 108098
- [158] A. Solanki et al., *Generative Adversarial Networks for Image-to-Image Translation*, 2021
- [159] A. Aggarwal et al., *Generative adversarial network: An overview of theory and applications*, International Journal of Information Management Data Insights, Vol. 1, 2021
- [160] M. Arjovsky et al., *Wasserstein GAN*, available at <https://arxiv.org/abs/1701.07875>
- [161] C. Frogner et al., *Learning with a Wasserstein Loss*, NIPS 2015, available at <https://arxiv.org/abs/1506.05439>
- [162] L. Rueschendorf, *The Wasserstein distance and approximation theorems*, Probability Theory and Related Fields, Vol. 70, 1985, pp. 117-129
- [163] A. Jolicoeur-Martineau, I. Mitliagkas, *Gradient penalty from a maximum margin perspective*, 2019, available at <https://doi.org/10.48550/arXiv.1910.06922>

- [164] J. Zhu et al., *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, 2017, available at <https://doi.org/10.48550/arXiv.1703.10593>
- [165] A. Radford et al., *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, available at <https://doi.org/10.48550/arXiv.1511.06434>
- [166] R. D. Hjelm et al., *Boundary-Seeking Generative Adversarial Networks*, 2017, available at <https://doi.org/10.48550/arXiv.1702.08431>
- [167] R. Sutton et al., *Policy Gradient Methods for Reinforcement Learning with Function Approximation*, Advances in Neural Information Processing Systems, Vol. 12, 2000
- [168] D. Berthelot et al., *BEGAN: Boundary Equilibrium Generative Adversarial Networks*, 2017, available at <https://arxiv.org/abs/1703.10717>
- [169] P. Isola et al., *Image-to-Image Translation with Conditional Adversarial Networks*, 2016, available at <https://arxiv.org/abs/1611.07004>
- [170] C. Hesse, *Pix2Pix online demo drawing*, available at <https://affinelayer.com/pixsrv/>
- [171] M. Mirza et al., *Conditional Generative Adversarial Nets*, 2014, available at <https://arxiv.org/abs/1410.8516>
- [172] Z. Pan et al., *Recent Progress on Generative Adversarial Networks (GANs): A Survey*, IEEE Access, Vol. 7, 2019
- [173] A. Aggarwal, *Generative adversarial network: An overview of theory and applications*, International Journal of Information Management, Vol. 1, 2021
- [174] A. Vaswani et al., *Attention Is All You Need*, NIPS 2017, available at <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [175] H. Zhang et al., *Self-Attention Generative Adversarial Networks*, Proceedings of Machine Learning Research, Vol. 97, 2019, pp. 7354-7363
- [176] D. Bau et al., *GAN dissection: Visualizing and understanding generative adversarial networks*, ICLR 2019
- [177] H. Huang et al., *An Introduction to Image Synthesis with Generative Adversarial Networks*, 2018, available at <https://arxiv.org/pdf/1803.04469.pdf>
- [178] A. Brock et al., *Large Scale GAN Training For High Fidelity Natural Image Synthesis*, ICLR 2019
- [179] M. S. Sabae et al., *StyleTF2: Generating Human Faces From Textual Description Using StyleGAN2*, 2022, available at <https://arxiv.org/pdf/2204.07924v1.pdf>
- [180] B. Kryhanovsky et al., *On the Probability of Finding Local Minima in Optimization Problems*, IEEE IJCNN 2006
- [181] G. Morse, K. O. Stanley, *Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks*, GECCO 2016, pp. 477-484
- [182] A.E. Eiben, J. Smith, *From Evolutionary Computation to the Evolution of Things*, Nature, Vol. 521, 2015, pp. 476-482
- [183] D. J. Montana, L. Davis, *Training feedforward neural networks using genetic algorithms*, IJCAI 1989, Vol. 1, 1989, pp. 762-767
- [184] R. S. Sexton et al., *Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation*, Decision Support Systems, Vol. 22, 1998, pp. 171-185
- [185] J. N. D. Gupta, R. S. Sexton, *Comparing backpropagation with a generic algorithm for neural network training*, Omega, Vol. 27, 1999, pp. 679-684
- [186] H. Huang et al., *A proposed iteration optimization approach integrating backpropagation neural network with genetic algorithm*, Expert Systems with Applications, Vol. 42, 2015, pp. 146-155

-
- [187] Haviluddin, R. Alfred, *A genetic-based backpropagation neural network for forecasting in time-series data*, ICSIT 2015, pp. 158-163
- [188] J. Gill et al., *Training back propagation neural networks with genetic algorithm for weather forecasting*, IEEE 8th ISISI 2010
- [189] C. Chandre Gowda, S. G. Mayya, *Comparison of Back Propagation Neural Network and Genetic Algorithm Neural Network for Stream Flow Prediction*, Journal of Computational Environmental Sciences, 2014
- [190] F. Yin et al., *A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters*, Materials & Design, Vol. 32, 2011, pp. 3457-3464
- [191] M. Hu, *Optimizing Back Propagation Neural Network with Genetic Algorithm for Man-hour Prediction in Chemical Equipment Design*, Chemical Engineering Transactions, Vol. 66, 2018, pp. 877-882
- [192] F.P. Such et al., *Deep Neuroevolution: Genetic algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning*, IEEE Transactions on Artificial Intelligence, 2017
- [193] Y. Sun et al., *Completely automated CNN architecture design based on blocks*, IEEE Transactions on Neural Networks and Learning Systems, Vol. 31, 2020, pp. 1242-1254
- [194] D. Floreano et al., *Neuroevolution: from Architectures to Learning*, Evolutionary Intelligence, Vol. 1, 2008, pp. 47-62
- [195] E. Galvan, P. Mooney, *Neuroevolution in Deep Neural Networks: Current Trends and Future Challenges*, IEEE Transactions on Artificial Intelligence, 2021
- [196] T. Elsken et al., *Neural Architecture Search*, Springer International Publishing, Cham, 2019, pp. 63-77
- [197] H. Kitano, *Designing neural networks using genetic algorithms with graph generation system*, Complex Systems, Vol. 4, 1990
- [198] R. Miiikkulainen et al., *Evolving deep neural networks*, available at <https://arxiv.org/abs/1703.00548>
- [199] E. Real et al., *Regularized evolution for image classifier architecture search*, AAAI 2019, AAAI Press, pp. 4780-4789
- [200] E. Real et al., *Large-scale evolution of image classifiers*, ICML17, Vol. 70, 2017, pp. 2902-2911
- [201] L. Xie, A. Yuille, *Genetic CNN*, ICCV 2017, pp. 1388-1397
- [202] E. Dufourq, B.A. Bassett, *EDEN: Evolutionary Deep Networks for Efficient Machine Learning*, PRASA-RobMech 2017, pp. 110-115
- [203] A. Martin et al., *Evodeep: A new evolutionary approach for automatic deep neural networks parametrisation*, Journal of Parallel and Distributed Computing, Vol. 117, 2018, pp. 180-191
- [204] Y. Sun et al., *Evolving deep convolutional neural networks for image classification*, IEEE Transactions on Evolutionary Computation, Vol. 24(2), 2020, pp. 394-407
- [205] C. Wang et al., *Evolutionary Generative Adversarial Networks*, IEEE Transactions on Evolutionary Computation, Vol. 23, 2019, pp. 921-934
- [206] J. Toutouh et al., *Spatial evolutionary generative adversarial networks*, GECCO 2020, pp. 1824-1831
- [207] V. F. Costa et al., *Neuroevolution of Generative Adversarial Networks*, Deep Neural Evolution, 2020, pp. 293-322
- [208] C. Coello, *Multi-objective Optimization*, Handbook of Heuristics, 2018, pp. 1-28
- [209] M. Baiocchi et al., *Multi-objective evolutionary GAN*, Deep Neural Evolution, 2020, pp. 293-322
- [210] C. He et al., *Evolutionary Multiobjective Optimization Driven by Generative Adversarial Networks (GANs)*, IEEE Transactions on Cybernetics, Vol. 51, no. 6, 2021, pp. 3129-3142

- [211] A. Jabbar et al., *A Survey on Generative Adversarial Networks: Variants, Applications, and Training*, ACM Computing Surveys, Vol. 65, Is. 8, 2022, pp. 1-49
- [212] F. Filthaut, *The LHCb Physics Programme*, 34th Rencontres de Moriond on QCD and High Energy Hadronic Interactions, 1999, pp. 409-414
- [213] I. Belyaev et al., *The history of LHCb*, The European Physical Journal H, Vol. 46, 2021
- [214] R. Aaij et al., *LHCb Collaboration*, Nuclear Physics A, Vol. 1005, 2021
- [215] R. Aaij et al., *Observation of a Narrow Pentaquark State, $P_c(4312)^+$, and of the Two-Peak Structure of the $P_c(4450)^+$* , Phys. Rev. Lett., Vol. 122, 2019
- [216] R. Aaij et al., *Observation of CP Violation in Charm Decays*, Phys. Rev. Lett., Vol. 122, 2019
- [217] The LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report*, CERN-LHCC-2012-007, 2012
- [218] T. Virdee, *Beyond the standard model of particle physics*, Philosophical Transactions A, Vol. 374, 2016
- [219] The LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report*, CERN, CERN-LHCC-2012-007, 2012
- [220] The LHCb collaboration, *High-Luminosity Large Hadron Collider (HL-LHC) - Preliminary Design Report*, CERN-2015-005, 2015
- [221] R. Alia et al., *LHC and HL-LHC: Present and Future Radiation Environment in the High-Luminosity Collision Points and RHA Implications*, IEEE Transactions on Nuclear Science, Vol. 65, 2017, pp. 448-456
- [222] L. Evans, P. Bryant, *LHC Machine*, Journal of Instrumentation, Vol. 3 2008
- [223] The ATLAS collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, Journal of Instrumentation, Vol. 3, 2008
- [224] The ALICE collaboration, *The ALICE experiment at the CERN LHC*, Journal of Instrumentation, Vol. 3, 2008
- [225] The CMS collaboration, *The CMS experiment at the CERN LHC*, Journal of Instrumentation, Vol. 3, 2008
- [226] P. Mouche, *Overall view of the LHC. Vue d'ensemble du LHC*, available at <https://cds.cern.ch/record/1708847>
- [227] M. Vanadia, *Beyond-the-Standard Model Higgs Physics using the ATLAS Experiment*, Proceedings of Science, Vol. 44, 2015
- [228] A. Tricomi et al., *The LHCf experiment at the LHC: Physics Goals and Status*, Nuclear Physics B - Proceedings Supplements, Vol. 196, 2009, pp. 30-35
- [229] G. Anelli et al., *The TOTEM experiment at the CERN large hadron collider*, Journal of Instrumentation, Vol. 3, 2008
- [230] J. Pinfold, *The MOEDAL experiment at the LHC - a progress report*, Universe, Vol. 5, 2019
- [231] J. Feng et al., *Letter of Intent: FASER - Forward Search Experiment at the LHC*, CERN-LHCC-2018-030, 2018
- [232] *Experiments at CERN*, available at <https://home.cern/science/experiments>
- [233] The LHCb collaboration, *The LHCb Detector at the LHC*, Journal of Instrumentation, Vol. 3, 2008
- [234] The LHCb collaboration, *Physics case for an LHCb Upgrade II - Opportunities in flavour physics, and beyond, in the HL-LHC era*, CERN-LHCC-2018-027, 2018
- [235] *CERN Bulletin*, Issue 50/2004, 2004, available at <https://cds.cern.ch/record/808300>
- [236] *The LHCb experiment website*, available at <https://home.cern/science/experiments/lhcb>

-
- [237] The LHCb collaboration, *LHCb Inner Tracker: Technical Design Report*, CERN-LHCC-2002-029, 2002
- [238] The LHCb collaboration, *LHCb Outer Tracker: Technical Design Report*, CERN-LHCC-2001-024, 2001
- [239] C. Elsasser, *The LHCb Silicon Tracker*, Journal of Instrumentation, Vol. 9, 2013
- [240] G. A. Cowan, the LHCb Silicon Tracker group, *Performance of the LHCb silicon tracker*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 699, 2013, pp. 156-159
- [241] R. Aaij et al., *Performance of the LHCb Vertex Locator*, Journal of Instrumentation, Vol. 9, 2014
- [242] The LHCb collaboration, *LHCb VELO (Vertex LOcator): Technical Design Report*, CERN-LHCC-2001-011, 2001
- [243] J. Alimena et al., *Searching for long-lived particles beyond the Standard Model at the Large Hadron Collider*, Journal of Physics G: Nuclear and Particle Physics, Vol. 47, 2020
- [244] E. Buchanan, *The LHCb Vertex Locator (VELO) Pixel Detector Upgrade*, Journal of Instrumentation, Vol. 12, 2017, C01013
- [245] P. R. Perez, *The LHCb Vertex Locator performance and Vertex Locator upgrade*, Journal of Instrumentation, Vol. 7, 2012, C12008
- [246] *Beetle Documentation*, available at <http://www.kip.uni-heidelberg.de/lhcb/index.html>
- [247] M. Agari et al., *Beetle - a radiation hard readout chip for the LHCb experiment*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 518, 2004, pp. 468-469
- [248] S. Lochner et al., *The Beetle Reference Manual - chip version 1.3, 1.4 and 1.5*, LHCb Technical Note, CERN-LHCb-2005-105, 2006
- [249] N. van Bakel et al., *Performance of the Beetle Readout Chip for LHCb*, In proceedings of: 8th Workshop on Electronics for LHC Experiments, 2002, available at <https://cds.cern.ch/record/1744659>
- [250] M. Agari et al., *Beetle - a radiation hard readout chip for the LHCb experiment*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 518, 2004, pp. 468-469
- [251] The LHCb collaboration, *RICH Technical Design Report*, CERN-LHCC-2000-037, 2000
- [252] S. Gambetta, the LHCb RICH group, *The LHCb RICH detectors: Operations and performance*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 952, 2020, 161882
- [253] The LHCb RICH group, *Performance of the LHCb RICH detector at the LHC*, The European Physical Journal C, Vol. 73, 2013
- [254] The LHCb collaboration, *LHCb calorimeters : Technical Design Report*, CERN-LHCC-2000-036, 2000
- [255] F. Machefert, *First year of running of the LHCb calorimeter system*, Physics Procedia, Vol. 37, 2012, pp. 421-428
- [256] Y. Guz, *The LHCb Hadron Calorimeter*, Journal of Physics: Conference Series, Vol. 160, 2009, 012054
- [257] *The LHCb electromagnetic calorimeter pictures*, available at <http://cds.cern.ch/record/835712>
- [258] I. Machikhiliyan, *Current status and performance of the LHCb electromagnetic and hadron calorimeters*, Journal of Physics: Conference Series, Vol. 293, 2011, 012052

- [259] The LHCb collaboration, *LHCb muon system: Technical Design Report*, CERN-LHCC-2001-010, 2001
- [260] A. Sharma, *Multiwire Gaseous Detectors: Basics and State-of-the-art*, Department of Physics, University of Maryland, US, 2000
- [261] T. Head, *The LHCb trigger system*, Journal of Instrumentation, Vol. 9, 2014, C09015
- [262] F. Legger et al., *TELL1: development of a common readout board for LHCb*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 535, 2004, pp. 497-499
- [263] The LHCb collaboration, *Trigger schemes (Online)*, available at <https://lhcb.web.cern.ch/speakersbureau/html/TriggerScheme.html>
- [264] J. Albrecht et al., *Performance of the LHCb High Level Trigger in 2012*, Journal of Physics: Conference Series, Vol. 513, 2014, 012001
- [265] E. Michielin, *The LHCb trigger in Run II*, Proceedings in: Incontri di Fisica delle Alte Energie - IFAE 2016, 2017
- [266] A. Trisovic, *Data preservation and reproducibility at the LHCb experiment at CERN*, Newnham College, University of Cambridge, CERN-THESIS-2018-174, 2018
- [267] The LHCb collaboration, *Framework TDR for the LHCb Upgrade: Technical Design Report*, CERN-LHCC-2012-007, 2012
- [268] R. Aaij et al., *Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC*, Journal of Instrumentation, Vol. 14, P04013
- [269] The LHCb collaboration, *LHCb Trigger and Online Technical Design Report*, CERN-LHCC-2014-016, 2014
- [270] F. Alessio, R. Jacobsson, *System-level Specifications of the Timing and Fast Control system for the LHCb Upgrade*, CERN Technical Note, CERN-LHCb-PUB-2012-001, 2012
- [271] F. Alessio et al., *A 40 MHz Trigger-free Readout Architecture for the LHCb Experiment*, Proceedings of the Topical Workshop on Electronics for Particle Physics, 2009, pp. 514-519
- [272] M. Bellato et al., *A PCIe Gen3 based readout for the LHCb upgrade*, Journal of Physics Conference Series, Vol. 513, 2014, 012023
- [273] The LHCb collaboration, *LHCb Tracker Upgrade Technical Design Report*, CERN-LHCC-2014-001, 2014
- [274] T. Kim, *SciFi - A large scintillating fibre tracker for LHCb*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 845, 2017, pp. 481-485
- [275] R. Aaij et al., *Allen: A High-Level Trigger on GPUs for LHCb*, Computing and Software for Big Science, Vol. 4, no. 7, 2020
- [276] R. Quagliani, *The LHCb Upgrade, Study of Double Charm B Decays with the LHCb Experiment at CERN and Track Reconstruction for the LHCb Upgrade*, Springer Theses, Springer, 2018, pp. 67-84
- [277] C. A. Beteta et al., *The SALT - Readout ASIC for Silicon Strip Sensors of Upstream Tracker in the Upgraded LHCb Experiment*, Sensors 22, 107, 2022
- [278] A. Massafferri, *SciFi, the new Tracker of the LHCb experiment*, Journal of Instrumentation, Vol. 15, 2020, C08006
- [279] The LHCb Scintillating Fibre Collaboration, *LHCb Scintillating Fibre Tracker Engineering Design Review Report: Fibres, Mats and Modules*, CERN Technical Note, LHCb-PUB-2015-008, 2015

- [280] C. Trippel et al., *Microlens-enhanced silicon photomultiplier arrays for LHCb SciFi Tracker Upgrade Ib*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 1040, 2022
- [281] J.M. de Cos et al., *Pacific: Sipm readout ASIC for lhcb upgrade*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 912, 2018, pp. 354-358
- [282] M. Fiorini, *The LHCb RICH Detector Upgrade*, Proceedings of Science, Vol. 314, 2018
- [283] Y. Guz, *LHCb Calorimeter Upgrade*, In Proceedings of: International Conference on Calorimetry for the High Energy Frontier, 2013, pp. 355-362
- [284] A. Cardini, *The LHCb Muon Upgrade*, Journal of Instrumentation, Vol. 9, 2014, C02014
- [285] The LHCb collaboration, *LHCb VELO Upgrade Technical Design Report*, CERN-LHCC-2013-021, 2013
- [286] A. F. Prieto et al., *Phase I Upgrade of the Readout System of the Vertex Detector at the LHCb Experiment*, IEEE Transaction on Nuclear Science, Vol. 67, Is. 4, 2020, pp. 732-739
- [287] K. Akiba et al., *Radiation Damage Effects and Operation of the LHCb Vertex Locator*, IEEE Transactions on Nuclear Science, Vol. 65, 2018, pp. 1127-1132
- [288] D. Dossett et al., *Radiation damage in the LHCb VELO*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 718, 2013, pp. 312-313
- [289] A. Affolder et al., *Radiation damage in the LHCb vertex detector*, Journal of Instrumentation, Vol. 8, 2013, P08002
- [290] T. Kohriki et al., *First observation of thermal runaway in the radiation damaged silicon detector*, IEEE Transactions on Nuclear Science, Vol. 43, 1996, pp. 1200-1202
- [291] G. Beck et al., *Analytic model of thermal runaway in silicon detectors*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 618, 2010, pp. 131-138
- [292] J. Allison et al., *Geant4 developments and applications*, IEEE Transactions on Nuclear Research, Vol. 53, 2006, pp. 270-278
- [293] A. Ferrari et al., *FLUKA: A multi-particle transport code*, CERN Yellow Reports: Monographs, CERN, 2005, pp. 1-405
- [294] M. Alexander et al., *Mapping the material in the LHCb vertex locator using secondary hadronic interactions*, Journal of Instrumentation, Vol. 13, 2018, P06008
- [295] R. Steinhagen, *LHC Beam Stability and Feedback Control*, available at <http://cds.cern.ch/record/1054849>
- [296] M. Kraan, *Part 10 (Sept 27 - Oct 2) of the production of the LHCb RF Box A-3*, Nikhef, available at <https://www.youtube.com/watch?v=ny67NR-8FiE>
- [297] M. Kraan, *Installation Upgrade RF Boxes (Full Version)*, Nikhef, available at https://www.youtube.com/watch?v=e5_r0_eTEtc
- [298] A. Nomerotski et al., *Evaporative CO₂ cooling using microchannels etched in silicon for the future LHCb vertex detector*, Journal of Instrumentation, Vol. 8, 2013, P04004
- [299] O. Francisco et al., *Microchannel cooling for the LHCb VELO Upgrade I*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2022, 166874
- [300] T. Poikela et al., *The VeloPix ASIC*, Journal of Instrumentation, Vol. 12, 2017, C01070

- [301] R. Ballabriga et al., *An introduction to the Medipix family ASICs*, Radiation Measurements, Vol. 136, 2020, 106271
- [302] X. Llopart et al., *Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 581, 2007, pp. 485-494
- [303] G. Crevatin et al., *Timepix3 readout system for time resolved experiments at synchrotron radiation facilities*, In proceedings of: 2016 IEEE NSS/MIC/RTSD, 2017
- [304] A. F. Prieto, P. V. Regueiro, *A Terabit Readout System for the LHCb VELO Upgrade I*, In proceedings of: 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC), 2018
- [305] A. F. Prieto et al., *Phase I Upgrade of the Readout System of the Vertex Detector at the LHCb Experiment*, IEEE Transactions on Nuclear Science, Vol. 67, no. 4, 2020
- [306] P. Moreira et al., *The GBT Project*, In proceedings of: The Topical Workshop on Electronics for Particle Physics, 2009, pp. 342-346
- [307] P. Moreira et al., *The GBT: A proposed architecture for multi-GB/s data transmission in high energy physics*, In proceedings of: Topical Workshop on Electronics for Particle Physics, 2007, pp. 332-336
- [308] F. Alessio, R. Jacobsson, *A new readout control system for the LHCb upgrade at CERN*, Journal of Instrumentation, Vol. 7, 2012, C11010
- [309] M. van Beuzekom et al., *VeloPix ASIC development for LHCb VELO upgrade*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 731, 2013, pp. 92-96
- [310] P. Leitao et al., *Test bench development for the radiation Hard GBTX ASIC*, Journal of Instrumentation, Vol. 10, 2015, C01038
- [311] S. Naik, *On-detector electronics for the LHCb VELO upgrade*, Journal of Instrumentation, Vol. 12, 2017, C02031
- [312] T. Poikela et al., *VeloPix: the pixel ASIC for the LHCb upgrade*, Journal of Instrumentation, Vol. 10, 2015, C01057
- [313] F. Krummenacher, *Pixel detectors with local intelligence: an IC designer point of view*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 305, 1991, pp. 527-532
- [314] M. George et al., *Xilinx, Linear Feedback Shift Registers in Virtex Devices*, document available online at <https://docs.xilinx.com/v/u/en-US/xapp210>
- [315] L. Eklund et al., *The VELO optical and power board*, LHCb Technical Note, CERN-LHCb-PUB-2021-012, 2021
- [316] R. Jacobsson et al., *The final LHCb readout supervisor "ODIN"*, In proceedings of: 8th Workshop on Electronics for LHC Experiments, 2002
- [317] F. Alessio, R. Jacobsson, *A new readout control system for the LHCb upgrade at CERN*, Journal of Instrumentation, Vol. 7, 2012, C11010
- [318] V. F. Lima, *LHCb Vertex Locator Upgrade Development and Rare b-quark Decays in LHCb*, Doctoral thesis, University of Liverpool, 2019
- [319] K. Akiba et al., *LHCb VELO Timepix3 telescope*, Journal of Instrumentation, Vol. 14, 2019, P05026
- [320] D. Hynds, *The Timepix telescope for charged particle tracking*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 730, 2013, pp. 50-53

- [321] K. Akiba et al., *The Timepix Telescope for high performance particle tracking*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 723, 2013, pp. 47-54
- [322] E. Dall’Occo, *The Timepix3 Telescope and Sensor R&D for the LHCb VELO Upgrade*, In proceedings of: 2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC), 2018
- [323] E. Buchanan, *Spatial Resolution Studies for the LHCb VELO Upgrade*, Doctoral Thesis, University of Bristol, UK, 2018
- [324] X. Llopart et al., *Timepix4, a large area pixel detector readout chip which can be tiled on 4 sides providing sub-200 ps timestamp binning*, Journal of Instrumentation, Vol. 17, 2022, C01044
- [325] K. Heijhoff et al., *Timing performance of the Timepix4 front-end*, Journal of Instrumentation, Vol. 17, 2022, P07006
- [326] B. van der Heijden et al., *SPIDR, a general-purpose readout system for pixel ASICs*, Journal of Instrumentation, Vol. 12, 2017, C02040
- [327] J. Visser et al., *SPIDR: a read-out system for Medipix3 & Timepix3*, Journal of Instrumentation, Vol. 10, C12028
- [328] *VC707 Evaluation Board for the Virtex-7 FPGA, User Guide*, 2019, available at https://www.xilinx.com/support/documents/boards_and_kits/vc707/ug885_VC707_Eval_Bd.pdf
- [329] G. Barrand et al., *GAUDI - A Software Architecture and Framework for building HEP Data Processing Applications*, Computer Physics Communications, Vol. 140, 2001, pp. 45-55
- [330] R. Currie et al., *Evolution of the LHCb Continuous Integration System*, EPJ Web of Conferences, Vol. 245, 2020
- [331] *Gaudi, LHCb Data Processing Application Framework*, Users Gaudi release v9, 2001
- [332] *The BRUNEL Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/brunel/>
- [333] *The BOOLE Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/boole/>
- [334] *The DAVINCI Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/davinci/>
- [335] *The MOORE Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/moore/>
- [336] *The Moore’s documentation*, available at <https://www.physi.uni-heidelberg.de/~mstahl/moore/>
- [337] *The GAUSS Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/gauss/>
- [338] M. Clemencic et al., *The LHCb Simulation Application, Gauss: Design, Evolution and Experience*, Journal of Physics: Conference Series, Vol. 331, Is. 3, 2011, 032023
- [339] G. Corti et al., *Software for the LHCb experiment*, IEEE Transactions on Nuclear Science, Vol. 53, 2006, pp. 1323 - 1328
- [340] K. Hennessy, *Calibration and Performance Monitoring of the LHCb Vertex Locator*, Journal of Physics: Conference Series, Vol. 396, 2012
- [341] T. Szumlak, C. Parkes, *Description of the Vetra Project and its Application for the VELO Detector*, LHCb Technical Note, CERN-LHCb-2008-022
- [342] M. Majewski, *Development of the LHCb VELO Monitoring Software Platform*, Acta Physica Polonica B, Vol. 48, 2017
- [343] M. Clemencic, B. Couturier, *A New Nightly Build System for LHCb*, IOP Publishing, Journal of Physics: Conference Series, Vol. 153, 2014, 052007
- [344] A. Dendek, *Calibration and monitoring of the SALT readout ASIC for the LHCb UT detector*, 22th Cracow EPIPHANY Conference on the Physics in LHC Run2, 2016

- [345] B. Rachwal, *A modular software framework for test-beam data analysis*, EPJ Web of Conferences, Vol. 214, 2019, 05023
- [346] P. Kopciewicz et al., *Vetra - Run 3 and 4 calibration software for LHCb silicon detectors*, LHCb Technical Report, 2022
- [347] W. Krupa, *Towards first observation of the $B_s^0 \rightarrow D_s^{*\mp} K^{*\pm}$ decay and calibration of the Upstream Tracker detector*, doctoral thesis, AGH-UST, 2022
- [348] *A GitLab repository of Vetra project*, available at <https://gitlab.cern.ch/lhcb/Vetra>
- [349] S. Ponce, *The LHCb-Starterkit Team, The DevelKit LHCb Run3 materials*, available at <https://lhcb.github.io/DevelopKit/03a-gaudi/>
- [350] *A GitLab repository of lb-stack-setup, the LHCb stack*, available at <https://gitlab.cern.ch/rmatev/lb-stack-setup>
- [351] *The LBCOM Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/lbcom/>
- [352] M. Majewski et al., *Calibration software platform for LHCb upgrade silicon detectors: STORCK and TITANIA*, CERN LHCb Technical Note, CERN-LHCb-PUB-2021-007, 2021
- [353] M. Majewski, *STORCK database documentation*, available at https://storck.docs.cern.ch/quick_start.html
- [354] SIMATIC WinCC Open Architecture Portal, *Product Information and Documentation*, available at <https://www.winccoa.com/>
- [355] L. Granado Cardoso et al., *A Framework for Hardware Integration in the LHCb Experiment Control System*, In proceedings of: ICALEPCS2015, 2015
- [356] M. Adinolfi et al., *LHCb data quality monitoring*, Journal of Physics: Conference Series, Vol. 898, 2017, 092027
- [357] S. Redford, *LHCb VELO Closing Control, Vertex Resolution and Luminosity Measurement*, Proceedings of Science, Vol. 113, 2011
- [358] F. Lazzari et al., *Real-time cluster finding for LHCb silicon pixel VELO detector using FPGA*, Journal of Physics Conference Series, Vol. 1525, 2020, 012044
- [359] C. Gaspar et al., *DIM, a portable, light weight package for information publishing, data transfer and inter-process communication*, Computer Physics Communications, Vol. 140, 2000, pp. 102-109
- [360] *DIM, Distributed Information Management System*, available at <http://dim.web.cern.ch/>
- [361] M. Frank et al., *Data stream handling in the LHCb experiment*, Journal of Physics: Conference Series, Vol. 119, 2008, 022024
- [362] *Experiment Control System, The LHCb PVSS Framework*, available at <http://lhcb-online.web.cern.ch/ecs/pvssintro.htm>
- [363] *PVSS - DIM Integration*, available at https://lhcb-online.web.cern.ch/ecs/fw/fw_dim_description.html
- [364] K. Hennessy et al., *Readout Firmware of the Vertex Locator for LHCb Run 3 and Beyond*, IEEE Transactions on Nuclear Science, Vol. 68, No. 10, 2021
- [365] C. Bozzi, S. Roiser, *Towards a computing model for the LHCb Upgrade*, The European Physical Journal Conferences, Vol. 214, 2019, 03045
- [366] P. Kopciewicz, *Daily Commissioning Software TWiki for VELO Upgrade I*, available at <https://lbtwiki.cern.ch/bin/view/VELO/VeloUpCommissioningSoft>
- [367] J. Barbosa et al., *Front-End Electronics Control and Monitoring for the LHCb Upgrade*, EPJ Web of Conferences, Vol. 214, 2019
- [368] P. Kopciewicz, *Storck object definition, DIM header documentation (the commissioning format)*, available at https://codimd.web.cern.ch/-LDkHKFaQHyp7Ld3rB11_Q#

-
- [369] C. Bartella et al., *VELO channels and cabling nomenclature*, LHCb Technical Note, EDMS No. 2397053, CERN, 2020
- [370] *An introduction to LHCb Software, LHCb Starterkit Lessons*, available at <https://lhcb.github.io/starterkit-lessons/first-analysis-steps/davinci.html>
- [371] *Timing and Fast Control System*, available at <https://lhcb-online.web.cern.ch/tfc/default.html>
- [372] E. Calvo et al., *The LHC Orbit and Trajectory System*, Proceedings DIPAC, 2003
- [373] *Raw Data Format for Run3 and ODIN bank*, LHCb Technical Note, available at <https://edms.cern.ch/ui/#!/master/navigator/document?D:100350539:100350539:subDocs>
- [374] A. F. Prieto, P. V. Regueiro, *Back-end firmware for the LHCb VELO upgrade phase I*, Proceedings of Science, Vol. 370, 2020
- [375] A. Prieto, K. Hennessy, *LHCb Upgrade VELO Front-End Data Format*, LHCb Technical Note, 2018
- [376] *The LHCb Project*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/lhcb/>
- [377] G. Giacomini, *Noise Characterization of Silicon Strip Detectors Comparison of Sensors with and without Integrated JFET Source-Follower*, doctoral thesis, 2007, available at <https://cds.cern.ch/record/2093586>
- [378] G. Barrand et al., *Abstract Interfaces for Data Analysis: Component Architecture for Data Analysis Tools*, In proceedings of: Computing in High Energy and Nuclear Physics, China, 2001, available at <https://cds.cern.ch/record/519004>
- [379] L. Granado Cardoso et al., *LHCb MiniDAQ control system*, EPJ Web of Conferences, Vol. 214, 2019, 01005
- [380] *PyDIM documentation*, available at <http://lhcbdoc.web.cern.ch/lhcbdoc/pydim/guide/index.html>
- [381] *STORCK - VELO Calibration Software GitLab repository*, available at <https://gitlab.cern.ch/velo-calibration-software/storck>
- [382] M. Majewski, *Application of machine learning methods for the analysis of the calibration of the LHCb VELO detector, studies of irradiated silicon pixel sensors and reconstruction of the neutrino interaction in LARTPC detector*, doctoral thesis, AGH-UST, 2022 (ongoing)
- [383] M. Adinolfi et al., *LHCb data quality monitoring*, Journal of Physics: Conf. Series 898 092027, 2017
- [384] R. Geertsema et al., *Charge collection properties of prototype sensors for the LHCb VELO upgrade*, Journal of Instrumentation, Vol. 16, 2021, P02029
- [385] K. Akiba et al., *Charged particle tracking with the Timepix ASIC*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 661, 2012, pp. 31-49
- [386] M. Rizzi et al., *Semiconductor Detectors and Principles of Radiation-matter Interaction*, Journal of Applied Sciences, Vol. 10, 2010, pp. 3141-3155
- [387] X. Cudie, *Design and characterization of 64K pixels chips working in single photon processing mode*, doctoral thesis, Sweden, 2007, available at cds.cern.ch/record/1056683
- [388] Z. He, *Review of the Shockley–Ramo theorem and its application in semiconductor gamma-ray detectors*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 463, 2001, pp. 250-267
- [389] L. Landau, *On the energy loss of fast particles by ionization*, J. Phys. (USSR), Vol. 8, 1944, pp. 201-205

- [390] M. Campbell et al., *Study of the charge sharing in a silicon pixel detector by means of α -particles interacting with a Medipix2 device*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 591, 2008, pp. 38-41
- [391] H. Driver et al., *Quantitative Expression of Cultural Relationships*, University of California Archaeology and Ethnology, Vol. 31 (4), 1932, pp. 211-256
- [392] A. Ezugwu et al., *A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects*, Engineering Applications of Artificial Intelligence, Vol. 110, 2022, 104743
- [393] P. Kaur et al., *A survey of clustering techniques and algorithms*, In Proceedings of: 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015
- [394] D. Xu et al., *A Comprehensive Survey of Clustering Algorithms*, Annals of Data Science, Vol. 2, 2015, pp. 165–193
- [395] R. Liu et al., *Variational Inference for Deblending Crowded Starfields*, 2021, available at <https://arxiv.org/abs/2102.02409>
- [396] Y. Zhang et al., *Crowded Cluster Cores: An Algorithm for Deblending in Dark Energy Survey Images*, Publications of the Astronomical Society of the Pacific, Vol. 127, 2015
- [397] E. Bertin, *SE-xtractor User Manual*, available at <https://sextractor.readthedocs.io/en/latest/>
- [398] J. Jakubek, *Precise energy calibration of pixel detector working in time-over-threshold mode*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 633, 2011, pp. 262-266
- [399] T. Panagiotis, *A silicon pixel detector for LHCb*, doctoral thesis, available at <https://cds.cern.ch/record/2238509>
- [400] A. Tehrani, *Test-beam measurements and simulation studies of thin pixel sensors for the CLIC vertex detector*, ETH Zurich, 2017
- [401] S. Tsigaridas et al., *Timewalk correction for the Timepix3 chip obtained with real particle data*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 930, 2019, pp. 185-190
- [402] S. Schramm, *Machine Learning at CERN: ATLAS, LHCb, and more*, Proceedings of Science, Vol. 340, 2019
- [403] C. Weisser, *The Search for Dark Photons at LHCb and Machine Learning in Particle Physics*, doctoral thesis, Massachusetts Institute of Technology, 2021, available at <https://dspace.mit.edu/handle/1721.1/142688>
- [404] L. Oliveira, *Jet-images - deep learning edition*, Journal of High Energy Physics, 2016
- [405] C. Shimmin et al., *Decorrelated jet substructure tagging using adversarial neural networks*, Phys. Rev. D, Vol. 96, 2017, 074034
- [406] A. Larkoski et al., *Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning*, Physics Reports, Vol. 841, 2020, pp. 1-63
- [407] The ATLAS collaboration, *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment*, ATLAS Phys. Pub., 2017, available at <http://cds.cern.ch/record/2255226>
- [408] M. Borisyak et al., *Towards automation of data quality system for CERN CMS experiment*, Journal of Physics: Conference Series, Vol. 898, 2017, 092041
- [409] CMS collaboration, *New Developments for Jet substructure Reconstruction in CMS*, CERN-CMS-DP-2017-027, available at <https://cds.cern.ch/record/2275226/>

-
- [410] CMS collaboration, *Performance of b tagging algorithms in proton-proton collisions at 13 TeV with Phase 1 CMS detector*, CERN-CMS-DP-2018-033, available at <https://cds.cern.ch/record/2627468>
- [411] *Machine Learning for Jet Physics workshop (ML4Jets)*, available at <https://indico.physics.lbl.gov/indico/event/546/> for 2017, <https://indico.cern.ch/event/745718/> for 2018 and <https://indico.cern.ch/event/980214/> for 2021
- [412] *Higgs Boson Machine Learning Challenge competition*, available at <https://www.kaggle.com/c/higgs-boson>
- [413] *MLTracks Kaggle competition*, available at <https://www.kaggle.com/competitions/trackml-particle-identification/overview>
- [414] *2018 IEEE World Congress on Computational Intelligence (IEEE WCCI 2018)*, plenary talks available at <https://ieeetv.ieee.org/event/ieee-wcci-2018>
- [415] *Yandex MLHEP 2019*, materials available at <https://github.com/yandexdataschool/mlhep2019>
- [416] R. Mankel, *Pattern Recognition and Event Reconstruction in Particle Physics Experiments*, Reports on Progress in Physics, Vol. 67 (4), 2004
- [417] B. Graham et al., *3D Semantic Segmentation with Submanifold Sparse Convolutional Networks*, available at <https://arxiv.org/abs/1711.10275>
- [418] L. Domine et al., *Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data*, 2019, available at <https://arxiv.org/abs/1903.05663>
- [419] G. Plastiras, *EdgeNet: Balancing Accuracy and Performance for Edge-based Convolutional Neural Network Object Detectors*, ICSDC 2019: Proceedings of the 13th International Conference on Distributed Smart Cameras, 2019, pp. 1-6
- [420] A. Tsaris et al., *The HEP.TrkX Project: Deep Learning for Particle Tracking*, Journal of Physics: Conference Series, Vol. 1085 (4), 2018, 042023
- [421] *The HEP.TrkX project*, website and documentation, available at <https://heptrkx.github.io/>
- [422] S. Farrell et al., *Novel deep learning methods for track reconstruction*, 2018, available at <https://arxiv.org/abs/1810.06111>
- [423] P. Komiske et al., *Energy flow networks: deep sets for particle jets*, Journal of High Energy Physics, 2019
- [424] M. De Cian, *Machine learning and parallelism in the reconstruction of LHCb and its upgrade*, EPJ Web of Conferences, Vol. 127, 2016
- [425] L. Anderlini, *Machine Learning for the LHCb Simulation*, 2021, available at <https://arxiv.org/abs/2110.07925>
- [426] Y. Wang, *Dynamic Graph CNN for Learning on Point Clouds*, available at <https://arxiv.org/abs/1801.07829>
- [427] P. Battaglia et al., *Relational inductive biases, deep learning, and graph networks*, 2018, [?]
- [428] J. You et al., *GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models*, 2018, available at <https://arxiv.org/abs/1802.08773>
- [429] M. Sajjadi et al., *Assessing Generative Models via Precision and Recall*, 2018, available at <https://arxiv.org/abs/1806.00035>
- [430] T. Fawcett, *An introduction to ROC analysis*, Pattern Recognition Letters, Vol. 27 (8), 2006, pp. 861-874
- [431] A. Bradley, *The use of the area under the ROC curve in the evaluation of machine learning algorithms*, Pattern Recognition, Vol. 30, 1997, pp. 1145-1159

- [432] M. Scholz et al., *Nonlinear Principal Component Analysis: Neural Network Models and Applications*, Principal Manifolds for Data Visualization and Dimension Reduction, pp. 44–67
- [433] A. Maevskiy et al., *Fast Data-Driven Simulation of Cherenkov Detectors Using Generative Adversarial Networks*, 2019, available at <https://arxiv.org/abs/1905.11825>
- [434] G. Khattak et al., *Three Dimensional Energy Parametrized Generative Adversarial Networks for Electromagnetic Shower Simulation*, In proceedings of: 2018 25th IEEE ICIP, 2018
- [435] G. Khattak et al., *Fast simulation of a high granularity calorimeter by generative adversarial networks*, The European Physical Journal C, Vol. 82, 2022
- [436] D. Salamani et al., *Deep Generative Models for Fast Shower Simulation in ATLAS*, In proceedings of: 2018 IEEE 14th International Conference on e-Science, 2018
- [437] V. Belavin et al., *Electromagnetic shower generation with Graph Neural Networks*, Journal of Physics: Conference Series, Vol. 1525, 2020, 012105
- [438] N. Akchurin et al., *On the Use of Neural Networks for Energy Reconstruction in High-granularity Calorimeters*, Journal of Instrumentation, Vol. 16, 2021, P12036
- [439] V. Chekalina, *Generative Models for Fast Calorimeter Simulation: the LHCb case*, EPJ Web of Conferences, Vol. 214, 2019, 02034
- [440] N. Kazeev, *Machine Learning for Particle Identification in the LHCb Detector*, doctoral thesis, 2020
- [441] D. Derkach et al., *Machine Learning based Global Particle Identification Algorithms at the LHCb Experiment*, EPJ Web of Conferences, Vol. 214, 2019
- [442] D. Derkach et al., *Machine-Learning-based global particle-identification algorithms at the LHCb experiment*, Journal of Physics: Conference Series, Vol. 1085 (4), 2018, 042038
- [443] D. Hnida, *Online track reconstruction for the LHCb Upgrade using machine learning*, Verhandlungen der Deutschen Physikalischen Gesellschaft, 2019
- [444] The LHCb collaboration, *LHCb detector performance*, International Journal of Modern Physics A, Vol. 30, 2015, 1530022
- [445] L. Anderlini et al., *Computing strategy for PID calibration samples for LHCb Run 2*, CERN-LHCb-PUB-2016-020, 2016, available at <https://cds.cern.ch/record/2199780>
- [446] R. Aaij et al., *Selection and processing of calibration samples to measure the particle identification performance of the LHCb experiment in Run 2*, EPJ Techniques and Instrumentation, Vol. 6, 2019
- [447] A. Rashwan et al., *MatrixNets: A New Scale and Aspect Ratio Aware Architecture for Object Detection*, 2020, available at <https://arxiv.org/abs/2001.03194>
- [448] T. Likhomanenko, *LHCb Topological Trigger Reoptimization*, Journal of Physics: Conference Series, Vol. 664, 2015, 082025
- [449] J. Duarte et al., *Fast inference of deep neural networks in FPGAs for particle physics*, Journal of Instrumentation, Vol. 13, 2018, P07027
- [450] B. Hackenberger, *Tensors all around us*, Croat Med J., Vol. 60, 2019, pp. 369-374
- [451] *PyTorch, a tensor library for deep learning*, available at <https://pytorch.org/docs/stable/index.html>
- [452] *Keras, a deep learning library for Theano and Tensorflow*, available at <https://faroit.com/keras-docs/1.2.0/>
- [453] *Tensorflow, an end-to-end open source platform for machine learning*, available at <https://www.tensorflow.org/>
- [454] M. Abadi et al., *TensorFlow: A system for large-scale machine learning*, 2016, available at <https://arxiv.org/abs/1605.08695>

-
- [455] *TensorBoard, the visualization tool for machine learning*, available at <https://www.tensorflow.org/tensorboard>
- [456] K. Wongsuphasawat et al., *Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow*, IEEE Transactions on Visualization and Computer Graphics, Vol. 24, 2018
- [457] *Theano documentation page*, available at <https://theano-pymc.readthedocs.io/en/latest/#>
- [458] *PyTorch Lightning, the PyTorch research framework*, available at <https://www.pytorchlightning.ai/>
- [459] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, In proceedings of: NeurIPS 2019
- [460] D. Koutsoukos et al., *Tensors: An abstraction for general data processing*, Proceedings of the VLDB Endowment, Vol. 14, 2021, pp. 1797-1804
- [461] A. Hoecker et al., *TMVA - Toolkit for Multivariate Data Analysis*, Proceedings of Science, ACAT, 2007
- [462] K. Albertsson et al., *TMVA 4, Toolkit for Multivariate Data Analysis with ROOT*, Users Guide, CERN-OPEN-2007-007, 2020
- [463] *TMVA - Toolkit for Multivariate Data Analysis with ROOT*, available at <http://tmva.sourceforge.net/>
- [464] *ROOT - an open source programming framework*, documentation available at <https://root.cern/>
- [465] R. Brun et al., *History of the ROOT system: conception, evolution and experience*, Proceedings of Science, Vol. 93, 2011
- [466] R. Brun et al., *ROOT: An object oriented data analysis framework*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 389, 1997, pp. 81-86
- [467] R. Brun et al., *ROOT, an object oriented data analysis framework*, CERN School of Computing, 2000, pp. 11-41, available at <https://cds.cern.ch/record/491486>
- [468] I. Antcheva et al., *ROOT — A C++ framework for petabyte data storage, statistical analysis and visualization*, Computer Physics Communications, Vol. 180 (12), 2009, pp. 2499-2512
- [469] D. Piparo et al., *A Parallelised ROOT for Future HEP Data Processing*, EPJ Web of Conferences, Vol. 214, 2019, 05033
- [470] *Manual of TMVA - Toolkit for Multivariate Data Analysis with ROOT*, available at <https://root.cern/manual/tmva/>
- [471] K. Albertsson, *Machine Learning with ROOT/TMVA*, EPJ Web of Conferences, Vol. 2454, 2020, 06019
- [472] F. James, *MINUIT, Function Minimization and Error Analysis Reference Manual*, available at <https://root.cern.ch/download/minuit.pdf>
- [473] W. Verkerke, *RooFit*, EPJ Web of Conferences, Vol. 4, 2010
- [474] W. Verkerke, D. Kirkby, *RooFit Users Manual v2.91*, available at https://root.cern.ch/download/doc/RooFit_Users_Manual_2.91-33.pdf
- [475] S. Ramos et al., *Discriminant Function Analysis*, The Encyclopedia of Applied Linguistics, 2012
- [476] A. Eckert-Gallup et al., *Kernel Density Estimation (KDE) with adaptive bandwidth selection for environmental contours of extreme sea state*, In proceedings of: OCEANS 2016 MTS/IEEE Monterey, 2016
- [477] M. Galli et al., *A New PyROOT: Modern, Interoperable and More Pythonic*, EPJ Web of Conferences, Vol. 245, 2020, 06004

- [478] T. Skwarnicki, *A study of the radiative CASCADE transitions between the Upsilon-Prime and Upsilon resonances*, doctoral thesis, 1986
- [479] J. Magnus, *Gauss on least-squares and maximum-likelihood estimation*, Archive for History of Exact Sciences, Vol. 76, 2022, pp. 425-430
- [480] A. Piotrowski, J. Napiorkowski, *The grouping differential evolution algorithm for multi-dimensional optimization problems*, Control and Cybernetics, Vol. 39, 2010
- [481] C. Bishop, C. Roach, *Fast curve fitting using neural networks*, Review of Scientific Instruments, Vol. 63, 1992
- [482] *The IntelliStat github project*, available at <https://github.com/PawelKopciewicz/IntelliStat/tree/main>
- [483] A. Savitzky, M. Golay, *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*, Analytical Chemistry, Vol. 36, 1964, pp. 1627-1639
- [484] F. Nielsen, *Cramér-Rao lower bound and information geometry*, Texts and Readings in Mathematics, Vol. 67, 2013, pp. 18-37
- [485] G. McLachlan et al., *Mixture Models: Inference and Applications to Clustering*, Journal of the American Statistical Association, Vol. 84, 1989, pp. 337-338
- [486] D. Reynolds, *Gaussian Mixture Models*, Encyclopedia of Biometrics, 2009, pp. 659-663
- [487] K. Cranmer, *Kernel estimation in high-energy physics*, Computer Physics Communications, Vol. 136, 2001, pp. 198-207
- [488] *RooKeysPdf class reference*, available at <https://root.cern.ch/doc/master/classRooKeysPdf.html#details>
- [489] C. Bishop, *Pattern Recognition and Machine Learning*, 2006
- [490] X. Mao et al., *Least Squares Generative Adversarial Networks*, 2016, available at <https://doi.org/10.48550/arXiv.1611.04076>
- [491] C. Wu, *Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis*, Ann. Statist., Vol. 14, 1986, pp. 1261-1295
- [492] *The reconstruction particle mass spectra github project*, available at https://github.com/MichalKacprzak99/reconstruction_particle_mass_spectra
- [493] G. Cowan et al., *Rapidsim v1.4.2, documentation and software*. available at <https://zenodo.org/record/1068329#.YtLfrYTP1PY>
- [494] Z. Yi et al., *DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*, available at <https://arxiv.org/abs/1704.02510>
- [495] *PyTorch implementation of GANs*, available at <https://github.com/eriklindernoren/PyTorch-GAN>
- [496] R. L. Workman et al. (Particle Data Group), *Prog. Theor. Exp. Phys. 2022*, 2022, available at <https://pdg.lbl.gov/2022/listings/rpp2022-list-B-zero.pdf>
- [497] *Flower Pollination Algorithm by Kopciewicz and Lukasik, application for Gauss distribution*, available at https://github.com/PawelKopciewicz/FP_optimizer
- [498] *Mutations in GAN*, available at <https://github.com/FrightenedFox/GAN-projects>