Department of Physics and Astronomy
University of Heidelberg

Master Thesis in Physics
submitted by

Sven Hoppner

born in Heidelberg (Germany)

2023

Machine Learning Studies for the Sexaquark Search in ALICE using Boosted Decision Trees

This Master Thesis has been carried out by Sven Hoppner at the
Physikalisches Institut in Heidelberg
under the supervision of
Prof. Dr. Klaus Reygers

Abstract

This thesis investigates the feasibility of searching for the proposed six-quark state called sexaquark at ALICE. The sexaquark (S) proposed by Gennys R. Farrar has a quark content of *uuddss* and is a possible dark matter candidate. If the sexaquark exists, then the sexaquark and the anti-sexaquark $\bar{S}$ would be produced in high-energy heavy-ion collisions at the LHC. This work focuses on the development of a reconstruction chain for a $\bar{S}$ produced in Pb–Pb collisions at center-of-mass energies of 5.02 TeV with subsequent annihilation in the detector material of the ALICE detector. The interaction channel $\bar{S} + n \rightarrow \Lambda + K^0_S$ was studied in simulated events. A complete reconstruction chain for this channel was developed, including the definition of background reducing cuts and the development of a boosted decision tree classifier based on the XGBoost library. The simulations showed a reconstruction efficiency of 2.4 % for the investigated channel, demonstrating the effectiveness of this reconstruction approach. Finally, calculations of the expected number of detected sexaquarks in real data were made, based on theoretical considerations and efficiency measurements with the simulation. The theoretical assumptions include a interaction cross section of 5 mb, proposed by Gennys R. Farrar, and a production cross section similar to the deuteron. In addition, estimates of the expected background were made, which, together with the expected number of detected sexaquarks, led to an expected significance range for a possible sexaquark search in all recorded data between 0.47σ and 6.5σ. Further simulations are necessary to improve the accuracy of this range, and with improvements to the reconstruction workflow and expansion of the search to multiple channels, a discovery of the sexaquark in ALICE might be possible. This research highlights the significant role of machine learning in the quest for new particles. The developed reconstruction method, employing advanced machine learning techniques, shows promising potential for detecting the sexaquark in upcoming data taking periods, and opens doors for further advancements in particle physics research.

Zusammenfassung

In dieser Arbeit wird die Durchführbarkeit der Suche nach dem vorgeschlagenen Sechs-Quark-Zustand namens Sexaquark bei ALICE untersucht. Das von Gennys R. Farrar vorgeschlagene Sexaquark (S) hat einen Quarkgehalt von $uuddss$ und ist ein möglicher Kandidat für dunkle Materie. Wenn das Sexaquark existiert, dann würden das Sexaquark und das Anti-Sexaquark $\bar{S}$ in hochenergetischen Schwerionenkollisionen am LHC erzeugt werden. Diese Arbeit konzentriert sich auf die Entwicklung einer Rekonstruktionskette für ein $\bar{S}$, das in Pb–Pb-Kollisionen bei Schwerpunktsenergien von 5.02 TeV mit anschließender Annihilation im Detektormaterial des ALICE-Detektors erzeugt wird. Der Wechselwirkungskanal $\bar{S}+n \rightarrow \Lambda K^0_S$ wurde in simulierten Events untersucht. Es wurde eine vollständige Rekonstruktionskette für diesen Kanal entwickelt, einschließlich der Definition von untergrundreduzierenden Cuts und der Entwicklung eines Boosted-Decision-Tree-Klassifikators auf der Grundlage der XGBoost-Bibliothek. Die Simulationen ergaben eine Rekonstruktionseffizienz von 2.4 % für den untersuchten Kanal, was die Wirksamkeit dieses Rekonstruktionsansatzes beweist. Auf Grundlage theoretischer Überlegungen und Effizienzmessungen mit der Simulation wurden Berechnungen zu der erwarteten Anzahl von nachweisbaren Sexaquarks in realen Daten durchgeführt. Zu den theoretischen Annahmen gehören ein von Gennys R. Farrar vorgeschlagener Wechselwirkungsquerschnitt von 5 mb und ein Produktionsquerschnitt ähnlich dem des Deuterons. Zusätzlich wurden Abschätzungen des erwarteten Hintergrunds vorgenommen, die zusammen mit der erwarteten Anzahl an Sexaquarks zu einem erwarteten Signifikanzbereich für eine mögliche Suche in allen aufgezeichneten Daten zwischen $0,47\sigma$ und $6,5\sigma$ führten. Weitere Simulationen sind notwendig, um die Genauigkeit dieses Bereichs zu verbessern, und mit Verbesserungen des Rekonstruktions-Workflows und der Ausweitung der Suche auf mehrere Kanäle könnte eine Entdeckung des Sexaquarks in ALICE möglich sein. Diese Forschung unterstreicht die wichtige Rolle von Machine Learning bei der Suche nach neuen Teilchen. Die entwickelte Rekonstruktionsmethode mithilfe von Machine Learning, zeigt ein vielversprechendes Potenzial für die Entdeckung des Sexaquarks in den kommenden Datenerfassungsperioden und öffnet Türen für weitere Fortschritte in der Teilchenphysikforschung.

Contents

List of Figures

List of Tables

1 Introduction

It is a widely accepted fact that our universe contains more mass than we are currently able to observe [1] and the source of this mass is what we call Dark Matter (DM). The source and composition of this dark matter is one of the most fundamental problems in modern physics. Many theories try to propose a solution to the dark matter problem, and some of them argue that it could be some kind of new unknown or already known particle. Many possible candidates have been proposed, with the current frontrunner being the so-called Weakly Interacting Massive Particles (WIMP). As the name suggests, WIMPs are heavy particles with masses between $\mathcal{O}(10\,\mathrm{GeV}/c^2)$ and $\mathcal{O}(1000\,\mathrm{GeV}/c^2)$ that interact with matter only through the weak and gravitational force. Extensive searches for WIMPs have been conducted over the years, but no conclusive signal has yet been observed [2]. Due to the lack of evidence for WIMPs, the focus of dark matter searches is now shifting to more exotic candidates. One such candidate was proposed in 2017 by Glennys Farrar [3]. This hypothetical particle is called the "sexaquark" with the mathematical symbol S and corresponds to a new stable six-quark state. The sexaquark is consistent with our current understanding of Quantum Chronodynamics (QCD) and the expected relic abundance of dark matter. Previous searches for the sexaquark have yielded inconclusive results, neither proving that the sexaquark exists nor that it does not, such as a search at the BaBaR experiment at the Stanford Linear Accelerator Center near Stanford University [4], looking for the sexaquark in the decay products of Υ decays, and a search at the CMS at CERN, where the interaction products of a sexaquark produced in the LHC particle collisions and the detector material could be observed [5][6]. The inconclusiveness of these earlier attempts led to a new search for S, which is currently being carried out by members of the ALICE collaboration. The ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment) detector is one of four major experiments at CERN (Conseil Européen pour la Recherche Nucléaire, European Organisation for Nuclear Research), which currently operates the world's largest hadron collider, the LHC (Large Hadron Collider). The sexaquark, if it exists, is possible to be created during the high-energy collisions and to interact with the detector material and be

detected indirectly via its reaction products. The number of sexaquarks produced is larger, the higher the center-of-mass energy of the collision, as well as the mass and cross section of the colliding particles is. The LHC with its high luminosity and high center of mass energies especially with heavy ion collisions such as Pb–Pb is the ideal place for this search. The use of the ALICE detector for this search is favored due to its great tracking and particle identification capabilities compared to the other experiments at the LHC. A previous master's thesis performed at ALICE showed the feasibility of such an analysis [7] and is now followed by an extensive search with simulations of S combined with a machine learning approach. The scope of this work is to support this analysis and to improve the existing search methods by using machine learning methods to separate signal from background in Monte Carlo simulations of the sexaquark. A classifier based on XGBoost will be developed. Furthermore, calculations of the expected detected signal as well as estimates of the expected background are performed.

This work is divided into five chapters: Chapter 1 gives a short introduction to the Standard Model of particle physics and the theoretical proposals and search strategies for the sexaquark. Chapter 2 discusses the experimental setup with a short introduction to the ALICE detector as well as some explanations on the working principles of boosted decision trees. Chapter 3 discusses the analysis performed, including the search strategy, simulations, channel selection, applied cuts, as well as the implementation, optimization and classification results of the XGBoost classifier. Chapter 4 describes the computational and physical results of the signal and background estimation. Finally, Chapter 5 gives a conclusion of the thesis and an outlook on further improvements.

1.1 Standard Model of Particle Physics

The Standard Model of Particle Physics (SM) is a generally accepted and thoroughly tested theory describing the composition of matter as we know it. It has been studied and developed since the second half of the 20th century and is therefore able to explain almost all current experimental results. In the SM, matter is composed of fundamental particles called fermions, which are divided into two categories: quarks and leptons. Fermions are spin $\frac{1}{2}$ particles, which are subdivided into three generations, getting heavier with each subsequent generation. Each generation consists of

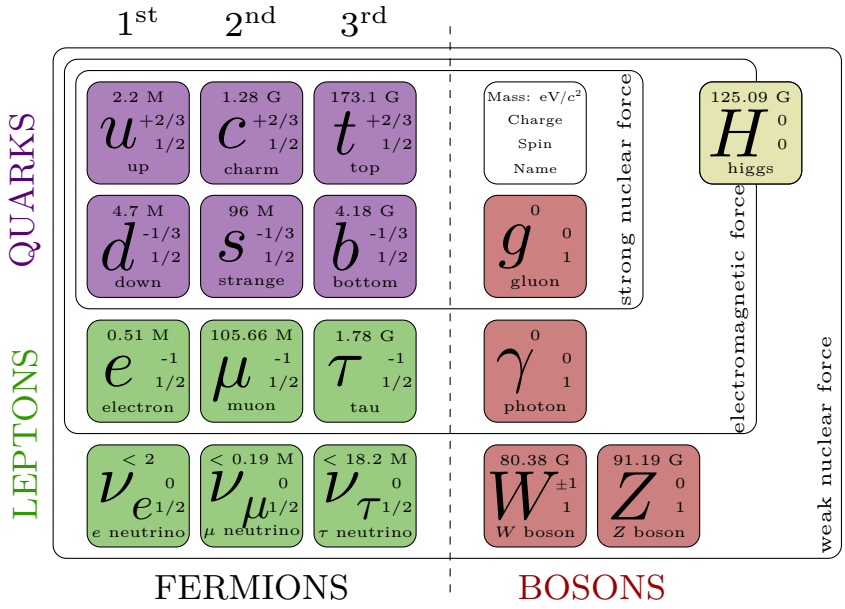


Figure 1.1: Elemental particles in the Standard Model of particle physics.

two particles, each with its respective antiparticle, which has the same mass, spin and lifetime, but opposite charge. An overview of the different elementary particles can be seen in Fig. 1.1. The three generations of leptons are the electron, muon and tau, each with its corresponding neutrino. Due to their neutral charge, neutrinos are suspected to be Majorana particles in theories beyond the SM, which are particles that are also their own antiparticle. Electrons and neutrinos have an infinite lifespan, while muons have a lifetime of 2.2×10^{-6} s and tau leptons have a lifetime of 2.9×10^{-13} s. Quarks are divided into up-type quarks, with a charge of $\frac{2}{3}$, namely up, charm and top quarks, and down-type quarks with a charge of $-\frac{1}{3}$, namely down, strange and bottom quarks. Quarks are bound to each other under normal energies and temperatures in a phenomenon called confinement. Together, quarks form hadrons, which can be classified into two groups: mesons and baryons. Mesons are whole-integer spin particles composed of one quark and one anti-quark, and baryons are half-integer spin particles composed of three quarks (three anti-quarks for anti-baryons) held together by the strong interaction. Interactions between particles occur through the exchange of particles called gauge bosons.In the SM, three of the four fundamental forces are explained by the exchange of the corresponding gauge boson. Photons are the exchange particles of the electromagnetic force, which acts on particles with an electric charge. The W and Z bosons are the exchange particles of the weak nuclear force, which is responsible for some particle decays,

such as β decay. The gluon is the exchange boson of the strong force, which affects both quarks and other gluons. The only fundamental force not included in the SM is gravity. The last addition to the SM was the Higgs boson in 2012 after its discovery at the LHC, which is responsible for most of the mass of fundamental particles. The electromagnetic and strong interactions are described by two theories. The theory that explains the electromagnetic force is called Quantum Electrodynamics (QED), and it describes how charged particles interact through the exchange of photons. Quantum Chromodynamics (QCD) is the corresponding field theory that describes the strong interaction. The strong force acts on quarks and gluons, which possess a special property called color charge, through the exchange of a gluon. Quarks can have one of three color charges, namely red, green and blue as well as corresponding anti-colors for anti-quarks. Hadrons, which are composed of quarks, must to be color neutral, where mesons are comprised of one color-anti-color pair and in baryons, each quark needs to have exactly one of the three colors. The interaction between quarks occurs between force-carrying gluons, which mediate the strong force. Gluons carry color charge as well and change the color of quarks by interacting with them. In total, there are eight different types of gluons, which are distinguished by the color charge they carry.

1.1.1 Exotic Hadrons

Apart from mesons, which consist of two quarks (one quark and one anti-quark), and baryons, which consist of three quarks, other hadrons with a quark content of more than three quarks have been proposed. These are called exotic hadrons. Since the postulation of the quark model in 1964 by Gell-Mann and Zweig independently, exotic hadrons were regarded as a possibility and are even mentioned as such in Gell-Mann's paper [8][9]. Many experiments have been searching for exotic hadrons with some being successful: the first particle proposed to be a tetraquark called $X(3872)$ was discovered by the Belle experiment in 2003 [10]. The observation of a tetraquark state candidate was announced in 2007 at the Belle experiment in Japan, with a $c\bar{c}d\bar{u}$ state called $Z(4430)$ [11]. The number in parentheses is the mass of the particle in MeV/c^2. Numerous tetraquarks were subsequently discovered: in June 2013, the BES III and Belle experiments in China and Japan, respectively, independently reported the $Z_C(3900)$ state [12][13] and in 2014, LHCb confirmed the existence of the $Z(4430)$ with a significance of 13.9σ [14]. Other tetraquarks discovered include

the $X(5568)$, $X(4274)$, $X(6900)$, $X(4500)$ and $X(4700)$ [15][16][17]. Searches for a possible pentaquark have been conducted since the mid-2000s but with no or questionable results [18][19] until 2015, when the LHCb collaboration at CERN identified two pentaquark states that are sometimes present as intermediate states in the decay of Λ_b^0. LHCb could verify those pentaquarks, $P_C^+(4380)$ and $P_C^+(4450)$, both with a quark content of $uudc\bar{c}$, with a significance of 9σ and 12σ respectively [20]. In 2019, LHCb also announced the discovery of the pentaquark $P_C^+(4312)$, with a significance of 5σ [21]. Particle states with more than five quarks have been proposed, but have not been discovered so far. Possible six-quark states include the H-dibaryon and the sexaquark proposed by R. Jaffe and G. Farrar respectively.

1.1.2 H-Dibaryon

Already 40 years ago, R. Jaffe proposed a possible *uuddss* state called the H-dibaryon [22]. It has quantum numbers $I = 0$ and $J^P = 0^+$ with spin $S = -2$ and baryon number $B = 2$. Jaffe suggested that it has a mass of $m_H \approx 2.15\,\mathrm{GeV}/c^2$, calculated using the MIT quark bag model [23]. Although Jaffe proposed it as a new stable particle at the time, its mass satisfies $m_H > m_\Lambda + m_p + m_e$ to make it unstable with a relatively short lifetime. The H-dibaryon is proposed to be a $\Lambda - \Lambda$ bound state, so a good reference here is the lifetime of a free Λ, which is of the order of $\mathcal{O}(10^{-10}\,\mathrm{s})$, but it may exceed this lifetime due to a binding energy of $B_H = 4.56 \pm 1.13_{\mathrm{stat}} \pm 0.63_{\mathrm{sys}}\,\mathrm{MeV}$ calculated using lattice QCD [24]. Therefore, a possible decay channel of the H-dibaryon is $H \to \Lambda + p + \pi^-$. Previous searches have been conducted at experiments around the world, including the Brookhaven National Laboratories (BNL) in the USA [25], Belle in Japan [26] and ALICE at CERN [27], without conclusive evidence for the existence of the H-dibaryon. It should be noted that these experiments had to focus on an H-dibaryon mass $m_H \geq 2\,\mathrm{GeV}/c^2$ to eliminate a possible neutron background. It is possible that instead of the loosely bound H-dibaryon state with a mass $m_H \geq 2\,\mathrm{GeV}/c^2$ that these experiments were trying to find, an existing *uuddss* state is actually strongly bound and stable with a mass lower than $2\,\mathrm{GeV}/c^2$.

1.1.3 Sexaquark

Following the experimental discoveries of four- and five-quark state particles in 2003 and 2015, in 2017 Glennys R. Farrar proposed a new particle in agreement with the Standard Model called the sexaquark [3]. It is proposed to be a composite of six strongly bound light quarks with a quark content of *uuddss*. Farrar reasons that this composite state of *uuddss* has a privileged status due to Fermi statistics, since it is the only combination of six light quarks for which the spatial wavefunction of S can be completely symmetric, while at the same time the spin, color, and flavor wavefunctions are completely antisymmetric. For other six quark states, spatial symmetry is not given, suggesting that S is the most tightly bound state of all the proposed six quark states. Due to its quark content, it has a neutral charge and is a boson with spin 0 with even parity and quantum numbers Q = 0, B = 2 and S = −2. It has a proposed mass $m_S \lesssim 2\,\mathrm{GeV}/c^2$ and depending on its exact mass it is either stable if $m_S < 2(m_p{+}m_e) = 1878\,\mathrm{MeV}/c^2$ or its lifetime τ_S is longer than the lifetime of the universe $\tau_{\mathrm{Univ.}}$ if $m_S < m_p{+}m_e{+}m_\Lambda = 2055\,\mathrm{MeV}/c^2$, since it could only decay via a doubly weak interaction, which would make it essentially stable. To confirm the mass of the sexaquark, lattice QCD calculations are needed, but they are still far away from realistic six-quark states [28]. On the other hand, using the constituent quark model to calculate the mass of the sexaquark as the sum of the effective masses of its quark constituents leads to a mass too high for stability of $2.1\,\mathrm{GeV}/c^2$. However, these calculations are not always applicable, since other hadron masses, such as pions, cannot be calculated in the same way. The possible stability of the sexaquark, combined with its neutral charge, makes it an ideal candidate for dark matter within the Standard Model, which is further discussed in section 1.3. Its mass, as well as its stability as a compact state, distinguishes it from the previously discussed H-dibaryon, which is why Farrar gave it the name sexaquark, denoted as S for “Sexaquark, Singlet, Scalar, Strong and Stable” [3]. Furthermore, previous searches for similar particles such as the H-dibaryon have severely disfavored or completely excluded states with masses above $2\,\mathrm{GeV}/c^2$ due to the high neutron background, which is a good explanation for why the S has eluded detection so far. In experimental setups, it bears a resemblance to the neutron and could be mistaken as such, which is why finding it requires a specialized search that explicitly looks for a neutral S = −2 particle. Possible search strategies and previous attempts are discussed in Chapter 1.2.

1.2 Proposed Search Strategies and Previous Searches for the Sexaquark

In her paper "A Stable Sexaquark: Overview and Discovery Strategies", Farrar suggests several possible strategies for discovering the sexaquark [29]. Her first proposed strategy is to look for a missing mass peak in the upsilon decay

$$\Upsilon \rightarrow \text{gluons} \rightarrow S\bar{\Lambda}\bar{\Lambda} \text{ or } \bar{S}\Lambda\Lambda + \text{pions and/or } \gamma. \tag{1.1}$$

Farrar argues that if every other final state particle has been detected and measured, a clear peak in the missing mass can be seen. She argues that only a few reconstructed events in this decay channel need to be reconstructed to provide conclusive evidence due to the high resolution of some up to state detectors on the order of $\mathcal{O}(20\,\text{MeV})$. A decay channel containing $\Lambda\Lambda/\bar{\Lambda}\bar{\Lambda}$ pairs is preferred because of the short decay length of Λ and the 64 % branching fraction in $p\pi^-$, resulting in high reconstruction efficiency of $\Lambda/\bar{\Lambda}$ and well measured 4 moments. Other decay channels in which the $\Lambda\Lambda$ or $\bar{\Lambda}\bar{\Lambda}$ pair could be replaced by $\Xi^- p$ or a single Λ which could be replaced by $K^- p$ could also be detected. Farrar notes that in principle any other combination of hyperons (baryons containing strange quarks and no charm, bottom or top quarks) and mesons with quantum numbers B= ± 2 and S= ∓ 2 would suffice, as long as no B- or S-bearing particle escapes detection. The first attempt to search for the sexaquark in Υ decays was made by the BABAR experiment in 2018 [30]. The sample studied consisted of 90×10^6 $\Upsilon(2S)$ and 110×10^6 $\Upsilon(3S)$ and the decay channel $\Upsilon \rightarrow S\bar{\Lambda}\bar{\Lambda}$ was examined. No experimental signal for the sexaquark was observed, but an upper limit with a 90 % confidence level on the combined $\Upsilon(2\text{S}, 3\text{S}) \rightarrow S\bar{\Lambda}\bar{\Lambda}$ branching ratio for $m_S < 2.05\,\text{GeV}/c^2$ was found to be in the range $(1.2 - 1.4) \times 10^{-7}$.

Farrar also mentions the possibility of detecting sexaquarks produced in hadronic collisions via characteristic decay chains after the annihilation of the $\bar{S}$ in the detector material. For this approach, she suggests searching at the LHC to take advantage of its high luminosity, since its estimated scattering cross section with a nucleon is rather small at $\sigma_{\text{SN}} \cdot \left(\frac{1}{4} - 1\right) \cdot \sigma_{\text{NN}}^{\text{el}} \approx (5 - 20)\,\text{mb}$. The annihilation reaction mentioned by Farrar is $\bar{S} + N \rightarrow \bar{\Xi}^{+,0} + X$, with $\bar{\Xi}^{+,0} \rightarrow \bar{\Lambda}\pi^{+,0}$ and $\bar{\Lambda} \rightarrow \bar{p}\pi^+$ or $\bar{S} + N \rightarrow \bar{\Lambda} + K_s^0 + X$. The resulting $\bar{S}$ should have a transverse momentum of $\langle p_T \rangle \lesssim \mathcal{O}(1\,\text{GeV}/c)$, which is similar to other hadrons. The advantage of this

$\bar{S} + n \rightarrow$	$\bar{S} + p \rightarrow$
$\bar{\Lambda} K^0$	$\bar{\Lambda} K^+$
$\bar{\Lambda} K^0 \pi^- \pi^+$	$\bar{\Lambda} K^+ \pi^- \pi^+$
$\bar{\Lambda} K^0 \pi^0 \pi^0$	$\bar{\Lambda} K^+ \pi^0 \pi^0$
$\bar{\Lambda} K^+ \pi^- \pi^0$	$\bar{\Lambda} K^0 \pi^+ \pi^0$
$\bar{p} K^0 K^0 \pi^+$	$\bar{p} K^+ K^+ \pi^0$
$\bar{p} K^0 K^+ \pi^0$	$\bar{p} K^+ K^0 \pi^+$
$\Xi^+ \pi^-$	

Table 1.1: Possible interaction channel of an anti-sexaquark with the detector material.

approach is that the observation of such a distinct production/decay chain provides unambiguous evidence that the interacted particle is a $B = -2$ and $S = +2$ neutral particle. The possible interaction channel with the detector material can be seen in Table 1.1.

There have been two attempts to search for the sexaquark at the CMS detector. First in 2018 by Florian Partous in his Master thesis on the feasibility of detecting the sexaquark at the CMS detector [5], where his thesis focused on the definition of background discriminating cuts and the search for an S-mass distribution bump in 150 million pp events. The result of this work was the estimation of an upper limit on the $\bar{S}$ production cross section with a 95 % confidence level, which was calculated to be $\sigma(pp \rightarrow \bar{S}) = 43\,\text{mb}$. For the calculation of the production cross section, the interaction cross section was assumed to be comparable to the inelastic neutron cross section $\sigma(\bar{S} + n)$. This upper limit was further updated in a subsequent search at CMS which is published in the Ph.D. thesis of Jarne de Clercq [6]. He obtained an upper limit on the product of production cross section and interaction cross section of $\sigma(pp \rightarrow \bar{S}) \times \sigma(\bar{S} + n \rightarrow K^0_S + \bar{\Lambda}^0) = 105^{+57.8}_{-32.4}\,\text{mb}^2$ with a 95 % confidence level. This search looks for the interaction products of sexaquarks produced in pp collisions, with the detector material. The interaction products are then reconstructed and after applying background reducing cuts, a boosted decision tree is used to separate the signal from background. This search was severely limited by the reconstruction efficiency of the sexaquark, which de Clercq reports to be of 0.0014 %. The low reconstruction efficiency, coupled with the low interaction probability of the sexaquark with the detector material could be the reason why no conclusive sexaquark signal was found. In this work, the search strategy of CMS is adopted with the intention to improve on the reconstruction efficiency, taking advantage of

the superior particle identification and tracking capabilities but suffering from lower luminosity.

Subsequently, a search for the sexaquark at ALICE was conducted as a master's thesis by Fabio Schlichtmann, who investigated the feasibility of detecting the sexaquark with the ALICE detector setup [7]. His thesis is the predecessor of this one and concluded that finding the sexaquark in ALICE would be challenging but possible. The investigation focused on the interaction of the S with the protons in the detector material rather than with the neutrons, and looked at detecting the interaction channels $\bar{S}+p \rightarrow \bar{p}+K^+ + K^0 + \pi^+$ and $\bar{S}+p \rightarrow \bar{\Lambda}+K^+ + \pi^- + \pi^+$. He concluded that of the order of $\mathcal{O}(10^1)$ to $\mathcal{O}(10^2)$ sexaquarks would be detectable within the investigated 2.17×10^8 Run 2 Pb–Pb events, but the reconstruction efficiency had to be estimated since the investigation was done without access to sexaquark simulations. With the simulations now available, a more accurate calculation of the expected number of sexaquarks should be possible, as well as an estimate of the expected background.

1.3 The dark matter candidate sexaquark

The search for a stable sexaquark is of particular interest because it could provide an explanation for dark matter (DM). For decades, Weakly Interacting Massive Particles (WIMPs) were the most favored explanation among possible dark matter candidates, but without any experimental evidence for their existence – despite extensive search efforts – the focus of dark matter research shifts to more exotic explanations, one of which is the sexaquark. By design, many of the criteria for a possible DM candidate are already met, such as stability. A dark matter candidate must be stable in the sense that the phenomenon that explains DM must have survived the time from the Big Bang to the present. As discussed earlier, the sexaquark is stable enough if its mass is below $m_S < m_p + m_e + m_\Lambda$, since its lifetime would be greater than the age of the universe. Furthermore, a suitable dark matter candidate had to evade any kind of detection until now, which for the sexaquark can be explained by its neutral charge, low interaction cross section and general similarity to neutrons in experimental settings. Therefore, the detection of the sexaquark requires an extensive large-scale specialized analysis on suitable experimental setups of the kind that have never been done before. In addition, a dark matter candidate

must match the observed DM relic density and the observed dark matter to baryon ratio $\Omega_{\mathrm{DM}}/\Omega_b = 5.3 \pm 0.1$ [31]. Farrar showed that a sexaquark in the mass range of 1860−1880 MeV/c^2 is perfectly consistent with this limit within 15 % and follows naturally from the Boltzmann distribution in the QGP with minimal assumptions for DM, which is equally composed of u,d and s quarks [32]. To preserve this relic abundance as the universe cools down requires that the decay rate of S in interactions between S and baryons, which occur via a Yukawa potential, is smaller than the expansion rate of the universe. For this to hold, the effective Yukawa vertex for decay is $g \lesssim$ few 10^{-6}, which is given by the low probability of fluctuations between sexaquark and dibaryon configurations [33]. Finally, a possible DM candidate has to be compatible with observed astrophysical phenomena. One of which is that the sexaquark must be compatible with the existence of neutron stars and supernovae. While some theoretical astrophysicists claim that a deeply bound *uuddss* state is incompatible with neutron stars, due to tensions with observations of hot proto-neutron stars, after the production of dibaryons from Λ baryons within the star [34], others claim that this dilemma can be resolved by quark deconfinement [35]. Both sides of the argument are based on speculations about the properties of a possible sexaquark and settling the debate will require proving either its existence and measuring its mass as well as cross sections or proving its non-existence. In either case, this thesis aims to contribute to this debate by providing tools and research that will bring us one step closer to the goal of detecting the sexaquark.

2 Experimental Setup and Used Tools

This Chapter discusses the experimental setup of the LHC and ALICE and a short introduction to boosted decision trees is also given.

2.1 CERN and the LHC

CERN (**C**onseil **E**uropéen pour la **R**echerche **N**ucléaire, European Organization for Nuclear Research) is one of the largest scientific organizations in the world, focused on deepening our current understanding of particle physics. CERN is located on the border between France and Switzerland, near Geneva on the Swiss side and Saint-Genis-Pouilly on the French side of the border. Most of its research is high-energy physics, in which particles are accelerated in several particle accelerators and brought to collision at specific points of interaction, where particle detectors are placed to measure the resulting particles. For this reason, construction of the world's largest particle accelerator, the Large Hadron Collider (LHC), began in 1994 and was completed in 2008. It has a circumference of 26.7 km and is housed about 175 m below the surface in a 3.8 m wide tunnel. A series of linear and ring accelerators such as the Proton Synchrotron (PS) and the Super Proton Synchrotron (SPS) are used to pre-accelerate the particles before they are injected into the LHC, where they reach center-of-mass energies of up to 13.6 TeV for proton-proton collisions and 5.02 TeV for lead–lead collisions. Along the beam pipe of the LHC there are four intersections where the four major experiments and their detectors are operating: ATLAS (**A** **T**oroidal **L**HC **A**pparatu**S**), the largest detector of the four, CMS (**C**ompact **M**uon **S**olenoid), LHCb (**L**arge **H**adron **C**ollider **b**eauty) and ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment) detector. A schematic view of the LHC and its main experiments at CERN can be seen in Fig. 2.1. The four experiments are used to study different aspects of particle and heavy ion collisions. ATLAS and CMS are both multipurpose detectors that excel in high p_{T} and high luminosity environments,

LHCb is specialized in detecting hadronic decays involving bottom or charm quarks. The ALICE collaboration is specifically focused on the study of the quark-gluon plasma (QGP), and excels in particle identification (PID) and tracking performance. A more detailed description of the ALICE detector can be found in the Chapter 2.2. The LHC has just finished the long shutdown 2 and many upgrades and repairs have been done on the different detectors. However, since calibration is still in progress, simulations anchored to Run 2 data were used for this thesis. Therefore, the upgrades will not be discussed in detail.

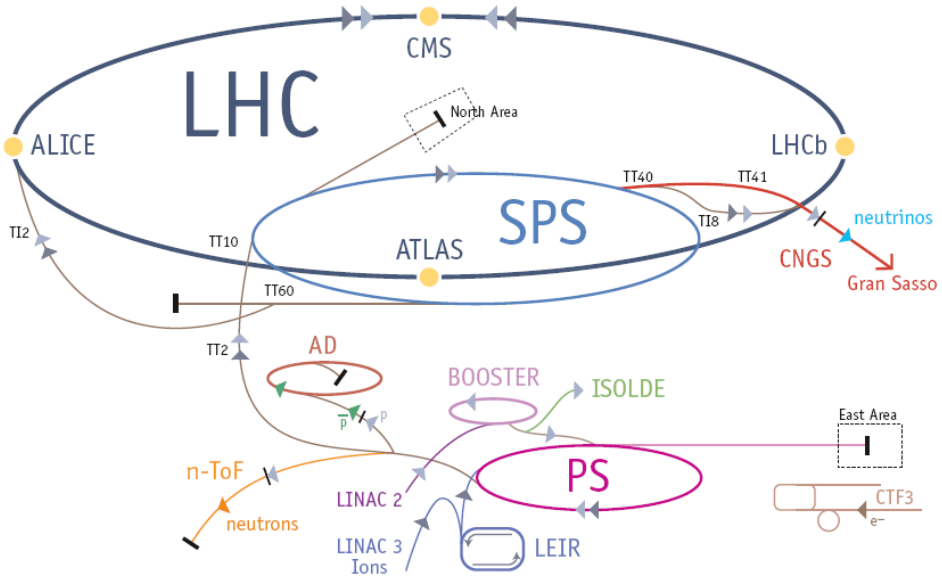


Figure 2.1: Schematic drawing of the different particle accelerators and facilities at CERN. The drawing is taken from Ref. [36].

2.2 The ALICE Detector

This section describes the ALICE detector setup in more detail. ALICE (**A** **L**arge **I**on **C**ollider **E**xperiment) is a general-purpose detector that is one of the four major experiments at CERN. It was built with the intention of studying heavy-ion collisions at the LHC. ALICE is optimized for the study of the quark–gluon plasma (QGP), for which it is specialized in particle identification and tracking even in high particle density environments of about 8000 charged particles per pseudorapidity interval [38]. ALICE measures lead–lead, proton–lead and proton–proton collisions with center-of-mass energies up to 5.02 TeV for lead–lead collisions and up

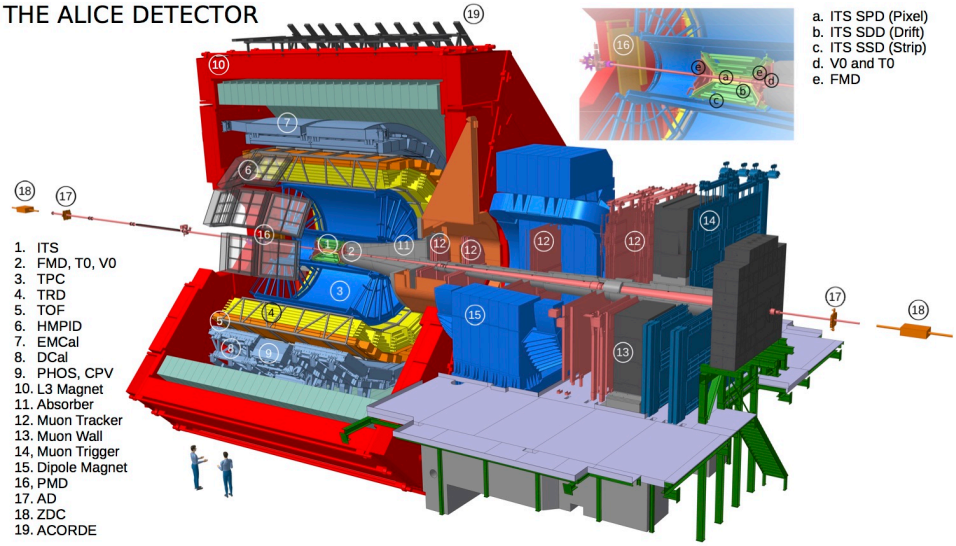


Figure 2.2: Schematic drawing of the ALICE detector. The central barrel (numbers 1–10) as well as the muon arm (numbers 11–15) are shown. The central barrel contains e.g. the Inner Tracking System (ITS), Time Projection Chamer (TPC) and Transition Radiation Detector (TRD) as well as a Time-Of-Flight Detector (TOF), several calorimeters and the 0.5 T solenoid. Drawing taken from Ref. [37].

to 13.6 TeV for proton–proton collisions. The detector is able to cover a wide range of transverse momenta from particles with a few hundred MeV/c up to $\mathcal{O}(20\,\mathrm{GeV}/c)$. This will be achieved with an onion-like structure of different detector systems dedicated to specific purposes and complementing each other. A schematic view of the ALICE detector is shown in Fig. 2.2. Coming from the center of the beam pipe, particles first pass through the Inner Tracking System (ITS) and then enter the Time Projection Chamber (TPC), ALICE's main tracking detector. Around the TPC, the 18 segments of the Transition Radiation Detector (TRD) and the Time-Of-Flight Detector (TOF) are built. The last layer of the detector setup marks a series of calorimeters, namely the Electromagnetic Calorimeter (EMCal), the Di-jet Calorimeter (DCal), the Photon Spectrometer (PHOS) and a Cherenkov detector (HMPID, acronym for High-Momentum Particle Identification). This entire detector assembly is enclosed in a solenoid called the L3 Magnet, which generates the standard operating magnetic field of 0.5 T. At this standard magnetic field strength, a minimum p_T of about 200 MeV/c is required for particles to reach the outer radius of the TPC, but tracks with lower p_T can still be reconstructed inside the TPC down to a few 100 MeV/c [39]. The ALICE setup is not symmetric along the beamline

due to the addition of the muon arm, whose main purpose is to detect muons, to the detector apparatus on the C-side of the detector (C for clockwise, referring to the direction of the beam in the LHC), which corresponds to the right side in Fig. 2.2. The left side is also called the A-side (anti-clockwise). In total, the detector occupies a space of $16 \times 16 \times 26\,\mathrm{m}^3$ and weighs about 10 000 t.

In the following sections, the TPC, which is the main detector used for the sexaquark search, as well as the ITS, TRD and the TOF detector will be explained in more detail.

2.2.1 Inner Tracking System

The Inner Tracking System (ITS) is the innermost detector of ALICE and consists of three different types of silicon semiconductor detectors as can be seen in the inlet of Fig. 2.2. The first part is a silicon pixel detector, followed by a silicon drift detector and finally a silicon strip detector. Each silicon detector layer consists of two successive layers for a total of six silicon detector layers. The ITS is able to cover a pseudorapidity range of $|\eta| < 0.9$ and it covers the radii between 39 and 430 mm. Since the ITS is the detector closest to the interaction point, it plays a crucial role in almost all measurements made in the central barrel of the ALICE detector. Its main purpose is to reconstruct the primary vertex with a very high resolution better than 100 µm and to identify and track particles that do not have enough momentum to reach the TPC. For tracks that are also reconstructed with the TPC, it is used to improve the momentum and pointing resolution. To minimize the impact on the particle trajectory, the ITS was specifically designed to have a small material budget. The ITS is also used to identify D-meson decays that have a decay length below 100 µm. For LHC Run 3, the ITS has been replaced by a new detector [40].

2.2.2 Time Projection Chamber

The Time Projection Chamber (TPC) is the main device for tracking and particle identification (PID) in ALICE. It consists mainly of a gas-filled cylinder with a radial active coverage of 0.83 to 2.50 m oriented along the beam axis. It is 5 m long,

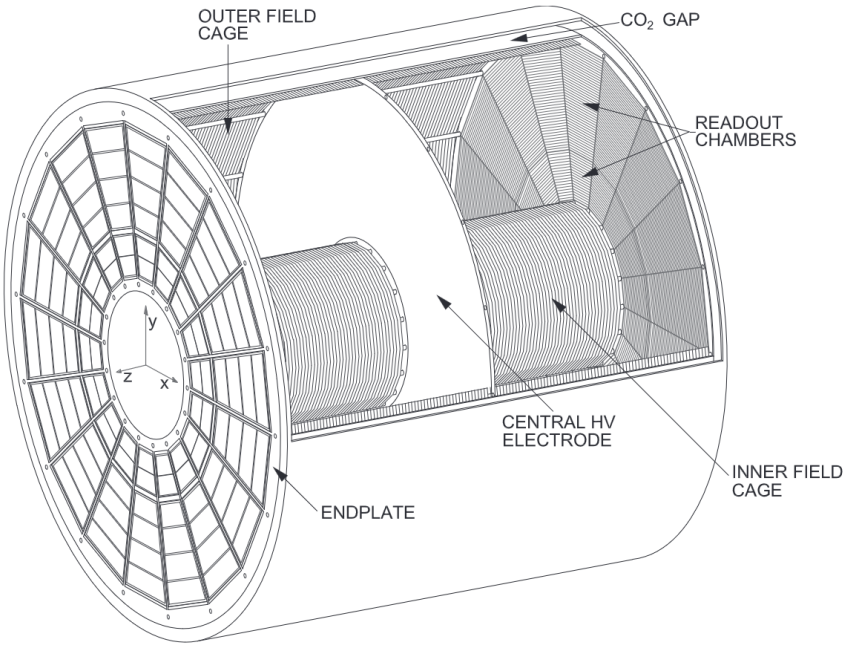


Figure 2.3: Schematic view of the Time Projection Chamber (TPC) of the ALICE detector. Picture taken from Ref. [41].

which means that it covers a pseudorapidity range of $|\eta| < 0.9$ similar to the ITS. A schematic drawing of the TPC can be seen in Fig. 2.3. The drum is filled with a gas mixture of argon and CO_2 at atmospheric pressure and the central electrode is charged to 100 kV, resulting in an axial electric field. Charged particles traversing the barrel ionize the gas, creating electrons in the process. These electrons drift along the field lines of the electric field towards either end of the detector where the readout chambers are located. The drift velocity of the electrons is approximately constant due to scattering from gas molecules. At the readout chambers, a signal is observed that is proportional to the energy lost by the particle as it passes through the chamber. At the readout chamber, 159 pad rows in the radial direction at each end record a two-dimensional image of the trajectory by assembling the x–y positions of the incoming drift electrons, which can then be reconstructed, together with the information of the drift velocity and the time of the chamber hits, to the exact three-dimensional path taken by the particle inside the barrel. The TPC can fully reconstruct charged particles with a transverse momentum of $p_{\mathrm{T}} \geq 100\,\mathrm{MeV}/c$ [42]. The energy loss of the charged particle in the TPC dE/dx can then be calculated

with the Bethe-Bloch formula using the energy deposited by the drift electrons:

$$-\frac{dE}{dx} = Kz^2\frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}\right] \tag{2.1}$$

All necessary parts of Eq. (2.1) are known for the TPC detector due to calibration. The dE/dx can then be used to give a hypothesis on the particle species based on the momentum of the traversing particle since the particles all follow a species specific curve in the dE/dx over momentum plot as seen in Fig. 2.4. The dE/dx values can

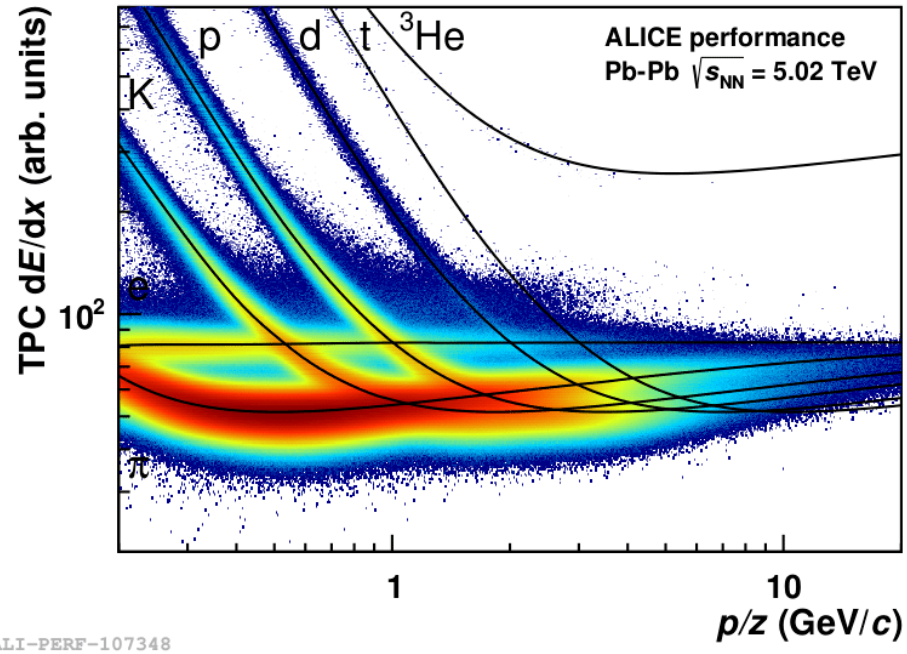


Figure 2.4: Particle Identification by dE/dx and momentum in the TPC. Picture taken from Ref. [41].

be estimated with an uncertainty of 5 %. From the plot, the distance of a particle dE/dx from all the particle lines is calculated in proportion to its uncertainty and summarized in a variable that represents the certainty that this particle belongs to a certain group: the $n\sigma$ value. This is the value used for PID estimation in this thesis.

2.2.3 Transition Radiation Detector

The Transition Radiation Detector (TRD) is used to provide triggering capabilities of high momentum electrons, jets, and light nuclei, and particle identification (PID) of electrons with a transverse momentum of $p_{\mathrm{T}} \geq 1\,\mathrm{GeV}/c$ [43]. It also plays a crucial role in correcting for space charge distortions in the TPC, as it is housed around the TPC in a radius of 2.9 m to 3.68 m. Transition Radiation (TR) is a form of electromagnetic radiation emitted when a highly relativistic charged particle passes through two materials with different dielectric constants. The amount of radiation emitted depends on the Lorentz factor γ of the passing particle, which makes it possible to distinguish lighter from heavier particles. Therefore, the TRD is divided into four different parts: The radiation region where the TR is generated, the drift region, a gas chamber filled with xenon and CO_2, the amplification region and the readout electronics. Charged particles passing through the drift region ionize the gas, producing electrons which, together with the electrons produced by the transition radiation, travel to the amplification region where the signal is amplified and finally read by the readout electronics. The electrons from the TR are the last to reach the readout chambers and appear as a second signal peak. Slower and heavier particles such as pions do not produce TR and can therefore be distinguished from lighter/faster particles such as electrons. Currently, the TR is not widely used in current analysis, so the main use for the TRD is to fit tracks produced in the TRD to tracks from the TPC, correcting for space charge distortions in the TPC. The TRD consists of 522 individual readout detector modules arranged in 18 supermodules, each six layers thick in the radial direction and consisting of 5 stacks along the beamline direction. The TRD covers a pseudorapidity range of $|\eta| < 0.84$ and the active radius is from 2.90 to 3.68 m [43]. For Run 3, broken TRD modules were repaired and a new online reconstruction software was implemented.

2.2.4 Time of Flight Detector

The Time of Flight (TOF) detector is located at a distance between 370 and 399 cm from the collision point. The TOF detector is used for particle identification by measuring the time of flight of the particles with a resolution better than 50 ps. The time-of-flight information is used together with the track length to calculate the velocity of the traversing particle, from which its mass can be determined if

the momentum is known. The detector consists of many Multigap Resistive Plate Chambers (MRPCs), which consist of resistive plates with ionizable gas between them. Traversing charged particles ionize the gas, creating free electrons that drift to a high-voltage electrode where they are collected, amplified, and measured. The TOF is mostly used to calculate the squared mass m^2 of the particles, since due to measurement inaccuracies the measured velocity v can be greater than the speed of light c, which would result in imaginary masses for the traversing particles [44].

2.3 Boosted Decision Trees

Machine learning methods are often divided into two classes: supervised and unsupervised. Unsupervised learning refers to algorithms that try to find patterns in unbalanced data sets. Supervised learning methods consist of algorithms such as neural networks, linear discriminant analysis, or decision trees. Supervised learning methods are classified by the availability of labeled examples used as training and testing input. The idea is that the machine learning algorithm is able to produce a function based on the input data that maps the features of that input to the labels of the output. Ideally, the learned function captures the underlying truth behind the input data, allowing it to correctly map features collected in real-world scenarios. When training a supervised learner, two edge cases must be avoided: overfitting and underfitting. Underfitting occurs when the trained classifier is not deep or complex enough to correctly capture the features of the data. Overfitting happens when the classifier is too complex and learns the statistical fluctuations of the training set, making it unable to generalize well enough in real-world scenarios. Balancing these two cases during training and keeping the classifier in a range where it is able to generalize well to the real data, but at the same time capture every piece of truth behind the data, is a delicate task [45].

Boosted decision trees (BDT) are one of these supervised machine learning methods and belong to the tree methods. BDTs are already successfully used in many areas of physics, including particle physics, due to their high classification performance while being fast to train and robust [46]. This makes them a popular choice over other supervised learners such as neural networks, which in general require more data to train the model and are computation intensive. Studies have also shown that tree methods, such as BDTs and random forests, outperform neural networks

in accuracy and training speed for tabular, unordered features [47]. Unordered features are tabular data where the order of the parameters given to the classifier does not matter. For example, if two parameters describe the positions x and y of the sample, it doesn't really matter if x is parameter 1 and y is parameter 2 or vice versa, as long as it is consistent across the data set.

Similar to random forests, BDTs use an ensemble of decision trees to classify their data. A single decision tree is easy to make, but can be quite unstable and not very informative, hence they are referred to as "weak learners". BDTs are based on the idea that many weak learners together form a good learner, with each additional tree improving the classifier by correcting the shortcomings of its predecessors. This is where the boosting part of BDTs comes in, since for each new tree, the training examples that were misclassified by the existing ensemble are given higher weights and play a larger role in building the new tree. By construction, the later trees are better at classifying examples misclassified by the previous ensemble, which is where much of the strength and versatility of BDTs comes from [48]. The final classification on the test and real data is not done on the final tree, but on the entire ensemble, with each tree making a weighted guess and the final result being the weighted sum of all predictions:

$$F(x) = \sum_{i=1}^{N_{\text{tree}}} \alpha_i h_i(x). \tag{2.2}$$

Here, N_{tree} denotes the total number of trees, α_i the weight of the i-th tree h_i and $F(x)$ is the final prediction. Training of the forest is done by minimizing a given loss function $L(y_j, F(x_j))$, where y_j denotes the desired output and $F(x_j)$ is given by Eq. (2.2). A new learner (a.k.a. tree) $h_j(x)$ is then added to the ensemble by minimizing the loss function with a data sample of size N:

$$\sum_{j=1}^{N} L_n(y_j, F_{n-1}(x_j) + \alpha_n h_n(x_j)) \to \min_{\alpha,h}. \tag{2.3}$$

Here, the loss is minimized for the new tree h_n by finding the appropriate weight α_n. This method of adding more weak learners to ensemble is what is commonly referred to as *boosting*. XGBoost (Extreme Gradient Boosting) is a machine learning algorithm first introduced by Chen and Guestrin in 2016 which builds on the mathematical ideas of boosted decision trees and improves the generic implementation with a focus on performance and scalability [49]. XGBoost uses an ensemble of decision trees as weak learners consisting of classification and regression trees

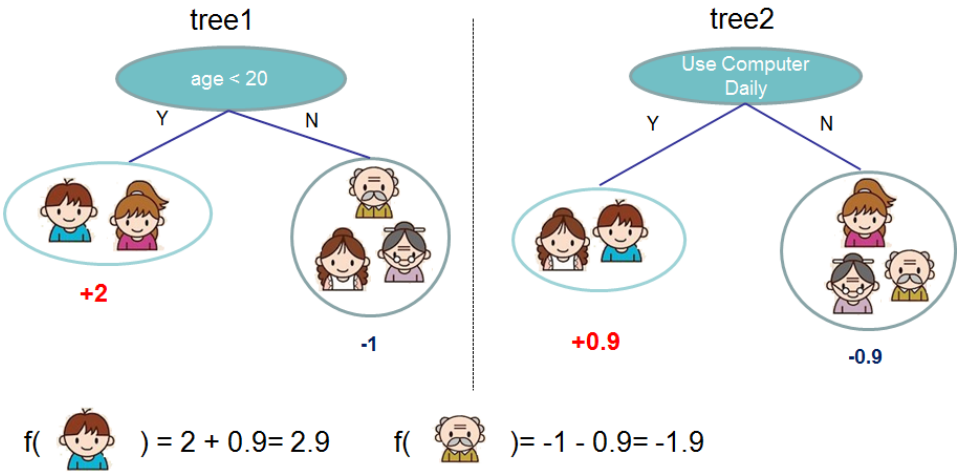


Figure 2.5: Example of a forest in XGBoost for an arbitrary case. The bottom part shows the calculation of the final score for two examples. Picture taken from Ref. [49].

(CART) and is the method of choice in this work. An example of a two-tree forest generated by XGBoost with its score function can be seen in Figure 2.5. It is a powerful algorithm that has gained popularity in the machine learning community due to its high accuracy and efficiency, and has gained a lot of publicity after being used by many winning teams in machine learning competitions. As its name suggests, XGBoost relies on the use of gradient descent to find a local minimum of the loss function while building new trees. XGBoost uses second-order approximations of the loss function in Eq. (2.3) to optimize the gradient boosting step. It is also aware of sparsity in the input data, which is handled by a defined default direction for missing data, when evaluating a split. Furthermore it features exact and approximate split-finding algorithms to improve scalability, accuracy, and training speed compared to other gradient boosting algorithms.

3 Analysis

3.1 Analysis Strategy

In this section, the analysis strategy is discussed in detail. Our attempt to detect the sexaquark will mainly follow Farrar's proposal to detect an anti-sexaquark, created as a result of a heavy ion collision inside the LHC and subsequently reconstruct the interaction of the anti-sexaquark with the detector material, as described in Chapter 1.2. The analysis will thus closely follow the previous attempt by De Clercq at CMS [6], with the goal of improving the reconstruction efficiency made possible by the better tracking and PID capabilities of ALICE. This work will mainly focus on identifying the anti-sexaquark in simulated Monte Carlo data and developing the necessary tools to do so. The application of the developed tools on real data goes beyond the scope of this thesis. An anti-sexaquark created in a collision inside the LHC and subsequently annihilated with the detector material can result in numerous different interaction channels (listed in Table 1.1), each of which leads to a different detector response and therefore a different approach to reconstruction and chance of accurately identifying the sexaquark. First, one has to choose a channel to reconstruct, and to this end simulations containing only sexaquarks and their interaction products were used to probe the detector response and evaluate which channel has the highest chance of successful reconstruction, as well as straightforward analysis procedures. The channel chosen was $\bar{S}+n \rightarrow \bar{\Lambda}+K^0_S \rightarrow \bar{p}+\pi^+ +\pi^- +\pi^+$ for reasons discussed in the following chapter 3.2. For this channel, the analysis is performed by first reconstructing $\bar{\Lambda}$ and K^0_S, and identifying their decay vertices consisting of a positive and a negative particle (V^0s). For this channel, simulations were performed with the sexaquark embedded in simulated proton–proton and lead–lead collisions. The identification of the sexaquark within the simulated data was performed using the ALICE TPC. The reconstruction of the particle tracks within the TPC followed the standard ALICE reconstruction workflow, but for the reconstruction of the V^0s of these particles, modifications to the standard approach had to be made. Within

the standard ALICE reconstruction framework, there is already a well-established and well-tested V^0 finder, but it is specialized to identify V^0s from particles originating at the primary vertex of the collision. As a result, in an attempt to reduce the immense combinatorial background, strong cuts are applied that strongly disfavor particles originating at secondary vertices, which also eliminates almost all of our signal V^0s. To mitigate this, a Custom V^0 Finder was developed that uses custom cuts to preserve our signal and additionally reduce primary V^0s. The Custom V^0 Finder is discussed in detail in Section 3.4.1. During the development of the Custom V^0 Finder, Monte Carlo truth information about the simulation was used to reconstruct the true V^0s of the decaying particles to develop and test the rest of the analysis. Once the V^0s are reconstructed, they are combined into sexaquark candidates. Each candidate consists of two V^0s sharing a common secondary vertex (SV). To find these secondary vertices, a V^0 pair finding algorithm has been developed. The algorithm uses the reconstructed position and momentum vectors of the two V^0s and propagates them in a straight line backwards to the point of closest approach (PoCA), which marks the secondary vertex of these two V^0s. The secondary vertices are computed for all possible combinations of V^0s for each event, and if at this point the distance of closest approach (DCA) is below a certain threshold (10 cm for the V^0 pair finder, but later cuts further reduce this value for candidates), the two V^0s together form a *sexaquark candidate*. The large combinatorial background for custom V^0 sexaquark candidates requires a series of background rejection cuts based on the DCA, vertex distance, and reconstructed V^0 masses. After applying these cuts, the remaining sexaquark candidates are subjected to a boosted decision tree (BDT) classifier, which further reduces the background based on topological and PID variables. The XGBoost library is used for the BDT. The XGBoost classifier is trained and tested on simulations with the goal of applying it to real data in future work, as application to real data is beyond the time constraints of this thesis.

$\bar{S} + n \rightarrow$	$\bar{S} + p \rightarrow$
$\bar{\Lambda} K^0_S$	$\bar{\Lambda} K^+$
$\bar{\Lambda} K^0_S \pi^- \pi^+$	$\bar{\Lambda} K^+ \pi^- \pi^+$
$\bar{p} K^0_S K^0_S \pi^+$	$\bar{p} K^+ K^0_S \pi^+$
$\Xi^+ \pi^-$	

Table 3.1: Simulated sexaquark interaction channels.

3.2 Pure Sexaquark Simulation and Channel Determination

3.2.1 Pure Sexaquark Simulation

To begin the analysis, a pure simulation of an anti-sexaquark interacting with the detector material was used to measure the detector response. The simulation itself was provided by Andrés Bórquez and was performed using the ALICE reconstruction framework with the GEANT4 environment [50]. Simulations in GEANT4 must be anchored to pre-existing runs, and the pure sexaquark simulations are all anchored to a Pb–Pb collision run (run number = 246225) with a CMS energy of 5.02 TeV. Since the anti-sexaquark is not part of the GEANT4 package, the simulation is limited to the products of the interaction between the $\bar{S}$ and the detector material. The momenta and directions of the daughter particles of the annihilation are determined using ROOT's TGenPhaseSpace class [51]. A $\bar{S}$ with a mass of exactly $1.8\,\mathrm{GeV}/c^2$ and a transverse momentum uniformly distributed between 0 and $5\,\mathrm{GeV}/c$ is collided with a neutron or proton at rest. The resulting particles are injected into the detector with the appropriate momentum and direction within a uniform spherical radius range between 5 and 180 cm. The initial anti-sexaquark has a uniform ϕ range between 0 and 360° and a uniform rapidity range between -1.8 and 1.8. The daughter particles of the interaction are then propagated through the detector using GEANT4.

The first simulations contain 10,000 interacted anti-sexaquarks for each interaction channel, which are listed in Table 3.1. These are all possible interaction channels that do not contain a π^0 as a daughter particle, since the π^0 reconstruction efficiency in ALICE is rather low. In Fig. 3.1, an event display of such a simulated anti-sexaquark annihilation is shown.

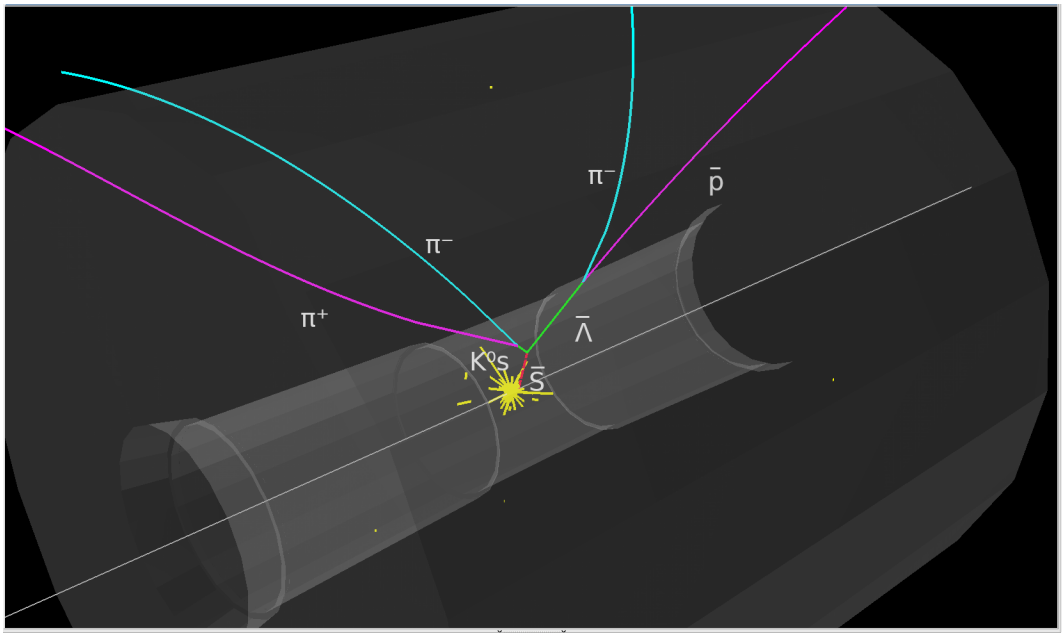


Figure 3.1: Labeled event display of a simulated anti-sexaquark interacting inside of the TPC in the channel $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0 \rightarrow \bar{p}\pi^+\pi^-\pi^+$. Positive daughters are drawn in purple, negative daughters in cyan and the neutral interaction products are depicted in green. The anti-sexaquarks path is depicted as red dotted line. The yellow lines correspond to background Λ and K_S^0. Picture provided by Andrés Bórquez.

3.2.2 Estimation of the Reconstruction efficiency

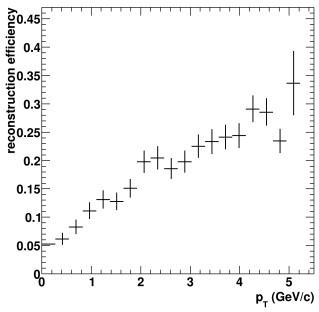
The response of the detector to a potential anti-sexaquark was investigated using the pure sexaquark simulations. First, the interaction channel with the highest probability of successfully detecting the sexaquark has to be selected for further investigation. For this reason, histograms of the detectors reconstruction efficiency of the sexaquark are made for each channel. An event was considered able to be reconstructed, if each final state particle of the interaction left a detectable track in the TPC. Reconstruction efficiency is defined as

$$rec = \frac{\#_{\text{able to reconstruct}}}{\#_{\text{total events}}} \tag{3.1}$$

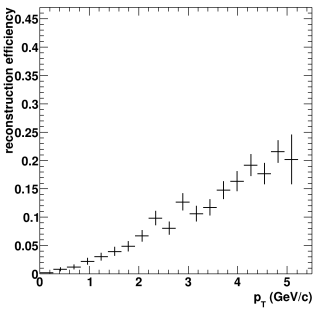
The reconstruction efficiencies of the sexaquarks for each channel are plotted as a function of the momentum of the initial sexaquark. These plots can be seen in Fig. 3.2, where the reconstruction efficiency is calculated for every momentum bin.

Figure 3.2 is divided into three rows. The first row represents the sexaquark interacting with a neutron, the second row represents the sexaquark interacting with a proton, while the third row is occupied by the specific interaction chain $\bar{S}+n \rightarrow \Xi^+ + \pi^-$. When reconstructing this channel, an additional decay level must be taken into account, since the Ξ^+ first decays into a π^+ and a $\bar{\Lambda}$, which then decays further.

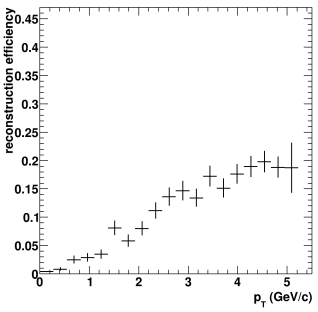
Comparing the histograms, three observations can be made. First, the more particles are produced in an interaction, the lower the reconstruction efficiency becomes. This can be seen by comparing the reconstruction efficiencies of 3.2a with 3.2b and 3.2c, where the channel $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0$ produces two daughter particles and four final granddaughter particles, while the other two $\bar{S}+n$ interactions produce four and six daughter and granddaughter particles, respectively. In addition, the $\bar{S}+n \rightarrow \Xi^+ + \pi^-$ channel also has a high chance of being reconstructed compared to the other $\bar{S} + n$ channels, although it requires an additional layer of reconstruction. This is explained by the decay products of Ξ^+, which are a $\bar{\Lambda}$ and a π^+, and therefore only four final state particles need to be reconstructed. Furthermore, the absence of a K_S^0 further improves the reconstruction efficiency of this channel. The same observation can be made in the $\bar{S} + p$ interactions, where the reconstruction efficiency of 3.2d around medium p_{T} ranges is higher at about 35 % than the other two channels at $20-30$ %, which can also be explained by the two daughter and three final particles for the first



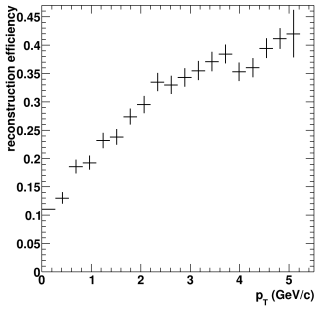
(a) $\bar{S} + n \rightarrow \bar{\Lambda} + K_S^0$



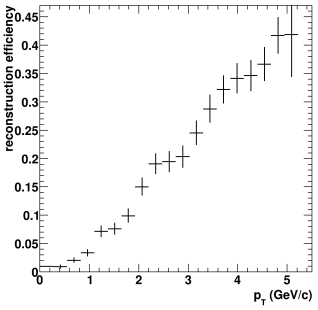
(b) $\bar{S} + n \rightarrow \bar{\Lambda} + K_S^0 + \pi^+ + \pi^-$



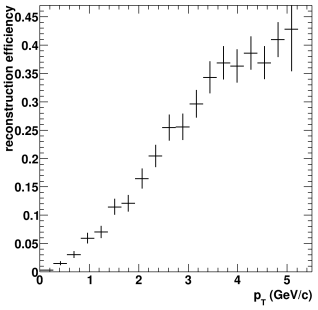
(c) $\bar{S} + n \rightarrow \bar{p} + K_S^0 + K_S^0 + \pi^+$



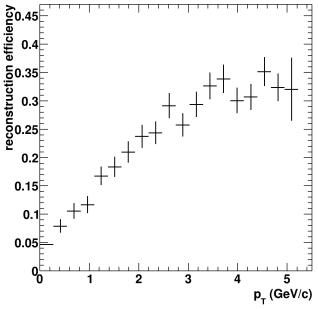
(d) $\bar{S} + p \rightarrow \bar{\Lambda} + K^+$



(e) $\bar{S} + p \rightarrow \bar{\Lambda} + K^+ + \pi^+ + \pi^-$



(f) $\bar{S} + p \rightarrow \bar{p} + K^+ + K_S^0 + \pi^+$



(g) $\bar{S} + n \rightarrow \Xi^+ + \pi^-$

Figure 3.2: Reconstruction efficiency plots of considered sexaquark interaction channel with the detector material. The reconstruction efficiency is calculated for every momentum bin, defined as $rec(p_\mathrm{T}) = \frac{\#_{\text{able to reconstruct}}(p_\mathrm{T})}{\#_{\text{total events}}(p_\mathrm{T})}$.

channel and four and five daughter and final particles for the other two channels. The reason for the decreasing reconstruction efficiency with increasing number of final particles is that for each additional particle, the chance of not fully reconstructing one of them increases.

The second observation is that if one compares the annihilation channels of the interaction $\bar{S} + n$ with $\bar{S} + p$, one can see that the reconstructability is higher across all $\bar{S} + p$ channels of each column. The difference between the two cases is that the daughter particles of the $\bar{S} + n$ interactions contain K_S^0 and the $\bar{S} + p$ interactions contain K^+. The K^+ has a higher probability to be reconstructed because it is a charged particle and its reconstruction occurs directly, while the K_S^0 decays can only be observed indirectly. Furthermore, it is not guaranteed that the K_S^0 can be reconstructed, since in 30.4 % the K_S^0 decays into two π^0, which are not reconstructed.

The third observation is that for all channels shown, the reconstruction efficiency increases steadily with transverse momentum up to $5\,\mathrm{GeV}/c$, which is the maximum possible p_T of an $\bar{S}$ in the simulation, generated uniformly between 0 and $5\,\mathrm{GeV}/c$. All distributions show a rapid decrease in reconstruction efficiency at low momenta, which can be explained by the occurrence of curling within the TPC. Since the $\bar{S}$ has such a low momentum, its daughter particles also have low momenta, which prevents them from leaving the TPC, resulting in curling that makes it difficult to reconstruct these tracks.

3.2.3 Channel Selection

Due to time constraints, one channel had to be selected for further investigation. The branching ratio of the sexaquark interactions has not yet been determined, so no branch can be favored a priori over others, so the main consideration in choosing the channel with which to continue the analysis was the feasibility of reconstruction. The choice fell on the interaction channel $\bar{S} + n \rightarrow \bar{\Lambda} + K_S^0$, although looking at Fig. 3.2, the channel with the highest reconstruction potential is the interaction of the sexaquark and the proton in anti-lambda and kaon. Other channels than these two have been discarded due to their increased particle count and thus reconstruction complexity. The reasons for choosing the neutron interaction channel over

the proton interaction channel were, first, that the reconstruction of the sexaquark-neutron interaction products requires finding two decay vertices (V^0s), whereas for the sexaquark-proton interaction one secondary decay vertex has to be found and combined with a particle track to find a common vertex. For finding V^0s, suitable software already exists within the standard ALICE reconstruction framework in the form of a V^0 finder. Finding the origin of two V^0s can easily be done by calculating the distance of closest approach of the reconstructed momentum lines of both V^0s. To find the origin of a V^0 and a charged particle, the distance of closest approach of a line and a helix must be determined, which, while doable, requires computations of similar complexity to a V^0 finder without readily available software, and developing such a software is beyond the scope of this thesis. Combining a track with a V^0 is also expected to have a significant combinatorial overhead, since there are generally more tracks than V^0s. Finally, the previous search for the sexaquark at CMS also looked for the $\bar{S} + n \to \bar{\Lambda} + K^0_S$, so it is a good choice for comparison between the two experimental setups.

3.3 Signal and Background Simulations with True V^0s

After selecting an interaction channel to study first, simulations were performed in which the sexaquark interaction products of this channel ($\bar{\Lambda}$ and K^0_S) were embedded in simulations of and Pb–Pb events acting as background. A more detailed look at the simulations is given in the following Section 3.3.1. The simulated event is reconstructed using the standard reconstruction framework, with the exception of the official V^0 finder, which is designed to find V^0s of particles coming from the primary vertex. Until the development of the Custom V^0 Finder was finalized, V^0s reconstructed with Monte Carlo truth information about decaying particles (a.k.a. true V^0s) were used to develop and test the XGBoost classifier. The process of finding true V^0s is described in Section 3.3.2. First, the sexaquark simulations were embedded in proton-proton collision simulations and a first test version of the classifier was developed. Subsequently, sexaquarks embedded in Pb–Pb collisions were used to further improve the classifier and investigate its performance. This version used in pp events was used as a proof of concept in a low background environment, and due to the lack of further insights, only the results of the Pb–Pb event simulations are described in Section 3.3.2.

3.3.1 Background Simulations with Embedded Sexaquarks

This section describes the generation of the MC events used in the analysis, provided by Andrés Bórquez. The simulation was generated by injecting the interaction products of the anti-sexaquark reaction with a neutron onto a generated MC event of a Pb–Pb collision at 5.02 TeV, where the underlying collision was created using the HIJING event generator [52]. The anti-sexaquark simulation was performed in a similar way to the pure signal simulations (see Section 3.2.1) with minimal adjustments to some of the parameters and the difference that this time only the annihilation channel selected in Section 3.2.3 ($\bar{S} + n \to \bar{\Lambda} + K^0_S$) was simulated. Again, only the interaction products of $\bar{S}$ and n are injected into the simulations, whose momenta are calculated using ROOT's TGenPhaseSpace class. The $\bar{S}$ has a fixed mass of $1.8\,\mathrm{GeV}/c^2$ and is injected in a flat rapidity range between -0.8 and 0.8 with a transverse momentum between 0 and $5\,\mathrm{GeV}/c$. The $\bar{S}$ then annihilates on a resting neutron and the calculated daughter particles are injected in a ϕ range between 0 and 360°. The difference is that the daughters are injected in a transverse radius range between 5 and 180 cm instead of the spherical radius. The daughter particles of the $\bar{S}$ annihilation, as well as the other particles of the collision produced by HIJING, are then propagated, transported, tracked and reconstructed using the ALICE reconstruction framework, which makes use of GEANT4. The simulations are anchored to existing run numbers, and there are 80 run numbers from the "LHC15o" run recorded in 2015 and 91 run numbers from the "LHC18r" run recorded in 2018. A table of all used run numbers can be seen in Tab. A.1. For each of these numbers, four simulations consisting of 250 events each were produced, resulting in a total of 171,000 simulated events, each containing an annihilated $\bar{S}$. These 171,000 events are the basis for all further analysis in this thesis.

3.3.2 XGBoost Classifier on True V^0 Candidates from Pb–Pb Collision Simulations

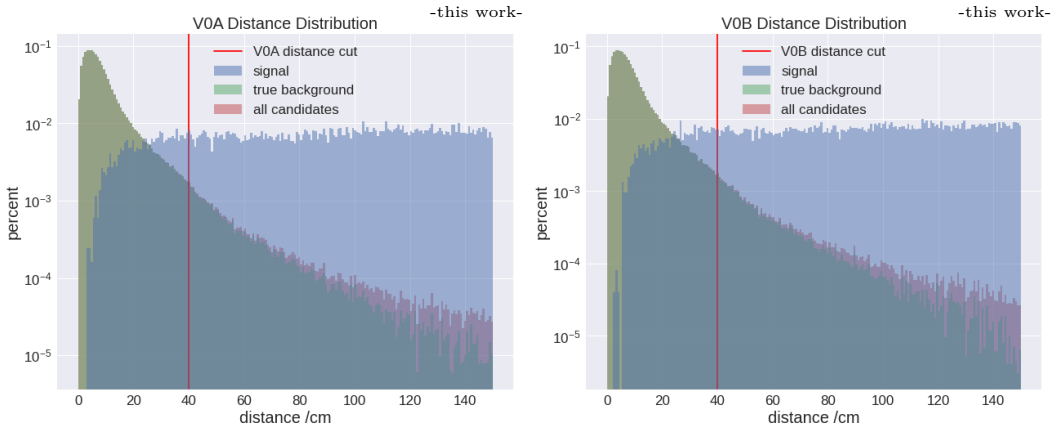
Monte Carlo True V^0 Finding

The specialization of the official V^0 finder on decaying particles coming from the primary vertex results in a strong suppression of V^0s from particles coming from

secondary vertices, and therefore the majority of V^0s coming from interacted sexaquarks are rejected as possible candidates. This requires the use of a Custom V^0 Finder and the use of Monte Carlo truth information to reconstruct V^0s until the Custom V^0 Finder was operational. True V^0 detection was accomplished by going through all MC generated particles and checking whether or not the particle decayed within the detector. The MC truth information was then used to identify its daughter particles and see if a positively charged daughter particle and a negatively charged daughter particle were produced. These particles were checked to see if they left a reconstructed particle track within the TPC, and if they did, their V^0 was reconstructed as the true position where the parent particle decayed, as well as its true momentum. From these, V^0 pairs were formed using the V^0 pair finding algorithm developed during this thesis, which is described in detail in Section 3.4.2.

True V^0 Candidate Cuts

Due to the ongoing development of the Custom V^0 Finder at the time, sexaquark candidates reconstructed from V^0s obtained using MC truth information were used to further develop the XGBoost classifier, as well as to consider and test possible high-impact features and cuts to be used in the final reconstruction task with V^0s found using the Custom V^0 Finder. This development on real V^0s is also a good way to explore how the analysis would play out on a perfect, error-free detector. After reconstruction, true V^0 finding, and V^0 pair finding of all $170,000$ events, a total of 65.8×10^6 sexaquark candidates are found, of which 43.5×10^3 are signal candidates. Of the resulting 68×10^6 background candidates, 65.1×10^6 belong to the true background (candidates where no V^0 is from the sexaquark) and 0.7×10^6 belong to candidates consisting of a V^0 from the sexaquark and a V^0 from another decayed particle. Signal candidates would make up less than $0.1\,\%$ of the training data, so two cuts are applied to reduce this imbalance. These two cuts target the distance of the V^0s to the primary vertex, and each of the V^0s is required to be further than $40\,\text{cm}$ from the primary vertex. The distance distributions of the two V^0s (V^0 A and V^0 B) of each candidate can be seen in Fig. 3.3. After these cuts, a total of 64.4×10^3 candidates remain, of which 36.3×10^3 belong to signal and 20.4×10^3 belong to true background candidates, which is a good signal-to-background ratio for training the classifier.



(a) (b)

Figure 3.3: Normalized V^0 distance of V^0 A and V^0 B to the primary vertex. The plots show the distribution of all candidates in red, the true background (candidates where neither A nor B stem from the sexaquark) in green and the true sexaquark signal candidates in blue. From these plots, cuts are derived to reduce the background, for which the threshold at 40 cm is shown as red vertical line.

XGBoost True V^0 Classifier Training & Results

Training a classifier model requires a set of features that the classifier uses to distinguish between signal and background candidates. These features must be descriptive and numerous enough that they can somehow be mapped to the underlying truth, but not too many so that training can be accomplished in a reasonable amount of time. After trying many different variables, the choice fell on a set of nine variables that the classifier should use to distinguish signal candidates from background candidates. The first four features give information about the Armenteros-Podolanski variables α and q_T for each reconstructed V^0: `V0<A/B>_ArmAlpha` and `V0<A/B>_ArmPt`. Cuts on Armenteros-Podolanski plots are a standard procedure for identifying V^0s as $K^0_S, \Lambda, \bar{\Lambda}$, or γ, and therefore a reasonable choice of features to separate the V^0s. The Armenteros-Podolanski variable α is defined as $\alpha = (p^+_\mathrm{L} - p^+_\mathrm{L})/(p^+_\mathrm{L} + p^-_\mathrm{L})$, where $p^{+/-}_\mathrm{L}$ are the momenta of the positive/negative particles of V^0 in the longitudinal direction relative to the reconstructed momentum vector of V^0, and similarly q_T is the momentum of the particles in the transverse direction relative to the reconstructed momentum vector of V^0 (named q_T to avoid confusion with p_T). An example of an Armenteros-Podolanski plot can be seen in Fig. A.1 in the appendix. Then there are five positional features: two for the distances between the primary vertex

and the V^0s `V0A_Dist` and `V0B_Dist`, one for the distance between the secondary vertex and the primary vertex `Dist_Vert`, and two for the distances between the V^0s and the secondary vertex `Dist_V0A_Vert` and `Dist_V0B_Vert`. Finally, there are three momentum related features which give the transverse momenta of the V^0s, `V0A_Transv_Mom` and `V0A_Transv_Mom`, as well as the opening angle between the V^0s momenta `Opening_Angle`. The sample of 64.4×10^3 acquired candidates is labeled such that a signal candidate (a candidate where both V^0s come from $S + N$ annihilation) is labeled 1, while true background and mixed candidates are labeled 0. The data set was shuffled and a $70 : 30$ split between training and test data was applied. As hyperparameters for the XGBoost classifier, mostly the default settings are used (see Section 3.4.5), with a small adjustment of the parameter `scale_pos_weight`, which is set to 0.25 to make the classifier more resistant to misclassifying background candidates. XGBoost creates a forest of 100 trees, each up to six layers deep. By counting how often each feature is used in cuts throughout the forest, the feature importance can be measured, which is plotted in Fig. 3.4. As can be seen, the most frequently used features describe the positional relationships between the secondary vertex and the V^0s, such as the opening angle and the distance between the V^0s and the SV, while PID features such as the Armenteros-Podolanski variables are less frequently used. To measure the performance of the classifier, the so-called receiver operating characteristic (ROC) curve is plotted, as shown in Fig. 3.5a. The ROC curve is a metric first introduced in 1941 for military radar receivers, hence its name, but is now commonly used in machine learning to illustrate the diagnostic ability of a binary classifier (a classifier that separates into only two classes). The ROC curve is drawn by plotting the true positive rate (TPR) against the false positive rate (FPR), which is given by:

$$TPR = \frac{TP}{TP + FN} \tag{3.2}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.3}$$

where TP is the number of true positives, FN is the number of false negatives, FP is the number of false positives and TN is the number of true negatives. The area under the ROC curve is generally used as a measure of a classifier. A perfect classifier has an area of 1, which is achieved for the training set, and almost achieved for the test set, which has an auc value higher than 0.995, since it is rounded up to 1. Fig. 3.5b shows a zoomed version of the ROC curve, showing the difference between the test and training curves. The curve shows that with a false positive rate of 0

(no misclassified background candidates), a true positive rate of nearly 0.8 can be achieved. A more readable plot is the posterior distribution of the classifier, shown in Fig. 3.6, where the distribution of test candidates is plotted as a function of the classifier score. One can see that a perfect separation between the true background and signal candidates is achieved, and also a very clear separation between the mixed and signal candidates. It can also be seen that mixed background candidates, where a V^0 comes from the sexaquark, can achieve higher scores and thus be more similar to sexaquark candidates. Training is performed for 100 rounds, and the evolution of the test and training AUC metrics, as well as the root mean squared error (RMSE), is shown in Fig. 3.7. Looking at the evolution of the AUC over the training rounds, one can see a high initial value of over 0.99 for both training and testing with a steep rise towards a perfect classifier. For the training sample, perfection is reached at iteration 40, at which point the AUC of the test sample also reaches a plateau where no further improvement in the AUC is made. This does not mean that there is no overall improvement, as can be seen by looking at Fig. 3.7b, where the RMSE for the test sample still improves up to iteration 60 to 70, with the training set RMSE still improving at training round 100. The improvement of the RMSE means that the classification results of successful signal/background classifications are pushed further to the corresponding label (1 for signal and 0 for background).

The tests on true V^0s show that very good sexaquark identification can be achieved with perfect detector measurements. Due to the nature of the MC truth information about momenta and position, some features were not considered because they have too much influence. One such feature is the *DCA* value, which due to the MC truth information gives a clear distinction between pairs of V^0s from a real common origin and V^0 pairs that are combined as part of the combinatorial background. After successful testing of the classifier on MC true V^0s, the classifier was ported to V^0s found by the Custom V^0 Finder, which is described in detail in the following chapters.

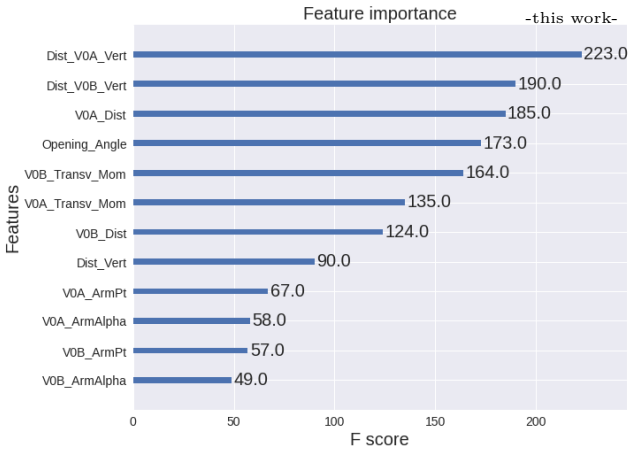
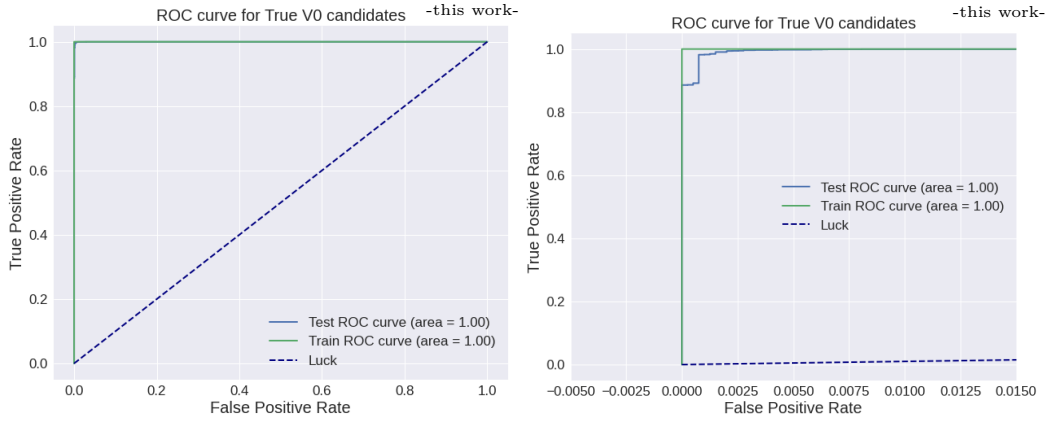


Figure 3.4: Feature Importance plot of XGBoost classification. The plot shows how often each of the features is used as a cut during classification.



(a) (b)

Figure 3.5: ROC curve of the XGBoost classifier used on the true V^0 data sample. On the left, the whole ROC curve is drawn and on the right, a zoomed in version of the same curve. The test curve is drawn in green and the training curve is drawn in blue. The dark blue diagonal represents a luck-based classifier which works by random guesses of the classes.

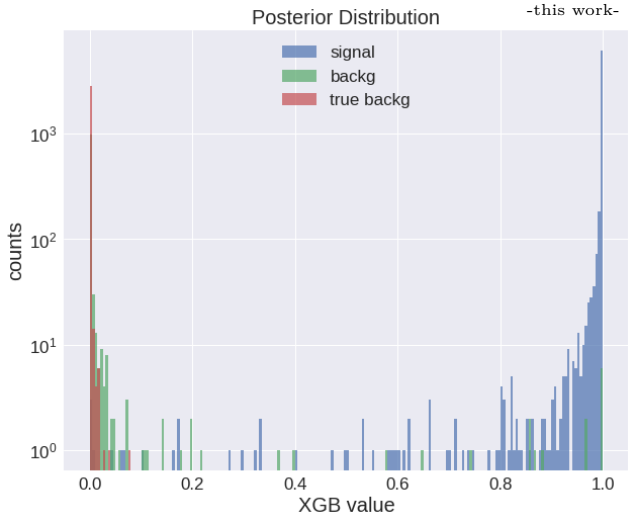
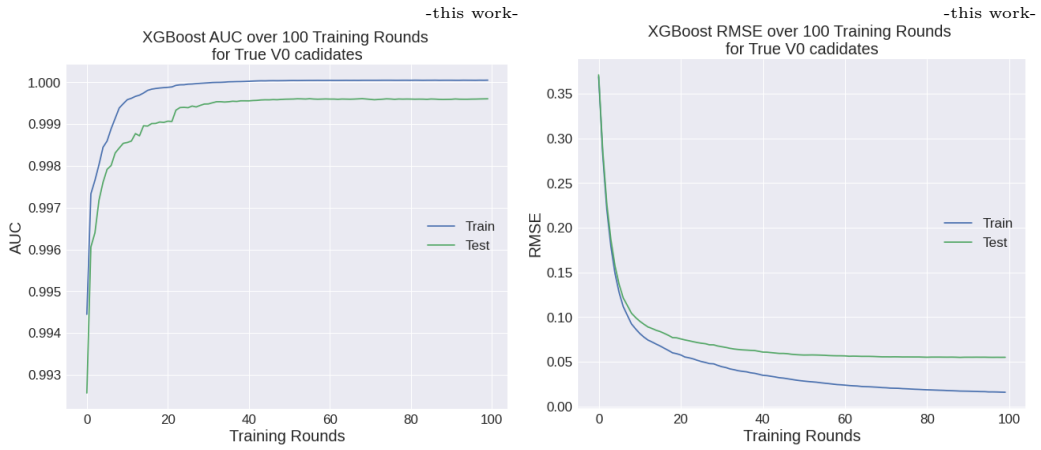


Figure 3.6: Posterior distribution of the XGBoost classifier. The counts of the different classes are shown as a function of the XGB score. Signal candidates are shown in blue, true background candidates in red and mixed background candidates are depicted in green.



(a) (b)

Figure 3.7: Training history of the XGBoost classifier on True V^0 candidates. On the left, the AUC metric and on the right, the RMSE is shown. The scores of the training set are plotted in blue and the scored of the test samples are shown in green.

3.4 Signal and Background Simulations with Custom V^0s

3.4.1 Custom V^0 Finder

Finding V^0s in an official reconstruction requires one of two code bases, the offline and the online V^0 finder. The online (or on-the-fly) V^0 finder is used in parallel with the official ALICE reconstruction, and despite its superior reconstruction of e.g. photons, small changes to it would require a complete reconstruction of the available data by the ALICE Collaboration. Such a request is not feasible at this stage of the analysis, so we focus on the offline V^0 finder.

The offline V^0 finder can be applied to already reconstructed tracks, making it possible to make adjustments and recalculate the vertices. The offline V^0 finder loops over all possible pairs of positively and negatively charged tracks and uses a Kalman filter algorithm to find the common vertices. A series of cuts are then applied to limit the number of background vertices found. The Custom V^0 Finder uses the same code as the official V^0 finder, but certain cuts had to be reduced and other cuts had to be applied simultaneously in order not to cut away the signal V^0s and not to be overwhelmed by the combinatorial background. The cuts used in both V^0 finders are shown in the Table 3.2, with the official V^0 finder cuts on the left and the custom finder cuts on the right.

Before attempting to find a V^0 in the Custom V^0 Finder, each positive particle track must satisfy that its $n\sigma_\pi$ value is less than 3, and each negative particle track must satisfy that its $n\sigma_\pi$ or $n\sigma_p$ value is less than 3, in order to discard V^0s that are unlikely to be either $\bar{\Lambda}$ or K^0_S, which reduces both the background and the necessary computation time immensely. Then, the official and Custom V^0 Finder require that the χ^2 value of the reconstructed vertex is less than 33 and that the impact parameter with respect to the primary vertex is greater than 1 cm. Now the custom and official V^0 finders differ on the cuts on the DCA of both tracks, the cosine pointing angle to the primary vertex (CPA), and the radius from the primary vertex R of the reconstructed vertex. The main driver of this change is the CPA value, since a particle with a CPA of 0.998 is a particle whose momentum vector points almost directly to the primary vertex, meaning that it most likely originated there. The official V^0 finder is designed to find particles created in the initial Pb–

Offline V^0 Finder	Custom V^0 Finder
	$n\sigma_{p/\pi}(\text{tracks}) \leq 3$
$\chi^2 < 33$	$\chi^2 < 33$
$b(\text{daughter}) > 0.1\,\text{cm}$	$b(\text{daughter}) > 0.1\,\text{cm}$
DCA(neg. track, pos. track) $< 1\,\text{cm}$	DCA(neg. track, pos. track) $< 0.6\,\text{cm}$
CPA(V^0) w.r.t. PV > 0.998	CPA(V^0) w.r.t. PV > 0.6
$0.9 < R < 100\,\text{cm}$	$50 < R < 200\,\text{cm}$
	$\eta(V^0) < 1$
	$p_\text{T}(V^0) > 1\,\text{GeV}/c$
	DCA(V^0, PV) $< 1.0\,\text{cm}$
	NClustersTPC(daughters) > 50
	$\eta(\text{daughter}) < 1.5$

Table 3.2: Comparison of cuts between the Offline V^0 Finder and Custom V^0 Finder.

Pb collision, but since the V^0s of the annihilated anti-sexaquark originate at the secondary vertex, they do not point to the primary vertex and are eliminated by the CPA cut. The CPA cut had to be reduced to 0.6 in order to keep enough signal, which resulted in a much higher background, and therefore stricter cuts on the DCA had to be applied to compensate. The minimum and maximum radius cuts were also increased, since the $\bar{S}$ V^0s coming from a secondary vertex are expected to decay farther away, simply because they are created farther away from the initial collision. The five additional cuts on the Custom V^0 Finder are applied to further reduce the combinatorial background. The cuts require the pseudorapidity η of the V^0s to be below 1 and that of the daughter particles to be below 1.5, the transverse momentum of the V^0s must be greater than $1\,\text{GeV}/c$, the DCA between the V^0 momentum and the primary vertex must be greater than 1 cm, and finally, each daughter track must leave at least 50 clusters inside the TPC to ensure good track reconstruction.

The V^0s found with the Custom V^0 Finder are then combined into pairs to create the sexaquark candidates. How the V^0 pair finding works is explained in the following section.

3.4.2 V^0 Pair Finding Algorithm

The particles found with the Custom V^0 Finder are assumed to be neutrally charged because they are formed by the intersection of a positively charged particle track and a negatively charged particle track. These V^0 particles therefore pass through the detector in a straight line. To find secondary vertices, where the V^0s might originate, one must find intersections of the momentum vectors of the V^0s. The probability of two lines intersecting in the three-dimensional detector is quite low for measured data, even if they belong to the same origin. Therefore, the distance of closest approach (DCA) is calculated, which can be done for two V^0s A and B with the position vectors $\mathbf{x}_A$ and $\mathbf{x}_B$ and the momentum vectors $\mathbf{p}_A$ and $\mathbf{p}_B$ using the following Equation [53]:

$$\mathrm{DCA} = \frac{|(\mathbf{p}_A \times \mathbf{p}_B) \cdot (\mathbf{x}_B - \mathbf{x}_A)|}{\|\mathbf{p}_A \times \mathbf{p}_B\|}. \tag{3.4}$$

This equation only holds true as long as the momentum vectors are not nearly parallel or close to being parallel within machine precision limits. If this is the case, the DCA is calculated as

$$\mathrm{DCA} = \frac{\|\mathbf{p}_A \times (\mathbf{x}_B - \mathbf{x}_A)\|}{\|\mathbf{p}_A\|}, \tag{3.5}$$

which is easily derived from geometric considerations. Now that the DCA is known, the point of closest approach (PoCA) must be determined on each line, since the point between these two will mark the position of the secondary vertex. In the case of parallel lines, this is easy to compute since one point can be arbitrarily set to the position vector of one of the lines and the other can be computed using geometric considerations:

$$\mathbf{P}_A = \frac{((\mathbf{x}_B - \mathbf{x}_A) \cdot \mathbf{p}_A) \cdot \mathbf{p}_A}{\|\mathbf{p}_A\|^2} \tag{3.6}$$

$$\mathbf{P}_B = \mathbf{x}_B \tag{3.7}$$

with $\mathbf{P}_{A/B}$ being the points of closest approach on the respective line. For the case of skewed lines, the calculation relies on solving the equation

$$\mathbf{x}_B + \omega \cdot \mathbf{p}_B = \mathbf{x}_A + \lambda \cdot \mathbf{p}_A + \mathrm{DCA} \cdot \frac{\mathbf{p}_A \times \mathbf{p}_B}{\|\mathbf{p}_A \times \mathbf{p}_B)\|}. \tag{3.8}$$

This results in in a linear system of equations with two unknown variables λ and ω and three equations with one for each spatial direction, so therefore it is definitely solvable. A possible solution is

$$\mathbf{d} = \mathrm{DCA} \cdot \frac{\mathbf{p}_A \times \mathbf{p}_B}{\|\mathbf{p}_A \times \mathbf{p}_B)\|} + \mathbf{x}_A - \mathbf{x}_B \tag{3.9}$$

$$\lambda = \frac{(p_{B,1}/p_{B,2}) \cdot d_2 - d_1}{p_{A,1} - p_{A,2} \cdot (p_{B,1}/p_{B,2})} \tag{3.10}$$

$$\omega = \frac{p_{A,1} \cdot \lambda + d_1}{p_{B,2}} \tag{3.11}$$

where the subscript i represents i-th component of the vectors. The points of closest approach are then given by

$$\mathbf{P}_A = \mathbf{x}_A + \lambda \cdot \mathbf{p}_A \tag{3.12}$$

$$\mathbf{P}_B = \mathbf{x}_B + \omega \cdot \mathbf{p}_B \tag{3.13}$$

The calculation of the solutions can fail, if either $p_{B,2}$ is zero, or $p_{A,1}$ and $p_{A,2}$ are equal to zero, but otherwise a definite answer is found. Should the very unlikely case happen that, one of these fail conditions is fulfilled, the Equations (3.9) to (3.11) are simply repeated with one other set of spatial directions until one with a possible solution is found.

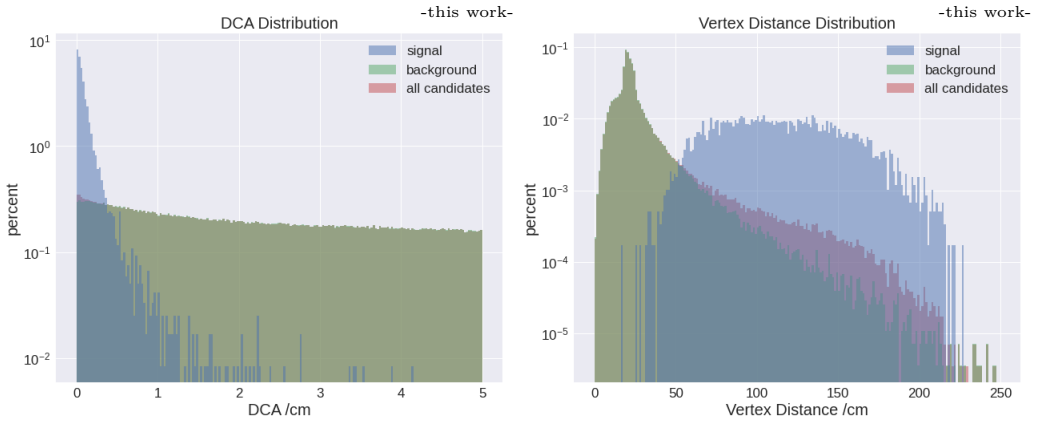
In the V^0 pair finding algorithm, the DCA and PoCAs are computed for each possible pair of V^0s of an event found by the Custom V^0 Finder. If the DCA of two V^0s is less than 10 cm, then a V^0 pair has been found and this pair is a candidate for an anti-sexaquark. In addition, another condition must be met, which is that the V^0s must come from a secondary vertex. The V^0s can only come from the secondary vertex if the V^0 particles are moving away from the SV and therefore the momenta of the V^0s are pointing away from the SV, which is given if in Eq. (3.12) and Eq. (3.13), λ and ω are negative. If both conditions are met, the secondary vertex of the sexaquark candidate is reconstructed as the center between these two points, and its momentum is reconstructed from the momentum vectors of the V^0s. Finally, the anti-sexaquark candidates are stored in a ROOT TTree in a flat format, which contains reconstructed topological information about the reconstructed secondary vertex, the two daughter V^0s and four granddaughter particles, as well as PID information about the granddaughter particles. The tree also contains MC truth information, such as the true PID and the signal or background label, about the

candidates for training and debugging purposes. This tree is then the basis for all further analysis.

3.4.3 Candidate Selection Cuts

The Custom V^0 Finder in combination with the V^0 pair finding algorithm is used to collect candidates suitable to be anti-sexaquarks. After applying both algorithms, 3.64×10^6 anti-sexaquark candidates were found among the 170,000 lead-lead events generated, of which only 4802 candidates are signal. Signal anti-sexaquark candidates are those where both V^0s come from the injected anti-sexaquark nucleon reaction, whereas *true background* candidates are those where neither of the V^0s come from it. In addition, there are numerous candidates where one V^0 is from a sexaquark and the other is not, which are simply referred to as background. Due to the overwhelming imbalance of the signal to background candidates, further cuts have to be made before a classifier can be trained on the data. For this reason, the distributions of several parameters of the dataset were examined and further cuts were applied to three variables. The first is the distance of closest approach (DCA) value, which marks the closest distance between the two propagated momentum lines of the two V^0s. The V^0 pair finding algorithm already applies a 10 cm cutoff to the DCA, but by further restricting this value, large amounts of background candidates can be ignored without significant loss to the signal population.
The DCA distribution is shown in Fig. 3.8a, from which a maximum 1.1 cm cutoff on the DCA was derived. This cut reduces the total number of candidates to 2.32×10^5 while still retaining 4742 of the signal candidates.
The next variable used to reduce the background is the distance of the reconstructed vertex from the primary vertex. Since the anti-sexaquark cannot decay on its own, it must first interact with the detector material and therefore its vertex is placed quite far from the primary vertex compared to the background particles. The vertex distance distribution can be seen in Fig. 3.8b. From this histogram, a lower bound cut was made at 38 cm, which reduces the total number of candidates by over 90 % to 2.97×10^4 and the number of signal candidates to 4730.

A final cut was made to the invariant mass of $\bar{\Lambda}$ and K^0_S, but for this to happen, the vertex of $\bar{\Lambda}$ and K^0_S has to be identified first. The candidates themselves consist of two V^0s, V^0 A and V^0 B, each of which could be the $\bar{\Lambda}$ or the K^0_S vertex. To identify



(a) (b)

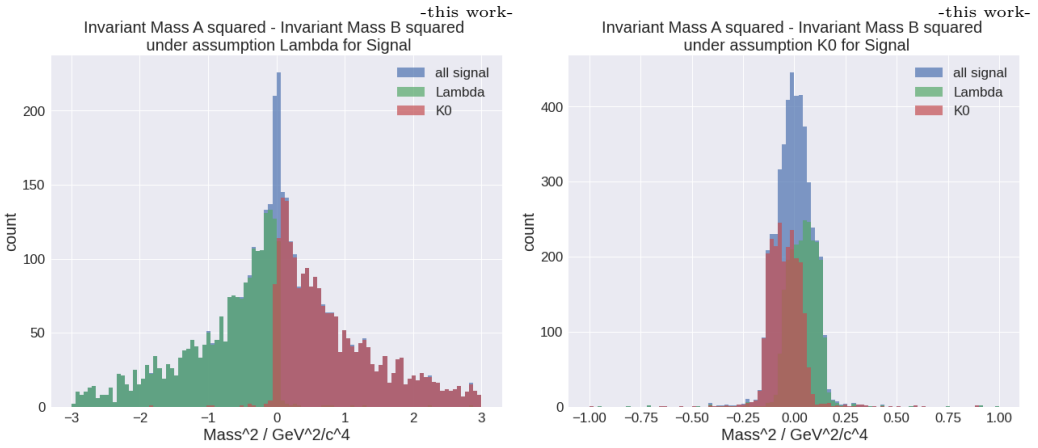
Figure 3.8: Distance of closest approach (DCA) (left) and vertex distance distributions (right) for signal and background candidates. DCA marks the minimal distance between the momentum lines of both V^0s of each candidate. Vertex distance marks the distance of the reconstructed sexaquark vertex at the point of closest approach to the primary vertex. The background candidates are plotted in green and the signal is plotted in blue.

which is which, the invariant mass of both V^0s is computed twice, assuming that it is either a $\bar{\Lambda}$ or a K_S^0. By assuming that both V^0s are lambdas and comparing their reconstructed masses under this assumption, a separation of K_S^0 and $\bar{\Lambda}$ vertices is possible. The invariant masses are reconstructed by taking the measured momenta of the daughter particles of the V^0s and combining them with the theoretical particle masses to calculate the energies of these daughter particles. The energies are calculated by:

$$E^2 = m^2 + p^2, \tag{3.14}$$

where E is the total energy, m the mass and p the momentum of the particle expressed in natural units ($c = 1$). Since K_S^0 decays into π^+ and π^-, and a $\bar{\Lambda}$ decays into π^+ and $\bar{p}$, the only difference is in the negative particle, which can be either a π^- or a $\bar{p}$, while the positive particle must always be a π^+. For this reason, the energy is calculated three times for each vertex: once for the positive particle and twice for the negative particle. Then the four-momenta of the daughter particles can be calculated. The four-momenta of the two daughters can then be summed to give the four-momentum of the V^0 particle, and the invariant mass squared can be calculated with

$$M^2 = E^2 - p^2 = E^2 - p_x^2 - p_y^2 - p_z^2. \tag{3.15}$$

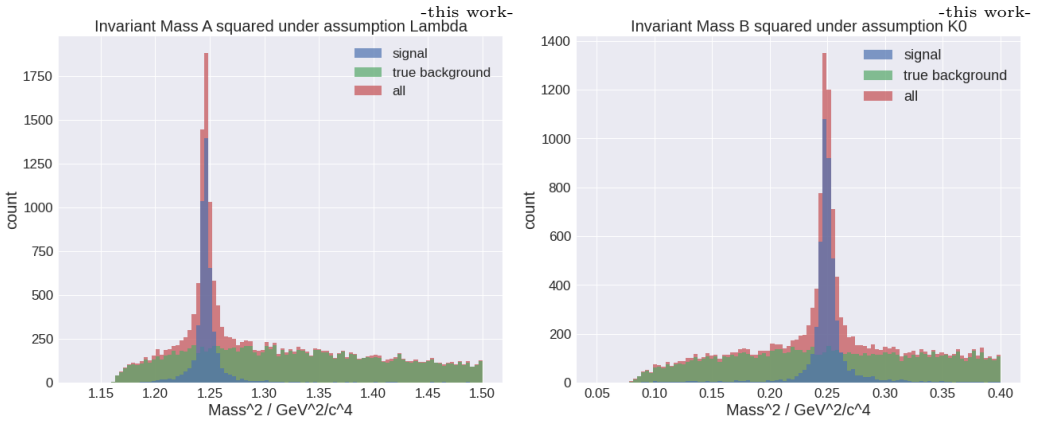


(a) (b)

Figure 3.9: Squared mass difference of V^0 A and V^0 B for signal candidates under the assumption, both particles are $\bar{\Lambda}$ on the left, and under the assumption, both particles are K_S^0 on the right. The cases, where particle A is the $\bar{\Lambda}$ are plotted in green, whereas cases where particle A is the K_S^0 are shown in red. The blue distribution shows both cases together. The overlap between $\bar{\Lambda}$ and K_S^0 cases is small enough on the left plot that it is used to separate $\bar{\Lambda}$ from K_S^0 with high accuracy.

Figure 3.9a shows the difference between the squared invariant mass of V^0 A and V^0 B, assuming that both are $\bar{\Lambda}$, and as can be seen, if this difference is negative, V^0 A was likely the $\bar{\Lambda}$, while if the difference is positive, vertex A was likely a K_S^0. The overlap between the $\bar{\Lambda}$ and K_S^0 distributions is small enough that a sufficient separation is achieved. Of the 4730 signal candidates, 4503 are correctly sorted by this criterion, which is a rate of over 95 %. The same criterion cannot be applied to the K_S^0 invariant masses, since there is no clear separation between the actual K_S^0 and $\bar{\Lambda}$ with their reconstructed masses as K_S^0, but a large overlap region around 0. The squared invariant mass difference of V^0 A and V^0 B, assuming both are K^0, is plotted in Fig. 3.9b.

With the candidates mostly sorted into $\bar{\Lambda}$ in V^0 A and K_S^0 in V^0 B, further selection cuts can be made based on the mass of the V^0 particles. The distribution of the squared invariant mass for V^0 A under the assumption that the particle is a $\bar{\Lambda}$ is plotted in Fig. 3.10a, while the same distribution for V^0 B under the assumption that it is a K_S^0 is plotted in Fig. 3.10b.



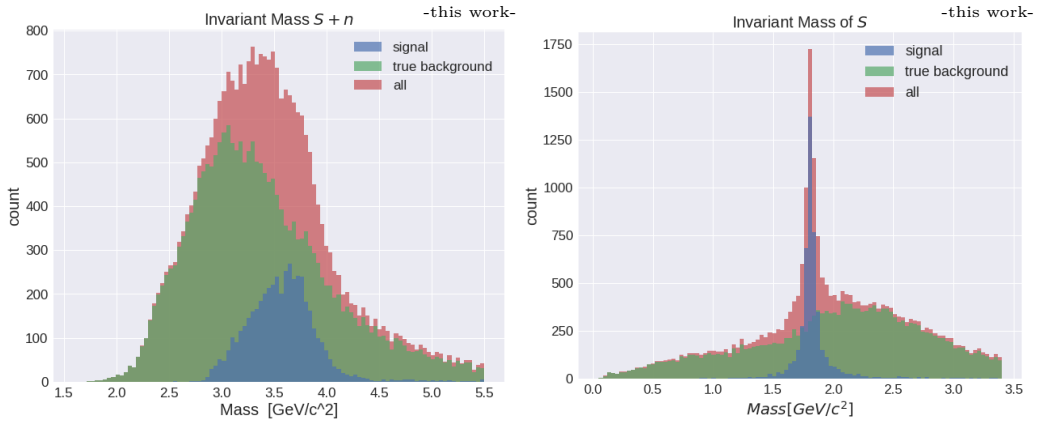
(a) (b)

Figure 3.10: Squared invariant masses of V^0 A and V^0 B under the assumption, that particle A is a $\bar{\Lambda}$ and particle B is a K^0. The plots show the distribution of all candidates in red, the true background (candidates where neither A nor B stem from the sexaquark) in green and the true sexaquark signal candidates in blue. From these plots, cuts are derived to reduce the background.

From these two plots, additional cuts are derived and applied based on the squared invariant masses. For V^0 A, the squared invariant mass is constrained to be below $1.35\,(\mathrm{GeV}/c^2)^2$, which reduces the total number of anti-sexaquark candidates by half while retaining over 98 % of the signal candidates. A further K^0_S squared invariant mass cut is applied, requiring the squared mass to be below $0.35\,(\mathrm{GeV}/c^2)^2$, reducing the total number of candidates by another 40 %, while keeping the total number of signal candidates above 4500. Harsher cuts could be applied, resulting in even higher background rejection with minimal signal loss, but further discrimination of the background will result in less training data for the final XGBoost classifier and even less test data to verify its performance. A comprehensive table of the applied candidate cuts can be found in Tab. 3.3.

	Total	Signal	Background	Signal Reduction (%)	Backg. Reduction (%)
Total	3639689	4802	3511785	-	-
DCA AB $\leq$ 1cm	231929	4742	223623	1.25	93.61
Dist Vert $\geq$ 40cm	29702	4730	22828	0.25	89.79
m_A(as $\bar{\Lambda}$) $\leq 1.35\,(\mathrm{GeV}/c^2)^2$	14461	4661	9800	1.46	57.07
m_B(as K^0_S) $\leq 0.35\,(\mathrm{GeV}/c^2)^2$	8786	4571	4215	1.93	56.98

Table 3.3: Table of applied cuts with signal and background counts and percentage reduction.



(a) (b)

Figure 3.11: Invariant masses of the $\bar{S}$ annihilation (left) and of the pure $\bar{S}$ (right) are depicted. The plots show the distribution of all candidates in red, the true background (candidates where neither A nor B stem from the sexaquark) in green and the true sexaquark signal candidates in blue. The invariant mass of the $\bar{S}$ shows a clear peak at 1.8 GeV, the mass set by the simulation.

From the calculated four-momenta of V^0 A and V^0 B, which are used to calculate their invariant mass, one can also calculate the invariant mass of the anti-sexaquark annihilation by adding their four momenta and using the square root of Eq. (3.15) for the invariant mass. The invariant mass of the annihilation after all applied cuts is plotted in Figure 3.11a. The invariant mass of $\bar{S}$ can then be easily calculated by subtracting the rest energy of the neutron from the energy of the annihilation and again using the square root of Eq. (3.15). The invariant mass of $\bar{S}$ after applying all the cuts is shown in Figure 3.11b. A clear peak at 1.8 GeV can be seen for the signal candidates, which is the mass of the injected anti-sexaquark.

3.4.4 XGBoost Training and Classification Features

The analysis done on the true V^0 sexaquark candidates provided some indications on what are better or worse features for discriminating the signal from the background, and all the features used for true V^0 classification are included in the features used during the custom V^0 sexaquark candidate classification. These features are already explained in detail (see Chapter 3.3.2) and include:

- Armenteros-Podolanski variables:
  - `V0<A/B>_ArmAlpha` - Armenteros-Podolanski variable α for V^0 A/B.
  - `V0<A/B>_ArmPt` - Armenteros-Podolanski variable q_T for V^0 A/B.
- Positional features:
  - `V0<A/B>_Dist` - Distance between the primary vertex and V^0 A/B .
  - `Dist_Vert` - Distance between secondary vertex and primary vertex.
  - `Dist_V0<A/B>_Vert` - Distance between V^0 A/B and secondary vertex.
- Momentum-related features:
  - `V0<A/B>_Transv_Mom` - Transverse momentum of V^0 A/B.
  - `Opening_Angle` - Opening angle between the momenta of V^0A and V^0B.

In addition to these features, thirteen more are used to classify the candidates of custom V^0s. The first is the already discussed DCA value, which can be used with the custom V^0s, since it does not automatically indicate whether two V^0s come from a common source, as it does for true V^0 candidates. Then eight features are used for PID information, which are given as the $n\sigma$ values of the final state particles to be either a pion or a proton: `V0<A/B>_<Pos/Neg>_NSigmaPion` and `V0<A/B>_<Pos/Neg>_NSigmaProton`. Finally, the last four features are the computed squared invariant masses of V^0 A and V^0 B under the assumption that they are either a $\bar{\Lambda}$ or a K^0_S (`V0<A/B>_Rec_invM_Lamb_2` and `V0<A/B>_Rec_invM_K0sh_2`), which have already been used in the candidate selection cuts and discussed in Section 3.4.3. In total, the classifier uses 25 features that provide information about positions and distances, momenta and opening angles, as well as Armenteros-Podolanski variables and nσ information. In addition to good and sufficient features, successful classification requires the right settings, or in the case of machine learning algorithms, good hyperparameters. The search for appropriate hyperparameters performed during this thesis is described in the following section.

3.4.5 XGBoost Hyperparameter Tuning

Hyperparameter optimization is a staple in any machine learning application, and this thesis is no different. Hyperparameters control how a machine learning classifier learns and behaves, and must be fine-tuned to ensure optimal operation of the classifier.

XGBoost Hyperparameters

The XGBoost library utilizes numerous parameters to govern the learning of the classifier. A comprehensive enumeration of the parameters to be optimized and a short description is given below:

- `n_estimators` [default = 100] - Number of gradient boosted trees and training rounds.
- `max_depth` [default = 6] - Maximum depth for tree learners.
- `grow_policy` [default = depthwise] - Depthwise or lossguide. Lossguide seeks out splits, which promise the greatest loss reduction, whereas depthwise favors splits near the root of the tree.
- `eta` [default = 0.3] - Learning rate.
- `gamma` [default = 0] - Minimum split loss. Sets the required reduction of the loss to justify another cut.
- `scale_pos_weight` [default = 1] - Regulates the balance between positive and negative weights. Usually set manually to the imbalance of the training sample: `scale_pos_weight` $= \frac{\sum \text{negative instances}}{\sum \text{positive instances}}$.
- `alpha` [default = 0] - L1 regularization term.
- `lambda` [default = 1] - L2 regularization term.
- `eval_metric` [default = logloss] - Evaluation metric used in estimating current tree performance. Chosen between `rsme`, `logloss`, `auc`, and `aucpr`.

Going through the list from top to bottom, `n_estimators` defines the number of trees the final classifier will have, and since exactly one tree is added per training round, it also defines the total number of training rounds performed. The `max_depth` parameter defines the maximum distance a leaf node can have from the root and the `grow_policy` parameter defines how new nodes are added to the tree. A depthwise grow policy will place new splits closest to the root of the tree, thus growing the tree layer by layer, while a loss-guided grow policy will estimate the node with the highest potential to reduce the loss function and try to split there. This could result in rather elongated trees, while depthwise growth guarantees wide trees. The learning rate is controlled by the parameter `eta` and defines the step size shrinkage between boosting rounds. Each new tree has its feature weights reduced by `eta`, so that each new tree has slightly less influence on the final classification than its predecessor, which is used to reduce overfitting. The `gamma` parameter sets a thresh-

old in expected loss reduction that each new split must meet. If a new split does not reduce the loss of the classifier enough, it is discarded, which is also used to prevent overfitting. For unbalanced datasets, the `scale_pos_weight` parameter can be defined to try to mitigate this imbalance. The `scale_pos_weight` parameter is a factor applied to the prediction of positive training samples, and increasing or decreasing it will increase or decrease the impact of false and true positive predictions, respectively. Usually, this parameter is set beforehand to the imbalance of the data set `scale_pos_weight` $= \frac{\sum \text{negative instances}}{\sum \text{positive instances}}$, but it can also be used to increase the priority of correctly identifying one class over the other. The hyperparameters `alpha` and `lambda` are used to set the L1 and L2 regularization terms, respectively. The L1 regularization is known as *lasso regression*, and the L2 regularization is also known as *ridge regression*. In lasso regression, the absolute magnitude of the weights is added to the loss function, scaled by the value `alpha`, while in ridge regression, the square of the weights is added to the loss function, scaled by the value `lambda`. Assuming that the loss of a regression is given by the mean square error (MSE), the combined loss function for ridge and lasso regression is given by:

$$\mathcal{L} = \sum_{i=1}^{N} (y_i - \hat{y}(x_i; \beta))^2 + \alpha \sum_{j=1}^{m} |\beta_j| + \lambda \sum_{j=1}^{m} \beta_j^2 \tag{3.16}$$

where N is the number of training samples, y_i is the true value of a function to be estimated, $\hat{y}$ is the regression function that estimates y_i based on the independent variable x_i and its weights β_j. The first term of Eq. (3.16) is the squared mean error, the second term is the lasso or L1 regularization term, where the absolute value of the weights is summed and scaled by the parameter α, and the last term is the ridge or L2 regularization term, scaled by the parameter λ, where the squared weights are summed and added to the loss function. Adding this regularization term reduces overfitting of a regressor or classifier. The L1 regularization tends to shrink the weight coefficients to zero, resulting in the dropping of less influential features. The L2 regularization, on the other hand, does not eliminate features, but rather asymptotically reduces the weight of features. A good balance of both regularization terms can prevent overfitting, eliminate unimportant features, and keep the weights in an appropriate range for good fitting and classification. Finally, the `eval_metric` parameter can be set to different evaluation metrics, although for this thesis we will choose between three of them: `rsme`, `auc` and `aucpr`. These evaluation metrics do not directly affect the training or classification performance, but are used to evaluate how well the classifier performs, so understanding their performance measurement

is still crucial for evaluating the goodness of the classifier. *RMSE* is an acronym for root mean square error and is given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}. \tag{3.17}$$

RMSE is a common error metric for regression and a staple for machine learning applications. *AUC* is an acronym for area under curve, where the curve is the ROC curve. The area under this curve is then used as a metric, where 1 is the highest possible value, meaning that the classifier is perfect and able to correctly classify all test samples. While a value of 0 is technically achievable, the lowest practical value of AUC is 0.5, which makes this classifier as good as randomly guessing the classes for the test samples. Values below 0.5 are obtained when the classifier has a higher probability of putting a sample in the wrong class than in the right class, at which point you can simply invert the classification to get a AUC value better than 0.5 again. *AUCPR* is the area under the precision-recall (PR) curve, in which the precision is plotted against the recall of a classifier. Precision is a measure of how many of our positive classifications are true, and recall is a measure of how many of the tested positive values were classified as such. In the formula, precision and recall are given by:

$$P = \frac{TP}{TP + FP} \tag{3.18}$$

$$R = \frac{TP}{TP + FN} \tag{3.19}$$

where P and R are the precision and recall respectively, TP is the number of true positives, FN is the number of false negatives, and FP is the number of false positives. Both the ROC and the PR curve are good measures of a classifier's performance and are often used interchangeably. The main difference is that the PR curve focuses more on precision. In the context of classification and optimization, these metrics are used for early stopping in training and pruning in optimization, which has an impact on performance and therefore needs to be part of the optimization parameters. Early stopping is the practice of prematurely stopping training after no improvement is seen on a validation set for a certain number of training rounds. Pruning is a practice where a set of parameters is discarded early if the classifier is already massively behind a classifier with a previous set of parameters early in the training process.

In addition to the parameters discussed above, XGBoost has a number of operational parameters that need to be defined but do not affect the classification result. These parameters include the number of cores XGBoost is allowed to use, the type of booster it should produce, how many output classes there are, and how verbose the program should be during training.

Hyperparameter Tuning Techniques

In the previous section, we discussed the different hyperparameters of XGBoost that need to be optimized. Now we will look at how to optimize them. For the optimization of hyperparameters, several techniques have been developed, which can mostly be divided into three categories: Grid Search, Random Search, and Bayesian Search. All searches try to find a set of parameters $\mathbf{x}$ within a user-defined parameter space that minimizes a given objective or loss function $f(\mathbf{x})$:

$$\mathbf{x}_{\text{opt}} \in \arg\min_{\mathbf{x} \in X} f(\mathbf{x}). \tag{3.20}$$

Grid Search divides the parameter space into a grid of equidistant points and tries each point in turn. While this search is time and resource consuming, it is thorough and guarantees good parameters as long as the possible space is large enough and the distance between points is small enough. This method can quickly become expensive in high-dimensional parameter spaces. Random Search, on the other hand, relies on randomly choosing parameters to test, and given enough guesses, will inevitably find good parameters, while still being computationally less expensive than grid search, but does not guarantee that all corners of the parameter space have been explored, or that the best found set is a minimum or even close to a minimum. Bayesian Search starts with random guesses similar to random search, but tries to learn based on previous attempts to suggest better parameters, resulting in good parameters with drastically reduced optimization time, hence it is heavily used in optimizing larger models. In this thesis, a Bayesian hyperparameter search is performed using the *Optuna* framework, and its working principle is explained below. Optuna is an open source project published in 2019 [54]. It stands out for its ease of use and setup, define-by-run principle, and efficient sampling and pruning algorithms that make it a state-of-the-art hyperparameter search algorithm. It is able to employ a number of hyperparameter optimization techniques such as random search and grid search, but its standard go-to technique is a Bayesian search method called the Tree-structured

Parzen Estimators (TPE) algorithm. To find the set of parameters that satisfy Eq. (3.20), Bayesian optimization algorithms employ an acquisition function that attempts to guide the algorithm to search near good observations (also known as exploitation) rather than searching in previously unseen parameter regions (also known as exploration). TPE uses the probability of improvement

$$\mathbb{P}(y \leq y^\star | \mathbf{x}, \mathcal{D}) := \int_{-\infty}^{y^\star} p(y|\mathbf{x}, \mathcal{D}) dy, \tag{3.21}$$

which gives the probability that, given a set of parameters $\mathbf{x}$ and previous observations $\mathcal{D}$ consisting of parameter sets-objective pairs $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, the objective y is lower than an objective threshold $y^\star$ to be specified by the user or the algorithm. To compute new sets of parameters to try, a set of prior observations $\mathcal{D}$ must be made, usually using a few rounds of optimization. To estimate $p(y|\mathbf{x}, \mathcal{D})$, the Bayes rule is applied first:

$$p(y|\mathbf{x}, \mathcal{D}) = \frac{p(\mathbf{x}|y, \mathcal{D})\, p(y)}{p(\mathbf{x})}, \tag{3.22}$$

and to estimate $p(\mathbf{x}|y, \mathcal{D})$, the previous observations are first separated into two groups $\mathcal{D}^{(l)}$ and $\mathcal{D}^{(g)}$. $\mathcal{D}^{(l)}$ is the group of observations where $y_n \leq y^\gamma$, where y^γ is the top γ quantile, and $\mathcal{D}^{(g)}$ are the remaining observations. In other words, the previous observations are separated into better and worse observations based on a certain quantile. Then $p(\mathbf{x}|y, \mathcal{D})$ is assumed:

$$p(x|y, \mathcal{D}) = \begin{cases} p(x|\mathcal{D}^{(l)}) & \text{if } y \leq y^\gamma, \\ p(x|\mathcal{D}^{(g)}) & \text{if } y > y^\gamma. \end{cases} \tag{3.23}$$

These two distributions, $p(\mathbf{x}|\mathcal{D}^{(l)})$ and $p(\mathbf{x}|\mathcal{D}^{(g)})$ are estimated using Parzen estimation, better known as Kernel Density Estimation (KDE), over each of the two sets of observations. Then, the probability of improvement from Eq. (3.21) is proportional to the density ratio of $p(\mathbf{x}|\mathcal{D}^{(l)})$ and $p(\mathbf{x}|\mathcal{D}^{(g)})$:

$$\mathbb{P}(y \leq y^\star | \mathbf{x}, \mathcal{D}) \propto \frac{p(\mathbf{x}|\mathcal{D}^{(l)})}{p(\mathbf{x}|\mathcal{D}^{(g)})}. \tag{3.24}$$

To choose the next set of parameters, one then selects a few parameter sets $\mathbf{x}_s$ drawn from $p(x|\mathcal{D}^{(l)})$ and chooses the one that satisfies

$$\mathbf{x}_{N+1} := \arg\max_{\mathbf{x} \in S} \frac{p(\mathbf{x}|\mathcal{D}^{(l)})}{p(\mathbf{x}|\mathcal{D}^{(g)})}. \tag{3.25}$$

With this new set of hyperparameters, the (potentially expensive) objective function $f(\mathbf{x}_{N+1})$ can be evaluated. Finally, the new pair of parameters and observations is added to the set of observations and the process is started again by dividing the observations into better and worse groups. For the interested reader, a deeper dive into the mathematics and selection criteria for the quantiles and thresholds can be found in Ref. [55]. While the search for the next set of parameters in TPE may be more complicated compared to grid or random search, it makes up for it with better predictions of good sets of parameters to try, resulting in fewer overall tests and fewer objective function evaluations. In most cases, calling the objective function requires training a model with the newly found hyperparameters, which can be very time consuming. Optuna reduces the time even further with its built-in pruning techniques, which allow to preemptively abort the training of a model as soon as it becomes foreseeable that a current set of parameters is outperformed by a previous set. The combination of Optuna's TPE and pruning algorithms, as well as its ease of setup and applicability to all machine learning tasks, make it the optimal choice for this thesis.

Optuna's Hyperparameter Tuning Objective Function

As mentioned in the previous section, the goal of hyperparameter tuning is to find a good (possibly the best) set of hyperparameters with which to train a classifier. To get a good set of parameters, one first has to know what "good" means, and then one has to translate this knowledge into a quantitative measure that the program can understand. For this, an appropriate objective function has to be constructed. Ideally, this objective is physically motivated and leads to a classifier that better distinguishes between signal and background candidates. Such an objective function is represented by the "Punzi criterion", introduced by Giovanni Punzi in 2003 [56]. The criterion is derived from the sensitivity of a search experiment with Poisson-distributed observables and is given as

$$P = \frac{\eta(t)}{a/2 + \sqrt{B(t)}}, \tag{3.26}$$

where η is the signal efficiency as a function of a possible set of cuts t, a is the desired number of sigmas corresponding to a two-tailed Gaussian test at significance level α, and $B(t)$ is the number of expected background events as a function of the

possible set of cuts t. The possible set of cuts in this case is simply the threshold on the output of the XGBoost classifier above which a candidate is considered a sexaquark, and the value of a is set to 3, since Punzi shows that lower numbers tend to generalize better to the Poisson distributions. Besides its good physical motivation, the Punzi criterion has many advantages over other optimization objectives, such as being independent of the cross section of the searched process, since only the signal efficiency is required. Another widely accepted measure to decide how decisive a measurement of a signal peak is compared to the background is the significance, which is defined as :

$$Sig = \frac{S}{\sqrt{S+B}}, \tag{3.27}$$

with S/B represent the number of signal and background candidates remaining after applying all cuts. The significance is given as the size of the σ intervals. Maximizing this function means maximizing the number of signal counts while minimizing the number of background counts, and in principle this would be applicable in this form, but it would not represent the real world scenario, since in the simulation each event contains a sexaquark that also interacts with the detector. This is a rather unlikely event, so the signal to background ratio would be biased, and to counteract this each signal count has to be multiplied by the expected production times the interaction probability of the sexaquark:

$$Sig = \frac{p_{\text{interacted } \bar{S}} \cdot S}{\sqrt{p_{\text{interacted } \bar{S}} \cdot S + B}}, \tag{3.28}$$

where $p_{\text{interacted } \bar{S}}$ is the probability that a anti-sexaquark is created in a collision and subsequently annihilated with the detector material. All these values could be estimated for the sexaquark, but the estimation comes with large uncertainties, a problem that does not arise with the Punzi criterion. Furthermore, the Punzi criterion does not have the problem of breaking down at low values of B, like other common “significance-like” optimization functions such as

$$Sig = \frac{S}{\sqrt{B}}. \tag{3.29}$$

Maximizing the Punzi criterion function then becomes the objective against which the performance of the hyperparameter tuning algorithms is measured. For the Punzi criterion (or any other objective function), the threshold of the classifier must be determined. The XGBoost classifier returns a value between 0 and 1 for each candidate, representing how confident the classifier is that a given candidate is signal,

where 0 means it is confident to not be signal and 1 means it is confident it is signal. All candidates fall on a distribution between these two values, with some background candidates having rather high prediction scores and some signal candidates having rather low prediction scores. This means that a threshold has to be set where everything above this threshold is declared to be signal and everything below is declared to be background, and the placement of this threshold has a strong impact on the significance. The fourth advantage of the Punzi criterion is that it is not fixed at a certain value (e.g. 0.5), but the optimal threshold is defined as the one that maximizes the Punzi criterion.

Optuna's Hyperparameter Tuning Results

This section discusses the results of the hyperparameter tuning. Optuna optimization was performed with maximization of the Punzi criterion as the objective (see Section 3.4.5) with a total of 500 trials, of which 25 were used as warm-up steps. During each iteration, a new classifier is trained on a newly shuffled dataset, and a 3-fold cross-validation was used to eliminate outliers. The k-fold cross-validation is done by partitioning the data set into k parts, followed by k training rounds, where $k-1$ parts are used for training and 1 part for testing, and then the result is averaged over all k trials. During optimization, the pruning options of Optuna are not used, since higher objective function results were obtained without them. After optimization, a maximum Punzi value of 0.1867 is achieved with the following parameters:

`Best trial:`	
`Value:`	0.207
`Params:`	
`eval_metric:`	rmse
`lambda:`	1.468×10^{-7}
`alpha:`	0.708
`eta:`	0.035
`scale_pos_weight:`	0.245
`n_estimators:`	129
`max_depth:`	16
`gamma:`	1.162×10^{-4}
`grow_policy:`	lossguide

Table 3.4: Table showcasing the hyperparameter optimization result.

Looking at the individual parameters, we can see that many of them differ significantly from the default values, with the most noticeable change being the learning rate `eta`, which is set to a very low value, resulting in a slow improvement during training, but increasing the chances of accurately finding a local minimum of the loss function during training. The optimized classifier has almost 30 % more trees, while being much deeper with a maximum depth of 16 compared to the default 6. Furthermore, the L2 regularization term `lambda` has been set nearly to 0, which is also the case for the minimum split loss `gamma`, but the L1 regularization term `alpha`, which is set to 0 by default, now has a rather high value of 0.7. In general, one can say that the optimized forest prefers to fully classify its training sample with deep and extensive trees with less regularization. As expected, `scale_pos_weight` is also set to a smaller value, which then favors the correct identification of background over signal, and thus leads the classifier to higher purities, accepting potential losses in efficiency. In Fig. 3.12 one can see the optimization history. The history shows the achieved value of the current best trial as a red line, while the individual trials are marked as blue dots. At the beginning of the history one can see the warm-up steps with comparatively low values at the beginning and a fast jump after the warm-up steps. After 192 trials the maximum was found with the current best objective value of 0.207. During the optimization, the built-in pruning is not used, since repeated optimizations showed that higher objective function values are achieved without it. The importance of each feature is shown in Fig. 3.13, which shows which features have the most impact on the objective. The histogram shows that the tree depth `max_depth` has the largest impact, since it allows a very deep classification of the training data. It is followed by the L1 regularization term `alpha` and then the parameter `scale_pos_weight`. The `alpha` parameter is generally responsible for removing unnecessary features from the feature space, and `scale_pos_weight` is important for correctly identifying background over signal. Next are the parameters responsible for the number and depth of the trees `n_estimators` and the learning rate `eta`. At the bottom of the list are the regulatory parameters that do not have much impact, such as the growth policy or the evaluation metric (which is not actively used since pruning and early stopping are disabled), followed by the L2 regularization term `lambda` and minimal split loss `gamma`, which are simply disabled.

With optimized hyperparameters, the custom V^0 dataset, and features to perform classification, training and quantitative testing of the XGBoost classifier could begin.

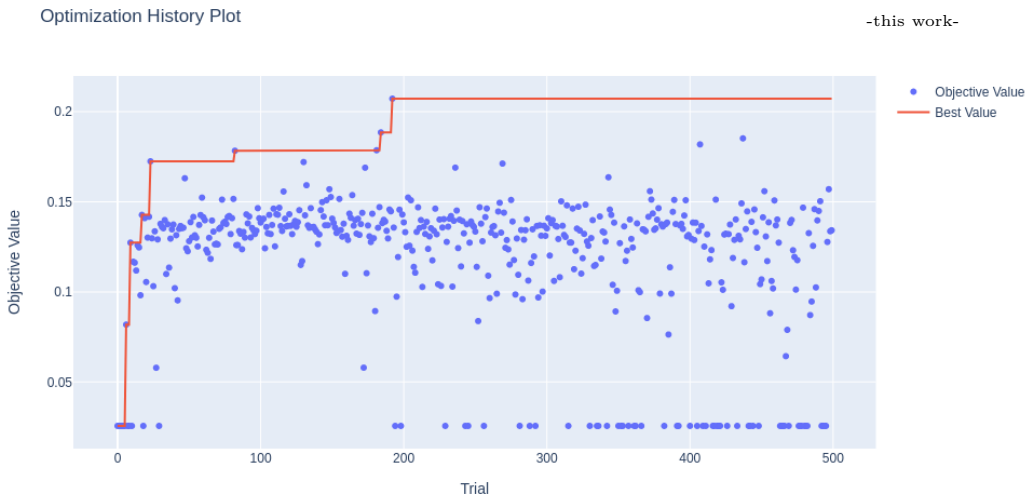


Figure 3.12: Optimization history of Optuna parameter optimization. The objective function value is plotted as a function of trials or optimization rounds. One can see the warm-up steps using random search at the lower end of the number of trials, until gaussian optimization kicks in after a couple of trials.

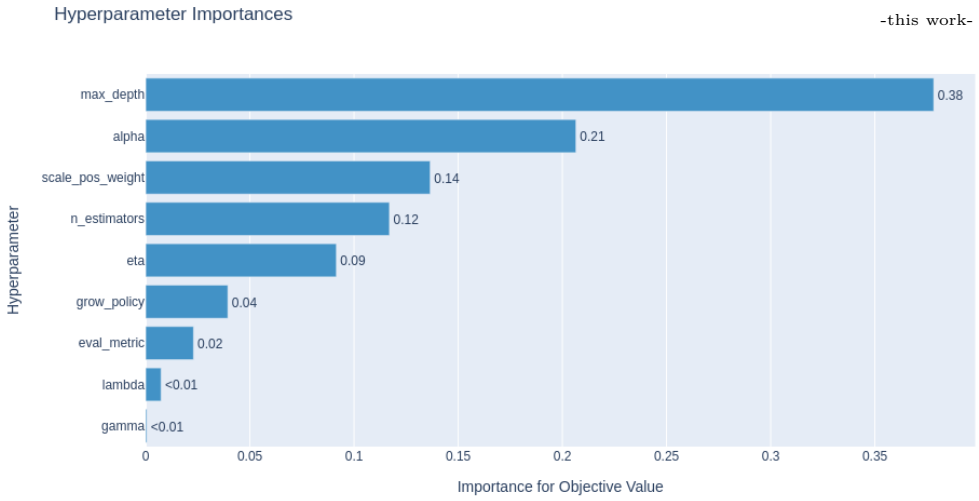


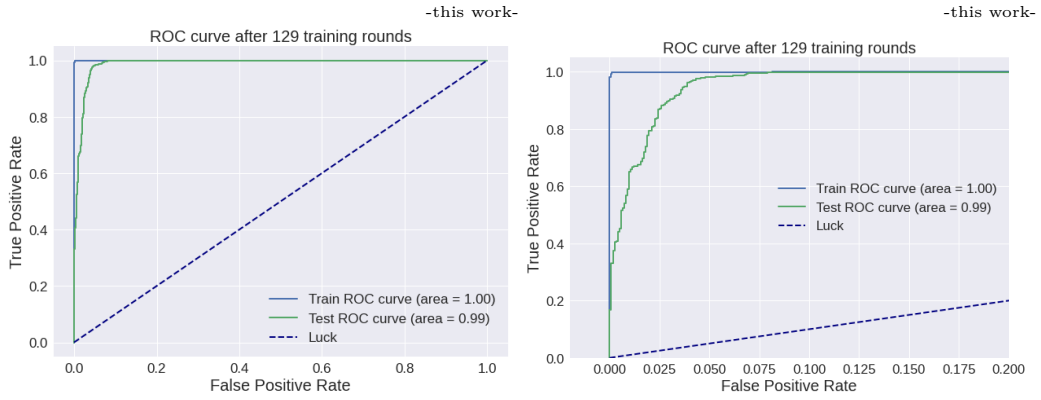
Figure 3.13: Parameter importance plot of Optuna hyperparameter optimization. The plot shows how much impact each of the optimized hyperparameters has on the result of the objective function.

3.4.6 XGBoost Classifier Training and Results on Pb–Pb Collision Simulations with Custom V^0s

This section discusses the results of training the XGBoost classifier. The classifier is trained and tested on the 8786 candidates left after applying all selection cuts (see Section 3.4.3), of which 4571 are signal candidates and 4215 are background candidates. The test to training ratio was set to $30 : 70$, resulting in 2636 test candidates, of which 1368 are signal candidates. The classifier was trained with the hyperparameters determined in Section 3.4.5, and then the performance of the classifier is evaluated. For this purpose, the ROC curve, the logarithmic loss and the feature importance of the classifier were determined.

ROC Curve & Logloss The ROC curve is shown in Fig. 3.14a with a zoomed version in Fig. 3.14b. Contrary to the true V^0 case, a clear distinction between the test and training curves can be observed, but the classifier still performs extremely well on the test sample with an AUC value of 0.99, while the training curve is still nearly perfect. One can also see that the difference from the perfect case is not symmetric, but rather shifted to the left, which shows the preference of the classifier to trade the true positive rate for a better false positive rate as a result of the optimization. The training history of the classifier can be seen in Fig. 3.15, where the AUC value of the ROC curve is plotted on the left and the logistic loss (logloss) is plotted on the right. From the AUC value, one can see that compared to the classification of true V^0s (see Fig. 3.7), the initial AUC as well as the final AUC are much lower for both training and test data. The training data eventually reaches an area of 1, and the final AUC for the test set is around 0.993. The logloss plot shows a steep decline in the logarithmic loss in the early training rounds with a flattening of both curves towards the final training rounds. The 129 training rounds are not enough to reach a final plateau for the training samples, but for the test samples the beginning of a plateau is reached at a logloss value of about 0.11.

Feature Importance The feature importance is plotted in Fig. 3.16, with XGBoost's built-in feature importance plot on the left in Fig. 3.16a and Shapley importance on the right in Fig. 3.16b. The built-in method simply counts how often a feature is used in a cut across all decision trees and uses that as a measure



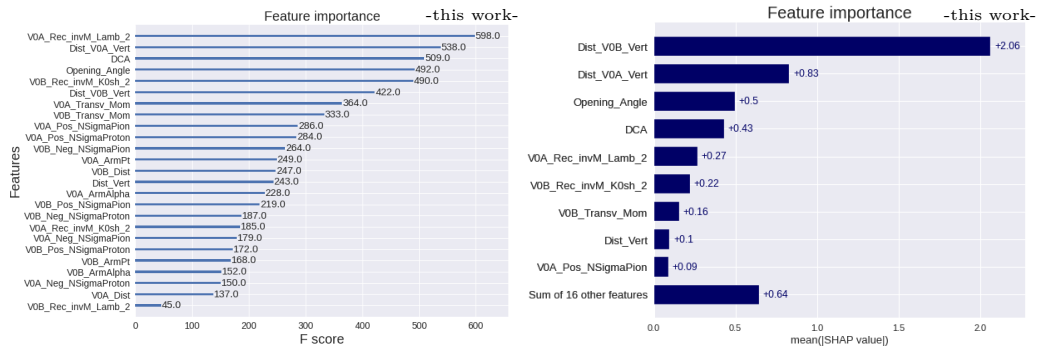
(a) (b)

Figure 3.14: ROC curve of the XGBoost classifier used on the custom V^0 data sample. On the left, the whole ROC curve is drawn and on the right, a zoomed in version of the same curve is depicted. The test curve is drawn in green and the training curve is drawn in blue. The dark blue diagonal represents how a luck-based classifier which works by random guesses of the classes would perform.

(a) (b)

Figure 3.15: Training history of the XGBoost classifier on True V^0 candidates. On the left, the AUC metric and on the right, the logloss is shown. The scores of the training set are plotted in blue and the scored of the test samples are shown in green.

of importance. You can see that the most frequently used features include `DCA`, `Dist_V0<A/B>_Vert`, `opening_angle`, as well as the reconstructed squared invariant masses `V0A_Rec_invM_Lamb_2` and `V0B_Rec_invM_K0sh_2`, while their counterparts `V0B_Rec_invM_Lamb_2` and `V0A_Rec_invM_K0sh_2` are at the lower end of the distribution. This is particularly interesting since the invariant masses were used to sort the candidates so that V^0 A most likely contained $\bar{\Lambda}$ and V^0 B most likely contained K^0_S, which the classifier picked up and focused on only those masses, ignoring the other two. The transverse momentum features are also quite high in the ranking, and the PID features mostly mark the middle to lower end of the importance distribution. The number of times a feature is used in a tree is a simple metric for extracting feature importance, but it may not be the best way to do so, since features used in cuts near the root of many trees have a huge impact on the final classification while occurring in small numbers, and therefore Shapley values are often used to determine importance. Shapley values were first introduced by Lloyd Shapley in 1951, for which he received the Nobel Prize in Economic Sciences in 2012 [57]. Shapley values are a concept in cooperative game theory for determining the contribution of each player to the outcome of the game. In the case of classifiers, Shapley values measure how large the impact of each feature on the final classification score is, as shown in Fig. 3.16b. We can see that the features describing the distance of the V^0s to the secondary vertex have by far the largest contribution to the score, followed by the opening angle between the V^0 momenta. The DCA is in fourth place, followed by the invariant masses, then a transverse momentum, the distance of the secondary vertex to the primary vertex, and the $n\sigma_\pi$ value of the positive trace of V^0 A. The remaining features are summed up at the bottom, but their contribution cannot simply be dismissed, since together they have a higher Shapley value than the opening angle feature. Comparing the two measures of feature importance, one can see similar features in similar places, but simply counting the importance of V^0 to the secondary vertex distance greatly underestimates it. Additionally, a Shapley beeswarm plot is shown in Fig. 3.17, which shows how different feature values affect the classification. Negative Shapley values mean that the classification of a sample is moving towards label 0 (background), and a positive Shapley value means that the classification is moving towards label 1 (signal). The value of the sample compared to other samples is visualized by the color of the dot. The plot shows that lower values for the distances and opening angles between the V^0s and the secondary vertex are more likely to be classified as signal, while higher values are more likely to be classified as background. For the invariant masses, average values are preferred for signal candidates, while lower and higher values are more likely to



(a) (b)

Figure 3.16: Feature Importance plot of XGBoost classification. The left plot is the in XGBoost integrated feature importance plot and shows how often each of the features is used as a cut during classification. The right plot uses the Shapley values to determine important features, which give an indication how big the contribution of a certain feature to the final classification was.

represent background values. For the features `Dist_Vert` and `V0B_Transv_Mom` we see that higher momenta and further away from the primary vertex are preferred for signal candidates.

Working Point Determination A crucial part of a classification task is the determination of the operating point, which represents a threshold above which candidates are classified as signal and below which they are classified as background. This threshold determines how many background candidates will survive and how much signal will be ignored, and setting an appropriate value requires a trade-off between the two. To this end, the relative and absolute numbers of surviving signal (true positive) and true background (false positive) candidates were plotted as a function of threshold, as shown in Fig. 3.18. For each plot, the data set was shuffled and the classifier was retrained 20 times from scratch. The faint blue and red lines show individual training, while the thick lines represent the average of all trials. It can be seen that the true positive rates/counts hardly differ from each other, while the false positive counts/rates show comparatively large variations, which is mainly due to the logarithmic scale, since small deviations on such a small scale have a more noticeable effect compared to the large scale of the true positive rates/counts. Furthermore, the false positive counts show a large threshold range at which no false positive candidate survives, with some trials having no false positives after a thresh-

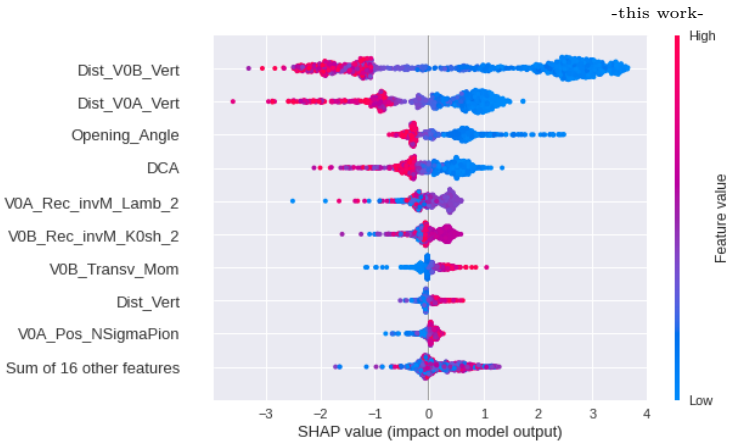
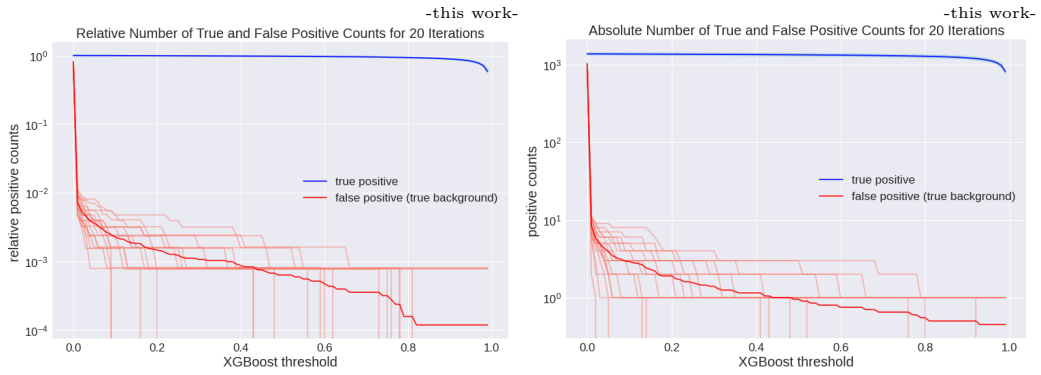


Figure 3.17: Shapley value beeswarm plot of XGBoost classifier. The plot shows how high and low values of a feature impact the classification.

old of 0.2, while other trials still have false positives at a threshold of 0.99. The main reason for this is a handful of convincing looking background candidates that achieve high XGB values when they are present in the test dataset, which happens in 30 % of the cases for each individual candidate.

In addition to the plots in Fig. 3.18, which show the signal and background counts as a function of threshold on the XBG score, the number of surviving backgrounds is plotted as a function of signal efficiency, which is shown in Fig. 3.19. For the calculation, 100 classifiers are trained, and for a threshold corresponding to different signal efficiencies, the background counts are measured and the mean and standard deviation of the error are calculated. The plot shows a plateau below a signal efficiency of 0.8 and an exponential increase above 0.8. From the background counts, the significance can be calculated using Eq. (3.28), where the number of sexaquarks is scaled by the estimated anti-sexaquark interaction probability calculated in Chapter 4. The significance can be seen in the Fig. 3.20a. Alternatively, the Punzi criterion can be plotted, as shown in Fig. 3.20b, and used to determine the working point. Comparing the two, it can be seen that both have a similar shape, but differ slightly at the maximum. The maximum of significance is around a signal efficiency of 0.85 with a rather flat distribution around the maximum, ranging from 0.82 to 0.9. The Punzi distribution has a sharper maximum at an efficiency of 0.9. Following the arguments of [56], the choice of the working point is determined by the Punzi criterion, which corresponds to an average threshold of 0.793 and an average true



(a) (b)

Figure 3.18: Absolute and relative numbers of true and false positive counts for the XGBoost classifier on custom V^0 data. The classifier is retrained on a shuffled data set 20 times, with each time being represented by a faint blue line for true positives and a faint red line for false positives. The thick lines represent the averages. The plot on the left shows the number of counts normalized by the total number of true background/signal candidates, respectively and the plot on the right shows the absolute number of counts.

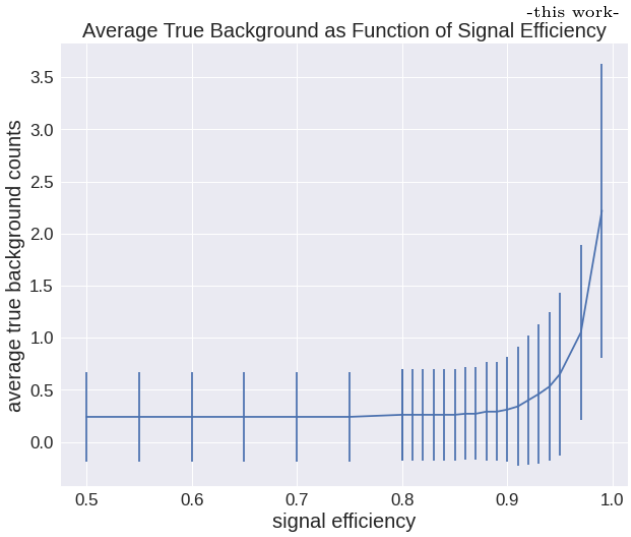


Figure 3.19: The average true background counts are depicted as a function of the signal efficiency. The counts were averaged over 100 reshuffled training rounds and the error is given by the standard deviation of these counts.

(a) (b)

Figure 3.20: Significance and Punzi criterion as a function of the signal efficiency. The values were determined in 100 training rounds with reshuffled data set and the error is given by the standard deviation.

background count of 0.33 ± 0.47. An example of the posterior distribution of the XGBoost classifier with the determined threshold is shown in Fig. 3.21.

3.4.7 Analysis of Surviving Background Candidates

Each individual candidate is assigned an identifier code, allowing its origin and thus its MC truth information to be traced. The candidate ID is a 16-digit integer number where the first six digits represent the run number to which the current simulation is anchored, the seventh digit represents the save file, where the candidate is stored from 0 to 3, and the following three digits are reserved for the event number. The last six digits correspond to the V^0 numbers of the candidates, where the first three are used for V^0 A and the last for V^0 B. The candidate numbers are used to investigate the background candidates that survive the XGBoost classifier. During the determination of the surviving background, which was performed by retraining the classifier 100 times, a total of three unique candidates survived the set threshold with a signal efficiency of 0.9. The ID numbers of the surviving candidates are 2460520249011013 (Cand. 1), 2975881230000004 (Cand. 2), and 2968990017008011 (Cand. 3). Cand. 1 is responsible for 89.5 % of the average 0.33 ± 0.47 surviving true background candidates, while Cand. 2 and Cand. 3 account for 7.9 and 2.6 %, respectively. These three candidates are examined using the MC truth information.

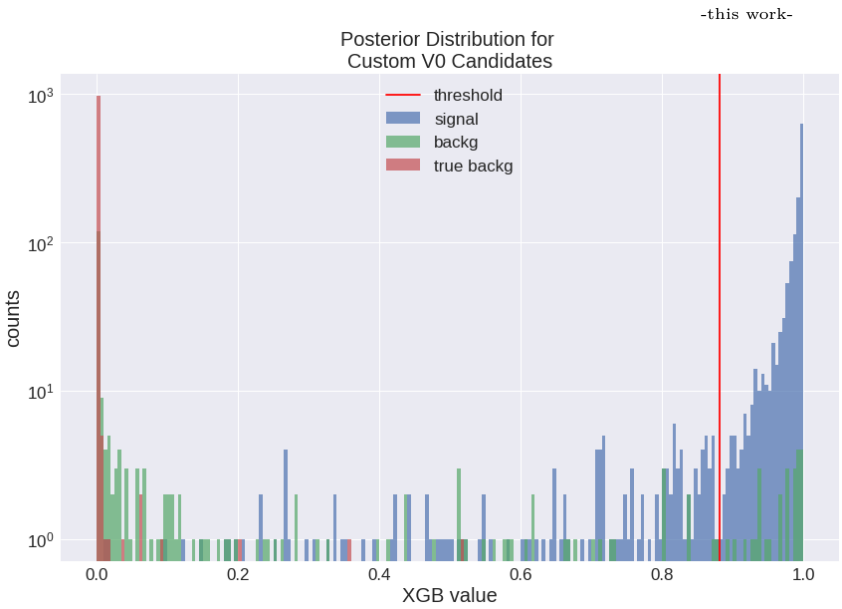
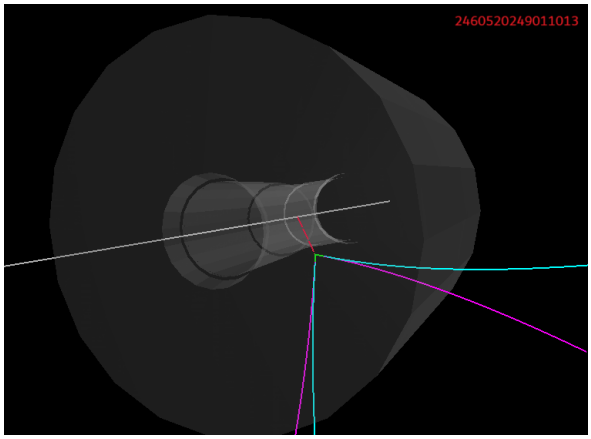


Figure 3.21: Posterior distribution of the XGBoost classifier. The counts of the different classes are shown as a function of the XGB score. Signal candidates are shown in blue, true background candidates in red and mixed background candidates are depicted in green. The determined classification threshold on the XGBoost value is shown in red.

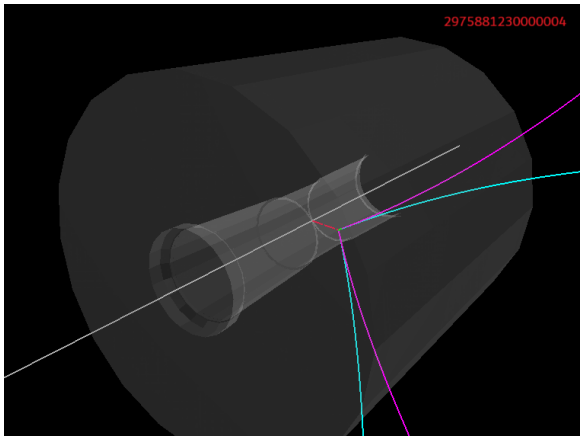
According to the MC truth information, Cand. 1 is a π^0 that decays into two γ. One of these γ produces a $\bar{p}$ and a π^+ after 23 cm, and the other γ produces a $\pi^+\pi^-$ pair 14 cm away from the reconstructed secondary vertex. These particles are in particular the final state particles of the analyzed $\bar{S}$ interaction chain $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0 \rightarrow \bar{p}\pi^+\pi^-\pi^+$. Furthermore, all of the candidate's features are in an ideal range for anti-sexaquark candidates, such as the reconstructed mass of V^0 A, assumed to be a Λ, which is 1.11 GeV and the reconstructed mass of V^0 B, assumed to be a K_S^0, which is 0.496 GeV. Due to the ideal properties, this candidate consistently achieves XGBoost classifier predictions of 0.98 and above. One possible interaction that could lead to such a candidate is a π^0 decay, where the two γ interact with nucleons in the detector to produce a proton, an antiproton, and a number of pions: $\gamma + N \rightarrow N + \bar{p} + p + n \cdot \pi^{+/-}$. From this interaction, the $\bar{p}$ and a π^+ are combined in the measured V^0, and the other produced particles are discarded. According to MC truth information, only the $\bar{p}$ and π^+ are produced from the γ, and the γ does not have enough energy to produce a $\bar{p}$ in the first place, so the exact interaction chain cannot be reconstructed. Further investigation is needed to identify the source of

this interaction and to determine whether it is due to a physical process or an effect of the simulation.

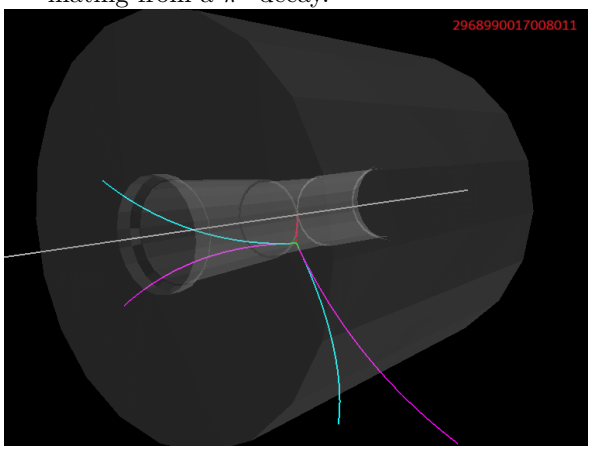
Candidates 2 and 3 both have the same end products, which are $\bar{p}$ and μ^+ at one V^0 and π^+ and μ^- at the other. A picture of the event display of these two candidates can be seen in Fig. 3.22b and Fig. 3.22c. For both candidates the first V^0 is identified as π^+ and the second V^0 is identified as K_S^0 and they do not share a common vertex. According to MC truth information, the K_S^0 comes from the primary particle and the mother of the π^+ is a ρ^0. The exact process of how these particles are created is up for investigation, but regardless of whether the reason is a physical process or a simulation artifact, both are likely caused by the same process due to the identical PID information of the candidates. In general, one can say that these two candidates are combinatorial backgrounds that happened to have all features in appropriate ranges to survive all cuts and achieve high scores in the classifier. Candidate 2 achieves classifier scores below 0.85 and candidate 3 achieves scores below 0.81, which is around the threshold, which is about 0.8. Candidates 2 and 3 achieve lower scores compared to Cand. 1, which means that most of the time they do not survive the XGBoost threshold and therefore only account for 10 % of the averaged surviving background. The reason why they sometimes survive the threshold is that due to the randomization, a particular set of candidates is not included in the training set and therefore the classifier lacks the necessary information to correctly identify them. This also means that with enough training data, these candidates can definitely be eliminated.



(a) Event display of the interaction of Candidate 1. The final state particles of this candidate are $\bar{p}\pi^+\pi^-\pi^+$ created by two interacted γ originating from a π^0 decay.



(b) Event display of the interaction of Candidate 2. The final state particles of this candidate are $\bar{p}\mu^+\pi^+\mu^-$.



(c) Event display of the interaction of Candidate 3. The final state particles of this candidate are $\bar{p}\mu^+\pi^+\mu^-$.

Figure 3.22: Event displays of surviving true background candidates. Positive particles are drawn in purple, negative particles in cyan and neutral particles in green. The pictures of the event displays were provided by Andrés Bórquez.

4 Signal and Background Estimations

In this chapter, calculations are made to estimate the number of sexaquarks produced in real collisions, as well as the interaction cross section of the sexaquark. These are then combined with measurements of the reconstruction efficiency of the sexaquark in Pb–Pb collision simulations to estimate the total number of sexaquarks that we should be able to find within the complete data set of Pb–Pb collisions recorded so far in ALICE during Run 2. These estimates will be compared with estimates of the expected background to determine whether or not a search within ALICE is feasible. These estimates are made for a sexaquark produced in a Pb–Pb collision and annihilating in the analyzed interaction channel $\bar{S} + n \rightarrow \bar{\Lambda} + K_s^0$.

4.1 Sexaquark Production Estimation

Estimating the feasibility of a sexaquark search first requires knowledge of the quantity at which the sexaquarks are produced within a data set, which requires knowledge of the number of S produced per Pb-Pb event $M_{S,\text{prod}}$. The deuteron is used as a basis for estimation. Deuterons (d) are dibaryons consisting of a proton and a neutron with a combined six-quark state *uududd* and a mass of 1875.6 MeV. The similarity of the proposed mass, as well as the six-quark deuterons and sexaquarks, make them an obvious choice for estimating the sexaquark production rate. Measurements of the deuteron $\mathrm{d}N_{\text{prod}}/\mathrm{d}y$ have already been made for Pb–Pb collisions at a center-of-mass energy per nucleon pair of $\sqrt{s_{NN}} = 2.76\,\text{TeV}$ for various centralities [58]. The centrality range is from 0 to 80 % and to get an estimate of the production rate for collisions of any possible centrality, the different results in [58] are averaged to get a value of $\mathrm{d}N_{d,\text{prod}}/\mathrm{d}y = 3.98 \times 10^{-2}$. This value has to be corrected with the number densities n of deuterons and sexaquarks to get the value

$\mathrm{d}N_{\mathrm{prod}}/\mathrm{d}y$ of the sexaquarks:

$$\mathrm{d}N_{S,\mathrm{prod}}/\mathrm{d}y = \mathrm{d}N_{d,\mathrm{prod}}/\mathrm{d}y \cdot \frac{n_S}{n_d} \tag{4.1}$$

The number density of a particle in a non-interacting hadron gas is given by [59]:

$$n = \frac{g}{2\pi^2} T \cdot \sum_{k=1}^{\infty} \frac{m^2}{k} s^{k+1} e^{k\mu/T} K_2(km/T), \tag{4.2}$$

where g is the spin degeneracy, given by $g = 2 \cdot spin + 1$, with $spin = 1$ for the deuteron and $spin = 0$ for the sexaquark, T is the temperature of the hadron gas, for which the chemical freeze-out temperature of the QGP ($T = 156\,\mathrm{MeV}$) is used. The mass of the particle is denoted by m, μ is the baryon chemical potential, which is set to 0, and K_2 is the modified Bessel function of the second kind. The sum in the Eq. (4.2) is alternating for fermions and constant for bosons, which is handled by setting the sign s to -1 for fermions and 1 for bosons. Before estimating the value of $\mathrm{d}N_S/\mathrm{d}y$ from the Eq. (4.1), a slight adjustment must be made to $\mathrm{d}N_d/\mathrm{d}y$: Since the measured value is based on deuterons produced in Pb–Pb collisions at 2.76 TeV, but our simulations use a center-of-mass energy of 5.02 TeV, this energy difference has to be taken into account. The number of particles produced in a collision is proportional to $s^{0.155}$ [60], where s is the squared center-of-mass energy of the collision. Therefore, the Eq. (4.1) is rewritten as:

$$(\mathrm{d}N_{S,\mathrm{prod}}/\mathrm{d}y)_{5.02\,\mathrm{TeV}} = (\mathrm{d}N_{d,\mathrm{prod}}/\mathrm{d}y)_{2.76\,\mathrm{TeV}} \cdot \frac{(5.02\,\mathrm{TeV})^{2\cdot 0.155}}{(2.76\,\mathrm{TeV})^{2\cdot 0.155}} \cdot \frac{n_S}{n_d}. \tag{4.3}$$

Now Eq. (4.3) can be used together with Eq. (4.2) to estimate the expected yield of sexaquarks, which is plotted in Figure 4.1 as a function of the expected mass of the sexaquark.

By assuming the mass of the anti-sexaquark to be $m_S = 1.8\,\mathrm{GeV}/c^2$, which is the $\bar{S}$ mass used in the simulations, one arrives at a expected yield of $\mathrm{d}N_{S,\mathrm{prod}}/\mathrm{d}y = 0.0245$ for the S. If one assumes a constant yield over the whole rapidity range, the number of S produced per event can be calculated with:

$$M_{S,\mathrm{prod}} = \mathrm{d}N_{S,\mathrm{prod}}/\mathrm{d}y \cdot \Delta y. \tag{4.4}$$

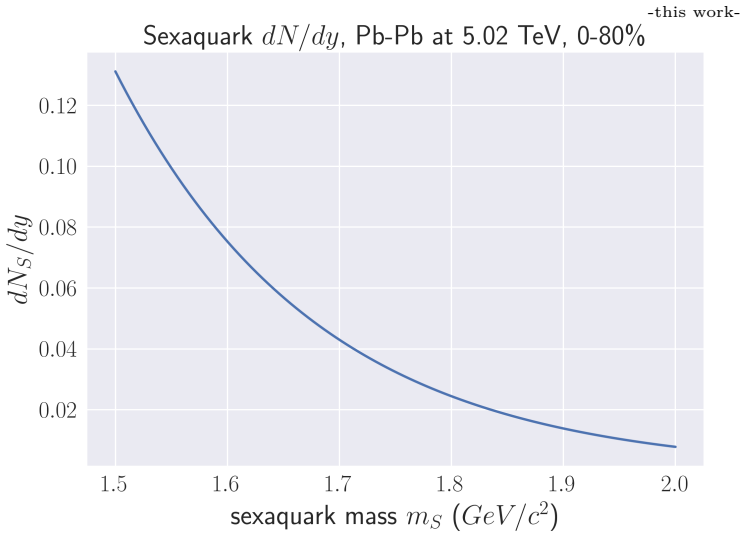


Figure 4.1: Estimated yield for the sexaquark as a function of $\bar{S}$ mass for Pb-Pb collisions at 5.02 TeV with a $0-80\%$ centrality. The estimation is based on the corresponding yield of deuterons, which is taken from [58] and averaged over all centralities. The estimated yield is calculated using Eq. (4.3) in combination with Eq. (4.2).

The TPC has an azimuthal coverage in pseudorapidity of $-0.8 < \eta < 0.8$, which results in $\Delta\eta = 1.6$. The translation of pseudorapidity into rapidity can be performed via [61]

$$y = \frac{1}{2} \log \frac{E + p_z}{E - p_z}, \tag{4.5}$$

where $E = \sqrt{p_{\mathrm{T}}^2 \cosh^2 \eta + m^2}$ and $p_z = p_{\mathrm{T}} \sinh \eta$. From this follows

$$y(\eta) = \frac{1}{2} \log \left(\frac{\sqrt{p_{\mathrm{T}}^2 \cosh^2 \eta + m^2} + p_{\mathrm{T}} \sinh \eta}{\sqrt{p_{\mathrm{T}}^2 \cosh^2 \eta + m^2} - p_{\mathrm{T}} \sinh \eta} \right), \tag{4.6}$$

which results in a rapidity range of $\Delta y = 1.206$, when assuming a value of $p_{\mathrm{T}} = 1.5\,\mathrm{GeV}/c$ for the transverse momentum and $m_S = 1.8\,\mathrm{GeV}$ for the S mass.

With the corresponding rapidity range, the expected number of produced sexaquark per event is given via Eq. (4.4) by $M_{S,\mathrm{prod}} = 0.0245 \cdot 1.206 = 0.0295$.

4.2 Sexaquark Interaction Rate Estimation

In order to be a possible dark matter candidate, the sexaquark has had to evade previous searches, which is largely explained by its proposed low interaction probability, which therefore greatly limits the number of detectable $\bar{S}$. For a quantitative estimate of the nucleon interaction probability, one first needs a value for the nucleon interaction cross section. In her paper, Farrar gives an estimate for the $S-$nucleon scattering cross section as $\sigma_{SN} \leq (\frac{1}{4} - 1)\sigma_{NN}^{el} \approx 5 - 20\,\text{mb}$ for $v/c \approx 1$ due to geometric considerations [3]. For further calculations, the lower limit of Farrar's first estimate of the scattering cross section is used with $\sigma_{SN} = 5\,\text{mb}$, which is in the order of normal hadrons. This value comes with large uncertainties and has a huge impact on the final estimated number of sexaquarks we expect to find. As Farrar notes in [62], where the possibility of $\bar{S}$ as a dark matter candidate is elaborated, and as such its interaction cross section must be compatible with experimental observations, leading to an upper limit estimate of $\sigma_{SN} < 10^{-29}\,\text{cm}$, which leads to $\sigma_{SN} < 0.01\,\text{mb}$. Such small annihilation cross sections undoubtedly lead to an unmeasurable low number of expected sexaquarks, and therefore this work must be considered a search for a new six-quark state $\bar{S}$, insead of a search for a dark matter particle $\bar{S}$. To calculate the interaction probability of a $\bar{S}$ with a neutron $p_{S,n}$ from the nucleon scattering cross section σ_{SN}, it is necessary to know the effective target thickness t_n, where $p_{S,n}$ is given by:

$$p_{S,n} = t_n \cdot \sigma_{SN}. \tag{4.7}$$

The target thickness can be calculated using

$$t_n = f \cdot X_0 \cdot N_A/M \cdot N \cdot 10^{-27}, \tag{4.8}$$

where f is the fraction of the radiation length that the particle must travel through to reach the TPC, X_0 is the radiation length of the detector, N_A is the Avogadro constant, M is the molar mass of the detector, N is the neutron number, and the factor 10^{-27} is to convert the target thickness from $[1/\text{cm}^2]$ to $1/\text{mb}$. From the material budget plot in Figure 4.2, one can estimate the fraction of radiation length at different stages of the detector. For our reconstruction, we need S that annihilate roughly in the center of the TPC at a radius of about 180 cm, since S that annihilate further away will be too far away to be detected, resulting in a fraction of the radiation length of about 0.125. The average molar mass M and neutron

number N can also be obtained from [63], where the average number of neutrons up to the center of the TPC is given by $\langle N \rangle = \langle A \rangle - \langle Z \rangle = 17.4 - 8.5 = 8.9$, where $\langle A \rangle$ is the mass number and $\langle Z \rangle$ is the average atomic number. Since the average atomic number $\langle A \rangle$ is also the molar mass (1 [amu] = 1[g/mol]), the molar mass M is given as $M = 17.4\,\mathrm{g/mol}$. Finally, the value of X_0 of the detector must be estimated. The main components of the detector are silicon and aluminum, for which the radiation length is given as $X_{0,\mathrm{Si}} = 21.82\,\mathrm{g/cm^2}$ and $X_{0,\mathrm{Al}} = 24.01\,\mathrm{g/cm^2}$, so an estimate slightly lower than these two values is used with $X_{0,\mathrm{Al}} = 20\,\mathrm{g/cm^2}$ to account for lighter parts in the detector setup. With these values, a final target thickness of $t_n = 7.4 \times 10^{-4}\,1/\mathrm{mb}$, giving a sexaquark-neutron interaction probability of $p_{S,n} = 0.00372$.

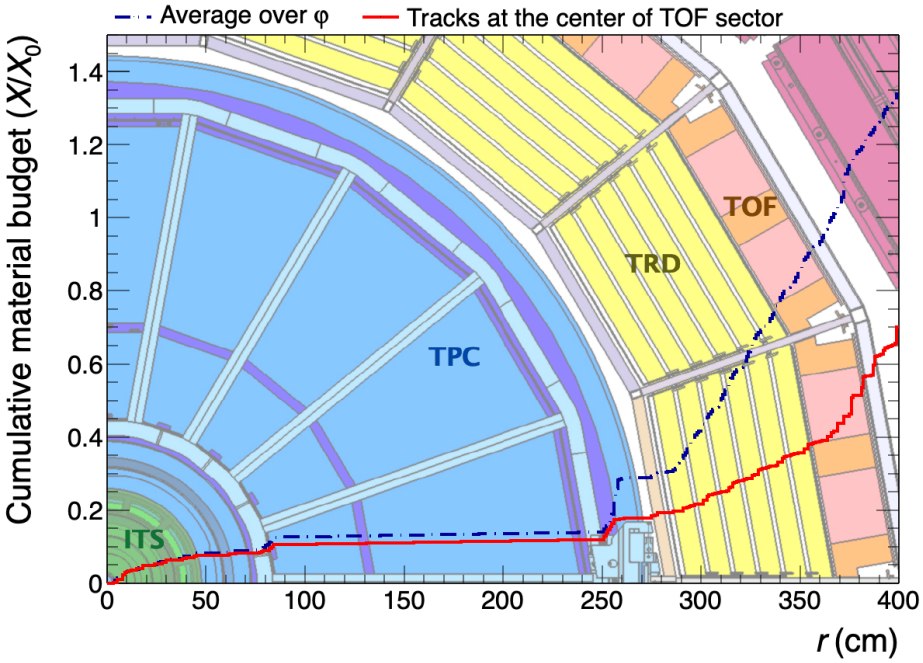


Figure 4.2: Material budget plot of ALICE from the beampipe up to the TOF detector. The plot depicts the cumulative distribution of material as a function of the radial distance from the beam pipe at the center of the TOF sectors as a red line and averaged over the azimuth angle as blue dotted line. The material budget is given in units of relative radiation length. Picture taken from Ref. [63].

4.3 Total Number of Events

Pb–Pb collisions produced in ALICE at a center-of-mass energy of 5.02 TeV since 2015 accumulate to a total number of 3.15×10^8 recorded Pb–Pb events[64]. It should be noted, however, that this number of events is not realistically usable in a proper analysis, since these events are not necessarily all perfectly reconstructed. By extending the criterion for the centrality of the S production estimate to 80 %, no events need to be discarded based on the centrality of the event. In addition, there are many reasons why a data sample may be unusable, such as errors in the ITS or reconstruction in the TPC. This analysis task does not rely on detector systems beyond the TPC, which makes it more robust against failures in various parts of ALICE, but realistically the number of usable events could be reduced by a large amount, but since this is very hard to estimate and the goal is rather to give an upper bound on the total number of reconstructable $\bar{S}$, the total number of events will be used for further studies. From the number of recorded events $N_{\text{Event}} = 3.15 \times 10^8$, one can estimate the total number of sexaquarks produced since 2015, which is $N_{S,\text{prod}} = N_{\text{Event}} \cdot M_{S,\text{prod}} = 9.29 \times 10^6$ of which $N_{S,\text{annih}} = N_{S,\text{prod}} \cdot p_{S,n} = 3.4 \times 10^4$ interact with the detector material.

4.4 Signal Efficiency & Background Estimation

The signal efficiency for the analyzed annihilation channel is estimated based on the Monte Carlo simulations. The different cuts and sources affecting the reconstruction efficiency are discussed. The simulation places annihilation products ($\bar{\Lambda}$ & K_s^0) of the sexaquark in the generated events and the decays of these particles are handled by the ALICE reconstruction framework, which also includes GEANT4. Each of the daughter particles has a decay channel in which only charged particles are produced and one in which only neutral particles are produced. The neutral decay channels are not attempted to be reconstructed, since for the $\bar{\Lambda}$ one product is a π^0 and for the K_s^0 both products are π^0, which has a very low reconstruction efficiency. Therefore, before even attempting the reconstruction, the efficiency is down to 44.2 %, since the branching ratio $\bar{\Lambda} \to \pi^+ + \bar{p}$ is 63.9 % and the branching ratio $K_s^0 \to \pi^+ + \pi^-$ is 69.2 %. Afterwards, all final state particles had to be reconstructed in the detector, for which the efficiency is easily available through the signal-only simulations 3.2,

where the efficiency of the TPC to reconstruct all particles was 18.3 %, which already includes the 44.2 % from the decay channels of the daughters. The detector efficiency for only the charged particles is given by $\epsilon_{\text{detector}} = 0.183/0.442 = 0.414 = 41.4\,\%$. Next, the sexaquark candidates are formed using the Custom V^0 Finder, which has an efficiency of 25.3 %, which is calculated by dividing the number of reconstructed Λ and K_S^0 signal V^0s by the total number of Λ and K_S^0 signal V^0s, for which both daughter particles were reconstructed in the TPC. The cuts applied to the sexaquark candidates and their efficiency are already discussed in Chapter 3.4.3, which will be listed in detail later, but overall the signal efficiency for the candidate cuts is $\epsilon_{\text{detector}} = 95.2\,\%$. Finally, the efficiency of the XGBoost classifier needs to be evaluated, which is simultaneously tied to the background rejection rate by setting the classifier's working point. This is explained in Section 3.4.6, where the threshold is set such that $\epsilon_{\text{XGBoost}} = 0.9$, which results in a number of surviving true background candidates of 0.33 ± 0.47 in the entire test set. With this number, the expected true background can be easily calculated, since the entire simulation consists of 171,000 events, 30 % of which is used for the test set, in which 0.33 ± 0.47 background candidates are found, resulting in $0.33 \pm 0.47 \frac{1}{51{,}300\text{events}} = 1 \pm 1.41 \frac{1}{153{,}900\text{events}}$.

The reconstruction efficiency can now be calculated just as easily by multiplying all efficiencies together, or by simply counting how many signal candidates survive all cuts. In the test sample, 1231 signal candidates survive after all cuts out of 51,300 events, resulting in a reconstruction efficiency of $\epsilon_{\bar{S}+n\to\bar{\Lambda}K_S^0} = 1231/51{,}300 = 0.0240$. There are two sources of efficiency reduction not yet accounted for in the efficiency: the branching ratio of $\bar{S}+n$, which leads to $\bar{S}+n \to \bar{\Lambda}K_S^0$, and the fact that a produced K^0 has a 50 % chance of being either a K_S^0 or a K_L^0, while the lifetime of the K_L^0 is too large to decay in the detector and thus be detectable. Estimating the branching ratio is not easy and would require lattice QCD calculations. For an approximate estimate, the different possible interaction channels from Table 1.1 are considered. For the $\bar{S}+n$ case, there are seven possible interaction channels, so a branching ratio of 1/10 is estimated, which is slightly lower than evenly distributed to account for the uncertainty in the estimate. With these two values, the overall efficiency becomes $\epsilon_{\bar{S}+n} = \epsilon_{\bar{S}+n\to\bar{\Lambda}K_S^0} \cdot BR \cdot p_{K_S^0/L} = 0.0012$. A comprehensive table of all reconstruction efficiencies can be found in Table 4.1.

Efficiency Step	**Efficiency Value**
Efficiency after decay channel branching ratios	44.2 %
Detector efficiency for reconstructing all charged particles	41.4 %
Custom V^0 Finder efficiency	25.3 %
Efficiency of cuts on sexaquark candidates	95.2 %
Efficiency of XGBoost classifier	90 %
Overall Reconstruction Efficiency in channel $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0$	2.40 %
Branching ratio BR	10 %
K^0 neutral particle decay probability ($p_{K_S^0/L}$)	50 %
Overall Efficiency ($\epsilon_{\bar{S}+n}$)	0.12 %

Table 4.1: Reconstruction efficiencies of the $\bar{S}$ reconstruction chain.

Expected Signal and Background Count The previous calculations can be combined to calculate the estimated number of anti-sexaquarks that one expects to measure in this one particular interaction channel from all available data. This can be calculated by

$$N_{\bar{S},\text{detect}} = N_{S,\text{prod}} \cdot p_{S,n} \cdot \epsilon_{\bar{S}+n}, \tag{4.9}$$

which gives $N_{\bar{S},\text{detect}} = 42$ expected detected antisexaquarks in all recorded data between 2015 and 2018. The expected number of background candidates can be calculated by multiplying the number of surviving background candidates per event by the total number of events:

$$N_{\text{bkg.}} = N_{\text{event}} \cdot n_{\text{bkg.}}, \tag{4.10}$$

which results in $N_{\text{bkg.}} = 2049 \pm 2890$. From the signal and background estimates, the expected significance can be calculated using Eq. (3.27) to be $S = 0.92\sigma$.Due to the large uncertainties of the background originating from the limited amount of simulated data, the significance is calculated again for a 2σ interval around the background to give a more realistic representation of the expected range of the significance. For the lower bound, the background vanishes, resulting in a significance of $S = \sqrt{N_{\bar{S},\text{detect}}} = 6.5\sigma$ and for the upper bound, 7827 background candidates are recorded, resulting in a significance of $S = 0.47\sigma$.

5 Discussion and Outlook

In this thesis, machine learning methods were applied to simulations of the anti-sexaquark to determine its detection efficiency and background suppression. The simulations, provided by Andrés Bórquez, consist of a total of 171,000 events in which an anti-sexaquark annihilates with the detector material of the ALICE detector in the evaluated interaction channel $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0$. Annihilation occurs in front of or inside the TPC and the resulting granddaughter particle tracks are recorded with the TPC detector. Two types of reconstruction have been evaluated in this work. The resulting particles are either paired together using a Custom V^0 Finder developed for this search by Andrés Bórquez, or using the true Monte Carlo information about the simulation. The V^0 pairs are then formed to find the secondary vertex of the $\bar{S}$ annihilation in a V^0 pair finding algorithm developed during this thesis. Information from the secondary vertex and the V^0s are then combined into anti-sexaquark candidates. Background reducing cuts on the candidates are identified, combining DCA, radial distance and invariant mass cuts. A BDT classifier using the XGBoost library is then used to further reduce the resulting background.

The analysis on true V^0s showed that a near-ideal classification can be achieved with perfect detector measurements. The analysis on true V^0 candidates also allowed to find good classification features due to the absence of possible artifacts induced by the detector simulation. In the case of custom V^0 candidates, hyperparameter optimization was used to maximize the Punzi criterion achieved by the classifier, and a signal and background analysis was performed. The optimal signal efficiency of the classifier was determined to be $\epsilon_{\text{XGBoost classifier}} = 0.9$, resulting in an average of 0.33 ± 0.47 true background counts within a 51,300 event dataset. The obtained signal efficiencies were combined with all other efficiencies measured in this thesis to obtain a combined annihilation channel reconstruction efficiency $\bar{S} + n \rightarrow \bar{\Lambda} K_S^0$ of $\epsilon_{\bar{S}+n\rightarrow\bar{\Lambda}K_S^0} = 0.0240 = 2.4\,\%$ and a complete reconstruction efficiency of $\bar{S}$ in the case of annihilation with one neutron of $\epsilon_{\bar{S}+n} = 0.0012$. The efficiencies were combined with calculations of the production and interaction probability of the $\bar{S}$

to calculate an estimate of the number of anti-sexaquarks expected to be detectable in an analysis including all measured data from 2015 and 2018, which resulted in 42 expected anti-sexaquarks for this particular channel. The estimated background in the complete data set is estimated to be 2049 ± 2890, which could lead to a sexaquark detection with a significance range between 0.47σ and 6.5σ, with a mean of 0.92σ, if the analysis is performed on the real data. Comparing these results with the previous search at CMS by de Clercq [6], where a reconstruction efficiency of $\epsilon_{\bar{S}+n\to\bar{\Lambda}K^0_S} = 0.0014\,\%$, it is clear that a search for $\bar{S}$ at ALICE is much more realistic, since the higher CMS luminosity does not outweigh the lower reconstruction efficiency in the range of three orders of magnitude. However, the calculations of expected sexaquarks and significance should be treated with caution, since the uncertainty on the theoretical values of σ_{SN} could be enormous and the actual value of σ_{SN} could also vary by several orders of magnitude. The efficiency estimates are based on the simulation of the $\bar{S}$ interaction, which may not fully capture the complexity of the real-world interaction. Many assumptions about unknown variables, such as the mass of the $\bar{S}$, had to be made in order to simulate the sexaquark, while others had to be simplified for the sake of time, which inevitably led to uncertainties in the reconstruction efficiencies. These simplified assumptions include, among others, the Fermi motion of the struck neutron, which was assumed to be at rest, and the uniform distribution of the interaction vertex of the $\bar{S} + n$ interaction, which does not take into account the material budget of ALICE. Overall, the search for the sexaquark using BDTs has potential when considering that only one out of seven possible interaction channels has been investigated, and in particular for the channel $\bar{S} + p \to \bar{\Lambda}K^+$ a similar or even higher reconstruction efficiency is expected (see Section 3.2.3).

Although a search on real Run 2 data seems promising and is already feasible, some improvements can and should be made before. First, the Fermi motion on the struck neutron should be included in the next simulation, as well as an annihilation range distribution scaled by the material budget of the detector. More simulations in general would also help the reconstruction, as it would lead to more training data, which will most likely improve the classification efficiency, as well as more test data, which will help by reducing the uncertainty in the background estimates, which is currently larger than the estimate itself. In addition, general-purpose simulations already performed by the ALICE collaboration, which do not include the sexaquark, could be used as background training and test data. Finally, a selection-cut optimization could be performed, where each cut applied to the reconstructed data

could be examined for reduction to obtain more signal and background candidates, shifting the task of background elimination more toward the classifier. In the case of the Custom V^0 Finder for example with its current efficiency of 25 %, a possible increase of the total efficiency by a factor two or more could reasonably be achieved. Finally, the reconstruction of the other interaction channels could be investigated, for which only two steps need to be taken for each additional channel: run new simulations and replace the V^0 pair finding algorithm with an algorithm that reconstructs the sexaquark candidates for that specific channel (e.g. a V^0 - particle track intersection finder for $\bar{S} + p \rightarrow \bar{\Lambda} K^+$). A search for all seven possible interaction channels could increase the expected significance by a factor of $\sqrt{7} = 2.65$. In addition, in LHC Run 3, ALICE will switch to a continuous readout, resulting in much higher luminosity and more data. In Run 3 and 4 an increase of available statistics by a factor of 50 is expected, which could increase the expected significance by a factor of $\sqrt{50} = 7.1$, assuming linear scaling of signal and background. This will allow for a more promising and extensive search and, with current efficiencies, should eventually allow for a discovery of the sexaquark or to confidently disprove the existence of the sexaquark.

A Appendix

A.1 Armenteros-Podolanski Plot

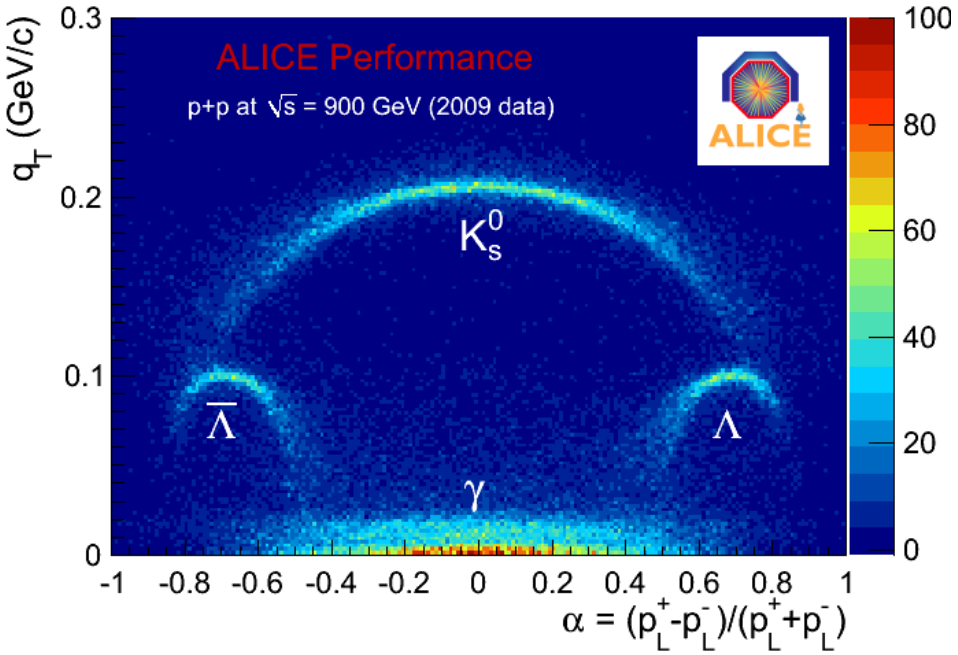


Figure A.1: Armenteros-Podolanski plot from the ALICE experiment using data from pp collisions at $\sqrt{s} = 900\,\text{GeV}$. The different V0 particles can be identified using the kinematics of their decay products. $p_{\text{L}}^{+/-}$ are the longitudinal momenta of the positively and negatively charged decay products with respect to the momentum vector of the V0 and q_{T} represents the transverse momentum of the positive decay product with respect to the momentum vector of the V0. Picture taken from Ref. [65].

A.2 Simulation Run Numbers

245683	245692	245700	245702	245705	245829	245831	245833	245923
245949	245952	245954	245963	246001	246003	246012	246036	246037
246042	246048	246049	246052	246053	246087	246089	246113	246115
246148	246151	246152	246153	246178	246180	246181	246182	246185
246217	246222	246225	246271	246272	246275	246276	246424	246428
246431	246434	246487	246488	246493	246495	246750	246751	246757
246758	246759	246760	246763	246765	246766	246804	246805	246807
246808	246809	246810	246844	246845	246846	246847	246851	246928
246945	246948	246980	246982	246984	246989	246991	246994	296690
296691	296693	296694	296752	296781	296784	296785	296786	296787
296790	296793	296794	296799	296835	296836	296838	296839	296848
296850	296851	296852	296894	296899	296900	296903	296930	296931
296932	296934	296935	296938	296941	296966	297031	297035	297085
297117	297118	297119	297123	297124	297128	297129	297132	297133
297193	297195	297196	297218	297221	297222	297278	297310	297311
297312	297315	297317	297332	297333	297335	297336	297363	297366
297367	297372	297379	297380	297405	297406	297413	297414	297415
297441	297442	297446	297450	297451	297452	297479	297483	297512
297537	297540	297541	297542	297544	297558	297588	297590	297595

Table A.1: Table with the run numbers, the $\bar{S}$ Pb–Pb simulations are anchored to. Numbers starting with 24 correspond to 2015 recorded data and numbers with 29 correspond to 2018 recorded data.

Literature

[1] S. McGaugh. "Predictions and Outcomes for the Dynamics of Rotating Galaxies." *Galaxies* 8.2 (2020). DOI: 10.3390/galaxies8020035.

[2] L. Roszkowski et al. "WIMP dark matter candidates and searches—current status and future prospects." *Reports on Progress in Physics* 81.6 (2018), p. 066201. DOI: 10.1088/1361-6633/aab913.

[3] G. R. Farrar. "Stable Sexaquark." *arXiv e-prints*, arXiv:1708.08951 (2017), arXiv:1708.08951. DOI: 10.48550/arXiv.1708.08951.

[4] R. Godang. "Search for Dark Matter in Upsilon Decays at BABAR Experiment." *Journal of Physics: Conference Series* 1468.1 (2020), p. 012045. DOI: 10.1088/1742-6596/1468/1/012045.

[5] F. Partous. "The Standard Model Strikes Back: Searching For Sexaquark Dark Matter At The LHC." MA thesis. Inter-University Institute For High Energies, 2018. URL: https://iihe.ac.be/sites/default/files/thesis-florian-partous-cms-master-2018pdf/thesis-florian-partous-cms-master-2018.pdf.

[6] J. T. De Clercq. "The Upgraded Outer Tracker for the CMS Detector at the High Luminosity LHC, and Search for Composite Standard Model Dark Matter with CMS at the LHC." 2019. URL: https://cds.cern.ch/record/2708026.

[7] F. L. Schlichtmann. "Reconstruction study of the S particle dark matter candidate at ALICE." MA thesis. Heidelberg Universiy, 2021. URL: https://www.physi.uni-heidelberg.de/Publications/MasterThesis_print.pdf.

[8] M. Gell-Mann. "A Schematic Model of Baryons and Mesons." *Phys. Lett.* 8 (1964), pp. 214–215. DOI: 10.1016/S0031-9163(64)92001-3.

[9] G. Zweig. *An SU_3 model for strong interaction symmetry and its breaking; Version 1.* Tech. rep. Geneva: CERN, 1964. DOI: 10.17181/CERN-TH-401.

[10] S.-K. Choi et al. "Observation of a Narrow Charmoniumlike State in Exclusive $B^{\pm} \to K^{\pm}\pi^{+}\pi^{-}J/\psi$ Decays." *Phys. Rev. Lett.* 91 (26 Dec. 2003), p. 262001. DOI: 10.1103/PhysRevLett.91.262001.

[11] G. Cotugno et al. "Charmed Baryonium." *Phys. Rev. Lett.* 104 (2010), p. 132005. DOI: 10.1103/PhysRevLett.104.132005.

[12] Z. Q. Liu et al. "Study of $e^{+}e^{-} \to \pi^{+}\pi^{-}J/\psi$ and Observation of a Charged Charmoniumlike State at Belle." *Phys. Rev. Lett.* 110 (25 June 2013), p. 252002. DOI: 10.1103/PhysRevLett.110.252002.

[13] M. Ablikim et al. "Observation of a Charged Charmoniumlike Structure in $e^{+}e^{-} \to \pi^{+}\pi^{-}J/\psi$ at $\sqrt{s}$=4.26 GeV." *Phys. Rev. Lett.* 110 (2013), p. 252001. DOI: 10.1103/PhysRevLett.110.252001.

[14] R. Aaij et al. "Observation of the Resonant Character of the $Z(4430)^{-}$ State." *Phys. Rev. Lett.* 112 (2014), p. 222002. DOI: 10.1103/PhysRevLett.112.222002.

[15] R. Aaij et al. "Observation of $J/\psi\phi$ Structures Consistent with Exotic States from Amplitude Analysis of $B^{+} \to J/\psi\phi K^{+}$ Decays." *Phys. Rev. Lett.* 118 (2 Jan. 2017), p. 022003. DOI: 10.1103/PhysRevLett.118.022003.

[16] R. Aaij et al. "Amplitude analysis of $B^{+} \to J/\psi\phi K^{+}$ decays." *Phys. Rev. D* 95 (), p. 012002. DOI: 10.1103/PhysRevD.95.012002.

[17] D. Dominguez. "Illustrations of the new pentaquark and tetraquarks discovered by LHCb" (2022). General Photo. URL: https://cds.cern.ch/record/2814136.

[18] C. Amsler et al. "Review of Particle Physics." *Physics Letters B* 667.1 (2008). Review of Particle Physics, pp. 1–6. DOI: 10.1016/j.physletb.2008.07.018.

[19] E. Oset & A Martinez Torres. "Critical view of the claimed Thetasup + pentaquark." *AIP Conference Proceedings* 1343.1 (2011). DOI: 10.1063/1.3574993.

[20] R. Aaij et al. "Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $^{0}_{b} \to J/\psi K^{-}p$ Decays." *Phys. Rev. Lett.* 115 (7 2015), p. 072001. DOI: 10.1103/PhysRevLett.115.072001.

[21] 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.* 122 (22 2019), p. 222001. DOI: 10.1103/PhysRevLett.122.222001.

[22] R. L. Jaffe. "Perhaps a Stable Dihyperon." *Phys. Rev. Lett.* 38 (5 1977), pp. 195–198. DOI: 10.1103/PhysRevLett.38.195.

[23] A. Chodos et al. "New extended model of hadrons." *Phys. Rev. D* 9 (12 June 1974), pp. 3471–3495. DOI: 10.1103/PhysRevD.9.3471.

[24] J. R. Green et al. "Weakly Bound H Dibaryon from SU(3)-Flavor-Symmetric QCD." *Phys. Rev. Lett.* 127 (24 2021), p. 242003. DOI: 10.1103/PhysRevLett.127.242003.

[25] R. E. Chrien. "H particle searches at Brookhaven." *Nucl. Phys. A* 629 (1998). Ed. by H. Ejiri et al., pp. 388C–397C. DOI: 10.1016/S0375-9474(97)00714-8.

[26] B. H. Kim et al. "Search for an H-Dibaryon with a Mass near $2m_\Lambda$ in $\Upsilon(1S)$ and $\Upsilon(2S)$ Decays." *Phys. Rev. Lett.* 110 (22 2013), p. 222002. DOI: 10.1103/PhysRevLett.110.222002.

[27] J. Adam et al. "Search for weakly decaying Λn and ΛΛ exotic bound states in central Pb–Pb collisions at sNN=2.76 TeV." *Physics Letters B* 752 (2016), pp. 267–277. DOI: https://doi.org/10.1016/j.physletb.2015.11.048.

[28] S. R. Beane et al. "Light nuclei and hypernuclei from quantum chromodynamics in the limit of SU(3) flavor symmetry." *Phys. Rev. D* 87 (2013), p. 034506. DOI: 10.1103/PhysRevD.87.034506.

[29] G. R. Farrar. "A Stable Sexaquark: Overview and Discovery Strategies" (2022).

[30] J. P. Lees et al. "Search for a Stable Six-Quark State at BABAR." *Physical Review Letters* 122.7 (2019). DOI: 10.1103/physrevlett.122.072002.

[31] P. A. R. Ade et al. "Planck 2015 results." *Astronomy & Astrophysics* 594 (2016), A13. DOI: 10.1051/0004-6361/201525830.

[32] G. R. Farrar. "A precision test of the nature of Dark Matter and a probe of the QCD phase transition." *arXiv: High Energy Physics - Phenomenology* (2022). DOI: 10.48550/arXiv.1805.03723.

[33] G. R. Farrar & G. Zaharijas. "Nuclear and nucleon transitions of the H dibaryon." *Physical Review D* 70.1 (2004). DOI: 10.1103/physrevd.70.014008.

[34] S. D. McDermott et al. "Deeply bound dibaryon is incompatible with neutron stars and supernovae." *Physical Review D* 99.3 (2019). DOI: 10.1103/physrevd.99.035013.

[35] M. Shahrbaf et al. "Sexaquark dilemma in neutron stars and its solution by quark deconfinement." *Physical Review D* 105.10 (2022). DOI: 10.1103/physrevd.105.103005.

[36] E. Mobs. "The CERN accelerator complex - 2019. Complexe des accélérateurs du CERN - 2019" (July 2019). General Photo. URL: https://cds.cern.ch/record/2684277.

[37] The ALICE Collaboration. "Performance of the ALICE experiment at the CERN LHC." *International Journal of Modern Physics A* 29.24 (2014), p. 1430044. DOI: 10.1142/s0217751x14300440.

[38] B. Alessandro et al., ALICE Collaboration. *ALICE: Physics Performance Report. ALICE physics performance : Technical Design Report*. Vol. 32. Technical Design Report ALICE. Geneva: CERN, 2005. DOI: 10.1088/0954-3899/32/10/001.

[39] K. Aamodt et al., ALICE Collaboration. "The ALICE experiment at the CERN LHC." *Journal of Instrumentation* 3.08 (2008), S08002–S08002. DOI: 10.1088/1748-0221/3/08/s08002.

[40] B. Abelev et al., ALICE Collaboration. *Technical Design Report for the Upgrade of the ALICE Inner Tracking System*. Tech. rep. CERN-LHCC-2013-024. ALICE-TDR-017. 2013. DOI: 10.1088/0954-3899/41/8/087002.

[41] *Upgrade of the ALICE Time Projection Chamber*. Tech. rep. 2013. URL: https://cds.cern.ch/record/1622286.

[42] V. Peskov et al. "Technical Design Report for the Upgrade of the ALICE Time Projection Chamber" (2014). DOI: 10.13140/RG.2.1.1761.7681.

[43] S. Acharya et al., ALICE Collaboration. "The ALICE Transition Radiation Detector: construction, operation, and performance." *Nucl. Instrum. Meth. A* 881 (2018), pp. 88–127. DOI: 10.1016/j.nima.2017.09.028.

[44] *ALICE Time-Of-Flight system (TOF): Technical Design Report*. Technical design report. ALICE. Geneva: CERN, 2000. URL: https://cds.cern.ch/record/430132.

[45] H. Allamy & R. Z. Khan. "Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study)" (2014), pp. 163–172. DOI: 10.3850/978-981-09-5247-1_017.

[46] Y. Coadou. "Boosted Decision Trees." *Artificial Intelligence for High Energy Physics*. WORLD SCIENTIFIC, 2022, pp. 9–58. DOI: 10.1142/9789811234033_0002.

[47] L. Grinsztajn et al. *Why do tree-based models still outperform deep learning on tabular data?* 2022. DOI: 10.48550/arXiv.2207.08815.

[48] Harris Drucker & Corinna Cortes. "Boosting Decision Trees." Vol. 8. Jan. 1995, pp. 479–485.

[49] T. Chen & C. Guestrin. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016. DOI: 10.1145/2939672.2939785.

[50] S. Agostinelli et al. "Geant4—a simulation toolkit." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8.

[51] R. Brun & F. Rademakers. "ROOT — An object oriented data analysis framework." *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 389.1 (1997). New Computing Techniques in Physics Research V, pp. 81–86. DOI: https://doi.org/10.1016/S0168-9002(97)00048-X.

[52] M. Gyulassy & X. Wang. "HIJING 1.0: A Monte Carlo program for parton and particle production in high energy hadronic and nuclear collisions." *Computer Physics Communications* 83.2-3 (1994), pp. 307–331. DOI: 10.1016/0010-4655(94)90057-4.

[53] W. Gellert & Van Nostrand Reinhold Company. *The VNR concise encyclopedia of mathematics*. English. 2nd ed. Van Nostrand Reinhold New York, 1989, 776 p., 56 p.

[54] T. Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework." *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019. DOI: 10.48550/arXiv.1907.10902.

[55] S. Watanabe. *Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance*. 2023. DOI: 10.48550/arXiv.2304.11127.

[56] G. Punzi. *Sensitivity of searches for new signals and its optimization*. 2003.

[57] L. S Shapley. "A Value for n-Person Games." *Contributions to the Theory of Games II*. Ed. by Harold W. Kuhn & Albert W. Tucker. Princeton: Princeton University Press, 1953, pp. 307–317.

[58] J. Adam et al. "Production of light nuclei and anti-nuclei in pp and Pb-Pb collisions at energies available at the CERN Large Hadron Collider." *Physical Review C* 93.2 (2016). DOI: 10.1103/physrevc.93.024917.

[59] P. Braun-Munzinger et al. "PARTICLE PRODUCTION IN HEAVY ION COLLISIONS." *Quark–Gluon Plasma 3*. WORLD SCIENTIFIC, 2004, pp. 491–599. DOI: 10.1142/9789812795533_0008.

[60] J. Adam et al. "Centrality Dependence of the Charged-Particle Multiplicity Density at Midrapidity in Pb-Pb Collisions at $\sqrt{s_{NN}} = 5.02$ TeV." *Phys. Rev. Lett.* 116 (2016), p. 222302. DOI: 10.1103/PhysRevLett.116.222302.

[61] J. Stachel & K. Reygers. *QGP Physics - from Fixed Target to LHC*. 2011. URL: https://www.physi.uni-heidelberg.de/~reygers/lectures/2011/qgp/qgp_02_kinematics.pdf.

[62] G. R. Farrar et al. "Dark Matter Particle in QCD" (2022). DOI: 10.48550/arXiv.2007.10378.

[63] S. Acharya et al. "Measurement of the low-energy antideuteron inelastic cross section." *Phys. Rev. Lett.* 125 (2020), p. 162001. DOI: 10.1103/PhysRevLett.125.162001.

[64] ALICE Collaboration. *ALICE upgrades during the LHC Long Shutdown 2*. 2023. DOI: 10.48550/arXiv.2302.01238.

[65] C. Lippmann. "Particle identification." *Nucl. Instrum. Methods Phys. Res., A* 666.arXiv:1101.3276 (2011). 61 pages, 30 figures, 148–172. 61 p. DOI: 10.1016/j.nima.2011.03.009.

Author's Contribution

The author was the main and sole developer behind the XGBoost classifier as well as the Optuna hyperparameter optimization. Furthermore, the author was the main developer of the V^0 pair finding algorithm. Analysis on the reconstruction efficiency of a pure simulated anti-sexaquark interaction of the detector was performed by the author. The author did **not** make the simulations used for this thesis or wrote the Custom V^0 Finder code, which was both done by Andés Bórquez. All complementary code used to handle the input data from the candidate trees, test and evaluate the performance of the classifier and plot the figures within this thesis marked with *this work* were written and deployed by the author. The author investigated and applied the background reducing cuts on anti- sexaquark candidates and performed the calculations on the expected amount of detectable sexaquarks, as well as the amount of expected surviving background candidates.

Acknowledgments

First of all, I would like to thank Prof. Dr. Klaus Reygers for giving me the opportunity to do my Master's thesis in the ALICE group and for all the help I received in writing my thesis as well as the constructive discussions and guidance throughout the thesis. He always had an open ear when I ran into problems or needed help writing my thesis, and always managed to take time out of his busy schedule to help with (often lengthy) discussions and suggestions on how to move forward.
I would also like to thank Prof. Dr. Norbert Herrmann for agreeing to be the second referee for this thesis.
I must express my gratitude to M.Sc. Andrés Bórquez, without whom this thesis would not be possible. He provided me with the simulations and is also the developer of the custom V0 finder that this thesis relies heavily on. He was also a kind and pleasant person to work with, and his cheerful attitude always lifted the spirits when things were not going as planned. He always provided me with the suggestions and information I needed and also proofread my thesis.
Thanks to Dr. Martin Kroesen for proofreading my thesis and helping me with machine learning questions.
Thanks go out to M.Sc. Felix Schlepper, who I shared an office with during most of the duration of my master thesis, who was also working on XGBoost classification, would help out whenever he could and was always fun company to have around.
I would like to thank the entire ALICE group for providing a very pleasant working environment and for their help whenever it was needed.
A big thank you goes out to my girlfriend, whose constant support and encouragement kept me going when times were tough and helped me get through this thesis.
Finally, I would like to thank my parents for their unwavering support and constant encouragement throughout my years of study and the process of researching and writing this thesis.

Selbstständigkeitserklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 8. Juli 2023

.....

Unterschrift