BERGISCHE UNIVERSITÄT WUPPERTAL

School of Mathematics and Natural Science
Experimental Particle Physics

Application of Novel Analysis Workflows in an ATLAS Search for Supersymmetry

Dissertation

Martin V. F. Errenst

Wuppertal, 10.03.2021

First examiner: Prof. Dr. Christian Zeitnitz
Second examiner: Dr. Markus Elsing

SPONSORED BY THE
Federal Ministry
of Education
and Research

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-20210811-094205-7
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20210811-094205-7]

DOI: 10.25926/6nfh-5242
[https://doi.org/10.25926/6nfh-5242]

Abstract

The size of the ATLAS collaboration results in a broad and diverse analysis software landscape. Previous and ongoing harmonization efforts aim to reduce the computational and storage footprint of analyses and provide common concepts to discuss and develop analysis software within the collaboration. In this dissertation, the recently introduced *CP* (*Combined Performance*) *algorithms* are studied in the context of an ongoing search for R-parity violating Supersymmetry in the final state of 1+ or 2 same-sign leptons and 4 to 15+ (b-)jets. A comparison between the reference implementation of the analysis and a new CP algorithm version allows for a detailed discussion of the configuration and computing performance.

Für Katharina

Acknowledgements

I want to thank my supervisors Christian Zeitnitz and Attila Krasznahorkay. You have introduced me to the ATLAS collaboration, which does not stop to amaze me. With your guidance, I was able to get an idea of the culture of the collaboration and make a small contribution to a monumental scientific, technological, and social project.

My dissertation would not have been possible without the Wolfgang Gentner scholarship, allowing me to work at CERN for three years. I am grateful for all the friends I have found and experiences I have gained during this time.

I also want to thank my parents, Christiane and Christoph. You have always encouraged a sense of independence and curiosity, which I see as essential traits in myself today.

Finally, I want to thank my wife, Katharina. Please know how deeply I appreciate your continuous love and support that started even before my very first lecture in physics over a decade ago.

Contents

Part I
Introduction

The *Large Hadron Collider* (LHC) [EB08] at CERN is a hadron collider facility, designed for proton-proton and lead ion collisions. It hosts multiple particle physics experiments in several caverns, roughly 100 m below the surface, outside of Geneva.

ATLAS [Col08] is a general-purpose detector at the LHC and is considered to be one of the leading experiments of particle physics. The first data taking campaign, called Run 1, occurred during 2010 − 2012 and proton-proton collisions were recorded at a centre of mass energy of $\sqrt{s} = 7$ TeV and $\sqrt{s} = 8$ TeV [Aad+13]; [Aab+16]. Run 1 finished with lead ion collisions in 2013. During Run 2 (2015 − 2018), the centre of mass energy was increased to $\sqrt{s} = 13$ TeV [19b]. This data is the basis for many published and ongoing physics studies. Run 3 is scheduled to start in 2022.

The ATLAS collaboration consists of thousands of active scientists from 181 institutions [Colc]. Data analyses within the collaboration are organized by research topic, resulting in communities focussing on selected particles (e.g. Higgs, top) or theories (e.g. Supersymmetry, Standard Model). Subgroups are set up to study specific models or processes in greater detail.

This multitude of analysis groups is partially mirrored in the analysis software landscape of ATLAS, where many *analysis frameworks* are developed and used in parallel. This is in agreement with Conway's law, stating that "any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure" [Con68].

After Run 1 of the LHC, an ongoing effort to centralize and harmonize ATLAS analysis software was started. A new format to store analysis object data, *xAOD*, was introduced to optimize the data handling [Buc+15]. The interfaces of *CP* (*Combined Performance*) *tools* were harmonized, which are defined and developed by the corresponding CP groups, who are responsible for a best-practice treatment of a given physics object (e.g. electrons and photons, muons, etc.) [FAR+15]; [Bir+14]. A derivation framework was introduced, centrally providing data formats containing selected events for multiple ATLAS analyses [Cat+15].

In preparation for Run 3, the harmonization of analysis software continues with the recent development of *CP algorithms*. CP algorithms are wrapping CP tools, resulting in schedulable entities that act on physics data. This step defines a central convention of the "steering" of analysis decisions and the corresponding configuration. Integrating CP algorithms in existing analysis frameworks is an ongoing process.

In this dissertation, the novel CP algorithms are used to reproduce the n-tuple production framework of a search for R-parity violating Supersymmetry with a final state of one lepton and multiple jets ($\geq$ 4), called the *RPV1L* analysis. This validates the current state of CP algorithms and can serve as a proof-of-concept of their integration with an underlying analysis framework.

The dissertation is organized as follows: A general introduction to particle physics is provided in Chapter 1. Chapter 2 introduces the ATLAS experiment at the LHC. The Chapters 3 and 4 describe the particle reconstruction, common data structures, and workflows used within the collaboration.

Part II contains a discussion of the RPV1L analysis. A theory basis for Supersymmetry is given in Chapter 5. The analysis is motivated and outlined in Chapter 6. Chapter 7 contains the background analysis of misidentified leptons from QCD multi-jet events. A summary of the analysis results is presented in Chapter 8.

Part III describes developments and studies made with the novel analysis infrastructure. Chapter 9 introduces CP algorithms in the general context of analysis frameworks in ATLAS. The reproduction of the RPV1L n-tuples with CP algorithms is discussed in Chapter 10. A study of the computing performance of CP algorithms is presented in Chapter 11.

Part IV summarizes and concludes this dissertation.

1 Theory

In this chapter, a qualitative description of the theory of particle physics is provided. Section 1.1 describes the Standard Model of particle physics. A brief description of proton-proton collisions is presented in Section 1.2. The interaction of particles with matter is described in Section 1.3. The theory of Supersymmetry is addressed in Chapter 5 of Part II.

1.1 Standard Model

The *Standard Model* (SM) of particle physics describes all known fundamental particles and their interactions. The discussion of the Standard Model follows Sections one and two of Griffiths' "*Introduction to elementary particles*" [Gri87, pp. 13-88]. Additional resources are cited when used.

Figure 1.1 shows a categorization of all particles of the SM. The particle masses are given in natural units, i.e. $\hbar = c = 1$. *Fermions*, which are either a *quark* or *lepton*, are the constituents of matter. *Bosons* act as force carriers, conveying the interactions between particles. Quarks and leptons are categorized into three generations, where each subsequent generation introduces particles with similar characteristics (e.g. charge, spin, etc.), but with a higher mass. Fermions of higher generations are unstable and decay after a characteristic lifetime.

There are three "up"-type quarks, *up*, *charm*, and *top*, with an electrical charge of $\frac{2}{3}e$. The "down"-type quarks, *down*, *strange*, and *bottom*, have an electrical charge of $-\frac{1}{3}e$. Quarks are the only type of fermions that are subject to the strong force, mediated by the *gluon*, which also couples to itself. With respect to the strong force, quarks can be in one of three states, commonly described through *colour charges* R, G, and B. In a very simplified description, gluons can change the colour of quarks by carrying a colour and a second anti-colour. Quarks are only observed in bound states with a combined neutral colour charge, i.e. as *mesons* ($q\bar{q}$ with colour and anti-colour) or *baryons* (qqq or $\bar{q}\bar{q}\bar{q}$ with R, G and B). There are also observations of exotic and unstable hadrons like tetraquarks ($qq\bar{q}\bar{q}$) [Cho+08] and pentaquarks ($qqqq\bar{q}$) [Aai+14]; [Aai+15].

A lepton generation consists of a charged particle and its corresponding *neutrino*. The charged leptons are called *electron*, *muon*, and *tau*. All quarks and charged leptons can interact through the electromagnetic force, mediated by the *photon*.

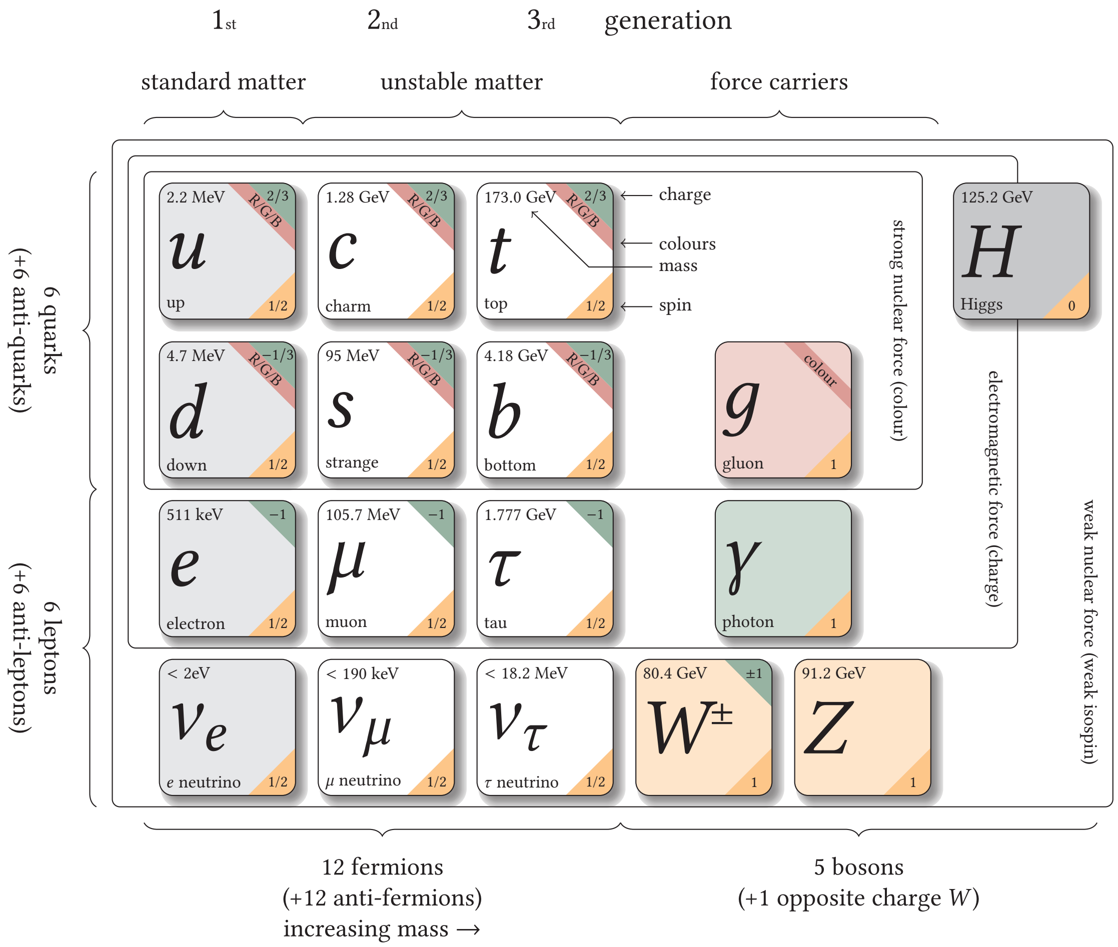


Figure 1.1: All fundamental particles of the Standard Model. Modified version of "*Standard model of physics*" [GB16], updated mass measurements according to "Review of Particle Physics" [Tan+18] in natural units.

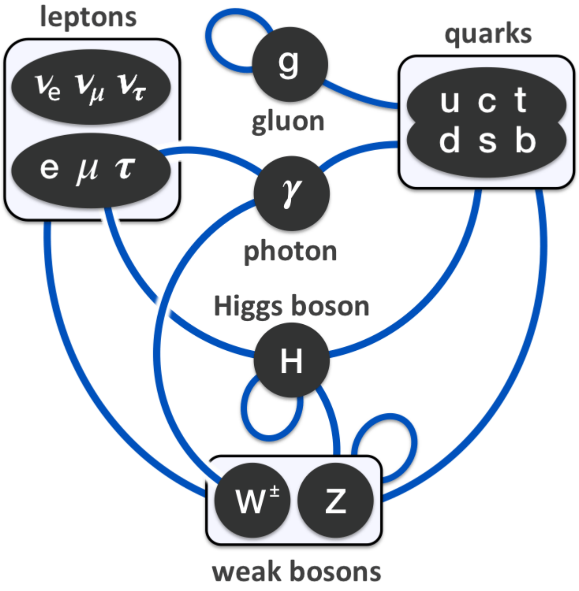


Figure 1.2: Particles of the Standard Model and their interactions [Dre14].

All fermions, including neutrinos, are subject to the weak force, which is mediated by the charged $W^\pm$ and neutral Z bosons. Therefore, neutrinos can only take part in weak interactions. Fermions can change their "flavour", i.e. lepton or quark type, only through charged weak interactions, involving a $W^\pm$.

Since the Z boson couples to all fermions, measurements of its lifetime can be used to prove that only three generations of quarks and leptons exist. Its lifetime would be shorter if there were a fourth generation, allowing for additional decay processes.

A schema displaying all possible interactions between fermions and bosons is shown in Figure 1.2.

All particles have an anti-particle equivalent with the same mass, but opposite charges. Some particles, like the photon, are their own anti-particle.

Noether's theorem states that symmetries in a physical theory result in conservation laws. For example, the invariance under translation in time yields the conservation law of energy. Therefore, theories are often expressed in terms of the mathematical groups $U(n)$ and $SU(n)$ of dimension n, representing these symmetries. Matrices in the group $U(n)$ are unitary, e.g. $\tilde{U}^*U = \mathbf{I}$. Elements of $SU(n)$ additionally have the determinant 1 [Gri87, pp. 117-119].

The SM is fully described by the $SU(3)\times SU(2)\times U(1)$ group, where each component is contributing a description of strong and electroweak interactions. $SU(3)$ results in the eight gluons, mediating the strong force. The remaining $SU(2) \times U(1)$ group describes the four gauge bosons $W^{1,2,3}$ and B. After *electroweak symmetrybreaking*, these eigenstates transform to the photon, $W^{\pm}$-, and Z-bosons [Tan+18, pp. 141, 161]; [Gri87, pp. 366-369].

The *Higgs* boson is the mediator between the Higgs field and quarks, charged leptons, $W^{\pm}$-, and Z-bosons. This interaction is a consequence of the *Higgs mechanism*, resulting in massive particles, which is especially relevant for explaining the existence of massive $W^{\pm}$ and Z bosons [Gri87, pp. 378-381, 401–405]. Until its observation in 2012 [Aad+12], the Higgs was the last predicted particle of the SM without sufficient evidence.

The Standard Model is unlikely to remain the last theory of particle physics since it is not able to address a range of issues:

1. **Parameter problem:** The Standard Model relies on over 20 free parameters, e.g. the quark and lepton masses. Theorists expect a mature model to be able to explain these masses, as opposed to relying on experimental measurements [Gri87, pp. 51-52]; [Man10, p. 483].

2. **Missing gravity:** There is no description of gravity in the SM. An extended SM that integrates gravity would also have to conform to the theory of general relativity [Man10, pp. 483-485].

3. **Dark matter:** Cosmological observations lead to evidence of the existence of *dark matter*. Dark matter interacts mostly through gravity with the SM particles and is only observed indirectly. It has no explanation in the SM [Man10, pp. 506-507].

4. **Hierarchy problem:** In the SM, the Higgs mass only results in the observed 125.2 GeV/c^2 if certain loop corrections cancel each other out in an excessively fine-tuned manner. Without these cancellations, the Higgs mass would be orders of magnitudes larger. Theories exhibiting such large discrepancies and fine-tuned behaviour are expected to be incomplete [Gri87, p. 412]; [Man10, pp. 510-511].

5. **Grand Unification:** The unification of the electromagnetic and weak interaction motivates the search for a theory that also includes the strong interaction. One observation is that the coupling constants α of each interaction are energy-dependent and their values could meet at very high energies of ~10^{16} GeV.

> In the SM, the coupling strength of the strong interaction never reaches the same value as the weak and electromagnetic force. This motivates the search for a new theory, where the coupling constants scale differently and reach the same value at high energies [Man10, p. 508]; [Gri87, pp. 405-406].

All of these issues motivate the construction of BSM (*Beyond Standard Model*) theories and the search for empirical evidence contradicting the Standard Model. One example is the search for Supersymmetry, and in particular, the RPV1L analysis discussed in Part II.

1.2 Proton-Proton Collisions

In a circular proton-proton collider, *bunches* of protons are accelerated in opposite directions and brought to a collision at designated interaction points in the particle detectors. The squared centre of mass energy of two colliding relativistic protons with energies E_1 and E_2, and momenta p_1 and p_2 is given by [Per87]

$$s = 2(E_1E_2 + p_1p_2) + 2M^2 \approx 4E_1E_2 \,.$$

The centre of mass system is at rest in the laboratory if $E_1 = E_2$, and virtually all of the energy is available for new particle creation. The reaction rate is described by

$$R = \sigma L$$

with the *luminosity* L and the interaction *cross section* σ, which is a measure for the probability of an interaction between the two colliding particles. The luminosity of two directed beams of relativistic particles is described by

$$L = fn\frac{N_1N_2}{A} \,,$$

with N_1 and N_2 as the numbers of particles in each bunch, n the number of bunches in each beam, A as the cross-sectional area of the beams, and the revolution frequency f.

Colliding protons are not behaving as single point-like particles but exhibit an internal structure. Richard Feynman first described this structure with the parton model, where a proton is a composite of three valence quarks, of which at least one is involved in the interaction. Additionally, internal quark-anti-quark pairs and gluons occur.

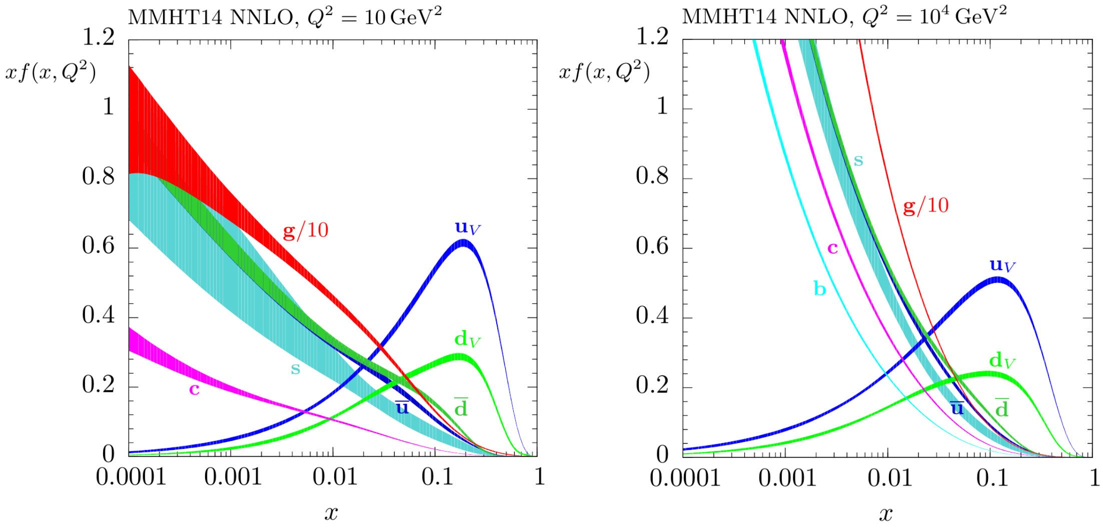


Figure 1.3: Parton density functions for the energy scale $Q^2 = 10$ GeV^2 and $Q^2 = 10^4$ GeV^2 with the associated 68% confidence intervals [Har+15].

Figure 1.3 shows the *parton density functions*, with respect to the fractional momentum x for an energy scale of $Q^2 = 10$ GeV^2 and $Q^2 = 10^4$ GeV^2, describing the probability of a parton contributing to the interaction. At higher energies, the interaction with virtual quarks and gluons in the proton becomes more likely for a small fractional momentum x [Per87]; [Tan+18]; [Har+15]. Interacting protons in collider experiments cause a characteristic radiation of quarks and gluons, forming new hadrons because of *confinement*, allowing quarks to only be stable in a bound state. Such cascading hadronic radiation is directed in a narrow cone and commonly referred to as a *jet* [Per87].

1.3 Interactions of Particles with Matter

High-energy physics experiments rely on particle detectors to identify the type, momentum, and energy of the particles of interest. To detect these particles, they need to interact with the detector medium. Chapter eight of Mann's "*An introduction to particle physics and the standard model*" [Man10, pp. 135-144] is followed to give an overview of common interactions of particles with matter.

Electromagnetically charged particles can interact with the nuclei and atomic electrons of the target matter. The key quantity to characterize these interactions is the *stopping power* $\frac{dE}{dx}$, describing the average loss of energy per depth travelled into the medium. An average range is computed as

$$R = \int_0^E \left(\frac{dE}{dx}\right)^{-1} dE\,.$$

Charged particles have three dominant types of interactions:

Ionization The incident charged particle scatters on atomic electrons in an inelastic process. The atomic electron potentially leaves the bound state with the atom, resulting in secondary electrons traversing the medium. Additionally, the remaining atomic electrons can radiate photons when they occupy the lower free state of the electron that has left.

Ionization through relativistic electrons, i.e. $v = \beta c$, is described by the *Bethe-Bloch* equation:

$$\frac{dE}{dx} = -\frac{4\pi N_A (ze)^2 e^2}{m_e c^2 \beta^2}\left(\frac{Z}{A}\right)\left[\ln\left(\frac{2m_e c^2 \beta^2}{\bar{I}}\gamma^2\right) - \beta^2\right]$$

with $\gamma = (1-\beta^2)^{-1/2}$, ionization potential $\bar{I}$, atomic number Z, nucleon number A, and Avogadro number $N_A = 10^{23} mol^{-1}$.

The number of ionized particles is proportional to the energy loss of the incident particle. The energy loss due to ionization decreases with increasing particle velocity, reaching a minimum value before growing again as $\ln(\gamma^2)$ (see Figure 1.4). At very high energies, $\frac{dE}{dx}$ becomes constant due to long-range interatomic screening effects, and particles become indistinguishable based on ionization loss. The Bethe-Bloch equation stops describing the physical reality at very low energies since it does not account for effects like electron capture or particle-anti-particle annihilation.

Radiation Charged particles emit *bremsstrahlung* if they interact with the charged nuclei in matter. Bremsstrahlung has a stopping power of $\frac{dE}{dx} = -\frac{E}{X_0}$. The *radiation length* X_0 describes the thickness of a given medium after which the energy of an incident particle is reduced by a factor of e (Euler's number). The critical energy for electrons at which ionization and radiation are of the same magnitude is $E_c \approx \frac{600}{Z}$ MeV, with the atomic number Z. Radiation loss is dominant for electrons and positrons at high energies.

Coulomb scattering Electrons and positrons are light enough to scatter on the Coulomb field of nuclei and atomic electrons of the medium. Multiple Coulomb scatterings result in a net scattering angle after traversing the medium. The width of the

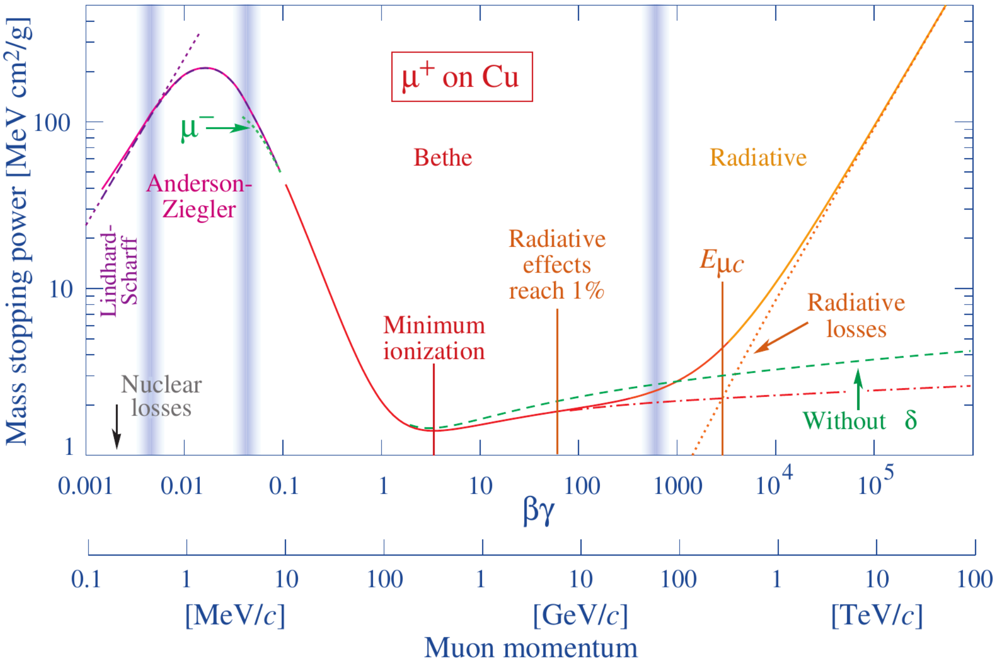


Figure 1.4: Stopping power for μ^+ particles at various momenta in copper ($Z = 29$). Image from "Review of Particle Physics" [Tan+18, p. 447].

angle-distribution is one effect that limits the measurement precision when measuring the direction of detected particles.

Figure 1.4 displays the stopping power $\frac{dE}{dx}$ for μ^+ in copper ($Z = 29$). Ionization is the most dominant effect for muons with a momentum of 10 MeV/c to 100 GeV/c, containing a point of minimum ionization where the stopping power is minimal. At higher energies, radiative losses become more dominant.

Photons have no charge, but as the mediators of the electromagnetic force, they still interact electromagnetically. The loss of intensity can be expressed as $\frac{dI}{dx} = -\mu I$, with μ as the *effective absorption coefficient*. μ^{-1} corresponds to the mean free path for absorption of a photon. Below are the three most dominant effects, expressed in terms of their cross section σ [Gri87, p. 199].

Photoelectric effect An incident photon is absorbed by an atomic electron, which in turn leaves its bound state. This effect dominates at low energies and in media with high atomic numbers:

$$\sigma_{photo} \approx \begin{cases} \frac{Z^5}{E^{7/2}}, \text{ for } E < m_e c^2 \\ \frac{Z^5}{E}, \text{ for } E > m_e c^2 \end{cases} .$$

Compton scattering The incident photon scatters off of an electron in the medium. This process is dominant at intermediate energies since the photon energy is too high to be absorbed by shell electrons and too low for pair production. The cross section of Compton scattering is estimated by

$$\sigma_{compton} \approx \frac{Z}{E}.$$

Pair production A photon with the energy $E > 2m_e \approx 1.022\,\text{MeV}$ can produce an electron-positron pair. Because of momentum conservation, this process cannot happen in free space. The photon needs a Coulomb field in the medium to recoil. This effect dominates for photons with $E > 10\,\text{MeV}$.

Figure 1.5 shows the cross sections for photons with different energies in carbon ($Z = 6$) and lead ($Z = 82$). The photoelectric effect is responsible for the various spikes, which depends on the number of atomic electrons and corresponding energy states. Compton scattering becomes dominant at around 10 to 100 keV. Pair production can occur while interacting with the Coulomb fields of nuclei or shell electrons (κ_{nuc} and κ_e) [Tan+18, p. 454].

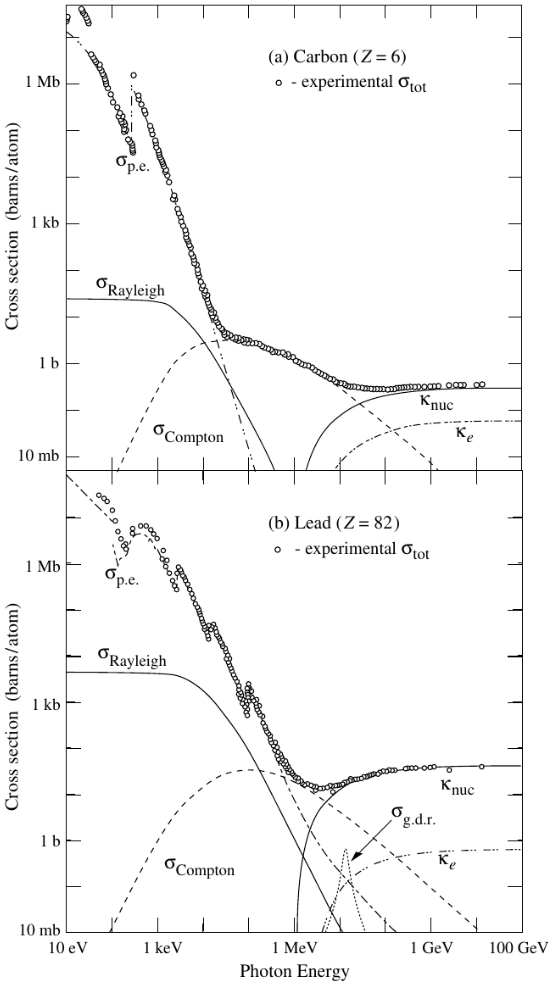


Figure 1.5: Cross section of photons at different energies in carbon ($Z = 6$) and lead ($Z = 82$). Image from "Review of Particle Physics" [Tan+18, p. 454].

2 The ATLAS Experiment at the LHC

The LHC [EB08] is a synchrotron accelerator for positively charged hadrons (protons or lead ions). The circular structure repeatedly accelerates the hadrons, increasing their kinetic energy with each cycle. Due to the increased kinetic energy, the bending strength of the accelerator magnets must be adjusted, such that the particle beam radius remains constant, hence the name *synchrotron* [Man10, pp. 118-120].

The particles are injected into the LHC ring in *bunches*, containing up to $\sim 10^{11}$ protons per bunch. These bunches travel through the ring in both directions, with defined *interaction points* at the experiment locations. Collisions are then a statistical process, where some particles in these two passing bunches interact, and others do not. An *event* is the measurement of the resulting particles of a single bunch crossing. Bunches are spaced, such that a *bunch crossing* occurs every 25 ns. As a consequence, experiments are exposed to collisions at a rate of up to 40 MHz. Multiple interactions can occur during the same bunch crossing, resulting in *in-time pile-up* interactions that superimpose in a given measurement [EB08]. *Out-of-time pile-up* refers to particles from uncorrelated interactions of previous or following bunch crossings. For example, the mean number of interactions per bunch crossing in 2018 averaged to $\langle \mu \rangle = 36.1$ [Expb]. An example of the resulting amount of data is given in Figure 2.1, displaying the reconstructed vertices and tracks with $p_T > 100$ MeV/c (see Chapter 3) for an event with 65 simultaneous interactions.

The original target peak luminosity for ATLAS and CMS was $L = 10^{34}\text{cm}^{-2}\text{s}^{-1}$. Experiments commonly report integrated luminosities for a given time period in the unit of inverse barn, e.g. $1\text{fb}^{-1} = 10^{39}\text{cm}^{-2}$, which is a measure for the amount of collected data.

The LHC operates during scheduled campaigns, called *Runs*. The accelerator and experiments have the opportunity to repair and upgrade their facilities during *Long Shutdowns* (LS), between each Run. Run 1 and Run 2 occurred during 2010 − 2013 and 2015 − 2018, respectively. Run 3 is scheduled for the 2022 − 2024 period. The LHC will be upgraded during LS3, to increase the delivered luminosity, i.e. the number of particle interactions per bunch crossing. This is referred to as the *High Luminosity LHC*

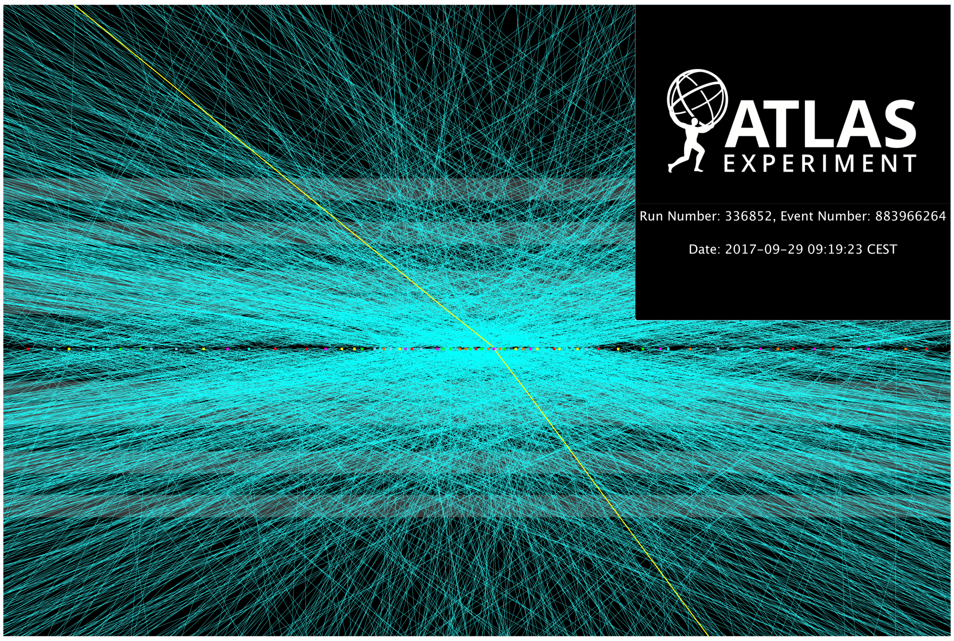

ATLAS
EXPERIMENT
Run Number: 336852, Event Number: 883966264
Date: 2017-09-29 09:19:23 CEST

Figure 2.1: $Z \longrightarrow \mu\mu$ candidate event with 65 additional reconstructed vertices recorded in 2017 [Expa]. Tracks with p_T > 100 MeV/c are displayed.

(HL-LHC) [Pro]; [15].

The ATLAS Experiment is one of two general-purpose detectors at the LHC. It consists of multiple barrel-shaped layers around the interaction point to track the path and measure the momentum of post-collision particles. Electromagnetic and hadronic calorimeters measure the energy deposition of the produced particles. A muon tracking system improves the muon detection accuracy, which is motivated by many physics studies [Col08]. Figure 2.2 contains an exemplary proton-proton collision event recorded in the ATLAS detector.

First official collaboration activities of ATLAS started in 1992, and the experiment construction finished in 2008 [Col08].

In Run 1, ATLAS recorded 47 pb^{-1}+5.5 fb^{-1} of proton-proton collisions with a centre-of-mass energy of $\sqrt{s}$ = 7 TeV [Aad+13] and 22.7 fb^{-1} at $\sqrt{s}$ = 8 TeV [Aab+16]. During Run 2, ATLAS recorded proton-proton collisions at an integrated luminosity of 139 fb^{-1} at $\sqrt{s}$ = 13 TeV [19b]. The integrated luminosity for Run 2 is stated after data quality selections, while the luminosity for Run 1 is the total amount of recorded events.

The ATLAS detector is upgraded between each Run of the LHC and is expected to operate at least until the end of the 2030s. During LS1, the inner detector was extended from 3 to 4 cylindrical pixel layers through the addition of the *Insertable B-Layer* (IBL). The IBL improves the vertex- and b-tagging performance, which tag particle objects

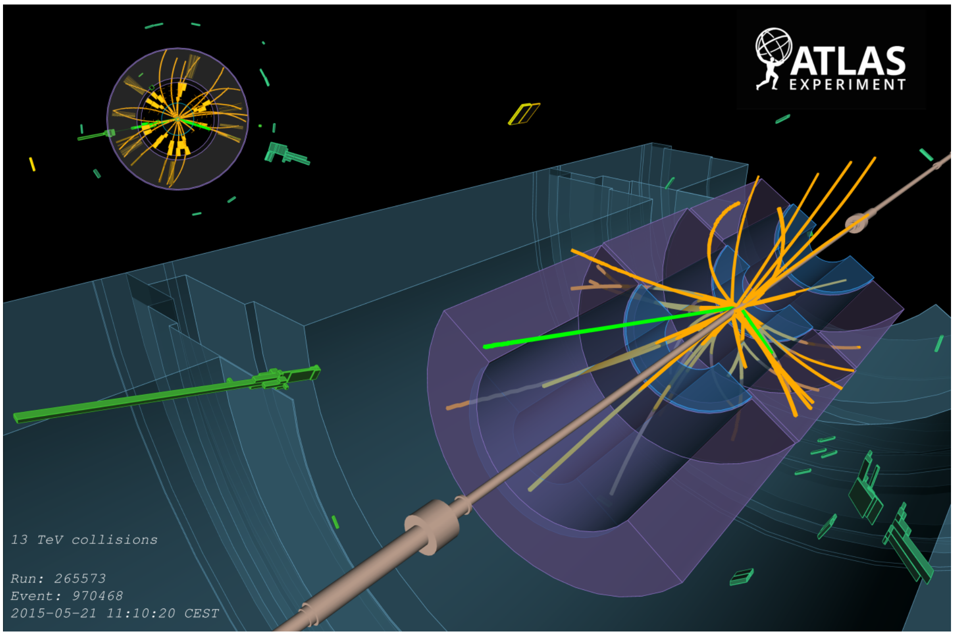


Figure 2.2: Event display of a $Z \longrightarrow e^{+}e^{-}$ candidate, at $\sqrt{s} = 13$ TeV. The arcs in the central section represent reconstructed tracks of particles. The stacked bars outside are proportional to the energy deposition in the electromagnetic calorimeters. The high energy electron-positron pair (green) is likely to be produced from a Z-boson decay [Col15].

with their corresponding original p-p collision position (vertex) and if they are a decay product of a bottom quark [ATL10]; [Abb+18]. During LS2, the installation of the *Muon New Small Wheel* and updates to the *Liquid Argon Calorimeter*, and *TDAQ* systems are planned and currently ongoing [11].

Before Run 4, as preparation for the HL-LHC, a thorough upgrade of the detector is planned. This includes a new tracking detector (ITk), updates to the calorimeter and muon spectrometer readout systems, and a new trigger architecture [ATL12]; [15].

An important goal of any LHC experiment in the coming Runs 3 and 4 is the search for hints of "new physics" (NP). The null observation of NP would be valuable as well since it would be "falsifying some of the paradigms that guided research in fundamental physics so far" [Cid+19]. Examples for NP studies are the search for Supersymmetry, exotic particles and dark matter. The study of flavour in the Standard Model, why masses between fermion families are so different, and the nature of neutrino masses, may lead to hints of NP as well [Cid+19].

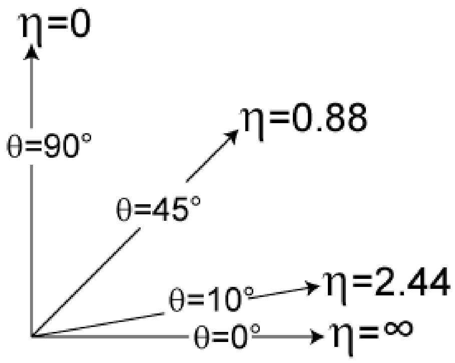


Figure 2.3: Pseudo-rapidity in forward direction [Jab07].

2.1 Coordinate System

The conventional coordinate system used in the context of ATLAS is right-handed, with its origin at the nominal interaction point at the centre of the detector [Col08]. This coordinate system aligns the z-axis along the beam direction, leaving the x-y-plane in a transverse direction. The positive x-axis points to the centre of the LHC ring, while the positive y-axis points upwards. Side A of the detector is located on the positive end on the z-axis. Consequently, side C lies on the negative side of the z-axis, facing west.

The azimuthal angle ϕ defines the rotation in the transversal x-y-plane around the beam axis. The polar angle θ is the angle from the beam axis. $\Delta R \equiv \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is often used to measure angular distances. The initial momentum of a proton-proton collision in the z-direction is unknown, which is motivating projections of related quantities in the x-y plane, e.g. $p_T = p \sin(\theta) = \sqrt{p_x^2 + p_y^2}$ and $E_T = E \sin(\theta)$. A commonly used measure is the *pseudo-rapidity* $\eta = -\ln\tan(\theta/2)$ (depicted in Figure 2.3). The pseudo-rapidity is useful since the total particle production cross section is following a flat distribution in η.

2.2 Detector and Trigger Systems

The detector description in this section follows the original paper “The ATLAS Experiment at the CERN Large Hadron Collider” [Col08]. Figure 2.4 illustrates the cylindrical detector layout and the major constituents of ATLAS.

The *Inner Detector* (ID) consists of the silicon *Pixel Detector*, the *SemiConductor Tracking Detector* (SCT) and the *Transition Radiation Tracker* (TRT), enabling charged particle track reconstruction in the range of $|\eta| < 2.5$. The silicon pixel detectors cover the central collision region and provide four measurements (including IBL [ATL10];

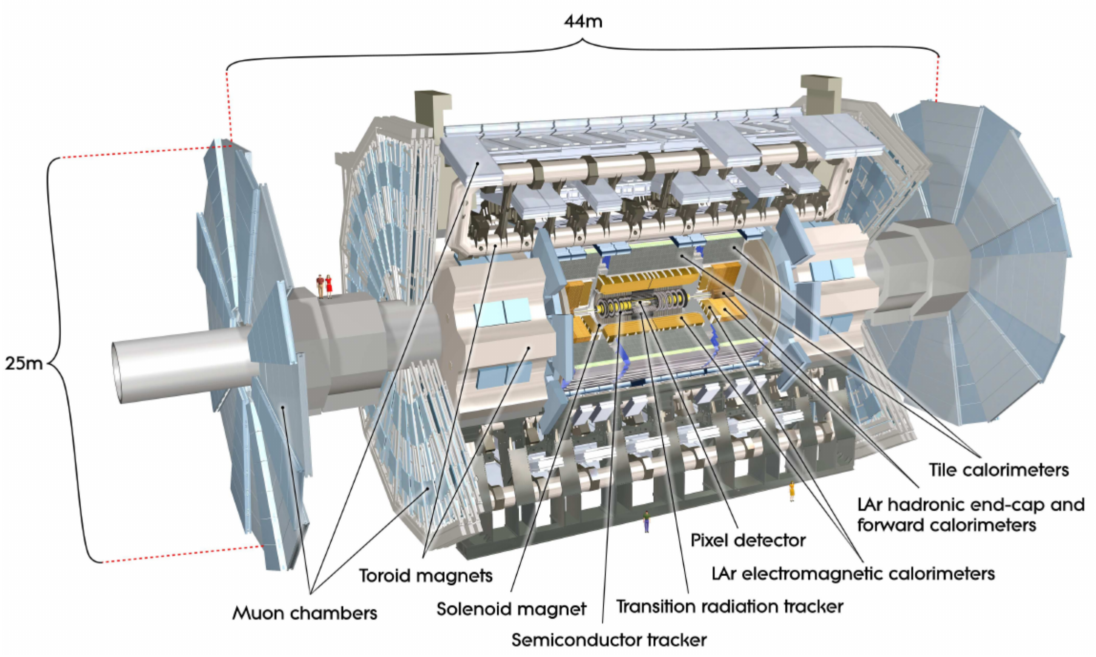

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

Figure 2.4: Illustration of the ATLAS detector and its constituents [Col08].

[Abb+18]). Eight additional hits per track are provided by the SCT. The TRT enables radially extended track reconstruction in the region of $|\eta| \leq 2.0$ and provides electron identification information.

Located around the ID is the solenoid magnet, creating a relatively homogeneous magnetic field with a flux density of 2 T in the centre. Eight barrel segments and two end-caps form the toroid magnets, which surround the calorimeters and create a magnetic field with a flux density of 0.5 T and 1 T for the muon detectors.

The calorimeter subsystem consists of detectors for electromagnetic and hadronic energy deposition measurements. Electromagnetic calorimetry is done by *Liquid-argon* (LAr) detectors with fine granularity for the η-range of the ID. The hadronic calorimetry is covered by scintillator tile calorimeters, separated into a larger centre barrel and two small extended barrel cylinders on either side of the centre. Both systems include multiple layers of end-caps to close the barrel region, where LAr detectors are also used for the hadronic calorimeters.

Outside of the calorimeter and partially immersed in the toroid magnets, is the muon spectrometer. It is composed of multiple trackers, again arranged in a barrel with end-cap shape, ensuring high precision measurements of muon tracks. The magnetic field is orthogonal to a muon trajectory in the muon spectrometer and has a bending power of 1.5 to 5.5 Tm in the region of $|\eta| < 1.4$. The bending power can go up to 7.5 Tm in the region of $1.6 < |\eta| < 2.7$.

The trigger system is reducing the rate of recorded events by selecting events with a pre-defined menu of observable properties. The first-level trigger is implemented in hardware and reduces the event rate to <100 kHz. A high-level trigger (HLT) reduces the number of recorded events to disk to about 1 kHz, by using commodity computer hardware to perform a fast reconstruction of physics objects. This reduction is essential to reduce the data rate to a manageable scale while still recording the most relevant events [ATL17c].

3 Event Reconstruction

Particles are reconstructed from detector measurements by combining the information of all detector systems. Charged particle tracks are reconstructed from hits in the inner detector and muon spectrometer layers. Energy deposits are measured in the calorimeters, mainly separating electromagnetic and hadronic processes. Each particle type, e.g. electrons or jets, is reconstructed differently because of their physical properties and to improve the reconstruction efficiency in the presence of their corresponding background processes.

This Chapter focuses on electron, muon, and (b-tagged) jet reconstruction in preparation for the RPV1L analysis, discussed in Part II.

3.1 Track and Vertex Reconstruction

Charged particle trajectories (*tracks*) are the basis for multiple particle reconstruction algorithms. Tracks with $p_T > 0.4$ GeV/c are reconstructed in the region of $|\eta| < 2.5$ by using the measurements of the inner detector [ATL17b]. The particle trajectory is bent in the magnetic field of the Solenoid with a resulting curvature inversely proportional to the momentum of the particle.

In a pre-processing step, the raw data from the pixel and SCT detectors is converted to clusters by grouping sensor readings where the deposited energy yields a charge above a threshold. The timing information of the TRT raw data is converted into calibrated drift circles.

Seeds are formed by three hits in the pixel and SCT detectors, with added requirements to increase purity and resolve ambiguities, e.g. hits used in multiple track seeds. A combinatorial Kalman filter extends the seeds into the following SCT and TRT layers to form track candidates.

The candidates are fitted, and ambiguities between track candidates are resolved by favouring tracks with matching clusters, high fit quality, and hits in each subsequent layer. Selected tracks are extended into the TRT before a final fit with the full information of all three detectors is done, and the quality of the tracks is compared to the silicon-only track candidates. A resulting track is expressed in terms of ϕ, θ, charge over momentum ($\frac{q}{p}$), and the impact parameters d_0 and z_0, describing the point of closest approach in the transverse and longitudinal plane to the primary vertex.

Reconstructed tracks are used to find the *primary vertex* of the proton-proton collision [ATL17d]. First, the tracks are used to create a vertex seed, assuming it to be located in the beam spot in the x-y-plane and using the mode of all z-coordinates of the tracks closest to the reconstructed beam spot centre. The seed is the starting point for an iterative fit where only the most compatible tracks contribute in each iteration. Incompatible tracks are removed and used to find additional vertices by repeating the same procedure. Only vertices with at least two associated tracks are valid primary vertex candidates. The candidate with the highest sum of squares of p_T is considered to be the primary vertex of the event.

3.2 Electron Reconstruction

Electrons[1] are created from the detector data in multiple steps [ATL19b]. The particle is reconstructed from tracks in the inner detector and clusters in the electromagnetic (EM) calorimeter. Additionally, it needs to be identified as a prompt electron, i.e. originating from the primary collision and isolated from other activities in the detectors. Finally, the electron charge is studied to suppress charge misidentifications.

All of these steps have corresponding selection efficiencies, resulting in a total efficiency for the measurement of electrons:

$$\varepsilon_{total} = \varepsilon_{EMclus} \times \varepsilon_{reco} \times \varepsilon_{id} \times \varepsilon_{iso} \times \varepsilon_{trig} = \left(\frac{N_{cluster}}{N_{all}}\right) \times \left(\frac{N_{reco}}{N_{cluster}}\right) \times \left(\frac{N_{id}}{N_{reco}}\right) \times \left(\frac{N_{iso}}{N_{id}}\right) \times \left(\frac{N_{trig}}{N_{iso}}\right),$$

with the efficiencies:

- ε_{EMclus}: Detection of electromagnetic calorimeter clusters from all produced electrons. This is estimated in simulations.
- ε_{reco}: Reconstruction of electrons from all detected clusters.
- ε_{id}: Identification of prompt electrons from all reconstructed electrons.
- ε_{iso}: Isolated electrons with respect to all identified electrons.

[1] In the context of reconstruction, this refers to both electrons and positrons.

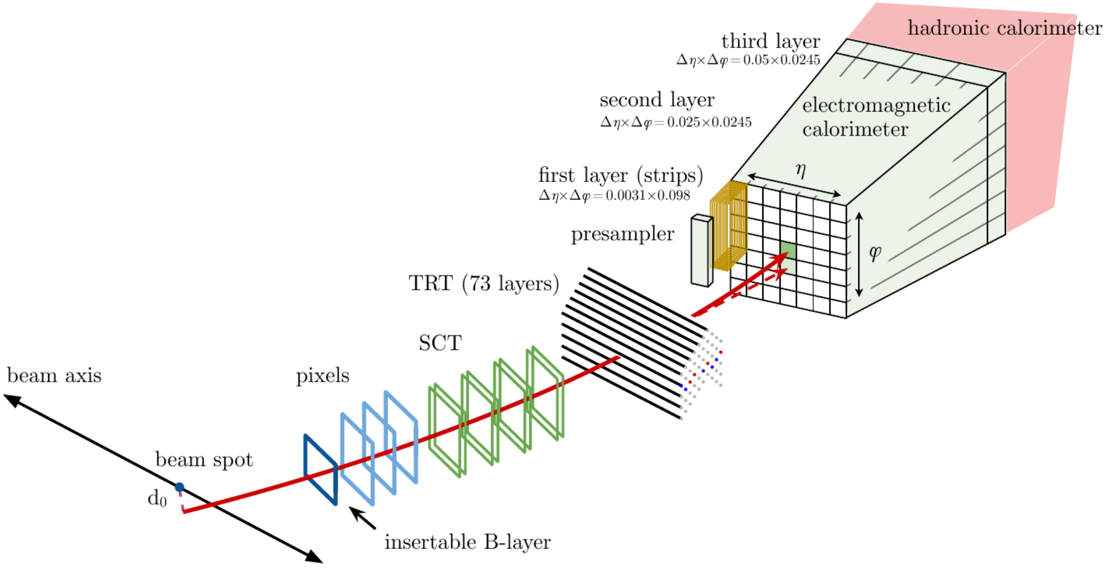


Figure 3.1: Illustration of the trajectory of an electron through the detector. The red line is the hypothetical path of an electron traversing the detectors. The dashed line is an exemplary path of a photon produced by the interaction of the electron with the material. [ATL19b]

- ε_{trig}: Isolated electrons with a matching trigger.

The selection efficiencies result in correction factors for experimentally determined electrons that need to be applied in every analysis. $Z \longrightarrow ee$ and $J/\Psi \longrightarrow ee$ decays are used to get unbiased samples of electrons via the tag-and-probe method[2], to get precise estimates of all selection efficiencies. Isolated electrons, selected for physics, are subject to large backgrounds from misidentified hadrons, electrons from photon conversion and non-isolated electrons from heavy-flavour decays. Correction *weights* for simulated events are extracted to match the measured efficiencies in data. Additional event weights, e.g. for the observed pile-up distribution, are required to achieve a good agreement between simulated events and events from data.

3.2.1 Reconstruction

The electron reconstruction, schematically displayed in Figure 3.1, is done by associating identified charged particle tracks in the inner detector to a localized cluster of deposited energy in the electromagnetic calorimeter. Extrapolated tracks are required to closely match to EM clusters in $\eta \times \phi$ space to form a final electron candidate.

[2] Also see Chapter 7 for a description of the tag-and-probe method.

EM calorimeter clusters without an associated track are reconstructed as photons. Electron reconstruction is done in the range of $|\eta| < 2.47$.

Cluster seeds in the EM calorimeter are formed by combining localized deposited energy in an $\eta \times \phi$ grid of size 3×5. The centre of the 3×5 grid moves in small steps until the whole calorimeter is covered. The energy of a reconstructed cluster is estimated in a window of 3×7 in the barrel region and 5×5 in the end caps. If two clusters are found in close proximity, i.e. overlap in an area of $\Delta\eta \times \Delta\phi = 5 \times 9$, the candidate with greater E_T is retained, if it is at least 10% larger. If it is within 10%, the candidate with the highest E_T in the centre of the cluster is kept.

Track candidates are fit with a special treatment for trajectories with larger losses due to bremsstrahlung. If multiple tracks match to the same calorimeter cluster in $\eta\times\phi$, the primary electron track is identified by considering the number of hits in the inner detector, E/p and p_T, and the associated vertex.

The energy of the electron candidate must be calibrated and is computed with the energy deposited in the calorimeter and the η and ϕ parameters of the associated track. Above $E_T = 15$ GeV/c, the electron reconstruction efficiency for good quality electrons, i.e. at least one hit in a pixel layer and at least seven hits in all silicon layers, varies from 97% to 99%.

3.2.2 Identification

Prompt electrons in the central region of $|\eta| < 2.47$ are identified through a likelihood-based (LH) method. This method uses measurements from the tracking and calorimeter systems and combined quantities. Multiple working points are defined to balance identification efficiency and background contamination through secondary electrons from photon conversion, misidentified electrons from multi-jet[3] events, and other sources of non-prompt electrons that are misidentified. The working points are called *Loose* (93% efficient for electrons with $E_T = 40$ GeV/c), *Medium* (88%), and *Tight* (80%). All working points require at least two hits in the pixel layer and seven hits in all silicon layers. Medium and Tight also require a hit in the innermost layer of the pixel detector, which increases the likelihood to identify a prompt electron originating from the proton-proton collision.

[3] Also see Chapter 7.

3.2.3 Isolation

To isolate prompt electrons from background processes, special isolation variables from calorimeter and inner detector measurements are introduced to quantify the differences in activity. A calorimeter based isolation variable is set by computing the deposited energy in a cone around the electron candidate cluster, with the energy in the core cells of the cluster subtracted. The track-based isolation variable is computed by summing the p_T of all tracks within a cone of radius ΔR, satisfying basic track quality requirements from the primary vertex. The contribution of the electron candidate track is subtracted, including extrapolated tracks close to the EM calorimeter cluster, to combine bremsstrahlung effects with the primary electron candidate. Multiple working points are introduced to combine both variables and isolate electrons with a known efficiency, e.g. *Fix (Tight)* with the requirements $E^{isol}_{T,cone}/p_T < 0.2$ and $p^{isol}_T/p_T < 0.15$.

A new isolation method to isolate prompt leptons was introduced during Run 2 [19a]; [Aab+18], which is the used method of isolation in the RPV1L analysis, discussed in Chapter 6. This method uses a boosted decision tree (BDT) to separate prompt and non-prompt leptons by using tracks of jets nearest to the lepton, as well as lifetime and isolation observables. The prompt lepton veto (*PLV*) method has a strong separation power for leptons with $p_T > 12$ GeV/c. A second low-p_T-BDT is used to increase the isolation efficiency below $p_T = 12$ GeV/c.

3.2.4 Charge Identification

The charge of an electron candidate is reconstructed from the curvature of the associated track in the inner detector. Misidentification can occur if the electron candidate is matched to an incorrect track, if the curvature is measured incorrectly, or if three tracks from pair production are assigned incorrectly. To suppress charge misidentification, a BDT is used, trained on track variables of single-electron events.

3.2.5 Calibration

The absolute energy scale of electrons has to be calibrated [ATL19a]. The impact of material in front of the calorimeter is minimized by applying a multivariate regression algorithm trained on simulated events, relying on an accurate description of the material in front of the calorimeters. The relative energy scales of different EM calorimeter layers is calibrated through studies of muon energy deposits and electron showers. This step is essential to correct the extrapolation of the energy to the full energy-range of electrons. Residual non-uniformities in the calorimeter response are corrected, and correction factors to match simulated samples to data are estimated.

The final energy estimate is derived from a large sample of events with $Z \rightarrow ee$ decays by comparing invariant mass distributions in data and simulation and extracting correction factors. Validation is done on independent samples from $J/\Psi \rightarrow ee$ decays. The accuracy is found to vary between 0.03% and 0.2%, depending on $|\eta|$. The calorimeter energy scale is stable with respect to time and changes in luminosity, with observed differences of up to a few per-mille in the endcap and less than one per-mille in the barrel calorimeter.

A coarse energy calibration is done during reconstruction, but analyses need to apply an improved calibration again, derived from a large data set recorded during the Run.

3.3 Muon Reconstruction

Muons are identified through charged particle tracks in the inner detector[4] (ID) and tracks in the muon spectrometer (MS), located in the outer sections of ATLAS [Aad+16a]. Tracks in the MS are reconstructed by analysing hit patterns in the muon chambers to form segments of multiple hits. Track candidates are defined by taking hit multiplicities and fit quality into consideration. At least two matching segments are required to build a track, except for the barrel-endcap transition region, where only a single segment is required. ID and MS tracks are used individually and in a combined way to define multiple types of muons:

- Combined (CB): Tracks from ID and MS are reconstructed independently. A global fit is run to yield the final track. MS hits may be added or removed to improve the fit.
- Segment-tagged (ST): An ID track is classified as a muon if its extrapolation to the MS has at least one associated segment.
- Calorimeter-tagged (CT): An ID track is classified as a muon if it is matched to an energy deposit in the calorimeter compatible with the minimum-ionizing-particle.
- Extrapolated (ME): The Muon trajectory is reconstructed based on MS tracks and lose requirements to select tracks originating from the interaction point. ME

[4] As described in Sections 3.1 and 3.2.

muons are used to extend the acceptance of muon reconstruction in the region of $2.5 < |\eta| < 2.7$, not covered by the ID.

Overlaps between muon types are resolved, giving priority to CB over ST over CT muons.

To suppress backgrounds, mainly from pion and kaon decays, muons are identified by analysing helper variables. The variables are q/p significance, defined as absolute difference between the q/p ratios of ID and MS, a comparison of ID and MS track momentum $\rho' = \frac{|p_T^{ID} - p_T^{MS}|}{p_T^{combined}}$, and the χ^2 of the combined track fit. With these variables and requirements on the number of hits in ID layers, four working points are defined: *Loose*, *Medium*, *Tight*, and *high-p_T*. The Loose working point considers all muon types, with CT and ST muons restricted to $|\eta| < 0.1$, CB muons in the range of $|\eta| < 2.5$, and ME muons for $2.5 < |\eta| < 2.7$. Medium only accepts CB and ME muons with requirements on ID and MS track momentum to suppress contamination from misidentified hadrons. Tight maximizes purity at the cost of selection efficiency by additional requirements on the combined fit quality and a two-dimensional cut on $\frac{q}{p}$ and ρ'. The High-p_T working point maximizes momentum resolution above 100 GeV/c. It requires CB muons to have enough hits in the MS and removes tracks in regions with suboptimal alignment.

Muons from heavy particle decays, e.g. W, Z, or Higgs, are often isolated from other particles. On the other hand, muons from semi-leptonic decays are often embedded in jets. For this reason, isolation requirements result in a good separation between "signal" and "background" muons. Similar to electrons, track-based and calorimeter-based isolation variables are defined. $p_T^{varcone30}$ is the scalar sum of all track p_T in a cone of variable size $\Delta R = min(10\,\text{GeVc}/p_T^{\mu}, 0.3)$, excluding the primary muon track. The calorimeter-based variable is $E_T^{topocone20}$, the sum of all E_T of the clusters in a cone of size $\Delta R = 0.2$, subtracting the contribution from the muon itself and pile-up effects. Multiple isolation working points are defined, e.g. *FixedCutLoose* with $p_T^{varcone30}/p_T^{\mu} < 0.15$ and $E_T^{topocone20}/p_T^{\mu} < 0.3$.

As for electrons in Section 3.2, the new *prompt lepton veto* working point is introduced and serves as the main isolation technique in the analysis discussed in this dissertation.

Finally, the muon momentum scale is calibrated with $J/\Psi \to \mu\mu$ and $Z \to \mu\mu$ decays. Simulated samples are corrected to improve the agreement to data and minimize systematic uncertainties. The muon reconstruction efficiency is close to 99% for muons with $5 < p_T < 100$ GeV/c in the region of $|\eta| < 2.5$. The isolation efficiency is between 93% and 100%.

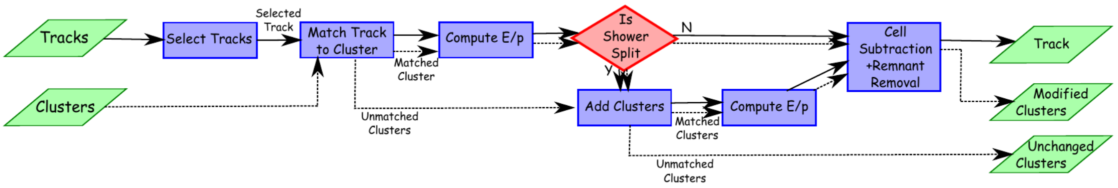


Figure 3.2: Flow chart of the particle flow method, used during jet reconstruction. [Aab+17a].

3.4 Jet Reconstruction and b-Tagging

The reconstruction of hadronic jets is implemented with the *particle flow* (PFlow) method and the anti-k_t algorithm [Aab+17a]. The PFlow method was introduced during Run 2 and improves the jet reconstruction by using the momenta of charged hadrons from inner detector tracks instead of the deposited energy in the calorimeters. This improves the accuracy and resolution of charged hadron measurements and improves the stability to pile-up interactions. Low-p_T charged particles from hadronic jets are included, which otherwise would be located outside of the jet cone due to the magnetic field in the inner detector. Flavour tagging is applied to mark jets originating from heavy-flavour quark decays, especially from b-quarks.

3.4.1 Jet Reconstruction

Jet reconstruction is performed on particle flow objects, consisting of calorimeter clusters and tracks originating from the primary vertex. In the forward region, only calorimeter clusters are used for particle flow reconstruction since no tracks are available. Calorimeter clusters are constructed from lateral and longitudinal segments, which allows for a three-dimensional reconstruction of particle showers, called *topological clusters.*

A flow diagram of the particle flow method is presented in Figure 3.2. Good quality tracks are selected with at least nine hits in the silicon detectors, with no missing pixel hits. Tracks with $p_T > 40$ GeV/c are excluded, since energetic particles are often poorly isolated from nearby activity. Tracks matching with Medium electrons or muons are not selected since tracks from charged hadrons are targeted. Next, tracks are matched to a single topological cluster with criteria on $\Delta\eta$ and $\Delta\phi$. The expected deposited energy in the calorimeter is computed based on the topological cluster position and the track momentum. It is used to estimate the probability of multiple topological clusters belonging to a single particle and split particle showers are recovered. It is also used to subtract the particle contribution to the cluster on a cell-by-cell basis or remove clusters with exceeding energy contributions. This cell-by-cell subtraction is performed around the extrapolated track position and a parameterized shower shape, extracted from simulated single $\pi^{\pm}$ particles. Rings are formed around the $\eta \times \phi$ position, and an averaged energy density per ring is compared to the expected energy deposit of the particle. Finally, checks on the remaining energy of the topological clusters are performed to identify if the shower is caused by a single particle or if multiple particles deposited energy in the volume.

Particle flow jets are then reconstructed with the anti-k_t algorithm with a radius of $R = 0.4$ [CSS08]. In the algorithm, the distances $d_{ij} = min\left(k_{ti}^{-2}, k_{tj}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2}$ and $d_{iB} = k_{ti}^{-2}$, between two pseudo-particles i and j, and between i and the beam (B) are introduced, with k_t as the transverse momentum of the pseudo-particle. If d_{ij} is smaller, i and j are recombined, and if d_{iB} is smaller, i is declared a jet and is removed from the list of candidates. The distances are recalculated, and the procedure is repeated until no pseudo-particles are left.

The resulting jet objects are calibrated over a range of $20 < p_T < 1500$ GeV/c. Multiple calibration stages restore the jet energy scale to that of truth jets [ATL17a]. For example, this includes corrections of jet directions with respect to the reconstructed primary vertex, corrections to pile-up effects, and calibration of simulated samples. *In situ* calibrations, applied to data, use well-measured reference objects, e.g. Z-bosons, to calibrate jet objects.

The *jet vertex tagger* (JVT) helps to suppress jets from pile-up [Aad+16b]. It uses information from associated tracks to reject jets not coming from the primary vertex.

3.4.2 Flavour Tagging

Flavour tagging refers to the process of identifying jets originating from decaying b-, c- or light-flavour hadrons. *B-tagging*, i.e. jets originating from a b-hadron, is of special interest since it is a major discriminant for the analysis discussed in Chapter 6. Other flavoured jets (c-hadrons and light-flavour jets), are the largest background of b-tagging [Aad+19]; [17]. Multiple b-tagging algorithms are applied, exploiting the long lifetime, high mass and high decay multiplicities of b-hadrons, and properties of the b-quark fragmentation. Due to their lifetime, b-hadrons have a significant mean flight length, leading to at least one displaced vertex, different from the primary vertex.

Low-level algorithms reconstruct the characteristic features of b-jets by either using individual properties of charged particle tracks associated with a hadronic jet or by combining multiple tracks to reconstruct displaced vertices. In a second stage, the low-level b-tagging results are combined in high-level algorithms using multivariate classifiers.

The DL1 high-level algorithm is based on a neural network, producing probabilities for a jet to be a b-jet, c-jet or a light-flavour jet. It provides multiple operation points, with efficiencies for b-tagging of 85%, 77%, 70%, and 60%, where lower efficiencies have improved background rejection performance. A recent improvement to the DL1 algorithm, referred to as DL1r, also uses the output of a new low-level algorithm, which is a recurrent neural network exploiting the correlation between track impact parameters [Cola].

3.5 Missing Transverse Momentum

Momentum conservation in the transverse plane implies that the transverse momenta of all particles should sum up to zero [18]. Imbalances indicate invisible particles, e.g. neutrinos or new weakly interacting particles. To estimate this imbalance, the missing transverse momentum, $\vec{E}_T^{miss}$, is reconstructed from all identified and calibrated particles in the x and y component:

$$E_{x(y)}^{miss} = E_{x(y)}^{miss,\mu} + E_{x(y)}^{miss,e} + E_{x(y)}^{miss,\gamma} + E_{x(y)}^{miss,\tau} + E_{x(y)}^{miss,jets} + E_{x(y)}^{miss,soft}.$$

Each term is calculated as the negative vectorial sum of p_T of energy deposits or charged particle trajectories of the corresponding particles. Energy deposits in the calorimeters and tracks are matched to reconstructed objects to minimize double-counting. Ambiguities between jets and electrons or photons and jets and muons are resolved. Electrons have priority over photons, taus and jets, and the latter are rejected if they share a calorimeter signal. For muons, certain criteria on track and vertex parameters are defined to isolate them from pile-up jets or excessive loss of energy in the calorimeter. Signals not associated with the reconstructed particles are collected in the *soft* term. The magnitude of missing transverse momentum is also attributed with a direction in the transverse plane, called ϕ^{miss}.

4 Data Formats and Processing in ATLAS

There are many data formats used in ATLAS particle reconstruction and analyses. The description of how these data formats are used at different stages of ATLAS data processing follows the original "*ATLAS Computing: technical design report*" [05].

4.1 Overview

Detected events passing the high-level trigger are streamed to the *Tier-0* computing facility as *RAW* data. The RAW data format is processed into the *Event Summary Data* (ESD) during *reconstruction*, which creates physics objects from "bare-metal data" recorded by the detector, as described in Chapter 3. AODs (*Analysis Object Data*) are derived from ESDs, containing potentially redundant, overlapping and ambiguous objects. For example, this could be a single electron that is also contained in a given jet cone.

Simulated data is using *generators* to create events at the interaction point, which are then propagated through a simulated detector. Recorded *hits* in the simulated detector material are then combined with *digitization* and *pile-up* effects to produce the *Raw Data Object* (RDO). Digitisation refers to the simulation of the detector read-out electronics. The trigger response is simulated on RDOs, which also produces a RDO format. Finally, RDOs go through the same reconstruction chain as RAW data to produce simulated AODs. Figure 4.1 shows a diagram of the simulation workflow with intermediate data formats.

A common software framework, *Athena* [Col19], is used for all these central processing steps. Analyses can be done directly in Athena, but it is also common for physics groups to develop their own analysis frameworks on top of the software package *ROOT* [Bru+19].

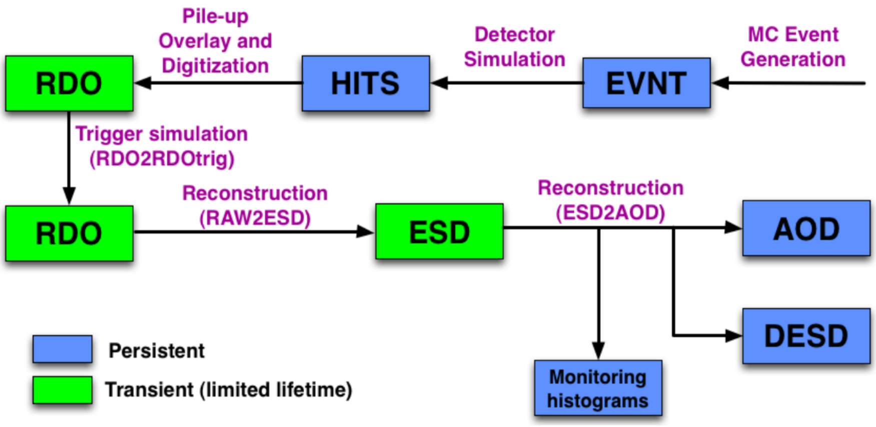


Figure 4.1: Simulation workflow in ATLAS. EVNT files contain events from generators that are propagated through the detector. Resulting HITS are digitized and overlaid with other HITS to simulate the detector response. The resulting RDO format is used as if it were the RAW data coming from the detector (including "Truth" information) [Bir+14].

4.2 Analysis Model in Run 2

The computing Model was updated for Run 2, as described in "*Update of the Computing Models of the WLCG and the LHC Experiments*" [Bir+14]. Most notably, a new *Event Data Model* (EDM) is implemented to replace the old AOD format. This new *xAOD* format was designed to be readable with a minimal amount of software packages, even without using the Athena framework.

Additionally, *dual-use tools* were introduced to provide the recommendations of *Combined Performance* (CP) groups both inside and outside of the Athena framework [FAR+15]. Each CP group implements the physics recommendations for their corresponding physics objects via CP tools, e.g. energy calibration, jet vertex or flavour tagging, as described in Chapter 3. CP tools have a harmonized interface to process single physics objects or containers of these objects and provide systematic uncertainties for their operation. Analyses implement basic object treatment by calling CP tools in C++ code on an event-by-event basis.

Systematic uncertainties are conventionally referred to by a string-name and applied in 1σ up or down variations. Each CP tool can check if it is affected by a given variation and report recommended systematic variations to consider.

Figure 4.2 shows the updated analysis workflow for Run 2. The xAOD output, produced in central production, can be used directly for analyses. Additionally, intermediate derived formats (DxAODs) that reduce the footprint of xAODs are prepared for

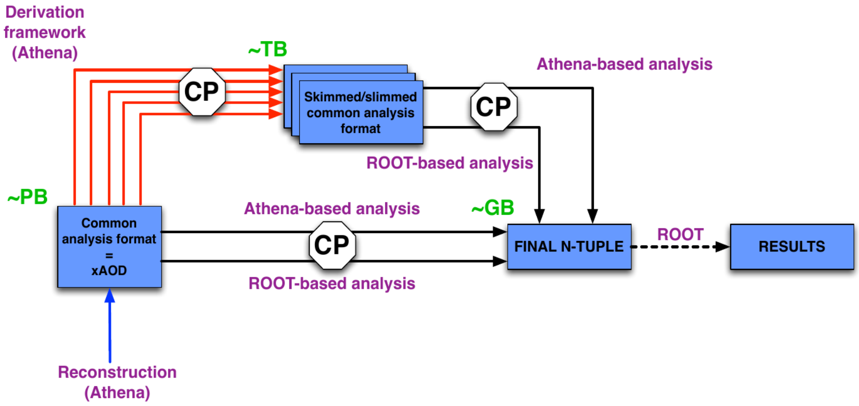


Figure 4.2: Analysis workflow for Run 2. Centrally produced xAODs can be used for analysis and also serve as input to the central derivation framework. Many analysis-specific derivation formats are centrally produced to provide more storage- and compute-efficient inputs to analyses. Analysis groups use “dual-use” CP tools that are usable in ROOT and Athena based frameworks to produce their analysis n-tuples [FAR+15].

various analysis groups. In both cases, ROOT or Athena based frameworks are making use of dual-use tools provided by the CP groups. Analysis groups often produce their own intermediate format as a *n-tuple*, storing variables in a flat structure.

The outlook section of “*Dual-use tools and systematics-aware analysis workflows in the ATLAS Run-2 analysis model*” [FAR+15] mentions that more work to improve and harmonize analysis software is expected. One expectation is the trend towards greater framework flexibility and use of dual-use components and the implementation of advanced analysis techniques in common CP tools. The possibility of dual-use algorithms is mentioned, which is an early motivation for the recently developed CP algorithms, discussed in Part III.

4.3 Derived Analysis Formats

The *derivation* framework, discussed in “*A New Petabyte-scale Data Derivation Framework for ATLAS*” [Cat+15], provides centrally produced reductions of xAODs. Analysis groups can define the content of a given derivation format by removing certain events (*skimming*), individual objects (*thinning*), an object type or part of its variables (*slimming*). These formats were introduced for Run 2 to address the problem of common and redundant preparations of reduced formats that was observed during Run 1. They typically contain all the necessary information to perform common analysis tasks.

During Run 2, the centralisation of DxAOD production was found to be very successful. Now, it is not recommended to perform analyses directly with xAODs, since many DxAOD formats are available and able to provide the necessary data at a more storage efficient way.[1]

4.4 Analysis Model in Future Runs

A increasing storage demand is expected for future Runs, which motivates a revision of the ATLAS production model. A projection of the expected disk resources is displayed in Figure 4.3. The "*Analysis Model Study Group for Run3*" (AMSG-R3) was formed to study the situation and create general recommendations [Elm+20].

The AMSG-R3 is recommending a new, very small sized, analysis format `DAOD_PHYSLITE` (~10 kB/event) and a larger format `DAOD_PHYS` (~50 kB/event), as a central starting point for most analyses. This would greatly reduce the number of existing derivation formats, and all analysis groups are requested to start their analyses with these formats.

Both derivations have a reduced amount of particle track, trigger and Monte Carlo generator information. Physics objects will already be calibrated in the `DAOD_PHYSLITE` format, which is not the case in current DxAODs. Additionally, lossy compression of floating-point information, with more significant digits than the physical resolution and uncertainties is recommended.

[1] For the ATLAS authorship qualification, I studied possible optimisations of the configuration of xAODs and DxAODs. The results of this study are presented in the public note "*Impact of ROOT file parameters on ATLAS Analysis Object Data*" [Col20].

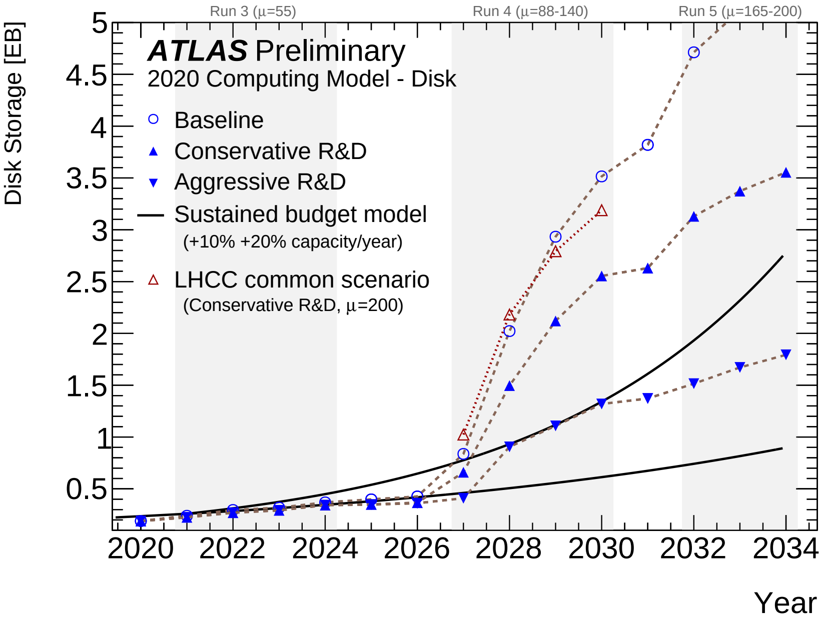


Figure 4.3: Projected disk requirements for ATLAS during future Runs. The baseline model refers to a similar operation as in Run 2. *Conservative R&D* models a scenario with some improvements, e.g. the addition of lossy compression. More *aggressive R&D*, e.g. a wide adoption of DAOD_PHYSLIGHT as the main analysis format, leads to the largest reduction in resource requirements [Cal+20].

Part II
RPV1L Analysis

As already motivated in Chapter 1, the Standard Model is not expected to be the last theory of particle physics. Supersymmetry (SUSY) models are one popular class of theories that are able to address many issues of the Standard Model, and many ATLAS analyses search for evidence of Supersymmetry. A summary of Supersymmetry theory is given in Chapter 5.

The *RPV1L* analysis, described in Chapter 6, is a search for R-parity violating SUSY processes. The final state in this analysis consists of one or more leptons or exactly two same sign leptons, with 4 to 15+ – possibly b-tagged – jets. The analysis considers the yields of multiple signal models in the two-dimensional N_{jet} by N_{bjet} distribution. All major backgrounds are estimated with data-driven methods. Chapter 7 contains the discussion of the background estimate of misidentified leptons from QCD multi-jet events. In Chapter 8, a short summary of the expected analysis results is presented.

The RPV1L analysis serves as a benchmark to validate the novel CP algorithms in Part III. The two same-sign lepton channel and various machine learning studies of the RPV1L analysis are not discussed in this dissertation to focus on the QCD multi-jet background and the CP algorithm validation.

5 Supersymmetry

David Griffiths is summarizing the characteristics of *Supersymmetry* (SUSY)[1] in his "*Introduction to elementary particles*" [Gri87, pp. 411-413] as a new fundamental symmetry that relates bosons and fermions. Each SM fermion has a bosonic *superpartner*, named by prefixing its name with an "s-", e.g. sup, sdown, selectron, et cetera. Correspondingly, SM bosons have fermionic superpartners, which are named by adding the postfix "-ino", e.g. gluino, photino, wino, et cetera.

By introducing new particles, the energy dependence of the three running coupling constants changes such that they can be unified on the *grand unified theory* (GUT) scale. SUSY is also able to give a "natural" solution to the hierarchy problem since boson and fermion contributions cancel each other in the Higgs loop corrections. In some SUSY models, the *lightest supersymmetric particle* (LSP) is colourless, electrically neutral, and stable and therefore a promising candidate for dark matter. Additionally, there are ways to integrate gravity into SUSY models. On the other hand, Supersymmetry models rely on many more free parameters than the SM itself.

If Supersymmetry were not broken, all postulated superpartners would have the same mass as their SM partner. No particles like this were observed in experiments. For example, no selectron with $spin = 0$ and $m_e = 0.511\ \mathrm{MeV/c^2}$ has been seen. Robert Mann mentions another convincing argument against unbroken Supersymmetry. If selectrons behaved exactly like electrons with $spin = 0$, they would bind to atoms without being subject to the Pauli principle, causing normal matter to implode [Man10, p. 513]. Therefore, Supersymmetry must be broken. It is assumed that superpartners are much heavier than their SM partners. The scale at which Supersymmetry is broken affects how well the model can address the hierarchy problem and if the coupling constants can be unified [Man10, pp. 513-514].

There is no experimental evidence supporting SUSY at this time.

[1] First development in a series of publications [GL71]; [GS71]; [VA73]; [WZ74b]; [WZ74a]; [FZ74]; [SS74].

Table 5.1: The fields of the MSSM and their $SU(3)\times SU(2)\times U(1)$ quantum numbers. Only one generation of quarks and leptons is listed. For each lepton, quark, and Higgs supermultiplet, there is a corresponding anti-particle multiplet of charge-conjugated fermions and their associated scalar partners [Tan+18, p. 790].

Supermultiplets	Superfield	Bosonic fields	Fermionic fields	$SU(3)$	$SU(2)$	$U(1)$
gluon/gluino	$\widehat{V}_8$	g	$\widetilde{g}$	8	1	0
gauge boson/ gaugino	$\widehat{V}$	$W^{\pm}, W^0$	$\widetilde{W}^{\pm}, \widetilde{W}^0$	1	3	0
	$\widehat{V}'$	B	$\widetilde{B}$	1	1	0
slepton/lepton	$\widehat{L}$	$(\widetilde{\nu}_L, \widetilde{e}_L^-)$	$(\nu, e^-)_L$	1	2	-1
	$\widehat{E}^c$	$\widetilde{e}_R^+$	e_L^c	1	1	2
squark/quark	$\widehat{Q}$	$\left(\widetilde{u}_L, \widetilde{d}_L\right)$	$(u, d)_L$	3	2	$1/3$
	$\widehat{U}^c$	$\widetilde{u}_R^*$	u_L^c	$\overline{3}$	1	$-4/3$
	$\widehat{D}^c$	$\widetilde{d}_R^*$	d_L^c	$\overline{3}$	1	$2/3$
Higgs/higgsino	$\widehat{H}_d$	$\left(H_d^0, H_d^-\right)$	$\left(\widetilde{H}_d^0, \widetilde{H}_d^-\right)$	1	2	-1
	$\widehat{H}_u$	(H_u^+, H_u^0)	$\left(\widetilde{H}_u^+, \widetilde{H}_u^0\right)$	1	2	1

5.1 Minimal Supersymmetric Standard Model

A commonly studied supersymmetric model is the *Minimal Supersymmetric Standard Model* (MSSM)[2]. To give an overview of the particles and characteristics of the MSSM, *"A Supersymmetry primer"* by Martin [Mar10, pp. 3-13, 54-72 and 102-107] and *"Review of Particle Physics"* by Tanabashi et al. [Tan+18, pp. 790-793] are followed.

In the supersymmetric extension of the Standard Model, each particle has a superpartner with a spin differing by $\frac{1}{2}$. Both superpartners form a *supermultiplet*. The simplest supermultiplet is the *chiral*, or *matter*, supermultiplet, consisting of a single Weyl fermion[3] combined with a complex scalar field. The *gauge*, or *vector*, supermultiplet consists of a massless spin-1 vector boson (2 helicity states) and a Weyl fermion. After spontaneous symmetry breaking, the components of gauge supermultiplets can gain mass. Other particle representations can be reduced to combinations of chiral and gauge supermultiplets.

Table 5.1 shows the supermultiplets of the MSSM. The gluon/gluino and gauge/-gaugino pairs are gauge supermultiplets. The matter supermultiplets consist of three generations of left-handed fermions, their superpartners, and all of their antiparticles.

[2] Originally developed by Fayet [Fay76]; [Fay77].

[3] Massless spin-$\frac{1}{2}$ fermion with two helicity states.

Left- and right-handed quarks and leptons behave as separate entities with different gauge transformation properties. For example, left-handed squarks, e.g. $\widetilde{u}_L$ and $\widetilde{d}_L$, couple to the W boson, while $\widetilde{u}_R$ and $\widetilde{d}_R$ do not. The "handedness" of bosonic superpartners (e.g. $\widetilde{e}_L$ and $\widetilde{e}_R$) refers to the helicity of their SM partner.

In the MSSM, there are two complex Higgs doublets and their higgsino superpartners, H_u and H_d, that couple either to "up"-type or "down-type" quarks or charged leptons.

Superpartners listed in Table 5.1 are not necessarily the mass eigenstates of the MSSM model. After electroweak symmetry breaking, the W^0 and B^0 gauge eigenstate mix to Z^0 and γ. With SUSY breaking, there can be mixing between electroweak gauginos and higgsinos, and within various sets of squarks, sleptons and Higgs scalars with the same electric charge.

Gaugino mass eigenstates after electroweak and Supersymmetry breaking are called *neutralino* if they are electrically neutral, and *chargino* if charged. There are four different neutralinos, $\widetilde{\chi}_i^0$ $(i = 1, 2, 3, 4)$, sorted ascendingly by mass. There are also two positive and two negative charginos, $\widetilde{\chi}_i^\pm$ $(i = 1, 2)$. The lightest neutralino, $\widetilde{\chi}_1^0$, is typically assumed to be the LSP.

The gluino is a colour octet fermion, and is often assumed to be heavier than neutralinos and charginos.

The masses of SUSY particles can not be arbitrarily large if the hierarchy problem should still be addressed by a Supersymmetry breaking SUSY model. This type of limited symmetry breaking is called *soft Supersymmetry* breaking. The MSSM requires an LSP mass that is not much larger than the TeV scale to address the hierarchy problem without fine-tuning loop corrections to the Higgs mass, but there is no fundamental requirement on the LSP mass. The MSSM predicts at least one neutral Higgs with the mass $m_H < 135$ GeV/c^2 [Mar10, p. 12].

5.2 R-Parity

As described by Martin in *"A Supersymmetry primer"* [Mar10, pp. 58-60 and 134-136], R-parity[4] is a common concept considered in the context of SUSY models. It is defined for every particle by the expression $P_R = (-1)^{3(B-L)+2s}$, with baryon number B, lepton number L and spin s.

[4] First mentioned by Farrar and Fayet [FF78].

The lagrangian of the MSSM is constructed with the superpotential

$$W_{MSSM} = \bar{u}\mathbf{y_u}QH_u - \bar{d}\mathbf{y_d}QH_d - \bar{e}\mathbf{y_e}LH_d + \mu H_u H_d, \tag{5.1}$$

which could be extended by R-parity violating terms. If R-parity conservation is assumed, terms of the form

$$W_{\Delta L=1} = \frac{1}{2}\lambda_{ijk}L_iL_j\overline{e}_k + \lambda'_{ijk}L_iQ_j\overline{d}_k + \mu'_iL_iH_u \tag{5.2}$$

and

$$W_{\Delta B=1} = \frac{1}{2}\lambda''_{ijk}\overline{u}_i\overline{d}_j\overline{d}_k \tag{5.3}$$

with family indices $i = 1, 2, 3$, are forbidden. The terms in equation 5.2 violate the total lepton number by 1, equivalently for equation 5.3 with the baryon number.

Corresponding B- and L-violation processes have not been observed, and there are experimental constraints on the contribution of these terms. The non-observation of proton decays, for example, gives one of the strongest limit. If λ' and λ'' were present and unsuppressed, the lifetime of a proton would be extremely short [Bar+05].

In the MSSM, *R-parity* is introduced as a new symmetry, which eliminates the possibility of B and L violating terms. The symmetry principle states that a candidate term is only allowed if the product of P_R is +1. Particles within the same supermultiplet have opposing R-parities. All SM particles and the Higgs bosons have $P_R = +1$, and all squarks, sleptons, gauginos and higgsinos have $P_R = -1$.

No mixing between sparticles and $P_R = +1$ particles is possible if R-parity is conserved. *R-parity conservation* (RPC) implies that every interaction vertex in the theory contains an even number of $P_R = -1$ sparticles, which results in multiple consequences:

1. The LSP with $P_R = -1$ must be stable. If it is electrically neutral, it only interacts weakly, it is therefore an attractive dark matter candidate.

2. Each particle other than the LSP must eventually decay into a state that contains an odd number of LSPs.

3. In colliders, sparticles can only occur in even numbers.

However, RPC is not a requirement for the theoretical consistency of the MSSM.

If *R-parity violation* (RPV) is allowed, both RPV violating terms can occur, but some theories only consider one of the violating terms. Minimum flavour violation is an example for only naturally occurring B-violating terms [CGH12], i.e. $\lambda'' \neq 0$, with λ''_{tsb}

as the largest RPV coupling. Other theories only have naturally occurring L-violating terms [LM20]. With RPV, the searches for SUSY in collider experiments look very different. The additional couplings allow for single sparticle production and imply an unstable LSP. For example $q\overline{q} \to \widetilde{\nu}$ or $slepton$ are mediated by λ'. The single squark production $qq \to \widetilde{q}$ is mediated by λ''.

However, if the RPV couplings are sufficiently small, the LSP will result in a displaced vertex or even decay outside of the detectors, where the latter case is not distinguishable to the RPC case.

5.3 Search for Supersymmetry at the LHC

The search for Supersymmetry is an important motivation for the LHC experiments. During Run 1, searches for new particles in ATLAS and CMS were focussed on direct productions of new particles close to the electroweak scale, i.e. below and around 1 TeV/c^2 [MP14]. This was motivated by many BSM theories, including the MSSM.

Searches continue for higher mass regions and processes with long-lived particles due to R-parity violation. Figure 5.1 shows a recent summary plot of exclusion limits from the Supersymmetry searches in ATLAS. Many mass limits exclude particles in the sub-1 TeV/c^2 range, with some limits going above 2 TeV/c^2.

There is no evidence of Supersymmetry so far.

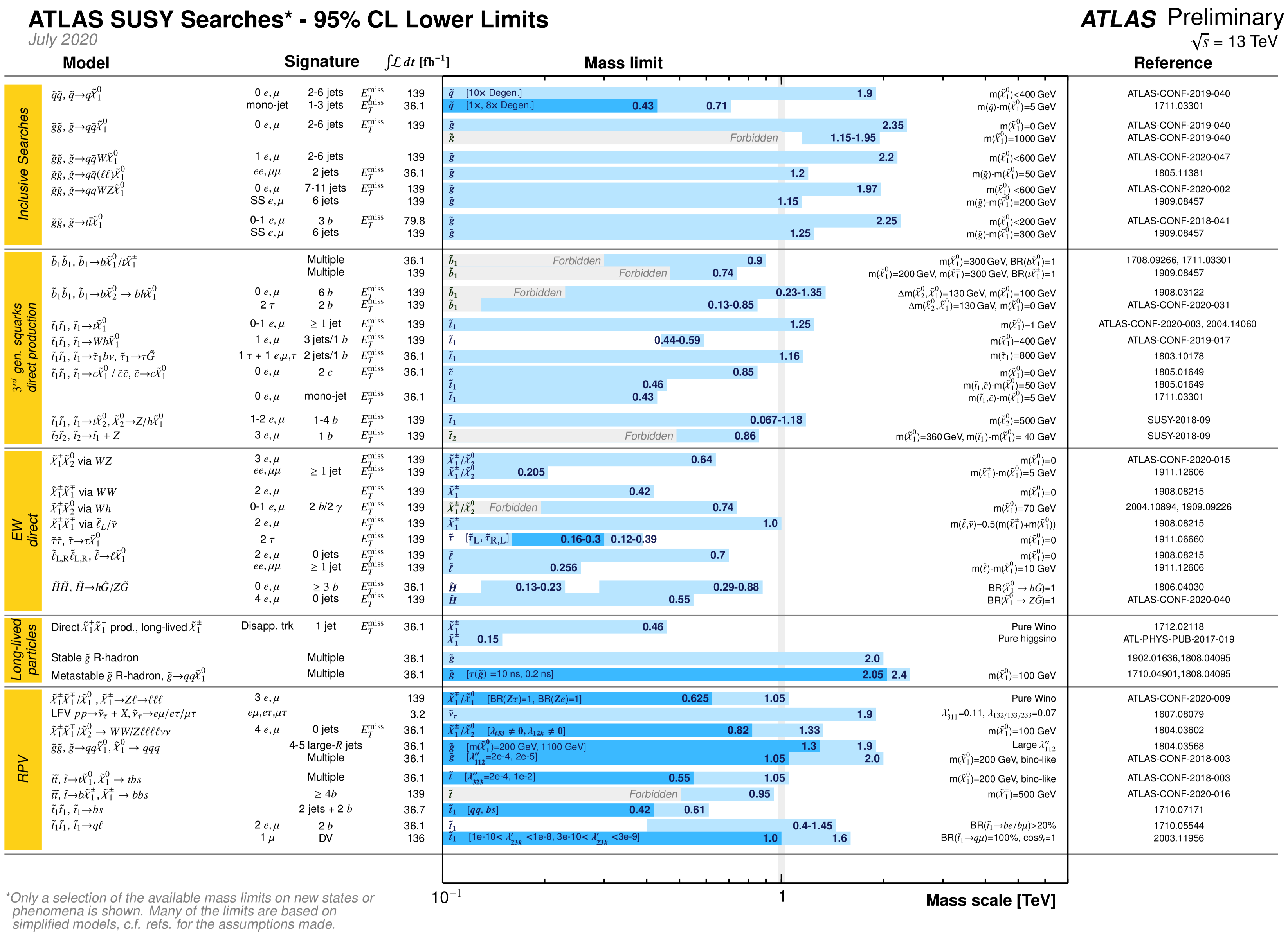


Figure 5.1: Exclusion limits of many ATLAS searches for supersymmetric particles. Some limits depend on additional assumptions on the mass of the intermediate states. This is in some cases indicated by darker bands showing different model parameters [20].

6 Analysis Overview

The *RPV1L* analysis searches for Supersymmetry using the full Run 2 dataset of the ATLAS detector [Col21]. This analysis is an extension of a previous analysis based on the data recorded in 2015 and 2016 [Aab+17c].

By analysing the final state of one or more leptons and 4 to 15+ – possibly b-tagged – jets, this analysis is covering a phase space that is not covered by other SUSY searches in ATLAS. This is mainly due to having no requirement on E_T^{miss} or derived variables, such as m_T, for background suppression.

The signal models are natural SUSY models with minimum flavour violation, where only one of the coupling constants in the RPV terms (see eq. 5.2 and 5.3) is considered to be non-zero [CGH12]. With MFV, the coupling λ''_{323} is orders of magnitude larger than other decays and therefore considered to produce the dominant process. In the previous iteration of the analysis, signals with direct gluino and stop production were targeted. The goal of this analysis is to also reach sensitivity for direct higgsino production. An overview of the benchmark signal models is given in Section 6.1. The details of the object definitions are discussed in Section 6.2.

The main measurement of this analysis is the distribution of *number of jets* (N_{Jet}) and the *number of b-tagged jets* (N_{bjet}), without relying on any additional variables. Data-driven methods are used to estimate the main backgrounds, as described in Section 6.3. Minor backgrounds are taken from Monte Carlo simulations. Finally, a simultaneous fit of all background models is performed on the two dimensional distribution of N_{Jet} and N_{bjet}.

6.1 Signals

Four benchmark models are considered, of which three only have baryon-number violating UDD couplings $\neq 0$ (cf. equation 5.3). One benchmark model has a lepton-number violating coupling $\lambda' \neq 0$ (cf. equation 5.2).

Compared to the previous publication, new signal models are introduced to to search for direct higgsino production. The 0-bjet region is sensitive to models without 3rd generation quarks, and the high-bjet region is sensitive to models where 3rd generation squarks (stop, sbottom) are the lightest squarks. In models where the lightest neutralinos or charginos are higgsino-like, the large top Yukawa coupling also enhances top quark production.

The benchmark signal models are:

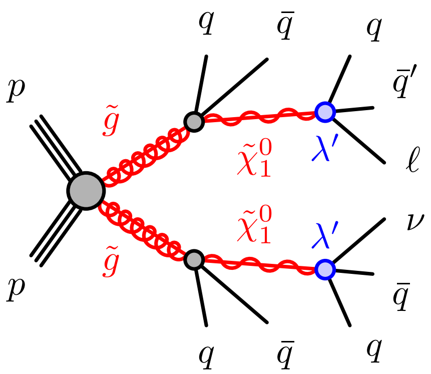


Figure 6.1: Feynman diagram for the gluino-LQD model. The non-zero RPV coupling is $\lambda'_{ijk} \neq 0$ with $i, j, k = 1, 2$.

Gluino-LQD The Gluino-LQD model is the only one with lepton-number violation, i.e. the only non-zero RPV couplings are $\lambda'_{ijk} \neq 0$ ($i, j, k = 1, 2$). The process is displayed in Figure 6.1. A direct gluino pair production decays into two light-flavour quarks and the lightest neutralino, which is also assumed to be the LSP. The LSP then decays via the non-zero RPV coupling λ' into two quarks and one lepton. No b-tagged jets are present in the final state.

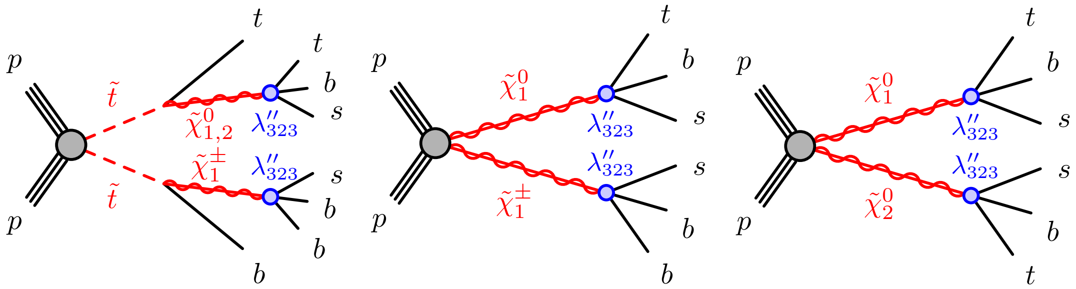


Figure 6.2: Feynman diagrams for the stop-EWKino model. The non-zero RPV coupling is $\lambda''_{323} \neq 0$.

Stop-EWKino Figure 6.2 shows the Feynman diagrams for the Stop-electroweakino (EWKino) model. A pair of two stops or two EWKinos (either neutralino or chargino) is produced. The mostly right-handed stops decay to a top and a neutralino or a bottom and a chargino. Then, the EWKinos decay via λ''_{323} to tbs or sbb. Direct EWKino production results in the same final states without the intermediate top or bottom quark. For this model, three scenarios are considered, where the LSP is either a pure bino, a pure wino or a pure higgsino, which is affecting the production cross sections of C1N1 and N1N2.

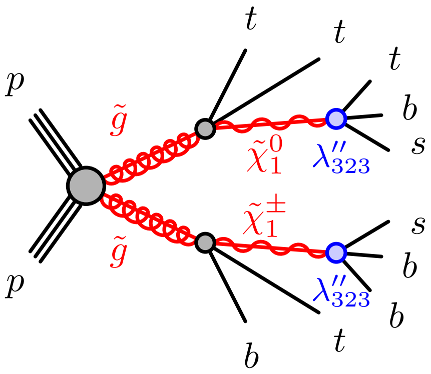


Figure 6.3: Feynman diagram for the gluino-neutralino model. The non-zero RPV coupling is $\lambda''_{323} \neq 0$. For the wino and higgsino LSP scenarios, the gluino may also decay to a chargino or second lightest neutralino.

Gluino-neutralino The gluino-neutralino model considers a pair production of gluinos that decay via a 3rd generation squark to $t\bar{t}$ and a neutralino, or $b\bar{b}$ and a neutralino, or tb and a chargino. The EWKinos then decay as described in the stop-EWKino model to tbs or sbb. Figure 6.3 shows a Feynman diagram of this process.

Gluino-stop In Figure 6.4, a Feynman diagram for the gluino-stop model is shown. A pair of gluinos is produced, and each decays to a top quark and a stop. The stop then decays via λ''_{323} to bs.

The first model is the only one where the final state has no b-tagged jets. In the final state of the other models, many (≥ 4) b-tagged jets are expected.

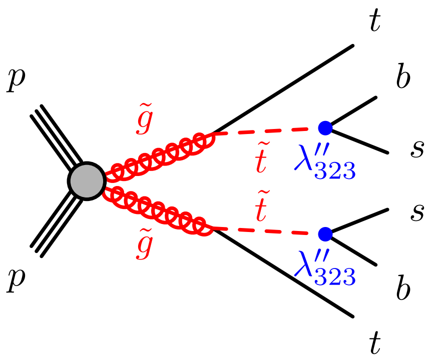


Figure 6.4: Feynman diagram for the gluino-stop model. The non-zero RPV coupling is $\lambda''_{323} \neq 0$.

6.2 Object Definition

The final state of this analysis consists of 4 to 15+ – possibly b-tagged – jets and either one or more or two same-sign leptons. A corresponding event display is shown in Figure 6.5. The objects are selected by following the recommendations of the corresponding CP groups and the SUSY group.[1] All events must pass the recommended good runs list and SCT, LAr, and Tile cleaning selections, which excludes events of bad quality due to the detector state during data taking. At least one signal lepton, either electron or muon, with $p_T > 27$ GeV/c and ≥ 4 jets with $p_T > 20$ GeV/c must be present.

Electrons are selected on a baseline level and a signal level. All requirements for both kinds of electrons are given in Table 6.1. Table 6.2 shows all triggers used to select data events with electrons. The trigger definitions differ between periods of data taking and are divided into a *standard* and a *support* trigger category. Standard triggers are used for the nominal selection, while the support triggers are necessary for the QCD fake lepton background analysis, discussed in Chapter 7. All triggers are defined as *high level triggers* (HLT), that select events based on online reconstructed physics objects. The nominal triggers are combined with a logical OR expression to form the standard lepton trigger of the analysis. They mainly select single electrons with Tight, Medium, or Loose isolation, depending on the p_T range. Support triggers have much looser requirements, necessitating them to be pre-scaled, i.e. not recording every selected event. For example, the average pre-scale factor, defined as selected luminosity/delivered luminosity, for the $5J30$ support trigger in 2016 is 0.824. The average pre-scale factor for the single electron support trigger in 2016 is 0.0056, i.e. recording much fewer events. Events

[1] See also the description of the event reconstruction in Chapter 3.

Table 6.1: Electron object definition of the RPV1L analysis.

Selection	Value
Preselection	
Algorithm	AuthorElectron
p_T	$\geq$ 10 GeV/c
η	< 1.37, > 1.52 and < 2.47
Quality	MediumLLH
Interaction point	$\left\lvert z_0^{PrimaryVertex} \right\rvert < 0.5\,\text{mm}$
Signal	
p_T	$>$ 15 GeV/c and $>$ 27 GeV/c for the leading lepton
Quality	TightLLH
Isolation	PLVTight
ECIDS	Loose (98%) (for 2 same sign leptons)
Interaction point	$\left\lvert d_0^{PrimaryVertex} \right\rvert / \sigma(d_0^{PV}) < 5$

Table 6.2: Electron triggers of the RPV1L analysis.

Year	Trigger
Standard	
2015	HLT_e24_lhmedium_L1EM20VH
	HLT_e60_lhmedium
	HLT_e120_lhloose
2016	HLT_e26_lhtight_nod0_ivarloose
-	HLT_e60_lhmedium_nod0
2018	HLT_e140_lhloose_nod0
Support (pre-scaled)	
2015	None available
2016	HLT_e24_lhmedium_nod0_L1EM20VH
	HLT_e26_lhmedium_5J30_0eta240_L1EM13VH_320
2017 &	HLT_e24_lhmedium_nod0_L1EM20VH
2018	HLT_e26_lhmedium_nod0_5J30_0eta240_L1EM13VH_320

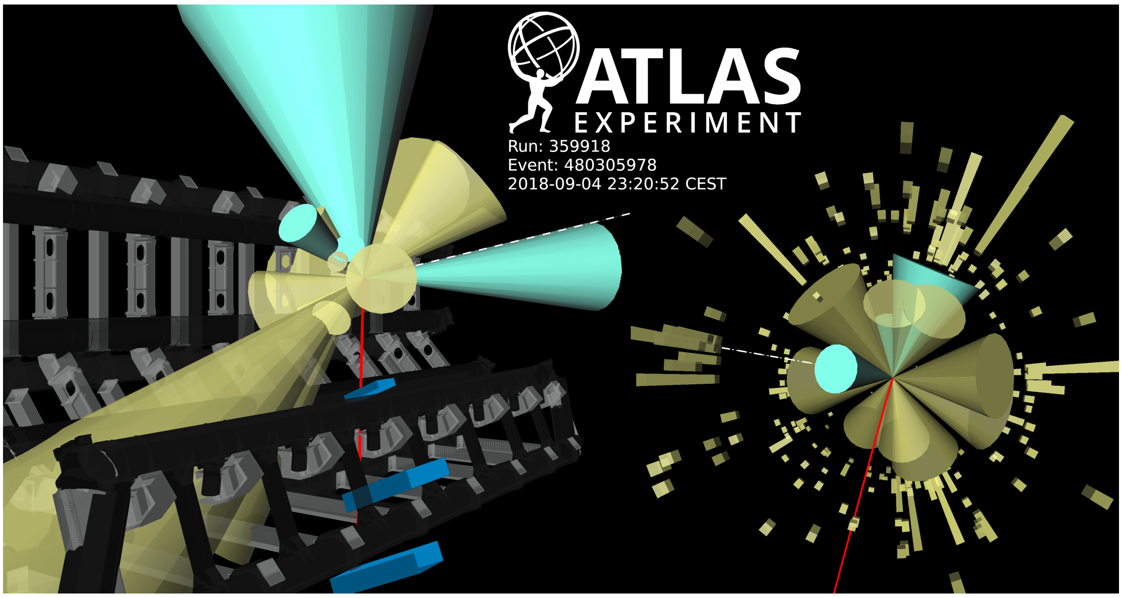


Figure 6.5: Display of an event with 15 jets with $p_T > 20$ GeV/c (yellow cones) and one muon with $p_T = 52.6$ GeV/c, shown as a red line that is crossing the blue muon chambers. Of all 15 selected jets, 4 are resulting from b-quark decays (blue cones).

with 4 and 5 jets are selected with the single lepton trigger. For events with ≥ 6 jets, a single lepton and 5 jets with $p_T \geq 30$ GeV/c trigger is used (cf. Support section in Table 6.2).

Muon selection happens similarly to the electron selection. The selection cuts and triggers are given in Table 6.3 and 6.4.

Jets are selected as described in Table 6.5. The jet vertex tagger rejects jets not originating from the primary collision vertex. This is important to distinguish jets produced in pile-up interactions from events with high jet multiplicities.

Overlap removal, i.e. resolving ambiguities between reconstructed particles, is following standard SUSY recommendations. Preselected jets are discarded if their quality is bad or if they overlap with a lepton. If b-tagged jets are close to a lepton with $p_T < 100$ GeV/c, the lepton is discarded instead. Leptons close to the remaining jets are also discarded to reject non-prompt leptons.

The missing transverse momentum, E_T^{miss}, is computed as described in Chapter 3 with baseline electrons, muons and jets. This process is following SUSY group recommendations as well.

Table 6.3: Muon definition of the RPV1L analysis.

Selection	Value
Preselection	
p_T	≥ 10 GeV/c
η	< 2.5
Quality	Medium
Interaction point	$\left\lvert z_0^{PrimaryVertex} \right\rvert < 0.5\,\mathrm{mm}$
Signal	
p_T	> 15 GeV/c and > 27 GeV/c for the leading lepton
Quality	Medium
Isolation	PLVTight
Interaction point	$\left\lvert d_0^{PrimaryVertex} \right\rvert / \sigma(d_0^{PV}) < 3$

Table 6.4: Muon triggers of the RPV1L analysis.

Year	Trigger
Standard	
2015	`HLT_mu20_iloose_L1MU15`
	`HLT_mu50`
2016 to	`HLT_mu26_ivarmedium`
2018	`HLT_mu50`
Support (pre-scaled)	
2015	`HLT_mu24`
2016 to	`HLT_mu24`
2018	`HLT_mu26_5J30_0eta240_L1MU10_3j20`

Table 6.5: Jet definition of the RPV1L analysis.

Selection	Value
Preselection	
Algorithm	Anti-k_t4PFlow
p_T	$\geq$ 20, 40, 60, 80, 100 GeV/c
Signal	
η	< 2.5
Jet vertex tagger	Tight
Signal bjet	
Algorithm	DL1r, 70% working point
p_T	≥ 20 GeV/c
η	< 2.5
Jet vertex tagger	Tight

6.3 Background Strategy

All major backgrounds are estimated in a data-driven approach, making the estimate independent of known mismodelling in simulated events with large (b-)jet multiplicities. The main backgrounds for the 1 lepton category are $W + jets$ events in the 0-bjet region and $t\bar{t} + jets$ events in all regions containing b-tagged jets. Minor backgrounds for the 1 lepton category are estimated from simulations and consist of electroweak processes, diboson, single top, ttV, ttH and 4-top events.

Simulated events and dedicated regions in data are used to validate the parameterization. The model parameters are estimated from data in a simultaneous likelihood fit in all jet and bjet regions. There is no direct dependence of this method on simulated events.

In the 0-bjet region, $W/Z + jets$ events are the most dominant background. Their propagation from one N_{jet} bin to the next is done with a combination of *staircase* [EKS85]; [Ber+89]; [Ger+12] and *poission* [Ger+12] scaling

$$r(j) = N_{j+1}^{W/Z+jets} / N_{j}^{W/Z+jets} = c_0 + \frac{c_1}{j+1}.$$

The resulting estimate is

$$N_{j,b}^{W/Z+jets} = f_{j,b}^{MC\,W/Z+jets} N_4^{W/Z+jets} \prod_{j'=4}^{j'=j-1} r(j').$$

The fraction of bjet events, $f_{j,b}^{MC\,W/Z+jets}$, is taken from simulated events. N_4^{W+jets}, N_4^{Z+jets}, c_0 and c_1 are left floating in the fit.

To estimate the $t\bar{t} + jets$ background, which is most dominant in bins with ≥ 1 bjets, a similar parameterization is used:

$$r(j) = c_0 + \frac{c_1}{j + c_2}.$$

The additional parameter c_2 is necessary to adjust for extra jets from initial or final state radiation. Three bins, $b, b-1$ and $b-2$, of the previous jet slices are used to estimate the number of bjets of the $j+1$th bin, i.e. $f_{j,b}$ from data, as shown in Figure 6.6:

$$N_{j,b}^{t\bar{t}+jets} = N_{j}^{t\bar{t}+jets} f_{j,b}$$

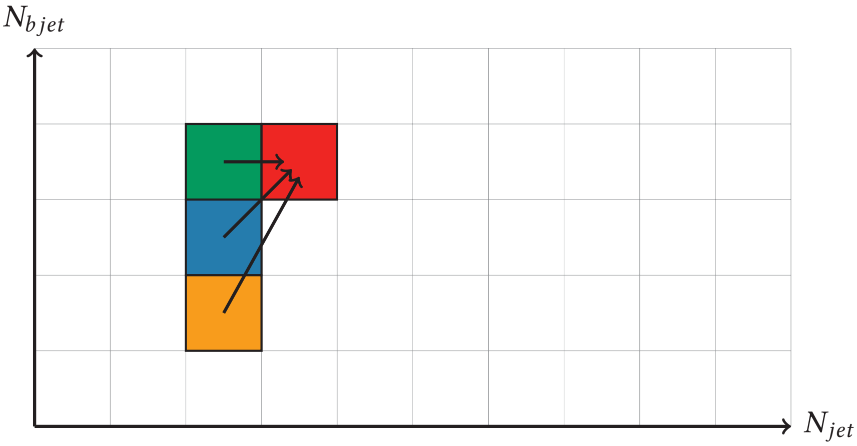


Figure 6.6: Parameterization scheme for the data-driven background estimate of $t\bar{t} + jets$. The fraction of bjet events, f, in preceding jet-slices is used to estimate the number of events in the current jet-slice.

and

$$\boxed{f_{(j+1),b}} = \boxed{f_{j,b} \cdot x_0} + \boxed{f_{j,(b-1)} \cdot x_1} + \boxed{f_{j,(b-2)} \cdot x_2}.^2 \tag{6.1}$$

With this parameterization, the estimation of arbitrarily high bjet multiplicities is possible. The initial bjet shape is taken from the 4-jet slice. This parameterization introduces 11 additional free parameters to the fit.

The final fit has a total of 26 free parameters in at least 51 bins. Therefore, the fit is over-constrained. A good description of the backgrounds is achieved over the whole jet multiplicity range (see Figure 6.7).

Misidentified leptons from QCD multi-jet events constitute a minor but not negligible background in the 1 lepton category. A good estimate is especially important for a stable fit of the $W + jets$ background model in the 0-bjet region. This is why a detailed study of this background is done, as discussed in Chapter 7.

Other background strategies of the RPV1L analysis, especially for the two same-sign lepton channel, are not addressed here for the sake of conciseness.

[2] Colors matching to boxes in Figure 6.6.

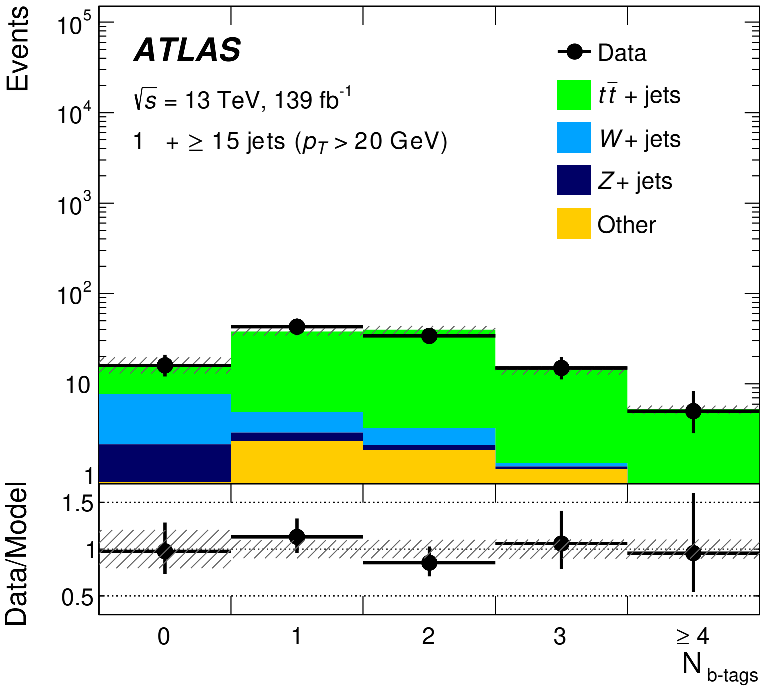


Figure 6.7: Event yield and the corresponding background estimates for the inclusive 15+ jet slice [Col21].

7 Background from Misidentified Leptons in QCD Multi-jet Events

Misidentified (fake) or non-prompt leptons in QCD multi-jet events are a minor background of the RPV1L analysis. Hadronic jets are misinterpreted as leptons and therefore change the number of leptons and jets in a given event. A good estimation of this effect is important for the gluino-LQD signal model and the $W + jets$ background fit since both fall into the 0-bjet region where this is a common occurrence. Figure 7.1 shows a comparison of data and simulated events, with a disagreement at low jet and bjet multiplicities originating from the QCD fake background. The disagreement at high multiplicities stems from known mismodelling in simulated events [Aab+17b].

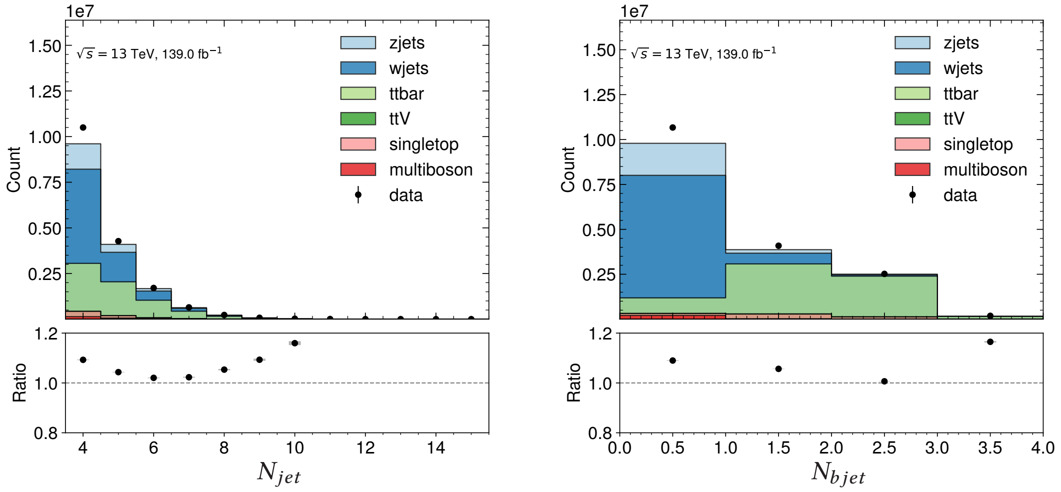
The number of fake leptons is estimated in a data-driven approach using the matrix method in the full Run 2 dataset. For the matrix method, two transfer factors ε^{misid} and ε^{real} are measured. ε^{real} is measured with the tag-and-probe method [Beh+13, pp. 331-332] in $Z \rightarrow \ell\ell$ events. ε^{misid} is measured in a QCD enriched region by subtracting all remaining simulated backgrounds from data. Both factors are parameterized as a function of leading lepton p_T and η. This approach is based on the previous iteration of this analysis [Aab+17c] with additional optimizations.

7.1 Matrix Method

The number of events with misidentified leptons that pass the signal selection, N^{misid}_{signal}, is estimated with the matrix method [Beh+13, pp. 334-337]:

$$\begin{pmatrix} N_{signal} \\ N_{baseline!signal} \end{pmatrix} = \begin{pmatrix} \varepsilon^{real} & \varepsilon^{misid} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} N^{real}_{baseline!signal} \\ N^{misid}_{baseline!signal} \end{pmatrix}. \tag{7.1}$$

Measured quantities are the number of events containing leptons passing the baseline, but not signal selection, $N_{baseline!signal}$, the number of events with leptons passing the signal selection, N_{signal}, and the transfer factors ε^{real} and ε^{misid}.



(a) **(b)**

Figure 7.1: N_{jet} and N_{bjet} for 2015 – 2018 electrons. The disagreement between data and simulated background events at low jet- and bjet-multiplicities is due to a missing description of misidentified leptons.

The transfer factors are defined as

$$\varepsilon = \frac{N_{signal}}{N_{baseline!signal}}$$

and give an estimate of how many misidentified (or true) leptons pass the signal selection.

To measure the factor of true leptons, ε^{real}, the tag-and-probe method is applied. In 2015 – 2018 data events, all events with $\geq$ 2 baseline leptons of same flavour and opposite charge are selected. These events are required to pass the nominal trigger. Additionally, one lepton has to pass the signal selection and is used as the tag. All combinations of the tag and the remaining baseline leptons are considered. The lepton pair with its invariant mass $m(\ell\ell)$ closest to the Z-mass m_Z:

$$|m(\ell\ell) - 91187.6\,MeV/c^2| \leq 10000\,MeV/c^2\,,$$

is selected as the tag-and-probe pair. The second lepton is very likely to be a true lepton, since it is produced in a $Z \longrightarrow \ell\ell$ decay. The transfer factor is estimated by counting the number of probe leptons that pass or do not pass the signal requirements

$$\varepsilon^{real} = \frac{N^{Probe}_{signal}}{N^{Probe}_{baseline!signal}}.$$

This factor is separately measured for electrons and muons.

The transfer factor ε^{misid} is measured by subtracting simulated background events from data in a QCD enriched region, interpreting the remaining events as QCD fake events:

$$\varepsilon^{misid} = \frac{(N^{data} - N^{MC})|_{signal}}{(N^{data} - N^{MC})|_{baseline!signal}}.$$

Events have to pass the support triggers with at least one baseline lepton (cf. Section 6.2). Events with 4 + 5 jets and 6+ jets are treated differently, because of the less pre-scaled support trigger that is available for events with many jets.

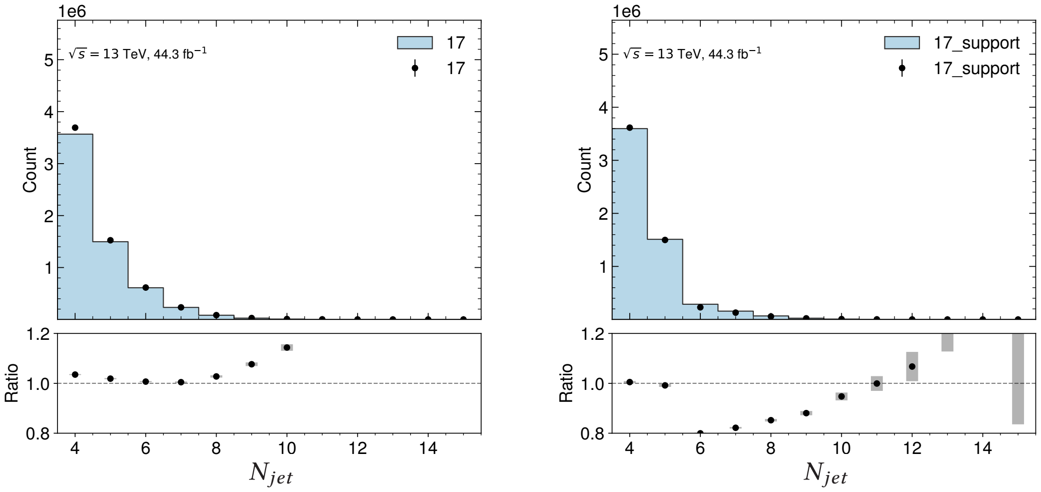
Both transfer factors are extracted as a function of leading lepton p_T and η, in the whole 2015 to 2018 period. An event-level weight, representing the contributions to the misidentified lepton background, is derived. Equation 7.1 is expressing the number of signal or baseline-and-not-signal events in terms of the number of real and misidentified lepton events that do not pass the signal requirement, so the equation needs to be inverted:

$$\begin{pmatrix} N^{real}_{baseline!signal} \\ N^{misid}_{baseline!signal} \end{pmatrix} = \frac{1}{\varepsilon^{real} - \varepsilon^{misid}} \begin{pmatrix} 1 & -\varepsilon^{misid} \\ -1 & \varepsilon^{real} \end{pmatrix} \begin{pmatrix} N_{signal} \\ N_{baseline!signal} \end{pmatrix}$$

$$\Rightarrow N^{misid}_{baseline!signal} = \frac{1}{\varepsilon^{real} - \varepsilon^{misid}} \left(-N_{signal} + \varepsilon^{real} N_{baseline!signal} \right). \tag{7.2}$$

Since this is expressing $N^{misid}_{baseline!signal}$ and not N^{misid}_{signal}, equation 7.2 needs to be multiplied with ε^{misid} again:

$$\begin{aligned} N^{misid}_{signal} &= \varepsilon^{misid} N^{misid}_{baseline!signal} & (7.3) \\ &= \frac{\varepsilon^{misid}}{\varepsilon^{misid} - \varepsilon^{real}} \left(N_{signal} - \varepsilon^{real} N_{baseline!signal} \right) & (7.4) \\ &= \sum_{baseline\ events} \frac{\varepsilon^{misid}}{\varepsilon^{misid} - \varepsilon^{real}} \left(\delta_{signal} - \varepsilon^{real} \delta_{baseline!signal} \right). & (7.5) \end{aligned}$$



(a) **(b)**

Figure 7.2: N_{jet} distribution for 2017 muon events, selected with the nominal (7.2a) and the support triggers (7.2b). Selecting events with jet $p_T \geq 20$ GeV/c results in problems with the support triggers used for 6+ jet events since it only becomes efficient at 30 GeV/c.

Equation 7.5 is the final expression to assign a QCD fake weight to each event. δ_{signal} is evaluated as 1, if the event has a signal lepton, otherwise $\delta_{baseline!signal} = 1$.

7.2 Trigger Selection and Efficiency Correction

The nominal triggers, as described in Section 6.2, have too tight isolation requirements for the ε^{misid} estimation since events with only one baseline, but no signal lepton, are not selected. Two separate support triggers are used:

1. Events with 4 + 5 jets: Pre-scaled single lepton trigger
2. Events with 6+ jets: Pre-scaled single lepton + 5 jets with $p_T > 30$ GeV/c

The corresponding pre-scale factors are included, if support triggered events are compared to nominally triggered events, or if simulated events are compared to data. Figure 7.2 shows that using the support trigger for 6+ jets events results in a selection bias.

The object definition of jets includes the condition $p_T \geq 20$ GeV/c, which is below the efficiency plateau of the secondary support triggers, as illustrated in Figure 7.3.

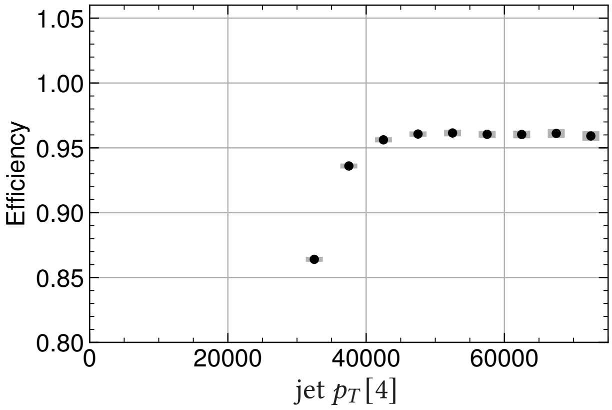


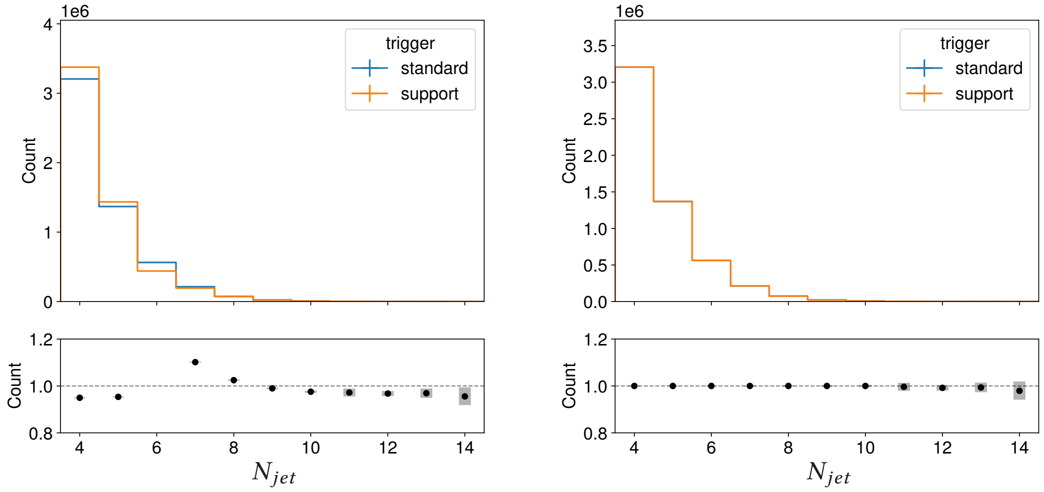
Figure 7.3: Trigger efficiency of the electron + 5jets trigger as a function of the fifth jet p_T. The trigger is only efficiently selecting jets with p_T > 40 GeV/c.

To reduce this effect, the single lepton trigger is also used for the 5 jet events, where this issue is most prominent.

This inefficiency can be corrected by comparing the N_{jet} distribution of nominally triggered events to the distribution of support triggered events (see Figure 7.4a), after a signal lepton selection. The ratio of nominal vs. support triggered events is applied as a correction factor to each event with 6+ jets, that is selected via the support trigger. A separate factor must be extracted for each period (2015, 2016, 2017, 2018) and data/simulated events. For all bins above 10 jets, the 10 jet correction factor is used. The corrected N_{jet} distribution for 2017 electrons is shown in Figure 7.4b.

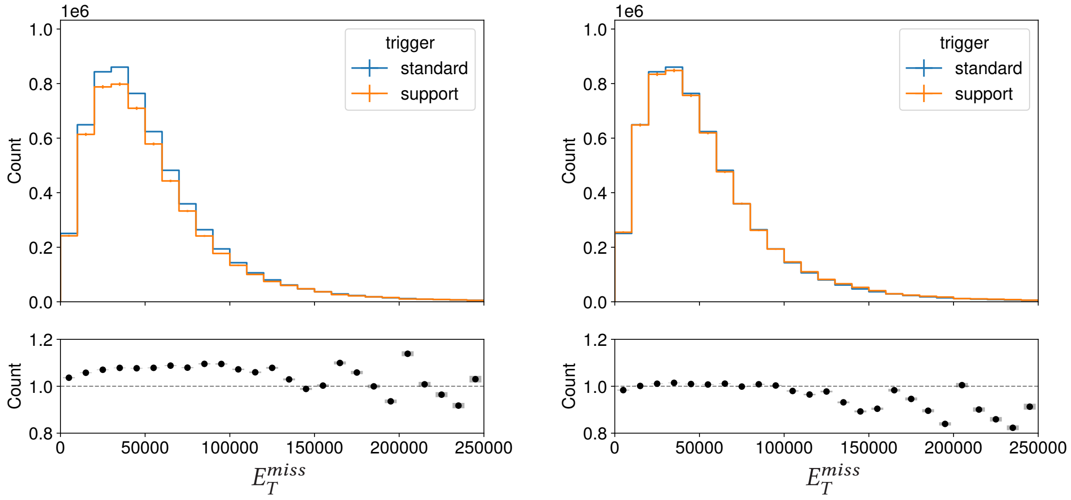
Figure 7.5 shows the E_T^{miss} distribution for 2017 electrons in data before (7.5a) and after (7.5b) the correction. With the correction, the agreement between nominal and support triggered events is much better in the region of $E_T^{miss} < 100$ GeV/c, which is the relevant region for the QCD fake background estimate.

The pre-scale and trigger correction factors must be applied to all support triggered data and simulated events.



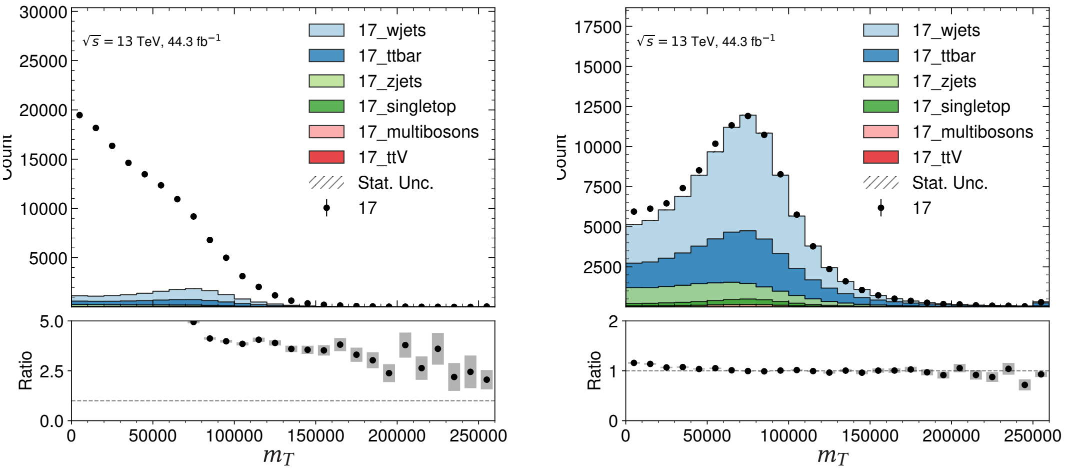
(a) **(b)**

Figure 7.4: Number of jet distributions for 2017 simulated events selecting electrons. The histograms show the distribution for nominally triggered and support triggered events before (7.4a) and after correction (7.4b).



(a) **(b)**

Figure 7.5: E_T^{miss} for 2017 data events selecting electrons with the nominal and support triggers. 7.5a shows the histogram before correcting the support trigger. 7.5b shows the distribution after correction.



(a)

(b)

Figure 7.6: Transverse mass in 2017 muon baseline-but-not-signal (7.6a) and signal (7.6b) selections. Due to calibrations of the simulated events, the data/mc agreement is much better for the signal selection.

7.3 m_T Normalization

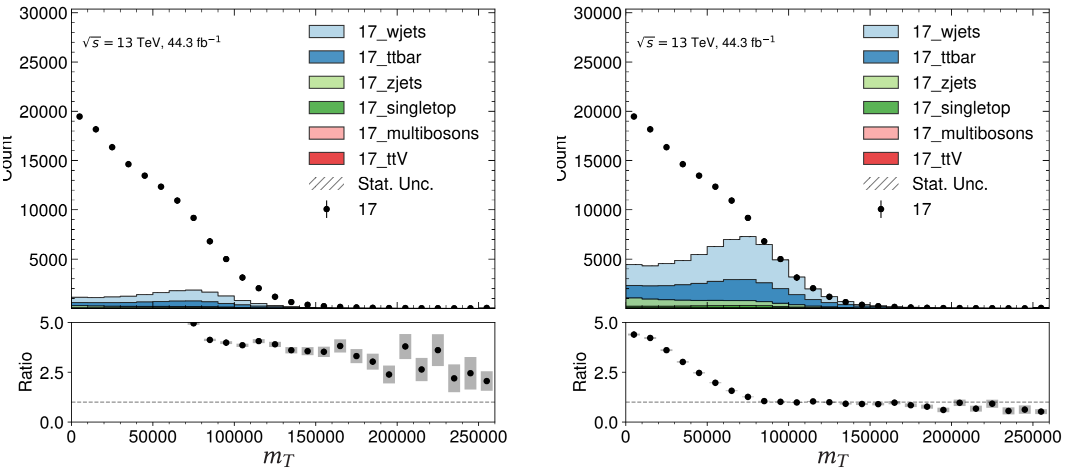
The transverse mass m_T is defined in terms of E_T^{miss} and the transversal momentum of the leading lepton p_T^ℓ [MP14]:

$$m_T = \sqrt{2 p_T^\ell E_T^{miss} \left(1 - \cos \Delta\phi \left(\ell, E_T^{miss}\right)\right)}. \tag{7.6}$$

Since there is no calibration available for simulated baseline-but-not-signal events, the m_T distribution in simulation is not matching well to data. Figure 7.6 shows data vs. simulation comparison of the m_T distribution for a baseline-but-not-signal (7.6a) and a signal selection (7.6b). A better agreement in the tail region of the baseline-but-not-signal case is expected, since all relevant backgrounds for that region are represented in simulation. The missing QCD fake background is only expected to impact the lower region of m_T, which is the visible gap in Figure 7.6b below ~50 GeV/c.

To correct for the missing calibration in simulated baseline-but-not-signal events, a calibration factor is extracted by fitting the simulation events to data in the region of [90, 150] GeV/c. The region below 90 GeV/c is excluded because of the expected missing QCD fakes that are to be estimated. The region above 150 GeV/c is excluded because of known mismodelling of the simulated events in the tail region. Different ranges were tested and found to lead to variations well below assigned uncertainties.

A separate normalization factor is extracted for 2015 − 2018, electrons and muons, 4 + 5 jets and 6+ jets, as well as baseline-but-not-signal and signal leptons. The largest factors occur for baseline-but-not-signal muon events. The normalization factor for signal events is always close to 1 (within 4%), but is also applied for consistency. This normalization is only applied to simulated events in the estimation of ε^{misid} and does not affect data. Figure 7.7 shows the m_T distribution of 2017 baseline-but-not-signal muon events before (7.7a) and after (7.7b) correction.



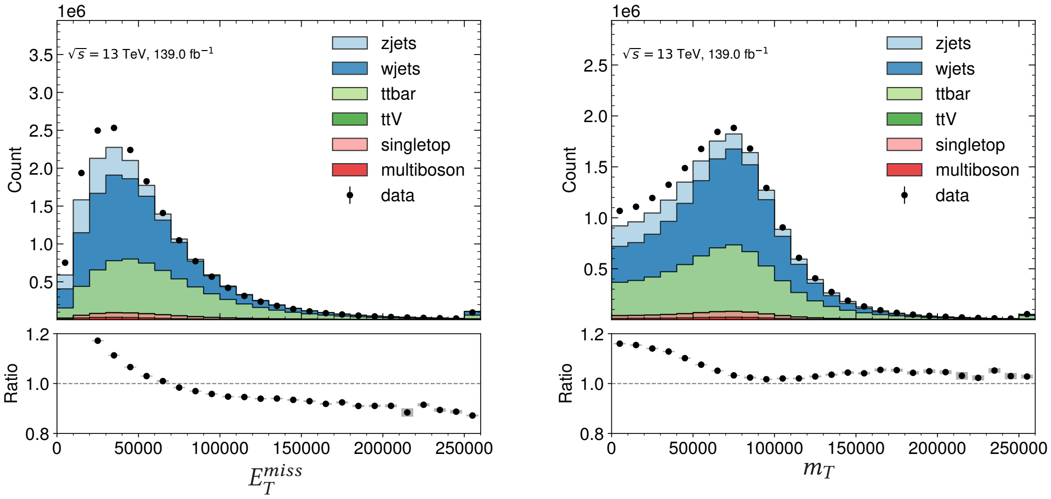
(a)

(b)

Figure 7.7: 2017 baseline-but-not-signal selected muon events without (7.7a) and with (7.7b) the normalization in m_T between [90, 150] GeV/c.

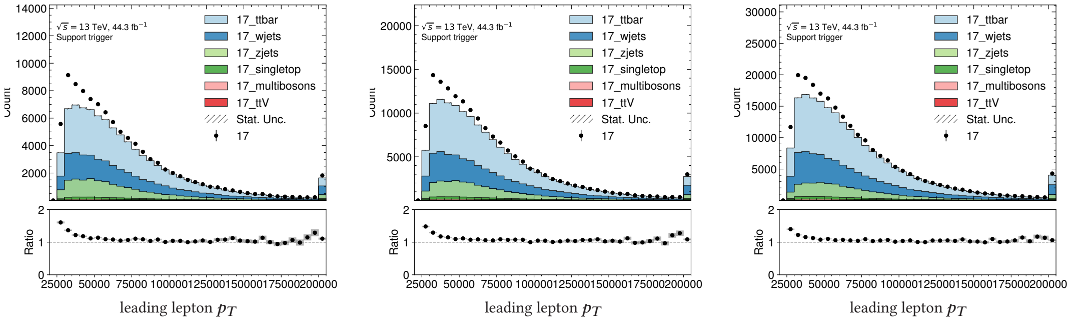
7.4 QCD Enhancing Cuts

To enhance the ratio of misidentified leptons from QCD events compared to other backgrounds before estimating ε^{misid}, QCD enhancing cuts are applied. As can be seen in Figure 7.8, the contribution of QCD fakes is mostly present in low E_T^{miss} and m_T regions. Therefore, an additional requirement on $E_T^{miss} + m_T < X$ GeV/c is introduced. Studies were done to quantify the impact of different cuts in the search for a beneficial trade-off between the total number of events and QCD enrichment. Figure 7.9 shows the leading lepton momentum for the three different cuts $E_T^{miss} + m_T < 70, 90, 110$ GeV/c. The gap between data and simulated events, i.e. the amount of QCD fakes, is largest for a cut at 70 GeV/c at a ≈ 50% lower yield, compared to a cut of 110 GeV/c. The final requirement for the estimation of ε^{misid} is selected as $E_T^{miss} + m_T < 90$ GeV/c.



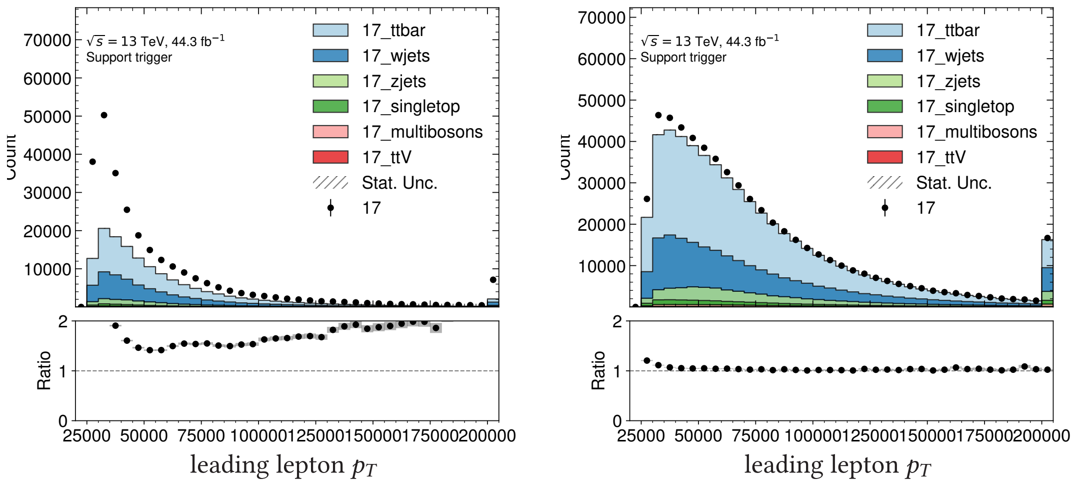
(a) **(b)**

Figure 7.8: E_T^{miss} and m_T for 2015 – 2018 electron selected events. The disagreement between data and simulation for E_T^{miss} comes from a known mismodelling in background processes with high jet multiplicities.



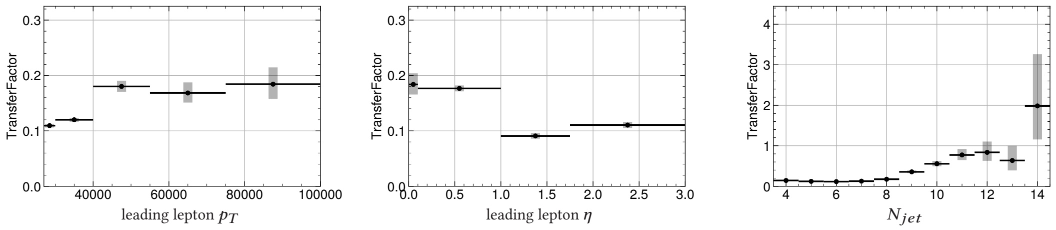
(a) **(b)** **(c)**

Figure 7.9: Leading lepton p_T for 2017 electrons with $E_T^{miss} + m_T < 70, 90, 110$ GeV/c.



(a) **(b)**

Figure 7.10: Leading lepton p_T for baseline-but-not-signal (7.10a) and signal (7.10b) electrons in 2017.



(a) **(b)** **(c)**

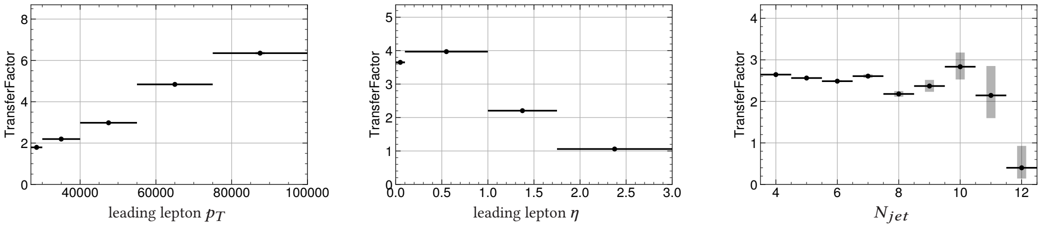
Figure 7.11: Fake transfer factor ε^{misid} for 2015 – 2018 electrons.

7.5 Estimating Fake Electrons

The number of misidentified electrons from multi-jet events is estimated by measuring ε^{real} and ε^{misid}, and applying them in the matrix method while including all mentioned corrections. Figure 7.10 shows the leading lepton p_T for the baseline-but-not-signal (7.10a) and signal (7.10b) events, which are the basis for the ratio in ε^{misid}.

ε^{misid} is measured in a selection with exactly one baseline lepton and ≥ 4 jets with $p_T > 20$ GeV/c. Figure 7.11 shows the resulting ε^{misid} for electrons from 2015 – 2018 data as a function of leading lepton p_T (7.11a), η (7.11b) and N_{jet} (7.11c). ε^{misid} can be assumed to be independent of N_{jet} for the lower jet bins.

The transfer factor ε^{real} is measured in data. Figure 7.12 shows the resulting factors as a function of leading lepton p_T (7.12a), η (7.12b) and N_{jet} (7.12c). Again, the transfer



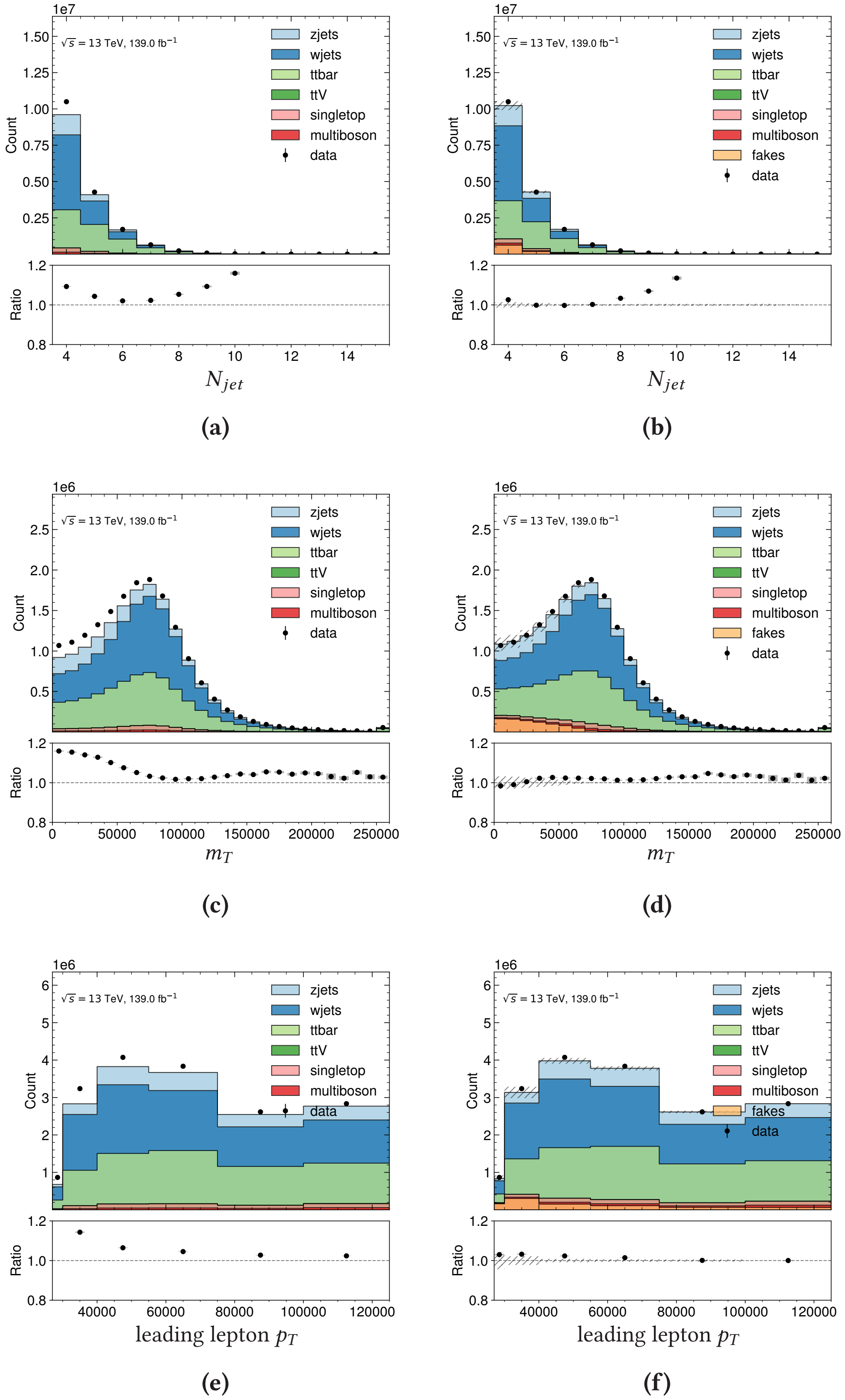
(a) **(b)** **(c)**

Figure 7.12: Real transfer factor ε^{real} for 2015 – 2018 electrons.

factor can be assumed to be sufficiently independent of N_{jet} in the lower jet bins.

Figure 7.13 shows the N_{jet}, m_T, and leading lepton p_T distributions with and without the resulting estimate of misidentified electrons.

In Figure 7.13, the statistical uncertainty is marked as a grey bar around the data points. A conservative 50% uncertainty is displayed as a hashed area, which is also used in the previous iteration of this analysis [Aab+17c]. This is a sufficient description of the uncertainty, since the final fit in the RPV1L analysis is insensitive to this uncertainty. The resulting estimate yields a good description of misidentified electrons from QCD multi-jet events.



(a)

(b)

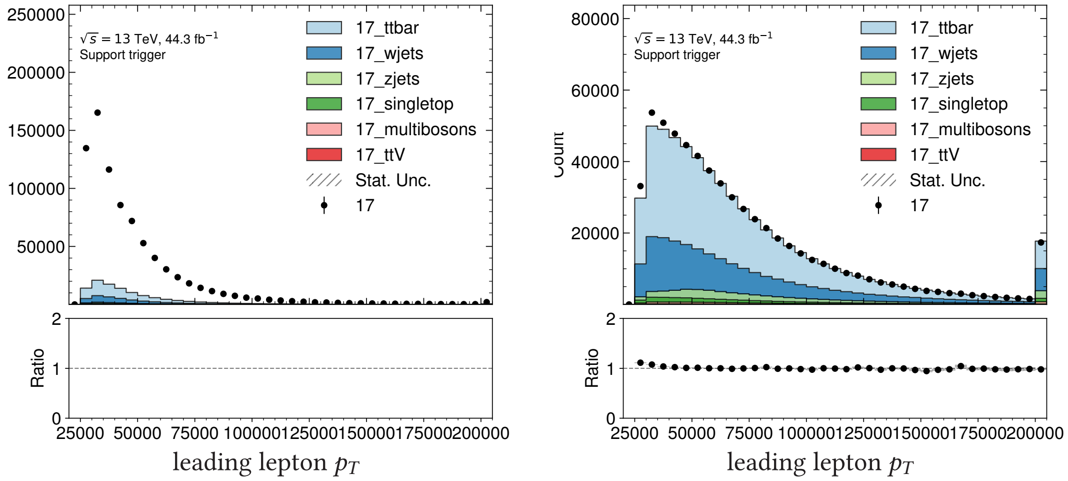
(c)

(d)

(e)

(f)

Figure 7.13: 2015 – 2018 electron events, N_{jet} without (7.13a) and with (7.13b) the fake estimate. 2015 – 2018 electron events, m_T without (7.13c) and with (7.13d) the fake estimate. 2015 – 2018 electron events, leading lepton p_T without (7.13e) and with (7.13f) the fake estimate.



(a) **(b)**

Figure 7.14: Leading lepton p_T for baseline-not-signal (7.14a) and signal (7.14b) 2017 muon events.

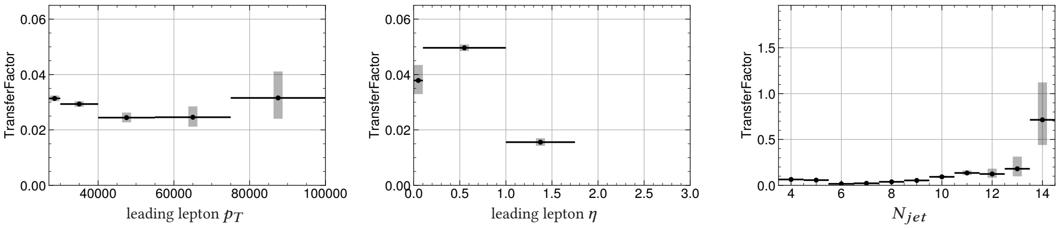
7.6 Estimating Fake Muons

Estimating the number of misidentified muons is analogous to the fake electron estimation. Figure 7.14 shows the leading lepton p_T for a baseline-but-not-signal and a signal selection. Compared to electrons (see Figure 7.10), the total number of events is larger and more QCD fakes contribute to the baseline-but-not-signal category. Fewer QCD fakes contribute to the signal category, since fake muons originate mostly from non-prompt muons of b-jet decays, opposed to electron fakes from misidentified jets and photon conversion.

Figure 7.15 shows the measured transfer factor ε^{misid} as a function of leading lepton p_T (7.15a), η (7.15b) and N_{jet} (7.15c). The factor is sufficiently independent of N_{jet}. For $|\eta| > 1.75$, the transfer factor ε^{misid} is 0.

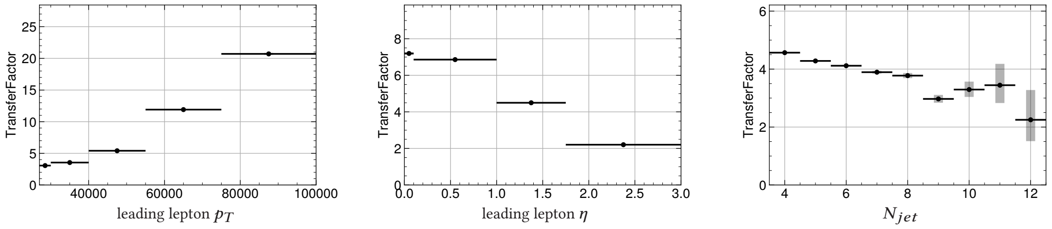
The measurements of ε^{real} are shown in Figure 7.16 as a function of leading lepton p_T (7.16a), η (7.16b) and N_{jet} (7.16c). The transfer factor behaves similar to the electrons ε^{real} (see Figure 7.12) and is sufficiently independent of N_{jet} in the lower jet bins.

The resulting yields of events with misidentified muons from QCD mulijet events is shown in Figure 7.17. A slight overestimation of all simulated backgrounds occurs with the estimate at N_{jet} = 6 and 7. Without the QCD fake estimate, the background is already matching the data, but a contribution to these bins is expected, which can be seen by comparing the same bins for electrons (see Figure 7.13b and 7.13a). In the final



(a) (b) (c)

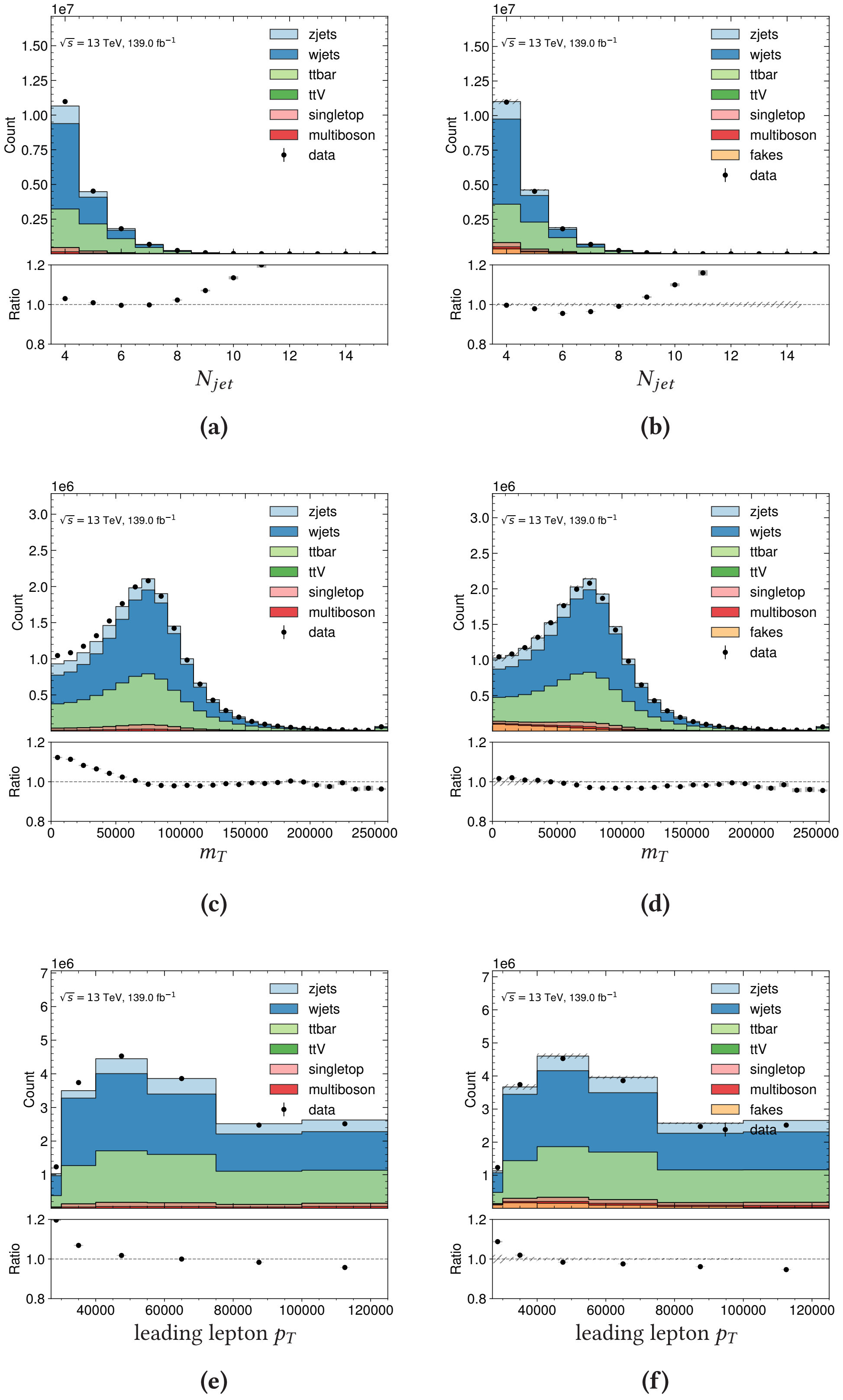
Figure 7.15: Fake transfer factor ε^{misid} for 2015 – 2018 muons.



(a) (b) (c)

Figure 7.16: Real transfer factor ε^{real} for 2015 – 2018 muons.

fit, the backgrounds from $t\bar{t} + jets$ and $W + jets$ is estimated from data, instead of the Monte Carlo simulations used here. This is why the QCD fake estimate contribution is assumed to be correct. Leading lepton p_T exhibits the same slight overestimation and an incomplete closure in the 27 to 30 GeV/c bin, which is likely due to mismodelling in $W + jets$ and $t\bar{t} + jets$ samples as well. A good description of m_T through the fake muon estimate is observed.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.17: 2015 – 2018 muon events, N_{jet} without (7.17a) and with (7.17b) the fake estimate. 2015 – 2018 muon events, m_T without (7.17c) and with (7.17d) the fake estimate. 2015 – 2018 muon events, leading lepton p_T without (7.17e) and with (7.17f) the fake estimate.

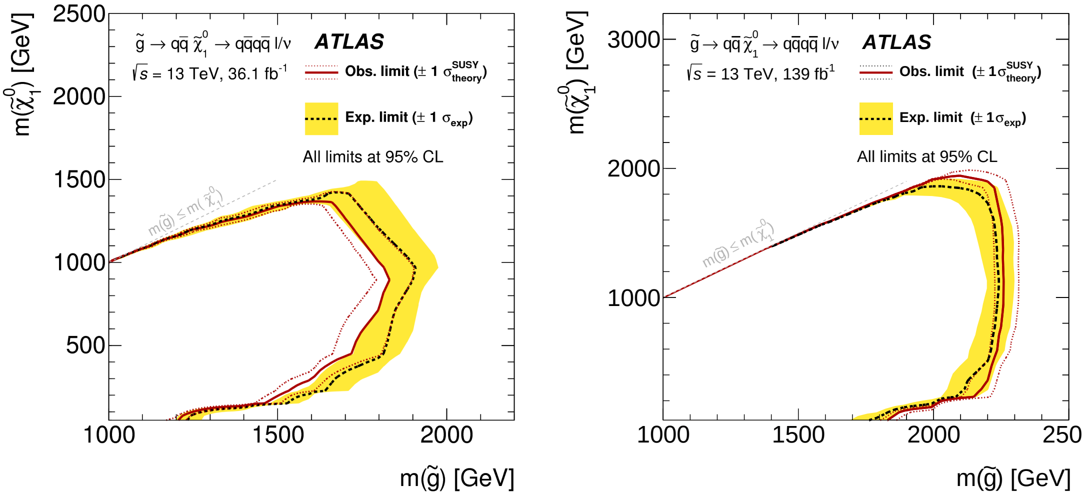
8 Results of the RPV1L Analysis

The RPV1L analysis found no significant excess in the signal regions.

Figure 8.1 shows the exemplary improvement for the expected exclusion limits in the Gluino-LQD model (see Section 6.1). The other signal models improve similarly.

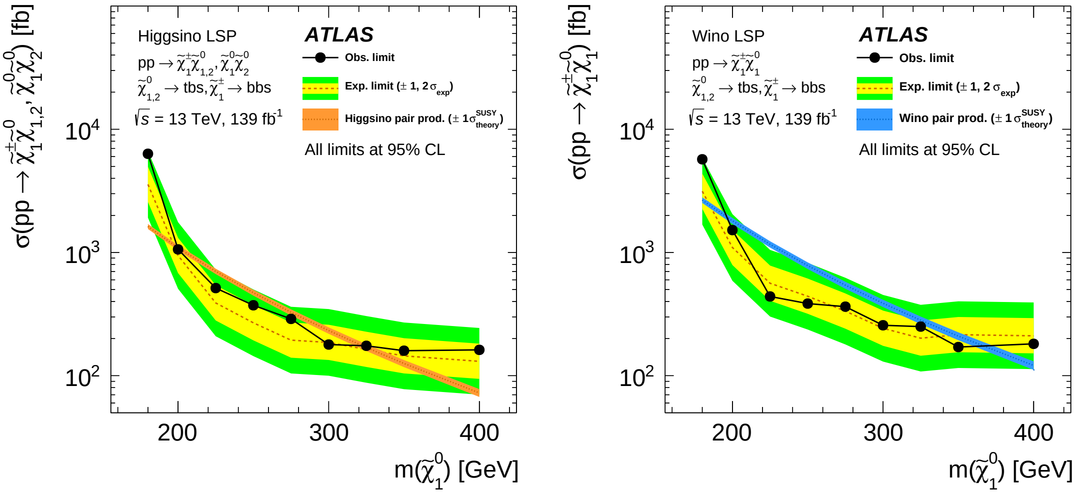
Sensitivity to direct electroweakino production is the main target of the current analysis iteration. Figure 8.2 shows the expected and observed exclusion for the higgsino LSP and wino LSP hypothesis. The analysis is able to exclude both neutralino hypothesis for the most part below ~320 GeV/c^2.

The RPV1L analysis is also sensitive to the Standard Model four-top production. A fit with a four-top signal is performed for a jet threshold of $p_T \geq 40$ GeV/c, resulting in a signal strength of $\mu_{tttt} = 2.0^{+0.9}_{-0.7}$, which is consistent with the original ATLAS measurement [Aad+20].



(a) **(b)**

Figure 8.1: Exclusion limits as a function of gluino and neutralino mass in the previous [Aab+17c] (left) and current analysis (right) [Col21].



(a) **(b)**

Figure 8.2: Expected exclusion of the higgsino LSP and wino LSP hypothesis [Col21].

Part III
Combined Performance Algorithms

ATLAS analyses are commonly performed with *analysis frameworks*. Analysis frameworks provide a configuration interface for physicists to implement and carry out analysis decisions, which define the order of physics object preparation, cleaning, filtering and selection. Technically, most frameworks are either based on the *AnalysisBase* or *AthAnalysis*, which implement a common event store and file operations to read (derived) xAODs. Both are provided as centrally managed releases in the ATLAS Athena repository [Col19].[1] The AnalysisBase release is a lightweight xAOD reading implementation on top of EventLoop and ROOT. AthAnalysis is based on the central production framework Athena.

Frameworks produce a small intermediate format that is common to a sub-group of analyses or even a single analysis. In the simplest case, this format is a ROOT n-tuple of all selected events and objects and is the starting point for final analysis tasks.

Examples of commonly used analysis frameworks in ATLAS are *SUSYTools*, *HWW*, *Hgam*, *CxAOD*, *xTau*, *xAODAnaHelpers*, and *AnalysisTop* [Sofb]. The names already indicate that frameworks often evolve in groups of analysers focussing on specific physics processes (e.g. SUSYTools, HWW, Hgam, AnalysisTop, etc.). Their feature-set ranges from being a simple toolkit of common operations (e.g. SUSYTools) to a standalone program implementing the analysis solely via configuration files (e.g. AnalysisTop). Many frameworks implement the same kind of operations, resulting in duplicated work between maintainers and suffering code quality since fewer developers work on the same code.

CP algorithms and a sensible default configuration in the form of standard sequences address the issues of existing frameworks. They reduce the duplication in existing analysis frameworks by extracting common operations into standardized building blocks. CP algorithms represent building blocks that are scheduled to prepare physics objects and perform common operations like *overlap removal* (OR) between ambiguous objects or the computation of missing transverse momentum (E_T^{miss}). These building blocks may also provide a common language to formulate analysis tasks within the collaboration and thus, improve the technical cooperation in the analysis community. Proto-

[1] https://gitlab.cern.ch/atlas/athena/

typing new analysis ideas may be simpler and result in code that is compatible with any other framework utilizing CP algorithms.

In this Part, the n-tuple production code of the RPV1L analysis is used as a reference. It is based on a modified version of the stop1l-xaod framework to read a derived xAOD format intended for Supersymmetry analyses (SUSY5) and produce the analysis n-tuples.

The stop1l-xaod framework is based on EventLoop and uses SUSYTools to calibrate and select reconstructed objects according to the recommendations of the SUSY subgroup, which in turn are based on the general CP recommendations. SUSYTools uses a list of key-value pairs to configure physics object definitions. For each object type (electrons, muons, etc.), multiple methods are implemented in SUSYTools to follow the CP recommendations of that type.

On top of these operations, the stop1l-xaod framework implements multiple algorithms, performing additional selection and preparation decisions on physics objects and events. In a simplified description, the SUSYTools configuration is preparing loosely selected objects and the stop1l-xaod algorithms apply a tighter selection.

The technical implementation and configuration of CP algorithms is presented in Chapter 9. Chapter 10 contains a comparison of the RPV1L analysis n-tuple production code, implemented with CP algorithms and the reference, as defined in Part II. It outlines the selection of electrons, muons and (b-)jets in both implementations. Validation is done in the form of event-cut-flow comparisons, comparing the number of events after defined physics decisions. Additionally, event yield comparisons in the N_{jet} vs. N_{bjet} distribution and direct comparisons of the n-tuples are presented. The computing performance of CP algorithms is analysed in Chapter 11.

9 Overview of CP Algorithms

Combined Performance Algorithms (CPAs) are a new development for ATLAS Run 3 analyses. CP algorithms wrap CP tools in dual-use algorithms, i.e. they are schedule-able in EventLoop- and Athena-based frameworks and are intended to harmonize the use of physics recommendations in ATLAS analysis software. As described in Chapter 4, CP tools implement physics recommendations for common object types (e.g. electrons, muons or jets) and are used in all ATLAS analysis frameworks. CP tools have the drawback that they are not schedule-able and require additional C++ code to be used in an analysis. Every analysis framework has to maintain the configuration and instantiation of each CP tool, which may require detailed domain-specific knowledge. CP algorithms, on the other hand, can be scheduled through the Python code of analysis job configuration.

CP algorithms also have the potential to improve the speed and efficiency of analysis code, since they offer a central location which can be optimized, rather than optimizing each analysis on its own. They are designed with early rejection of events as a central concept, which can reduce time spent on undesirable events. Additionally, CPAs provide a uniform approach to evaluate systematic uncertainties of the underlying tools.

CP algorithms are used by scheduling them at the very beginning of a user analysis job. There is a centrally provided default sequence of CP algorithms for each object type, e.g. the electron sequence, which apply the configured selections, cleaning and calibration procedures. After their execution, CP algorithms store the prepared physics objects in the event store. Finally, the analysis user-code reads the objects from the event store without the necessity to call CP tools.

Multiple working points, e.g. for different isolation working points, can be configured through partial copies of the CPA sequences, and result in separate *decorations* to the final object. A decoration in ATLAS software refers to the addition of new information to already existing and otherwise constant objects. For example, electrons could be decorated with a flag indicating if that particular electron passes a certain p_T requirement or not.

The details of the CPA implementation may still be subject to change, but CP algorithms are recommended for all Run 3 analysis developments. The general approach is unlikely to change, but more work will go into the configuration layer of CPAs to improve the integration with existing analysis frameworks and workflows.

Framework developers agreed to investigate the CP algorithm integration in preparation for Run 3. First prototypes of an integration with SUSYTools are already implemented. A proof-of-concept integration of CP algorithms in the RPV1L analysis framework, utilizing SUSYTools, is discussed in Chapter 10.

9.1 Infrastructure

CP algorithms and their sequence configurations are implemented in Python, where high-level code results in instantiated C++ objects in EventLoop or Athena. The instantiation itself is handled by the framework. Only the Python configuration and the C++ code of the algorithms is user-facing.

Algorithms are scheduled by being accumulated in an `AnaAlgSequence`, which is a container class that manages the order of algorithm execution [Sofa]. A sequence can have stages, e.g. *calibration*, *selection*, and *efficiency*, to group together multiple algorithms. This simplifies the manipulation of sequences, e.g. by removing the calibration stage of a sequence copy to prepare a second parallel selection based on the same calibration.

The sequence also manages `metaConfig` and `dynConfig` objects for each algorithm. A `metaConfig` object holds the stage name, affecting systematic variations and input or output property names of the corresponding algorithm. `dynConfig` is used to configure properties of the algorithm at configuration-time dynamically. With these two helper constructs, CP algorithms can be dynamically added, removed or changed in a given sequence without breaking dependencies between the algorithms. A simple example where this is necessary is preselection. With a scheduled selection algorithm, a subsequent algorithm can declare a preselection property to only work on events that pass this selection. The preselection condition cannot hold a hard-coded value; otherwise, it would depend directly on the previous selection algorithm, resulting in a less flexible sequence.

CP algorithms inherit from `EL::AnaAlgorithm`[1] [Colb]; [Col19], providing methods to access the event store, manage histograms and basic message logging capabilities. The dual-use nature of this class allows for simple unit testing and integration into most existing ATLAS analysis codebases.

Every `AnaAlgorithm`, and therefore every CPA, implements an `execute()` method, which is called during the event loop for every event and contains the body

[1] https://gitlab.cern.ch/atlas/athena/tree/21.2/PhysicsAnalysis/D3PDTools/AnaAlgorithm

of work of the algorithm. CP algorithms commonly apply CP tools in their `execute` method to break down the workload into individual components. These tools are often exposed to the Python configuration to make the algorithm and its tools fully configurable.

Algorithms also commonly implement the `initialize()` and `finalize()` methods. In `initialize()`, the algorithm performs initial preparations, e.g. retrieving handles to tools and event store objects or allocating resources for its operation. `finalize()` is called after the event loop is finished and contains whatever code the algorithm needs to execute at that point.

Algorithms can read, copy, and write objects and object containers to and from the event store. Object names in the event store are typically exposed to the Python configuration as configurable properties, e.g. allowing for type-agnostic algorithms to be applied without any changes to the C++ code.

Systematically varied objects, e.g. electrons after applied calibration, are stored in the event store with different names. These types of objects are specified in the Python configuration by adding the suffix _%SYS% to the object name in the event store. The suffix _NOSYS corresponds to the nominal version of the object, without any systematic variation applied. On the C++ side, systematic handles automatically resolve these suffixes to point to the correct container. CP algorithms that work on systematically varied objects loop over all relevant systematic variations in their `execute` method. By accessing the event store objects through systematic handles, the system can optimize object reading such that each algorithm is only applied on relevant systematic variations. Unaffected systematic variations are resolved by creating shallow copies[2] pointing to the nominal container.

9.2 Centrally Provided CP Algorithms

The CP algorithms and default sequence configurations are part of the ATLAS Athena repository[3] [Colb]; [Col19]. This Section gives an overview of the existing packages, which are intended to provide everything an analysis developer needs to implement CP recommendations in their framework.

[2] Shallow copies are thin objects that only hold variables with modified values and refer to the original container for all unmodified variables. *Deep copies* also copy unmodified variables, resulting in a complete and independent representation of the data.

[3] https://gitlab.cern.ch/atlas/athena/-/tree/21.2/PhysicsAnalysis/Algorithms

Asg The Asg package contains auxiliary algorithms to deal with very common analysis tasks. An event selection sequence does initial event filtering through GoodRunsList, PrimaryVertex and event cleaning algorithms.

The `GeneratorAnalysisSequence` decorates simulated objects with the corresponding truth information. A `PileupAnalysisSequence` and the corresponding algorithm corrects simulated events to the corresponding pileup profile of the data run. Multiple algorithms help with creating shallow or deep copies of existing objects and creating links to the original object. Additionally, there are algorithms to create simple output ROOT n-tuples or xAOD files from objects in the event store. The Asg package also contains the necessary algorithms to prepare a complete list of relevant systematic variations. An overlap removal sequence and the corresponding algorithm processes multiple object containers via overlap removal CP tools to resolve ambiguous objects from each event. Finally, a range of general selection, bookkeeping, and histogramming algorithms are available for generic analysis tasks.

EGamma The EGamma package contains multiple CP algorithms and sequences to apply CP recommendations for electrons and photons. Both sequences do general selections, calibrations, isolation and output copy operations, as described in Chapter 3. In the default configuration, both sequences only allow for a single isolation working point, but this can be circumvented with multiple partial sequences. The packages contain many algorithms wrapping the electron and photon CP tools, e.g. the `EgammaCalibrationAndSmearingAlg` to calibrate electrons or photons.

Similar packages exist for muon, tau, and jet CP recommendations, with the corresponding sequences and algorithms.

FTag This package implements flavour tagging operations, e.g. identifying jets from b-quark decays. The `FTagAnalysisSequence` modifies a given jet sequence to correctly use b-tagging working points and compute the corresponding identification efficiencies.

MET To analyse the E_T^{miss} of a given event, all other reconstructed objects need to be considered since it is derived from the remaining transverse momentum, after removing all identified objects. The `MetAnalysisSequence` reads the reconstructed leptons and jets. Various algorithms derive E_T^{miss} and handle corresponding systematic variations.

Table 9.1: Available options to the `makeMuonAnalysisSequence` function.

Option	Description
`dataType`	Data, MC, or fast simulation
`workingPoint`	Isolation and identification working point (Tight, Medium, Loose, VeryLoose, HighPt, LowPtEfficiency)
`deepCopyOutput`	If true, output containers are stored with a copy of all information. (default `False`)
`shallowViewOutput`	If true, output containers are stored as a shallow (“view”) copy, pointing to the original objects in the event store for all branches that are not changed. Mutually exclusive with `deepCopyOutput`. (default `True`)
`postfix`	Postfix to decoration and algorithm names. Useful if the muon sequence is scheduled multiple times.
`ptSelectionOutput`	Decides if the p_T selection is applied to the output container or not. (default `False`)
`qualitySelectionOutput`	Decides if the quality requirements are applied to the output container or not. (default `True`)
`enableCutflow`	Store cutflow information of intermediate steps of this sequence. (default `False`)
`enableKinematicHistograms`	Write the kinematic histogram. (default `False`)

Trigger A trigger analysis sequence identifies which triggers have passed for a given event and provides the correct pre-scale factors in case of simulated events.

Tracking In this package, the `VertexSelectionAlg` helps with studying available vertices and identifying the primary vertex.

9.3 Workflow Example: Muon Sequence

In general, CP algorithms are intended to run before the analysis framework code. The CP algorithms decorate the physics objects, following the standard recommendations and the sequence configuration. Standard sequences of CP algorithms are created through `make*Sequence` functions in Python. The output of these sequences provide efficient access to all selected objects in the processed events.

This Section contains a discussion of the default muon analysis sequence[4] to give an example of how CP algorithm sequences work in detail [Colb]; [Col19]. Table 9.1 lists all available options of the muon sequence at the time of writing this dissertation. These options represent the first configuration options exposed to the user. It is still possible to configure scheduled algorithms and their tools afterwards. First, the sequence function prepares the default values for the meta configuration:

```
seq.addMetaConfigDefault ("selectionDecorNames", [])
seq.addMetaConfigDefault ("selectionDecorNamesOutput", [])
seq.addMetaConfigDefault ("selectionDecorCount", [])
```

These lists hold the decorations created by the sequence, often used for preselection to skip undesirable events early. A separate list of decorations is managed for the output container, including or excluding the p_T or quality selection decorations, depending on the configuration. The first algorithm scheduled in the sequence is the `MuonEtaCutAlg`:

```
1  alg = createAlgorithm( ’CP::AsgSelectionAlg’, ’MuonEtaCutAlg’ + postfix )
2  addPrivateTool( alg, ’selectionTool’, ’CP::AsgPtEtaSelectionTool’ )
3  alg.selectionTool.maxEta = 2.5
4  alg.selectionDecoration = ’selectEta’ + postfix + ’,as_bits’
5  seq.append( alg,
6      inputPropName = ’particles’,
7      outputPropName = ’particlesOut’,
8      stageName = ’selection’,
9      metaConfig = {
10         ’selectionDecorNames’ : [alg.selectionDecoration],
11         ’selectionDecorNamesOutput’ : [alg.selectionDecoration],
12         ’selectionDecorCount’ : [2]
13     },
14     dynConfig = {
15         ’preselection’ : lambda meta : "&&".join (meta["selectionDecorNames"][:])}
16 )
```

The algorithm is created with a type and a name. Additionally, a private instance of the `CP::AsgPtEtaSelectionTool` is assigned to the algorithm. In line 3, the `maxEta` property of the selection tool is set to 2.5, adding the requirement of $|\eta| \leq 2.5$ for all muons. Objects that do not satisfy this requirement are not removed. Instead, the decoration `selectEta` is added to mark objects that pass this requirement. Finally, the configured algorithm is appended to the sequence. It is declared as part of the `selection` stage, and the `metaConfig` and `dynConfig` parameters are used to add the decoration `selectEta` to the global and output decoration lists. This way, the following algorithms’ preselection will ensure that their workloads are only executed for objects and events that pass the MuonEtaCutAlg.

[4] https://gitlab.cern.ch/atlas/athena/-/blob/21.2/PhysicsAnalysis/Algorithms/MuonAnalysisAlgorithms/python/MuonAnalysisSequence.py

The remaining analysis sequence continues similarly. Each algorithm is first created, its configuration defined, and appended to the sequence. User parameters of the sequence generation method can include or exclude the addition of individual algorithms.

The following algorithms filter muons based on their vertex properties, apply calibrations and decorate whether muons are isolation, in addition to p_T and scale factor information.

The muon analysis sequence is an example of a linear sequence of algorithms that prepare muons from simulated or reconstructed data. For more in-depth requirements, the individual algorithms can be changed, removed, reordered or appended. Multiple sequences can be used to use different working points.

10 Reproducing RPV1L N-Tuples with CP Algorithms

To test the current state of CP algorithms and their sequences, the n-tuple production of the RPV1L analysis (see Chapter 6) is re-implemented with CP algorithms. The goal is to implement the selection, calibration and generally recommended physics object treatment for the objects relevant to the RPV1L analysis using CP algorithms. These objects are electrons, muons and – potentially b-tagged – jets.

The n-tuple production code of the RPV1L analysis is a good test project for multiple reasons. First, the underlying stop1l-xaod framework is migrated to use the `AnaAlgorithm` class, which is also the technical basis for CP algorithms. This results in a compatible Python configuration of the sequences. Next, the RPV1L analysis uses `SUSYTools` (ST), which is a widely adopted analysis package within the SUSY group of ATLAS. Successfully using CP algorithms with ST would show the usability of the CP algorithm workflow and could serve as a prototype of their integration in common ATLAS analysis frameworks. Finally, reproducing the RPV1L analysis with CP algorithms serves as a "real world" validation, proving that CP algorithms provide all necessary physics analyses tools.

10.1 Technical Setup

The validation is performed with a simulated test sample of 10000 $t\bar{t}$ events[1] with the original implementation, referred to as *reference*, and a new CP algorithm version.

Figure 10.1 shows a diagram of the RPV1L analysis job structure in the stop1l-xaod framework. A sequence of algorithms processes each event during the event loop in the given order. A shared SUSYTools instance is created during initialization and provides helper methods to the algorithms implementing the CP recommendations.

[1] For reference, the exact sample is: `mc16_13TeV.407342.PhPy8EG_A14_ttbarHT1k5_hdamp258p75_nonallhad.deriv.DAOD_SUSY5.e6414_s3126_r9364_p4172`.

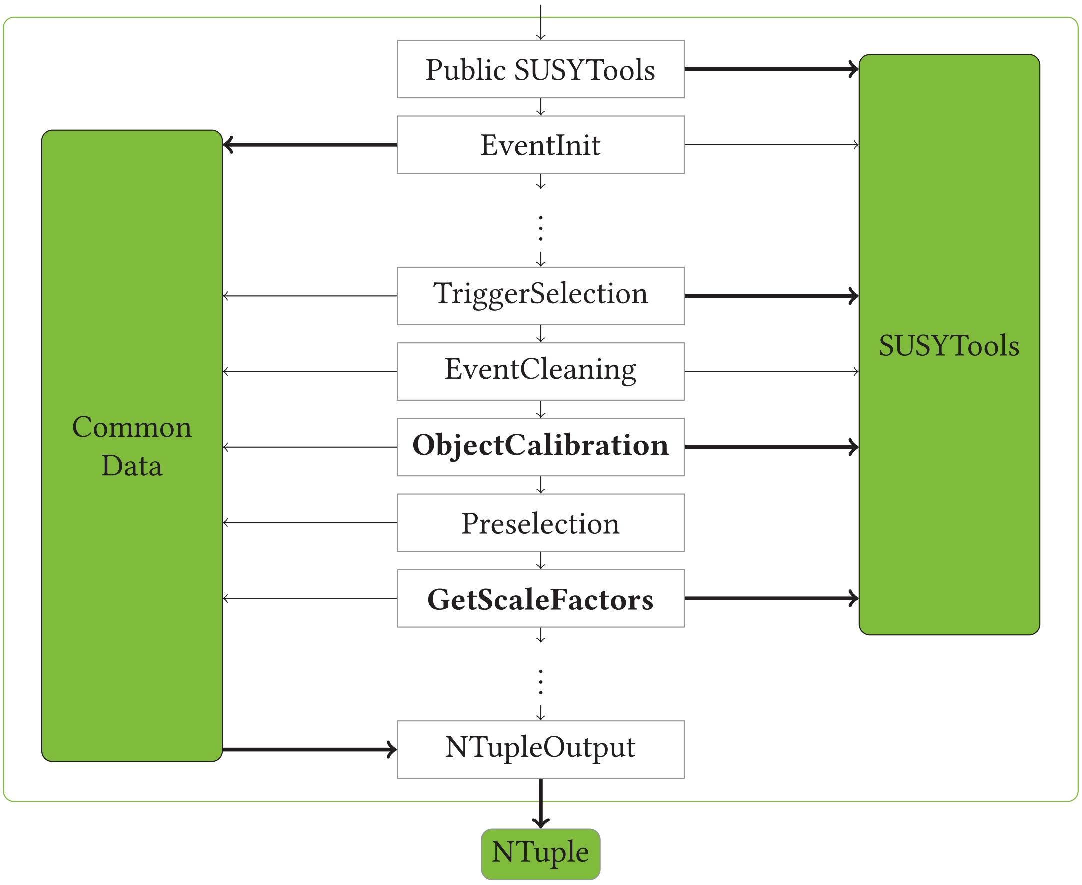


Figure 10.1: Algorithm sequence of the stop1l-xaod framework in the RPV1L analysis. A shared SUSYTools instance provides object selection and other helper methods to every algorithm. Data to be stored in the n-tuple output, is collected in the `CommonData` singleton structure. Bold arrows indicate relations of particular interest (singleton instantiation, output, and relevant for object selection). The `ObjectCalibration` and `GetScaleFactors` algorithms are of special interest since they implement all object definitions and scale factor calculations. Several algorithms that do not affect event selections or object definitions are omitted for readability (marked with "...").

The TriggerSelection, ObjectCalibration, and GetScaleFactors algorithms are the most frequent clients of SUSYTools and have implicit calls to CP tools that have to be replaced in the CP algorithm implementation.

During the initialization of ST, *pile-up reweighting* (PRW) is applied to simulated events, to match the observed pile-up profile in data. PRW is not applied twice if it was run before, e.g. in a corresponding CP algorithm. A singleton class, CommonData, is created by the EventInit algorithm during initialization and is used within stop1l-xaod to track all event and object data to be stored in the final n-tuple. N-tuple writing is implemented through the NTupleOutput algorithm.

The reference implementation executes the sequence exactly as presented in Figure 10.1. The CP algorithm implementation schedules all object definition sequences before the beginning of the diagram in Figure 10.1. In this way, all objects (electrons, muons, and (b-)jets) are calibrated and decorated with all selections, before the stop1l-xaod sequence starts. A new re-implementation of SUSYTools accesses these decorations instead of calling CP tools. This lightweight implementation of SUSYTools is re-implementing only a selected list of methods and forwards all other methods to the original implementation. The changed methods and their purpose are:

- GetJets: Apply CP recommendations to the whole jet container. Calls FillJet and applies b-tagging.
- FillMuon, -Electron, -Jet: Apply CP recommendations to an individual jet, muon, or electron object.
- IsSignalElectron: Decorate an electron object with signal definition flags.
- OverlapRemoval: Do overlap removal between provided electron, muon and jet containers.
- BtagSF: Calculate the b-tagging scale factor.
- JVT_SF: Calculate the JVT scale factor.
- GetTotalMuonSF: Calculate the total muon scale factor.
- GetTotalElectronSF: Calculate the total electron scale factor.

Only minor technical changes to the reference implementation of stop1l-xaod and SUSYTools are made. A shared interface class of SUSYTools is used in all stop1l-xaod algorithms, to make the reference and lightweight ST implementations configurable from Python.

The job configuration of the reference happens in two places. First, a regular SUSYTools configuration file of key-value pairs sets a very loose baseline definition of all objects. Second, the stop1l-xaod algorithm sequence defines – potentially tighter – objects in the algorithm properties. This is mainly done via properties of the `ObjectCalibration` algorithm.

The new CP algorithm implementation adds a layer of configuration, by providing configuration options in the `makeSequence` methods, as well as in the underlying algorithm properties. To achieve the same object definitions as the reference, the CP algorithm sequences must be configured to select objects in the same order and with the same parameters as the original SUSYTools methods.

10.2 Object Definitions

The physics objects are read from the centrally produced derivation files and reconstructed as described in Chapter 3. Events recorded during bad detector states are removed and ambiguous (overlapping) objects resolved. The remaining objects are calibrated and selected, e.g. by isolation working point, p_T, η, et cetera, according to the analysis objectives.

The electrons and muons of the RPV1L analysis are categorized in a baseline and a signal set, where the latter is always more selective than the baseline selection. Jets have to fulfil certain baseline conditions and are flavour tagged to identify jets from decaying b-quarks.

Figure 10.2 shows a diagram of stored and intermediate object containers. The containers named “Jets”[2], “Electrons”, and “Muons” are read from the DxAOD file in both versions. In the reference implementation, these containers are read in `ObjectCalibration` to produce intermediate containers, prefixed with `calib`. These containers are then treated as follows with the default SUSYTools implementation:

1. Calibration and object decoration via calls to SUSYTools helper methods.

2. Ambiguities between objects in the `calib*` containers are resolved in overlap removal, marking objects that pass OR without removing any.

[2] “Jets” is just a simplified description. For reference, the real container name is “AntiKt4EMPFlowJets”.

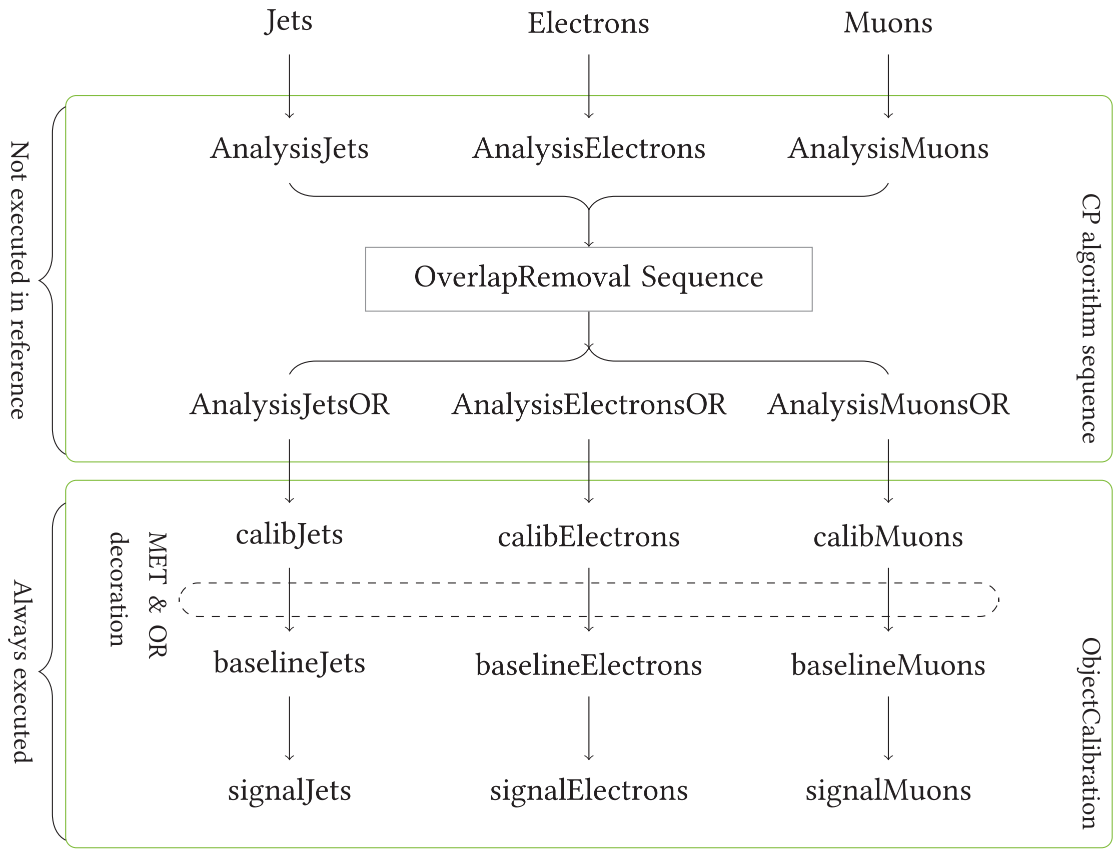


Figure 10.2: Intermediate physics object containers in CP algorithms and the stop1l-xaod framework. The first containers are read from the inputfile. In the reference, these are directly used as a source for the `calib*` containers, i.e. skipping the first box. In the CP algorithm implementation, the file containers are processed to pre-select and decorate all objects according to the sequence configurations. The output of these sequences is then the source for all `calib*` containers. Each following container may contain less selected objects but additional object decorations. E_T^{miss} calculation in the CP algorithm sequence is done similarly to the overlap removal sequence, reading `AnalysisJets` and leptons passing the baseline selection. The resulting `AnalysisMET` container is read on the stop1l-xaod side (dashed circle) instead of calling a SUSYTools method to compute E_T^{miss}.

3. E_T^{miss} is calculated with objects in `calib*` containers that are passing baseline requirements, i.e. not including overlap removal.

4. Finally, filtered baseline and signal objects are stored separately in the `Common-Data` structure. These selections are based only on the prepared decorations, e.g. `baseline(electron)` and `passOR(electron)` to define baseline electrons.

In the CP algorithm version, the basic objects from the DxAOD file are read in the first algorithm of each sequence, and stop1l-xaod is only executed with the lightweight SUSYTools instance:

1. The jet and flavour-tagging sequences prepare `AnalysisJets` with the corresponding object decorations. Similarly, electron and muon sequences prepare the corresponding lepton containers.

2. The overlap removal sequence reads these objects to remove ambiguous objects in all `Analysis*` containers and, as opposed to the stop1l-xaod implementation, produces new containers that potentially only stores objects passing the overlap removal (configurable).

3. Similarly, the E_T^{miss} sequence reads `AnalysisJets` and leptons passing baseline requirements to compute the missing transverse momentum. The output is stored in a `AnalysisMET` container. (Not shown in Figure 10.2)

4. `Analysis*OR` objects are used in `ObjectCalibration` to prepare the intermediate `calib*` containers by calling the lightweight ST methods.

5. `AnalysisMET` is read instead of calling SUSYTools method to compute E_T^{miss}. (dashed line)

6. The final baseline and signal containers are created as in the reference implementation.

In the following subsections, the technical details are highlighted that are necessary to select the same objects in the CP algorithm implementation compared to the reference.

10.2.1 Jet and b-Jet Definition

The jet container is read in the `ObjectCalibration` algorithm, and each jet is processed with SUSYTools helper methods. Each jet is calibrated and decorated with jet vertex and b-tagging tagging information.[3] A second, looser, b-tagging working point is prepared for overlap removal to ensure that b-tagged jets are retained over leptons in most relevant cases. The baseline and signal requirements on p_T and η are checked, and corresponding flags are decorated. Each jet that passes baseline and jet vertex tagging (JVT) requirements is assigned the overlap removal priority (2), over jets that only pass baseline (1) or neither of both requirements (0). If a signal jet is simulated, it is also decorated with the truth information about its origin from a b-decay. The resulting container of calibrated jets still contains all jets, with only added decorations about passing the mentioned decorations, indicating fulfillment of the requirements mentioned above.

Jets that pass baseline requirements and overlap removal are stored in the `baseJets` container. Signal jets are a strict subset of `baseJets`, and are stored in the `signalJets` container. A last container is created to store all signal jets that are b-tagged.

Jet vertex and b-tagging have associated scale factors that are computed in the `GetScaleFactor` algorithm. In this algorithm, the nominal scale factor and a scale factor for each affecting systematic uncertainty is computed by calling SUSYTools helper methods.

In the CP algorithm implementation, all implicit calls to CP tools of the ST helper methods are removed by using the lightweight ST implementation. The default jet and flavour tagging sequences provide most of the required functionality, and only minor changes are necessary to achieve the same jet selection as in the reference. For example, certain decorations had to be renamed to match the expected decorations in SUSYTools, e.g. the JVT flag `passJVT` and the corresponding scale factor name. The p_T and η requirements for forward jets are changed to match the default ST configuration. Finally, two simple decoration algorithms are added to decorate each jet with the baseline flag and overlap removal priority.

Calibrating jets in CP algorithms results in very small differences in jet energy properties on the order of 10^{-7}. These differences appear on a binary comparison between the produced n-tuples, but derived quantities, such as E_T^{miss}, are affected as well. The impact of these differences is discussed in Section 10.3.

[3] See Section 3.4 for more details on jet reconstruction.

10.2.2 Electron and Muon Definition

Reconstructed electrons and muons, see Sections 3.2 and 3.3 respectively, are read in the `ObjectCalibration` algorithm. Individual muons are calibrated and decorated with baseline and signal p_T and η requirements. Muons originating from the primary vertex are decorated with the baseline flag and an overlap removal priority of 2. Additionally, two isolation working points (`isol`, `isolHighPt`) and the signal identification (ID) working point are decorated. Each muon passing the baseline, signal ID, and isolation requirements, is decorated with the signal flag. Finally, muons are marked as "bad" if the associated track does not fulfill certain requirements, e.g. originating from cosmic events, if the muon track does not match the collision region. All "good" muons that pass baseline requirements and overlap removal are stored in the `baseMuons` container, while signal muons are stored in `signalMuons`.

Similar procedures are applied for the electron selection, except for the check of bad or cosmic tracks. Electrons have additional decorations to identify charge flips or cut on ambiguity classifiers.

The CP algorithm sequence for muon preparation is extended with simple decorator algorithms for the baseline requirement and overlap removal priority again. Additionally, a short algorithm is included to identify bad muons by their associated track parameters in the same way as it is implemented in the SUSYTools reference. Signal muon objects are decorated with multiple isolation working points, such that they can be read in the lightweight SUSYTools methods. Several scale factor algorithms are added to the sequence, e.g. to compute the scale factors for isolation and identification. Muon trigger scale factors can only be computed after overlap removal is applied, which is why the corresponding algorithms are scheduled after the OR sequence and therefore only decorate muons in the `AnalysisMuonOR` container.

The electron CP algorithm sequence requires only minor changes as well, e.g. the decorator algorithms for baseline and OR priority. All scale factors for electrons can be computed by individual algorithms in the electron sequence, even before overlap removal.

Slight differences in the total scale factors for electrons and muons, with a maximum on the order 10^{-7}, are observed on the binary level, without affecting the final cutflow or yield results.

10.2.3 Overlap Removal in CP Algorithms

The current default overlap removal sequence manages individual algorithms to remove overlapping objects between two object types. One algorithm manages a single tool, e.g. to apply overlap removal to individual muons and jets. On the other hand, the reference implementation in SUSYTools uses the `ToolBox` helper class that implements recommended configurations and prepares all overlap removal tools at once.

To match the behaviour between both approaches, the default CP algorithm sequence for overlap removal has to be extended in terms of configurable algorithm properties. The `MuPFJetORT` is missing the b-jet label to include b-tagging information in its OR decisions. Additionally, options for removing calorimeter muons in the electron-muon OR tool and "ghost association" in the muon-jet OR tool are added to reflect the configuration of the ToolBox approach. In SUSYTools, user priorities are set for pre-selected jets (1) and selected jets and leptons (2). These user priorities are explicitly enabled in the lepton-jet OR tools. With these changes, the processed events behave the same in both setups. By default, the output container of the `OverlapAnalysisSequence` is only storing objects that pass overlap removal.

10.2.4 Missing Transverse Momentum

The calculation of missing transverse momentum is done without applied overlap removal.[4] In the reference implementation of `ObjectCalibration`, the `GetMET` method calculates E_T^{miss} by calling the corresponding CP tools on all `calib` -containers.

In the CP algorithm implementation, that function call is completely replaced by just reading `AnalysisMET` from the event store. The standard sequence provides all necessary options, and only minor changes are necessary to reflect the SUSYTools configuration. The `softTermKey` is set to `PVSoftTrk`, the `doJetJVT` option is enabled, and `doTrackMet` is disabled. In the `MetBuilderAlg`, the `finalKey` parameter is set to `Final`.

The input of the E_T^{miss} sequence consists of the `AnalysisJets`, as defined above, and selections of the muon and electron containers, only containing objects that meet the baseline requirements. This selection is stored in temporary containers.

Minor differences in E_T^{miss} and the derived m_T in a binary comparison have no effect on the resulting event-cutflow and yields.

[4] See Section 3.5 for more details on E_T^{miss} reconstruction.

10.2.5 Systematic Variations

Every CP tool declares the systematic variations of its operations and can be executed with these variations applied. CP tools are typically evaluated in one-sigma up and down variations, in addition to the nominal execution. The reference implementation of the RPV1L analysis offers two approaches to systematic variations:

1. The analysis is executed with the nominal object definitions. In the `GetScaleFactors` algorithm, a loop over all weight-affecting systematic variations is performed to calculate the variations of scale factors with respect to the nominal signal objects.

2. The analysis is executed only with one kinematics-affecting systematic variation. In this mode, no weight-affecting variations are evaluated, and an individual n-tuple production job is required for each kinematics-affecting systematic variation.

In both approaches the systematic variation is applied to the entire SUSYTools collection of CP tools. After the required operations are performed, SUSYTools is reset to the nominal mode.

CP algorithms treat systematic variations differently. Each CP algorithm that is potentially affected by systematic variations executes a loop over all configured systematic variations in its `execute()` method. For each loop iteration, a different container is prepared, e.g. `AnalysisMuons_%SYS%`, where `%SYS%` is replaced with NOSYS for the nominal execution and otherwise the current systematic variation name. If the algorithm is not affected by the current variation, the container points to the nominal container, in order to be transparent to subsequent algorithms that might need to access the systematically varied container. There is no differentiation between kinematics- and weight-affecting systematic variations in CP algorithms in favour of a simplified configuration. However, it is possible to evaluate weight-affecting variations only by creating systematically varied decorations on the nominal container. Through this mechanism, e.g. scale factor decorations can be evaluated without creating a systematically varied container for the relevant weight-affecting variations.

In order to integrate the CP algorithm approach to systematic variations with the stop1l-xaod framework, the sequences are configured to either run a selected list of weight-affecting systematics or a list of kinematics-affecting systematics. Both versions always also run the nominal evaluation to define a default container for unaffected algorithms to refer to. The `ObjectCalibration` algorithm is modified such that jets, muons and electrons loop over all systematic containers apply the decorations used in the analysis. Only the nominal container or a selected kinematics-affecting systematic variation is written out to the n-tuple. In `GetScaleFactors`, all objects of weight-affecting variations are read as systematically varied decorations from the nominal container, and scale factors are computed for all corresponding objects that match the signal requirements. This implementation is not tuned to yield the same values as the reference but only added to compare the computational workloads in both setups.

The strength of the CP algorithm approach is observable if the analysis evaluates multiple kinematics-affecting systematic variations since individual unaffected algorithms resolve the systematically varied container by pointing to the nominal container. In comparison, the reference implementation requires complete re-executions of the whole process for each variation, which does contain redundant operations. The performance differences of both setups with systematic variations are discussed in Section 11.4.

10.3 N-Tuple Comparison

10000 $t\bar{t}$ events are processed as a benchmark for the n-tuple production with the reference and the new CP algorithm implementation of the RPV1L analysis. In the resulting outputs, only minor differences in scale factors, jet energy calibrations, E_T^{miss}, and derived quantities are observed. An event by event comparison shows that these differences never exceed a maximum of 0.006%, as shown in Table 10.1. Figure 10.3 shows the ratio of the jet energy and m_T distributions of both versions. Discrepancies in the binary comparison are negligible and likely caused by differences due to floating-point operations.

Both versions contain the same events with one lepton and at least four jets, as can be seen in Listing 10.4. At every intermediate step, the algorithms of the stop1l-xaod framework select the same number of events. Another quantity to consider is the two-dimensional distribution of the number of jets vs. number of b-tagged jets. Listing 10.5 shows the identical yields for both versions. This shows that the configuration of both versions are identical with respect to object definitions (ID, isolation, etc.) and overlap removal.

Table 10.1: Maximum relative differences observed in a binary comparison of the n-tuples of both setups with 10000 $t\bar{t}$ events.

Variable	Difference in %
met	1.44e-05
met_phi	5.19e-05
mt	0.006
minmax_mass	1.19e-07
subleadmt	7.60e-05
jet_e	1.19e-07
zll_met	4.02e-07
sf_no_pu	1.33e-07
sf_total	2.24e-07

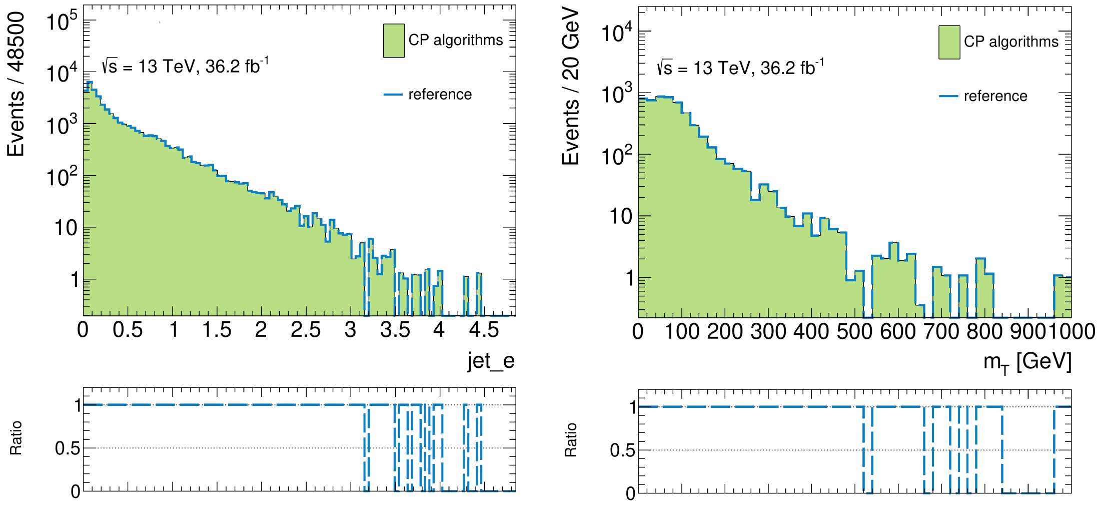


Figure 10.3: Jet energy and m_T distributions of 10000 $t\bar{t}$ events in the CP algorithm and reference setups. The lower plots show the ratio between both versions, i.e. 1 corresponds to an exact match.

Figure 10.4: Identical event cutflow of the CP algorithm and the reference implementation.

```
mc16_13TeV.407342.Ph in Preselection     Unweighted events  Weighted events   n el           el   n mu           mu
All Events                                           14000      1.00499e+07  14000  1.00499e+07  14000  1.00499e+07
DxAOD skimming                                       10000      7.17944e+06  10000  7.17944e+06  10000  7.17944e+06
Truth HT filter                                      10000      7.17944e+06  10000  7.17944e+06  10000  7.17944e+06
Lepton trigger                                        6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
GRL                                                   6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
Tile                                                  6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
LAr                                                   6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
SCT                                                   6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
Core                                                  6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
Primary vertex                                        6167      4.41258e+06   6167  4.41258e+06   6167  4.41258e+06
Jet/MET cleaning                                      6146      4.39712e+06   6146  4.39712e+06   6146  4.39712e+06
Bad muon veto                                         6146      4.39712e+06   6146  4.39712e+06   6146  4.39712e+06
>= 1 baseline lepton                                  5590      3.99818e+06   5590  3.99818e+06   5590  3.99818e+06
>= 1 signal lepton                                    4888      3.49619e+06   4888  3.49619e+06   4888  3.49619e+06
Assign lepton channel                                 4888      3.49619e+06   2390  1.70613e+06   2672  1.91223e+06
Leading lep pt >= 27 GeV                              4822      3.44761e+06   2372  1.69288e+06   2624  1.8769e+06
Signal lepton is leading lepton                       4736      3.38578e+06   2330  1.66197e+06   2580  1.84598e+06
MET > 0 GeV                                           4736      3.38578e+06   2330  1.66197e+06   2580  1.84598e+06
MET < 18446744073709551615 GeV                        4736      3.38578e+06   2330  1.66197e+06   2580  1.84598e+06
MET > 0 GeV or photon                                 4736      3.38578e+06   2330  1.66197e+06   2580  1.84598e+06
>= 2 jets                                             4726      3.38136e+06   2321  1.65829e+06   2578  1.84451e+06
>= 4 jets || >=2 b-jets || >=2 SS leptons || emu+bjet 4510      3.24446e+06   2199  1.58174e+06   2475  1.77901e+06
```

Figure 10.5: Identical weighted yields of N_{jet} vs. N_{bjet} in the CP algorithm and reference implementation.

```
pt>40 GeV,
btag njets = 4         njets = 5         njets = 6         njets = 7         njets = 8        njets = 9        njets >= 10
== 0 831.22 +/- 66.17  844.92 +/- 66.85  661.58 +/- 58.81  375.60 +/- 44.28  199.71 +/- 31.63 59.08 +/- 17.19  34.44 +/- 13.04
== 1 2065.76 +/- 102.55 2641.52 +/- 115.54 2032.74 +/- 100.60 1344.64 +/- 82.07 640.10 +/- 55.91 241.49 +/- 34.73 70.93 +/- 20.01
== 2 1080.43 +/- 73.06  1706.02 +/- 91.97  1525.75 +/- 86.22  1165.39 +/- 75.22 685.45 +/- 58.92 251.43 +/- 35.50 75.96 +/- 19.72
== 3 38.48 +/- 13.62   128.48 +/- 24.12  184.50 +/- 29.86  98.72 +/- 21.31   122.08 +/- 24.22 43.95 +/- 14.10  13.16 +/- 7.60
>= 4 0.00 +/- 0.00     10.33 +/- 7.33    9.47 +/- 6.70     32.60 +/- 11.70   22.14 +/- 10.12  8.78 +/- 6.28    0.00 +/- 0.00

sum over table = 19246.84
```

11 Performance Analysis

This Chapter contains a study of the computing performance of the RPV1L n-tuple implementation with CP algorithms and its reference. The computing workload of both implementations is expected to be similar if no systematic variations are evaluated. Differences in computing performance might be caused by inefficiencies in the job configuration or redundant operations. A study of the computing performance could help to point out the critical sections of the configuration, resulting in "best-practice" recommendations for CP algorithm users.

The event sampling[1] based [Cora] profiler Intel® VTune™[2] [Corc] is used to study the n-tuple production in detail. Occasionally, the `ittnotify` library is used to limit the data collection of VTune to the event loop by starting the collection in the first event of the first algorithm and stopping in the `finalize` method of the `EventInit` algorithm.

Two essential metrics are reported by the production jobs themselves, namely "wall clock time" of the event processing and the rate of events per second. Events per second are measured both including and excluding the initialization time of the process. A rate without initialization time is a better approximation of real-world analyses since n-tuple production is typically run on large amounts of data. The constant initialization time becomes negligible in this case. For this reason, the event rate without initialization time is used as the final metric to evaluate the performance of a given setup.

All performance measurements are executed on the same computer:

- Intel® Core(TM) i7-6700 CPU @ 3.40GHz, 4 physical cores, 8 logical cores
  2× 128kB L1 (instruction, data), 1MB L2 (unified), 8MB L3 (unified)
- 8GB DDR4 RAM @ 2133 MHz
- SSD: SAMSUNG MZ7LN256 256GB, 520 MB/s seq. write, up to 550 MB/s read, 100K IOPs random read, up to 90K IOPS random write

[1] *Hardware event* is referring to the data of dedicated performance monitoring units in the CPU. Regular readout of these counters enables profilers to derive an estimated utilization of the CPU with little overhead in execution time.

[2] As part of Parallel Studio XE 2020.4.912.

During performance measurements, no other processes are executed on the computer, except for the usual system processes. This minimizes fluctuations between measurements. Wall times and event rates are measured as the average of 10 repetitions with the corresponding standard deviation. Occasionally, the minimum or maximum of 10 repetitions is provided to give an estimate of the best-case performance.

Event sampling based profiling is prone to larger fluctuations since performance metrics are collected at regular intervals during the process execution. These *samples* can vary compared to a second profile with the same configuration. Thus, the results of profiles are only used in relative measurements between changes to the configuration and to locate interesting sections of the code.

The n-tuple code depends on a published analysis release, distributed via the CERN VM file system [Blo+20]. Software read from this network storage is automatically cached locally. For this reason, the first execution of a given process can be slower than any subsequent execution.

To improve the reproducibility between multiple executions, the Linux kernel caches are dropped by calling `sysctl vm.drop_caches=3`, freeing up the RAM of any cached pages and slab objects, e.g. dentries and inodes [RM]. This also emulates a typical analysis process, where reading data from disk or even network storages is a regular operation.

11.1 Initial Performance Evaluation

An initial evaluation of the CP algorithm implementation and the reference of the n-tuple production code is done to assess the general performance of both setups. At the time of this initial test, the CP algorithm implementation did not include the E_T^{miss} and scale factor calculation, and the corresponding default SUSYTools methods are used. This might also lead to some redundant calculations since the CP algorithm sequences already contain some algorithms for the calculation of scale factors. No systematic variations are evaluated in this initial test, which can be disadvantageous for the CP algorithm implementation since it executes some infrastructure to address systematic variations by default. This includes, for example, the `SysListLoaderAlg`, which prepares a list of all applicable systematic variations.

Table 11.1 shows the average wall times and event rates for the n-tuple production of 10000 $t\bar{t}$ events in both setups. As expected, the CP algorithm implementation is slightly slower than the reference in this setup, which amounts to a ~11% longer execution time and a ~10% lower event processing rate.

Table 11.1: Wall time and event rate of processing 10000 $t\bar{t}$ events with the CP algorithm and the reference implementation of the RPV1L n-tuple producton code. The average and standard deviation of 10 repetitions is shown. No optimizations are applied.

	Wall time (s)	Event rate (Hz)
Reference	229.70 ± 2.72	43.44 ± 0.52
CP algorithms	255.30 ± 3.29	39.08 ± 0.50

11.1.1 Performance Snapshot

Intel® VTune™ provides a *performance snapshot* profile with the intention to guide the user through detailed profiling types. Figure 11.1 contains a screenshot of the resulting recommendations for the reference. The recommendations for the CP algorithm implementation are almost identical, except for a reported 37.3% (vs. 37.7%) of retired instructions[3] at the *microarchitecture exploration* stage, which is likely within the uncertainty of event sampling based profiling.

This summary emphasizes three points. First, a *hotspot* analysis should be run to isolate single sections of the code responsible for a large part of the total runtime. These hotspots are typically promising candidates for optimization. The second suggestion is the *threading* profile to measure utilization of all available CPU cores. This profiling type is not very relevant for this study since the underlying ATLAS analysis release is not intended to provide multithreading capabilities. The last profiling type, *microarchitecture exploration*, points out issues with micro-operation dispatching, branch prediction and port utilization. This is a common occurrence in ATLAS event processing code since event-by-event processing is hard to predict by nature [Err16]. Each subsequent event is physically independent. Additionally, many C++ classes are instantiated to treat the possibly occurring physics objects in a given event, which results in unlocalized code that is hard to be cached in the instruction cache of the CPU.

11.1.2 Hotspot Analysis

The hotspot profile reveals that a significant proportion of the n-tuple production process is caused by calls to various `isAffectedBySystematic` methods of multiple CP tools in both versions. Figure 11.2 shows the six most time-consuming function calls, including a list of callers for the first two.

[3] *Retired instructions* are all instructions the CPU fully executes and that are actually required to execute the current program [Corb].

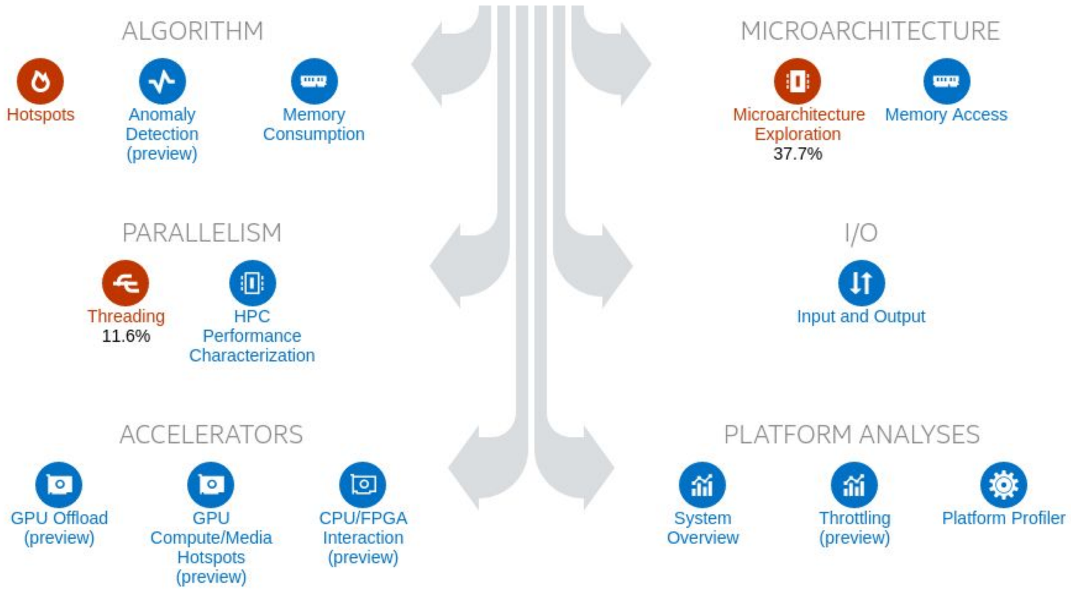


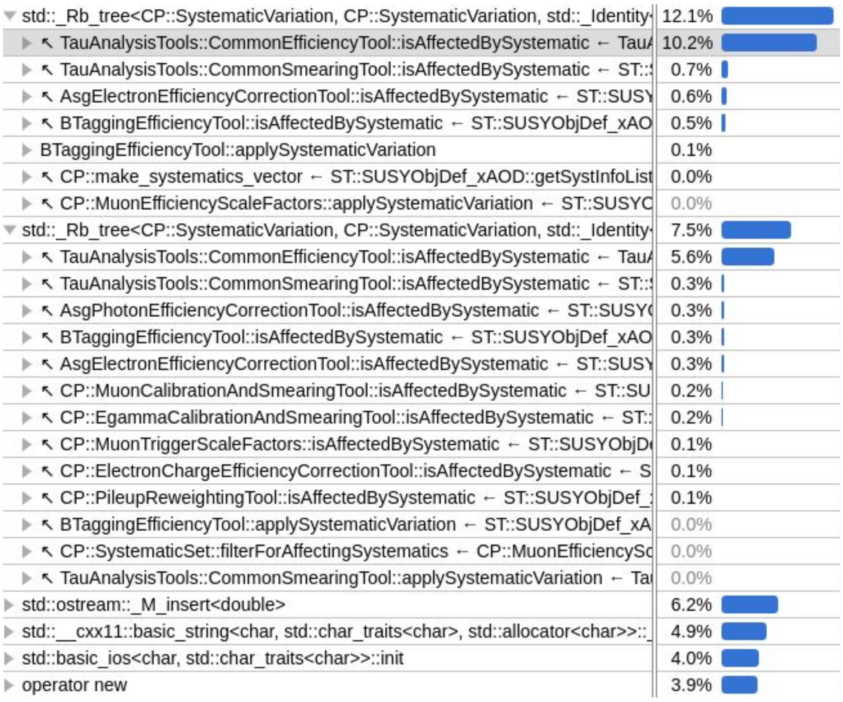
Figure 11.1: Recommendations of the performance snapshot profile for the reference n-tuple production, recommending a hotspot, threading and microarchitecture profile. The result of the CP algorithm version is almost identical, except for 37.3% retired instructions at the microarchitecture explaination stage.

Table 11.2: Wall time and event rate with caching the systematics list in `GetScaleFactors`. The average and standard deviation of 10 repetitions is given.

	Wall time (s)	Event rate (Hz)
Reference	72.00 ± 0.63	137.99 ± 1.02
CP algorithms	99.10 ± 2.39	100.33 ± 2.33

All of these function calls are ultimately caused by `getSystInfoList()`, which assembles a list of all available and relevant systematic variations for the current configuration. This function is used in the stop1l-xaod algorithm `GetScaleFactors`, which calculates and stores all relevant scale factors, potentially dependent on systematic uncertainties. The algorithm calls `getSystInfoList()` during initialization, but also for every event during the event loop. The list of relevant systematic variations does not change for each event and can be cached during initialization instead.

Table 11.2 shows the total run time and event rate after implementing a fix for the `GetScaleFactors` hotspot. A comparison to Table 11.1 shows that this simple fix improves the reference by a factor of ~3 and the CP algorithm implementation by a factor of ~2.5. This optimization only affects analyses using the stop1l-xaod framework



```
▼ std::_Rb_tree<CP::SystematicVariation, CP::SystematicVariation, std::_Identity   12.1%
  ▶ ↖ TauAnalysisTools::CommonEfficiencyTool::isAffectedBySystematic ← TauA        10.2%
  ▶ ↖ TauAnalysisTools::CommonSmearingTool::isAffectedBySystematic ← ST::S          0.7%
  ▶ ↖ AsgElectronEfficiencyCorrectionTool::isAffectedBySystematic ← ST::SUSY        0.6%
  ▶ ↖ BTaggingEfficiencyTool::isAffectedBySystematic ← ST::SUSYObjDef_xAO           0.5%
  ▶ BTaggingEfficiencyTool::applySystematicVariation                                0.1%
  ▶ ↖ CP::make_systematics_vector ← ST::SUSYObjDef_xAOD::getSystInfoList            0.0%
  ▶ ↖ CP::MuonEfficiencyScaleFactors::applySystematicVariation ← ST::SUSYC          0.0%
▼ std::_Rb_tree<CP::SystematicVariation, CP::SystematicVariation, std::_Identity    7.5%
  ▶ ↖ TauAnalysisTools::CommonEfficiencyTool::isAffectedBySystematic ← TauA         5.6%
  ▶ ↖ TauAnalysisTools::CommonSmearingTool::isAffectedBySystematic ← ST::S          0.3%
  ▶ ↖ AsgPhotonEfficiencyCorrectionTool::isAffectedBySystematic ← ST::SUSY(         0.3%
  ▶ ↖ BTaggingEfficiencyTool::isAffectedBySystematic ← ST::SUSYObjDef_xAO           0.3%
  ▶ ↖ AsgElectronEfficiencyCorrectionTool::isAffectedBySystematic ← ST::SUSY        0.3%
  ▶ ↖ CP::MuonCalibrationAndSmearingTool::isAffectedBySystematic ← ST::SU           0.2%
  ▶ ↖ CP::EgammaCalibrationAndSmearingTool::isAffectedBySystematic ← ST::           0.2%
  ▶ ↖ CP::MuonTriggerScaleFactors::isAffectedBySystematic ← ST::SUSYObjD            0.1%
  ▶ ↖ CP::ElectronChargeEfficiencyCorrectionTool::isAffectedBySystematic ← S        0.1%
  ▶ ↖ CP::PileupReweightingTool::isAffectedBySystematic ← ST::SUSYObjDef_           0.1%
  ▶ ↖ BTaggingEfficiencyTool::applySystematicVariation ← ST::SUSYObjDef_xA          0.0%
  ▶ ↖ CP::SystematicSet::filterForAffectingSystematics ← CP::MuonEfficiencySc       0.0%
  ▶ ↖ TauAnalysisTools::CommonSmearingTool::applySystematicVariation ← Ta           0.0%
▶ std::ostream::_M_insert<double>                                                   6.2%
▶ std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>>::_ 4.9%
▶ std::basic_ios<char, std::char_traits<char>>::init                                4.0%
▶ operator new                                                                      3.9%
```

Figure 11.2: Most time-consuming function calls as the fraction of the total time in the CP algorithm implementation. Many functions are called by `isAffectedBySystematic` methods of various CP tools, which happens in both, the CP algorithm and the reference implementation.

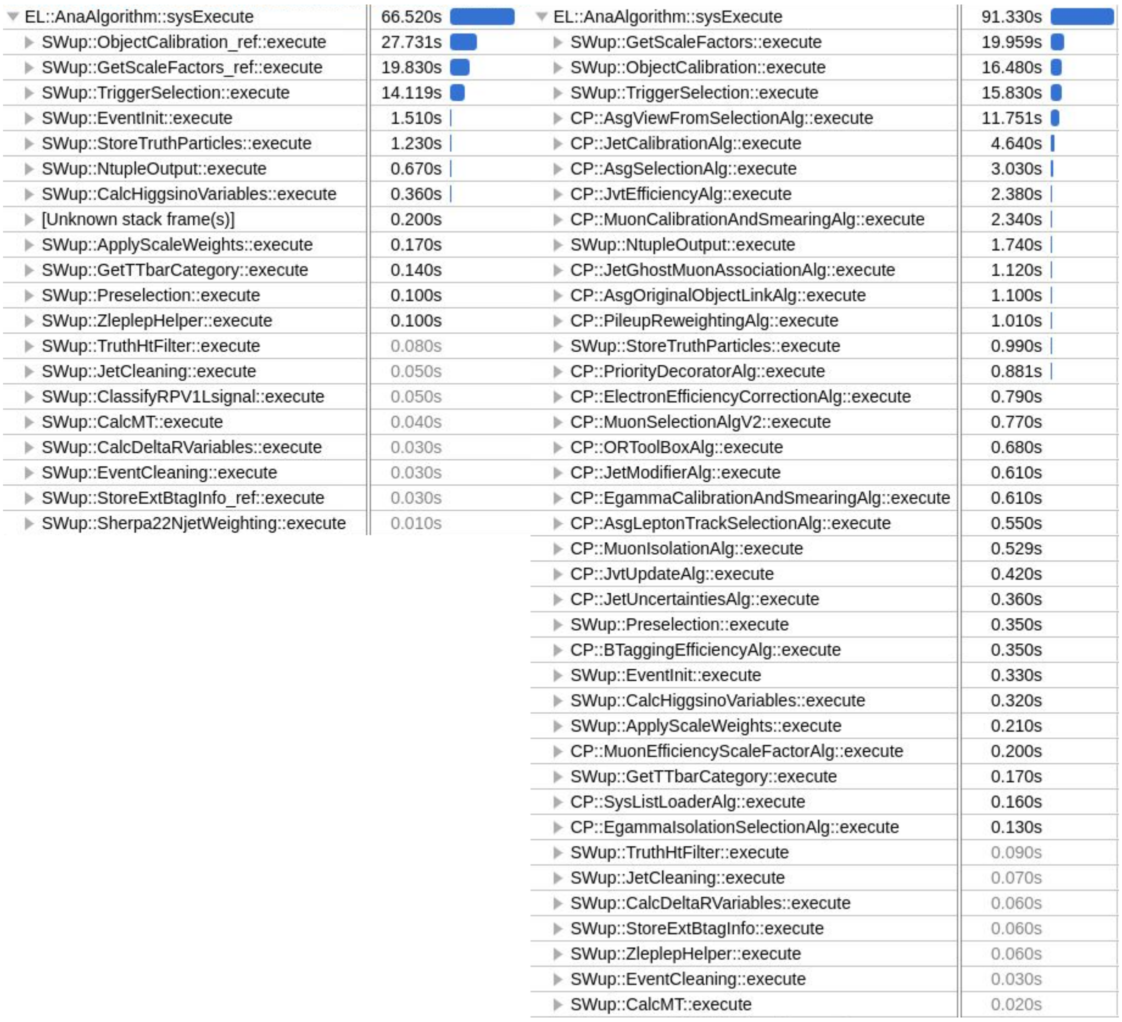
since the original issue is the inefficient usage of a method provided by SUSYTools.

Including this optimization, the difference between the reference and the CP algorithm version becomes more pronounced. The event rate of the CP algorithm implementation is ~27% lower compared to the reference.

11.1.3 Hotspot Analysis with GetScaleFactors Fix

A second hotspot analysis is done to assess the differences between the reference and the CP algorithm implementation after the first hotspot is resolved. Figure 11.3 shows a top-down call graph showing the `execute` methods of all involved algorithms, which are called on every event in the file. It shows a difference between both setups that is not localized in a single location of the program.

In the reference (left), the algorithms `ObjectCalibration`, `GetScaleFactors` and `TriggerSelection` account for ~93% of the run time spent in all `execute` methods. These algorithms implement the object selection, compute scale factors and check if given objects pass certain triggers of interest.



EL::AnaAlgorithm::sysExecute	66.520s
SWup::ObjectCalibration_ref::execute	27.731s
SWup::GetScaleFactors_ref::execute	19.830s
SWup::TriggerSelection::execute	14.119s
SWup::EventInit::execute	1.510s
SWup::StoreTruthParticles::execute	1.230s
SWup::NtupleOutput::execute	0.670s
SWup::CalcHiggsinoVariables::execute	0.360s
[Unknown stack frame(s)]	0.200s
SWup::ApplyScaleWeights::execute	0.170s
SWup::GetTTbarCategory::execute	0.140s
SWup::Preselection::execute	0.100s
SWup::ZleplepHelper::execute	0.100s
SWup::TruthHtFilter::execute	0.080s
SWup::JetCleaning::execute	0.050s
SWup::ClassifyRPV1Lsignal::execute	0.050s
SWup::CalcMT::execute	0.040s
SWup::CalcDeltaRVariables::execute	0.030s
SWup::EventCleaning::execute	0.030s
SWup::StoreExtBtagInfo_ref::execute	0.030s
SWup::Sherpa22NjetWeighting::execute	0.010s

EL::AnaAlgorithm::sysExecute	91.330s
SWup::GetScaleFactors::execute	19.959s
SWup::ObjectCalibration::execute	16.480s
SWup::TriggerSelection::execute	15.830s
CP::AsgViewFromSelectionAlg::execute	11.751s
CP::JetCalibrationAlg::execute	4.640s
CP::AsgSelectionAlg::execute	3.030s
CP::JvtEfficiencyAlg::execute	2.380s
CP::MuonCalibrationAndSmearingAlg::execute	2.340s
SWup::NtupleOutput::execute	1.740s
CP::JetGhostMuonAssociationAlg::execute	1.120s
CP::AsgOriginalObjectLinkAlg::execute	1.100s
CP::PileupReweightingAlg::execute	1.010s
SWup::StoreTruthParticles::execute	0.990s
CP::PriorityDecoratorAlg::execute	0.881s
CP::ElectronEfficiencyCorrectionAlg::execute	0.790s
CP::MuonSelectionAlgV2::execute	0.770s
CP::ORToolBoxAlg::execute	0.680s
CP::JetModifierAlg::execute	0.610s
CP::EgammaCalibrationAndSmearingAlg::execute	0.610s
CP::AsgLeptonTrackSelectionAlg::execute	0.550s
CP::MuonIsolationAlg::execute	0.529s
CP::JvtUpdateAlg::execute	0.420s
CP::JetUncertaintiesAlg::execute	0.360s
SWup::Preselection::execute	0.350s
CP::BTaggingEfficiencyAlg::execute	0.350s
SWup::EventInit::execute	0.330s
SWup::CalcHiggsinoVariables::execute	0.320s
SWup::ApplyScaleWeights::execute	0.210s
CP::MuonEfficiencyScaleFactorAlg::execute	0.200s
SWup::GetTTbarCategory::execute	0.170s
CP::SysListLoaderAlg::execute	0.160s
CP::EgammaIsolationSelectionAlg::execute	0.130s
SWup::TruthHtFilter::execute	0.090s
SWup::JetCleaning::execute	0.070s
SWup::CalcDeltaRVariables::execute	0.060s
SWup::StoreExtBtagInfo::execute	0.060s
SWup::ZleplepHelper::execute	0.060s
SWup::EventCleaning::execute	0.030s
SWup::CalcMT::execute	0.020s

Figure 11.3: Time spent in individual algorithms in the reference (left) and the CP algorithm implementation (right).

For the CP algorithm implementation, these three algorithms only account for ~57% of the total time and the workload appears to be distributed. The absolute time spent in ObjectCalibration is lower, compared to the reference, since physics definitions are now executed in the corresponding CP algorithms. For example, the jet calibration, is extracted from ObjectCalibration into the JetCalibration algorithm.

Jet calibration takes up ~4 s in the CP algorithm implementation, but the reference profile shows only ~1.7 s within the object calibration algorithm of the reference. These absolute time measurements can vary between profiles, but this difference is large enough to suggest a difference in the configuration. This is likely caused by differences in the preselection before jet calibration. In the GetJets and FillJet methods of the reference, a set of jet property checks are executed before jet calibration is applied. For the CP algorithm setup, fewer preselection cuts are applied, resulting in more jets being calibrated, only to be filtered by baseline and signal requirements later. A similar difference also occurs in the case of muon calibration.

In both versions of the ObjectCalibration, a significant part of the time in the event loop is spent in trigger matching methods that check if a given particle activated a specific trigger.

Time spent in GetScaleFactors and TriggerSelection does not change significantly since both setups perform the same operations in these algorithms. The runtime of GetScaleFactors mostly consists of calculations of JVT, electron and muon scale factors. For TriggerSelection, most of its self-time comes from finding relevant triggers that passed for a given event.

A fourth time-consuming section, only occurring in the CP algorithm case, is the AsgViewFromSelectionAlg. The overhead originates from making deep copies of calibrated electron, muon and jet containers in the current test. This is an inefficiency in the configuration of the corresponding CP algorithm sequences of this test and not of the default sequences.

In summary, there are opportunities for performance optimizations in the reference and the CP algorithm implementation of the n-tuple production code. Additionally, there is a measurable difference between the CP algorithm implementation and the reference, which is caused by the current configuration of the test.

Table 11.3: Average and maximum event rates of 10 repetitions of processing 10000 $t\bar{t}$ events in the reference and the CP algorithm implementation with and without doing deep copies of the output electron and muon containers. In this test, the `GetScaleFactor` algorithm is configured to not evaluate any weight-affecting variations, resulting in higher event rates than shown in Table 11.2.

	Mean event rate (Hz)	Max. event rate (Hz)
Reference	190.66 ± 2.25	195.13
CP algorithms with deep copies	115.58 ± 4.34	120.97
CP algorithms without deep copies	130.52 ± 1.58	133.49

11.2 Deep Copies of Containers

The version of the CP algorithm sequences that is profiled in Section 11.1 creates deep copies of the `AnalysisElectrons` and `-Muons` containers. Deep copies ensure that the resulting container is independent and complete, i.e. there are no links to a separate container in the event store. With this operation, the `AsgViewFromSelectionAlg` is a dominant component in the VTune profiles. There is no benefit of doing deep copies at this point since the objects are read from memory in the stop1l-xaod framework. The decision of enabling or disabling deep copies is typically located at the end of CP algorithm sequences.

Table 11.3 shows the event rates of 10 repetitions of processing 10000 $t\bar{t}$ events in the reference and the CP algorithm setup with and without deep copies. By avoiding deep copies, the event rate in the CP algorithm setup improves from 115.59 ± 4.34 Hz to 130.52 ± 1.58 Hz. This observation motivates an efficient treatment of intermediate containers by avoiding deep copies when the containers are solely processed in memory.

11.3 Preselection in CP Algorithms

Profiling the complete implementation of CP algorithms with weight-affecting systematic variations reveals that a large fraction of time is spent in different functions as the reference. 12.7% of the time is spent in the `__popcountdi2` builtin method, counting bits in a bit set. This is followed by 6.2% in the `CxxUtils::ConcurrentBitset::Impl::count`, which is also the callee of the above population count. In the reference, the most dominant function call is the `new` operator with 7.3%, which is 5.4% in the CP algorithm version.

The stack trace of both population count methods shows that they are caused by retrieving the data or resolving the original object links from shallow copied containers. As described in Section 10.2.5, each systematic variation introduces new lightweight copies of the processed object containers in CP algorithms. The fact that these operations appear in the profile suggests that individual calls are cheap, but they scale with the number of intermediate containers and systematic variations.

One minor cause is identified as the preselection condition on many CP algorithms. For example, the muon sequence manages a list of selection decorations, e.g. p_T and η requirements, and algorithms can be configured such that they are only executed on objects where all preselection requirements are met. Each requirement in the preselection list involves a read access to the corresponding container to retrieve the status of the flag for the current event. Since these requirements often accumulate and imply each other, e.g. by only checking the η requirements on events where the p_T requirements are already met, it is sufficient to only use the latest preselection flag.

In a simple test, the electron and muon sequences are modified to only access the last preselection since these sequences use this mechanism considerably. With this reduction of preselection evaluations, the VTune profile reports 10.9% of the runtime coming from `__popcountdi2` and 5.5% from `count`. Therefore, this represents a slight improvement that is easy to implement and should be considered best-practice whenever preselections are used accumulatively.

Table 11.4 shows the event rates while checking either all or just the last preselection flag. The process that evaluates weight-affecting systematic variations improves through this change since each preselection is checked for every systematic variation. The process that only runs over a single kinematics-affecting variation is not measurably improving by this approach since this behaviour scales with the number of evaluated systematic variations

The remaining calls to the population count methods are still responsible for a significant proportion of the job execution if many systematic variations are evaluated. This is due to distributed access to various systematically varied containers, which is a fundamental design decision for CP algorithms. There is no design improvement to solve this problem other than reducing the number of CP algorithms and their operations to a minimum.

The machine instruction `popcnt` is available on all computer architectures that support the SSE 4.2 instruction set [Cor20]. Recompiling the `ConcurrentBitset` class with the `gcc` option `-mpopcnt`[4], improves the performance of this code sec-

[4] Included in `-march=native`, but used exclusively here.

Table 11.4: Average and maximum event rates of 10 repetitions of processing 10000 $t\bar{t}$ events while either checking all or the last preselection flags in the electron and muon sequences. "CP algorithms (weights)" refers to the nominal execution, where also multiple weight-affecting systematic variations are evaluated. "CP algorithms (kinematics)" refers to a job where only a single kinematics-affecting systematic variation is evaluated.

	Mean event rate (Hz)	Max. event rate (Hz)
CP algorithms (weights)	43.34 ± 0.28	43.63
CP algorithms (weights) one presel.	44.78 ± 0.37	45.31
CP algorithms (kinematic)	113.69 ± 1.85	116.73
CP algorithms (kinematic) one presel.	114.12 ± 1.02	116.34

tion. The average event rate of processing 10000 $t\bar{t}$ events increases to $50.21 \pm 0.64\,Hz$, which is a ~11% improvement, matching the previously reported 10.9% spent in the builtin population count. The largest remaining hotspot in a new VTune profile is the allocation operator with 6.6%, similar to the reference.

ATLAS software is usually compiled with very few assumptions about the underlying architecture because of unpredictable computing environments in the Collaboration. For this reason, the `-mpopcnt` option is not used in the following studies.

11.4 Systematic Variations

As already mentioned in Section 10.2.5, the two approaches to kinematics-affecting and weight-affecting systematic variations could introduce differences in the computing performance between the reference and the CP algorithm setup. Various tests are performed to evaluate these differences in both setups. All tests include the previously discussed optimizations.

11.4.1 Scaling

Figure 11.4 shows the event rate scaling of the CP algorithm setup with respect to the number of processed variations. A list of 36 weight-affecting and 14 kinematics-affecting systematic variations is passed to the CP algorithm sequences, and each container is also processed in the `ObjectCalibration` or `GetScaleFactor` algorithms of the CP algorithm implementation. The event rate is reduced with an increasing number of variations due to the workload of preparing a systematically varied container for each variation. Each CP algorithm that modifies a systematically varied

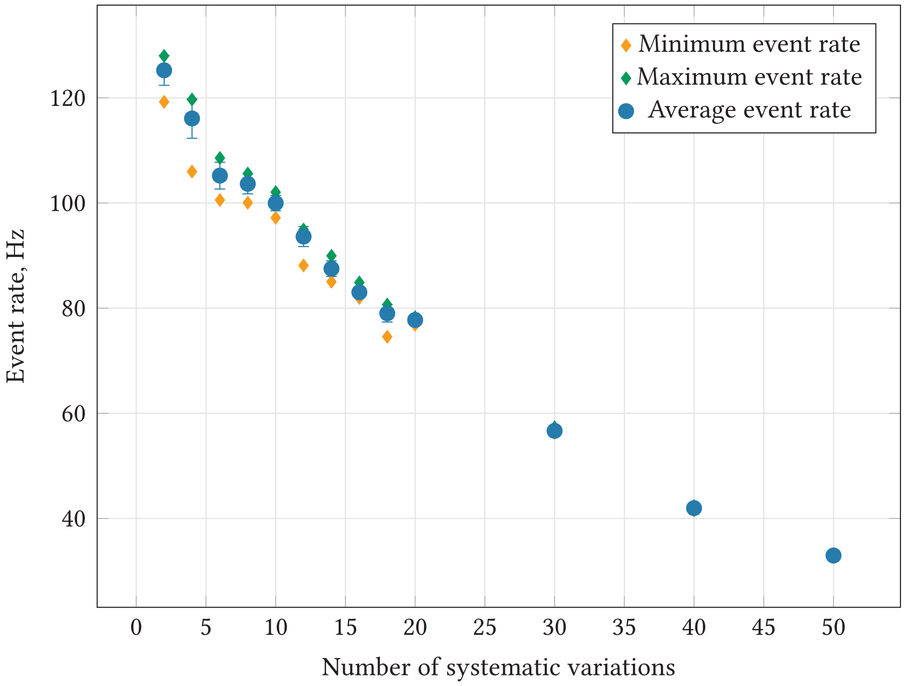


Figure 11.4: Event rate of processing 10000 $t\bar{t}$ events in the CP algorithm setup with an increasing number of systematic variations. Each test is repeated 10 times, showing the mean, standard deviation, and minimum/maximum.

container creates a new shallow copy of it. The slowdown of the event rate is unexpected for the first 36 variations since no systematically varied container should be created and only systematically varied decorations to the nominal containers are applied.

11.4.2 Weight-Affecting Variations

In the previous test it is found that processing the nominal containers and additionally calculating systematically varied scale factors is slower in CP algorithms than in the reference. This difference is also visible in Table 11.4, which amounts to a 2 to 3 times lower event rate. To further test this, 36 weight-affecting systematic variations (e.g. `JET_JvtEfficiency__1down` and `1up`) are evaluated in addition to the nominal containers. Table 11.5 shows the event rates of the CP algorithm and the reference setup, only including the evaluation of weight-affecting variations. The CP algorithm implementation is ~3 times slower in this case. A third measurement is performed, where weight-affecting variations are not evaluated in `GetScaleFactors`, which shows that the overhead is located in the scheduled CP algorithms.

The overhead is unexpected since only weight-affecting systematic variations are

Table 11.5: Average and maximum event rates of 10 repetitions of processing 10000 $t\bar{t}$ events with only weight-affecting variations in addition to the nominal containers. Turning off the evaluation of weight-affecting variations in the `GetScaleFactors` algorithm is not affecting the event rate by much. This shows that the overhead is mostly caused by operations in the CP algorithm sequence. Forcing all algorithms that create scale factor decorations to only consider the nominal container increases the event rate by a factor of ~2. Avoiding the output property in scale factor calculating algorithms removes the overhead from unnecessary particle container copies.

	Mean event rate (Hz)	Max. event rate (Hz)
CPA	45.12 ± 0.26	45.42
CPA (w/o `GetScaleFactor`)	44.99 ± 0.89	45.97
CPA (forced to NOSYS)	91.53 ± 1.68	93.69
CPA (no output properties)	110.70 ± 1.71	113.47
Reference	138.87 ± 1.01	140.36

considered, which mostly affect scale factor decorations. CP algorithms should be able to automatically optimize this situation such that no unnecessary copies of systematically varied containers are created, and only the nominal containers are decorated with systematically varied scale factors.

In a separate test, the automatic system of handling systematic variations is manually disabled for the corresponding scale factor algorithms, which in turn are forced only to evaluate the weight-affecting variations for decorations to the nominal container (NOSYS). In this proof-of-concept approach, the event rate roughly doubles from $45.12 \pm 0.26\,Hz$ to $91.53 \pm 1.68\,Hz$. This is an extreme and non-standard optimization, strictly assuming that no weight-affecting variations would ever be evaluated on top of kinematics-affecting systematic variations. Such manual configurations should be avoided in regular analyses.

The final cause of this inefficiency is found to be a subtle misconfiguration in the test setup. Each scale factor calculating algorithm had a defined output property name for the corresponding particle container. With this, the system automatically assumed a systematically varied particle container and therefore created an unnecessary copy. Removing this property from all scale factor algorithms results in the intended behaviour and an average event rate of $110.70 \pm 1.71\,Hz$. The centrally provided default sequences do not contain this misconfiguration. Figure 11.5 shows the updated event rate scaling with 36 weight-affecting and 14 kinematics-affecting systematic variations. Each weight-affecting variation introduces little overhead, and kinematics-affecting variations show an expected slowdown.

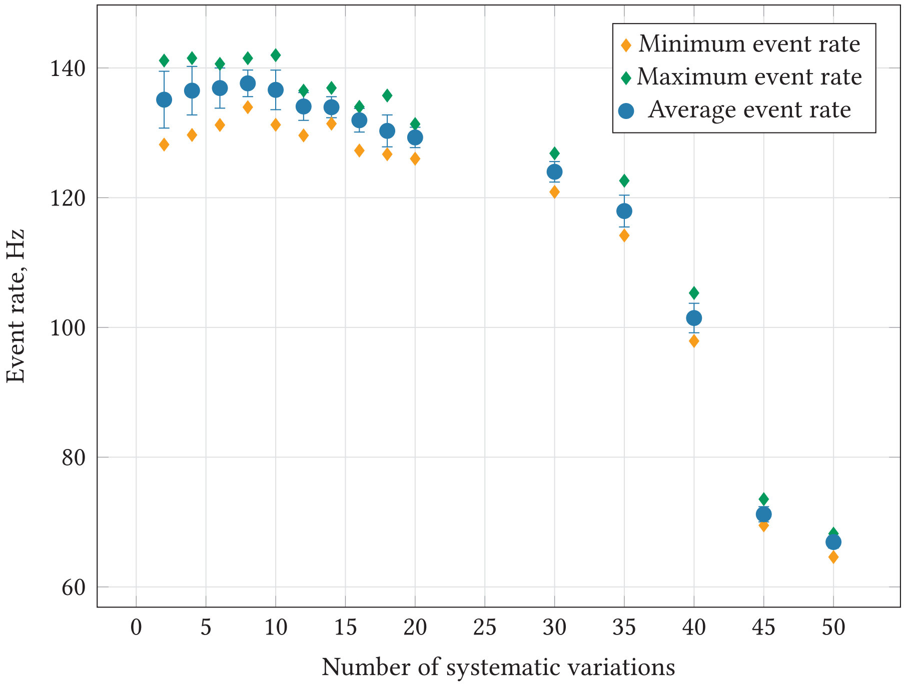


Figure 11.5: Event rate scaling of processing 36 weight-affecting and 14 kinematics-affecting systematic variations (ordered) 10 times in the CP algorithm implementation. Each weight-affecting variations adds little overhead, if no output container of the underlying particle type is configured in all scale factor algorithms.

11.4.3 Kinematics-Affecting Variations

Processing a single kinematic-affecting systematic variation could also be slightly slower in the CP algorithm setup since the nominal containers are always required to serve as a default fall-back if a given CP algorithm is not affected by the variation. The average event rate in this case is $115.28 \pm 1.02\,Hz$ for the CP algorithm setup and $115.37 \pm 1.72\,Hz$ for the reference implementation. In this one-to-one comparison, both event rates are within the uncertainties, which shows that the CP algorithm approach can be as efficient as the reference.

The strength of the CP algorithm approach is observable if multiple kinematics-affecting systematic variations are evaluated in one job. In this case, the reference implementation requires individual jobs for each variation. A comparison is done by executing the nominal job, including weight-variations and 14 additional kinematics-affecting systematic variations. The results are shown in Table 11.6, where the CP algorithm approach is up to ~8 times faster. This comparison is biased by not storing systematically varied containers in the n-tuple in the CP algorithm version.

Processing all selected systematic variations at once in a CP algorithm job can be significantly faster than re-executing the whole analysis for each variation. A resulting best-practice recommendation could be to omit systematic variations in CP algorithms during development and schedule an n-tuple production with all systematic variations at once when the analysis is matured. This recommendation depends on how efficiently the used analysis framework treats weight- and kinematics-affecting systematic variations.

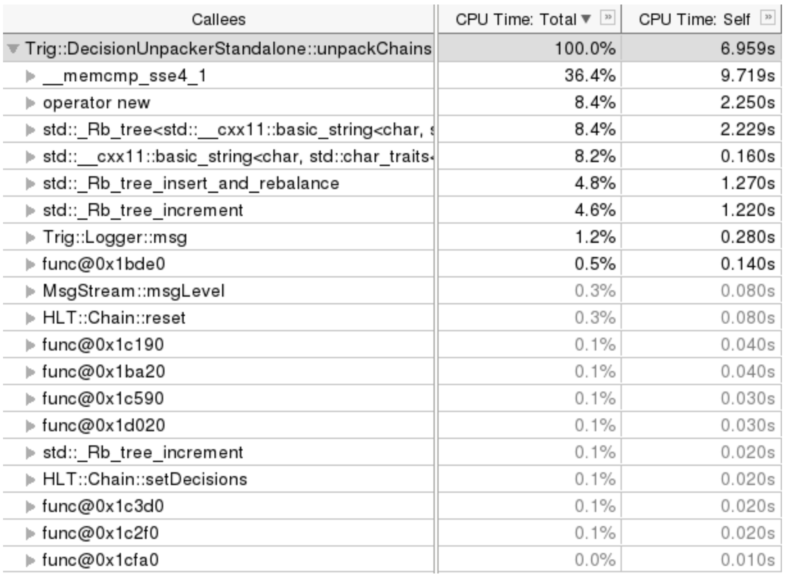
It is also recommended to minimize the number of CP algorithms that need to re-run if their input containers are affected by a systematic variation. For example, the overlap removal sequence needs to re-run for any systematic variation that affects its input objects. Any subsequent algorithm that works on the output of the overlap removal sequence is evaluated as well. Therefore, ensuring that most processing is done on the individual object-sequences before overlap removal is most efficient.

11.5 TrigDecisionTool Improvement

A profile of an unrelated analysis revealed an inefficiency in the `TrigDecisionTool`, which is also used in the RPV1L analysis. This tool provides the `isPassed()` method, which determines if a given event passes a specific trigger. The profiled analysis spends roughly 40.7% of all clock cycles in the `isPassed()` method and more

Table 11.6: Average event rate and total time of 10 repeated executions. The CP algorithm version is executing all weight- and kinematics-affecting variations. The reference implementation is split into nominal and weight-affecting variations, and individual executions for single kinematics-affecting systematic variations. The total time of all reference jobs is summed to be comparable to the CP algorithm execution.

	EventRate (Hz)	Total time (s)
CP algorithms	65.96 ± 1.11	151.00 ± 2.72
REF (nominal + weight variations)	131.33 ± 4.97	75.80 ± 3.06
REF (`EG_RESOLUTION_ALL__1down`)	119.42 ± 2.18	83.10 ± 1.51
REF (`EG_RESOLUTION_ALL__1up`)	117.89 ± 2.89	84.30 ± 2.33
REF (`JET_GroupedNP_1__1down`)	110.67 ± 3.60	89.90 ± 2.98
REF (`JET_GroupedNP_1__1up`)	111.94 ± 2.08	88.90 ± 1.76
REF (`JET_GroupedNP_2__1down`)	111.66 ± 3.25	89.10 ± 2.59
REF (`JET_GroupedNP_2__1up`)	111.74 ± 2.10	89.10 ± 1.58
REF (`JET_GroupedNP_3__1down`)	112.00 ± 3.18	88.90 ± 2.70
REF (`JET_GroupedNP_3__1up`)	114.93 ± 2.70	86.60 ± 2.20
REF (`MET_SoftTrk_ResoPara`)	125.36 ± 3.13	79.50 ± 1.86
REF (`MET_SoftTrk_ResoPerp`)	125.80 ± 1.75	79.00 ± 1.18
REF (`MET_SoftTrk_ScaleDown`)	126.62 ± 2.02	78.50 ± 1.36
REF (`MET_SoftTrk_ScaleUp`)	126.76 ± 2.32	78.40 ± 1.50
REF (`MUON_ID__1down`)	116.36 ± 1.24	85.50 ± 0.81
REF (`MUON_ID__1up`)	114.94 ± 2.88	86.50 ± 2.42
REF (sum)	-	1263.10 ± 8.12



Callees	CPU Time: Total ▼	CPU Time: Self
▼ Trig::DecisionUnpackerStandalone::unpackChains	100.0%	6.959s
▶ __memcmp_sse4_1	36.4%	9.719s
▶ operator new	8.4%	2.250s
▶ std::_Rb_tree<std::__cxx11::basic_string<char, s	8.4%	2.229s
▶ std::__cxx11::basic_string<char, std::char_traits<	8.2%	0.160s
▶ std::_Rb_tree_insert_and_rebalance	4.8%	1.270s
▶ std::_Rb_tree_increment	4.6%	1.220s
▶ Trig::Logger::msg	1.2%	0.280s
▶ func@0x1bde0	0.5%	0.140s
▶ MsgStream::msgLevel	0.3%	0.080s
▶ HLT::Chain::reset	0.3%	0.080s
▶ func@0x1c190	0.1%	0.040s
▶ func@0x1ba20	0.1%	0.040s
▶ func@0x1c590	0.1%	0.030s
▶ func@0x1d020	0.1%	0.030s
▶ std::_Rb_tree_increment	0.1%	0.020s
▶ HLT::Chain::setDecisions	0.1%	0.020s
▶ func@0x1c3d0	0.1%	0.020s
▶ func@0x1c2f0	0.1%	0.020s
▶ func@0x1cfa0	0.0%	0.010s

Figure 11.6: VTune profile of what portion of the `unpackChains()` method is spent in which operation.

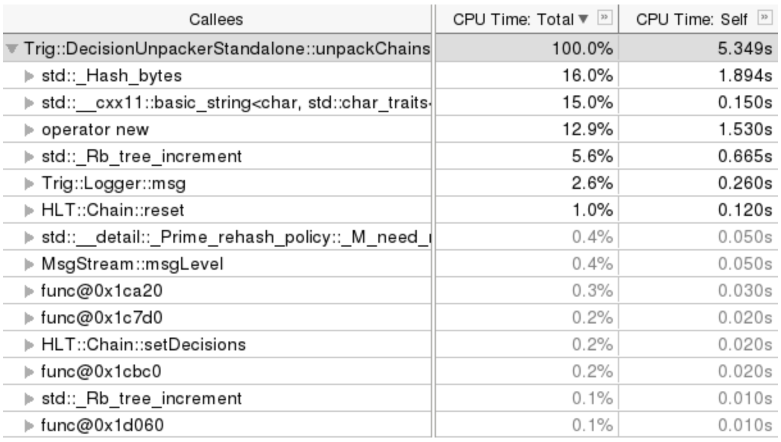
specifically in the call to `unpackChains()`, which makes it a promising hotspot to investigate.

Since there are up to 3500 triggers to test for, some analyses call this method multiple times for each event. The code section causing the inefficiency is:

```
for( auto& cntrchain : cache ) {
  unsigned cntr = cntrchain.first;
  cntrchain.second->reset();
  cntrchain.second->setDecisions(
              get32BitDecision( cntr, raw ),
              get32BitDecision( cntr, passedthrough ),
              get32BitDecision( cntr, prescaled),
              get32BitDecision( cntr, resurrected ) );
  output[ cntrchain.second->getChainName() ] = cntrchain.second; // <<< Expensive line
}
```

This section stores a new key-value pair in the `std::map output` container, which is cleared before the for-loop. Figure 11.6 shows a breakdown of how much time is spent in subsequent function calls.

A lot of time is spent comparing values, which is due to the fact that `std::map` is an ordered container. Replacing it with a `std::unordered_map` is a simple improvement. The same profiling information for the improved version is shown in Figure 11.7, proving that the `std::unordered_map` does not do the expensive key comparison anymore. Now only 24.4% of the total clock cycle count is spent in the `isPassed()` method. The total wall time is reduced from 72.8 ± 0.4 s to 55.5 ± 0.8 s, which corresponds to a reduction of roughly 24%.



Callees	CPU Time: Total ▼	CPU Time: Self
▼ Trig::DecisionUnpackerStandalone::unpackChains	100.0%	5.349s
▶ std::_Hash_bytes	16.0%	1.894s
▶ std::__cxx11::basic_string<char, std::char_traits	15.0%	0.150s
▶ operator new	12.9%	1.530s
▶ std::_Rb_tree_increment	5.6%	0.665s
▶ Trig::Logger::msg	2.6%	0.260s
▶ HLT::Chain::reset	1.0%	0.120s
▶ std::__detail::_Prime_rehash_policy::_M_need_	0.4%	0.050s
▶ MsgStream::msgLevel	0.4%	0.050s
▶ func@0x1ca20	0.3%	0.030s
▶ func@0x1c7d0	0.2%	0.020s
▶ HLT::Chain::setDecisions	0.2%	0.020s
▶ func@0x1cbc0	0.2%	0.020s
▶ std::_Rb_tree_increment	0.1%	0.010s
▶ func@0x1d060	0.1%	0.010s

Figure 11.7: VTune profile of `unpackChains()` after implementing the fix.

Table 11.7: Walltime for number of calls to `isPassed()` in seconds for different underlying containers in the demonstrator.

Version	1	500	1000	2000	3408 (all)
`std::map`	16	21	25	33	46
`std::unordered_map`	11	13	16	23	34
`std::vector`	9	11	17	32	58

A demonstrator, where only `isPassed()` is called for each event a variable number of times, was developed[5]. It compares three implementations of the above-mentioned code. One with the default `std::map`, a `std::unordered_map` and a `std::vector<std::pair<...>>`. In Table 11.7 the execution times of this demonstrator are shown for each version and a variable number of triggers.

The original implementation performs worse in every case. The `std::vector` implementation can be even faster for fewer checked triggers, likely due to better cache alignment. Since some analysis workflows require many trigger checks and since the `std::vector` implementation is less clear to read, the `std::unordered_map` implementation is the recommended solution to this issue. The fix has been merged into the upstream repository and went into the analysis release 21.2.40.

11.6 Summary of the Performance Analysis

The performance analysis of the CP algorithm implementation and a reference version of the RPV1L n-tuple production highlights multiple ways to optimize the event throughput. An initial profile revealed the unnecessary creation of a list of relevant

[5] gitlab.cern.ch/merrenst/trigdecsdemonstrator

systematic variations in the `GetScaleFactor` algorithm for each event, affecting both versions of the analysis. By creating the list once during the algorithms initialization, the performance of this particular analysis can be improved by a factor of 2.5 to 3. A technical optimization of the `TrigDecisionTool` is implemented and potentially improving all future analyses in ATLAS.

The modular and flexible design of CP algorithms is found to introduce some overhead in certain configurations. Creating deep copies of intermediate containers should be avoided whenever possible. The remaining overhead can be attributed to shallow copies of intermediate containers that are required to implement the modular design of CP algorithm sequences. A minor improvement is possible by reducing access to these containers through the optimization of preselection conditions. If SSE 4.2 instructions are available, the overhead of using shallow copies could be further reduced by 10% by compiling the underlying C++ libraries with the compiler flag `-mpopcnt`. The availability of SSE 4.2 is usually not assumed since ATLAS software is used in many different computing environments.

CP algorithms automatically manage the execution of multiple systematic variations. If only weight-affecting systematic variations are evaluated on top of the nominal execution, it is recommended to ensure that scale factor calculating algorithms do not define an output property for the corresponding particle container. Otherwise, the CP algorithm implementation can be significantly slower by creating unnecessary container copies, e.g. resulting in a ~3 times slower execution of this particular analysis. However, the automatic management of systematic variations is found to work very well when multiple kinematics-affecting variations are evaluated. The CP algorithm implementation is ~8 times faster in this particular test by evaluating all variations in a single execution, as opposed to the reference where individual processes need to be configured for each kinematics-affecting variation.

Part IV
Conclusion

The ATLAS collaboration has a large and diverse analysis community, also resulting in a diverse analysis software landscape. Previous and ongoing harmonization efforts of the analysis software and workflows aim to reduce the amount of duplicated work between analysis software developers and ensure efficient resource usage. A common basis for physics object preparation was recently introduced with the CP algorithms for Run 3 analyses. CP algorithms wrap the widely used CP tools in schedulable objects, standardizing the configuration of the most common CP tools. This reduces the amount of work analysis framework developers need to do in order to implement CP recommendations.

In this dissertation, a prototype implementation of the RPV1L analysis based on the novel CP algorithms is compared to the original production framework. The RPV1L analysis is a search for supersymmetric particles with R-parity violation in the final state of 1+ or 2 same-sign leptons and 4 to 15+ (b-)jets. A computing performance analysis of the reference and the CP algorithm implementation assesses the current state of CP algorithms.

The RPV1L analysis, discussed in Part II, uses the whole Run 2 dataset to iterate on a previous analysis of the 2015 – 2016 data. It improves the exclusion limits for various R-parity violating benchmark models. Additionally, the analysis reaches sensitivity for direct electroweakino production. The background estimate of misidentified leptons from QCD multi-jet events is discussed in greater detail. A good estimate for the QCD background is extracted, which is essential for a stable fit in the 0-b-jet control region for the purpose of statistical interpretation.

The n-tuple production code of the RPV1L analysis is re-implemented with the novel CP algorithms, as discussed in Part III. Minor discrepancies in a binary comparison are observed without a significant effect on the final object selection. Identical event cutflows and yields for all selected electrons, muons and (b-)jets are achieved. The new CP algorithm implementation applies all recommended object treatments, including overlap removal and E_T^{miss} and scale factor calculations.

A computational performance analysis of the reference and CP algorithm implementation reveals inefficiencies in the RPV1L framework. Resolving these issues increased the execution speed by a factor of 2.5 to 3. Fundamental differences in the job structure between the CP algorithm implementation and the reference are highlighted. The performance of CP algorithms is found to be significantly better than the reference if kinematics-affecting systematic variations are evaluated. The reference only supports a single variation per execution and needs to be rerun for each variation, whereas the CP algorithm implementation evaluates all variations in a single process.

The development and integration of CP algorithms is an ongoing process, but the default sequences provide all essential features to reproduce an analysis based on SUSYTools. During discussions in the collaboration, it became clear that the general structure of CP algorithms is unlikely to change drastically. On the other hand, the configuration layer of CP algorithms is identified to require further work. Sequences are difficult to split into sub-sequences, and the handling of systematic variations via regular expressions is not always easy to grasp. A clear separation between "user-" and "expert-" level configurations could help with a good integration of CP algorithms in existing and future frameworks.

The greatest value of centrally provided CP algorithms lies in the harmonization of analysis frameworks and the formation of a common technical language to prototype and develop future analysis software in ATLAS.

Bibliography

[05] *ATLAS Computing: technical design report.* Technical Design Report ATLAS. Geneva: CERN, 2005. URL: https://cds.cern.ch/record/837738.

[11] *Letter of Intent for the Phase-I Upgrade of the ATLAS Experiment.* CERN-LHCC-2011-012. LHCC-I-020. Geneva: CERN, 2011-11. URL: https://cds.cern.ch/record/1402470.

[15] *ATLAS Phase-II Upgrade Scoping Document.* Tech. rep. CERN-LHCC-2015-020. LHCC-G-166. Geneva: CERN, 2015-09. URL: https://cds.cern.ch/record/2055248.

[17] *Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run.* Tech. rep. ATL-PHYS-PUB-2017-013. Geneva: CERN, 2017-07. URL: http://cds.cern.ch/record/2273281.

[18] *E_T^{miss} performance in the ATLAS detector using 2015-2016 LHC p-p collisions.* Tech. rep. ATLAS-CONF-2018-023. Geneva: CERN, 2018-06. URL: http://cds.cern.ch/record/2625233.

[19a] *Analysis of $t\bar{t}H$ and $t\bar{t}W$ production in multilepton final states with the ATLAS detector.* Tech. rep. ATLAS-CONF-2019-045. Geneva: CERN, 2019-10. URL: https://cds.cern.ch/record/2693930.

[19b] *Luminosity determination in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC.* Tech. rep. ATLAS-CONF-2019-021. Geneva: CERN, 2019-06. URL: http://cds.cern.ch/record/2677054.

[20] *SUSY July 2020 Summary Plot Update.* ATL-PHYS-PUB-2020-020. Geneva: CERN, 2020-07. URL: http://cds.cern.ch/record/2725258.

[Aab+16] Morad Aaboud et al. “Luminosity determination in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector at the LHC”. In: *Eur. Phys. J. C* 76.12 (2016), p. 653. DOI: 10.1140/epjc/s10052-016-4466-1. arXiv: 1608.03953 [hep-ex].

[Aab+17a] M. Aaboud et al. "Jet reconstruction and performance using particle flow with the ATLAS Detector". In: *The European Physical Journal C* 77.7 (2017-07). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5031-2. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5031-2.

[Aab+17b] M. Aaboud et al. "Measurement of jet activity produced in top-quark events with an electron, a muon and two b-tagged jets in the final state in pp collisions at $\sqrt{s}$ = 13 with the ATLAS detector". In: *The European Physical Journal C* 77.4 (2017-04). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-4766-0. URL: http://dx.doi.org/10.1140/epjc/s10052-017-4766-0.

[Aab+17c] Morad Aaboud et al., The ATLAS Collaboration. "Search for new phenomena in a lepton plus high jet multiplicity final state with the ATLAS experiment using $\sqrt{s}$ = 13 TeV proton-proton collision data". In: *JHEP* 09 (2017), p. 088. DOI: 10.1007/JHEP09(2017)088. arXiv: 1704.08493 [hep-ex].

[Aab+18] M. Aaboud et al. "Evidence for the associated production of the Higgs boson and a top quark pair with the ATLAS detector". In: *Physical Review D* 97.7 (2018-04). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.072003. URL: http://dx.doi.org/10.1103/PhysRevD.97.072003.

[Aad+12] Georges Aad et al., The ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett. B* 716.arXiv:1207.7214. CERN-PH-EP-2012-218 (2012-08). Comments: 24 pages plus author list (38 pages total), 12 figures, 7 tables, revised author list, 1–29. 29 p. DOI: 10.1016/j.physletb.2012.08.020. URL: https://cds.cern.ch/record/1471031.

[Aad+13] Georges Aad et al. "Improved luminosity determination in pp collisions at sqrt(s) = 7 TeV using the ATLAS detector at the LHC". In: *Eur. Phys. J. C* 73.8 (2013), p. 2518. DOI: 10.1140/epjc/s10052-013-2518-3. arXiv: 1302.4393 [hep-ex].

[Aad+16a] G. Aad et al. “Muon reconstruction performance of the ATLAS detector in proton–proton collision data at $\sqrt{s} = 13$ TeV”. In: *The European Physical Journal C* 76.5 (2016-05). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4120-y. URL: http://dx.doi.org/10.1140/epjc/s10052-016-4120-y.

[Aad+16b] G. Aad et al. “Performance of pile-up mitigation techniques for jets in pp collisions at $\sqrt{s} = 8$ TeV using the ATLAS detector”. In: *The European Physical Journal C* 76.11 (2016-10). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4395-z. URL: http://dx.doi.org/10.1140/epjc/s10052-016-4395-z.

[Aad+19] G. Aad et al. “ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ TeV”. In: *The European Physical Journal C* 79.11 (2019-11). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7450-8. URL: http://dx.doi.org/10.1140/epjc/s10052-019-7450-8.

[Aad+20] Georges Aad et al. “Evidence for $t\bar{t}t\bar{t}$ production in the multilepton final state in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Eur. Phys. J. C* 80.11 (2020), p. 1085. DOI: 10.1140/epjc/s10052-020-08509-3. arXiv: 2007.14858 [hep-ex].

[Aai+14] Roel Aaij et al. “Observation of the resonant character of the $Z(4430)^-$ state”. In: *Phys. Rev. Lett.* 112.22 (2014), p. 222002. DOI: 10.1103/PhysRevLett.112.222002. arXiv: 1404.1903 [hep-ex].

[Aai+15] Roel Aaij et al. “Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \to J/\psi K^- p$ Decays”. In: *Phys. Rev. Lett.* 115 (2015), p. 072001. DOI: 10.1103/PhysRevLett.115.072001. arXiv: 1507.03414 [hep-ex].

[Abb+18] B. Abbott et al. “Production and integration of the ATLAS Insertable B-Layer”. In: *JINST* 13 (2018), T05008. DOI: 10.1088/1748-0221/13/05/T05008. arXiv: 1803.00844 [physics.ins-det].

[ATL10] ATLAS Collaboration. *ATLAS Insertable B-Layer Technical Design Report.* ATLAS-TDR-19. 2010. URL: https://cds.cern.ch/record/1291633.

[ATL12] Collaboration ATLAS. *Letter of Intent for the Phase-II Upgrade of the ATLAS Experiment.* CERN-LHCC-2012-022. LHCC-I-023. Draft version for comments. Geneva: CERN, 2012-12. URL: https://cds.cern.ch/record/1502664.

[ATL17a] ATLAS Collaboration. “Jet energy scale measurements and their systematic uncertainties in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Rev. D* 96 (2017), p. 072002. DOI: 10.1103/PhysRevD.96.072002. arXiv: 1703.09665 [hep-ex].

[ATL17b] ATLAS Collaboration. “Performance of the ATLAS track reconstruction algorithms in dense environments in LHC Run 2”. In: *Eur. Phys. J. C* 77 (2017), p. 673. DOI: 10.1140/epjc/s10052-017-5225-7. arXiv: 1704.07983 [hep-ex].

[ATL17c] ATLAS Collaboration. “Performance of the ATLAS trigger system in 2015”. In: *Eur. Phys. J. C* 77 (2017), p. 317. DOI: 10.1140/epjc/s10052-017-4852-3. arXiv: 1611.09661 [hep-ex].

[ATL17d] ATLAS Collaboration. “Reconstruction of primary vertices at the ATLAS experiment in Run 1 proton–proton collisions at the LHC”. In: *Eur. Phys. J. C* 77 (2017), p. 332. DOI: 10.1140/epjc/s10052-017-4887-5. arXiv: 1611.10235 [hep-ex].

[ATL19a] ATLAS Collaboration. “Electron and photon energy calibration with the ATLAS detector using 2015–2016 LHC proton–proton collision data”. In: *JINST* 14 (2019), P03017. DOI: 10.1088/1748-0221/14/03/P03017. arXiv: 1812.03848 [hep-ex].

[ATL19b] ATLAS Collaboration. “Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton–proton collision data at $\sqrt{s} = 13$ TeV”. In: *Eur. Phys. J. C* 79 (2019), p. 639. DOI: 10.1140/epjc/s10052-019-7140-6. arXiv: 1902.04655 [hep-ex].

[Bar+05] R. Barbier et al. “R-Parity-violating supersymmetry”. In: *Physics Reports* 420.1-6 (2005-11), pp. 1–195. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2005.08.006. URL: http://dx.doi.org/10.1016/j.physrep.2005.08.006.

[Beh+13] Olaf Behnke et al., eds. *Data analysis in high energy physics: A practical guide to statistical methods.* Weinheim, Germany: Wiley-VCH, 2013. ISBN: 978-3-527-41058-3, 978-3-527-65344-7, 978-3-527-65343-0.

[Ber+89] F.A. Berends et al. “Multijet production in W, Z events at pp colliders”. In: *Physics Letters B* 224.1 (1989), pp. 237–242. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(89)91081-2. URL: https://www.sciencedirect.com/science/article/pii/0370269389910812.

[Bir+14] I Bird et al. *Update of the Computing Models of the WLCG and the LHC Experiments*. CERN-LHCC-2014-014. LCG-TDR-002. 2014-04. URL: https://cds.cern.ch/record/1695401.

[Blo+20] Jakob Blomer et al. *The CernVM File System: v2.7.5*. Version 2.7.5. 2020-10. DOI: 10.5281/zenodo.4114078. URL: https://doi.org/10.5281/zenodo.4114078.

[Bru+19] Rene Brun et al. *root-project/root: v6.18/02*. Version v6-18-02. 2019-08. DOI: 10.5281/zenodo.3895860. URL: https://doi.org/10.5281/zenodo.3895860.

[Buc+15] Andrew Buckley et al. *Implementation of the ATLAS Run 2 event data model*. ATL-SOFT-PROC-2015-003. 7. Geneva: CERN, 2015-05. DOI: 10.1088/1742-6596/664/7/072045. URL: https://cds.cern.ch/record/2014150.

[Cal+20] P Calafiura et al. *ATLAS HL-LHC Computing Conceptual Design Report*. Tech. rep. CERN-LHCC-2020-015. LHCC-G-178. Geneva: CERN, 2020-09. URL: https://cds.cern.ch/record/2729668.

[Cat+15] James Catmore et al. *A New Petabyte-scale Data Derivation Framework for ATLAS*. ATL-SOFT-PROC-2015-041. 7. Geneva: CERN, 2015-05. DOI: 10.1088/1742-6596/664/7/072007. URL: https://cds.cern.ch/record/2016628.

[CGH12] Csaba Csáki, Yuval Grossman, and Ben Heidenreich. “Minimal flavor violation supersymmetry: A natural theory forR-parity violation”. In: *Physical Review D* 85.9 (2012-05). ISSN: 1550-2368. DOI: 10.1103/physrevd.85.095009. URL: http://dx.doi.org/10.1103/PhysRevD.85.095009.

[Cho+08] S.K. Choi et al. “Observation of a resonance-like structure in the $pi^{\pm}\psi'$ mass distribution in exclusive $B \longrightarrow K\pi^{\pm}\psi'$ decays”. In: *Phys. Rev. Lett.* 100 (2008). Ed. by Dongchul Son and Sun Kun Oh, p. 142001. DOI: 10.1103/PhysRevLett.100.142001. arXiv: 0708.1790 [hep-ex].

[Cid+19] Xabier Cid Vidal et al. "Report from Working Group 3: Beyond the Standard Model physics at the HL-LHC and HE-LHC". In: *Report on the Physics at the HL-LHC,and Perspectives for the HE-LHC*. Ed. by Andrea Dainese et al. Vol. 7. 2019-12, pp. 585–865. DOI: 10.23731/CYRM-2019-007.585. arXiv: 1812.07831 [hep-ph].

[Cola] ATLAS Collaboration. *Expected performance of the 2019 ATLAS b-taggers*. URL: http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/FTAG-2019-005/ (visited on 2021-01-27).

[Colb] The ATLAS Collaboration. *Athena, 21.2, Gitlab*. Commit hash: 277d4cea6ff8807d0de1f1faf2f927901fad756d. URL: https://gitlab.cern.ch/atlas/athena/-/tree/21.2/PhysicsAnalysis/Algorithms (visited on 2020-08-24).

[Colc] The ATLAS Collaboration. *The Collaboration*. URL: http://atlas.cern/discover/collaboration (visited on 2020-11-28).

[Col08] The ATLAS Collaboration. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.

[Col15] ATLAS Collaboration. "ATLAS event at 13 TeV - Zee candidate - run: 265573, lb: 18, event: 970468". General Photo. 2015-05. URL: https://cds.cern.ch/record/2019370.

[Col19] The ATLAS Collaboration. *Athena*. Version 21.2.129. 2019-04. DOI: 10.5281/zenodo.3932810. URL: https://doi.org/10.5281/zenodo.3932810.

[Col20] The ATLAS Collaboration. *Impact of ROOT file parameters on ATLAS Analysis Object Data*. Tech. rep. ATL-SOFT-PUB-2020-002. Geneva: CERN, 2020-07. URL: https://cds.cern.ch/record/2723199.

[Col21] ATLAS Collaboration. "Search for R-parity violating supersymmetry in a final state containing leptons and many jets with the ATLAS experiment using $\sqrt{s}$ = 13 TeV proton-proton collision data". In: (2021). arXiv: 2106.09609 [hep-ex].

[Con68] Melvin E. Conway. *How Do Committees Invent?* 1968. URL: http://www.melconway.com/Home/Committees_Paper.html (visited on 2020-11-28).

[Cora] Intel Corporation. *Hardware Event-based Sampling Collection.* URL: https://software.intel.com/content/www/us/en/develop/documentation/vtune-help/top/analyze-performance/hardware-event-based-sampling-collection.html (visited on 2020-12-14).

[Corb] Intel Corporation. *Instructions Retired Event.* URL: https://software.intel.com/content/www/us/en/develop/documentation/vtune-help/top/analyze-performance/custom-analysis/custom-analysis-options/hardware-event-list/instructions-retired-event.html (visited on 2021-03-01).

[Corc] Intel Corporation. *Intel® VTune™ Profiler Product Page.* URL: https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/vtune-profiler.html (visited on 2020-12-14).

[Cor20] Intel Corporation. *Intel 64 and IA-32, Architectures Software Developer's Manual.* Combined Volumes:1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D and 4. 2020-11. URL: https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html.

[CSS08] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "The anti-k_t jet clustering algorithm". In: *JHEP* 04 (2008), p. 063. DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189 [hep-ph].

[Dre14] Eric Drexler. *Elementary Particle Interactions in the Standard Model.* Image File. 2014. URL: https://en.wikipedia.org/wiki/File:Elementary_particle_interactions_in_the_Standard_Model.png (visited on 2020-07-15).

[EB08] Lyndon Evans and Philip Bryant. "LHC Machine". In: *JINST* 3 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[EKS85] S.D. Ellis, R. Kleiss, and W.J. Stirling. "W's, Z's and jets". In: *Physics Letters B* 154.5 (1985), pp. 435–440. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(85)90425-3. URL: https://www.sciencedirect.com/science/article/pii/0370269385904253.

[Elm+20] Johannes Elmsheuser et al. “Evolution of the ATLAS analysis model for Run-3 and prospects for HL-LHC”. In: *EPJ Web Conf.* 245 (2020), p. 06014. DOI: 10.1051/epjconf/202024506014.

[Err16] Martin Errenst. “Performance Optimization of the ATLAS Detector Simulation”. Presented 2016-09-05. 2016-09. URL: https://cds.cern.ch/record/2227212.

[Expa] ATLAS Experiment. *$Z \rightarrow \mu\mu$ candidate event with 65 additional reconstructed primary vertices recorded in 2017.* URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/EventDisplayRun2Physics#High_pileup_displays (visited on 2021-01-04).

[Expb] ATLAS Experiment. *Number of Interactions per Crossing – 2018 pp Collisions.* URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2#Pileup_Interactions_and_Data_Tak (visited on 2021-01-04).

[FAR+15] Steven FARRELL et al. *Dual-use tools and systematics-aware analysis workflows in the ATLAS Run-2 analysis model.* ATL-SOFT-PROC-2015-019. 3. Geneva: CERN, 2015-05. DOI: 10.1088/1742-6596/664/3/032007. URL: https://cds.cern.ch/record/2015229.

[Fay76] Pierre Fayet. “Supersymmetry and Weak, Electromagnetic and Strong Interactions”. In: *Phys. Lett. B* 64 (1976), p. 159. DOI: 10.1016/0370-2693(76)90319-1.

[Fay77] Pierre Fayet. “Spontaneously Broken Supersymmetric Theories of Weak, Electromagnetic and Strong Interactions”. In: *Phys. Lett. B* 69 (1977), p. 489. DOI: 10.1016/0370-2693(77)90852-8.

[FF78] Glennys R. Farrar and Pierre Fayet. “Phenomenology of the production, decay, and detection of new hadronic states associated with supersymmetry”. In: *Phys. Lett. B* 76 (1978), p. 575. DOI: 10.1016/0370-2693(78)90858-4.

[FZ74] S. Ferrara and B. Zumino. “Supergauge invariant Yang-Mills theories”. In: *Nucl. Phys. B* 79 (1974), p. 413. DOI: 10.1016/0550-3213(74)90559-8.

[GB16] David Galbraith and Carsten Burgard. *Standard model of physics.* TikZ Example. 2016. URL: http://www.texample.net/tikz/examples/model-physics/ (visited on 2020-07-15).

[Ger+12] Erik Gerwick et al. “Scaling patterns for QCD jets”. In: *Journal of High Energy Physics* 2012.10 (2012-10). DOI: 10.1007/jhep10(2012)162. URL: https://doi.org/10.1007/jhep10(2012)162.

[GL71] Y.A. Golfand and E.P. Likhtman. “Extension of the Algebra of Poincare Group Generators and Violation of P Invariance”. In: *JETP Lett.* 13 (1971). [Pisma Zh. Eksp. Teor. Fiz. **13** (1971) 452], p. 323.

[Gri87] David J Griffiths. *Introduction to elementary particles*. TextBook Physics. New York, NY: Wiley, 1987. DOI: 10.1002/9783527618460. URL: https://cds.cern.ch/record/1260972.

[GS71] J.-L. Gervais and B. Sakita. “Field theory interpretation of supergauges in dual models”. In: *Nuclear Physics B* 34.2 (1971), pp. 632–639. ISSN: 0550-3213. DOI: https://doi.org/10.1016/0550-3213(71)90351-8. URL: http://www.sciencedirect.com/science/article/pii/0550321371903518.

[Har+15] L. A. Harland-Lang et al. “Parton distributions in the LHC era: MMHT 2014 PDFs”. In: *The European Physical Journal C* 75.5 (2015-05). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-015-3397-6. URL: http://dx.doi.org/10.1140/epjc/s10052-015-3397-6.

[Jab07] JabberWok. *Pseudorapidity2*. Image File. 2007. URL: https://en.wikipedia.org/wiki/File:Pseudorapidity2.png (visited on 2020-08-19).

[LM20] Daniel E. López-Fogliani and Carlos Muñoz. “Searching for supersymmetry: the $\mu\nu$SSM”. In: *The European Physical Journal Special Topics* 229.21 (2020-12), pp. 3263–3301. ISSN: 1951-6401. DOI: 10.1140/epjst/e2020-000114-9. URL: http://dx.doi.org/10.1140/epjst/e2020-000114-9.

[Man10] Robert Mann. *An introduction to particle physics and the standard model*. Boca Raton, FL: CRC Press, 2010. URL: https://cds.cern.ch/record/1225889.

[Mar10] Stephen P. Martin. “A Supersymmetry primer”. In: *Perspectives on supersymmetry. Vol.2*. Ed. by Gordon L. Kane. Vol. 21. 2010, pp. 1–153. DOI: 10.1142/9789812839657_0001. arXiv: hep-ph/9709356.

[MP14] I. Melzer-Pellmann and P. Pralavorio. “Lessons for SUSY from the LHC after the first run”. In: *Eur. Phys. J. C* 74 (2014), p. 2801. DOI: 10.1140/epjc/s10052-014-2801-y. arXiv: 1404.7191 [hep-ex].

[Per87] Donald Hill Perkins. *Introduction to high energy physics; 3rd ed.* Menlo Park, CA: Addison-Wesley, 1987. URL: https://cds.cern.ch/record/108804.

[Pro] HiLumi HL-LHC Project. *Project Schedule.* URL: https://project-hl-lhc-industry.web.cern.ch/content/project-schedule (visited on 2020-12-05).

[RM] Rik van Riel and Peter W. Morreale. *Documentation for /proc/sys/vm/*.* URL: https://www.kernel.org/doc/Documentation/sysctl/vm.txt (visited on 2020-11-24).

[Sofa] ATLAS Offline Software. *ATLAS Software Documentation: Configuring Systematics Algorithm Sequences.* Likely to change regularly. Corresponding git repository located at https://gitlab.cern.ch/atlas-sw-git/atlassoftwaredocs (HEAD: 2aeb4d653034d66cc09131f53f580f636cf3342d). URL: https://atlassoftwaredocs.web.cern.ch/AnalysisTools/ana_alg_sys_config/ (visited on 2020-08-24).

[Sofb] ATLAS Offline Software. *Common CP Algorithms Outlook.* Likely to change regularly. Corresponding git repository located at https://gitlab.cern.ch/atlas-sw-git/atlassoftwaredocs (HEAD: 758ee34bd089125058edf8da15fe8aa029344a9d). URL: https://atlassoftwaredocs.web.cern.ch/ABtutorial/cpalg_outlook/ (visited on 2020-11-18).

[SS74] Abdus Salam and J. Strathdee. “Super-symmetry and non-Abelian gauges”. In: *Phys. Lett. B* 51 (1974), p. 353. DOI: 10.1016/0370-2693(74)90226-3.

[Tan+18] M. Tanabashi et al., Particle Data Group. “Review of Particle Physics”. In: *Phys. Rev. D* 98 (3 2018-08), p. 030001. DOI: 10.1103/PhysRevD.98.030001. URL: https://link.aps.org/doi/10.1103/PhysRevD.98.030001.

[VA73] D.V. Volkov and V.P. Akulov. “Is the neutrino a goldstone particle?” In: *Phys. Lett. B* 46 (1973), p. 109. DOI: 10.1016/0370-2693(73)90490-5.

[WZ74a] J. Wess and B. Zumino. "Supergauge invariant extension of quantum electrodynamics". In: *Nucl. Phys. B* 78 (1974), p. 1. DOI: 10.1016/0550-3213(74)90112-6.

[WZ74b] J. Wess and B. Zumino. "Supergauge transformations in four dimensions". In: *Nucl. Phys. B* 70 (1974), p. 39. DOI: 10.1016/0550-3213(74)90355-1.

I hereby declare that

1. I have written the submitted dissertation with the title “Application of Novel Analysis Workflows in an ATLAS Search for Supersymmetry” independently by myself,

2. using only the sources and aids documented therein. I have marked all passages taken over verbatim or in content as such, indicating the source.

3. I have not submitted this dissertation in the current or another version to a different department or at another university.

4. I agree with the presence of audience members who are not members of the examination board.

Wuppertal,

Martin V. F. Errenst